OPERATOR'S MANUAL

CR23X Micrologger

Revision: 11/06

®

Warranty and Assistance

The **CR23X MICROLOGGER** is warranted by CAMPBELL SCIENTIFIC, INC. to be free from defects in materials and workmanship under normal use and service for thirty-six (36) months from date of shipment unless specified otherwise. Batteries have no warranty. CAMPBELL SCIENTIFIC, INC.'s obligation under this warranty is limited to repairing or replacing (at CAMPBELL SCIENTIFIC, INC.'s option) defective products. The customer shall assume all costs of removing, reinstalling, and shipping defective products to CAMPBELL SCIENTIFIC, INC. CAMPBELL SCIENTIFIC, INC. will return such products by surface carrier prepaid. This warranty shall not apply to any CAMPBELL SCIENTIFIC, INC. products which have been subjected to modification, misuse, neglect, accidents of nature, or shipping damage. This warranty is in lieu of all other warranties, expressed or implied, including warranties of merchantability or fitness for a particular purpose. CAMPBELL SCIENTIFIC, INC. is not liable for special, indirect, incidental, or consequential damages.

Products may not be returned without prior authorization. The following contact information is for US and International customers residing in countries served by Campbell Scientific, Inc. directly. Affiliate companies handle repairs for customers within their territories. Please visit www.campbellsci.com to determine which Campbell Scientific company serves your country. To obtain a Returned Materials Authorization (RMA), contact CAMPBELL SCIENTIFIC, INC., phone (435) 753-2342. After an applications engineer determines the nature of the problem, an RMA number will be issued. Please write this number clearly on the outside of the shipping container. CAMPBELL SCIENTIFIC's shipping address is:

CAMPBELL SCIENTIFIC, INC.
RMA#_____
815 West 1800 North
Logan, Utah 84321-1784

CAMPBELL SCIENTIFIC, INC. does not accept collect calls.

CR23X MEASUREMENT AND CONTROL MODULE
TABLE OF CONTENTS

PDF viewers note: These page numbers refer to the printed version of this document. Use the Adobe Acrobat® bookmarks tab for links to specific sections.

PROGRAMMING

1. FUNCTIONAL MODES

i

DATA RETRIEVAL/COMMUNICATION

PROGRAM EXAMPLES

8.   PROCESSING AND PROGRAM CONTROL EXAMPLES

INSTRUCTIONS

9.   INPUT/OUTPUT INSTRUCTIONS

10.   PROCESSING INSTRUCTIONS

11.   OUTPUT PROCESSING INSTRUCTIONS

12.   PROGRAM CONTROL INSTRUCTIONS

MEASUREMENTS

13.   CR23X MEASUREMENTS

INSTALLATION

14. INSTALLATION AND MAINTENANCE

APPENDICES

A. GLOSSARY

B. CONTROL PORT SERIAL I/O INSTRUCTION 15

C. ADDITIONAL TELECOMMUNICATIONS INFORMATION

E. ASCII TABLE

F. DYNAGAGE SAP-FLOW (P67)

G. CALLBACK (CR23X INITIATED TELECOMMUNICATIONS)

H. CALL ANOTHER DATALOGGER VIA PHONE OR RF

I. TD OPERATING SYSTEM ADDENDUM FOR CR510, CR10X, AND CR23X MANUALS

SELECTED OPERATING DETAILS

1. **Storing Data** - Data are stored in Final Storage only by Output Processing Instructions and only when the Output Flag (Flag 0) is set. (Sections OV4.1.1 and 3.7.1)

2. **Storing Date and Time** - Date and time are stored with the data in Final Storage ONLY if the Real Time Instruction 77 is used. (Section 11)

3. **Data Transfer** - On-line data transfer from Final Storage to peripherals (printer, Storage Module, etc.) occurs only if enabled with Instruction 96 in the datalogger program. (Sections 4 and 12)

4. **Final Storage Resolution** - All Input Storage values are displayed ($\boxed{*}$ $\boxed{6}$ mode) as high resolution with a maximum value of 99999. However, the default resolution for data stored in Final Storage is low resolution, maximum value of 6999. Results exceeding 6999 are stored as 6999 unless Instruction 78 is used to store the values in Final Storage as high resolution values. (Sections 2.2.1 and 11)

5. **Floating Point Format** - The computations performed in the CR23X use floating point arithmetic. CSI's 4 byte floating point numbers contain a 23 bit binary mantissa and a 6 bit binary exponent. The largest and smallest numbers that can be stored and processed are 9×10^{18} and 1×10^{-19}, respectively. (Section 2.2.2)

6. **Erasing Final Storage** - Data in Final Storage can be erased without altering the program by using the $\boxed{*}$ $\boxed{A}$ Mode to repartition memory. (Section 1.5.2)

7. **ALL memory can be erased** and the CR23X completely reset by entering 98765 for the number of bytes allocated to Program Memory. ($\boxed{*}$ $\boxed{A}$ Window 5, Section 1.5.2)

CAUTIONARY NOTES

1. Damage will occur to the analog input circuitry if voltages in excess of ±16 V are applied for a sustained period. Voltages in excess of ±8 V will cause errors and possible overranging on other analog input channels.

2. Do not download an operating system (OS) written for a particular datalogger model into the hardware of another datalogger model. The datalogger will sustain damage and must be returned to the factory for repair. This is of concern only when updated operating systems are purchased from Campbell Scientific.

3. When using the CR23X with the rechargeable battery option, remember that the sealed lead acid batteries are permanently damaged if deep discharged. The cells are rated at a 7 Ahr capacity but experience a slow discharge even in storage. It is advisable to maintain a continuous charge on the battery, whether in operation or storage (Section 14).

4. When connecting external power to the CR23X, first, remove the green power connector from the CR23X panel. Then insert the positive 12 V lead into the right-most terminal of the green connector. Next, insert the ground lead to the left terminal. Double check polarity before plugging the green connector into the panel.

5. Voltages in excess of 5 volts should not be applied to a control port.

6. The CR23X contains desiccant to protect against excess humidity. To reduce vapor transfer into the ENC 12/14 or ENC 16/18 enclosure, plug the cable entry conduit with Duct Seal, a putty-type sealant available at most electrical supply houses. DO NOT totally seal enclosures equipped with lead acid batteries. Hydrogen concentration may build up to explosive levels.

CR23X MICROLOGGER OVERVIEW

Read the Selected Operating Details and Cautionary Notes at the front of the Manual before using the CR23X.

The CR23X Micrologger combines precision measurement with processing and control capability in a single battery operated system.

Campbell Scientific, Inc. provides three documents to aid in understanding and operating the CR23X:

1. *This Overview*
2. *The CR23X Operator's Manual*
3. *The CR23X Prompt Sheet*

This Overview introduces the concepts required to take advantage of the CR23X's capabilities. Hands-on programming examples start in Section OV4. Working with a CR23X will help the learning process, so don't just read the examples, turn on the CR23X and do them. If you want to start this minute, go ahead and try the examples, then come back and read the rest of the Overview.

- *The sections of the Operator's Manual which should be read to complete a basic understanding of the CR23X operation are the Programming Sections 1-3, the portions of the data retrieval Sections 4 and 5 appropriate to the method(s) you are using (see OV5), and Section 14 which covers installation and maintenance.*

- *Section 6 covers details of serial communications. Sections 7 and 8 contain programming examples. Sections 9-12 have detailed descriptions of the programming instructions, and Section 13 goes into detail on the CR23X measurement procedures.*

The Prompt Sheet is an abbreviated description of the programming instructions. Once familiar with the CR23X, it is possible to program it using only the Prompt Sheet and on-line prompts as a reference, consulting the manual if further detail is needed.

OV1. PHYSICAL DESCRIPTION

The CR23X Micrologger with the alkaline batteries is shown in Figure OV1-1. It is powered with 10 "D" cells and has only the power switch on the base. The rechargeable CR23X has rechargeable lead acid cells. In addition to the power switch, it has a charger input plug and an LED which lights when the charging circuit is active. Rechargeable CR23Xs should always be connected to a solar panel or AC charger. The lead acid batteries provide backup in event of a power failure but are permanently damaged if their voltage drops below 11.76 volts. Campbell Scientific does not warrant batteries.

The 16 character keyboard is used to enter programs, commands and data; these can be viewed on the 24 character x 2 line LCD display.
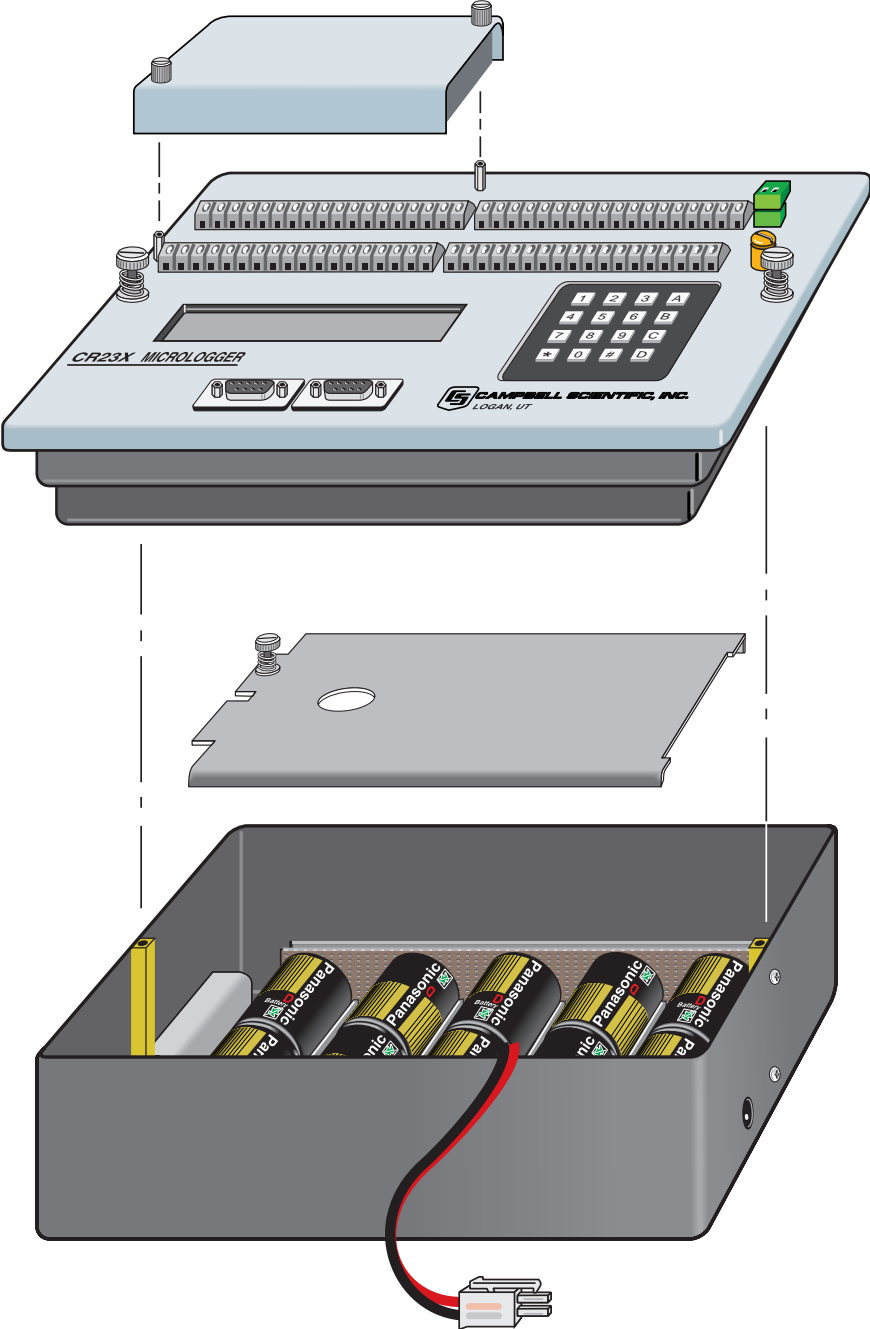
FIGURE OV1-1. CR23X Micrologger

ANALOG INPUTS

Input/Output Instructions
1 Volt (SE)
2 Volt (DIFF)
4 Ex-Del-Se
5 AC Half Br
6 Full Br
7 3W Half Br
8 Ex-Del-Diff
9 6W Full Br
11 Temp (107)
12 RH-(207)
13 Temp-TC SE
14 Temp-TC DIFF
16 Temp-RTD
27 Interval-Freq.
28 Vibrating Wire Meas
29 INW Press
131 Enhanced Vib. Wire

PULSE INPUTS
Input/Output Instructions
3 Pulse

Continuous
Analog Outputs
133 Analog

EXCITATION OUTPUTS

Input/Output Instructions
4 Ex-Del-Se
5 AC Half Br
6 Full Br
7 3W Half Br
8 Ex-Del-Diff
9 Full Br-Mex
11 Temp (107)
12 RH (207)
22 Del w/Opt Ext
28 Wire Meas
29 INW Press

DIGITAL I/O PORTS

Input/Output Instructions
3 Pulse
15 Serial I/O
20 Set Ports
21 Pulse Port
25 Read Ports
100-110, 118 SDM and SDI12
Instructions
134 AM25T

Program Control Instructions
83 If Case < F
86 Do
88 If X <=> Y
89 If X <=> F
91 If flag, port
92 If Time

Command Codes:
4X Set port x high
49 Switched 12 V on
5X Set port x low
59 Switched 12 V off
6X Toggle port x
7X Pulse port x
95 Port Subr.
96 Port Subr.
97 Port Subr.
98 Port Subr.

Signal Ground (⏚), for
Analog
Pulse
Excitation
Sensor Shields

External
12 Volt
Power Input

Power Ground (G), for
5V
SW-12
12V
Control I/O

Earth Ground
Connect 12ga
or larger wire to
earth ground

04:REF_TEMP
+21.93

CR23X MICROLOGGER

CAMPBELL SCIENTIFIC, INC.
Logan, UT

SN:

MADE IN USA

CS I/O

COMPUTER
RS232
(OPTICALLY ISOLATED)

SERIAL I/O
Telecommunications
Program Control Instructions
96 Storage Module, Printer, Serial Out
97 Initiate Telecommunications
120 TGT1 GOES Satellite (CS I/O only)
121 ARGOS Satellite (CS I/O only)
122 INMARSAT-C Satellite (CS I/O only)
123 TGT1 Programming

Switched
12 Volts

FIGURE OV1-2. CR23X Panel and Associated Programming Instructions

CR23X MICROLOGGER OVERVIEW

The 9-pin serial CS I/O port provides connection to data storage peripherals, such as the SM192/716 Storage Module, and provides serial communication to computer or modem devices for data transfer or remote programming (Section 6). *This 9 pin port does NOT have the same pin configuration as the 9 pin serial ports currently used on most personal computers.* An SC32A is required to interface the CS I/O port to a PC or other RS-232 serial port (Section 6). An optically isolated computer RS-232 port is also provided for direct connection to PCs and other RS-232 devices.

The panel contains four terminal strips which are used for sensor inputs, excitation, control input/outputs, etc. Figure OV1-2 shows the CR23X panel and the associated programming instructions.

OV1.1 WIRING TERMINALS

Wiring terminals are provided on the CR23X to allow connection of external sensors and other devices.

OV1.1.1 ANALOG INPUTS

The terminals labeled 1H to 12L are analog voltage inputs. These numbers (black) refer to the high and low inputs to the differential channels 1 through 12. In a differential measurement, the voltage on the H input is measured with respect to the voltage on the L input. When making single-ended measurements, either the H or L input may be used as an independent channel to measure voltage with respect to the CR23X analog ground ($\doteq$). The single-ended channels are numbered sequentially starting with 1H (blue); e.g., the H and L sides of differential channel 1 are single-ended channels 1 and 2; the H and L sides of differential channel 2 are single-ended channels 3 and 4, etc.

The analog input terminal strips have an insulated cover to reduce temperature gradients across the input terminals. The cover is required for accurate thermocouple measurements (Section 13.4).

OV1.1.2 EXCITATION OUTPUTS

The terminals labeled EX1, EX2, EX3, and EX4 are precision, switched excitation outputs used to supply programmable excitation voltages for resistive bridge measurements. DC or AC excitation at voltages between -5000 mV and +5000 mV are user programmable (Section 9).

OV1.1.3 CONTINUOUS ANALOG OUTPUTS (CAO)

Two CAO channels supply continuous output voltages under program control, for use with strip charts, x-y plotters, or proportional controllers.

OV1.1.4 PULSE INPUTS

The terminals labeled P1, P2, P3, and P4 are the pulse counter inputs for the CR23X. They are programmable for high frequency pulse, low level AC, or switch closure (Section 9, Instruction 3).

OV1.1.5 DIGITAL I/O PORTS

Terminals C1 through C8 are digital Input/Output ports. On power-up they are configured as input ports, commonly used for reading the status of an external signal. High and low conditions are: 3 V < high < 5.5 V; -0.5 V < low < 0.8 V.

Configured as outputs the ports allow on/off control of external devices. A port can be set high (5 V $\pm$ 0.1 V), low (<0.1 V), toggled or pulsed (Sections 3, 8.3, and 12).

Ports C5 through C8 can be configured as pulse counters for switch closures (Section 9, Instruction 3) or used to trigger subroutine execution (Section 1.1.2).

Built in Zener diodes on the eight control ports limit input voltage to acceptable levels of < = 5.6 VDC. Do not apply voltages greater than 16 VDC. A voltage of 5.0 VDC is preferred.

OV1.1.6 GROUNDS

The CR23X has ground terminals marked $\doteq$ and G. Signal returns of analog inputs and their associated shields along with excitation voltage returns are to be tied to the $\doteq$ terminals located in the analog input terminal strips. The G terminals (Power Grounds) are intended to carry return currents from the 5 V, SW12, 12 V, and C1-C8 outputs. Tying these potentially large return currents to G terminals keeps these currents from flowing through and corrupting analog measurements. Offset voltage errors in single-ended measurements can occur for large (50 mA) currents flowing into the $\doteq$ terminals in the analog input terminal strips.

Return currents from the CAO and pulse-counter channels should be tied to the ⏚ terminals in the CAO and pulse-counter terminal strip to prevent them from flowing through the analog measurement section.

The ground lug is also marked ⏚ and provides a rugged ground path from the individual ⏚ and G terminals to earth or chassis ground for ESD protection.

Review Section 14.7 for complete grounding recommendations.

OV1.1.8 5V OUTPUTS

The 5 V (±4.0%) output is commonly used to power peripherals such as the QD1 Incremental Encoder Interface, AVW1 or AVW4 Vibrating Wire Interface.

The 5 V output is common with pin 1 on the CS I/O 9 pin connector; 200 mA is the maximum combined output.

OV1.1.9 CS I/O

The 9 pin CS I/O port contains lines for serial communication between the CR23X and external devices such as computers, printers, Campbell modems, Storage Modules, etc. **This port does NOT have the same configuration as the 9 pin serial ports currently used on most personal computers**. It has a 5 VDC power line which is used to power peripherals such as Storage Modules. The same 5 VDC supply is used for the 5 V output on the lower right terminal strip. It has a 12 VDC power line used to power other peripherals such as the COM200 phone modem. Section 6 contains technical details on serial communication.

OV1.1.10 COMPUTER RS-232 PORT

This port is an optically isolated standard 9 pin RS-232 DCE/DTE port. It can be connected directly to the serial port of most personal computers. A 6 foot 9 to 9 pin serial cable and a 9 to 25 pin adapter are included with the CR23X to connect this port to a PC serial port.

OV1.1.11 SWITCHED 12 VOLT

The switched 12 volt output can be used to power sensors or devices requiring an unregulated 12 volts. The output is limited to 600 mA at 50°C (360 mA at 80°C) current. The switched 12 volt port is addressed as "Port 9" in a datalogger program.

When the port is set high, the 12 volts is turned on; when the port is low, the switched 12 volts is off (Section 8.12).

OV1.2 CONNECTING POWER TO THE CR23X

The CR23X should be powered by any clean, battery backed 12 VDC source. The green power connector on the wiring panel is a plug in connector that allows the power supply to be easily disconnected. The power connection is reverse polarity protected. The datalogger should be earth or chassis ground during routine operation. See Section 14 for details on power supply connections and grounding.

When primary power falls below 11.0 VDC, the CR23X stops executing its programs. The Low Voltage Counter (∗B window 9) is incremented by one each time the primary power falls below 11.0 VDC and E10 is displayed. A double dash (--) in the 9th window of the ∗B mode indicates that the CR23X is currently in a low primary power mode. (Section 1.6)

The datalogger program and stored data remain in memory, and the clock continues to keep time when power is disconnected. The clock and SRAM are powered by an internal lithium battery. (Section 14.11.2)

OV2. MEMORY AND PROGRAMMING CONCEPTS

OV2.1 INTERNAL MEMORY

The standard CR23X has 512 Kilobytes of Flash Electrically Erasable Programmable Read Only Memory (EEPROM), 128 Kilobytes Static Random Access Memory (SRAM), and 1 Megabyte of Flash RAM. As an option, the CR23X can be purchased with 4 Megabyte Flash for final storage. Operating system EEPROM stores the operating system, user programs, and labels. SRAM is used for final storage data and running the user program. Final Storage Flash is used for data storage. The use of the Input, Intermediate, and Final Storage in the measurement and data processing sequence is shown in Figure OV2.1-2. The five areas of SRAM are:

1. **System Memory** - used for overhead tasks such as compiling programs, transferring data, etc. The user cannot access this memory.

2. **Active Program Memory** - available for user entered programs.

3. **Input Storage** - Input Storage holds the results of measurements or calculations. The ⌧ ⌧ Mode is used to view Input Storage locations for checking current sensor readings or calculated values. Input Storage defaults to 64 locations. Additional locations can be assigned using the ⌧ ⌧ Mode.

4. **Intermediate Storage** - Certain Processing Instructions and most of the Output Processing Instructions maintain intermediate results in Intermediate Storage. Intermediate storage is automatically accessed by the instructions and cannot be accessed by the user. The default allocation is 64 locations. The number of locations can be changed using the ⌧ ⌧ Mode.

5. **Final Storage** - Final processed values are stored here for transfer to printer, solid state Storage Module or for retrieval via telecommunication links. Values are stored in Final Storage only by the Output Processing Instructions and only when the Output Flag is set in the user's program. Approximately 570,000 locations are allocated to Final Storage from SRAM on power up. This number is reduced if Input or Intermediate Storage is increased.

   While the total size of these three areas remains constant, memory may be reallocated between the areas to accommodate different measurement and processing needs (⌧ ⌧ Mode, Section 1.5).

6. **Alphanumeric Labels** - The CR23X can be programmed through EDLOG (PC208W software) to assign alphanumeric labels to Input Storage and Final Storage locations. Labels must consist of letters, numbers, or the underscore (_), and must not begin with a number.

**Flash Memory
(EEPROM)
Total 512 Kbytes**

Operating System
(128 Kbytes)

Active Program
(32 Kbytes Code)

Stored Programs
(32 Kbytes Code)
(32 Kbytes Labels)

**Temporary Copy of
Current Program**
Saved during
download if download
is aborted
(64 Kbytes)

Alphanumeric Labels
(32 Kbytes)

Unassigned
(192 Kbytes)

How it works:
The **Operating System** is loaded into Flash Memory at the factory. **System Memory** is used while the CR23X is running for calculations, buffering data and general operating tasks.

Any time a user loads a program into the CR23X, the program is compiled in SRAM and stored in the **Active Program** areas. If the CR23X is powered off and then on, the Active Program is loaded from Flash and run.

The Active Program is run in SRAM to maximize speed. The program accesses **Input Storage** and **Intermediate Storage** and stores data into **Final Storage** for later retrieval by the user.

The Active Program can be copied into the **Stored Programs** area. While 98 program "names" are available, the number of programs stored is limited by the available memory. Stored programs can be retrieved to become the active program. While programs are stored one at a time, all stored programs are erased simultaneously. That is because the flash memory can only be written to once before it must be erased and can only be erased in 16 Kbytes blocks.

(Memory Areas separated by dashed lines: ▬ ▬ ▬ ▬ ▬ ▬ can be re-sized by the user.)

1 byte per character stored. 9 bytes per input location label. All final storage label characters plus 2 bytes per table name (array ID name) and field name.

**SRAM/FLASH
Total 1152 Kbytes**

32K SRAM

System Memory
4096 Bytes

Active Program
Default
2048 Bytes

Input Storage
Default
112 Bytes
28 Locations

Intermediate Storage
Default
256 Bytes
64 Locations

96K SRAM

Final Storage 1 and 2
98,304 Bytes
49,154 Locations

1M FLASH

Final Storage 1 and 2
917,504 Bytes
458,752 Locations
or

4M FLASH

Final Storage 1 and 2
4,292,610 Bytes
2,146,305 Locations

Final Storage 1 Only
131,072 Bytes
65,536 Locations

Memory available only to system

Memory shared between Program, Input Storage, and Intermediate Storage

Memory allocable to Final Storage 1 and 2 only

Memory available only to Final Storage area 1

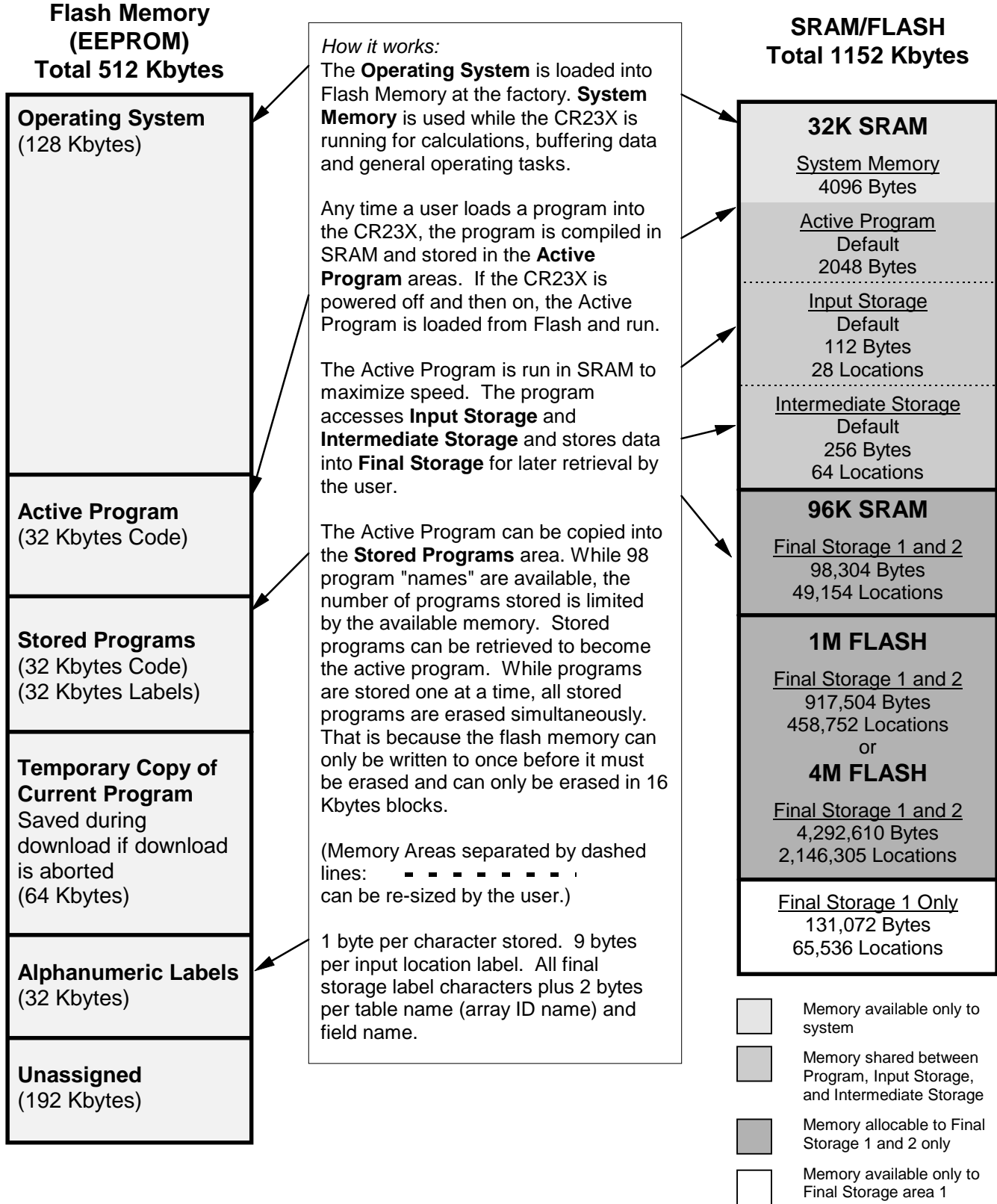**FIGURE OV2.1-1. CR23X Memory**

OV2.2 PROGRAM TABLES, EXECUTION INTERVAL AND OUTPUT INTERVALS

The CR23X must be programmed before it will make any measurements. A program consists of a group of instructions entered into a **program table**. The program table is given an **execution interval** which determines how frequently that table is executed. When the table is executed, the instructions are executed in sequence from beginning to end. After executing the table, the CR23X waits the remainder of the execution interval and then executes the table again starting at the beginning.

The interval at which the table is executed generally determines the interval at which the sensors are measured. The interval at which data are stored is separate from how often the table is executed, and may range from samples every execution interval to processed summaries output hourly, daily, or on longer or irregular intervals.

Programs are entered in Tables 1 and 2. Subroutines, called from Tables 1 and 2, are entered in Subroutine Table 3. The size of program memory can be fixed or automatically allocated by the CR23X (Section 1.5).

Table 1 and Table 2 have independent execution intervals, entered in units of seconds with an allowable range of 1/100 to 6553.5 seconds. Subroutine Table 3 has no execution interval, since it is called from Table 1, Table 2, or an interrupt subroutine.

OV2.2.1 THE EXECUTION INTERVAL

The execution interval specifies how often the program in the table is executed, which is usually determined by how often the sensors are to be measured. *Unless two different measurement rates are needed, use only one table*. A program table is executed sequentially starting with the first instruction in the table and proceeding to the end of the table.

Table 1. Execute every x sec. $0.01 \le x \le 6553.5$	Table 2. Execute every y sec. $0.01 \le y \le 6553.5$	Table 3. Subroutines
Instructions are executed sequentially in the order they are entered in the table. One complete pass through the table is made each execution interval unless program control instructions are used to loop or branch execution.	*Table 2 is used if there is a need to measure and process data on a separate interval from that in Table 1.*	*A subroutine is executed only when called from Table 1 or 2.*
Normal Order: MEASURE PROCESS CHECK OUTPUT COND. OUTPUT PROCESSING		Subroutine Label Instructions End
		Subroutine Label Instructions End
		Subroutine Label Instructions End

FIGURE OV2.2-1. Program and Subroutine Tables

Each instruction in the table requires a finite time to execute. If the execution interval is less than the time required to process the table, an execution interval overrun (table overrun) occurs; the CR23X finishes processing the table and waits for the next execution interval before initiating the table. When a table overrun occurs, $^T{}_o$ appears in the lower right corner of the display in the Running Table mode ($\boxed{*}$ $\boxed{0}$). Overruns and table priority are discussed in Section 1.1.

OV2.2.2. THE OUTPUT INTERVAL

The interval at which output occurs must be an integer multiple of the execution interval (e.g., a table cannot have a 10 minute execution interval and output every 15 minutes).

A single program table can have many different output intervals and conditions, each with a unique data set (Output Array). Program Control Instructions are used to set the Output Flag. The Output Processing Instructions which follow the instruction setting the Output Flag determine the data output and its sequence. Each additional Output Array is created by another Program Control Instruction checking a output condition, followed by Output Processing Instructions defining the data set to output.

OV2.3 CR23X INSTRUCTION TYPES

Figure OV2.3-1 illustrates the use of three different instruction types which act on data. The fourth type, Program Control, is used to control output times and vary program execution. Instructions are identified by numbers.

1. **INPUT/OUTPUT INSTRUCTIONS** (1-29, 100-110, 113-118, 130-134; Section 9) control the terminal strip inputs and outputs (Figure OV1.1-2), storing the results in Input Storage (destination). Multiplier and offset parameters allow conversion of linear signals into engineering units. The Digital

I/O Ports and CAO analog output ports are also addressed with I/O Instructions.

2. **PROCESSING INSTRUCTIONS** (30-68, Section 10) perform numerical operations on values located in Input Storage and store the results back in Input Storage. These instructions can be used to develop high level algorithms to process measurements prior to Output Processing.

3. **OUTPUT PROCESSING INSTRUCTIONS** (69-82, Section 11) are the only instructions which store data in Final Storage. Input Storage values are processed over time to obtain averages, maxima, minima, etc. There are two types of processing done by Output Instructions: **Intermediate** and **Final**.

    Intermediate processing normally takes place each time the instruction is executed. For example, when the Average Instruction is executed, it adds the values from the input locations being averaged to running totals in Intermediate Storage. It also keeps track of the number of samples.

    Final processing occurs only when the Output Flag is high (Section 3.7.1). The Output Processing Instructions check the Output Flag. If the flag is high, final values are calculated and output. With the Average, the totals are divided by the number of samples and the resulting averages sent to Final Storage. Intermediate locations are zeroed and the process starts over. *The Output Flag, Flag 0, is set high by a Program Control Instruction which must precede the Output Processing Instructions in the user entered program.*

4. **PROGRAM CONTROL INSTRUCTIONS** (83-98, 111, 120-123, 220; Section 12) are used for logic decisions, conditional statements, and to send data to peripherals. They can set flags and ports, compare values or times, execute loops, call subroutines, conditionally execute portions of the program, etc.

CR23X MICROLOGGER OVERVIEW

INPUT/OUTPUT INSTRUCTIONS

Specify the conversion of a sensor signal to a data value and store it in Input Storage. Programmable entries specify:
(1) the measurement type
(2) the number of channels to measure
(3) the input voltage range
(4) the Input Storage Location
(5) the sensor calibration constants used to convert the sensor output to engineering units

I/O Instructions also control analog outputs and digital control ports.

INPUT STORAGE

Holds the results of measurements or calculations in user specified locations. The value in a location is written over each time a new measurement or calculation stores data to the locations.

PROCESSING INSTRUCTIONS

Perform calculations with values in Input Storage. Results are returned to Input Storage. Arithmetic, transcendental and polynomial functions are included.

OUTPUT PROCESSING INSTRUCTIONS

Perform calculations over time on the values updated in Input Storage. Summaries for Final Storage are generated when a Program Control Instruction sets the Output Flag in response to time or events. Results may be redirected to Input Storage for further processing. Examples include sums, averages, max/min, standard deviation, histograms, etc.

Output Flag set high

INTERMEDIATE STORAGE

Provides temporary storage for intermediate calculations required by the OUTPUT PROCESSING INSTRUCTIONS; for example, sums, cross products, comparative values, etc.

FINAL STORAGE

Final results from OUTPUT PROCESSING INSTRUCTIONS are stored here for on-line or interrogated transfer to external devices (Figure OV5.1-1). When memory is full, new data overwrites the oldest data.
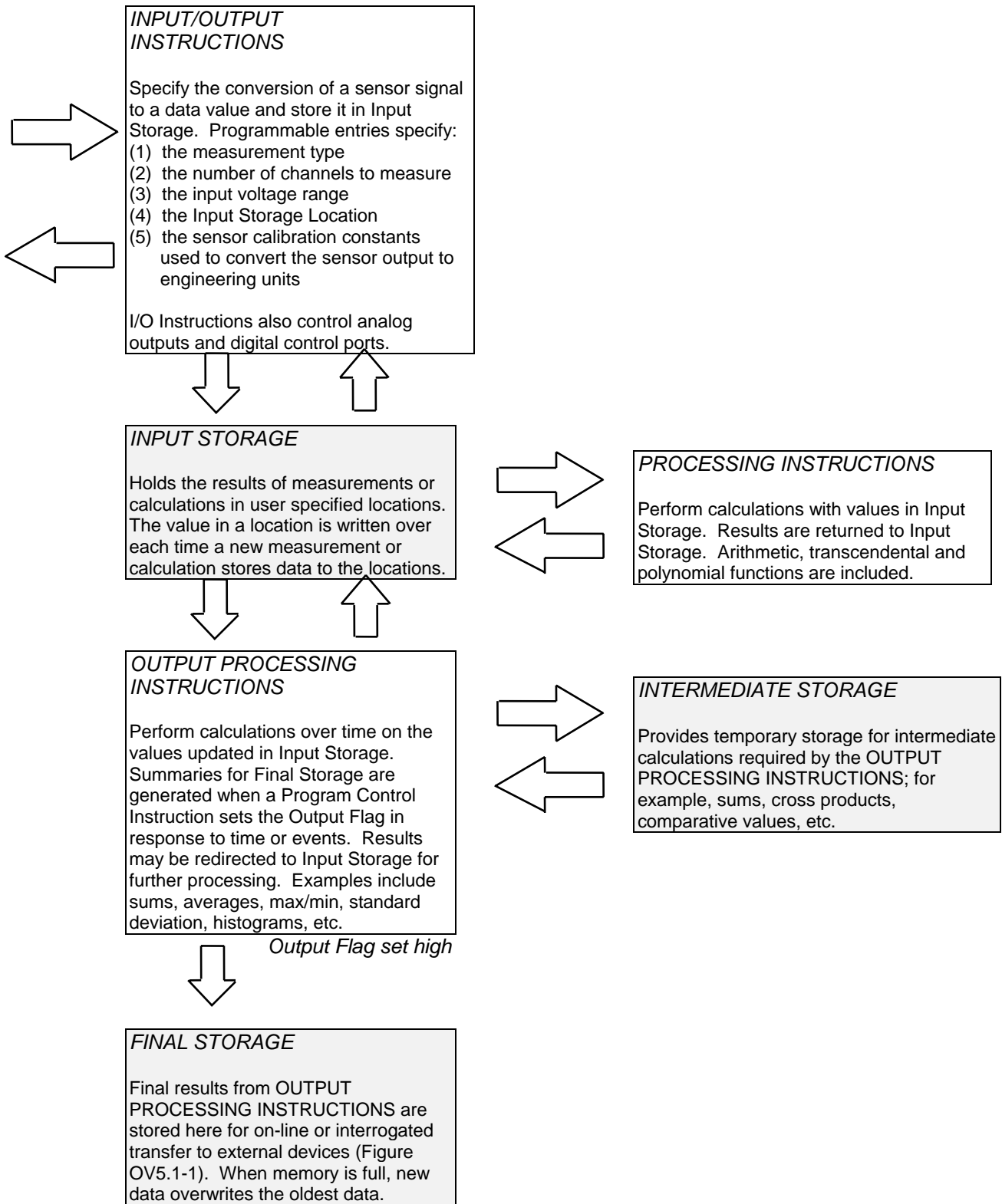
FIGURE OV2.3-1. Instruction Types and Storage Areas

OV3. COMMUNICATING WITH CR23X

The user can communicate with the CR23X through either the integral keyboard and two line LCD display, or through a telecommunications link with a terminal or computer. The preferred method for routine operation is through a telecommunications link with a personal computer running Campbell Scientific's PC208 or PC208W Datalogger Support Software. These packages contain a program editor (EDLOG), datalogger communications, automated telecommunications data retrieval, a data reduction program (SPLIT), and programs to retrieve data from Campbell Scientific Storage Modules.

Some situations, however, require an alternate communications method. The integral keyboard is convenient for cursory on-site inspection of datalogger functions. It can also be used when becoming familiar with the dataloggers functional modes as outlined in Sections OV3.1 through OV5 and Section 1.

A third communications alternative is through a dumb terminal or a computer terminal emulator program through a telecommunications link. Several arcane commands are used in this mode as outlined in Section 5. The most useful command to most CR23X users is the **7H** command, which places the CR23X in the Remote Keyboard Mode. This mode uses the same commands as when communicating on-site through the integral keyboard and display. A common way to use this mode is to enter it through the terminal emulator program in PC208 or PC208W. Once the telecommunications link is established, CR-LF (carriage return - line feed) is issued from the PC by hitting the <Enter> key several times while in the terminal emulator. The CR23X will respond by sending an asterisk (*) to the PC screen. At the *, 7H followed by a CR-LF is issued. The CR23X will respond with a greater-than symbol (>). From the >, the functional modes can be entered as outlined in Section 1.

OV3.1 CR23X KEYPAD/DISPLAY

On power-up, the "HELLO" message is displayed while the CR23X checks memory. The total size of memory is then displayed (1664 K bytes of memory).

Using the keypad, work through the direct programming examples in this overview in addition to using EDLOG and you will have the basics of CR23X operation as well as an appreciation for the help provided by the software and the CR23X on-line help.

The display will turn off automatically if not continuously updated. The display will stay on if continuously updated such as occurs in the $\boxed{*}\,\boxed{5}$ and $\boxed{*}\,\boxed{6}$ modes. Otherwise, it will turn off automatically to save 4 mA of power. Time to display shut off is 3 minutes if left in the $\boxed{*}\,\boxed{0}$ mode, or 6 minutes if left in other modes not continuously updating the screen. While in the $\boxed{*}\,\boxed{0}$ mode, the screen can be manually turned off by pressing the $\boxed{\#}$. Press any other key to turn it back on.

OV3.1.1 FUNCTIONAL MODES

CR23X/User interaction is broken into different functional MODES (e.g., programming the measurements and output, setting time, manually initiating a block data transfer to Storage Module, etc.). The modes are referred to as Star ($\boxed{*}$) Modes since they are accessed by first keying $\boxed{*}$, then the mode number or letter. Table OV3.1-1 lists the CR23X Modes.

Because the display uses approximately 4 mA when active, it is automatically turned off if not updated for three minutes, except in the $\boxed{*}\,\boxed{6}$ mode, where it is left on indefinitely. The display can be turned off from the keypad in the $\boxed{*}\,\boxed{0}$ mode by pressing #. Pressing any key except the # key will cause the display to be turned back on after it has been turned off.

TABLE OV3.1-1. $\boxed{*}$ Mode Summary

Key	Mode
$\boxed{*}\,\boxed{0}$	Compile program, log data and indicate active Tables
$\boxed{*}\,\boxed{1}$	Program Table 1
$\boxed{*}\,\boxed{2}$	Program Table 2
$\boxed{*}\,\boxed{3}$	Program Table 3, subroutines only
$\boxed{*}\,\boxed{4}$	Parameter Entry Table
$\boxed{*}\,\boxed{5}$	Display/set real time clock
$\boxed{*}\,\boxed{6}$	Display/alter Input Storage data, toggle flags or control ports.
$\boxed{*}\,\boxed{7}$	Display Final Storage data
$\boxed{*}\,\boxed{8}$	Final Storage data transfer to peripheral
$\boxed{*}\,\boxed{9}$	Storage Module commands
$\boxed{*}\,\boxed{A}$	Memory allocation/reset
$\boxed{*}\,\boxed{B}$	Signature/status
$\boxed{*}\,\boxed{C}$	Security
$\boxed{*}\,\boxed{D}$	Save/load program, set display contrast, power up settings, ID, etc.
$\boxed{*}\,\boxed{\#}$	Used with TGT1 satellite transmitter

OV3.1.2 KEY DEFINITION

Keys and key sequences have specific functions when using the keypad or a computer/terminal in the remote keyboard state (Section 5). Table OV3.1-2 lists these functions. In some cases, the exact action of a key depends on the mode the CR23X is in and is described with the mode in the manual.

TABLE OV3.1-2 Key Description/Editing Functions

Keys $\boxed{A}$, $\boxed{B}$, $\boxed{C}$, and $\boxed{D}$ repeat when continuously pressed. Repetitions occur slowly at first and then speed up.

Key	Action
Any key	Turn on display (except #)
$\boxed{0}$ - $\boxed{9}$	Key numeric entries into display
$\boxed{*}$	Enter Mode (followed by Mode Number)
$\boxed{A}$	Enter/Advance
$\boxed{B}$	Back up
$\boxed{C}$	Change the sign of a number or index a parameter
$\boxed{D}$	Show Help when "?" is on display Enter the decimal point
$\boxed{\#}$	Turns off display in $\boxed{*}\,\boxed{0}$ Shows output table name in $\boxed{*}\,\boxed{7}$ Clear the rightmost digit keyed into the display
$\boxed{\#}\,\boxed{A}$	Advance to next instruction in program table ($\boxed{*}\,\boxed{1}$, $\boxed{*}\,\boxed{2}$, $\boxed{*}\,\boxed{3}$) or to next Output Array in Final Storage ($\boxed{*}\,\boxed{7}$)
$\boxed{\#}\,\boxed{B}$	Back up to previous instruction in program table or to previous Output Array in Final Storage
$\boxed{\#}\,\boxed{D}$	Delete entire instruction
$\boxed{\#}\,\boxed{0}$	(then A or CR) Back up to the start of the current array.

When using a computer/terminal to communicate with the CR23X (Telecommunications remote keyboard state) there are some keys available in addition to those found on the keypad. Table OV3.1-3 lists these keys.

TABLE OV3.1-3. Additional Keys Allowed in Telecommunications

Key	Action
-	Change Sign, Index (same as C)
CR	Enter/advance (same as A)
S or ^S	Stops transmission of data (10 second time-out; any character restarts)
C or ^C	Aborts transmission of Data

OV3.2 USING COMPUTER WITH DATALOGGER SUPPORT SOFTWARE

Direct datalogger communication programs in the datalogger support software (PC208W) provide menu selection of tools to perform the datalogger functions (e.g., set clock, send program, monitor measurements, and collect data). The user also has the option of directly entering keyboard commands via a built-in terminal emulator (Section OV3.3).

When using the support software, the computer's baud rate, port, and modem types are specified and stored in a file for future use.

The simplest and most common interface is to connect the optically isolated 9 pin "Computer RS-232" port to a 9 pin PC RS-232 port. An adapter is supplied with the CR23X for connection to a 25 pin PC RS-232 port. Otherwise, an SC32A can be used on the CS I/O port. The SC32A converts and optically isolates the voltages passing between the CR23X and the external terminal device.

The SC12 Two Peripheral cable which comes with the SC32A is used to connect the CS I/O port of the CR23X to the 9 pin port of the SC32A labeled "Datalogger". Connect the "Terminal/Printer" port of the SC32A to the serial port of the computer with a straight 25 pin cable or, if the computer has a 9 pin serial port, a standard 9 to 25 pin adapter cable.

OV3.3 ASCII TERMINAL OR COMPUTER WITH TERMINAL EMULATOR

Devices which can be used to communicate with the CR23X include standard ASCII terminals and computers programmed to function as a terminal emulator. See Section 6.7 for details.

To communicate with any device, the CR23X enters its Telecommunications Mode and responds only to valid telecommunications commands. Within the Telecommunications Mode, there are 2 "states"; the Telecommunications Command state and the Remote Keyboard state. Communication is established in the Telecommunications command state. One of the commands is to enter the Remote Keyboard state (Section 5).

The Remote Keyboard state allows the keyboard of the computer/terminal to act like the CR23X keypad. Various datalogger modes may be entered, including the mode in which programs may be keyed in to the CR23X from the computer/terminal.

OV4. PROGRAMMING THE CR23X

A datalogger program is created on a computer using EDLOG. A program can also be entered directly into the datalogger using the keypad. Section OV4.3 describes options for loading the program into the CR23X.

OV4.1 PROGRAMMING SEQUENCE

In routine applications, the CR23X measures sensor output signals, processes the measurements over some time interval and stores the processed results. A generalized programming sequence is:

1. Enter the execution interval. In most cases, the execution interval is determined by the desired sensor scan rate.

2. Enter the Input/Output instructions required to measure the sensors.

3. If processing in addition to that provided by the Output Processing Instructions (step 5) is required, enter the appropriate Processing Instructions.

4. Enter the Program Control Instruction to test the output condition and set the Output Flag when the condition is met. For example, use

    Instruction 92 to output based on time.

    Instruction 86 to output every execution interval.

    Instruction 88 or 89 to output based on a comparison of values in input locations.

This instruction must precede the Output Processing Instructions which store data in Final Storage. Instructions are described in Sections 9 through 12.

5. Enter the Output Processing Instructions to store processed data in Final Storage. The order in which data are stored is determined by the order of the Output Processing Instructions in the table.

6. Repeat steps 4 and 5 for additional outputs on different intervals or conditions.

> **NOTE**: The program must be executed for output to occur. Therefore, the interval at which the Output Flag is set must be evenly divisible by the execution interval. For example, with a 2 minute execution interval and a 5 minute output interval, the output flag will only be set on the even multiples of the 5 minute intervals, not on the odd. Data will be output every 10 minutes instead of every 5 minutes.

Execution intervals and output intervals set with Instruction 92 are synchronized with real time starting at midnight.

OV4.2 INSTRUCTION FORMAT

Instructions are identified by an instruction number. Each instruction has a number of parameters that give the CR23X the information it needs to execute the instruction.

The CR23X Prompt Sheet has the instruction numbers in red, with the parameters briefly listed in columns following the description. Some parameters are footnoted with further description under the "Instruction Option Codes" heading. The CR23X also has on-line help available when a "?" appears on the display. Help is accessed by pressing D .

For example, Instruction 73 stores the maximum value that occurred in an Input Storage location over the output interval.

P73 Maximum
    1:    Reps
    2:    TimeOption
    3:    Loc

The instruction has three parameters (1) REPetitionS, the number of sequential Input

Storage locations on which to find maxima, (2) TIME, an option of storing the time of occurrence with the maximum value, and (3) LOC, the first Input Storage location operated on by the Maximum Instruction. The codes for the TIME parameter are listed in the "Instruction Option Codes".

The repetitions parameter specifies how many times an instruction's function is to be repeated. For example, four 107 thermistor probes may be measured with a single Instruction 11, Temp-107, with four repetitions. Parameter 2 specifies the input channel of the first thermistor (the probes must be connected to sequential channels). Parameter 4 specifies the Input Storage location in which to store measurements from the first thermistor. If location 5 were used and the first probe was on channel 1, the temperature of the thermistor on channel 1 would be stored in input location 5, the temperature from channel 2 in input location 6, etc.

Detailed descriptions of the instructions are given in Sections 9-12. Entering an instruction into a program table is described in OV5.

OV4.3 ENTERING A PROGRAM

Programs are entered into the CR23X in one of three ways:

1. Keyed in using the CR23X keypad.
2. Loaded from a pre-recorded listing using the ⁎ D Mode. There are 2 types of storage/input:
    a. Stored on disk/sent from computer.
    b. Stored/loaded from Storage Module.
3. Loaded from internal Flash Memory or Storage Module upon power-up.

A program is created by keying it directly into the datalogger as described in Section OV5, or on a PC using PC208W.

Program files (.DLD) can be downloaded directly to the CR23X using PC208W. Communication via direct wire, telephone, cellular phone, or Radio Frequency (RF) is supported.

Programs on disk can be copied to a Storage Module with the appropriate software. Using the ⁎ D Mode to save or load a program from a Storage Module is described in Section 1.8.

Once a program is loaded in the CR23X, the program will be stored in flash memory and will automatically be loaded and run when the datalogger is powered-up.

The program on power up function can also be achieved by using a Storage Module. Up to 8 programs can be stored in the Storage Module, the programs may be assigned any of the numbers 1-8. If the Storage Module is connected when the CR23X is powered-up the CR23X will automatically load program number 8, provided that a program 8 is loaded in the Storage Module (Section 1.8). The program from the Storage Module will replace the active program in flash memory.

OV5. PROGRAMMING EXAMPLES

The following examples stress direct interaction with the CR23X using the keypad. At the beginning of each example is an EDLOG listing of the program. You can also participate in the example by entering the program in EDLOG and sending it to the CR23X and viewing measurements with PC208W. (See the PC208W manual for guidance.) You can also work through the examples with the 16 key keypad. You will learn the basics of CR23X operation as well as an appreciation for the help provided by the software.

We will start with a simple programming example. There is a brief explanation of each step to help you follow the logic. When the example uses an instruction, press D on parameters marked with "?" for parameter descriptions. Alternatively, find the instruction on the Prompt Sheet and follow through the description of the parameters. Using the Prompt Sheet or on-line help while going through these examples will help you become familiar with their respective formats. Sections 9-12 have more detailed descriptions of the instructions.

Turn on the CR23X. The programming steps in the following examples use the keystrokes possible on the keypad. With a terminal, some responses will be slightly different.

When the CR23X is powered up, the display will show:

Display	Explanation
HELLO	On power-up, the CR23X displays "HELLO" while it checks the memory

after a few seconds delay

1664 Kbytes memory	The size of the machine's total memory

When the CR23X is turned on, it tests the FLASH memory and loads the current program to RAM. After the program compiles successfully, the CR23X begins executing the program. If a key is pressed while the CR23X is testing memory ("HELLO" is on the display), there will be a 128 second delay before compiling and running the program. This can be used to edit or change the program before it starts running.

In order to ensure that there is no active program in the CR23X, load an empty program using the ✳ D Mode:

Key	Display Will Show: (ID:Data)	Explanation
✳	Mode	Enter mode
D	13:Enter Command 00	Enter ✳ D Mode
7	13: 00 7	7 is command to load program from flash
A	07:Program ID 00	Execute command 7, CR23X is ready for program number
0	07:Program ID 00 00	Load Program 0 (empty program)
A		Execute program load, after a short wait, the display will show
	Prog. operation complete	Indicating that the command is complete.

OV5.1 SAMPLE PROGRAM 1

EDLOG Listing Program 1:

```
*Table 1 Program
01:         5.0          Execution Interval (seconds)

1: Panel Temperature (P17)
  1:        1           Loc [ CR23XTemp ]

2: Do (P86)
  1:        10          Set Output Flag High

3: Sample (P70)
  1:        1           Reps
  2:        1           Loc [ CR23XTemp ]
```

In this example the CR23X is programmed to read its panel temperature (using a built in thermistor) every 5 seconds and to send the results to Final Storage.

Key	Display Will Show: (ID:Data)	Explanation
✳	Mode	Enter mode.
1	Mode 01 Go To 0000	Enter Program Table 1.
A	Scan Interval execution +0000	Advance to execution interval (In seconds)
5	Scan Interval +0.0000 5	Key in an execution interval of 5 seconds.
A	01:P00	Enter the 5 second execution interval and advance to the first program instruction location.
1 7	01:P00 17	Key in Instruction 17 which directs the CR23X to measure the panel temperature in degrees C. This is an Input/Output Instruction.
A	Panel Temp 01:Loc 0000	Enter Instruction 17 and advance to the first parameter.
1	01:Loc 0000 1	The input location to store the measurement, location 1.

Key	Display Will Show: (ID:Data)	Explanation
[A]	02:P00	Enter the location # and advance to the second program instruction.

The CR23X is now programmed to read the panel temperature every 5 seconds and place the reading in Input Storage Location 1. The program can be compiled and the temperature displayed (note that it is not yet storing data).

Key	Display Will Show: (ID:Data)	Explanation
[*] [0]	Running Table 1	Exit Table 1, enter [*] [0] Mode, compile table and begin logging.
[*] [6]	Mode 06 Enter Loc 0001	Enter [*] [6] Mode (to view Input Storage).
[A]	0001: 21.234	Advance to first storage location. Panel temp. is 21.234°C (display shows actual temperature so exact value will vary).

Wait a few seconds:

	01:21.423	The CR23X has read the sensor and stored the result again. The internal temp is now 21.423 ^{o}C. The value is updated every 5 seconds when the table is executed. At this point the CR23X is measuring the temperature every 5 seconds and sending the value to Input Storage. No data are being saved. The next step is to have the CR23X send each reading to Final Storage. (Remember, the Output Flag must be set first.)
[*] [1]	Mode 01 Go To	Exit [*] [6] Mode. Enter 0000 program table 1.

Key	Display Will Show: (ID:Data)	Explanation
[2] [A]	02:P00	Advance to 2nd instruction location (this is where we left off).
[8] [6]	02:P00 86	This is the DO instruction (a Program Control Instruction).
[A]	Do 01:CMD 00	Enter 86 and advance to the first parameter (which will specify the command to execute).
[1] [0]	01:CMD 00 10	This command sets the Output Flag (Flag 0) high.
[A]	03:P00	Enter 10 and advance to third program instruction.
[7] [0]	03:P00 70	The SAMPLE instruction. It directs the CR23X to take a reading from an Input Storage location and send it to Final Storage (an Output Processing Instruction).
[A]	Sample 01:Reps 0000	Enter 70 and advance to the first parameter (repetitions).
[1]	01:Reps 0000	There is only one input location to sample; repetitions = 1.
[A]	02:Loc 0000	Enter 1 and advance to second parameter (Input Storage location to sample).
[1]	02:Loc 0000 1	Input Storage Location 1, where the temperature is stored.
[A]	04:P00	Enter 1 and advance to fourth program instruction.
[*]	Mode	Exit Table 1.
[0]	Running Table 1	Enter [*] [0] Mode, compile program, log data.

The CR23X is now programmed to measure the internal temperature every 5 seconds and send each reading to Final Storage. Values in Final Storage can be viewed using the `*` `7` *Mode.*

	Display Will Show:	
Key	(ID:Data)	Explanation
`*` `7`	Mode 07	Enter `*` `7` Mode. The Loc 13 Data Storage Pointer (DSP) is at Location 13 (in this example).
`A`	Array ID 01: +0102	Advance to the first value, the Output Array ID. 102 indicates the Output Flag was set by the second instruction in Program Table 1.
`A`	02: +21.231	Advance to the first stored temperature.
`A`	Array ID 01: +0102	Advance to the next output array. Same Output Array ID.
`A`	02: +21.42	Advance to 2nd stored temp, 21.42 deg. C.

There are no date and time tags on the data. They must be put there with Output Instruction 77. Instruction 77 is used in the next example.

If a terminal is used to communicate with the CR23X, Telecommunications Commands (Section 5) can be used to view entire Output Arrays (in this case the ID and temperature) at the same time.

OV5.2 SAMPLE PROGRAM 2

EDLOG Listing Program 2:

*Table 1 Program
  01:    60.0   Execution Interval (seconds)

1: Panel Temperature (P17)
  1:      1   Loc [CR23XTemp]

2: Thermocouple Temp (DIFF) (P14)
  1:      1   Reps
  2:      21   ± 10 mV 60 Hz Rejection
  3:      5   DIFF Channel
  4:      1   Type T (Copper-Constantan)
  5:      1   Ref Temp Loc [CR23XTemp]

  6:      2   Loc [TCTemp   ]
  7:      1.0   Mult
  8:      0.0   Offset

3: If time is (P92)
  1:      0   Minutes (Seconds --) into a
  2:      60   Interval (same units as above)
  3:      10   Set Output Flag High

4: Real Time (P77)
  1:      110   Day,Hour/Minute

5: Average (P71)
  1:      2   Reps
  2:      1   Loc [CR23XTemp]

6: If time is (P92)
  1:      0   Minutes (Seconds --) into a
  2:      1440   Interval (same units as above)
  3:      10   Set Output Flag High

7: Real Time (P77)
  1:      110   Day,Hour/Minute

8: Maximum (P73)
  1:      1   Reps
  2:      10   Value with Hr-Min
  3:      2   Loc [TCTemp   ]

9: Minimum (P74)
  1:      1   Reps
  2:      10   Value with Hr-Min
  3:      2   Loc [TCTemp   ]

10:    Serial Out (P96)
  1:      71   SM192/SM716/CSM1

This second example is more representative of a real data collection application. Once again, the panel temperature is measured, but it is used as a reference temperature for the differential voltage measurement of a type T (copper-constantan) thermocouple; the CR23X is initially supplied with a short type T thermocouple connected to differential channel 5.

When using a type T thermocouple, the copper lead (blue) is connected to the high input of the differential channel, and the constantan lead (red) is connected to the low input.

A thermocouple produces a voltage that is proportional to the difference in temperature between the measurement and the reference junctions.

To make a thermocouple (TC) temperature measurement, the temperature of the reference junction (in this example, the panel temperature) must be measured. The CR23X takes the reference temperature, converts it to the equivalent TC voltage relative to 0°C, adds the measured TC voltage, and converts the sum to temperature through a polynomial fit to the TC output curve (Section 13.4).

Instruction 14 directs the CR23X to make a differential TC temperature measurement. The first parameter in Instruction 14 is the number of times to repeat the measurement. Enter 1, because in this example there is only one thermocouple. If there were more than 1 TC, they could be wired to sequential channels, and the number of thermocouples entered for repetitions. The CR23X would automatically advance through the channels sequentially and measure all of the thermocouples.

Parameter 2 is the voltage range to use when making the measurement. The output of a type T thermocouple is approximately 40 microvolts per degree C difference in temperature between the two junctions. The ±10 mV scale will provide a range of +1000/40 = +250°C (i.e., this scale will not overrange as long as the measuring junction is within 250°C of the panel temperature). The resolution of the ±10 mV range is 0.33 µV or 0.008°C because a differential measurement is being made.

Parameter 3 is the analog input channel on which to make the first, and in this case only, measurement.

Parameter 4 is the code for the type of thermocouple used. This information is located on the Prompt Sheet, in the on-line help, or in the description of Instruction 14 in Section 9. The code for a type T (copper-constantan) thermocouple is 1.

Parameter 5 is the Input Storage location in which the reference temperature is stored. Parameter 6 is the Input Storage location in which to store the measurement (or the first measurement; e.g., if there are 5 repetitions and the first measurement is stored in location 3, the final measurement will be stored in location 7). Parameters 7 and 8 are the multiplier and offset. A multiplier of 1 and an offset of 0 outputs the reading in degrees C. A multiplier of 1.8 and an offset of 32 converts the reading to degrees F.

In this example, the sensor is measured once a minute, and the day, time, and average temperature are output every hour. Once a day the day, time, maximum and minimum temperatures and the times they occur will be output.

Final Storage data will be sent to Storage Module. Remember, all on-line data output to a peripheral device is accomplished with Instruction 96 (Sections 4.1 and 12).

The first example described program entry one keystroke at a time. This example does not show the "A" key. Remember, "A" is used to enter and/or advance (i.e., between each line in the example below). This format is similar to the format used in EDLOG.

It's a good idea to have both the manual and the Prompt Sheet handy when going through this example. Also look at the on-line help, key , whenever "?" is displayed on the screen. You can find the program instructions and parameters on the Prompt Sheet and can read their complete definitions in the manual.

To obtain daily output, the If Time instruction is again used to set the Output Flag and is followed by the Output Instructions to store time and the daily maximum and minimum temperatures and the time each occurs.

Any Program Control Instruction which is used to set the Output Flag high will set it low if the conditions are not met for setting it high. Instruction 92 above sets the Output Flag high every hour. The Output Instructions which follow do not output every hour because they are preceded by another Instruction 92 which sets the Output Flag high at midnight (and sets it low at any other time). This is a unique feature of Flag 0. The Output Flag is automatically set low at the start of each table (Section 3.7).

OV5.3 EDITING AN EXISTING PROGRAM

When editing an existing program in the CR23X, entering a new instruction inserts the instruction; entering a new parameter replaces the previous value.

To insert an instruction, enter the program table and advance to the position where the instruction is to be inserted (i.e., PXX in the display) key in the instruction number, and then key A. The new instruction will be inserted at that point in the table, advance through and enter the parameters. The instruction that was at that point and all instructions following it will be pushed down to follow the inserted instruction.

An instruction is deleted by advancing to the instruction number (P in display) and keying #D (Table 4.2-1).

To change the value entered for a parameter, advance to the parameter and key in the correct value then press A. Note that the new value is not entered until A is keyed.

SAMPLE PROGRAM 2

Instruction (Loc:Entry)	Parameter (Par#:Entry)	Description
⎡*⎤ ⎡1⎤		Enter Program Table 1
01:60		60 second (1 minute) execution interval
Key ⎡#⎤⎡D⎤ until is displayed	01:P00	Erase previous Program before continuing.
01:P17		Measure panel temperature
	01:1	Store temp in Location 1
02:P14 (differential)		Measure thermocouple temperature
	01:1	1 repetition
	02:21	Range code (10 mV, 60 Hz Rejection)
	03:5	Input channel of TC
	04:1	TC type: copper-constantan
	05:1	Reference temp is stored in Location 1
	06:2	Store TC temp in Location 2
	07:1	Multiplier of 1
	08:0	No offset
03:P92		If Time instruction
	01:0	0 minutes into the interval
	02:60	60 minute interval
	03:10	Set Output Flag 0

The CR23X is programmed to measure the thermocouple temperature every sixty seconds. The If Time instruction sets the Output Flag at the beginning of every hour. Next, the Output Instructions for time and average are added.

Instruction # (Loc.:Entry)	Parameter (Par.#:Entry)	Description
04:P77		Output Time instruction
	01:110	Store Julian day, hour, and minute
05:P71		Average instruction
	01:1	one repetition
	02:2	Location 2 - source of TC temps. to be averaged
06:P92		If Time instruction
	01:0	0 minutes into the interval
	02:1440	1440 minute interval (24 hrs.)
	03:10	Set Output Flag 0
07: P77		Output Time instruction
	01:100	Store Day of Year
08: P73		Maximize instruction
	01:1	One repetition
	02:10	Output time of daily maximum in hours and minutes
	03:2	Data source is Input Storage Location 2.

09: P74		Minimize instruction
	01:1	One repetition
	02:10	Output the time of the daily minimum in hours and minutes
	03:2	Data source is Input Storage Location 2.

The program to make the measurements and to send the desired data to Final Storage has been entered. At this point, Instruction 96 is entered to enable data transfer from Final Storage to Storage Module.

10:P96		Activate Serial Data Output.
	1:71	Output Final Storage data to Storage Module.

The program is complete. (Here the example reverts back to the key by key format.)

Key	Display	Explanation
⁎ 5	00:21:32	Enter ⁎ 5 Mode. Clock running but perhaps not set
correctly.		
A	05:0000	Advance to location for year.
1 9 9 6	05:1996	Key in year (1996).
A	05:0000	Enter and advance to location for Julian day.
1 9 7	05:197	Key in Julian day.
A	05:0021	Enter and advance to location for hours and minutes (24 hr. time).
1 3 2 4	05:1324	Key in hrs.:min. (1:24 PM in this example).
A	:13:24:01	Clock set and running.
⁎ 0	Running Table 1	Exit ⁎ 5 , compile Table 1, commence logging data.

OV6. DATA RETRIEVAL OPTIONS

There are several options for data storage and retrieval. These options are covered in detail in Sections 2, 4, and 5. Figure OV6.1-1 summarizes the various possible methods.

Regardless of the method used, there are three general approaches to retrieving data from a datalogger.

1) On-line output of Final Storage data to a peripheral storage device. On a regular schedule, that storage device is either "milked" of its data or is brought back to the office/lab where the data is transferred to the computer. In the latter case, a "fresh" storage device is usually left in the field when the full one is taken so that data collection can continue uninterrupted.

2) Bring a storage device to the datalogger and milk all the data that has accumulated in Final Storage since the last visit.

3) Retrieve the data over some form of telecommunications link, whether it be RF, telephone, cellular phone, short haul modem, or satellite. This can be performed under program control or by regularly scheduled polling of the dataloggers. Campbell Scientific's Datalogger Support Software automates this process.

Regardless of which method is used, the retrieval of data from the datalogger does NOT erase those data from Final Storage. The data remain in the ring memory until:

They are written over by new data (Section 2.1)

Memory is reallocated or the CR23X is reset (Section 1.5)

Table OV6.1-1 lists the instructions used with the various methods of data retrieval.

TABLE OV6.1-1. Data Retrieval Methods and Related Instructions

Method	Instruction/Mode	Section in Manual
Storage Module	Instruction 96	4.1, 12
	∗ 8	4.2
	∗ 9	4.5
Telecommunications	Telecommunications Commands	5
	Instruction 97	12
	Instruction 99	12
Printer or other Serial device	Instruction 96	4.1, 12
	Instruction 98	12
	∗ 8	4.2

CR23X MICROLOGGER OVERVIEW
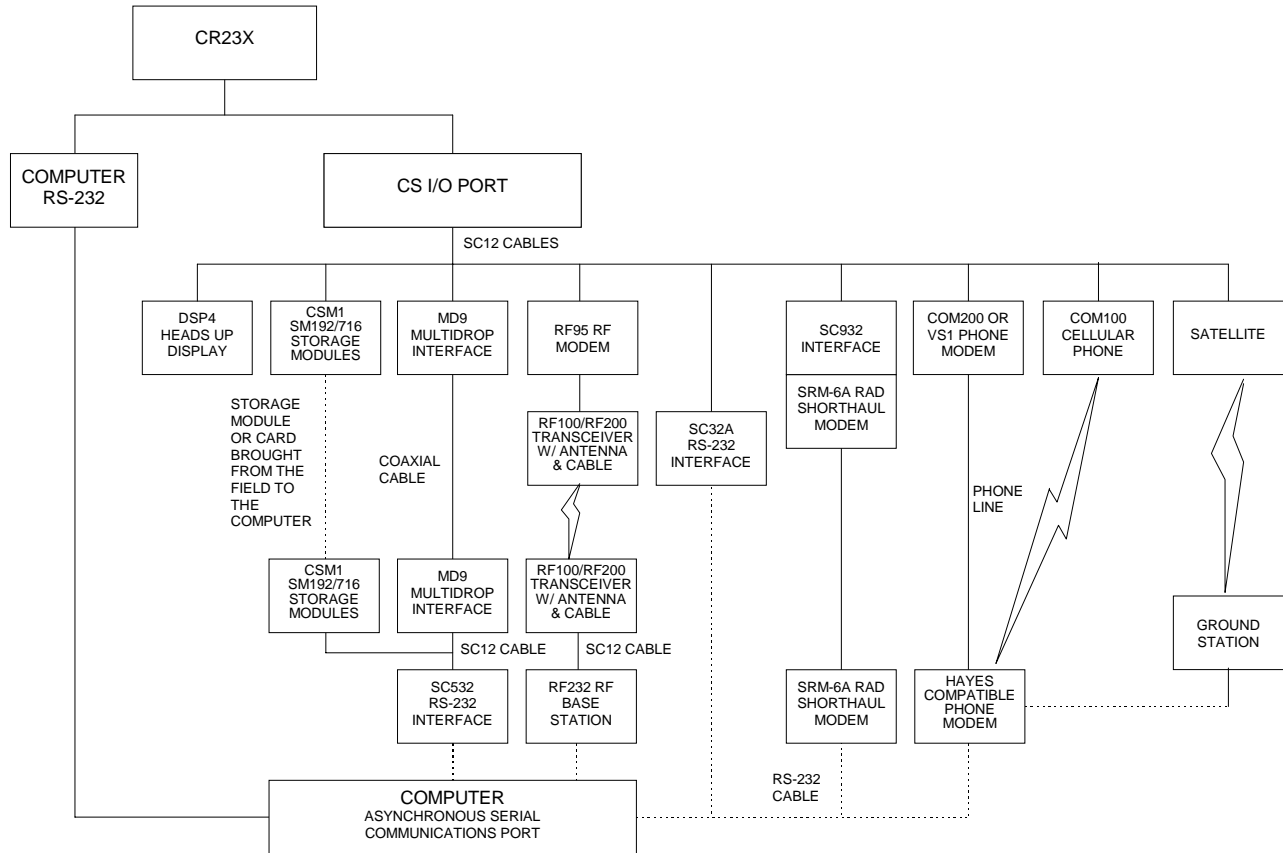
NOTES: 1.   ADDITIONAL METHODS OF DATA RETRIEVAL ARE:
   A.   SATELLITE TRANSMISSION
   B.   DIRECT DUMP TO PRINTER
   C.   VOICE PHONE MODEM TO VOICE PHONE OR PC WITH HAYES COMPATIBLE PHONE MODEM

2.   THE DSP4 HEADS UP DISPLAY ALLOWS THE USER TO VIEW DATA IN INPUT STORAGE. ALSO BUFFERS FINAL STORAGE DATA AND WRITES IT TO PRINTER OR STORAGE MODULE.

3.   ALL CAMPBELL SCIENTIFIC RS-232 INTERFACE PERIPHERALS HAVE A FEMALE 25 PIN RS-232 CONNECTOR.

4.   THE "COMPUTER RS-232" PORT HAS A FEMALE 9 PIN CONNECTOR.

FIGURE OV6.1-1. Data Retrieval Hardware Options

OV7. SPECIFICATIONS

Electrical specifications are valid for -25° to 50°C range unless otherwise specified. To maintain electrical specifications, yearly recalibrations are recommended.

PROGRAM EXECUTION RATE
Program is synchronized with real-time up to 100 Hz. Two fast (250 µs integration) single-ended measurements can write to final storage at 100 Hz. Burst measurements are possible at rates up to 1.5 kHz over short intervals.

CLOCK ACCURACY
±1 minute per month

ANALOG INPUTS
DESCRIPTION: 12 differential or 24 single-ended, individually configured. Channel expansion provided through AM416 Relay Multiplexers and AM25T Thermocouple Multiplexers.

ACCURACY: ±0.025% of FSR; 0° to 40°C
±0.05% of FSR; -25° to 50°C
±0.075% of FSR; -40° to 80°C (optional)
±5 µV offset voltage error is possible with SE measurements.

RANGES AND RESOLUTION

Input Range (mV)	Resolution (µV) Diff.	SE	Accuracy (mV) (-25° to 50°C)
±5000	166	333	±5.00
±1000	33.3	66.6	±1.00
±200	6.66	13.3	±0.20
±50	1.67	3.33	±0.05
±10	0.33	0.66	±0.01

INPUT SAMPLE RATES: Includes the measurement time and conversion to engineering units. Differential measurements incorporate two integrations with reversed input polarities to reduce thermal offset and common mode errors. Fast measurement integrates the signal for 250 µs; slow measurement integrates for one power line cycle (50 or 60 Hz).

Fast single-ended voltage:	2.1 ms
Fast differential voltage:	3.1 ms
Slow single-ended voltage (60 Hz):	18.3 ms
Slow differential voltage (60 Hz):	35.9 ms
Fast differential thermocouple:	6.9 ms

INPUT REFERRED NOISE: Typical for ±10 mV Input Range; digital resolution dominates for higher ranges.

Fast differential	0.60 µV rms
Slow differential (60 Hz)	0.15 µV rms
Fast single-ended	1.20 µV rms
Slow single-ended (60 Hz)	0.30 µV rms

COMMON MODE RANGE: ±5 V.

DC COMMON MODE REJECTION: >100 dB.

NORMAL MODE REJECTION: 70 dB (60 Hz with slow diff. measurement).

SUSTAINED INPUT VOLTAGE WITHOUT DAMAGE: ±16 VDC max.

INPUT CURRENT: ±2.5 nA typ., ±10 nA max. at 50°C.

INPUT RESISTANCE: 20 Gohms typical.

ANALOG OUTPUTS
DESCRIPTION: 4 switched, active only during measurement one at a time; 2 continuous.

RANGE: Programmable between ±5 V

RESOLUTION: 333 µV

ACCURACY: ±5 mV; ±2.5 mV (0° to 40°C).

CURRENT SOURCING: 50 mA for switched; 15 mA for continuous.

CURRENT SINKING: 50 mA for switched, 5 mA for continuous (15 mA for continuous with Boost selected in P133).

FREQUENCY SWEEP FUNCTION: The switched outputs provide a programmable swept frequency, 0 to 5 V square wave for exciting vibrating wire transducers.

RESISTANCE MEASUREMENTS
MEASUREMENT TYPES: The CR23X provides ratiometric measurements of 4- and 6-wire full bridges, and 2-, 3-, and 4-wire half bridges. Precise, dual polarity excitation using any of the 4 switched outputs eliminates DC errors. Conductivity measurements use a dual polarity 0.75 ms excitation to minimize ionic polarization errors.

ACCURACY: ±0.02% of FSR (±0.015%, 0° to 40°C) plus bridge resistor error.

PERIOD AVERAGING MEASUREMENTS
DESCRIPTION: The average period for a single cycle is determined by measuring the duration of a specified number of cycles. Any of the 24 SE analog inputs can be used; signal attenuation and AC coupling is typically required.

INPUT FREQUENCY RANGE: Signal centered around ground.

Max. Input Frequency	Min. signal (Peak to Peak) @ Max. Freq.
10 kHz	2 mV
20 kHz	5 mV
30 kHz	10 mV
200 kHz	500 mV

RESOLUTION: 12 ns divided by the number of cycles measured.

ACCURACY: ±0.03% of reading.

PULSE COUNTERS
DESCRIPTION: Four 8-bit or two 16-bit inputs selectable for switch closure, high frequency pulse, or low-level AC. Counters read at 10 or 100 Hz.

MAXIMUM COUNT RATE: 2.5 kHz and 25 kHz, 8-bit counter read at 10 Hz and 100 Hz, respectively; 500 kHz, 16-bit counter.

SWITCH CLOSURE MODE
Minimum Switch Closed Time: 5 ms.
Minimum Switch Open Time: 6 ms.
Maximum Bounce Time: 1 ms open without being counted.

HIGH FREQUENCY PULSE MODE
Minimum Pulse Width: 1 µs.
Maximum Input Frequency: 500 kHz.
Voltage Thresholds: Count upon transition from below 1.5 V to above 3.5 VDC. Larger transitions required at high frequencies because of 0.5 µs time constant filter.
Maximum Input Voltage: ±20 V.

LOW LEVEL AC MODE
Internal AC coupling removes DC offsets up to ±0.5 V.
Input Hysteresis: 15 mV.
Maximum AC Input Voltage: ±20 V.

Frequency Range	Min. sine wave rms
1.0 Hz to 1 kHz	20 mV
0.5 Hz to 10 kHz	200 mV
0.3 Hz to 16 kHz	1000 mV

DIGITAL I/O PORTS
DESCRIPTION: 8 ports selectable as binary inputs or control outputs. Ports C5-C8 capable of counting switch closures and high frequency.

HIGH FREQUENCY MAX: 2.5 kHz

OUTPUT VOLTAGES (no load): high 5.0 V ±0.1 V; low < 0.1 V.

OUTPUT RESISTANCE: 500 ohms.

INPUT STATE: high 3.0 to 5.5 V; low -0.5 to 0.8 V.

INPUT RESISTANCE: 100 kohms.

SDI-12 INTERFACE SUPPORT
DESCRIPTION: Digital I/O Ports C5-C8 support SDI-12 asynchronous communication; up to 10 SDI-12 sensors can be connected to each port.

EMI and ESD PROTECTION
Encased in metal with gas discharge tubes on the panel, the CR23X has EMI filtering and ESD protection on all input and output connections.

CE COMPLIANCE (as of 01/98)
APPLICATION OF COUNCIL DIRECTIVE(S): 89/336/EEC as amended by 89/336/EEC and 93/68/EEC

STANDARD(S) TO WHICH CONFORMITY IS DECLARED:
ENC55022-1: 1995 and ENC50082-1: 1992

CPU AND INTERFACE
PROCESSORS: Hitachi 6303; Motorola 68HC708 supports communications.

MEMORY: 1 M Flash stores 500K data values; 512K Flash stores OS and user programs with 128K battery-backed SRAM. Optional 4 M Flash available.

DISPLAY: 24-character-by-2-line LCD.

SERIAL INTERFACES: Optically isolated RS-232 9-pin interface for computer or modem. CS 9-pin I/O interface for peripherals such as card storage module or modem.

BAUD RATES: Selectable at 300, 1200, 2400, 4800, 9600, 19.2K, 38.4K, and 76.8K. ASCII protocol is one start bit, one stop bit, eight data bits, no parity.

SYSTEM POWER REQUIREMENTS
VOLTAGE: 11 to 16 VDC.

TYPICAL CURRENT DRAIN: 2 mA quiescent with display off (2.5 mA max), 7 mA quiescent with display on, 45 mA during processing, and 70 mA during analog measurement.

INTERNAL BATTERIES: 7 Ahr alkaline or 7 Ahr rechargeable base; low-profile base without batteries optional. 1800 mAhr lithium battery for clock and SRAM backup typically provides 10 years of service.

EXTERNAL BATTERIES: Any 11 to 16 V battery may be connected; reverse polarity protected.

PHYSICAL SPECIFICATIONS
SIZE: 9.5" x 7.0" x 3.8" (24.1 cm x 17.8 cm x 9.6 cm). Terminal strips extend 0.4" (1.0 cm) and terminal strip cover extends 1.3" (3.3 cm) above the panel surface.

WEIGHT: 3.6 lbs (1.6 kg) with low-profile base, 8.3 lbs (3.8 kg) with alkaline base, 10.7 lbs (4.8 kg) with rechargeable base.

WARRANTY
3 years against defects in materials and workmanship.

SECTION 1. FUNCTIONAL MODES

1.1 DATALOGGER PROGRAMS - ⁕ 1 , ⁕ 2 , ⁕ 3 , AND ⁕ 4 MODES

Data acquisition and processing functions are controlled by user-entered instructions contained in program tables. Programming can be separated into 2 tables, each having its own user-entered execution interval. A third table is available for programming subroutines which may be called by instructions in Tables 1 or 2 or by special interrupts. The ⁕ 1 and ⁕ 2 Modes are used to access Tables 1 and 2. The ⁕ 3 Mode is used to access Subroutine Table 3.

The ⁕ 4 Mode Table is a table of values used in the program that someone can change while the rest of the program is protected. These values may be used for sensor calibrations or to select optional sensors. The ⁕ 4 Table is only available when a special program created by EDLOG is loaded in the CR23X.

When a program table is first entered, the display shows the mode (table) number on the first line and 0000 on the second line. Keying an "A" will advance the editor to the scan interval. If there is an existing program in the table, keying an instruction location number prior to "A" will advance directly to the instruction (e.g., 5 will advance to the fifth instruction in the table).

1.1.1 SCAN (EXECUTION) INTERVAL

The scan interval is entered in units of seconds as follows:

1/100 .... 1 second, in multiples of 1/100 (.01)

1 .... 6553.5 seconds, in multiples of 1/10 (0.1)

Execution of the table is repeated at the rate determined by this entry. The table will not be executed if 0 is entered.

The sample rate for a CR23X measurement is the rate at which the measurement instruction can be executed (i.e., the measurement made, scaled with the instruction's multiplier and offset, and the result placed in Input Storage).

Additional processing requires extra time. The throughput rate is the rate at which a measurement can be made and the resulting value stored in Final Storage. The maximum throughput rate for fast single-ended measurements, other than burst measurements, is 600 measurements per second (24 measurements repeated 25 times per second with the settling time set at 100 µs with Instruction P132).

If the specified execution interval for a table is less than the time required to process that table, the CR23X finishes processing the table and waits for the next occurrence of the execution interval before again initiating the table (i.e., when the execution interval has elapsed and the table is still executing, that execution is skipped). Since no advantage is gained in the rate of execution with this situation, it should be avoided by specifying an execution interval adequate for the table processing time.

> **NOTE:** Whenever the processing time of the user's program exceeds a table's execution interval, an error is logged in memory. The number of overrun errors can be displayed and reset in the B mode (Section 1.6) or using the Telecommunications A command (Section 5.1). An overrun will also cause "T_o" to appear in the lower right corner of the display. "T_o" will appear on the first table overruns and continue to be displayed until table overruns stop and ⁕ 0 or another ⁕ mode command is entered.

In some cases, the processing time may exceed the execution interval only when the Output Flag is set and extra time is consumed by final Output Processing. This may be acceptable. For example, suppose it is desired to sample some phenomena every 0.1 seconds and output processed data every 10 minutes. The processing time of the table which does this is less than 0.1 seconds except when output occurs (every 10 minutes). With final output the processing time is 1 second. With the execution interval set at 0.1 seconds, and a one second lag between samples once every 10 minutes, 9 measurements out of 5000 (.18%) are missed: an acceptable statistical error for most populations.

1.1.2 SUBROUTINES

Table 3 is used to enter subroutines which may be called with Program Control Instructions in Tables 1 and 2 or other subroutines. The group of instructions which form a subroutine must begin with Instruction 85, Label Subroutine, and end with Instruction 95, End (Section 12).

Subroutines 95, 96, 97, and 98 have the unique capability of being executed when a port goes high (ports 5, 6, 7, and 8 respectively). Any of these subroutines will interrupt Tables 1 and 2 (Section 1.1.3) when the appropriate port goes high. When the port goes high, the processor awakes within a few microseconds. The port triggers on the rising edge (i.e., when it goes from low to high). If the port stays high, the subroutine is not called again.

1.1.3 TABLE PRIORITY/INTERRUPTS

Table 1 execution has priority over Table 2. If Table 2 is being executed when it is time to execute Table 1, Table 2 will be interrupted. After Table 1 processing is completed, Table 2 processing resumes at the interruption point. If the execution interval of Table 2 coincides with Table 1, Table 1 is executed first, then Table 2.

Interrupts by Table 1 are not allowed in the middle of an instruction or while output to Final Storage is in process (flag 0 is set high). The interrupt occurs as soon as the instruction is completed or flag 0 is set low.

Special subroutines 95, 96, 97, and 98, initiated by a port going high (Section 1.1.2), can interrupt either Table 1 or 2 or can occur when neither is being executed. These subroutines can interrupt a table while the Output Flag is set. When the port goes high during the execution of a table, the instruction being executed is completed before the subroutine is run (i.e., as if the subroutine was called by the next instruction). For more information, refer to Section 12 (P85 Label Subroutine).

1.1.4 ✳ 4 PARAMETER ENTRY TABLE

The CR23X ✳ 4 mode is a table with up to one hundred values. Each value corresponds to an instruction parameter in the datalogger program. When the datalogger compiles the program, values in the ✳ 4 table are transferred to the corresponding instruction parameter. The datalogger program must be created using EDLOG which allows instruction parameters to be assigned to the ✳ 4 table.

In a network of datalogger stations, the ✳ 4 table can be used to simplify site customization and the procedure of information entry. Once a generalized program is developed, application specific details, e.g., sensor calibration, can be entered without accessing the ✳ 1 and ✳ 2 program tables or the ✳ 3 subroutine table.

ASSIGNING PARAMETERS TO ✳ 4 - EDLOG

The only way to implement the ✳ 4 mode is through EDLOG. The datalogger program is generated in EDLOG in the normal way.

To assign a parameter to a ✳ 4 location, position the cursor on the desired parameter and press the "@" key. EDLOG then prompts for the location number in the ✳ 4 table to be assigned to the associated parameter. After a valid number is entered, EDLOG marks the parameter with "@@nn" to the right of the parameter description, where "nn" is the ✳ 4 location number.

Any program parameter or execution interval can be marked for inclusion in the table, as illustrated below.

PROGRAM

```
*      Table 1 Program
  01:       0.0       Execution Interval
                      (seconds) @@0

01:    Volts (SE) (P1)
       1:   1         Reps
       2:   1         ±2.5 mV Slow Range
       3:   1         SE Channel
       4:   1         Loc [ _____ ]
       5:   1         Mult @@1
       6:   0         Offset @@2
```

In the above example, ✳ 4 location 0 is assigned to the program table execution interval, and locations 1 and 2 to the multiplier and offset of the measurement instruction. Note that a default execution interval of zero means the program will not execute until an alternative interval is entered in location 00 of the ✳ 4 mode. A default multiplier and offset of 1 and 0 means that the measurement value is in units of millivolts. A different

multiplier and offset can be entered in ∗ 4 locations 1 and 2, respectively.

A ∗ 4 location can be used in only one program parameter. For example, ∗ 4 locations 0, 1, and 2 used in the example cannot be reused in another instruction in the same program.

If the ∗ 4 feature is enabled in EDLOG when printing a program to a printer or disk file, the ∗ 4 list is printed at the end of the file.

Once the EDLOG created program has been sent to the CR23X, it can be saved in the Flash memory program storage area using the ∗ D Mode (Section 1.8).

CHANGING VALUES IN ∗ 4 TABLE

Enter the ∗ 4 Mode by keying "∗ 4 "; "04:00" is then displayed. At this point it is possible to jump to any valid ∗ 4 location by keying the desired location number and pressing the A key. For example, when the display shows 04:00 and the desired location is 80, key in the number 80, press the A key and the display will show "80:XXXXX." where XXXXX. is the value stored in location 80. Pressing the "A" key advances to the next ∗ 4 location, and the "B" key backs up to the previous location. If a ∗ 4 location is not assigned in the datalogger program, it can not be displayed in the ∗ 4 mode.

To enter a value in a ∗ 4 location, advance to the desired location, key in the number and enter it by pressing the "A" key. The value is not entered if the "A" key is not pressed.

Entering a new value causes the datalogger to stop logging. Logging resumes when the program is compiled. Upon compiling (∗ 0 or ∗ 6), all current ∗ 4 values are incorporated into the program. For this reason, whenever changes are made in the ∗ 4 mode, make sure that all ∗ 4 values are correct before exiting the ∗ 4 mode.

Removing or adding an instruction to a program residing in the datalogger disables the ∗ 4 mode. An instruction parameter may be edited without any adverse affect. If the ∗ 4 mode is disabled, it may be reactivated by downloading the program to the datalogger or, if the program was saved to Flash storage,

retrieving the program from the stored program area.

The ∗ C mode (Section 1.7) may be used to secure the datalogger program and the ∗ 4 mode entries. The lowest level of security prevents access to the ∗ 1 , ∗ 2 , and ∗ 3 modes. Higher levels of security block ∗ 4 .

The CR23X will not respond to the ∗ 4 command if any of the following conditions exist.

- the program that was downloaded does not contain any ∗ 4 assignments.

- a program that was downloaded has since been hand edited.

- Security is blocking access to ∗ 4 .

1.1.5 COMPILING A PROGRAM

When a program is first loaded, or if any changes are made in the ∗ 1 , ∗ 2 , ∗ 3 , ∗ 4 , ∗ A , or ∗ C Modes, the program must be compiled before it starts running. The compile function checks for programming errors and optimizes program information for use during program execution. If errors are detected, the appropriate error codes are indicated on the display (Section 3.10). The compile function is executed when the ∗ 0 , ∗ 6 , or ∗ B Modes are entered and prior to saving a program listing in the ∗ D Mode. The compile function is only executed after a program change has been made and any subsequent use of any of these modes will return to the mode without recompiling.

When the ∗ 0 or ∗ B Mode is used to compile, all output ports and flags are set low, the timer is reset, and data values contained in Input and Intermediate Storage are reset to zero.

When ∗ 0 is used, one of the following status lines will be displayed: Running Table 1, Running Table 2, No Active Program, Running Table 1, 2, Tables Not Running.

When the ∗ 6 Mode is used to compile data values contained in Input Storage, the state of flags, control ports, and the timer (Instruction 26) are unaltered. Compiling always zeros Intermediate Storage.

1.2 SETTING AND DISPLAYING THE CLOCK - ✳ 5 MODE

The ✳ 5 Mode is used to display or set time. When "∗5" is entered, time is displayed. It is updated approximately once a second or longer depending on the rate and degree of data collection and processing taking place. The sequence of time parameters displayed in the ✳ 5 Mode is given in Table 1.2-1.

To set the year, day, or hours and minutes, enter the ✳ 5 Mode and advance to display the appropriate value. Key in the desired number and enter the value by keying "A". When a new value for hours and minutes is entered, the seconds are set to zero and current time is again displayed. To exit the ✳ 5 Mode, key "∗" and the mode you wish to enter.

When the time is changed, a partial recompile is done automatically to synchronize the program with real time.

Changing time affects the output and execution intervals in which time is changed. Because time can only be set with a 1 second resolution, execution intervals of 1 second or less remain constant. Averaged values will still be accurate, though the interval may have a different number of samples than normal. Totalized values will reflect the different number of samples. The pulse count instruction will use the previous interval's value if an option has been selected to discard odd intervals, otherwise it will use the count accumulated in the interval.

TABLE 1.2-1. Sequence of Time Parameters in ∗ 5 Mode

Key	Display ID:DATA	Description
∗ 5	HH:MM:SS	Display current time
A	Year XXXX	Display/enter year
A	Day of Year XXXX or MMDD XXXX	Display/enter day of year 1-365(366) or press D and enter as month and day, such as 1012 for October 12. D toggles back to Day of Year.
A	Time HHMM	Display/enter hours:minutes
A	Seconds SS	Display/enter seconds

1.3 DISPLAYING/ALTERING INPUT MEMORY, FLAGS, AND PORTS - ✳ 6 MODE

The ✳ 6 Mode is used to display and/or change Input Storage values and to toggle and display user flags and ports. If the ✳ 6 Mode is entered immediately following any changes in program tables, the program will be compiled and run.

> **NOTE:** Input Storage data and the state of flags, control ports, and the timer (Instruction 26) are UNALTERED whenever program tables are altered and recompiled with the ✳ 6 Mode. Compiling always zeros Intermediate Storage.

TABLE 1.3-1. ✳ 6 Mode Commands

Key	Action
∗ 6	Enter ✳ 6 mode
A	Advance to next input location or enter new value
B	Back-up to previous location
C	Change value in first input location on display (followed by keyed in value, then "A")
D	Display/alter user flags 1 through 8
1	Display/alter user flags 11 through 18
0	Display/alter ports
#	Display current location and allow a location number to be keyed in, followed by "A" to jump to that location

1.3.1 DISPLAYING AND ALTERING INPUT STORAGE

When "∗6" is entered, the keyboard/display will read "Mode 06 Enter Loc". One can advance to view the value stored in input location 1 by keying "A". To go directly to a specific location, key in the location number before keying "A". For example, to view the value contained in Input Storage location 20, key in "*6 20 A". The left portion of the display shows the location number and the 9-character label assigned to that location in the programming portion (EDLOG) of Campbell Scientific's PC208W datalogger support software. If the value stored in the location being monitored is the result of a program instruction, the value on the display will be the result of the most recent scan and will be

updated each time the instruction is executed. When using the ⁂ 6 Mode from a remote terminal, a number (any number) must be sent before the value shown will be updated.

Input locations can be used to store parameters for use in computations. To store a value in a location, or change the current value, key "C" while monitoring the location, followed by the desired number and "A".

If an algorithm requires parameters to be manually modified during execution of the Program without interruption of the Table execution process, the ⁂ 6 Mode can be used. (If parameters will not need modification, it is better to load them from the program using Instruction 30.) If initial parameter values are required to be in place before program execution commences, use Instruction 91 at the beginning of the program table to prevent the execution until a flag is set (see the next section). Initial parameter values can be entered into input locations using the ⁂ 6 Mode C command. The flag can then be set to enable the table(s).

If the program is altered and compiled with ⁂ 0 Mode, all values previously entered via ⁂ 6 C will be set to zero. To preserve ⁂ 6 C entered values, compile with ⁂ 6 after changing the program.

1.3.2 DISPLAYING AND TOGGLING USER FLAGS

If D is keyed (for Flags 1 to 8), or 1 is keyed (for Flags 11 to 18) while the CR23X is displaying a location value, the current status of the user flags will be displayed in the following format: "0001.0010". The characters represent the flags, the left-most digit is Flag 1 (or 11) and right most is Flag 8 (or 18). A "0" indicates the flag is low and a "1" indicates the flag is high. In the above example, Flags 4 (or 14) and 7 (or 17) are set. To toggle a flag, simply press the corresponding number. To return to displaying the input location, press "A".

Entering appropriate flag tests into the program allows manual control of program execution. For example, to manually start the execution of Table 2: enter Instruction 91 as the first instruction in Table 2. The first parameter is 25 (do if Flag 5 is low), the second parameter is 0, go to end of program table. If Flag 5 is low, all subsequent instructions in Table 2 will be

skipped. Flag 5 can be toggled from the ⁂ 6 Mode, effectively starting and stopping the execution of Table 2.

1.3.3 DISPLAYING AND TOGGLING PORTS

> **NOTE:** The switched 12 V port is displayed as "control port 9." Other port options are not available on the switched 12 volt channel.

The status of the CR23X ports can be displayed by hitting "0" while looking at an input location (e.g., ⁂ 6 A 0). Ports are displayed left to right as SW12, C8, C7, ... , C1 (opposite to the flags). A port configured as output can be toggled by hitting its number while in the port display mode. There is no effect on ports configured as inputs.

On power up all ports are configured as inputs. Instruction 20 is used to configure a port as an output. Ports are also configured as outputs by any program control commands which uses the port as an output (pulse, set high, set low, toggle).

1.4 COMPILING AND LOGGING DATA - ⁂ 0 MODE

When the ⁂ 0 Mode is entered after programming the CR23X, the program is compiled and the display shows "Running Table" followed by the active program table numbers. The display is not updated after entering ⁂ 0 .

> **NOTE:** All output ports are set low, the timer is reset, and data values in Input and Intermediate Storage are RESET TO ZERO whenever the program tables are altered and the Program is recompiled with the ⁂ 0 Mode. The same is true when the programs are compiled with ⁂ B or ⁂ D .

To minimize current drain, the CR23X should be left in the ⁂ 0 Mode when logging data, and by turning off the display by pressing # .

1.5 MEMORY ALLOCATION - ⁂

1.5.1 INTERNAL MEMORY

When powered up, "Hello" is displayed while a self check is performed. The total system memory is then displayed in K bytes. The size

of memory can be displayed in the ⌗∗⌗ ⌗0⌗ mode. A "--" after the number displayed means that the memory test was aborted. The number shown indicates how far the test progressed before aborted.

Input Storage is used to store the results of Input/Output and Processing Instructions. The values stored in input locations may be displayed using the ⌗∗⌗ ⌗6⌗ Mode (Section 1.3).

Intermediate Storage is a scratch pad for Output Processing Instructions. It is used to store the results of intermediate calculations necessary for averages, standard deviations, histograms, etc. Intermediate Storage is not accessible by the user.

Final Storage holds stored data for a permanent record. Output Instructions store data in Final Storage when the Output Flag is set (Section 3.7). The data in Final Storage can be monitored using the ⌗∗⌗ ⌗7⌗ Mode (Section 2.3).

Each Input or Intermediate Storage location requires 4 bytes of memory. Each Final Storage location requires 2 bytes of memory. Low resolution data points require 1 Final Storage location and high resolution data points require 2. Section 2 describes Final Storage and data retrieval in detail.

Figure 1.5-1 lists the basic memory functions and the amount of memory allotted to them.
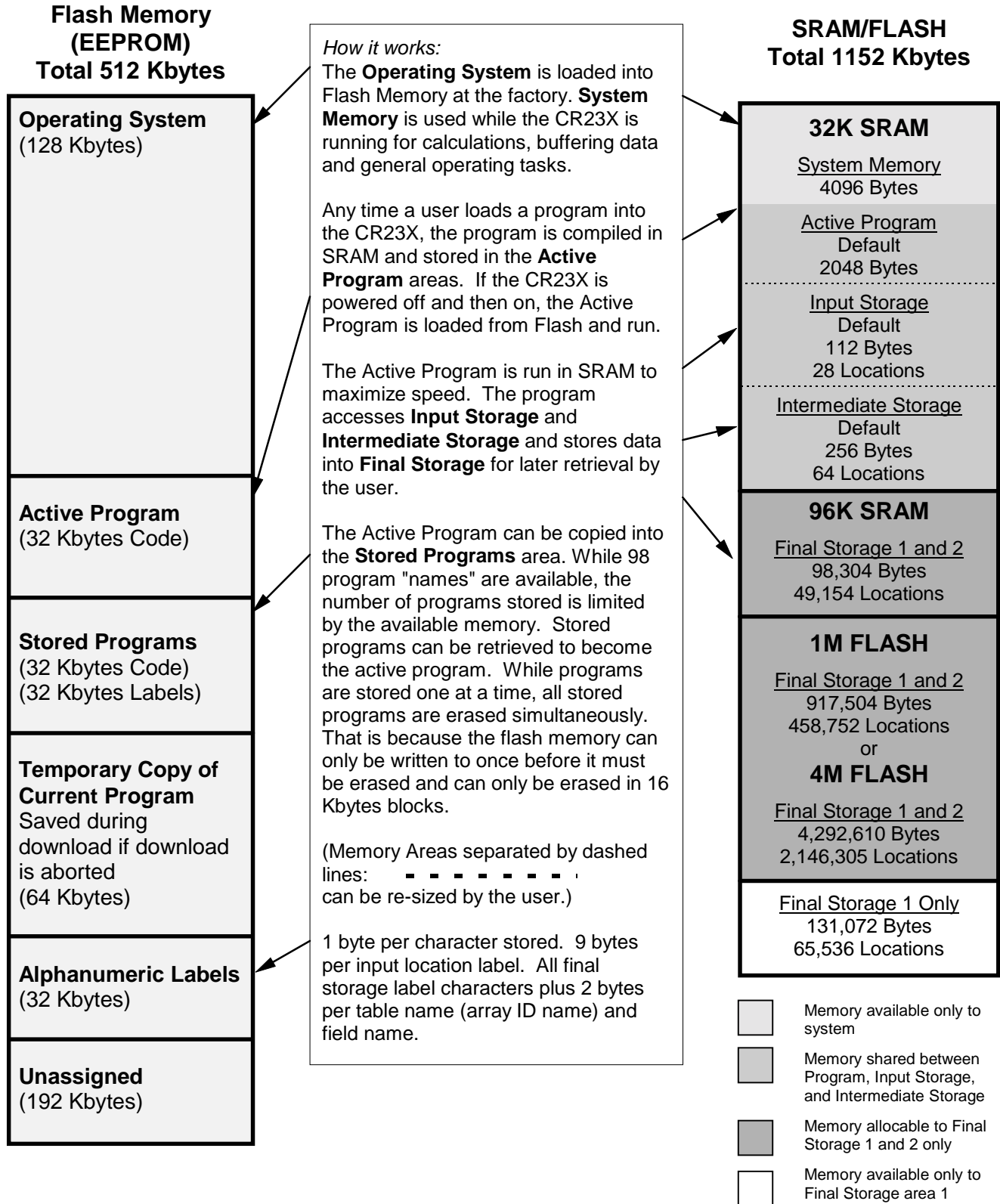
Flash Memory (EEPROM) Total 512 Kbytes

Operating System
(128 Kbytes)

Active Program
(32 Kbytes Code)

Stored Programs
(32 Kbytes Code)
(32 Kbytes Labels)

Temporary Copy of Current Program
Saved during download if download is aborted
(64 Kbytes)

Alphanumeric Labels
(32 Kbytes)

Unassigned
(192 Kbytes)

How it works:

The **Operating System** is loaded into Flash Memory at the factory. **System Memory** is used while the CR23X is running for calculations, buffering data and general operating tasks.

Any time a user loads a program into the CR23X, the program is compiled in SRAM and stored in the **Active Program** areas. If the CR23X is powered off and then on, the Active Program is loaded from Flash and run.

The Active Program is run in SRAM to maximize speed. The program accesses **Input Storage** and **Intermediate Storage** and stores data into **Final Storage** for later retrieval by the user.

The Active Program can be copied into the **Stored Programs** area. While 98 program "names" are available, the number of programs stored is limited by the available memory. Stored programs can be retrieved to become the active program. While programs are stored one at a time, all stored programs are erased simultaneously. That is because the flash memory can only be written to once before it must be erased and can only be erased in 16 Kbytes blocks.

(Memory Areas separated by dashed lines: ▪ ▪ ▪ ▪ ▪ ▪ ▪ ▪ can be re-sized by the user.)

1 byte per character stored. 9 bytes per input location label. All final storage label characters plus 2 bytes per table name (array ID name) and field name.

SRAM/FLASH Total 1152 Kbytes

32K SRAM

System Memory
4096 Bytes

Active Program
Default
2048 Bytes

Input Storage
Default
112 Bytes
28 Locations

Intermediate Storage
Default
256 Bytes
64 Locations

96K SRAM

Final Storage 1 and 2
98,304 Bytes
49,154 Locations

1M FLASH

Final Storage 1 and 2
917,504 Bytes
458,752 Locations
or
4M FLASH

Final Storage 1 and 2
4,292,610 Bytes
2,146,305 Locations

Final Storage 1 Only
131,072 Bytes
65,536 Locations

Memory available only to system

Memory shared between Program, Input Storage, and Intermediate Storage

Memory allocable to Final Storage 1 and 2 only

Memory available only to Final Storage area 1

FIGURE 1.5-1. CR23X Memory

1.5.2 ⁕ A MODE

```
┌─────────────────────────────────────────┐
│ CAUTION:  Reallocating memory will result │
│ in all data being lost.                   │
└─────────────────────────────────────────┘
```

The ⁕ A Mode is used to 1) determine the number of locations allocated to Input Storage, Intermediate Storage, Final Storage Area 2, Final Storage Area 1, and Program Memory; 2) repartition this memory; 3) check the number of bytes remaining in Program memory; 4) erase Final Storage; and 5) to completely reset the datalogger.

A second Final Storage area (Storage Area 2) can be allocated in the ⁕ A Mode. The default number of locations allocated for Storage Area 2 is 0. Final Storage Area 1 is the source from which memory is taken when Final Storage Area 2 is increased. When Final Storage Area 2 is reduced, Final Storage Area 1 memory is increased.

When ⁕ A is entered, the first number displayed is the number of memory locations allocated to Input Storage. The "A" key is used to advance through the next 6 windows. Table 1.5-2 describes what the values in the ⁕ A Mode represent.

Memory allocation defaults at reset to the values in Table 1.5-1.

The sizes of Input, Intermediate, Final Storage Area 2, and Program Memory may be altered by keying in the desired value and entering it by keying "A".

The maximum size of Input and Intermediate Storage and the minimum size of Final Storage are determined by the memory installed (Table 1.5-1). A minimum 64 Input location and 65,536 Final Storage Area 1 locations will ALWAYS be retained. The size of Intermediate Storage may be reduced to 0.

TABLE 1.5-2. Description of ⁕ A Mode Data

Keyboard Entry	Display ID: Data	Description of Data
⁕ A	01: Input Locations XXXX	**Input Storage Locations.** Default 28, minimum of 1, maximum of 7138. This value can be changed by keying in the desired number.
A	02: Intermediate Locs XXXX	**Intermediate Storage Locations.** Default 64, maximum of 7137. This value can be changed by keying in the desired number. **Enter 0 then recompile, and the CR23X will assign the exact number needed**. Entering 0 may also result in the CR23X erasing all data whenever the program is changed and compiled.
A	03: Final Storage 2 XXXXX	**Final Storage Area 2 Locations** (CR23X-4M). Default 0, minimum of 0, maximum of 507,905 (2,080,769). Valid inputs are 0…32,769 or 49,153 and 32,768*N, where N is an integer. Changing this number automatically reallocates Final Storage Area 1.
A	04: Final Storage 1 XXXXX	**Final Storage Area 1 Locations** (CR23X-4M). Default 573,441 (2,146,305), minimum of 65,536, maximum of 573,441 (2,146,305). This number is automatically altered when the memory allocation for Final Storage Area 2 is changed.
A	05: Alloc. Program Bytes +XXXXX	**Bytes allocated for user program.** Default 2048, minimum 116, maximum 28,552. The number of bytes to assign to program memory can be keyed in to change the size of program memory. Changing the size of program memory results in all data being erased. **Enter 0 and the CR23X will assign the exact number needed above 116**. Entering 0 will also result in the CR23X erasing all data whenever the program is changed and compiled. **Key in 98765 to completely reset datalogger.**

A	06: Prog. Bytes Unused +XXXXX	**Bytes free in program memory.** The user cannot change this window. It is a function of window 5 and the program.
A	07: Prog. Bytes Available +XXXXX	The user cannot change this window. It is a function of Window 5 and total available memory.
A	08: Label Bytes Used +XXXXX	The user cannot change this window. It is a function of the program.
A	09: Label Bytes Free +XXXXX	The user cannot change this window. It is a function of Window 8 and the program.

Input Storage, Intermediate Storage, and Final Storage are erased when memory is repartitioned. This feature may be used to clear memory without altering programming. The number of locations does not actually need to be changed; the same value can be keyed in and entered.

If Intermediate Storage size is too small to accommodate the programs or instructions entered, the "E:04" ERROR CODE will be displayed in the ⁎ 0 , ⁎ 6 , and ⁎ B Modes. The user may remove this error code by entering a larger value for Intermediate Storage size. Intermediate Storage and Program Memory can be automatically allocated by entering 0 for their size. When automatic allocation is used, all data are erased any time the program is exchanged and recompiled. Final Storage size is maximized by limiting Intermediate Storage and Program Memory to the minimum necessary. The size of Final Storage and the rate at which data are stored determines how long it will take for Final Storage to fill, at which point new data will write over old.

After repartitioning memory, the program must be recompiled. Compiling erases Intermediate Storage. Compiling with ⁎ 0 erases Input Storage; compiling with ⁎ 6 leaves Input Storage unaltered.

ENTERING 98765 for the number of bytes to allocate for program memory COMPLETELY RESETS THE CR23X. All memory is erased including any stored programs and memory is checked. Memory allocation returns to the default. The reset operation requires approximately 5 minutes for a CR23X. Memory reset can be aborted by pressing any key on the keypad, or raising the ring line high.

1.6 MEMORY TESTING AND SYSTEM STATUS - ⁎ B

The ⁎ B Mode is used to check the status of the program's operating system, memory, and lithium battery. Table 1.6-1 describes what the values seen in the ⁎ B Mode represent.

A signature is a number which is a function of the data and the sequence of data in memory. It is derived using an algorithm which assures a 99.998% probability that if either the data or its sequence changes, the signature changes. The algorithm used to calculate the signature is described in Appendix C.

The signature of the program memory is used to determine if the program tables have been altered. The program signature is calculated only at compile time. In the background FLASH memory of the program is periodically checked against RAM memory of the program. If a byte is different, an E08 watchdog error is flagged.

During the self check on reset, the signature computed for the OS is compared with a stored signature to determine if a failure has occurred. The operating system (OS) signature is calculated in the background of 8 bytes per second and is updated at least once every three days. It is also done when memory is reset or a new operating system is downloaded.

The contents of windows 6 and 7, Operating System (OS) version and version revision, are helpful in determining what OS is in the datalogger. As different versions are released, there may be operational differences. When calling Campbell Scientific for datalogger assistance, please have these numbers available.

Window 13 is a real time display (updated every 0.1 seconds) of the "Program Time", the time it takes Table 1 to execute. The resolution is 0.407 µs, and the range is 6.826 seconds. To read this time as part of a datalogger program, see the description for Instruction P130.

TABLE 1.6-1. Description of ⌗ B **Mode Data**

Keyboard Entry	Display ID: Data	Description of Data
⌗ B	01: +XXXXX	Program memory Signature. The value is dependent upon the programming entered and memory allotment. If the program has not been previously compiled, it will be compiled and run.
A	02: +XXXXX	Operating System (OS) Signature
A	03: XXXXX	Memory Size, Kbytes (Flash + SRAM). "--" indicates that a full memory reset was aborted.
A	04: XX	Number of E08 occurrences (Key in 88 to reset)
A	05: XX	Number of table overrun occurrences (Key in 88 to reset)
A	06: +X.XXXX	Operating System version number
A	07: XXXX.	Version revision number
A	08: +X.XXXX	Lithium battery voltage (measured daily)
A	09: XX	Low 12 V battery detect counter (Key in 88 to reset)
A	10: XX	Extended memory error counter (Key in 88 to reset)
A	11: +X.XXXX	Extended Memory time of erase, seconds. If >5 at room temperature, flash memory may be wearing out. Contact CSI for replacement information.
A	12: XX	Low 5 V counter (Key 88 to reset)
A	13: +X.XXXX	Program Time (0.407 μsec resolution, range is 6.826 seconds, above 6.826, time = 6.826 + displayed value)
A	14: +X.XXXX	Panel Temperature (updated at least every 1 to 2.8 minutes)
A	15: XX	Coprocessor Revision
A	16: XX	Coprocessor Status
A	17: XX	CPLD Revision

TABLE 1.7-1. [*] [C] Mode Entries

SECURITY DISABLED

Keyboard Entry	Display ID: Data	Description
[*] [C]	01: XXXX	Non-zero password blocks entry to [*] [1] , [*] [2] , [*] [3] , [*] [A] , and [*] [D] Modes, telecommunication S command.
[A]	02: XXXX	Non-zero password blocks [*] [4] , [*] [5] , and [*] [6] except for display.
[A]	03: XXXX	Non-zero password blocks [*] [5] , [*] [6] , [*] [7] , [*] [8] , [*] [9] , [*] [B] , and all telecommunications commands except A, L, N, and E.

SECURITY ENABLED

Keyboard Entry	Display ID: Data	Description
[*] [C]	12: 0000	Enter password. If correct, security is temporarily unlocked through that level.
[A]	01: XX	Level to which security has been disabled. 0 -- Password 1 entered (everything unlocked) 1 -- Password 2 entered 2 -- Password 3 entered

1.7 [*] [c] MODE -- SECURITY

The [*] [C] Mode is used to block access to the user's program information and certain CR23X functions. There are 3 levels of security, each with its own 4 digit password. Setting a password to a non-zero value "locks" the functions secured at that level. The password must subsequently be entered to temporarily unlock security through that level. Passwords are part of the program. If security is enabled in the active program, it is enabled as soon as the program is run when the CR23X is powered up.

When security is disabled, [*] [C] will advance directly to the window containing the first password. A non-zero password must be entered in order to advance to the next window. Leaving a password 0, or entering 0 for the password disables that and subsequent levels of security.

Security may be temporarily disabled by entering a password in the [*] [C] Mode or using the telecommunications L command (Section 5.1). The password entered determines what operations are unlocked (e.g., entering password 2 unlocks the functions secured by passwords 2 and 3). Password 1

(everything unlocked) must be entered before any passwords can be altered.

When security is temporarily disabled in the [*] [C] Mode, entering [*] [0] will automatically re-enable security to the level determined by the passwords entered.

The telecommunications L command temporarily changes the security level. After hanging up, security is reset.

1.8 [*] [D] MODE -- SAVE OR LOAD PROGRAM

The [*] [D] Mode is used to save or load CR23X programs, to set the degree to which memory is cleared on powerup, to set the datalogger ID, to set communication to full or half duplex, and to set the display's contrast level.

Programs ([*] [1] , [*] [2] , [*] [3] , [*] [4] , [*] [A] , [*] [C] , and [*] [D] Mode data) may be stored to and from computers, internal flash memory, and Storage Modules. Several programs can be stored in the CR23X Flash Memory and later recalled and run using the [*] [D] Mode or Instruction 111.

SECTION 1. FUNCTIONAL MODES

PC208W automatically makes use of the ✱ D Mode to upload and download programs from a computer. Appendix C gives some additional information on Commands 1 and 2 that are used for these operations.

When "✱D" is keyed in, the CR23X will display "13: Enter Command". A command (Table 1.8-1) is entered by keying the command number and "A".

TABLE 1.8-1. ✱ D Mode Commands

Command	Description
1	Send (Print) ASCII Program
2	Load ASCII Program, ✱ 0 Compile
2--	Load ASCII Program, ✱ 6 Compile (canceled by ✱ D 1 0 mode)
6	Store Program in Flash
7	Load Program from Flash
7N	Save/Load/Clear Program from Storage Module N
8	Set Datalogger ID
9	Set Full/Half Duplex
10	Set Powerup Options
11	Set Display Contrast Level
12	Set Initial Baud/Set RS232 Power
13	Set Compile Option

If the CR23X program has not been compiled when the command to save a program is entered, it will be compiled before the program is saved. When a program is loaded, it is immediately compiled and run. When a command is complete, "13:0000" is displayed; ✱ D must be entered again before another command can be given.

If a program download is aborted, the CR23X will reload the program in its flash into RAM, compile it, and run it.

TABLE 1.8-2. Program Load Error Codes

E 94	Program Storage Area full
E 95	Program does not exist in flash
E 96	Storage Module not connected or wrong address
E 97	Data not encountered within 30 sec.
E 98	Uncorrectable errors detected
E 99	Wrong type of file or Editor Error

1.8.1 INTERNAL FLASH PROGRAM STORAGE

Several programs can be stored in the CR23X Flash Memory and later recalled and run using the ✱ D Mode. The Flash Electrically Erasable Programmable Read Only Memory is non-volatile memory that can only be erased in 16K blocks. The CR23X has 512K of Flash EEPROM memory, one 16K block is reserved for storing extra programs.

When a program is loaded and compiled, it is saved as the active program. The active program will be automatically loaded and run when the CR23X is powered up. Automatic loading of the program can be aborted by pressing any key while "Hello" is showing on the CR23X display; the display will show "Program Aborted." (If a Storage Module with a program 8 is connected when the CR23X powers-up, the Storage Module program 8 will be loaded into the CR23X and become the active program.)

The active program can be stored in internal flash memory program storage with ✱ D command 6 (Table 1.8-3). Programs can be retrieved with ✱ D command 7 (Table 1.8-4).

TABLE 1.8-3. Storing Program in Internal Flash

Key entry	Display
✱ D	13: Enter Command 00
6 A	06: Program ID 00

You may now enter one of the following options:

xx A	Save active program as number xx, xx may be 1-98.
A B	Scroll forward and backward through saved program numbers. The numbers are displayed in the order saved.
9 9 A 9 9 A	Clear all saved programs.
0 A	Display number of bytes free in saved program area.

TABLE 1.8-4. Retrieving a Program from Internal Flash

Key entry   **Display**
⁜ D   13: Enter Command
00
7 A   07: Program ID
00

You may now enter one of the following options:

xx A   Retrieve program number xx (the most recent xx saved). To have the program compile like ⁜ 6 (no resetting of input locations, flags, or ports) press C (xx--) before A.

0 A   Erase active program (i.e., load a blank program; memory allocation and Final Storage are reset).

A
B   Scroll forward and backward through saved program numbers.

Scrolling through the program names begins with the oldest program. "A" advances to the next newer program, "B" backs up to the next older program. While scrolling, at any time typing in a number (xxA) will cause a save or a retrieve operation.

Each program saved takes up the memory required for the program + 6 bytes.

Flash memory can only be written to once before being erased. Because it can only be erased in 16K blocks, if one stored program is to be erased, all must be erased. To allow revising a program and storing it with the same number (name) as an earlier version, the same number can be used by more than one saved program. When retrieving a program, the programs are searched beginning with the last program saved; the most recently saved version will be retrieved. An older program with a duplicate name cannot be retrieved. When the flash program memory is full, all programs must be erased before any more can be added (error 94 will be displayed).

1.8.2 PROGRAM TRANSFER WITH STORAGE MODULE

Storage Modules can store up to eight separate programs. The Storage Module and Keyboard/Display or Modem/Terminal must both be connected to the CR23X. After keying

⁜ D , the command 7N, is entered (N is the Storage Module address 1-8, Section 4.4.1). Address 1 will work with any Storage Module address; the CR23X will search for the lowest address Storage Module that is connected. The command to save, load, or clear a program and the program number (Table 1.8-5) is entered. After the operation is finished "13:0000" is displayed. Error 96 indicates that the Storage Module is not connected or the wrong address was given.

TABLE 1.8-5 Transferring a Program using a Storage Module

Key entry   **Display**
⁜ D   13: Enter Command
00
7 N A   7N: Save, Load, Clr
00

(N is Storage Module address 1-8)

You may now enter one of the following options:

1 x   Save Program x to Storage Module (x = 1-8)
2 x   Load Program x from Storage Module (x = 1-8)
3 x   Erase Program x in Storage Module (x = 1-8)

The datalogger can be programmed on power-up using a Storage Module. If a program is stored as program number 8, and the Storage Module is connected to the datalogger I/O at power-up, program number 8 is automatically loaded into the active program area of the datalogger and run.

1.8.3 SET DATALOGGER ID

Command 8 is used to set the datalogger ID. The ID can be moved to an input location with Instruction 117 and can then be sampled as part of the data.

TABLE 1.8-6 Setting Datalogger ID

Key Entry   **Display**
⁜ D   13: Enter Command
00
8 A   08: Datalogger ID
0XXX

Where XXX are 0s or the current ID. You may now key in the ID (1-12, 14-254).

1.8.4 FULL/HALF DUPLEX

The ✳ D Mode can also be used to set communications to full or half duplex. The default is full duplex, which works best in most situations.

TABLE 1.8-7. Setting Duplex

Key entry	Display
✳ D	13: Enter Command 00
9 A	09: Comm Duplex 0x

If x=0 the CR23X is set for full duplex.
If x=1 the CR23X is set for half duplex.

You may now change the option:

0 A	Set full duplex
1 A	Set half duplex

1.8.5 SETTING POWERUP OPTIONS

Setting options for the Program on Powerup allows the user to specify what information to retain from when the datalogger was last on. This allows Flag/Port status, the User Timer, and the Input/Intermediate Storage to be cleared or not cleared.

Table 1.8-8. Setting Powerup Options

Key entry	Display
✳ D	13: Enter Command 00
1 0 A	10: Power Up Option 0X

Where X is the powerup option currently selected. You may now change the option:

0 A	Clears input locations, ports, flags, user timer, and intermediate storage locations.
1 A	Clears intermediate storage only (leaves Input Storage, Flags/Ports, and User Timer as is).
2 A	Doesn't clear anything.

1.8.6 SETTING DISPLAY CONTRAST

The CR23X automatically adjusts the LCD display contrast for temperature within two seconds after power-up. If necessary, the user can fine tune the default contrast in the ✳ D mode. The user entered adjustment is valid only for the specific temperature range wherein the adjustment was made. If the CR23X temperature moves out of that range, the default setting for the next range controls the contrast. See "Telecommunications" R command information on changing default settings in each temperature range.

TABLE 1.8-9.

Key Entry	Display	Comments
✳ D	13: Enter Command 00	
1 1 A	11: Program Stopped 11: Dsply Contrast xxxx	xxxx is the current setting. Key in new setting followed by an A or . . .
A	11: D Dark C Light xxxx	Press A to darken, B to lighten
✳ 0	LOG	Save setting, restart program

1.8.7 SET INITIAL BAUD / SET RS232 POWER

Table 1.8-11 shows the option codes available for setting the initial baud rate. Setting the initial baud rate forces the CR23X to try the selected baud rate first when connecting with a device. By indexing the option, the "Computer RS232" port can be powered up. Power up of the RS232 port puts 9 volts on pins 1 (DTR) and 8 (RTS), and 8 volts on pin 2 (TX).

TABLE 1.8-10. Set Initial Baud Rate / Set RS232 Power

Key Entry	Display	Comments
`*` `D`	13:Enter Command 00	
`1` `2` `A`	12: Connect Baud Rate 00	
`X` `C` `A`	12: Connect Baud Rate 0X--	Enter Baud Rate Code X (Table 1.8-11). Index (--) is optional.

TABLE 1.8-11. Baud Rate Codes

X = 0   300 Baud
X = 1   1200 Baud
X = 4   2400 Baud
X = 5   4800 Baud
X = 2   9600 Baud
X = 6   19.2 K Baud
X = 7   38.4 K Baud
X = 3   76.8 K Baud
X-- = RS232 Power On

TABLE 1.8-12. Set Program Compile Option

Key Entry	Display	Comments
`*` `D`	13:Enter Command 00	
`1` `3` `A`	13: Compile Option 00	
`1` `A`	13: Compile Option 01	Sets Compile like `*` `6`

TABLE 1.8-13. Compile Option Codes

0	Compile like `*` `0` (See Section 1.4)
1	Compile like `*` `6` (See Section 1.3)
2	Do not clear intermediate storage

1.8.8 SET PROGRAM COMPILE OPTION

Table 1.8-13 shows the option codes available for setting the program compile option. This setting will affect the program compile when the program is downloaded from the PC or a SM192/716 Storage Module. It also affects compiling with `*` `B` and the arcane `*` `D` `7` command. Keyboard or Remote Keyboard compiling with `*` `6` and `*` `0` is not affected by this setting. If a .DLD file has this setting, it will affect the compile operation AFTER the .DLD file is downloaded.

SECTION 2. INTERNAL DATA STORAGE

2.1 FINAL STORAGE AREAS, OUTPUT ARRAYS, AND MEMORY POINTERS

Final Storage is the memory where final processed data are stored. Final Storage data are transferred to your computer or external storage peripheral.

The size of Final Storage is expressed in terms of memory locations or bytes. A low resolution data point (4 decimal characters) occupies one memory location (2 bytes), whereas a high resolution data point (5 decimal characters) requires two memory locations (4 bytes). Table 1.5-1 shows the default allocation of memory locations to Program, Input, Intermediate, and the two Final Storage areas. The ✱ Ⓐ Mode is used to reallocate memory or erase Final Storage (Section 1.5).

The default size of Final Storage with standard memory is 586,568 low resolution memory locations.

Final Storage can be divided into two parts: Final Storage Area 1 and Final Storage Area 2.

Final Storage Area 1 is the default storage area and the only one used if the operator does not specifically allocate memory to Area 2.

Two Final Storage Areas may be used to:

1. Output different data to different devices.

2. Separate archive data from real time display data. In other words, you can record a short time history of real time data and separately record long term, archive data.

3. Record both high speed data (fast recording interval) and slow data without having the high speed data write over the slow data.

Each Final Storage Area can be represented as ring memory (Figure 2.1-1) on which the newest data are written over the oldest data.

The Data Storage Pointer (DSP) is used to determine where to store each new data point in the Final Storage area. The DSP advances to the next available memory location after each new data point is stored.
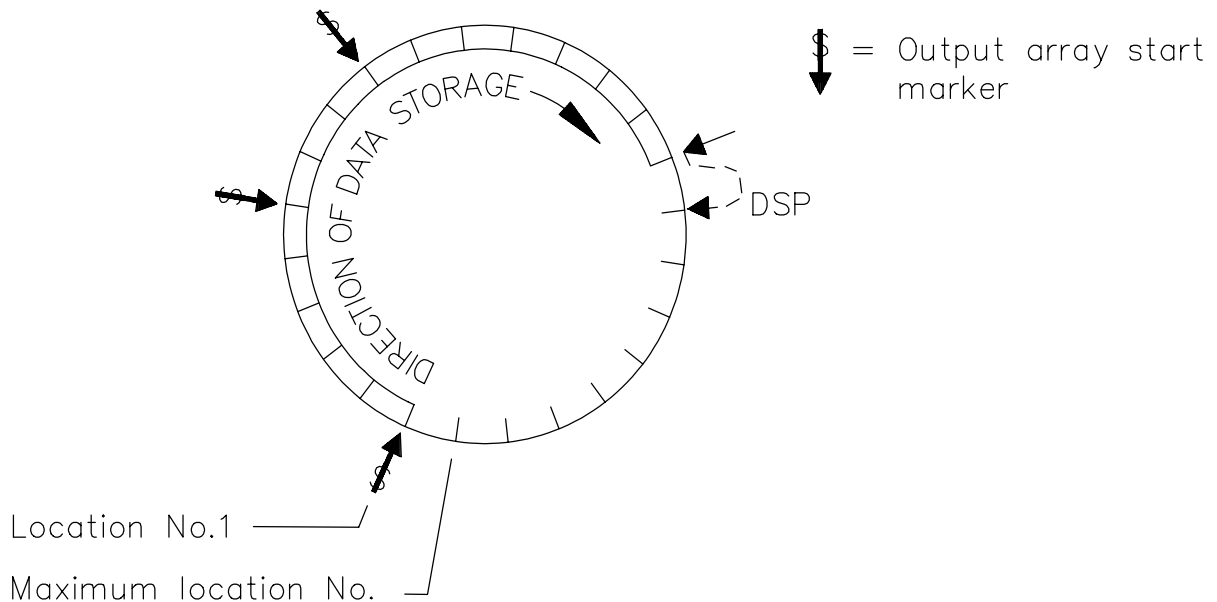

FIGURE 2.1-1. Ring Memory Representation of Final Data Storage

SECTION 2. INTERNAL DATA STORAGE

Output Processing Instructions store data into Final Storage only when the Output Flag is set. The string of data stored each time the Output Flag is set is called an **OUTPUT ARRAY**. The first data point in the output array is a 3 digit **OUTPUT ARRAY ID**. This ID number is set in one of two ways:

1. In the default condition, the ID consists of the program table number and the Instruction Location Number of the instruction which set the Output Flag for that particular array of data. For example, the ID of 118 in Figure 2.1-2 indicates that the 18th instruction in Table 1 set the Output Flag.

2. The output array ID can be set by the user with the second parameter of Instruction 80 (Section 11). The ID can be set to any positive integer up to 511. This option allows the user to make the output array ID independent of the programming. The program can be changed (instructions added or deleted) without changing the output array ID. This avoids confusion during data reduction, especially on long term projects where program changes or updates are likely.
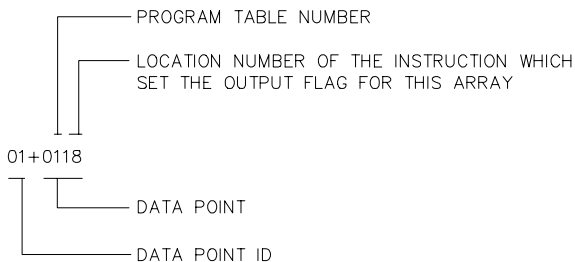
```
        ┌──── PROGRAM TABLE NUMBER

        │  ┌── LOCATION NUMBER OF THE INSTRUCTION WHICH
        │  │   SET THE OUTPUT FLAG FOR THIS ARRAY
        │  │
   01+0118

     │  │
     │  └──── DATA POINT

     └─────── DATA POINT ID
```

FIGURE 2.1-2. Output Array ID

> **NOTE:** If Instruction 80 is used to designate the active Final Storage Area and parameter 2 is 0, the output array ID is determined by the position of Instruction 80 or by the position of the instruction setting the Output Flag, whichever occurs last.

A start-of-array marker ($ in Figure 2.1-1) is written into Final Storage with the Output Array ID. This marker is used as a reference point from which to number the data points of the output array. The start of array marker occupies the same Final Storage location as the Array ID and is transparent for all user operations.

Data are stored in Final Storage before being transmitted to an external device. There are 4 pointers for each Final Storage Area which are used to keep track of data transmission. These pointers are:

1. Display Pointer (**DPTR**)
2. Printer Pointer (**PPTR**)
3. Telecommunications (Modem) Pointer (**MPTR**)
4. Storage Module Pointer (**SPTR**)

The **DPTR** is used to recall data to the keyboard/display. The positioning of this pointer and data recall are controlled from the keyboard (⁎ 7 Mode).

The **PPTR** is used to control data transmission to a printer or other serial device. Whenever on-line printer transfer is activated (Instruction 96), data between the PPTR and DSP are transmitted. The PPTR may also be positioned via the keyboard for manually initiated data transmission (⁎ 8 Mode).

The **MPTR** is used in transmitting data over a telecommunications interface. When telecommunications is first entered, the MPTR is set to the same location as the DSP. Positioning of the MPTR is then controlled by commands from the external calling device (Section 5.1).

The **SPTR** is used to control data transmission to a Storage Module. When on-line transfer is activated by Instruction 96, data is transmitted each time an output array is stored in Final Storage IF THE STORAGE MODULE IS CONNECTED TO THE CR23X. If the Storage Module is not connected, the CR23X does not transmit the data nor does it advance the SPTR to the new DSP location. It saves the data until the Storage Module is connected. Then, during the next execution of Instruction 96, the CR23X outputs all of the data between the SPTR and the DSP and updates the SPTR to the DSP location (Section 4.1)

The SPTR may also be positioned via the keyboard for manually initiated data transfer to the Storage Module (⁎ 8 Mode, Section 4.2).

> **NOTE:** All memory pointers are set to the DSP location when the datalogger compiles a program. ALWAYS RETRIEVE UNCOLLECTED DATA BEFORE MAKING PROGRAM CHANGES.

2.2 DATA OUTPUT FORMAT AND RANGE LIMITS

Data are stored internally in Campbell Scientific's Binary Final Storage Format (Appendix C.2). Data may be sent to Final Storage in either LOW RESOLUTION or HIGH RESOLUTION format.

2.2.1 RESOLUTION AND RANGE LIMITS

Low resolution data is a 2 byte format with 4 significant digits and a maximum magnitude of +6999. High resolution data is a 4 byte format with 5 significant digits and a maximum possible output value of +99999 (see Table 2.2-1 below).

TABLE 2.2-1. Resolution Range Limits of CR23X Data

Resolution	Zero	Minimum Magnitude	Maximum Magnitude
Low	0.000	+0.001	+6999.
High	0.0000	+ .00001	+99999.

The resolution of the low resolution format is reduced to 3 significant digits when the first (left most) digit is 7 or greater. Thus, it may be necessary to use high resolution output or an offset to maintain the desired resolution of a measurement. For example, if water level is to be measured and output to the nearest 0.01 ft., the level must be less than 70 ft. for low resolution output to display the 0.01 ft. increment. If the water level was expected to range from 50 to 80 ft. the data could either be output in high resolution or could be offset by 20 ft. (transforming the range to 30 to 50 ft.).

2.2.2 INPUT AND INTERMEDIATE STORAGE DATA FORMAT

While output data have the limits described above, the computations performed in the CR23X are done in floating point arithmetic. In Input and Intermediate Storage, the numbers are stored and processed in a binary format with a 23 bit binary mantissa and a 6 bit binary exponent. The largest and smallest numbers that can be stored and processed are 9×10^{18} and 1×10^{-19}, respectively. The size of the number determines the resolution of the arithmetic. A rough approximation of the resolution is that it is better than 1 in the seventh digit. For example, the resolution of 97,386,9$\underline{2}$4 is better than 10. The resolution of 0.0086731$\underline{9}$24 is better than 0.000000001.

A precise calculation of the resolution of a number may be determined by representing the number as a mantissa between .5 and 1 multiplied by 2 raised to some integer power. The resolution is the product of that power of 2 and 2^{-24}. For example, representing 478 as .9336 $\ast$ 2^9, the resolution is $2^9 \ast 2^{-24} = 2^{-15} = 0.0000305$. A description of Campbell Scientific's floating point format may be found in the description of the J and K Telecommunications Commands in Appendix C.

2.3 DISPLAYING STORED DATA - ✳ 7 MODE

(Computer/terminal users refer to Section 5 for instructions on entering the Remote Keyboard State.)

Final Storage may be displayed by using the ✳ 7 Mode. Key ✳ 7 .

If you have allocated memory to Final Storage Area 2, the display will show:

> Mode 07: Storage Area
> 00

Select which Storage Area you wish to view:

> 00 or 01 = Final Storage Area 1
> 02 = Final Storage Area 2

If no memory has been allocated to Final Storage Area 2, this first window will be skipped.

The next window displays the current DSP location. Pressing A advances you to the Output array ID of the oldest Array in the Storage Area. To locate a specific Output Array, enter a location number that positions the Display Pointer (DPTR) behind the desired data and press the "A" key. If the location number entered is in the middle of an Output Array, the DPTR is automatically advanced to the first data point of the next Output Array. Repeated use of the "A" key advances through the Output Array. Data and the alphanumeric label assigned by EDLOG are displayed. The "B" key backs the DPTR through memory.

The memory location of the data point is displayed by pressing the "#" key. At this point, another memory location may be entered, followed by the "A" key to jump to the start of

the Output Array equal to or just ahead of the location entered. Whenever a location number is displayed by using the "#" key, the corresponding data point can be displayed by pressing the "C" key.

The same element in the next Output Array with the same ID can be displayed by hitting [#] [A]. The same element in the previous array can be displayed by hitting [#] [B]. If the element is 1 (Array ID), then [#] [A] advances to the next array and [#] [B] backs up to the previous array. [#] [0] [A] backs up to the start of the current array.

The keyboard commands used in the [*] [7] Mode are summarized in Table 2.3-1.

Advancing the DPTR past the Data Storage Pointer (DSP) displays the oldest data point. Upon entering the [*] [7] Mode, the oldest Output Array can be accessed by pressing the "A" key.

TABLE 2.3-1. [*] [7] **Mode Command Summary**

Key	Action
[A]	Advance to next data point
[B]	Back-up to previous data point
[#]	Display location number of currently displayed data point value
[C]	Display value of current location
[#][A]	Advance to same element in next Output Array with same ID
[#][B]	Back-up to same element in previous Output Array with same ID
[#][0][A]	Back-up to the start of the current Final Data Storage Array
[*]	Exit [*] [7] Mode

SECTION 3. INSTRUCTION SET BASICS

The instructions used to program the CR23X are divided into four types: Input/Output (I/O), Processing, Output Processing, and Program Control. I/O Instructions are used to make measurements and store the readings in input locations or to initiate analog or digital port output. Processing Instructions perform mathematical operations using data from Input Storage locations and place the results back into specified Input Storage locations. Output Processing Instructions provide a method for generating time or event dependent data summaries from processed sensor readings residing in specified Input Storage locations. Program Control Instructions are used to direct program execution based on time and or conditional tests on input data and to direct output to external devices.

Instructions are identified by a number. There are a fixed number of parameters associated with each instruction to give the CR23X the information required to execute the instruction. The set of instructions available in the CR23X is determined by the CR23X Operating System.

3.1 PARAMETER DATA TYPES

There are 3 different data types used for Instruction parameters: Floating Point (FP), 4 digit integers (4), and 2 digit integers (2). The parameter data type is identified in the listings of the instruction parameters in Sections 9-12. Different data types are used to allow the CR23X to make the most efficient use of its memory.

Floating Point parameters are used to enter numeric constants for calibrations or mathematical operations. While it is only possible to enter 5 digits (magnitude +.00001 to +99999.), the internal format has a much greater range (1×10^{-19} to 9×10^{18}, Section 2.2.1). Instruction 30 can be used to enter a number in scientific notation into an input location.

3.2 REPETITIONS (Reps)

The repetitions parameter on many of the I/O, Processing, and Output Processing Instructions is used to repeat the instruction on a number of sequential Input Channels or Input Storage locations. For example, if you are making 4 differential voltage measurements on the same voltage range, wire the inputs to sequential channels and enter the Differential Voltage Measurement Instruction once with 4 repetitions, rather than entering 4 separate measurement instructions. The instruction will make 4 measurements starting on the specified channel number and continuing through the 3 succeeding differential channels. The results will be stored in the specified input location and the 3 succeeding input locations. Averages for all 4 measurements can be calculated by entering the Average Instruction with 4 repetitions.

When several of the same type of measurements will be made, but the calibrations of the sensors are different, it requires less time to make the measurements using one measurement with repetitions and then apply the calibrations with a scaling array (Inst. 53) than it does to enter the instruction several times in order to use a different multiplier and offset. This is due to set up and calibration time for each measurement instruction. However, if time is not a constraint, separate instructions may make the program easier to follow.

3.3 ENTERING NEGATIVE NUMBERS

Before or after keying in a number, press Ⓒ or "-" to change the number's sign. On floating point numbers a minus sign (-) will appear to the left of the number. Excitation voltages in millivolts for I/O Instructions are 4 digit integers; when Ⓒ is keyed 2 minus signs (--) will appear to the right of the number indicating a negative excitation. Even though this display is the same as that indicating an indexed input location, (Section 3.4) there is no indexing effect on excitation voltage.

3.4 INDEXING INPUT LOCATIONS AND CONTROL PORTS

When used within a loop, the parameters for input locations and the commands to set, toggle, or pulse a port can be Indexed to the loop counter. The loop counter is added to the indexed value to determine the actual Input Location or Port the instruction acts on. Normally the loop counter is incremented by 1 after each pass through the loop. Instruction 90, Step Loop Index, allows the increment step

to be changed. See Instructions 87 and 90, Section 12, for more details.

To index an input location (4 digit integer) or set port command (2 digit integer) parameter, ⒸC or "-" is pressed after keying the value but before entering the parameter. Two minus signs (--) will be displayed to the right of the parameter.

3.5 VOLTAGE RANGE AND OVERRANGE DETECTION

The voltage RANGE code parameter on Input/Output Instructions is used to specify the full scale range of the measurement and the integration period for the measurement (Table 3.5-1).

The full scale range selected should be the smallest that will accommodate the full scale output of the sensor being measured. Using the smallest possible range will result in the best resolution for the measurement.

Four different integration sequences are possible. The relative immunity of the integration sequences to random noise is: Slow 60 Hz rej. = Slow 50 Hz rej. > Fast 60 Hz rej. = Fast 50 Hz rej. > 250 µs integ. The 60 Hz rejection integration rejects noise from 60 Hz AC line power. The 50 Hz rejection is for countries whose electric utilities operate at 50 Hz (Section 13.1).

When a voltage input exceeds the range programmed, the value which is stored is set to the maximum negative number and displayed as -99999 in high resolution or -6999 in low resolution.

An input voltage greater than +8 volts on one of the analog inputs will result in errors and possible overranging on the other analog inputs. Voltages greater than 16 volts may permanently damage the CR23X.

3.6 OUTPUT PROCESSING

Most Output Processing Instructions have both an Intermediate Data Processing operation and a Final Data Processing operation. For example, when the Average Instruction, 71, is initiated, the intermediate processing operation increments a sample count and adds each new Input Storage value to a cumulative total residing in Intermediate Storage. When the Output Flag is set, the final processing operation divides the cumulative total by the number of samples to find the average. The average is then stored in final storage and the cumulative total and number of samples are set to zero in Intermediate Storage.

Final Storage Area 1 (Sections 1.5, 2.1) is the default destination of data output by Output Processing Instructions. Instruction 80 may be used to direct output to either Final Storage Area 2 or to Input Storage.

Output Processing Instructions requiring intermediate processing sample the specified input location(s) each time the Output Instruction is executed, NOT each time the location value is updated by an I/O Instruction. For example: Suppose a temperature measurement is initiated by Table 1 which has an execution interval of 1 second.

TABLE 3.5-1. Input Voltage Ranges and Codes

Range Code			Full Scale Range	Resolution Differential*
Fast 250 µs Integ.	Slow 60 Hz Reject.	Slow 50 Hz Reject.		
10	20	30	Autorange**	
11	21	31	±10 mV	0.33 µV
12	22	32	±50 mV	1.67 µV
13	23	33	±200 mV	6.66 µV
14	24	34	±1000 mV	33.3 µV
15	25	35	±5000 mV	166. µV

* Differential measurement, resolution for single-ended measurement is twice value shown.

**Autoranging may not adequately measure inputs with extremely noisy or rapidly changing signals. Autoranging channels may occasionally measure on a higher range than is required.

The instructions to output the average temperature every 10 minutes are in Table 2 which has an execution interval of 10 seconds. The temperature will be measured 600 times in the 10 minute period, but the average will be the result of only 60 of those measurements because the instruction to average is executed only one tenth as often as the instruction to make the measurement.

Intermediate Processing can be disabled by setting Flag 9 which prevents Intermediate Processing without actually skipping over the Output Instruction.

All of the Output Processing Instructions store processed data values when and only when the Output Flag is set (Section 3.7.1). The Output Flag (Flag 0) is set at desired intervals or in response to certain conditions by using an appropriate Program Control Instruction (Section 12).

3.7 USE OF FLAGS: OUTPUT AND PROGRAM CONTROL

There are 18 flags which may be used in CR23X programs. Two of the flags are dedicated to specific functions: Flag 0 causes Output Processing Instructions to write to Final Storage, and Flag 9 disables intermediate processing. Flags 1-8 and 11-18 may be used as desired in programming the CR23X. Flags 0 and 9 are automatically set low at the beginning of each execution of the program table. Flags 1-8 and 11-18 remain unchanged until acted on by a Program Control Instruction or until manually toggled from the ⁎ 6 Mode.

TABLE 3.7-1. Flag Description

Flag 0	-	Output Flag
Flag 1 to 8	-	User Flags
Flag 11 to 18	-	User Flags
Flag 9	-	Intermediate Processing Disable Flag

Flags are set with Program Control Instructions. The Output Flag (Flag 0) and the Intermediate Programming Disable Flag (Flag 9) will always be set low if the set high condition fails. The status of flags 1 through 8 and 11 through 18 does not change when a conditional test is false.

3.7.1 THE OUTPUT FLAG

A group of processed data values is placed in Final Data Storage by Output Processing Instructions when the Output Flag (Flag 0) is set high. This group of data is called an Output Array. The Output Flag is set using Program Control Instructions according to time or event dependent intervals specified by the user. The Output Flag is set low at the beginning of each execution of the program table.

Output is most often desired at fixed intervals; this is accomplished with Instruction 92, If Time. Output is usually desired on the even interval, so Parameter 1, time into the interval, is 0. The time interval (Parameter 2), in minutes, is how often output will occur; i.e., the Output Interval. The command code (Parameter 3) is 10, causing Flag 0 to be set high. The time interval is synchronized to 24 hour time; output will occur on each integer multiple of the Output Interval starting from midnight (0 minutes). If the Output Interval is not an even divisor of 1440 minutes (24 hours), the last output interval of the day will be less than the specified time interval. Output will occur at midnight and will resume synchronized to the new day. Instruction 92 is followed in the program table by the Output Instructions which define the Output Array desired.

Each group of Output Processing Instructions creating an Output Array is preceded by a Program Control Instruction that sets the Output Flag.

> **NOTE**: If the Output Flag is already set high and the test condition of a subsequent Program Control Instruction acting on Flag 0 fails, the flag is set low. This eliminates entering another instruction to specifically reset the Output Flag before proceeding to another group of Output Instructions with a different output interval.

3.7.2 THE INTERMEDIATE PROCESSING DISABLE FLAG

The Intermediate Processing Disable Flag (Flag 9) suspends intermediate processing when it is set high. This flag is used to restrict sampling for averages, totals, maxima, minima, etc., to times when certain criteria are met. The flag is automatically set low at the beginning of each execution of the program table.

As an example, suppose it is desired to obtain a wind speed rose incorporating only wind speeds greater than or equal to 4.5 m/s. The wind speed rose is computed using the Histogram Instruction 75, and wind speed is stored in input location 14, in m/s. Instruction 89 is placed just before Instruction 75 and is used to set Flag 9 high if the wind speed is less than 4.5 m/s:

TABLE 3.7-2. Example of the Use of Flag 9

Inst. Loc.	Param. No.	Entry	Description
X	P	89	If wind speed < 4.5 m/s
	1	14	Wind speed location
	2	4	Comparison: <
	3	4.5	Minimum wind speed for histogram
	4	19	Set Flag 9 high
X+1	P	75	Histogram
X+2	P	86	Do
	1	29	Set Flag 9 Low

> **NOTE**: Flag 9 is automatically reset the same as Flag 0. If the intermediate processing disable flag is already set high and the test condition of a subsequent Program Control Instruction acting on Flag 9 fails, the flag is set low. This feature eliminates having to enter another instruction to specifically reset Flag 9 before proceeding to another group of test conditions.

3.7.3 USER FLAGS

Flags 1 through 8 and 11 through 18 are not dedicated to a specific purpose and are available to the user for general programming needs. The user flags can be manually toggled from the keyboard in the ⁎ 6 Mode (Section 1.3) or through telecommunications with PC208W datalogger support software. By inserting the flag test (Instruction 91) at appropriate points in the program, the user can use the ⁎ 6 Mode to manually direct program execution.

3.8 PROGRAM CONTROL LOGICAL CONSTRUCTIONS

Most of the Program Control Instructions have a command code parameter which is used to specify the action to be taken if the condition tested in the instruction is true. Table 3.8-1 lists these codes.

TABLE 3.8-1. Command Codes

0	Go to end of program table[3]
1-9, 79-99	Call Subroutine 1-9, 79-99[1]
10-19	Set Flag 0-9 high
111-118	Set Flag 11-18 high
20-29	Set Flag 0-9 low
211-218	Set Flag 11-18 low
30	Then Do
31	Exit loop if true
32	Exit loop if false
41-48	Set Port 1-8 high[2]
49	Set Switched 12 V high[2]
51-58	Set Port 1-8 low[2]
59	Set Switched 12 V low[2]
61-68	Toggle Port 1-8[2]
71-78	Pulse Port 1-8[2]

[1]   95, 96, 97, and 98 are special subroutines which can be called by Control ports 6, 7, and 8 going high; see Instruction 85 for details (Section 12).

[2]   The ports can be indexed to the loop counter (Section 8.4).

[3]   If this command is executed while in a subroutine, execution jumps directly to the end of the table that called the subroutine.

3.8.1 IF THEN/ELSE COMPARISONS

Program Control Instructions can be used for If then/else comparisons. When Command 30 (Then do) is used with Instructions 83 or 88-92, the If Instruction is followed immediately by instructions to execute if the comparison is true. The Else Instruction (94) is optional and is followed by the instructions to execute if the comparison is false. The End Instruction (95) ends the If then/else comparison and marks the beginning of the instructions that are executed regardless of the outcome of the comparison (see Figure 3.8-1).

```
        ┌───── IF (83 OR 88—92 WITH COMMAND 30)
        │      │
        │      TRUE
        │   ┌──┘
        │   │  ELSE (94)
        │   │
        │   │  FALSE
        │   │    │
        │   │    ▼
        └───┴─▶ END (95)
                 │
                 ▼
```
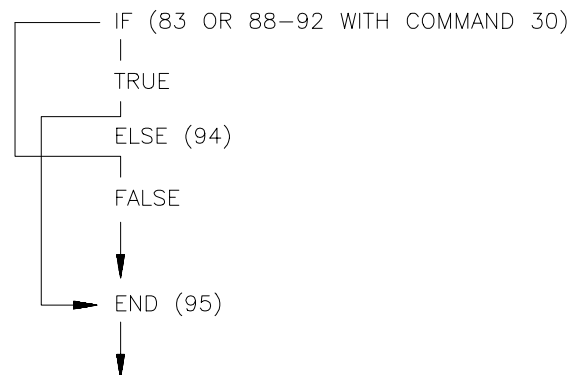
FIGURE 3.8-1. If Then/Else Execution Sequence

```
            ┌────── IF A  (88-92 WITH COMMAND 30)
            │
            │   ┌── IF B  (88-92 WITH COMMAND 30)
            │   │
            │   │   INSTRUCTIONS TO
            │   │   EXECUTE IF A AND B
EITHER FALSE│   │   ARE BOTH TRUE
            │   │
            │   │
            │   └─► END B (95)
            │
            └─────► END A (95)
```
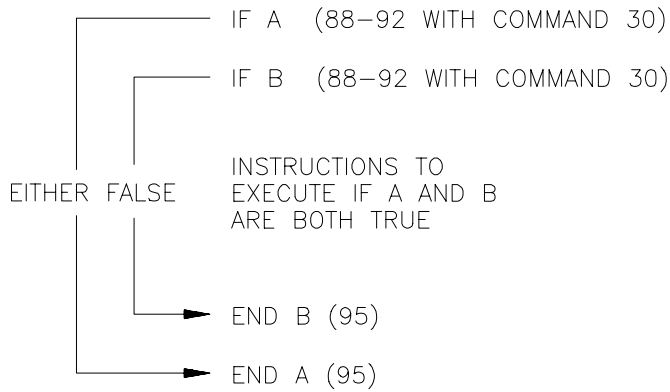
FIGURE 3.8-2. Logical AND Construction

If Then/Else comparisons may be nested to form logical AND or OR branching. Figure 3.8-2 illustrates an AND construction. If conditions A and B are true, the instructions included between IF B and the first End Instruction will be executed.

If either of the conditions is false, execution will jump to the corresponding End Instruction, skipping the instructions between.

A logical OR construction is also possible. Figure 3.8-3 illustrates the instruction sequence that will result in subroutine X being executed if either A or B is true.

```
    IF A  (88-92 with command 30)
        Call subroutine X (86, command=X)
    ELSE  (94)
        IF B  (88-92 with command 30)
            Call subroutine X (86, command=X)
        END B (95)
    END A (95)
```

FIGURE 3.8-3. Logical OR Construction

> **NOTE:** A logical OR can also be constructed by setting a user flag if a comparison is true. (The flag is cleared before making comparisons.) After all comparisons have been made, execute the desired instructions if the flag is set.

The Begin Case Instruction 93 and If Case Instruction 83 allow a series of tests on the value in an input location. The case test is started with Instruction 93 which specifies the location to test. A series of Instruction 83s are

then used to compare the value in the location with fixed values. When the value in the input location is less than the fixed value specified in Instruction 83, the command in that Instruction 83 is executed, and execution branches to the END Instruction 95 which closes the case test (see Instruction 93, Section 12).

3.8.2 NESTING

A branching or loop instruction which occurs before a previous branch or loop has been closed is nested. The maximum nesting level is 11 deep. Loop Instruction 87 and Begin Case Instruction 93 both count as 1 level. Instructions 83, 86, 88, 89, 91, and 92 each count as one level when used with the Command "30" which is the "Then Do" command. Use of Else, Instruction 94, also counts as one nesting level each time it is used. For example, the AND construction above is nested 2 deep while the OR construction is nested 3 deep.

Subroutine calls do not count as nesting with the above instructions, though they have their own nesting limit (maximum of 7, see Instruction 85, Section 12). Branching and loop nesting start at zero in each subroutine.

Any number of groups of nested instructions may be used in any of the three Programming Tables. The number of groups is only restricted by the program memory available.

3.9 INSTRUCTION MEMORY AND EXECUTION TIME

Each instruction requires program memory and uses varying numbers of Input, Intermediate, and Final Storage locations. Tables 3.9-1 to 3.9-4 list the memory used by each instruction and the approximate time required to execute it.

When attempting to make a series of measurements and calculations at a fast rate, it is important to examine the time required for the automatic calibration sequence and possibly make use of the program controlled calibration, Instruction 24. Section 13.9 describes the calibration process.

TABLE 3.9-1. CR23X Input/Output Instruction Execution Time (ms)

Inst.	Desc.	InLoc	Bytes	1-4 or NA	10	Fast 11-12	Fast 13-14	15	20	21-22	60Hz 23-24	25	30	31-32	50Hz 33-34	35	40	Fast 41-42	60Hz 43-45	50	Fast 51-52	50Hz 53-55
1	VOLT (SE)	R	15		(-1.8)+11.8*R	1.5+1.6R	0.5+1.6R	0.5+1.6R	4.4+117.8*R	17.5+18.2R	0.1+18.2R	0.4+9.9R	(-8.3)+140.0*R	21.0+21.6R	0.0+21.6R	0.4+11.6R	7.5+52.5*R	1.4+9.9R	0.4+9.9R	4.7+61.0*R	1.4+11.6R	0.4+11.6R
2	VOLT (DIFF)	R	15		0.4+14.4*R	0.5+2.6R	0.5+2.6R	0.5+2.6R	4.5+161.7*R	0.0+35.9R	0.0+35.9R	0.0+19.7R	3.2+191.9*R	0.0+42.5R	0.0+42.5R	0.0+23.0R	4.4+97.2*R	0.0+19.7R	0.0+19.7R	3.2+114.6*R	0.0+23.0R	0.0+23.0R
3	PULSE	R	15	0.8+0.8R																		
4	EX - DEL. - SE (with delay)	R	20		0.3 + 12.4 * R / 0.3*R*(12.4+5* Delay)	1.6+1.6R	0.6+1.6R	0.6+1.6R	4.5 + 118.5 * R / 4.5+R*(118.5+5* Delay)	17.7+18.3R	0.2+18.3R	0.5+10.0R	3.0 + 140.1 * R / 3.0+R*(140.1+5* Delay)	21.2+21.6R	0.2+21.6R	0.5+11.7R	4.7 + 53.2 * R / 4.7+R*(53.2+5* Delay)	1.5+10.0R	1.5+10.0R	5.4 + 61.6 * R / 5.4+R*(61.6+5* Delay)	1.5+11.7R	0.5+11.7R
5	AC HALF BR	R	18		12.2 + 24.2 * R	3.5+3.0R	2.4+3.2R	2.4+3.2R	3.5 + 205.1 * R	19.2+36.1R	2.2+36.1R	1.9+20.1R	4.6 + 241.4 * R	23.0+42.8R	2.0+42.8R	1.6+23.6R	6.1 + 106.8 * R	3.1+19.9R	1.9+20.1R	4.8 + 123.6 * R	2.8+23.4R	1.6+23.6R
6	FULL BR	R	18		12.0 + 32.8 * R	2.6+5.1R	2.6+5.1R	2.6+5.1R	13.6 + 327.5 * R	0.6+71.3R	1.6+71.6R	1.6+39.1R	2.8 + 389.5 * R	1.7+84.8R	1.7+84.8R	1.5+45.8R	4.9 + 200.4 * R	1.9+39.1R	1.6+39.1R	4.8 + 233.8 * R	2.0+45.6R	1.5+45.8R
7	3W HALF BR	R	18		16.6 + 34.1 * R	4.3+5.8R	3.2+7.0R	3.2+7.0R	148.9 + 314.6 * R	19.1+72.8R	2.7+72.4R	1.5+41.5R	176.0 + 360.9 * R	22.5+86.2R	1.9+86.2R	2.1+47.9R	47.0 + 181.4 * R	1.5+41.5R	1.6+41.5R	56.6 + 210.7 * R	3.2+46.7R	2.1+47.9R
8	EX - DEL. - DIFF (with delay)	R	20		0.4 + 12.2 * R / 0.4*2*R*(12.2+5* Delay)	1.7+1.7R	0.7+1.7R	0.7+1.7R	3.6 + 118.7 * R / 3.6+2*R*(118.7+5* Delay)	17.7+18.3R	0.2+18.3R	0.5+10.0R	3.6 + 140.5 * R / 3.6+2*R*(140.5+5* Delay)	21.2+21.6R	0.3+21.6R	1.3+12.2R	2.9 + 53.9 * R / 2.9+2*R*(53.9+5* Delay)	1.6+10.0R	0.5+10.0R	5.4 + 62.0 * R / 5.4+2*R*(62.0+5* Delay)	1.6+11.7R	0.5+11.6R
9	FULL BR-MEX V1 = 15 / V1 = 25	R	19		15.1 + 45.8 * R	2.0+10.7R / 3.0+10.9R / 0.4+46.1R	1.3+11.1R / 3.0+10.9R / 0.4+46.1R	2.5+11.2R / 3.0+10.9R / 0.4+46.1R	3.9 + 679.3 * R	0.0+144.0R / 1.3+77.8R / 1.3+111.7R	0.0+144.0R / 1.3+77.8R / 1.3+111.7R	0.0+79.2R / 0.0+45.9R / 0.0+79.2R	4.4 + 805.8 * R	0.0+170.5R / 0.8+91.3R / 0.9+125.6R	0.0+171.4R / 0.0+91.7R / 0.9+125.6R	0.0+93.1R / 0.0+52.3R / 0.2+86.5R	78.5 + 332.5 * R	0.4+78.9R / 0.0+45.8R / 0.8+79.4R	0.0+79.2R / 0.0+45.9R / 0.0+79.2R	57.5 + 411.5 * R	0.0+92.3R / 0.0+52.2R / 0.4+86.3R	0.0+93.1R / 0.0+52.2R / 0.2+86.5R
10	BATT VOLT	1	4	2.5																		
11	TEMP (107)	R	15	3.2+3.6R		3.2+3.6R				17.8+56.9R				21.1+63.6R				17.8+56.9R			21.1+63.6R	
12	RH (207)	R	17	3.0+4.3R																		
13	TEMP-TC SE	R	18		(-1.3) + 17.2 * R	3.0+3.6R	2.0+3.6R	2.0+3.7R	5.6 + 98.6 * R	19.2+20.2R	1.7+20.2R	2.0+12.0R	5.6 + 115.2 R	22.6+23.6R	1.7+23.6R	2.0+13.7R	4.0 + 41.5 * R	3.1+11.9R	2.0+12.0R	2.7 + 46.9 * R	3.1+13.6R	2.0+13.6R
14	TEMP-TE DIF	R	18		(-0.5) + 18.1 * R	2.2+4.7R	2.2+4.7R	1.8+4.7R	5.4 + 115.4 * R	1.6+37.9R	1.6+37.9R	1.3+21.7R	5.6 + 134.9 R	1.7+44.6R	1.7+44.6R	1.6+25.0R	3.1 + 67.4 * R	1.6+21.7R	1.6+21.8R	4.3 + 76.9 * R	2.0+25.0R	2.0+25.0R
15	SERIAL I/O	R	15			NOT	READY	YET														
16	TEMP-RTD	R		0.8+1.3R																		
17	TEMP-PANEL	1 OR 5	4	2.8																		
18	TIME	1	7	2																		
19	SIGNATURE		4	40.2																		
20	PORT SET	0	6	6.6																		
21	PULSE PORT	0	10	0.8+10H																		
22	DELAY-EXCITE	0	9	0.5+del(ms)																		
23	BURST	X	35																			
24	CALIBRATION	19	4	0.3																		
25	READ PORT	1	6																			
26	TIMER	1 OR 0	4																			
27	PERIOD AVG	R	19																			
28	VIBR. WIRE	R	21																			
29	INW PS9105	2	88	157																		
100	TDR			2.7+2.7R																		
101	SDM-INT8			1.7+1.0R																		
102	SDM-SW8A			0.9+2.1R																		
103	SDM-AO4																					
104	SDM-CD16																					
105	SDI-12 REC.					NOT	READY	YET														
106	SDI-12 SEN.					NOT	READY	YET														
107	SDM CSAT3			4.4																		
108	SDM-UDG01			135																		
109	SDMX50																					
110	SDM GRP TRIG																					
113	SDM-SIO4																					
114	SET TIME																					
115	SDM BAUD																					
116	GENERIC SDM																					
117	DATALOG ID																					
118	SDM-OBDII																					
130	STATUS MON																					
131	SLOPE IND VW																					
132	SETTLING TIME																					
133	CAO				(-8.0) + 49.0 * R				(-10.0) + 195.0 * R				5.0 + 226.0 * R				(-30.0) + 133.0 * R			(-10.0) + 150.0 * R		
134	AM25T MX																					
220	DISPLAY OUT																					

TABLE 3.9-2. Processing Instruction Memory and Execution Times R = No. of Reps.

INSTRUCTION	INPUT LOC.	MEMORY INTER. LOC.	PROG. BYTES	EXECUTION TIME (ms)
30 Z=F	1	0	9	+ = 0.5 + 0.1 $*$ exponent - = 0.5 + 0.3 * exponent
31 Z=X	1	0	6	0.4
32 Z=Z+1	1	0	4	0.4
33 Z=X+Y	1	0	8	0.7
34 Z=X+F	1	0	10	0.6
35 Z=X-Y	1	0	8	0.7
36 Z=X$*$Y	1	0	8	0.7
37 Z=X$*$F	1	0	10	0.7
38 Z=X/Y	1	0	8	1.5
39 Z=SQRT(X)	1	0	6	4.5
40 Z=LN(X)	1	0	6	4.3
41 Z=EXP(X)	1	0	6	3.4
42 Z=1/X	1	0	6	1.5
43 Z=ABS(X)	1	0	6	0.6
44 Z=FRAC(X)	1	0	6	0.6
45 Z=INT(X)	1	0	6	0.7
46 Z=X MOD F	1	0	10	1.8
47 Z=X^Y	1	0	8	7.7
48 Z=SIN(X)	1	0	6	4.1
49 SPA. MAX	1 or 2	0	8	0.6 + 0.5 * swath
50 SPA. MIN	1 or 2	0	8	0.8 + 0.5 * swath
51 SPA. AVG	1	0	8	1.2 + 0.5 * swath
52 RUNNING AVG	1	(R $*$ par 4) + R + 1	11	0.3 + 2.7R
53 A$*$X+B	4	0	36	2.3
54 BLOCK MOVE	R	0	10	0.2 + 0.1R
55 POLYNOMIAL	R	0	31	0.3 + (0.2 $*$ order + 0.7)*R
56 SAT. VP	1	0	6	2.2
57 WDT-VP	1	0	10	3.9
58 LP FILTER	R	R + 1	13	0.5 + 1.5R
59 X/(1-X)	1	0	9	0.1 + 1.4R
60 FFT	see instruction	see instruction	13	see instruction
61 INDIR. MOVE	1	0	6	0.5
62 COV/COR	see instruction	see instruction	18	see instruction
63 PARA.EXTN.	0	0	10	0.1
64 PAROSCIENTIFIC	3	0	6	7.5
65 BULK LOAD	8	0	36	2.5
66 ARC TAN	1	0	8	4.4
67 DYNAGAGE	2 or 6	0	43	7.6
68 4 DIG PARA. EXTN.	0	0	18	0.1

TABLE 3.9-3. CR23X Output Instructions R = No. of Reps.

INSTRUCTION	INTER. LOC.	FINAL MEMORY VALUES	BYTES	OPTION	FLAG O LOW	FLAG 0 HIGH
69 WIND VECTOR	2+9R	2R	12	00,	2.7+8.3R	3.9+38.7R
		3R		01,	2.7+8.3R	4.5+15.6R
		4R		02,	2.4+8.8R	3.8+34.7R
		2R		10,	2.1+8.2R	0.0+27.3R
		3R		11,	2.1+8.2R	3.7+15.8R
		4R		12,	1.6+6.4R	3.1+27.4R
70 SAMPLE	0	R	6	0.1	0.5+0.2R	
71 AVERAGE	1+R	R	7		0.6+0.3R	1.4+1.7R
72 TOTALIZE	R	R	7		0.4+0.3R	0.9+0.6R
73 MAXIMUM	1R	R	8	0,	0.6+0.5R	1.6+1.4R
	2R	2R, 3R		01,10,11	0.6+0.5R	1.6+2.8R
74 MINIMUM	1R	R	8	0,	0.6+0.5R	1.6+1.4R
	2R	2R, 3R		01,10,11	0.6+0.5R	1.6+2.8R
75 HISTOGRAM	1+bins∗R	BINS*R	24		0.7+1.7R	0.8+(2.8+0.7*BIN)R
77 REAL TIME	0	1 TO 4	4		0.1	2.2
78 RESOLUTION	0	0	3		0.4	0.4
79 SMPL ON MM	R	R	7		0.2	0.7+0.3R
80 STORE AREA[1]	0	0	7		0.2	0.2
81 RAINFLOW HIST	see instruction		42			
82 STD. DEV.	1+3R	R	7		0.8+1.0R	3.3+7.0R

TABLE 3.9-4. Program Control Instruction Memory and Execution Times

INSTRUCTION	MEMORY INTER. LOC.	MEMORY PROG. BYTES	EXECUTION TIME (ms)
83 IF CASE <F	0	10	0.23
85 LABEL SUBR.	0	3	0.08
86 DO	0	6	0.17
87 LOOP	1	10	0.16
88 IF X<=>Y	0	11	0.42
89 IF X<=>F	0	13	0.32
90 LOOP INDEX	0	3	0.40
91 IF FLAG/PORT	0	7	0.22
92 IF TIME	1	12	0.21
93 BEGIN CASE	1	8	0.16
94 ELSE	0	4	0.16
95 END	0	4	0.22
96 SERIAL OUT	0	3	
97 INIT.TELE.	7	17	
98 SEND CHAR.	0	3	
120 GOES SAT	0 or 2	5	
121 ARGOS SAT	0	8	

3.10 ERROR CODES

There are four types of errors flagged by the CR23X: Compile, Run Time, Editor, and ⌖ D Mode. Compile errors are errors in programming which are detected once the program is entered and compiled for the first time (⌖ 0 , ⌖ 6 , or ⌖ B Mode entered). If a programming error is detected during compilation, an E is displayed with the 2 digit error code. The Instruction Location Number of the Instruction which caused the error is displayed to the right of the error code (e.g., E23 105; 105 indicates that the fifth instruction in Table 1 caused error 23). Error 22, missing END, will indicate the location of the instruction which the compiler cannot match with an END instruction.

Run time errors are detected while the program is running. The number of the instruction being executed at the time the error is detected is displayed to the right of the error code (e.g., E09 06 indicates that an Instruction 6 in the program is attempting to store data in input locations beyond those allocated). Run time errors 9 and 31 are the result of programming errors. While E08 will display the number of the instruction that was being executed when the error occurred, it is unlikely that the instruction has anything to do with the error.

If there is a run time error in a table with a fast execution interval, the error may be written to the display so frequently that it seems the CR23X is not responding to the keyboard. Once the program is stopped, normal function will return. To stop the program some entry must be changed which requires recompiling (Section 1.1.4). For example, enter 0 for the execution interval of Table 1 (i.e., enter ⌖ 1 A 0 A as fast as possible). The program can easily be stopped by pressing any key while the CR23X is displaying "Hello" after applying power (turn the CR23X off and then on again). This delays program execution for about two minutes, allowing the program to be changed.

Error 8 is the result of a hardware and software "watchdog" that checks the processor state, software timers, and program related counters. The watchdog will attempt to reset the processor and program execution if it finds that the processor has bombed or is neglecting standard system updates, or if the counters are out of allowable limits. Error code 08 is flagged when the watchdog performs this reset. E08 is

occasionally caused by voltage surges or transients. Frequent repetitions of E08 are indicative of a dead lithium battery, a hardware problem, or a software bug. Check the lithium battery voltage (⌖ B). If the lithium battery voltage is good (2.4 volts or higher), contact Campbell Scientific for assistance as a hardware or software bug is indicated. The CR23X keeps track of the number of times (up to 99) that E08 has occurred. The number can be displayed and reset in the ⌖ B Mode (Section 1.6) or with the Telecommunications A command (Section 5.1).

Error 10 is displayed if the primary power drops below 11 volts. When this happens, the CR23X stops executing programs. The low voltage counter (⌖ B Window 9, Section 1.6) counts the number of times the voltage drops below 11 volts and displays a double dash (--) if the CR23X is currently in a low voltage shut down. Low voltage shut down terminates when voltage is raised above 11 volts.

Editor errors are detected as soon as an incorrect value is entered and are displayed immediately. ⌖ B Mode errors indicate problems with saving or loading a program.

TABLE 3.10-1. Error Codes

Code	Type	Description
03	Editor	Program table full
04	Compile	Intermediate Storage full
05	Compile	Storage Area #2 not allocated
08	Run Time	CR23X reset by watchdog timer
09	Run Time	Insufficient Input Storage
10	Run Time	Low battery voltage
11	Editor	Attempt to allocate more Input or Intermediate Storage than is available
12	Compile	Duplicate ⌖ 4 ID
13	Run Time	Low 5 V supply
20	Compile	SUBROUTINE encountered before END of previous subroutine
21	Compile	END without IF, LOOP or SUBROUTINE
22	Compile	Missing END
23	Compile	Nonexistent SUBROUTINE
24	Compile	ELSE in SUBROUTINE without IF

25	Compile	ELSE without IF
26	Compile	EXIT LOOP without LOOP
27	Compile	IF CASE without BEGIN CASE
30	Compile	IF and/or LOOP nested too deep
31	Run Time	SUBROUTINES nested too deep
32	Compile	Instruction 3 and interrupt subroutine use same port
40	Editor	Instruction does not exist
41	Editor	Incorrect execution interval
60	Compile	Insufficient Input Storage
61	Compile	Burst Measurement Scan Rate too short
62	Compile	N<2 in FFT
68	Compile	Instruction 118 without enough Instructions 68 or 63
80	Compile	Valid entries for parameter 1 of P80 are 1, 2, 3 (-- is illegal)
92	Compile	Instruction 92, intervals in seconds: Time into Interval > 59 or Interval > 60
93	Compile	Save labels full
94	[*] [D] Mode	Program Storage Area full
95	[*] [D] Mode	Program does not exist in Flash memory
96	[*] [D] Mode	Addressed device not connected or wrong address (see Table 1.8-2)
97	[*] [D] Mode	Data not received within 30 seconds
98	[*] [D] Mode	Uncorrectable errors detected
99	[*] [D] Mode	Wrong file type or editor error
101	TGT1	No response
102	TGT1	Bad Star Pound entry
105	Illegal	Baud C5-C8
106	Illegal	Channel C5-C8
107	Compile	Second CSAT3 Instruction not nested

SECTION 4. EXTERNAL STORAGE PERIPHERALS

External data storage devices are used to provide a data transfer medium that the user can carry from the test site to the lab and to supplement the internal storage capacity of the CR23X, allowing longer periods between visits to the site. The standard data storage peripheral for the CR23X is the Storage Module (Section 4.4). Output to a printer or related device is also possible (Section 4.3).

Data output to a peripheral device can take place ON-LINE (automatically, as part of the CR23X's routine operation) or it can be MANUALLY INITIATED. On-line data transfer is accomplished with Instruction 96 (Section 4.1). Manual initiation is done in the ✱ 8 *Mode (Section 4.2).*

The CR23X can output data to multiple peripherals. The CR23X activates the peripheral it sends data to in one of two ways (Section 6.2):

1. *A specific pin in the CS I/O connector is dedicated to that peripheral; when that pin goes high, the peripheral is enabled. This is referred to as "PIN-ENABLED" or simply "ENABLED".*

2. *The peripheral is synchronously addressed by the CR23X. This is referred to as "ADDRESSED".*

Modems are pin-enabled. Only one modem device may be connected to the CR23X at any one time.

The SM192, SM716, and CSM1 Storage Modules are addressed. The CR23X can tell when the addressed device is present. The CR23X will not send data meant for the Storage module if the Storage Module is not present (Section 4.4.2).

The ✱ 9 *Mode (Section 4.5) allows the user to communicate directly with the Storage Module and to perform several functions, including review of data, battery test, review of Storage Module status, etc.*

Cassette tape data storage is not supported by the CR23X.

4.1 ON-LINE DATA TRANSFER - INSTRUCTION 96

All on-line data output to a peripheral device is accomplished with Instruction 96. (Instruction 96 can also be used to transfer data from one Final Storage Area to the other, Section 8.8, 12). This instruction must be included in the datalogger program for on-line data transfer to take place. Instruction 96 should follow the Output Processing Instructions, but only needs to be included once in the program table unless both Final Storage areas are in use. The suggested programming sequence is:

1. Set the Output Flag.

2. If both Final Storage Areas are in use or if you wish to set the Output Array ID, enter Instruction 80 (Section 11).

3. Enter the appropriate Output Processing Instructions.

4. Enter Instruction 96 to enable the on-line transfer of Final Storage data to the specified device. If outputting to more than one device, Instruction 96 must be entered separately for each device.

5. Repeat steps 2 through 4 if you wish to output data to the other Final Storage Area and the peripheral.

SECTION 4. EXTERNAL STORAGE PERIPHERALS

Instruction 96 has a single parameter which specifies the peripheral to send output to. Table 4.1-1 lists the output device codes.

TABLE 4.1-1. Output Device and Baud Rate Codes

Code	Baud Rate
0	300
1	1200
2	9600
3	76800
4	2400
5	4800
6	19200
7	38400

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Option Device

ADDRESSED PRINT DEVICE, y = Baud code
1y   = Printable ASCII
2y   = Comma Separated ASCII
3y   = Binary Final Storage Format
7N   = Storage Module N (N=1-8; Section 4.4.2)
        (Stored in Binary Format)
7N-- = Output File Mark to Storage Module N

SERIAL PRINTER, COMPUTER, OR PIN-ENABLED PRINT DEVICE, y = Baud code (SDE pulled high)
4y   = Printable ASCII (CS I/O)
4y-- = Printable ASCII (RS-232)
5y   = Comma Separated ASCII (CS I/O)
5y-- = Comma Separated ASCII (RS-232)
6y   = Binary Final Storage Format (CS I/O)
6y-- = Binary Final Storage Format (RS-232)

TRANSFER DATA TO OTHER FINAL STORAGE AREA
80 = New data only
81 = All data

The source of data for Instruction 96 is the currently active Final Storage Area as set by Instruction 80 (the default is Final Storage Area 1 at the beginning of each program table execution).

If the CR23X is using a port (CS I/O or RS-232) for other I/O tasks when Instruction 96 is executed, the output request is put in a queue and program execution continues. As the port becomes available, each device in the queue gets its turn.

An output request is not put in the queue if the same device is already in the queue. The data contained in the queue (and which determine a unique entry) are the device, baud rate (if applicable), and the Final Storage Area.

When an entry reaches the top of the queue, the CR23X sends all data accumulated since the last transfer to the device up to the location of the DSP at the time the device became active.

Printer output can be either pin-enabled or addressed. However, there is not a pin specifically dedicated to print enable. When a pin-enabled print output is specified, the SDE line, which is normally used in the addressing sequence, is used as a print enable. This allows some compatibility with the CR21, 21X, and CR7 dataloggers which have a Print Enable line. The pin-enabled print option will result in garbage being sent to the print peripheral if an addressed device is also connected to the CR23X (i.e., SM192 or SM716 etc.). The SDC99 Synchronous Device Interface can convert a print device to an Addressed peripheral (Section 6.2).

The STORAGE MODULE address is important only when using more than one Storage Module. The universal address that will find the Storage Module with lowest number address is "1". If a Storage Module is not connected, the CR23X will not advance the SPTR (Section 2.1) and the Storage Module drops out of the queue until the next time Instruction 96 is executed. Section 4.4 contains specifics on the Storage Modules.

TABLE 4.2-1.   ⟨8⟩ Mode Entries

Key	Display ID: DATA	Description
⟨*⟩⟨8⟩ Mode	08: Storage Area 00	Key 1 or 2 for Storage Area. (This window is skipped if no memory has been allocated to Final Storage Area 2.)
⟨A⟩	01: Device Code XX	Key in Output Device Option. See Table 4.1-1.
⟨A⟩	02: Start Location XXXXX	Start of dump location. Initially the SPTR or PPTR location; a different location may be entered if desired.
⟨A⟩	03: End Location XXXXX	End of dump location. Initially the DSP location; a different location may be keyed in if desired.
⟨A⟩	04: Number Starts 00	Ready to dump. To initiate dump, key any number, then A. While dumping, "04 activated; key aborts" and the location number will be displayed. "Output complete" will be displayed when the dump is complete. (Any key aborts transmission after completion of the current data block.)

4.2 MANUALLY INITIATED DATA OUTPUT - ⟨*⟩ ⟨8⟩ MODE

Data transfer to a peripheral device can be manually initiated in the ⟨*⟩ ⟨8⟩ Mode. The ⟨*⟩ ⟨8⟩ Mode allows the user to retrieve a specific block of data, on demand, regardless of whether or not the CR23X is programmed for on-line data output.

If external storage peripherals are not left on-line, the maximum time between collecting data must be calculated to ensure that data in Final Storage are not lost due to write-over. To calculate this time it is necessary to know: (1) the size of Final Storage, (2) the number of Output Arrays being generated, (3) the number of low and/or high resolution data points per Output Array, and (4) the rate at which Output Arrays are placed into Final Storage. When calculating the number of data points per Output Array, remember to add 1 data point per array for the Output Array ID.

For example, assume that 586,568 locations are assigned to Final Storage (⟨*⟩ ⟨A⟩ Mode), and that 1 Output Array, containing the Array ID (1 memory location), 9 low resolution data points (9 memory locations) and 5 high resolution data points (10 memory locations), is stored each hour. In addition, an Output Array with the Array ID and 5 high resolution data points (11 memory locations) is stored daily. This is a total of 491 memory locations per day ((20 x 24) + 11). 586,568 divided by 491 = 1194 days. Therefore, the CR23X would have to be visited every 1194 days to retrieve data, because write-over would

begin on the 1195th day. The site should be visited more frequently than this for routine maintenance. Thus data storage capacity would not be a factor in determining how frequently to visit the site.

The output device codes used with the ⟨*⟩ ⟨8⟩ Mode are the same as those used with Instruction 96 (Table 4.1-1), with the exception of the option to transfer data from one Final Storage area to the other (80, 81). Table 4.2-1 lists the keystrokes required to initiate a ⟨*⟩ ⟨8⟩ data dump.

4.3 PRINTER OUTPUT FORMATS

Printer output can be sent in binary Final Storage Format (Appendix C.2), Printable ASCII, or Comma Separated ASCII. These ASCII formats may also be used when data from the Storage Modules or Telecommunications are stored on disk with Campbell Scientific's PC208W software.

4.3.1 PRINTABLE ASCII FORMAT

In the Printable ASCII format, each data point is preceded by a 2 digit data point ID and a (+) or (−) sign. The ID and fixed spacing of the data points make particular points easy to find on a printed output. This format requires 10 bytes per data point to store on disk.

Figure 4.3-1 shows both high and low resolution data points in a 12 data point Output Array. The example data contains Day, Hour-Minute, and Seconds in the 2nd - 4th data points. REMEMBER! You must specifically program

the CR23X to output the date and time values. The Output Array ID, Day, and Time are always 4 character numbers, even when high resolution output is specified. The seconds resolution is 0.1 seconds.

Each full line of data contains 8 data points (79 characters including spaces), plus a carriage return (CR) and line feed (LF). If the last data point in a full line is high resolution, it is followed immediately with a CR and LF. If it is low resolution, the line is terminated with a space, CR and LF. Lines of data containing less than 8 data points are terminated similarly after the last data point.

4.3.2 COMMA SEPARATED ASCII

Comma Separated ASCII strips all IDs, leading zeros, unnecessary decimal points and trailing zeros, and plus signs. Data points are separated by commas. Arrays are separated by Carriage Return Line Feed. Comma Separated ASCII requires approximately 6 bytes per data point. Example:

1,234,1145,23.65,-12.26,625.9
1,234,1200,24.1,-10.98,650.3

4.4 STORAGE MODULE

The Storage Module stores data in battery backed RAM. Backup is provided by an internal lithium battery. The RAM is internal on the SM192/716 and on a PCMCIA card in the CSM1. Operating power is supplied by the CR23X over pin 1 of the CS I/O port. Whenever power is applied to the CS I/O port (after having been off), the Storage Module places a File Mark in the data (if a File Mark is not the last data point already in storage).

The File Mark separates data. For example, if you retrieve data from one CR23X, disconnect the Storage Module and connect it to a second CR23X, a File Mark is automatically placed in the data. This mark follows the data from the first CR23X but precedes the data from the second.

The SM192 has 192K bytes of RAM storage; the SM716 has 716K bytes. Both can be configured as either ring or fill and stop memory. The size of memory in the CSM1 depends on the PC Card used. The CSM1 is always fill and stop.
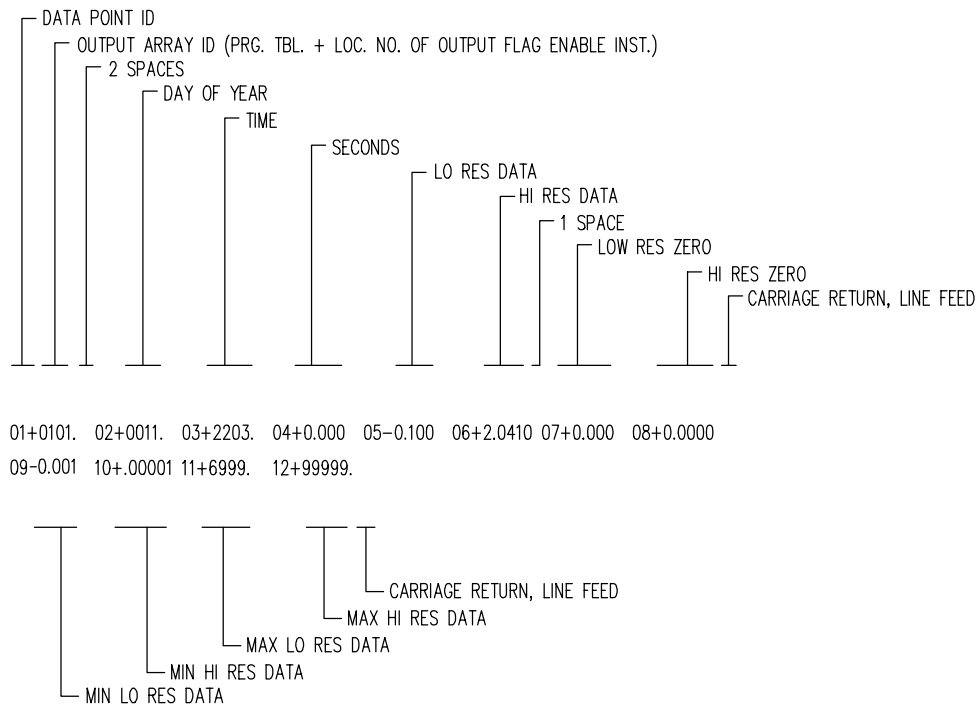
FIGURE 4.3-1. Example of CR23X Printable ASCII Output Format

4.4.1 STORAGE MODULE ADDRESSING

The CSM1 does not support individual addresses. Use address "1" when sending data to the CSM1.

The SM192/716 Storage Modules can have individual addresses. Different addresses allow 1) up to 8 Storage Modules to be connected to the CR23X during on-line output, 2) different data to be output to different Modules, and 3) transfer of data from a Module that is left with the CR23X to a Module that is hand carried to the site for data transfer (✳ 9 Mode).

Storage Modules are assigned addresses (1-8) either through the ✳ 9 Mode or with the PC208W software. The default address when the Storage Module is reset is "1". Unless you are using one of the features which require different addresses, you need not assign any other address.

Address 1 is also a universal address when sending data or commands to a storage module with Instruction 96, ✳ 8 , or ✳ 9 . When address 1 is entered in the ✳ 9 Mode (default) or in the device code (71, Table 4.2-1) for Instruction 96 or the ✳ 8 Mode, The CR23X searches for the Storage Module with the lowest address that is not full (fill and stop configuration only) and addresses it. In other words, if a single Storage Module is connected, and it is not full, address 1 will address that Storage Module regardless of the address that is assigned to the Module.

Address 1 would be used with Instruction 96 if several Storage Modules with different addresses were connected to the CR23X and were to be filled sequentially. The Storage modules would be configured as fill and stop. When the lowest addressed Module was full data would be written to the next lowest addressed Module, etc.

4.4.2 STORAGE MODULE USE WITH INSTRUCTION 96

When output to the Storage Module is enabled with Instruction 96, the Storage Module(s) may be either left with the CR23X for on-line data transfer and periodically exchanged, or brought to the site for data transfer.

USE OF STORAGE MODULE TO PICK UP DATA

The CR23X is capable of recognizing whether or not the Storage Module is connected. Each time Instruction 96 is executed and there is data to output, the CR23X checks for the presence of a Storage Module. If one is not present, the CR23X does not attempt to output data. Instead, the CR23X saves the data and continues its other operations without advancing the Storage Module Pointer (SPTR, Section 2.1).

When the user finally does connect the Storage Module to the CR23X, two things happen:

1. Immediately upon connection, a File Mark is placed in the Storage Module Memory following the last data stored (if a File Mark wasn't the last data point already in storage).

2. During the next execution of Instruction 96, the CR23X recognizes that the Storage Module (SM) is present and outputs all data between the SPTR and the DSP location.

The File Mark allows the operator to distinguish blocks of data from different dataloggers or from different visits to the field.

To be certain that the Storage Module has been connected to the CR23X during an execution of P96, the user can:

• Leave the Storage Module connected for a time period longer than an execution interval *or*

• Use the SC90 9-Pin Serial Line Monitor. The SC90 contains an LED which lights up during data transmission. The user connects the SM to the CR23X with the SC90 on the line and waits for the LED to light. When the light goes off, data transfer is complete and the SM can be disconnected from the CR23X.

4.4.3 ✳ 8 DUMP TO STORAGE MODULE

In addition to the on-line data output procedures described above, output to the Storage Module can be manually initiated in the ✳ 8 Mode. The procedure for setting up and transferring data is as follows:

1. Connect the Storage Module to the CR23X using the SC12 cable.

2. Key in the appropriate commands as listed in Table 4.2-1.

4.5 [*] [9] MODE -- SM192/716 STORAGE MODULE COMMANDS

The CSM1 does not support the [*] [9] Mode Commands.

The [*] [9] Mode is used to issue commands to the SM192/716 Storage Module, from the CR23X. These commands are like [*] Modes for the Storage Module and in some cases are directly analogous to the CR23X [*] Modes. Command 7 enters a mode used to review stored data, and 8 is used to transfer data between two Storage Modules connected to the CR23X. The operations with the Storage Module are not directly analogous as may be seen in Table 4.5-1 which lists the

commands (e.g., when reviewing data, #A advances to the start of the next Output Array rather than to the same element in the next array with the same ID).

When [*] [9] is keyed, the CR23X responds: 09:01

1 is the default address for the Storage Module (Section 4.4.1). If you have more than 1 Storage Module connected, enter the address of the desired Storage Module. Address 1 will always work if only one Module is connected. Key A and the CR23X responds: 9N:00 Where N is the address which was entered.

You may now enter any of the commands in Table 4.5-1 (key in the command number and enter with A). Most commands have at least one response. Advance through the responses and return to the [*] [9] command state by keying A.

TABLE 4.5-1. [*] [9] Commands for Storage Module

COMMAND	DISPLAY	DESCRIPTION
1	01: 0000	RESET, enter 248 to erase all data and programs. While erasing, the SM checks memory. The number of good chips is then
	01: XX	displayed (6 for SM192, 22 SM716).
3	03: 01	INSERT FILE MARK, 1 indicates that the mark was inserted, 0 that it was not.
4	04: XX	DISPLAY/SET MEMORY CONFIGURATION enter the appropriate code to change configuration 0=ring, 1=fill & stop
5		DISPLAY STATUS (A to advance to each window)
	01: ABCD	Window 1:
	AB	Storage pointer location (chip no.)
	CD	Total good RAM chips (1-22)
	02: ABCD	Window 2:
	AB	Display pointer location (chip no.)
	C	Unloaded Batt. Chk. 0=low, 1=OK
	D	No. of Programs stored (Max=8)
	03: A0CD	Window 3:
	A	Errors logged (up to 9)
	0	Not Used
	C	Memory Config. (0=ring, 1=fill&stop)
	D	Memory Status (0=not full, 1=full)
	04: XXXXX	PROM signature (0 if bad PROM)
6	06: 0X	BATTERY CHECK UNDER LOAD (0=low, 1=OK)
7	07: 00	DISPLAY DATA, Select the Storage Module Area with these codes:
	0	Dump pointer to SRP
	1	File 1, current file
	2	File 2, previous to file 1
	3	File 3, previous to file 2
	4	File 4, previous to file 3
	5	File 5, previous to file 4
	7	Display pointer to SRP
	9	Oldest data to SRP
	1-5	will loop within file boundaries, 0,7,9 allow display to cross boundaries

	07:XXXXXX	SM location at end of area selected. Key A to advance to first data. If another location is keyed in SM will jump to 1st start of array following that location.
		Review data with:
		A Advance and display next data point
		B Back-up one data point
		# Display location, C to return to data
		#A Advance to next start of Array
		#B Back-up to start of Array
		#D Return to ⁹ command mode
8		DUMP TO ANOTHER STORAGE MODULE
	08:00	Select Area as in 7 above
	01:XXXXXX	First Loc. in area selected/Enter Loc. to start dump
	02:XXXXXX	Final Loc. in area selected/Enter Loc. to end dump
	03:XX	Enter destination SM address
9		DISPLAY ADDRESSES OF CONNECTED SM
	XXXXXXXX	1 = occupied, 0 = unoccupied
	87654321	(Addresses 8-1 from left to right)
10		CHANGE ADDRESS
	10:0X	X is current address, enter address to change to (1-8)

SECTION 5. TELECOMMUNICATIONS

Campbell Scientific has developed a software package which automates data retrieval and facilitates the programming of Campbell Scientific dataloggers and the handling of data files. This package, PC208W, has been designed to meet most needs in datalogger support and telecommunications. Therefore, information in this section is not necessary for most datalogger applications.

Telecommunications is used to retrieve data from Final Storage directly to a computer/terminal and to program the CR23X. Any user communication with the CR23X that makes use of a computer or terminal is done through Telecommunications.

Telecommunications can take place over a variety of links including:

- *SC32A and ribbon cable/SC929 cable*
- *Telephone*
- *Cellular phone*
- *Radio frequency*
- *Short haul modem and twisted pair wire*
- *Multi-drop interface and coax cable*

This section does not cover the technical interface details for any of these links. Those details are covered in Section 6 and in the individual manuals for the devices.

Data retrieval can take place in either BINARY or ASCII. The BINARY format is 5 times more compact than ASCII. The shorter transmission times for binary result in lower long distance costs if the link is telephone and lower power consumption with an RF link. On "noisy" links shorter blocks of data are more likely to get through without interruption.

For more efficient data transfer, binary data retrieval makes use of a signature for error detection. The signature algorithm assures a 99.998% probability that if either the data or its sequence changes, the signature changes. Campbell Scientific's PC208W Datalogger Support Software uses the binary format for data transfer.

This section does not furnish sufficient detail to write telecommunications software. Appendix C contains some details of binary data transfer and Campbell Scientific's binary data format. The emphasis of this section is on the commands that a person would use when manually (i.e., keyed in by hand) interrogating or programming the CR23X via a computer/terminal. These commands and the responses to them are sent in the American Standard Code for Information Interchange (ASCII).

The telecommunications commands allow the user to perform several operations including:

- *trouble shoot a problematic communications link*
- *check the datalogger's status*
- *monitor data in Input Storage and review data in Final Storage*
- *retrieve Final Storage data in either ASCII or BINARY*
- *open communications with the Storage Module*
- *remote keyboard programming*

The Remote Keyboard State (Section 5.2) allows the user with a computer/terminal to use the same commands as on the CR23X keypad.

5.1 TELECOMMUNICATIONS COMMANDS

When a modem/terminal rings the CR23X, the CR23X should answer almost immediately. Several carriage returns (CR) must be sent to the CR23X to allow it to set its baud rate to that of the modem/terminal (300, 1200, 2400, 4800, 9600, 19.2K, 38.4K, or 76.8K). Once the baud rate is set, the CR23X will send back the prompt, "∗", signaling that it is ready to receive a command.

SECTION 5. TELECOMMUNICATIONS

GENERAL RULES governing the telecommunications commands are as follows:

1. ∗ from datalogger means "ready for command".

2. All commands are of the form: [no.]letter, where the number may or may not be optional.

3. Valid characters are the numbers **0-9**, the capital letters **A-U**, the colon (**:**), and the carriage return (**CR**).

4. An illegal character increments a counter and zeros the command buffer, returning a **∗**.

5. **CR** to datalogger means "execute".

6. A carriage return followed immediately by a line feed character (**CRLF)** from datalogger means "executing command".

7. ANY character besides a **CR** sent to the datalogger with a legal command in its buffer causes the datalogger to abort the command sequence with **CRLF∗** and to zero the command buffer.

8. All commands return a response code, usually at least a checksum.

9. The checksum includes all characters sent by the datalogger since the last ∗, including the echoed command sequence, excluding only the checksum itself. The checksum is formed by summing the ASCII values, without parity, of the transmitted characters. The largest possible checksum value is 8191. Each time 8191 is exceeded, the CR23X starts the count over; e.g., if the sum of the ASCII values is 8192, the checksum is 0.

10. Commands that return Campbell Scientific binary format data (i.e., **F** and **K** commands) return a signature (see Appendix C.3).

The CR23X sends ASCII data with 8 bits, no parity, one start bit, and one stop bit.

After the CR23X answers a ring, or completes a command, it waits about 40 seconds (127 seconds in the Remote Keyboard State) for a valid character to arrive. It "hangs up" if it does not receive a valid character in this time interval. Some modems are quite noisy when not on line; it is possible for valid characters to appear in the noise pattern. To insure that this situation does not keep the CR23X in telecommunications, the CR23X counts all the invalid characters it receives from the time it answers a ring, and terminates communication after receiving 150 invalid characters.

The CR23X continues to execute its measurement and processing tasks while servicing the telecommunication requests. If the processing overhead is large (short Execution Interval), the processing tasks will slow the telecommunication functions. In a worst case situation, the CR23X interrupts the processing tasks to transmit a data point every 0.1 second.

The best way to become familiar with the Telecommunication Commands is to try them from a terminal connected to the CR23X via the SC32A (Section 6.7.1) or other interface. Commands used to interrogate the CR23X in the Telecommunications Mode are described in the following Table.

TABLE 5.1-1. Telecommunications Commands

Command	Description
[F.S. Area]**A**	SELECT AREA/STATUS - If 1 or 2 does not precede the A to select the Final Storage Area, the CR23X will default to the Area last used (initially this is Area 1). All subsequent commands other than A will address the area selected. Datalogger returns **R**eference, the DSP location; the number of **F**illed Final Storage locations; **V**ersion of datalogger; Final Storage **A**rea; **L**ocation of MPTR (the location number may be 1 to 7 characters long); **E**rrors #1, #2, #3, and #4 where #1 is the number of E08's, #2 is the number of overrun errors, #3 is the number of times the program stopped due to low voltage, and #4 is the number of times the 5 V supply dropped below 5V (all are cleared by entering 8888A; #2 is also cleared at time of program compilation); size of total **M**emory in CR23X in Kbytes; the lithium **B**attery voltage; and **C**hecksum. All in the following format:
	R+xxxxx. F+xxxxx. Vxx Axx L+xxxxxxx. Exx xx xx xx Mxxxx B+x.xxxx Cxxxx
	If data is stored while in telecommunications, the A command must be issued to update the Reference to the new DSP.
[no. of arrays]**B**	BACK-UP - MPTR is backed-up the specified number of Output Arrays (no number defaults to 1) and advanced to the nearest start of array. CR23X sends the Area, MPTR Location, and Checksum:
	Ax L+xxxxxxx Cxxxx
[YR:DAY:HR:MM:SS]**C**	RESET/SEND TIME - If time is entered the time is reset. If only 2 colons are in the time string, HR:MM:SS is assumed; 3 colons means DAY:HR:MM:SS. If only the C is entered, time is unaltered. CR23X returns year, Julian day, hr:min:sec, and Checksum:
	Yxx Dxxxx Txx:xx:xx Cxxxx
[no. of arrays]**D**	ASCII DUMP - If necessary, the MPTR is advanced to the beginning of the next array. CR23X sends the number of arrays specified (no number defaults to 1) or the number of arrays between MPTR and Reference, whichever is smaller, CRLF, FSA, Location, Checksum.
E	End call. Datalogger sends CRLF only.
[no. of loc.]**F**	BINARY DUMP - Used by CSI software for data retrieval. See Appendix C.
[F.S. loc. no.]**G**	MOVE MPTR - MPTR is moved to specified Final Storage location. The location number must be entered. CR23X sends Area, Location, and Checksum:
	Ax L+xxxxxxx Cxxxx
7H or **2718H**	REMOTE KEYBOARD - CR23X sends the prompt ">" and is ready to execute standard keyboard commands (Section OV3). Aborted by pressing any key (except *, D , #) on local keypad.

[loc. no.]**I**   Display/change value at Input Storage location. CR23X sends the value stored at the location. A new value and CR may then be sent. CR23X sends checksum. If no new value is sent (CR only), the location value will remain the same.

3142J   TOGGLE FLAGS AND SET UP FOR K COMMAND - Used in the Monitor Mode and with the Heads Up Display. See Appendix C for details.

2413J   SET UP FOR K COMMAND - Used in the Monitor Mode and with the Heads Up Display. Similar to the 3142J command but does not toggle flags and ports. See Appendix C for details. (Available first in CR23X OS Version 1.7; also indicated by "V4" returned to the A command, see above.)

K   CURRENT INFORMATION - In response to the K command, the CR23X sends datalogger time, user flag status, the data at the input locations requested in the J command, and Final Storage Data if requested by the J command. Used in the Monitor Mode and with Heads Up Display. See Appendix C for details.

[Password]**L**   Unlocks security (if enabled) to the level determined by the password entered (See * C Mode, Section 1.7). CR23X sends security level (0-3) and checksum:

Sxx Cxxxx

[X]**M**   Connect to Storage Module with address 'X' and enter the Storage Module's Telecommunications Mode (see Storage Module manual). The Storage Module can also be accessed through the * 9 Commands while in the Remote Keyboard Mode (Section 4.5 and the Storage Module manual).

1N   Connect phone modem to RF modem at phone to RF base station.

P   Command to set 9.8304 MHz crystal coefficients. Coefficients are set at the factory and normally should not be altered. Displays:

E clock Hz - 2.4576 MHz

to give better resolution. Also, gets rid of out of bounds check that used to load default back in. If not set, P27 and other timing instructions will have problems.

19287P will display the 2 coefficient numbers for you (e.g. 0.8944 .8133). If the 2 numbers are -99999, the datalogger needs to be calibrated.

19287:1600P will calibrate it for you if your crystal is exactly 2.4576 MHz E Clock. This temporary fix will improve the performance of a datalogger that failed the 19287P test.

R   Command to set the display's contrast dependence on temperature. Typing **R** alone will display the current settings to you (## ## ## ##

##). Typing 8 numbers, separated by colons, followed by an R, will reset the default settings. Example:

140:110:90:65:50:45:34:30R <crlf>

The setting of the eight contrast temperature bins is initially done at Campbell Scientific. Below are the contrast settings of one type of LCD screen for temperatures from <-15 to >+50 °C. A user can also adjust the value of the current bin by entering the * D mode while in Remote Keyboard Mode. The minimum contrast setting is 0. The maximum setting is 255.

Contrast Setting	Temperature ºC
140	< -15
110	-10 to -15
90	-5 to -10
65	-5 to +5
50	+5 to +30
45	+30 to +40
35	+40 to +50
30	> +50

S   Returns Mode A Memory Allocation registers (first group of 01: to 09:) and Mode B Status/On-board Firmware registers (second group of 01: to 17:)

T   SDM-SIO4 talk through command

Address: Port T
Address = 0..15
Port = 0..4

nnnnU   Returns V[value] C[checksum] where nnnn refers to an input location, port, or flag, V is the value at the input location, port or flag, and C is the checksum. For nnnn = 90ff, then nnnn refers to flag ff. For nnnn = 91pp, then nnnn refers to port pp. For nnnn<9000, then nnnn refers to input location nnnn.

Examples (xxxx is checksum, true or high for flags is non-zero, false or low is zero):
2U<CR> returns
V+73.650 Cxxxx. (input location 2 equals 73.650)

9003U<CR> returns
V+1.0000 Cxxxx (flag 3 is high).

9107U<CR> returns
V+0.0000 Cxxxx (port 7 is low)

nnnn:[value]**:**[checksum]**U**   Loads the input location, port, or flag referred to by nnnn with value if the checksum is correct. nnnn refers to input location, flag, or port as above. The datalogger returns the same as the nnnnU command.

5-5

Examples:
14:-3.2450:xxxxU<CR> returns
V-3.2450 C1357 (sets input location 14 to -3.2450)

9003:1:xxxxU<CR> returns
V1.0000 Cxxxx (sets flag 3 high)

9105:0:xxxxU<CR> returns
V0.0000 Cxxxx (sets port 5 low)

5.2 REMOTE PROGRAMMING OF THE CR23X

Remote programming of the CR23X can be accomplished with the PC208W software or directly through the Remote Keyboard State.

The PC208W Datalogger Support Software was developed by Campbell Scientific for use with IBM or compatible PC's. Datalogger programs are developed on the computer using the program editor (Edlog in PC208W or the separate Short Cut program generator) and downloaded to the datalogger using PC208W's Connect screen.

Alternatively, a terminal session can be opened using any terminal program (e.g., Microsoft's Hyperterminal) and connecting via serial cable to the CR23X. PC208W's Connect screen also offers such a Terminal window. The CR23X can be placed in the Remote Keyboard State by sending either "**7H**" or "**2718H**" and a carriage return (CR). The CR23X responds by sending a CR, line feed (LF), and the prompt '>'. The CR23X is then ready to receive the standard keyboard commands; it recognizes all the standard CR23X keyboard characters plus several additional characters, including the decimal point, the minus sign, and Enter (CR) (Section OV3.2).

> **NOTE:** Entering * 0 returns the CR23X to the telecommunications command state.

Remember that entering * 0 will compile and run the CR23X program if program changes have been made.

The 7H Command is generally used with a terminal for direct entry since H makes use of a destructive backspace and does not send control Q between each entry. The 2718H Command functions the same as it does for other Campbell Scientific dataloggers (deleting an entry causes the entire entry to be sent, "control Q" is sent after each user entry).

It is important to remember that the Remote Keyboard State is still within Telecommunications. Entering * 0 exits the Remote Keyboard State and returns the datalogger to the Telecommunications Command State, awaiting another command. The user can step back and forth between the Telecommunications Command State and the Remote Keyboard State.

7H (or 2718H)

Telecommunications Command State → Remote Keyboard State

* 0 ←
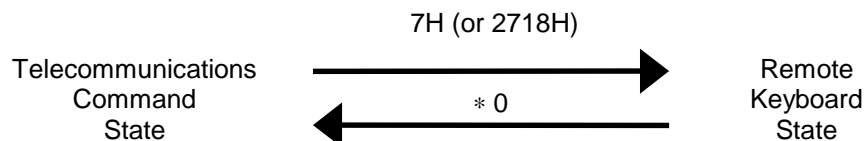
SECTION 6. 9-PIN SERIAL INPUT/OUTPUT

External communication peripherals normally connect to the CR23X through two 9-pin subminiature D-type socket connectors located on the front panel (Figure 6.1-1). An optically isolated RS-232 port is provided for direct connection to RS-232 devices such as a PC. Optical isolation provides immunity from ground loop problems that can degrade single-ended measurement accuracy in systems with multiple ground connections. The CS I/O interface utilizes (0 - 5) V signal levels and is to be used to connect to Campbell Scientific communications peripherals. Either 9-pin interface can be used for telecommunications, however, only one of these two interfaces can be active at once. The first 9-pin interface to receive a RING becomes the active interface until a telecommunications sequence is terminated. When either port is active, "Busy with COM" will show on the CR23X display. If "Busy with COM" is displayed after disconnecting the telecommunications device, pressing any key will activate the key pad after a 2 to 3 second delay.

NOTE: *Serial communications is not reliable over cable greater than 50 feet in length.*

6.1 COMPUTER RS-232 9-PIN DESCRIPTION

Direct connection of the CR23X to a PC is most conveniently done through the "Computer RS232" port (Figure 6.2-1). Table 6.1-1 gives a brief description of each "Computer RS232" pin.

The Computer RS-232 port is a DCE device when connected to a PC with a serial cable. It also doubles as a DTE device when connected to a modem device through a null-modem cable.

Maximum input = $\pm$ 25V
Minimum Output = $\pm$ 5V
Typical Output = $\pm$ 7V

TABLE 6.1-1. Computer RS-232 Pin-Out

ABR = Abbreviation for the function name
PIN = Pin number
O   = Signal Out of the CR23X to a RS-232 device
I   = Signal Into the CR23X from a RS-232 device

PIN	ABR	I/O	Description
1	DTR	O	data terminal ready
2	TX	O	asynchronous transmit
3	RX	I	asynchronous receive
4			not used
5	GND		isolated ground
6			not used
7	CTS	I	clear to send
8	RTS	O	request to send
9	RING	I	ring

The CR23X is supplied with a six foot 9 pin to 9 pin serial cable and a 9 to 25 pin adapter to facilitate connection to a PC RS-232 port.
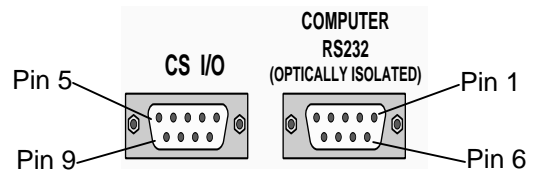
FIGURE 6.1-1. Serial Communication Interfaces

6.2 CS I/O 9-PIN DESCRIPTION

All Campbell Scientific communication peripherals connect to the CR23X through the 9-pin subminiature D-type socket connector located on the front of the Wiring Panel labeled "CS I/O" (Figure 6.1-1). Table 6.2-1 gives a brief description of each pin.

TABLE 6.2-1. Pin Description

ABR = Abbreviation for the function name.
PIN = Pin number.
O = Signal Out of the CR10X to a peripheral.
I = Signal Into the CR10X from a peripheral.

PIN	ABR	I/O	Description
1	5 V	O	5V: Sources 5 VDC, used to power peripherals.
2	SG		Signal Ground: Provides a power return for pin 1 (5V), and is used as a reference for voltage levels.
3	RING	I	Ring: Raised by a peripheral to put the CR10X in the telecommunications mode.
4	RXD	I	Receive Data: Serial data transmitted by a peripheral are received on pin 4.
5	ME	O	Modem Enable: Raised when the CR10X determines that a modem raised the ring line.
6	SDE	O	Synchronous Device Enable: Used to address Synchronous Devices (SDs), and can be used as an enable line for printers.
7	CLK/HS	I/O	Clock/Handshake: Used with the SDE and TXD lines to address and transfer data to SDs. When not used as a clock, pin 7 can be used as a handshake line (during printer output, high enables, low disables).
8	+12 VDC		
9	TXD	O	Transmit Data: Serial data are transmitted from the CR10X to peripherals on pin 9; logic low marking (0V) logic high spacing (5V) standard asynchronous ASCII, 8 data bits, no parity, 1 start bit, 1 stop bit, 300, 1200, 9600, 76,800 baud (user selectable).
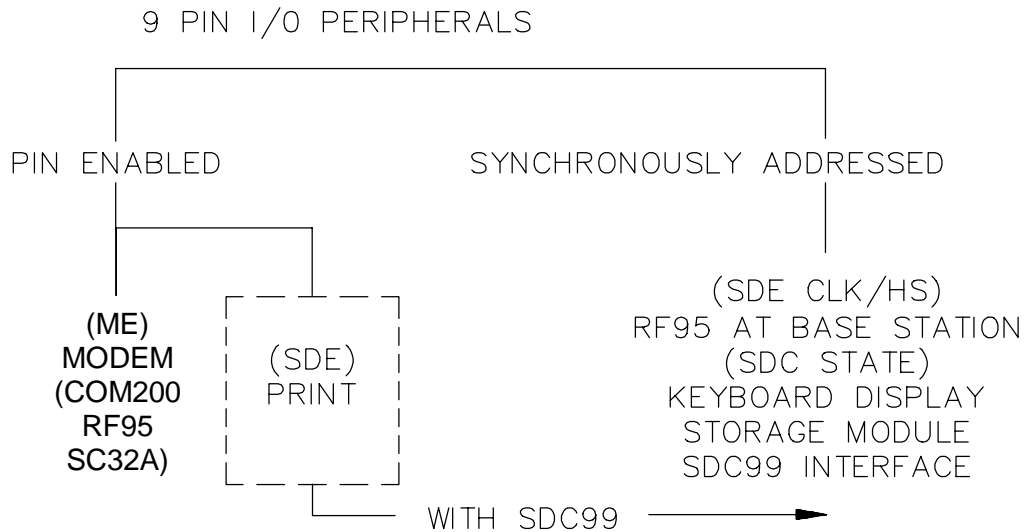
FIGURE 6.2-1. Hardware Enabled and Synchronously Addressed Peripherals

6.2.1 ENABLING AND ADDRESSING PERIPHERALS

While several peripherals may be connected in parallel to the CS I/O port, the CR23X has only one transmit line (pin 9) and one receive line (pin 4, Table 6.2-1). The CR23X selects a peripheral in one of two ways: 1) A specific pin is dedicated to that peripheral and the peripheral is enabled when the pin goes high; we will call this pin-enabled or simply enabled. 2) The peripheral is addressed; the address is sent on pin 9, each bit being synchronously clocked using pin 7. Pin 6 is set high while addressing.

6.2.1.1 PIN-ENABLED PERIPHERALS

Modem Enable (pin 5) is dedicated to a specific device. Synchronous Device Enable (pin 6) can either be used as a Print Enable or it can be used to address Synchronous Devices (Section 6.2.5).

Modem Enable (ME), pin 5, is raised to enable a modem that has raised the ring line. Modem/terminal peripherals include Campbell Scientific phone modems and computers or terminals using the SC32A or SC929 interface. The CR23X interprets a ring interrupt (Section 6.2.2) to come from a modem if the device raises the CR23X's Ring line, and holds it high until the CR23X raises the ME line. Only one modem/ terminal may be connected to the CR23X.

Print Peripherals are defined as peripherals which have an asynchronous serial communications port used to RECEIVE data transferred by the CR23X. In most cases the print peripheral is a printer, but could also be an on-line computer or other device.

Synchronous Device Enable (SDE), pin 6, may be used to enable a print peripheral only when no other addressable peripherals are connected to the CS I/O connector. Use of the SDE line as an enable line maintains CR23X compatibility with printer-type peripherals which require a line to be held high (Data Terminal Ready) in order to receive data.

If output to both a print peripheral and an addressable peripheral is necessary the SDC99 Synchronous Device Interface is required. With the SDC99 the print peripheral functions as an addressable peripheral. If the SDC99 is not used, the print peripheral receives the address and data sent to the addressed peripheral. Synchronous addressing appears as garbage characters on a print peripheral.

6.2.1.2 ADDRESSED PERIPHERALS

The CR23X has the ability to address Synchronous Devices (SDs). SDs differ from enabled peripherals in that they are not enabled solely by a hardware line (Section 6.2.1.1); an SD is enabled by an address synchronously clocked from the CR23X (Section 6.2.5).

Up to 16 SDs may be addressed by the CR23X. Unlike an enabled peripheral, the CR23X establishes communication with an addressed peripheral before data are transferred. During data transfer an addressed peripheral uses pin 7 as a handshake line with the CR23X.

Synchronously addressed peripherals include Storage Modules, SDC99 Synchronous Device Interface (SDC99), and RF95 RF Modem when configured as a synchronous device. The SDC99 interface is used to address peripherals which are normally modem enabled (Figure 6.2-1).

6.2.2 RING INTERRUPTS

There are two peripherals that can raise the CR23X's ring line; modems and the RF Modem configured for synchronous device for communication (RF-SDC). The RF-SDC is used when the CR23X is installed at a telephone to RF base station.

When the Ring line is raised, the processor is interrupted, and the CR23X determines which peripheral raised the Ring line through a process of elimination (Figure 6.2-3). The CR23X raises the CLK/HS line forcing all SDs to drop the ring line. If the ring line is still high the peripheral is a modem, and the ME line is raised. If the ring line is low the CR23X addresses the Keyboard Display and RF-SDC to determine which device to service. (Section 6.2.5)

After the CR23X has determined which peripheral raised the Ring line, the hierarchy is as follows:

A modem which raises the Ring line will interrupt and gain control of the CR23X. A ring from a modem aborts data transfer to pin-enabled and addressed peripherals.
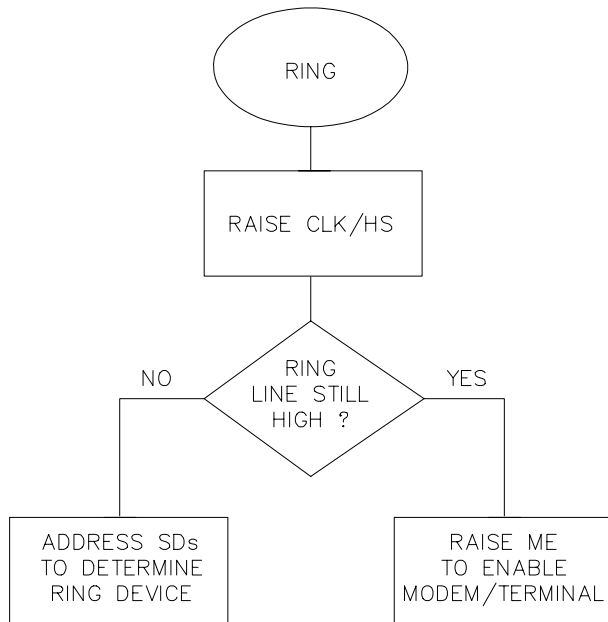
FIGURE 6.2-3. Servicing of Ring Interrupts

6.2.4 MODEM/TERMINAL PERIPHERALS

The CR23X considers any device with an asynchronous serial communications port which raises the Ring line (and holds it high until the ME line is raised) to be a modem peripheral. Modem/terminals include Campbell Scientific phone modems, and most computers, terminals, and modems using the SC32A Optically Isolated RS-232 Interface, the SC932 RS-232 DCE Interface, or the SC929 cable.

When a modem raises the Ring line, the CR23X responds by raising the ME line. The CR23X must then receive carriage returns spaced at least 50 ms apart until it can establish baud rate. When the baud rate has been set, the CR23X sends a carriage return, line feed, "∗".

The ME line is held high until the CR23X receives an "E" to exit telecommunications. The ME is also lowered if a character is not received after 40 seconds in the Telecommunications Command State (2 minutes in the Remote Keyboard State).

Some modems are quite noisy when not on line; it is possible for valid characters to appear in the noise pattern. For this reason, the CR23X counts all the invalid characters it receives from the time it answers a ring, and terminates communication (lowers the ME line and returns to the ⁕ 0 Mode) after receiving 150 invalid characters.

6.2.5 SYNCHRONOUS DEVICE COMMUNICATION

Synchronous Devices (SDs) differ from enabled peripherals (Section 6.2.1) in that they are not enabled solely by a hardware line. An SD is enabled by an address synchronously clocked from the CR23X. Up to 16 SDs may be addressed by the CR23X, requiring only three pins of the 9-pin connector.

Synchronous Device Communication (SDC) discussed here is for those peripherals which connect to the 9-pin serial port. This should not be confused with Synchronous Device for Measurement (SDM) peripherals connected to control ports 1, 2, and 3. (Although the communication protocol for SDMs is very similar, their addressing is independent of SDC addresses and they do not have a ring line.)

SD STATES

The CR23X and the SDs use a combination of the Ring, Clock Handshake (CLK/HS) and Synchronous Device Enable (SDE) lines to establish communication. The CR23X can put the SDs into one of six states.

STATE 1, the SD Reset State

The CR23X forces the SDs to the reset/request state by lowering the SDE and CLK/HS lines. The SD cannot drive the CLK/HS or RXD lines in State 1, however, it can raise the Ring line if service is needed. The SD can never pull the Ring low if a Modem/Terminal is holding it high. Data on TXD is ignored by the SD.

STATE 2, the SD Addressing State

The CR23X places the SDs in the addressing state by raising CLK/HS followed by or simultaneously raising SDE (Figure 6.2-4). TXD must be low while SDE and CLK/HS are changing to the high state.

State 2 requires all SDs to drop the Ring line and prepare for addressing. The CR23X then synchronously clocks 8 bits onto TXD using CLK/HS as a clock. The least significant bit is transmitted first and is always logic high. Each bit transmitted is stable on the rising edge of CLK/HS. The SDs shift in bits from TXD on the rising edge of CLK/HS provided by the CR23X. The CR23X can only address one device per State 2 cycle. More than one SD may respond

to the address, however. State 2 ends when the 8th bit is received by the SD.

SDs implemented with shift registers decode the 4 most significant bits (bits 4, 5, 6, and 7) for an address. Bit 0 is always logic high. Bits 1, 2, and 3 are optional function selectors or commands. Addresses established to date are shown in Table 6.2-2 and are decoded with respect to the TXD line.

TABLE 6.2-2. SD Addresses

	B7	B6	B5	B4	B3	B2	B1	B0
VS1	0	1	1	0	X	X	X	1
SDC99 Printer	0	0	0	0	X	X	X	1
Storage Module	0	0	0	1	X	X	X	1
RF95 Modem	0	0	1	1	X	X	X	1

STATE 3, the SD Active State

The SD addressed by State 2, enters State 3. All other SDs enter State 4. An active SD returns to State 1 by resetting itself, or by the CR23X forcing it to reset.

Active SDs have different acknowledgment and communication protocols. Once addressed, the SD is free to use the CLK/HS, TXD, and RXD lines according to its protocol with the CR23X. The CR23X may also pulse the SDE line after addressing, as long as the CLK/HS and SDE are not low at the same time.

STATE 4, the SD Inactive State

The SDs not addressed by State 2 enter State 4, if not able to reset themselves (e.g., SM192 Storage Module). Inactive SDs ignore data on the TXD line and are not allowed to use the CLK/HS or RXD lines. Inactive SDs may raise the Ring line to request service.

STATE 5

State 5 is a branch from State 1 when the SDE line is high and the CLK/HS line is low. The SDs must drop the Ring line in this state. This state is not used by SDs. The CR23X must force the SDs back to the reset state from State 5 before addressing SDs.

STATE 6

State 6 is a branch from State 1, like State 5, except the SDE line is low and the CLK/HS line is high. The SDs must drop the Ring line in this state.

6.2.6 MODEM/TERMINAL AND COMPUTER REQUIREMENTS

The CR23X considers any device with an asynchronous serial communications port which raises the Ring line (and holds it high until the ME line is raised) to be a modem peripheral. Modems include Campbell Scientific phone modems, and most computers, terminals, and modems using the SC32A Optically Isolated RS-232 Interface.
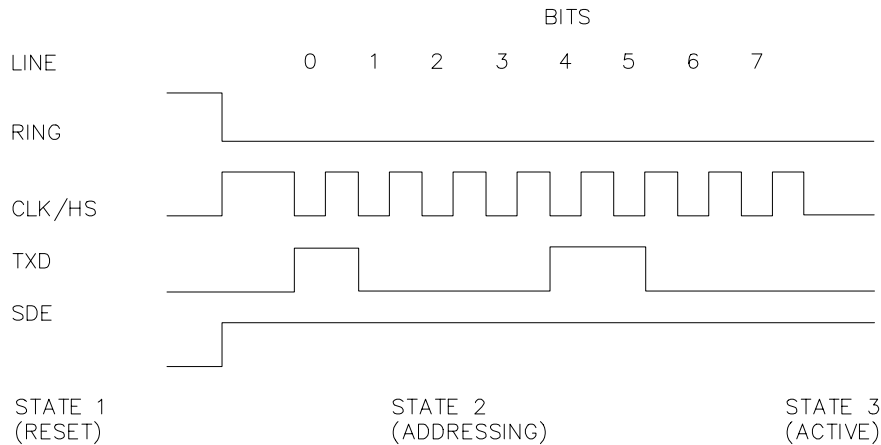
FIGURE 6.2-4. Addressing Sequence for the RF Modem

6.2.6.1 SC32A INTERFACE TO COMPUTER

Most computers require the SC32A Optically Isolated RS-232 Interface to communicate to the CS I/O port. (Direct connection to the CR23X is allowed through the "Computer RS-232" port.) The SC32A can pass data up to 19.2 K baud. The SC32A raises the CR23X's ring line when it receives characters from a modem, and converts the CR23X's logic levels (0 V logic low, 5V logic high) to RS-232 logic levels.

The SC32A 25-pin port is configured as Data Communications Equipment (DCE) (see Table 6.2-3) for direct connection to Data Terminal Equipment (DTE), which includes most PCs and printers.

When the SC32A receives a character from the terminal/computer (pin 2), 5 V is applied to the datalogger Ring line (pin 3) for one second or until the Modem Enable line (ME) goes high. The CR23X waits approximately 40 seconds to receive carriage returns, which it uses to establish baud rate. After the baud rate has been set the CR23X transmits a carriage return, line feed, "∗", and enters the Telecommunications Command State (Section 5). If the carriage returns are not received within the 40 seconds, the CR23X "hangs up".

TABLE 6.2-3. SC32A Pin Description

PIN = Pin number
O = Signal Out of the SC32A to a peripheral
I = Signal Into the SC32A from peripheral

25-PIN FEMALE PORT:

PIN #	I/O	ABBREVIATION
1		GROUND
2	I	TX
3	O	RX
4	I	RTS (POWER)
5	O	CTS
6	O	DSR
7		GROUND
8	O	DCD
20	I	DTR (POWER)

9-PIN MALE PORT:

PIN #	ABBREVIATION
1	+5V INPUT
2	GROUND
3	RING
4	RX
5	ME
6	SDE
9	TX

NOTE: The SC32A has a jumper, which when used, passes data only when the ME line is high and the SDE line is low. The function of the jumper is to block data sent to SDs from being received by a computer/terminal used to initiate data transfer. Synchronous data will appear as garbage characters on a computer/terminal.

6.2.6.2 SC932 INTERFACE TO MODEMS

Most modems have an RS-232 port configured as DCE. For connection of the CS I/O port to DCE devices such as modems and some computers, the SC932 9-pin to RS-232 DCE Interface should be used. The SC932 supports baud rates up to 19.2 K. Faster baud rates may be possible, depending on the device being interfaced.

6.2.6.3 COMPUTER/TERMINAL REQUIREMENTS

Computer/terminal peripherals are usually configured as Data Terminal Equipment (DTE). Pins 4 and 20 are used as handshake lines, which are set high when the serial port is enabled. Power for the SC32A RS-232 section is taken from these pins. For equipment configured as DTE (see Table 6.2-4) a direct ribbon cable connects the computer/terminal to the SC32A. Clear to Send (CTS) pin 5, Data Set Ready (DSR) pin 6, and Data Carrier Detect (DCD) pin 8 are held high by the SC32A (when the RS-232 section is powered) which should satisfy hardware handshake requirements of the computer/terminal.

Table 6.2-4 lists the most common RS-232 configuration for Data Terminal Equipment.

TABLE 6.2-4. DTE Pin Configuration

PIN = 25-pin connector number
ABR = Abbreviation for the function name
O = Signal Out of terminal to another device
I = Signal Into terminal from another device

PIN	ABR	I/O	FUNCTION
2	TD	O	Transmitted Data: Data is transmitted from the terminal on this line.
3	RD	I	Received Data: Data is received by the terminal on this line.
4	RTS	O	Request to Send: The terminal raises this line to ask a receiving device if the terminal can transmit data.
5	CTS	I	Clear to Send: The receiving device raises this line to let the terminal know that the receiving device is ready to accept data.
20	DTR	O	Data Terminal Ready: The terminal raises this line to tell the modem to connect itself to the telephone line.
6	DSR	I	Data Set Ready: The modem raises this line to tell the terminal that the modem is connected to the phone line.
8	DCD	I	Data Carrier Detect: The modem raises this line to tell the terminal that the modem is receiving a valid carrier signal from the phone line.
22	RI	I	Ring Indicator: The modem raises this line to tell the terminal that the phone is ringing.
7	SG		Signal Ground: Voltages are measured relative to this point.

6.2.6.3 COMMUNICATION PROTOCOL/TROUBLE SHOOTING

The ASCII standard defines an alphabet consisting of 128 different characters where each character corresponds to a number, letter, symbol, or control code.

An ASCII character is a binary digital code composed of a combination of seven "bits", each bit having a binary state of 1 (one) or 0 (zero). For example, the binary equivalent for the ASCII character "1" is 0110001 (decimal 49).

ASCII characters are transmitted one bit at a time, starting with the 1st (least significant) bit. During data transmission the marking condition is used to denote the binary state 1, and the spacing condition for the binary state 0. The signal is considered marking when the voltage is more negative than minus three volts with respect to ground, and spacing when the voltage is more positive than plus three volts.

Most computers use 8-bits (1 byte) for data communications. The 8th bit is sometimes used for a type of error checking called parity-checking. Even parity binary characters have an even number of 1's, odd-parity characters have an odd number of 1's. When parity checking is used, the 8th bit is set to either a 1 or a 0 to make the parity of the character correct. The CR23X ignores the 8th bit of a character that is receives, and transmits the 8th bit as a binary 0. This method is generally described as "no parity".

To separate ASCII characters, a Start bit is sent before the 1st bit and a Stop bit is sent after the 8th bit. The start bit is always a space, and the stop bit is always a mark. Between characters the signal is in the marking condition.

Figure 6.2-5 shows how the ASCII character "1" is transmitted. When transmitted by the CR23X using the SC32A RS-232 interface spacing and marking voltages are positive and negative, as shown. Signal voltages at the CS I/O port are 5V in the spacing condition, and 0V in the marking condition.
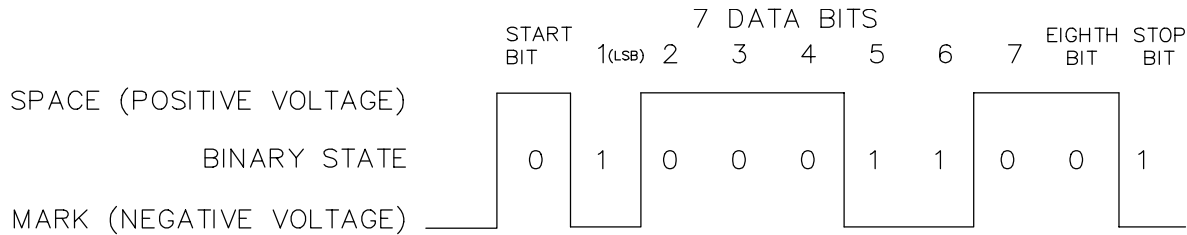
FIGURE 6.2-5. Transmitting the ASCII Character 1

BAUD RATE

BAUD RATE is the number of bits transmitted per second. The CR23X can communicate at 300, 1200, 4800, 9600, 19200, 38400, and 76800 baud. In the Telecommunications State, the CR23X will set its baud rate to match the baud rate of the computer/terminal. Some baud rates, particularly those above 9600, may not be supported by all CSI communications equipment.

Typically the baud rate of the modem/computer/ terminal is set either with dip switches, or programmed from the keyboard. The instrument's instruction manual should explain how to set it.

DUPLEX

Full duplex means that two devices can communicate in both directions simultaneously. Half duplex means that the two devices must send and receive alternately. Full duplex should always be specified when communicating with Campbell Scientific peripherals and modems. However, communication between some Campbell Scientific modems (such as the RF95 RF modem) is carried out in a half duplex fashion. This can affect the way commands should be sent to and received from such a modem, especially when implemented by computer software.

To overcome the limitations of half duplex, some communications links expect a terminal sending data to also write the data to the screen. This saves the remote device having to echo that data back. If, when communicating with a Campbell Scientific device, characters are displayed twice (in pairs), it is likely that the terminal is set to half duplex rather than the correct setting of full duplex.

IF NOTHING HAPPENS

If the CR23X is connected to the SC32A RS-232 interface and a modem/terminal, and an "∗" is not received after sending carriage returns:

1. Verify that the CR23X has power AT THE 12V AND GROUND INPUTS, and that the cables connecting the devices are securely connected.

2. Verify that the port of the modem/terminal is an asynchronous serial communications port configured as DTE (see Table 6.2-4). The most common problems occur when the user tries to use a parallel port, or doesn't know the port assignments, i.e. COM1 or COM2. IBM, and most compatibles come with a Diagnostic disk which can be used to identify ports, and their assignments. If the serial port is standard equipment, then the operators manual should give you this information.

3. Verify that there is 5 volts between the CR23X 5V and G terminals. Call Campbell Scientific technical support if the voltage is less than 4.8 volts.

Some serial ports, e.g., the Super Serial Card for Apple computers, can be configured as DTE or DCE with a jumper block. Pin functions must match Table 6.2-4.

If you are using a computer to communicate with the datalogger, communication software must be used to enable the serial port and to make the computer function as a terminal. The port should be enabled for 300, 1200, or 9600 baud, 8 data bits, 1 stop bit, and no parity. Campbell Scientific's PC208W provides this function.

If you are not sure that your computer/terminal is sending or receiving characters, there is a simple way to verify it. Make sure that the duplex is set to full. Next, take a paper clip and connect one end to pin 2, and the other end to pin 3 of the serial port. Each character typed on the keyboard will be displayed only if transmitted from the terminal on pin 2, and received on pin 3 (if duplex is set to half, the character will be displayed once if it is not transmitted, or twice if it is transmitted).

IF GARBAGE APPEARS

If garbage characters appear on the display, check that the baud rate is supported by the CR23X. If the baud rate is correct, verify that the computer/terminal is set for 8 data bits, and no parity. Garbage will appear if 7 data bits and no parity are used. If the computer/terminal is set to 8 data bits and even or odd parity, communication cannot be established.

6.3 USE OF INSTRUCTION 96

Instruction 96 is used for on-line data transfer to peripherals (Section 4.1). Each peripheral connected to the CR23X requires an Instruction 96 with the appropriate parameter. If the CR23X is already communicating on the 9-pin connector when Instruction 96 is executed, the instruction puts the output request in a "queue" and program execution continues. As the 9-pin connector becomes available, each device in the queue will get its turn until the queue is empty.

Instruction 96 is aborted if a modem raises the Ring line. Data transfer to an addressed peripheral is aborted if the ring line is raised by an RF Modem configured as a synchronous device. Transfer of data is not resumed until the next time Instruction 96 is executed and the datalogger has exited telecommunications.

The ⁕ 8 Mode is used to manually initiate data transfer from Final Storage to a peripheral. An abort flag is set if any key on the CR23X or terminal is pressed during the transfer. Data transfer is stopped and the memory location displayed when the flag is set. During ⁕ 8 data transfer the abort flag is checked as follows:

1. Comma separated ASCII - after every 32 characters.
2. Printable ASCII - after every line.
3. Binary - after every 256 Final Storage locations.

TABLE 6.3-1. CS I/O Pin Description

ABR = Abbreviation for the function name.
PIN = Pin number.
O = Signal Out of the CR23X to a peripheral.
I = Signal Into the CR23X from a peripheral.

PIN	ABR	I/O	Description
1	5 V	O	5V: Sources 5 VDC, used to power peripherals.
2	SG		Signal Ground: Provides a power return for pin 1 (5V), and is used as a reference for voltage levels.
3	RING	I	Ring: Raised by a peripheral to put the CR23X in the telecommunications mode.
4	RXD	I	Receive Data: Serial data transmitted by a peripheral are received on pin 4.
5	ME	O	Modem Enable: Raised when the CR23X determines that a modem raised the ring line.
6	SDE	O	Synchronous Device Enable: Used to address Synchronous Devices (SDs), and can be used as an enable line for printers.
7	CLK/HS	I/O	Clock/Handshake: Used with the SDE and TXD lines to address and transfer data to SDs. When not used as a clock, pin 7 can be used as a handshake line (during printer output, high enables, low disables).
8	12 V	O	12 V: Sources 12 VDC, used to power 12 VDC peripherals.
9	TXD	O	Transmit Data: Serial data are transmitted from the CR23X to peripherals on pin 9; logic low marking (0V) logic high spacing (5V) standard asynchronous ASCII, 8 data bits, no parity, 1 start bit, 1 stop bit, 300, 1200, 9600, 76,800 baud (user selectable).

SECTION 7. MEASUREMENT PROGRAMMING EXAMPLES

This section gives some examples of Input Programming for common sensors used with the CR23X. These examples detail only the connections, Input, Program Control, and Processing Instructions necessary to perform measurements and store the data in engineering units in Input Storage. Output Processing Instructions are omitted. It is left for the user to program the necessary instructions to obtain the final data in the form desired. NO OUTPUT TO FINAL STORAGE WILL TAKE PLACE WITHOUT ADDITIONAL PROGRAMMING.

The examples given in this section would likely be only fragments of larger programs. In general, the examples are written with the measurements made by the lowest numbered channels, the instructions at the beginning of the program table, and low number Input Storage locations used to store the data. It is unlikely that an application and CR23X configuration exactly duplicates that assumed in an example. THESE EXAMPLES ARE NOT MEANT TO BE USED VERBATIM; SENSOR CALIBRATION AND INPUT LOCATIONS SELECTED MUST BE ADJUSTED FOR THE ACTUAL CIRCUMSTANCES. UNLESS OTHERWISE NOTED, ALL EXCITATION CHANNELS ARE SWITCHED ANALOG OUTPUT.

7.1 SINGLE-ENDED VOLTAGE/ SWITCHED 12 V TERMINAL - CS500

The CS500 is a modified Vaisala 50Y Humitter temperature and relative humidity sensor. It has high level linear output of 0 to 1 V for the temperature range of -40° to +60°C and relative humidity of 0 to 100%. It is measured with Instruction 1 (Volts SE). The multiplier for temperature is found with the following relationship [60°C - (-40°C)] / [1000 mV - 0 mV] = 0.1°C/mV. The offset is -40°C. The multiplier for relative humidity is [100 % - 0 %] / [1000 mV - 0 mV] = 0.1 %/mV and the offset is 0 %. The CS500 is powered by the CR23X's 12 V battery and draws <2 mA of current while on. Battery power can be conserved by turning the CS500 on just prior to making the measurement and turning it off after the measurement is completed. This is done with the Switched 12 V terminal on the CR23X wiring panel.

CONNECTIONS

The CS500 output is measured using two single-ended voltage measurements on analog inputs 5 and 6. Single-ended analog inputs are labeled in blue on the CR23X wiring panel. A wiring diagram on connections between the CR23X and the CS500 is given in Figure 7.1-1.

> **CAUTION:** The Switched 12 V Control terminal will be permanently damaged if 12 V is applied to it. Do not connect 12 V to the Switched 12 V Control terminal.

PROGRAM

```
;Turn CS500 on.
;
01:    Do (P86)
       1:  49          Set Switched 12 V High

;Allow CS500 to warm up and stabilize on
;the Temperature and Relative Humidity.
;
02:    Excitation with Delay (P22)
       1:  3           Ex Channel
       2:  0           Delay W/Ex (units = 0.01 sec)
       3:  10          Delay After Ex (units = 0.01 sec)
       4:  0           mV Excitation

;Measure Temperature.
;
03:    Volts (SE) (P1)
       1:  1           Reps
       2:  25          ±5000 mV Slow 60 Hz
                       Rejection Range
       3:  5           SE Channel
       4:  1           Loc [ Temp_C   ]
       5:  .1          Mult
       6: -40          Offset
```

SECTION 7. MEASUREMENT PROGRAMMING EXAMPLES

;Measure Relative Humidity.
;
```
04:    Volts (SE) (P1)
       1:   1         Reps
       2:   25        ±5000 mV Slow 60 Hz
                      Rejection Range
       3:   6         SE Channel
       4:   2         Loc [ RH_pct   ]
       5:    .1       Mult
       6:   0         Offset
```

;Turn CS500 off.
;
```
05:    Do (P86)
       1:  59          Set Switched 12 V Low
```

INPUT LOCATIONS

1 Temp_C
2 RH_pct
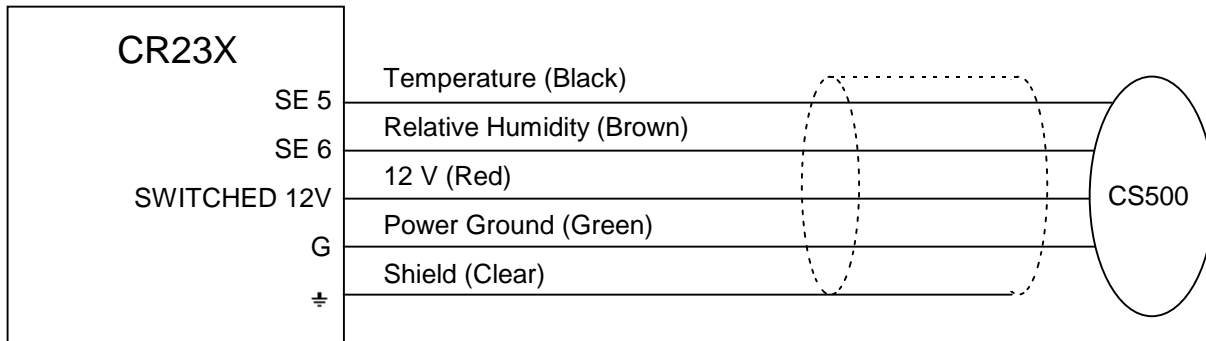
FIGURE 7.1-1. Wiring Diagram for CS500
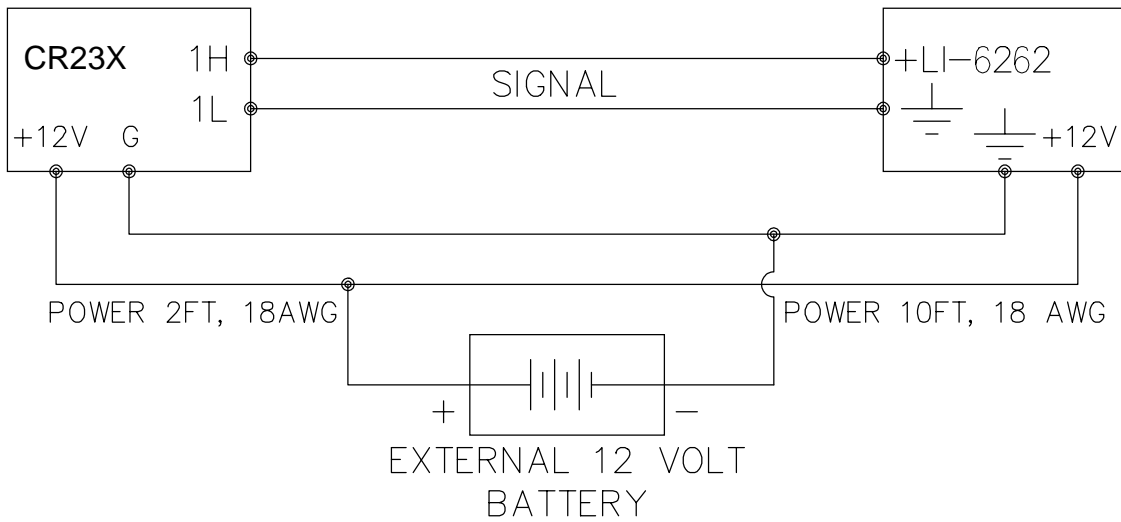
FIGURE 7.1-2. Typical Connection for Active Sensor with External Battery

7.2 DIFFERENTIAL VOLTAGE MEASUREMENT

Some sensors either contain or require active signal conditioning circuitry to provide an easily measured analog voltage output. Generally, the output is referenced to the sensor ground. The associated current drain usually requires a power source external to the CR23X. A typical connection scheme where AC power is not available and both the CR23X and sensor are powered by an external battery is shown in Figure 7.2-1. Since a single-ended measurement is referenced to the CR23X ground, any voltage difference between the sensor ground and CR23X ground becomes a measurement error. A differential measurement avoids this error by measuring the signal between the 2 leads without reference to ground. This example analyzes the potential error on a differential CO_2 measurement using a LI-COR CO_2/H_2O analyzer, model LI-6262.

The wire used to supply power from the external battery is 18 AWG with an average resistance of 6.5 ohms/1000 ft. The power leads to the CR23X and LI-6262 are 2 ft and 10 ft, respectively. Typical current drain for the LI-6262 is 1000 mA. When making measurements, the CR23X draws about 35 mA. Since voltage is equal to current multiplied by resistance (V=IR), ground voltages at the LI-6262 and the CR23X relative to battery ground are:

$$LI\text{-}6262 \text{ ground} =$$
$$1A * 6.5 \text{ ohms}/1000 \text{ ft} * 10 \text{ ft} = +0.065 \text{ V}$$

$$CR23X \text{ ground} =$$
$$0.035A * 6.5 \text{ ohms}/1000 \text{ ft} * 2 \text{ ft} = +0.0005 \text{ V}$$

Ground at the LI-6262 is 0.065 V higher than ground at the CR23X. The LI-6262 can be programmed to output a linear voltage (0 to 100 mV) that is proportional to differential CO_2, 100 µmol/mol full scale, or 1 µmol/mol/mV. If the output is measured with a single-ended voltage measurement, it is 0.065 V or 65 µmol/mol high. If this offset remained constant, it could be corrected in programming. However, it is better to use a differential voltage measurement which does not rely on the current drain remaining constant. The program that follows illustrates the use of Instruction 2 to make the measurement. A multiplier of 1 is used to convert the millivolt output into µmol/mol.

PROGRAM

```
01:   Volt (Diff) (P2)
      1:   1        Reps
      2:   23       ±200 mV 60 Hz Rejection
                    Range
      3:   1        DIFF Channel
      4:   1        Loc [ umol_mol  ]
      5:   1        Mult
      6:   0        Offset
```

7.3 THERMOCOUPLE TEMPERATURES USING CR23X REFERENCE

The use of the built in CR23X thermocouple reference thermistor is described in the introductory programming example (Section OV4).

7.4 THERMOCOUPLE TEMPERATURES USING AN EXTERNAL REFERENCE JUNCTION

When a number of thermocouple measurements are made at some distance from the CR23X, it is often better to use a reference junction box located at the site rather than using the CR23X panel for the reference junction. This reduces the required length of expensive thermocouple wire as regular copper wire can be used between the junction box (J-box) and CR23X. In addition, if the temperature gradient between the J-box and the thermocouple measurement junction is smaller than the gradient between the CR23X and the measurement junction, thermocouple accuracy is improved. In the following example, an external reference junction is used on 5 thermocouple measurements. A Campbell Scientific 107 Temperature Probe is used to measure the reference temperature. The connection scheme is shown in Figure 7.4-1.
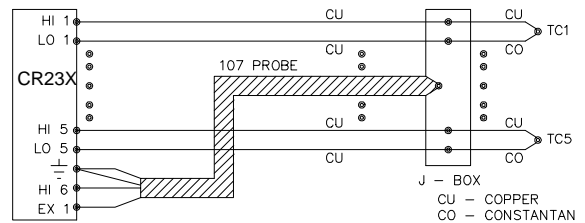
FIGURE 7.4-1. Thermocouples with External Reference Junction

The temperature of the 107 Probe is stored in input location 1 and the thermocouple temperatures in Locations 2-6.

PROGRAM

```
1:  Temp (107) (P11)
    1:      1          Reps
    2:      1          SE Channel
    3:      1          Excite w/E1+reps
    4:      1          Loc [ REF_TEMP ]
    5:      1.0        Mult
    6:      0          Offset

2:  Thermocouple Temp (DIFF) (P14)
    1:      5          Reps
    2:      21         10 mV, 60 Hz Reject, Slow
                       Range
    3:      1          DIFF Channel
    4:      1          Type T (Copper-Constantan)
    5:      1          Ref Temp (Deg. C) Loc [
                       REF_TEMP ]
    6:      2          Loc [ TC_1    ]
    7:      1.0        Mult
    8:      0.0        Offset
```

7.5 107 TEMPERATURE PROBE

Instruction 11 is designed to excite and measure the Campbell Scientific 107 Thermistor Probe (or the thermistor portion of the 207 temperature and relative humidity probe) and convert the measurement into temperature (^{o}C). In this example, temperatures are obtained from three 107 probes. The measurements are made on single-ended channels 1-3, and the temperatures are stored in input locations 1-3.

CONNECTIONS

The black leads from the probes go to excitation channel 1, the purple and clear leads go to a ⏚ channel, and the red leads go to single- ended channels 1, 2, and 3 (high and low sides of differential channel 1 and high side of 2).

PROGRAM

```
1:  Temp (107) (P11)
    1:      3          Reps
    2:      1          SE Channel
    3:      1          Excite all reps w/E1
    4:      1          Loc [ TEMP_1   ]
    5:      1.0        Mult
    6:      0.0        Offset
```

7.6 ANEMOMETER WITH PHOTOCHOPPER OUTPUT

An anemometer with a photochopper transducer produces a pulsed output which is monitored with the Pulse Count Instruction, configured for High Frequency Pulses. The anemometer used in this example is the R.M. Young Model No. 12102D Cup Anemometer which has a 10 window chopper wheel. The photochopper circuitry is powered from the CR23X 12V supply. Supplemental charging, AC or solar, should be used with the CR23X. If a charging source is not practical, back-up batteries should be used to compensate for the increased current drain.

Wind speed is desired in meters per second. There is a pulse each time a window in the chopper wheel, which revolves with the cups, allows light to pass from the source to the photoreceptor. Because there are 10 windows in the chopper wheel, there are 10 pulses per revolution. Thus, 1 rpm is equal to 10 pulses per 60 seconds (1 minute) or 6 rpm = 1 pulse per second. The manufacturer's calibration for relating wind speed to rpm is:

$$\text{Wind speed (m/s)} = 0.01632 \text{ m/s/rpm} \times \text{rpm} + 0.2 \text{ m/s}$$

Pulse count instruction has the option of converting counts to frequency in Hz (counts/second). The multiplier and offset to convert Hz to meters per second are:

$$\text{m/s} = 0.01632 \text{ m/s/rpm} \times 6 \text{ rpm/Hz} \times X\text{Hz} + 0.2 \text{ m/s}$$

$$= 0.0979 \text{ m/s/Hz} \times X\text{Hz} + 0.2 \text{ m/s}$$

```
                                  ANEMOMETER
┌─────────┐
│  + 12 ●─┼────────────────────● +12V SUPPLY
│         │
│ CR23X   │
│   ⏚     ●────────────────────● GROUND AND SIGNAL REFERENCE
│         │
│   P1  ●─┼────────────────────● PULSE OUT
└─────────┘
```
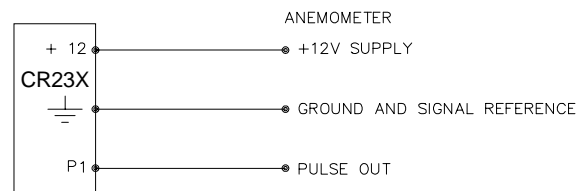
FIGURE 7.6-1. Wiring Diagram for Anemometer

PROGRAM

```
1: Pulse (P3)
  1:     1        Reps
  2:     1        Pulse Channel 1
  3:    20        High Frequency, Output Hz
  4:     1        Loc [ WS_m_s   ]
  5:      .0979   Mult
  6:      .2      Offset
```

7.7 TIPPING BUCKET RAIN GAUGE WITH LONG LEADS

A tipping bucket rain gauge is measured with the Pulse Count Instruction configured for Switch Closure. Counts from long intervals will be used (an option in Parameter 3), as the final output desired is total rainfall (obtained with Instruction 72, Totalize). If counts from long intervals were discarded, less rainfall would be recorded than was actually measured by the gauge (assuming there were counts in the long intervals). Output is desired in millimeters of precipitation. The gauge is calibrated for a 0.01 inch tip; a multiplier of 0.254 is used.
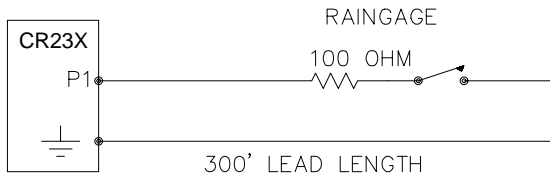
FIGURE 7.7-1. Wiring Diagram for Rain Gauge with Long Leads

In a long cable there is appreciable capacitance between the lines, which is discharged across the switch when it closes. In addition to shortening switch life, a transient may be induced in other wires, packaged with the rain gauge leads, each time the switch closes. The 100 ohm resistor protects the switch from arcing and the associated transient from occurring, and should be included any time leads longer than 100 feet are used with a switch closure.

PROGRAM

```
1: Pulse (P3)
  1:     1        Reps
  2:     1        Pulse Channel 1
  3:     2        Switch Closure, All Counts
  4:     1        Loc [ InchRain  ]
  5:      .254    Mult
  6:     0.0      Offset
```

7.8 100 OHM PRT IN 4 WIRE HALF BRIDGE

Instruction 9 is the best choice for accuracy where a 100 ohm Platinum Resistance Thermometer (PRT) is separated from other bridge completion resistors by a lead length having more than a few thousandths of an ohm resistance. In this example, it is desired to measure a temperature in the range of -10 to 40ºC. The length of the cable from the CR23X to the PRT is 500 feet.
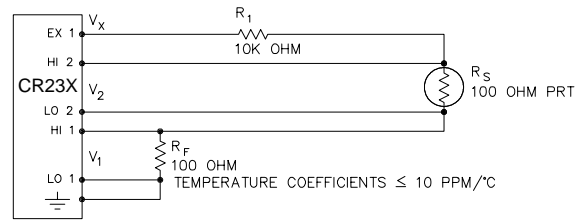
FIGURE 7.8-1. Wiring Diagram for PRT in 4 Wire Half Bridge

Figure 7.8-1 diagrams the circuit used to measure the PRT. The 10 kohm resistor allows the use of a high excitation voltage and a low input range. This insures that noise in the excitation does not have an effect on signal noise. Because the fixed resistor (R_f) and the PRT (R_s) have approximately the same resistance, the differential measurement of the voltage drop across the PRT can be made on the same range as the differential measurement of the voltage drop across R_f. The use of the same range eliminates any range translation error that might arise from the 0.01% tolerance of the range translation resistors in the CR23X.

If the voltage drop across the PRT (V_2) is kept on the 50mV range, self heating of the PRT should be less than 0.001ºC in still air. The resolution of the measurement is increased as the excitation voltage (V_x) is increased as long as the Input Range is not exceeded. The voltage drop across the PRT is equal to V_x multiplied by the ratio of R_s to the total resistance, and is greatest when R_s is greatest (R_s=115.54 ohms at 40ºC). To find the maximum excitation voltage that can be used, we assume V_2 equal to 50mV and use Ohm's Law to solve for the resulting current, I.

7-5

I = 50mV/Rs = 50mV/115. 54 ohms = 0.433mA

Next solve for V_x:

$$V_x = I(R_1 + R_s + R_f) = 4.42V$$

If the actual resistances were the nominal values, the CR23X would not overrange with V_x = 4.4V. To allow for the tolerances in the actual resistances, it is decided to set V_x equal to 4.2 volts (e.g., if the 10 kohms resistor is 5% low, $R_s/(R_1 + R_s + R_f) = 115.54/9715.54$, and V_x must be 4.204V to keep V_s less than 50mV).

The result of Instruction 9 when the first differential measurement (V_1) is not made on the 5V range is equivalent to R_s/R_f. Instruction 16 computes the temperature (^oC) for a DIN 43760 standard PRT from the ratio of the PRT resistance to its resistance at 0^oC (R_s/R_0). Thus, a multiplier of R_f/R_0 is used in Instruction 9 to obtain the desired intermediate, R_s/R_0 ($=R_s/R_f \times R_f/R_0$). If R_s and R_0 were each exactly 100 ohms, the multiplier would be 1. However, neither resistance is likely to be exact. The correct multiplier is found by connecting the PRT to the CR23X and entering Instruction 9 with a multiplier of 1. The PRT is then placed in an ice bath (0^oC; $R_s = R_0$), and the result of the bridge measurement is read using the ⁕ 6 Mode. The reading is R_s/R_f, which is equal to R_0/R_f since $R_s = R_0$. The correct value of the multiplier, R_f/R_0, is the reciprocal of this reading. The initial reading assumed for this example was 0.9890. The correct multiplier is: $R_f/R_0 = 1/0.9890 = 1.0111$.

The fixed 100 ohm resistor must be thermally stable. Its precision is not important because the exact resistance is incorporated, along with that of the PRT, into the calibrated multiplier. The 10 ppm/^oC temperature coefficient of the fixed resistor will limit the error due to its change in resistance with temperature to less than 0.15^oC over the -10 to 40^oC temperature range. Because the measurement is ratiometric (R_s/R_f), the properties of the 10 kohm resistor do not affect the result.

A terminal input module (Model 4WPB100) can be used to complete the circuit shown in Figure 7.8-1.

PROGRAM

1: Full Bridge w/mv Excit (P9)
1:	1	Reps
2:	22	50 mV, 60 Hz Reject, Slow, Ex Range
3:	22	50 mV, 60 Hz Reject, Slow, Br Range
4:	1	DIFF Channel
5:	1	Excite all reps w/Exchan 1
6:	4400	mV Excitation
7:	1	Loc [Rs_Ro]
8:	1.0111	Mult
9:	0.0	Offset

2: Temperature RTD (P16)
1:	1	Reps
2:	1	R/R0 Loc [Rs_Ro]
3:	2	Loc [TEMP_degC]
4:	1.0	Mult
5:	0.0	Offset

7.9 100 OHM PRT IN 3 WIRE HALF BRIDGE

The temperature measurement requirements in this example are the same as in Section 7.9. In this case, a three wire half bridge, Instruction 7, is used to measure the resistance of the PRT. The diagram of the PRT circuit is shown in Figure 7.9-1.
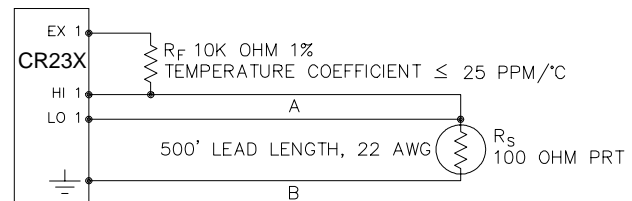
FIGURE 7.9-1. 3 Wire Half Bridge Used to Measure 100 ohm PRT

As in the example in Section 7.8, the excitation voltage is calculated to be the maximum possible, yet allow the +50mV measurement range. The 10 kohm resistor has a tolerance of ±1%; thus, the lowest resistance to expect from it is 9.9 kohms. We calculate the maximum excitation voltage (V_x) to keep the voltage drop across the PRT less than 50mV:

$$0.050V > V_x \, 115.54/(9900 + 115.54); \ \ V_x < 4.33V$$

The excitation voltage used is 4.3V.

The multiplier used in Instruction 7 is determined in the same manner as in Section 7.8. In this example, the multiplier (R_f/R_0) is assumed to be 100.93.

The 3 wire half bridge compensates for lead wire resistance by assuming that the resistance of wire A is the same as the resistance of wire B. The maximum difference expected in wire resistance is 2%, but is more likely to be on the order of 1%. The resistance of R_s calculated with Instruction 7, is actually R_s plus the difference in resistance of wires A and B. The average resistance of 22 AWG wire is 16.5 ohms per 1000 feet, which would give each 500 foot lead wire a nominal resistance of 8.3 ohms. Two percent of 8.3 ohms is 0.17 ohms. Assuming that the greater resistance is in wire B, the resistance measured for the PRT (R_0 = 100 ohms) in the ice bath would be 100.17 ohms, and the resistance at 40ºC would be 115.71. The measured ratio R_s/R_0 is 1.1551; the actual ratio is 115.54/100 = 1.1554. The temperature computed by Instruction 16 from the measured ratio would be about 0.1ºC lower than the actual temperature of the PRT. This source of error does not exist in the example in Section 7.8, where a 4 wire half bridge is used to measure PRT resistance.

The advantages of the 3 wire half bridge are that it only requires 3 lead wires going to the sensor and takes 2 single- ended input channels, whereas the 4 wire half bridge requires 4 wires and 2 differential channels.

A terminal input module (Model 3WHB10K) can be used to complete the circuit in Figure 7.9-1. It uses a ±0.01% ±8 ppm precision resistor.

PROGRAM

```
1: 3W Half Bridge (P7)
  1:      1         Reps
  2:     22         50 mV, 60 Hz Reject, Slow
                    Range
  3:      1         SE Channel
  4:      1         Excite all reps w/Exchan 1
  5:   4300         mV Excitation
  6:      1         Loc [ Rs_Ro    ]
  7:  100.93        Mult
  8:    0.0         Offset
```

```
2: Temperature RTD (P16)
  1:      1         Reps
  2:      1         R/R0 Loc [ Rs_Ro    ]
  3:      2         Loc [ TEMP_degC ]
  4:    1.0         Mult
  5:    0.0         Offset
```

7.10 100 OHM PRT IN 4 WIRE FULL BRIDGE

This example describes obtaining the temperature from a 100 ohm PRT in a 4 wire full bridge (Instruction 6). The temperature being measured is in a constant temperature bath and is to be used as the input for a control algorithm. The PRT in this case does not adhere to the DIN standard (alpha = 0.00385) used in the temperature calculating Instruction 16. Alpha is defined as (R_{100}/R_0-1)/100, where R_{100} and R_0 are the resistances of the PRT at 100ºC and 0ºC, respectively. In this PRT, alpha is equal to 0.00392.
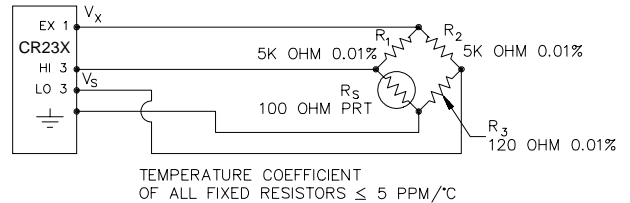
FIGURE 7.10-1. Full Bridge Schematic for 100 ohm PRT

The result (X) given by Instruction 6 is 1000 V_s/V_x (where V_s is the measured bridge output voltage, and V_x is the excitation voltage) which is:

$$X = 1000 (R_s/(R_s+R_1)-R_3/(R_2+R_3))$$

The resistance of the PRT (R_s) is calculated with the Bridge Transform Instruction 59:

$$R_s = R_1 X'/(1-X')$$

Where

$$X' = X/1000 + R_3/(R_2+R_3)$$

Thus, to obtain the value R_s/R_0, ($R_0 = R_s$ @ 0ºC) for the temperature calculating Instruction 16, the multiplier and offset used in Instruction 6 are 0.001 and $R_3/(R_2+R_3)$, respectively. The multiplier used in Instruction 59 to obtain R_s/R_0 is R_1/R_0 (5000/100 = 50).

It is desired to control the temperature bath at 50°C with as little variation as possible. High resolution is desired so the control algorithm will be able to respond to minute changes in temperature. The highest resolution is obtained when the temperature range results in an output voltage (V_s) range which fills the measurement range selected in Instruction 6. The full bridge configuration allows the bridge to be balanced (V_s = 0V) at or near the control temperature. Thus, the output voltage can go both positive and negative as the bath temperature changes, allowing the full use of the measurement range.

The resistance of the PRT is approximately 119.7 ohms at 50°C. The 120 ohm fixed resistor balances the bridge at approximately 51°C. The output voltage is:

$$V_s = V_x [R_s/(R_s+R_1) - R3/(R_2+R_3)]$$

$$= V_x [R_s/(R_s+5000) - 0.023438]$$

The temperature range to be covered is 50 ±10°C. At 40°C R_s is approximately 115.8 ohms, or:

$$V_s = -802.24 \times 10^{-6} V_x$$

even with an excitation voltage (V_x) equal to 5000mV, V_s can be measured on the ±10mV scale (40°C = 115.8 Ω = -4.01mV, 60°C = 123.6 Ω = 3.428mV). There is a change of approximately 4mV from the output at 40°C to the output at 51°C, or 364μV/°C. With a resolution of 0.33μV on the ±10mV range, this means that the temperature resolution is 0.0009°C.

The ±5 ppm per °C temperature coefficient of the fixed resistors was chosen so that their ±0.01% accuracy tolerance would hold over the desired temperature range.

The relationship between temperature and PRT resistance is a slightly nonlinear one. Instruction 16 computes this relationship for a DIN standard PRT where the nominal temperature coefficient is 0.00385/°C. The change in nonlinearity of a PRT with the temperature coefficient of 0.00392/°C is minute compared with the slope change. Entering a slope correction factor of 0.00385/0.00392 = 0.98214 as the multiplier in Instruction 16 results in a calculated temperature which is well within the accuracy specifications of the PRT.

PROGRAM

```
1: Full Bridge (P6)
  1:       1        Reps
  2:      21        10 mV, 60 Hz Reject, Slow
                    Range
  3:       3        DIFF Channel
  4:       1        Excite all reps w/Exchan 1
  5:    5000        mV Excitation
  6:       1        Loc [ Rs_Ro    ]
  7:      .001      Mult
  8:    .02344      Offset

2: BR Transform Rf[X/(1-X)] (P59)
  1:       1        Reps
  2:       1        Loc [ Rs_Ro    ]
  3:      50        Multiplier (Rf)

3: Temperature RTD (P16)
  1:       1        Reps
  2:       1        R/R0 Loc [ Rs_Ro     ]
  3:       2        Loc [ TEMP_degC ]
  4:    .98214      Mult
  5:     0.0        Offset
```

7.11 PRESSURE TRANSDUCER - 4 WIRE FULL BRIDGE

This example describes a measurement made with a Druck PDCR 10/D depth measurement pressure transducer. The pressure transducer was ordered with passive temperature compensation for use with positive or negative excitation and has a range of 5 psi or about 3.5 meters of water. The transducer is used to measure the depth of water in a stilling well.

Instruction 6, 4 Wire Full Bridge, is used to measure the pressure transducer. The high output of the semiconductor strain gauge necessitates the use of the ±50mV input range. The sensor is calibrated by connecting it to the CR23X and using Instruction 6 with a multiplier of 1 and an offset of 0, noting the readings (* 6 Mode) with 10 cm of water above the sensor and with 334.6 cm of water above the sensor. The output of Instruction 6 is 1000 V_s/V_x or millivolts per volt excitation. At 10 cm the reading is 0.19963mV/V and at 334.6 cm the reading is 6.6485mV/V. The multiplier to yield output in cm is:

(334.6 - 10)/(6.6485-.19963) = 50.334 cm/mV/V

The offset is determined after the pressure transducer is installed in the stilling well. The sensor is installed 65 cm below the water level at the time of installation. The depth of water at this time is determined to be 72.6 cm relative to the desired reference. When programmed with the multiplier determined above and an offset of 0, a reading of 65.12 is obtained. The offset for the actual measurements is thus determined to be 72.6 - 65.12 = 7.48 cm.

The lead length is approximately 10 feet, so there is no appreciable error due to lead wire resistance. Example 7.13 shows a means of compensating for long lead lengths.
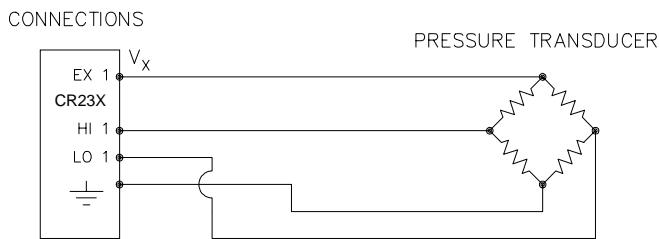
FIGURE 7.11-1. Wiring Diagram for Full Bridge Pressure Transducer

PROGRAM

```
1:  Full Bridge (P6)
    1:       1           Reps
    2:      22           50 mV, 60 Hz Reject, Slow
                         Range
    3:       1           DIFF Channel
    4:       1           Excite all reps w/Exchan 1
    5:    5000           mV Excitation
    6:       1           Loc [ WATER_cm  ]
    7:   50.334          Mult
    8:    7.48           Offset
```

7.12 LYSIMETER - 6 WIRE FULL BRIDGE

When a long cable is required between a load cell and the CR23X, the resistance of the wire can create a substantial error in the measurement if the 4 wire full bridge (Instruction 6) is used to excite and measure the load cell. This error arises because the excitation voltage is lower at the load cell than at the CR23X due to voltage drop in the cable. The 6 wire full bridge (Instruction 9) avoids this problem by measuring the excitation voltage at the load cell. This example shows the errors

one would encounter if the actual excitation voltage was not measured and shows the use of a 6 wire full bridge to measure a load cell on a weighing lysimeter (a container buried in the ground, filled with plants and soil, used for measuring evapotranspiration).

The lysimeter is 2 meters in diameter and 1.5 meters deep. The total weight of the lysimeter with its container is approximately 8000 kg. The lysimeter has a mechanically adjustable counterbalance, and changes in weight are measured with a 250 pound (113.6 kg) capacity Sensotec Model 41 tension/compression load cell. The load cell has a 4:1 mechanical advantage on the lysimeter (i.e., a change of 4 kg in the mass of the lysimeter will change the force on the load cell by 1 kg-force or 980 N).
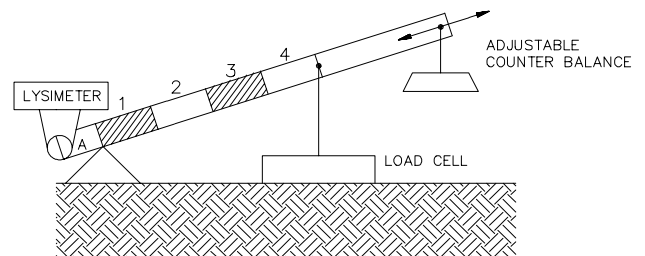
FIGURE 7.12-1. Lysimeter Weighing Mechanism

The surface area of the lysimeter is 3.1416 m^2 or 31,416 cm^2, so 1 cm of rainfall or evaporation results in a 31.416 kg change in mass. The load cell can measure ±113.6 kg, a 227 kg range. This represents a maximum change of 909 kg (28 cm of water) in the lysimeter before the counterbalance would have to be readjusted.

There is 1000 feet of 22 AWG cable between the CR23X and the load cell. The output of the load cell is directly proportional to the excitation voltage. When Instruction 6 (4 wire half bridge) is used, the assumption is that the voltage drop in the connecting cable is negligible. The average resistance of 22 AWG wire is 16.5 ohms per 1000 feet. Thus, the resistance in the excitation lead going out to the load cell added to that in the lead coming back to ground is 33 ohms. The resistance of the bridge in the load cell is 350 ohms. The voltage drop across the load cell is equal to the voltage at the CR23X multiplied by the ratio of the load cell resistance, R_s, to the total resistance, R_T, of the circuit. If Instruction 6 were used to measure the load

7-9

cell, the excitation voltage actually applied to the load cell, V_1 would be:

$$V_1 = V_x \, R_S/R_T = V_x \, 350/(350+33) = 0.91 \, V_x$$

Where V_x is the excitation voltage. This means that the voltage output by the load cell would only be 91% of that expected. If recording of the lysimeter data was initiated with the load cell output at 0 volts, and 100mm of evapotranspiration had occurred, calculation of the change with Instruction 6 would indicate that only 91mm of water had been lost. Because the error is a fixed percentage of the output, the actual magnitude of the error increases with the force applied to the load cell. If the resistance of the wire was constant, one could correct for the voltage drop with a fixed multiplier. However, the resistance of copper changes 0.4% per degree C change in temperature. Assume that the cable between the load cell and the CR23X lays on the soil surface and undergoes a 25°C diurnal temperature fluctuation. If the resistance is 33 ohms at the maximum temperature, then at the minimum temperature, the resistance is:

$$(1-25 \times 0.004)33 \text{ ohms} = 29.7 \text{ ohms}$$

The actual excitation voltage at the load cell is:

$$V_1 = 350/(350+29.7) \, V_x = .92 \, V_x$$

The excitation voltage has increased by 1%, relative to the voltage applied at the CR23X. In this case, where we were recording a 91mm change in water content, there would be a 1mm diurnal change in the recorded water content that would actually be due to the change in temperature. Instruction 9 solves this problem by actually measuring the voltage drop across the load cell bridge. The drawbacks to using Instruction 9 are that it requires an extra differential channel and the added expense of a 6 wire cable. In this case, the benefits are worth the expense.

The load cell has a nominal full scale output of 3 millivolts per volt excitation. If the excitation is 3.3 volts, the full scale output is 9.9 millivolts; thus, the ±10 millivolt range is selected. The calibrated output of the load cell is 3.106mV/V1 at a load of 250 pounds. Output is desired in millimeters of water with respect to a fixed point. The calibration in mV/V1/mm is:

$$3.106\text{mV/V}_1/250\text{lb} \times 2.2\text{lb/kg} \times 3.1416\text{kg/mm}/4 =$$

$$0.02147\text{mV/V}_1/\text{mm}$$

The reciprocal of this gives the multiplier to convert mV/V1 into millimeters. (The result of Instruction 9 is the ratio of the output voltage to the actual excitation voltage multiplied by 1000, which is mV/V1):

$$1/0.02147\text{mV/V}_1/\text{mm} = 46.583\text{mm/mV/V}_1$$

The output from the load cell is connected so that the voltage increases as the mass of the lysimeter increases. (If the actual mechanical linkage was as diagrammed in Figure 7.12-1, the output voltage would be positive when the load cell was under tension.)

When the experiment is started, the water content of the soil in the lysimeter is approximately 25% on a volume basis. It is decided to use this as the reference, (i.e., 0.25 x 1500mm = 375 mm). The experiment is started at the beginning of what is expected to be a period during which evapotranspiration exceeds precipitation. Instruction 9 is programmed with the correct multiplier and no offset. After hooking everything up, the counterbalance is adjusted so that the load cell is near the top of its range; this will allow a longer period before readjustment is necessary. The result of Instruction 9 (monitored with the ⁎ 6 Mode) is 109. The offset needed to give the desired initial value of 375mm is 266. However, it is decided to add this offset in a separate instruction so that the result of Instruction 9 can be used as a ready reminder of the strain on the load cell (range = ±140mm). When the strain on the load cell nears its rated limits, the counterbalance is readjusted and the offset recalculated to provide a continuous record of the water budget.

The program table has an execution interval of 10 seconds. The average value in millimeters is output to Final Storage (not shown in Table) every hour. The average is used, instead of a sample, in order to cancel out the effects of wind loading on the lysimeter.

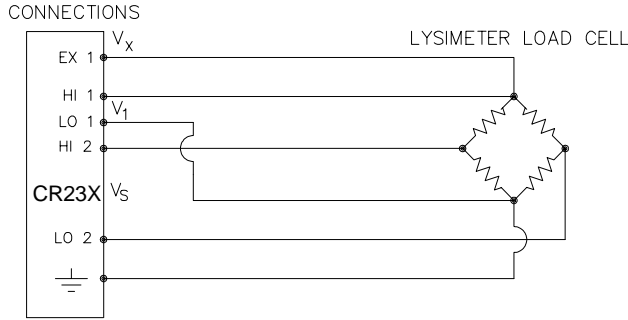**FIGURE 7.12-2. 6 Wire Full Bridge Connection for Load Cell**

PROGRAM

1: Full Bridge w/mv Excit (P9)
```
    1:      1          Reps
    2:     25          5000 mV, 60 Hz Reject,
                       Fast, Ex Range
    3:     21          10 mV, 60 Hz Reject, Slow,
                       Br Range
    4:      1          DIFF Channel
    5:      1          Excite all reps w/Exchan 1
    6:   3300          mV Excitation
    7:      1          Loc [ RAW_MEAS  ]
    8:  46.583         Mult
    9:    0.0          Offset
```

2: Z=X+F (P34)
```
    1:      1          X Loc [ RAW_MEAS  ]
    2:    266          F
    3:      2          Z Loc [ MEAS_OFFS ]
```

7.13 227 GYPSUM SOIL MOISTURE BLOCK

Soil moisture is measured with a gypsum block by relating the change in moisture to the change in resistance of the block. An AC Half Bridge (Instruction 5) is used to determine the resistance of the gypsum block. Rapid reversal of the excitation voltage inhibits polarization of the sensor. Polarization creates an error in the output so the fast integration time is used. The output of Instruction 5 is the ratio of the output voltage to the excitation voltage; this output is converted to gypsum block resistance with Instruction 59, Bridge Transform.

The Campbell Scientific 227 Soil Moisture Block uses a Delmhorst gypsum block with a 1 kohm bridge completion resistor (there are also series capacitors to block DC current and degradation due to electrolysis. Using data supplied by Delmhorst, Campbell Scientific has computed

coefficients for a 5th order polynomial to convert block resistance to water potential in bars. There are two polynomials: one to optimize the range from -0.1 to -2 bars, and one to cover the range from -0.1 to -10 bars (the minus sign is omitted in the output). The -0.1 to -2 bar polynomial requires a multiplier of 1 in the Bridge Transform Instruction (result in kohms) and the -0.1 to -10 bar polynomial requires a multiplier of 0.1 (result in 10,000s of ohms). The multiplier is a scaling factor to maintain the maximum number of significant digits in the coefficients of the polynomial.

In this example, we wish to make measurements on 6 gypsum blocks and output the final data in bars. The soil where the moisture measurements are to be made is quite wet at the time the data logging is initiated, but is expected to dry beyond the -2 bar limit of the wet range polynomial. The dry range polynomial is used, so a multiplier of 0.1 is entered in the bridge transform instruction.

When the water potential is computed, it is written over the resistance value. The potentials are stored in input locations 1-6 where they may be accessed for output to Final Storage. If it was desired to retain the resistance values, the potential measurements could be stored in Locations 7-12 by changing Parameter 3 in Instruction 55 to 7.
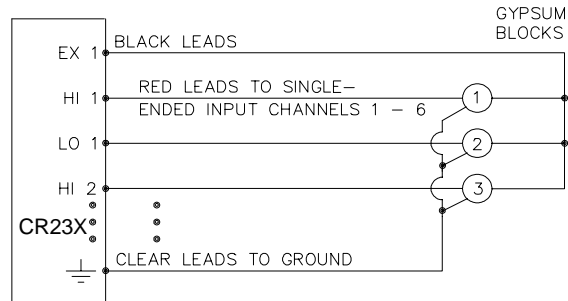
FIGURE 7.13-1. 6 Gypsum Blocks Connected to the CR23X

PROGRAM

1: AC Half Bridge (P5)
```
    1:      6          Reps
    2:     14          1000 mV, Fast Range
    3:      1          SE Channel
    4:      1          Excite all reps w/Exchan 1
    5:   1000          mV Excitation
    6:      1          Loc [ Vs_Vx_1   ]
    7:    1.0          Mult
    8:    0.0          Offset
```

2: BR Transform Rf[X/(1-X)] (P59)
```
  1:      6         Reps
  2:      1         Loc [ Vs_Vx_1   ]
  3:       .1       Multiplier (Rf)
```

3: Polynomial (P55)
```
  1:      6         Reps
  2:      1         X Loc [ Vs_Vx_1   ]
  3:      1         F(X) Loc [ Vs_Vx_1   ]
  4:       .15836   C0
  5:      6.1445    C1
  6:     -8.4189    C2
  7:      9.2493    C3
  8:     -3.1685    C4
  9:       .33392   C5
```

7.14 NONLINEAR THERMISTOR IN HALF BRIDGE (CAMPBELL SCIENTIFIC MODEL 101)

Instruction 11, 107 Thermistor Probe, automatically calculates temperature by transforming the millivolt reading with a 5th order polynomial. Instruction 55, Polynomial, can be used to calculate temperature of any nonlinear thermistor, provided the correlation between temperature and probe output is known, and an appropriate polynomial fit has been determined. In this example, the CR23X is used to measure the temperature of 5 Campbell Scientific 101 Probes (used with the CR21). Instruction 4, Excite, Delay, and Measure, is used because the high source resistance of the probe requires a long input settling time (see Section 13.3.1). The signal voltage is then transformed to temperature using the Polynomial Instruction.

The manual for the 101 Probe gives the coefficients of the 5th order polynomial used to convert the output in millivolts to temperature (E denotes the power of 10 by which the mantissa is multiplied):

```
        C0        -53.7842
        C1          0.147974
        C2         -2.18755E-4
        C3          2.19046E-7
        C4         -1.11341E-10
        C5          2.33651E-14
```

The CR23X will only allow 5 significant digits to the right or left of the decimal point to be entered from the keyboard. The polynomial cannot be applied exactly as given in the 101 manual. The initial millivolt reading must be scaled if the

coefficients of the higher order terms are to be entered with the maximum number of significant digits. If 0.001 is used as a multiplier on the millivolt output, the coefficients are divided by 0.001 raised to the appropriate power (i.e., C0=C0, C1=C1/0.001, C2=C2/.000001, etc.). With this adjustment, the coefficients entered in Parameters 4-9 of Instruction 55 become:

```
        C0            -53.784
        C1            147.97
        C2           -218.76
        C3            219.05
        C4           -111.34
        C5             23.365
```


FIGURE 7.14-1. 101 Thermistor Probes Connected to CR23X

PROGRAM

1: Excite-Delay (SE) (P4)
```
  1:      5         Reps
  2:     24         1000 mV, 60 Hz Reject,
                    Slow Range
  3:      1         SE Channel
  4:      1         Excite all reps w/Exchan 1
  5:     10         Delay (units 0.01 sec)
  6:    400         mV Excitation
  7:      1         Loc [ V_1      ]
  8:      1.0       Mult
  9:      0.0       Offset
```

2: Polynomial (P55)
```
  1:      1         Reps
  2:      1         X Loc [ V_1      ]
  3:      1         F(X) Loc [ V_1      ]
  4:    -53.784     C0
  5:    147.97      C1
  6:   -218.76      C2
  7:    219.05      C3
  8:   -111.34      C4
  9:     23.365     C5
```

7.15 WATER LEVEL - GEOKON'S VIBRATING WIRE PRESSURE SENSOR

The vibrating wire sensor utilizes a change in the frequency of a vibrating wire to sense pressure. Figure 7.15-1 illustrates how an increase in pressure on the diaphragm decreases the tension on the wire attached to the diaphragm. A decrease in the wire tension decreases the resonant frequency in the same way that loosening a guitar string decreases its frequency.

Vibrating Wire Measurement Instruction 28 excites the "plucking" and "pickup" coils shown in Figure 7.15-1 with a "swept" frequency. A "swept" frequency is a group of different frequencies that are sent one right after another starting with the lowest frequency and ending with the highest. The lowest and highest frequencies are entered by the user in units of hundreds of Hz. This swept frequency causes the wire to vibrate at each of the individual frequencies. Ideally, all of the frequencies except the one matching the resonant frequency of the wire will die out in a very short time. The wire will vibrate with the resonant frequency for a relatively long period of time, cutting the lines of flux in the "plucking" and "pickup" coils and inducing the same frequency on the lines to the CR23X. Instruction 28 then accurately measures how much time it takes to receive a user specified number of cycles.

The vibrating wire requires temperature compensation. A nonlinear thermistor built into the probe is measured using Instruction 4, a single-ended half bridge measurement with excitation, and calculated with Instruction 55, a fifth order polynomial instruction.

Campbell Scientific's AVW1 or AVW4 Vibrating Wire Sensor Interface provides transformer isolation between the datalogger and vibrating wire sensors.

The transformer strips off any DC noise on the signal, improving the ability to detect cycles.
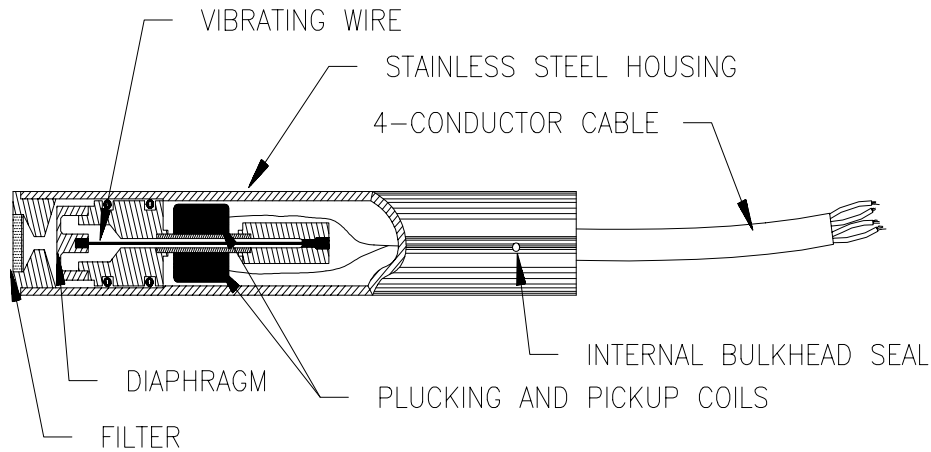
FIGURE 7.15-1. A Vibrating Wire Sensor

The following calculations are based on using a Geokon model 4500 Vibrating Wire sensor. An individual multiplier and offset must be calculated for each sensor used in a system.

MULTIPLIER

The fundamental equation relating frequency to pressure is

$$P = -F_x G + B \quad \text{where}$$

P = pressure, PSI
G = the Gage Factor obtained from the sensors calibration sheet in PSI/digit. The units of a digit are $Hz^2(10^{-3})$.
B = offset
$F_x = f^2 Hz^2(10^{-3})$, where f is frequency.

Instruction 28 measures period, T, of the vibrating wire in milliseconds (ms) and returns a measured value, X, of

$$X = 1/(T^2 (ms)^2) = f^2(10^{-6})Hz^2$$

A multiplier of -1000 in Instruction 28 converts the measurement to digits, as shown below.

$$-F_x = -X(-10^3) = -f^2(10^{-3})Hz^2$$

To calculate the multiplier, convert Geokon's gage factor, G, to the desired units (i.e., feet of water per digit) and multiply by -1000 digits/kHz^2.

TEMPERATURE CORRECTION

The temperature correction is applied as follows.

$$P_T = P + C * (t_1 - t_0), \quad \text{where}$$

P_T = Pressure corrected for temperature, °C
C = Temperature coefficient, PSI/°C (from Geokon calibration sheet)
t_0 & t_1 = Initial and current temperatures, °C.

The temperature coefficient, C, must be converted to units compatible with the gage factor, G.

WELL MONITORING EXAMPLE

In this example the vibrating wire sensor is used to monitor water table height (Figure 7.15-2). The desired data is the distance from the lip of

the well to the water surface. The sensor is vented to atmosphere to eliminate measurement errors due to changes in barometric pressure. The water level is expected to stay within 40 to 80 feet of the lip so the 50 psi pressure sensor is placed approximately 100 feet below the lip of the well. The calibration data from Geokon is provided in Table 7.15-1.

TABLE 7.15-1 Calibration Data for Sensor 3998	
Gage Factor (psi/digit)	Temp. Coeff. (psi/°C)
0.0151	-0.0698

The multiplier, m, is calculated to convert the reading to feet of water.

m = 0.0151 (psi/digit) * 2.3067 (ft of water/psi) * -1000 digits/kHz^2 = -34.831 ft of water/kHz^2

After the probe reaches thermal equilibrium, the initial temperature, t_0, is measured to be 24°C.

The water column above the sensor is referred to as the "Reading". The Reading decreases with increasing "Distance" from lip of well to water surface so the Distance is computed by subtracting the Reading from the Offset as shown in Figure 7.15-2.

The "Initial Distance" to the water surface is measured with a chalked line to be 47.23 feet below the lip. The first time the program is executed, the program calculates the offset (Offset = Distance + Reading) required to obtain a reading of 47.23 feet. The offset is stored in Location 4 and applied to subsequent measurements.

NOTE: Following program compilation in the [*] [0] Mode, all input locations are set to zero. This fact is utilized to detect the first execution following a program compilation.

The example assumes the sensor has been connected as shown here.
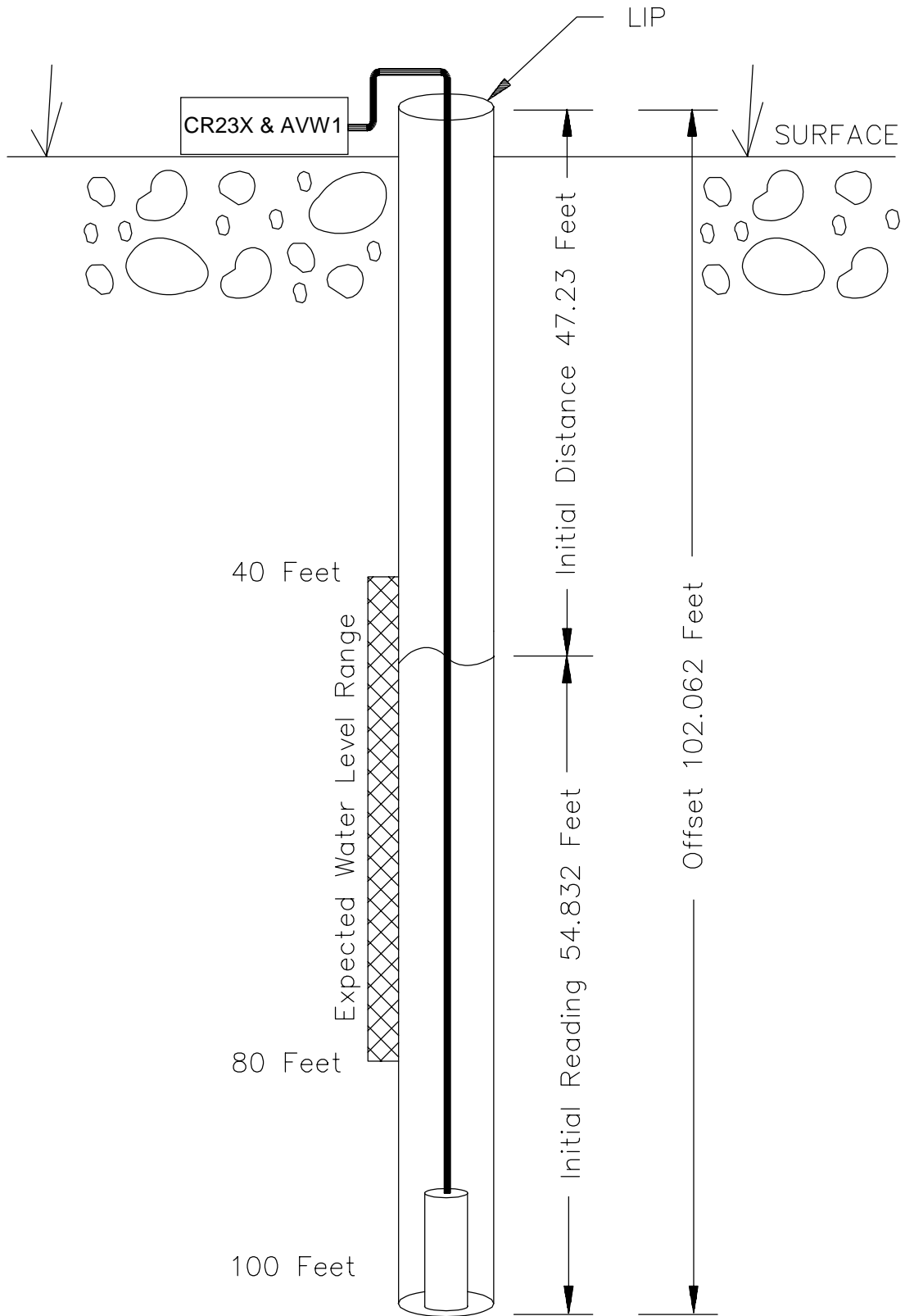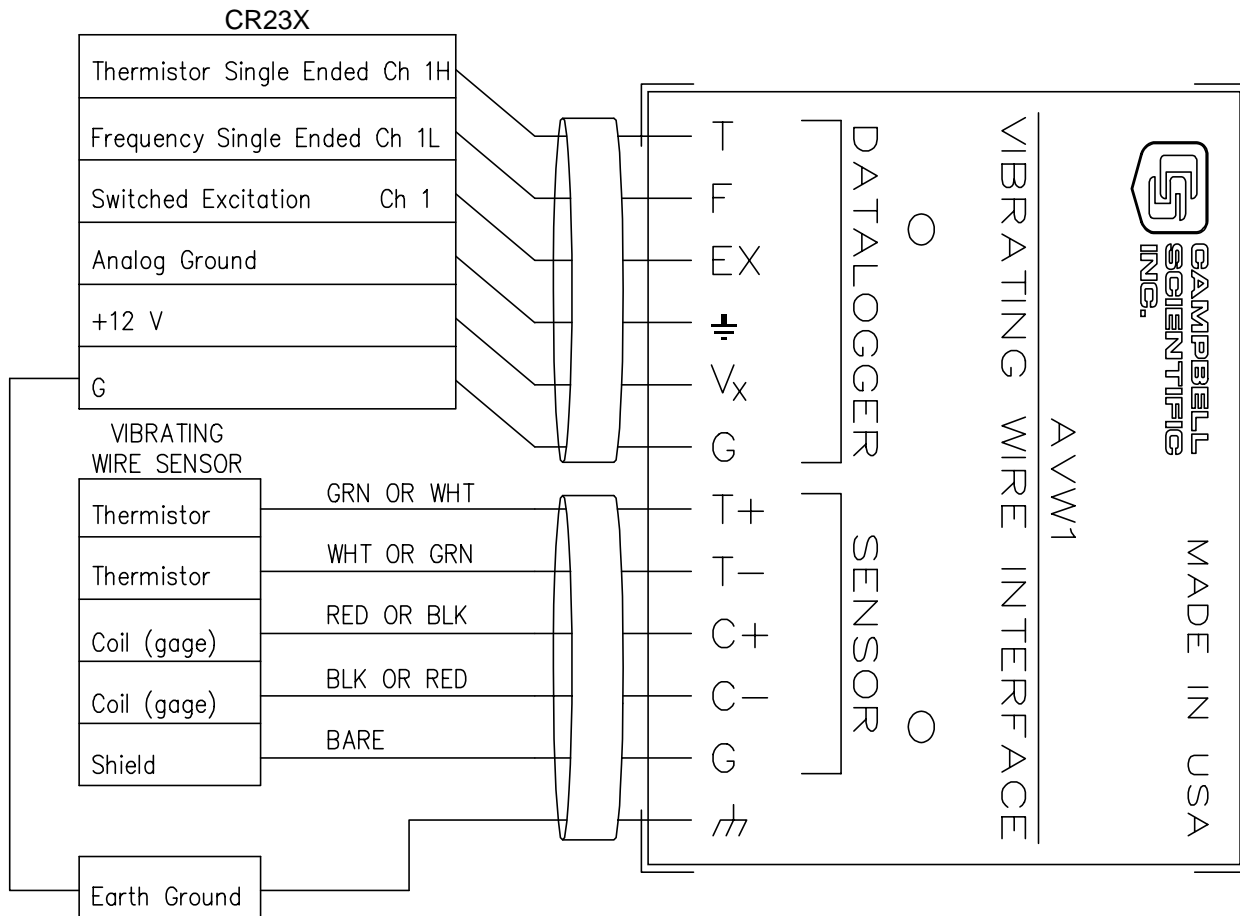
FIGURE 7.15-2. Well Monitoring Example

FIGURE 7.15-3. Hook up to AVW1

PROGRAM

AVW1 & CR23X USED TO MEASURE 1
GEOKON VIBRATING WIRE SENSOR.

* Table 1 Program
  01:      60   Execution Interval (seconds)

01:     Excite-Delay (SE) (P4)
        1:     1   Reps
        2:    15   ±5000 mV Fast Range
        3:     1   SE Channel
        4:     1   Excite all reps w/Exchan 1
        5:     1   Delay (units 0.01 sec)
        6: 2500   mV Excitation
        7:     1   Loc [Temp   ]
        8:    .001   Mult
        9:     0   Offset

02:     Polynomial (P55)
        1:     1   Reps
        2:     1   X Loc [Temp   ]
        3:     1   F(X) Loc [Temp   ]
        4: -104.78   C0
        5: 378.11   C1
        6: -611.59   C2
        7: 544.27   C3
        8: -240.91   C4
        9:   43.089   C5

03:     Vibrating Wire (SE) (P28)
        1:     1   Reps
        2:     2   SE Channel
        3:     1   Excite all reps w/Exchan 1
        4:    24   Starting Freq. (units = 100 Hz)
        5:    32   End Freq. (units = 100 Hz)
        6:   500   No. of Cycles
        7:     0   Rep Delay (units = 0.01 sec)
        8:     2   Loc [Pressure]
        9: -34.836   Mult
        10:    0   Offset

```
04:   Z=X+F (P34)
      1:    1        X Loc [ Temp      ]
      2:  -24         F
      3:    3        Z Loc [ Temp_Comp ]

05:   Z=X*F (P37)
      1:    3        X Loc [ Temp_Comp ]
      2:  -.0698      F
      3:    3        Z Loc [ Temp_Comp ]

06:   Z=X+Y (P33)
      1:    3        X Loc [ Temp_Comp ]
      2:    2        Y Loc [ Pressure  ]
      3:    2        Z Loc [ Pressure  ]

07:   IF (X<=>F) (P89)
      1:    5        X Loc [ Cmpile_Ck ]
      2:    1         =
      3:    0         F
      4:   30        Then Do

08:   Z=X+F (P34)
      1:    2        X Loc [ Pressure  ]
      2:  47.23       F
      3:    4        Z Loc [ Offset    ]

09:   Z=F (P30)
      1:    1         F
      2:    0        Exponent of 10
      3:    5        Z Loc [ Cmpile_Ck ]

10:   End (P95)

11:   Z=X-Y (P35)
      1:    4        X Loc [ Offset    ]
      2:    2        Y Loc [ Pressure  ]
      3:    6        Z Loc [ Distance  ]
```

7.16 PAROSCIENTIFIC "T" SERIES PRESSURE TRANSDUCER

The Paroscientific "T" series transducer has two resonating quartz crystals that output frequency signals for temperature and pressure. The pressure output requires temperature compensation. Instruction 27 measures the outputs and returns period in microseconds (refer to Table 7.16-1).

TABLE 7.16-1 Period Averaging Inst. 27

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Gain/output option
03:	2	Single Ended Input Channel
04:	4	# Cycles to measure
05:	4	Time out (0.01 sec, at least the maximum duration of the # of cycles specified + 1 1/2 cycles.)
06:	4	Destination input location
07:	FP	Multiplier
08:	FP	Offset

GAIN/OUTPUT OPTION, PARAMETER 2

The nominal pressure frequency range is 32 to 42 kHz over the stated pressure range (period of $3.1*10^{-5}$ to $2.4*10^{-5}$). The nominal temperature frequency is 172 kHz, ± 50 ppm/$^{\circ}$C (period of $5.8*10^{-6}$). The gain/output option (Parameter 2) is based on the maximum input frequency. For the stated temperature and pressure frequency ranges, the gain codes are 3 and 4 (refer to Table 7.16-2). Codes 1 and 2 are not used with Paroscientific transducers. For period, the output option is 0.

TABLE 7.16-2 Input Frequency Gain Codes

Range Code	Peak to Peak Volts Required @ Max. Freq.*		Maximum Frequency	
1	2	mV	10	kHz
2	5	mV	20	kHz
3	20	mV	30	kHz
4	500	V	100	kHz

0x   Output period in microseconds
1x   Output frequency in kHz
     where x is range code

*   AC voltage; must be centered around CR23X ground.

TIME OUT, PARAMETER 5

The "time out", Parameter 5, specifies the maximum length of time the instruction waits on each repetition to receive the number of cycles specified in Parameter 4. The time out units are 0.01 seconds. The minimum time out is the time required to receive the specified number of cycles at the maximum expected frequency. Time out calculations using a recommended 9000 and 5000 cycles for temperature and pressure at the maximum frequency are shown below.

Time out for temperature: 6,
$5.22 = (5.8*10^{-6})(9000/0.01)$

Time out for pressure: 16,
$15.5 = (3.1*10^{-5})(5000/0.01)$

If the time out expires before the requested number of cycles are read, -99999 is stored in the input location (Parameter 6). After reading the correct number of cycles, program execution advances to the next instruction; the time out may be over estimated without losing execution speed.

CONVERTING MICROSECONDS TO °C AND PSI

Instruction 64 (Paroscientific Processing) reports the temperature in °C and the pressure in PSI using the measurements made by Instruction 27 (period average).

The pressure (P) and temperature (T) in microseconds are converted to psi and °C according to the following relationships provided by Paroscientific.

$$P = C (1-T_0^2/Tau^2)(1-D(1-T_0^2/Tau^2))$$
$$T = Y_1U + Y_2U^2 + Y_3U^3$$

where:

$C = C_1 + C_2U + C_3U^2$

$D = D_1 + D_2U$

$T_0 = T_1 + T_2U + T_3U^2 + T_4U^3 + T_5U^4$

$U = U_t - U_0$

Tau = measured period, in microseconds, of pressure

U_t = measured period, in microseconds, of temperature

Values for the calibration coefficients (U_0, Y_1, Y_2, Y_3, C_1, C_2, C_3, D_1, D_2, T_1, T_2, T_3, T_4, T_5) are provided by Paroscientific.

ENTERING THE COEFFICIENTS

Coefficients are entered using CR23X Instruction 65 (Bulk Load). A calibration sheet from Paroscientific which accompanies the transducer lists the 14 coefficients. Table 7.16-3 shows the correct format for entering the coefficients. The coefficients shown are for transducer Serial Number 30135. Your coefficients will be different.

TABLE 7.16-3 Coefficient Entry Format for Paroscientific "T" Series Pressure Transducer Instruction 30

Coeff.	Value	Entry
U_0	5.860253	5.8603
Y_1	-3970.348	-3970.3
Y_2	-7114.265	-7114.3
* Y_3	102779.1	102.78
C_1	70.29398	70.294
C_2	6.610141	6.6101
C_3	-119.2867	-119.29
* D_1	0.0308837	30.884
D_2	0.0	0.0
T_1	26.33703	26.337
T_2	0.8516985	0.85170
T_3	21.80118	21.801
T_4	0.0	0.0
T_5	0.0	0.0

* Y_3 and D_1 coefficients are entered as $Y_3/1000$ and D_1*1000.

Only the first 5 digits of each 7 digit coefficient are entered in the datalogger. The maximum error that occurs due to using five digit coefficients is 0.001 psi throughout a 15 psi range at any temperature between -30 and +60°C. The maximum error in temperature calculations is 0.2°C over the same temperature range.

CONNECTIONS

Figure 7.16-1 details the components required for connecting the transducer to the CR23X. The user supplied components are commonly available at commercial electronic stores.
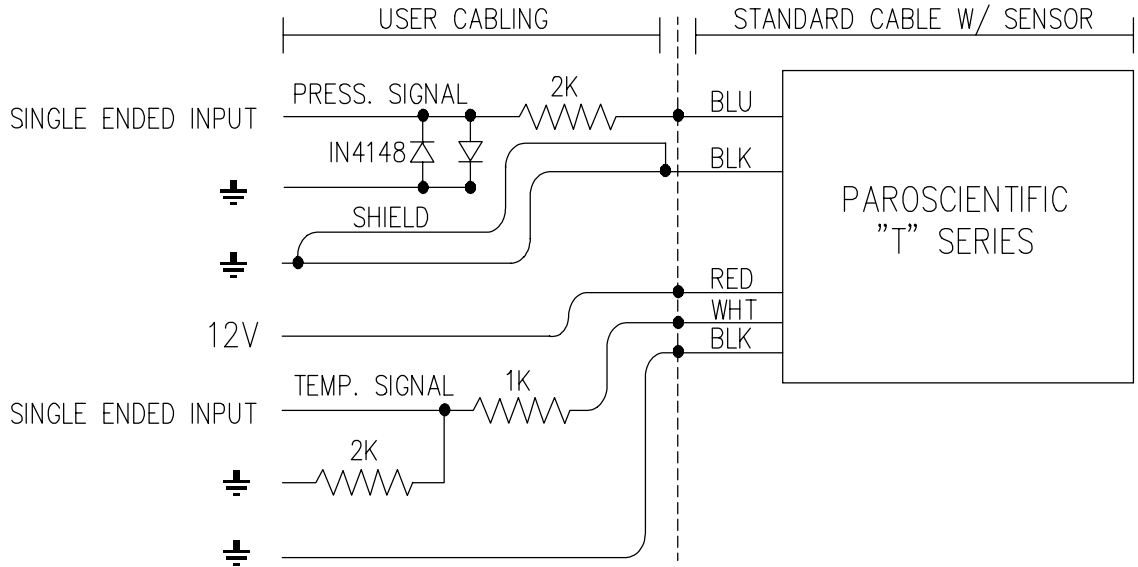
CR23X
CONNECTIONS

FIGURE 7.16-1. CR23X/Paroscientific "T" Series Transducer Wiring Diagram

PROGRAM EXAMPLE

The following example reads the coefficients from a subroutine only when the datalogger program is compiled. The coefficients are stored in Input Locations 3 through 16. The temperature frequency is read on single-ended Channel 1 and stored in Input Location 1. Pressure is measured on single-ended Channel 2 and stored in Location 2. Instruction 64 converts the readings to engineering units. Temperature (°C), pressure (psi), and signature are stored in Locations 19 through 21, respectively. The signature is based on the 14 coefficients. Record the signature once the coefficients are correctly entered in the program. The recorded signature may be used to identify any changes made to the coefficients at a later date. Instructions to output the readings to Final Storage are not included in this example.

PROGRAM

```
*  Table 1 Program
   01:      60        Execution Interval (seconds)
```

;Load the coefficients if the program has
;just compiled.
;

```
01:    IF (X<=>F) (P89)
       1:    17        X Loc [ Comp_Chk  ]
       2:    1         =
       3:    0         F
       4:    1         Call Subroutine 1
```

;Measure the temperature period in
;microseconds.
;

```
02:    Period Average (SE) (P27)
       1:    1         Reps
       2:    4         Input Gain = 1
       3:    1         SE Channel
       4:  9000        No. of Cycles
       5:    6         Timeout (units = 0.01 seconds)
       6:    1         Loc [ Temp_us   ]
       7:    1         Mult
       8:    1         Offset
```

;Measure the pressure period in
;microseconds.
;

```
03:    Period Average (SE) (P27)
       1:    1         Reps
       2:    4         Input Gain = 1
       3:    2         SE Channel
       4:  5000        No. of Cycles
       5:    15        Timeout (units = 0.01 seconds)
       6:    2         Loc [ Press_us  ]
       7:    1         Mult
       8:    0         Offset
```

;Find Temperature and Pressure.
;
```
04:    Paroscientific (P64)
       1:    1          Loc [ Temp_us  ]
       2:    19         Loc [ Temp_C   ]
```

* Table 3 Subroutines

```
01:    Beginning of Subroutine (P85)
       1:    1          Subroutine 1

02:    Bulk Load (P65)
       1:    5.8603    F
       2: -3970.3      F
       3: -7114.3      F
       4:    102.78    F
       5:    70.294    F
       6:    6.6101    F
       7:  -119.29     F
       8:    30.884    F
       9:    3          Loc [ U0     ]

03:    Bulk Load (P65)
       1:    0          F
       2:    26.337    F
       3:    .85170    F
       4:    21.801    F
       5:    0          F
       6:    0          F
       7:    1          F
       8:    0          F
       9:    11         Loc [ D2     ]

04:    End (P95)
```

End Program

INPUT LOCATION LABELS:

1 Temp_us	12 T1
2 Press_us	13 T2
3 U0	14 T3
4 Y1	15 T4
5 Y2	16 T5
6 Y3	17 Comp_Chk
7 C1	18 plce_hold
8 C2	19 Temp_C
9 C3	20 Press_psi
10 D1	21 Signature
11 D2	

7.17 4 TO 20 MA SENSOR USING CURS100 TERMINAL INPUT MODULE

A dew point sensor has a 4 to 20 mA output over the dew point temperature range of -40° to +70°C. The dew point sensor output may be measured by the CR23X using the CUS100 Terminal Input Module (TIM). The CUS100 uses a 100 Ω, $\pm$ 0.01 % resistor to convert the 4 to 20 mA range to 400 to 2000 mV. The millivolt range was found using the relationship $V = IR$, where V is voltage, I is current, and R is resistance, e.g. the voltage at -40°C is given by $V = 4\ mA * 100\ \Omega = 400\ mV$. The dew point sensor is measured with Instruction 2 (Volt Diff). The multiplier for dew point temperature is found with the following relationship [70°C - (-40°C)] / [2000 mV - 400 mV] = 0.06875°C/mV. The offset is found by taking the linear relationship °C = mV $*$ Mult + Offset and solving for the Offset. At -40°C the voltage is 400 mV, thus the Offset = -40 - [400 mV $*$ 0.06875°C/mV] = -67.5°C.

CONNECTIONS
The dew point sensor is measured with a differential voltage measurement on differential analog input 4. The CURS100 TIM and dew point sensor are wired to the CR23X wiring panel as shown in Figure 7.17-1.

PROGRAM

```
01:    Volt (Diff) (P2)
       1:    1          Reps
       2:    25         ±5000 mV Slow 60 Hz
                        Rejection Range
       3:    4          DIFF Channel
       4:    1          Loc [ Dew_Pnt_C ]
       5:    .06875     Mult
       6: -67.5         Offset
```

INPUT LOCATIONS

1 Dew_Pnt_C
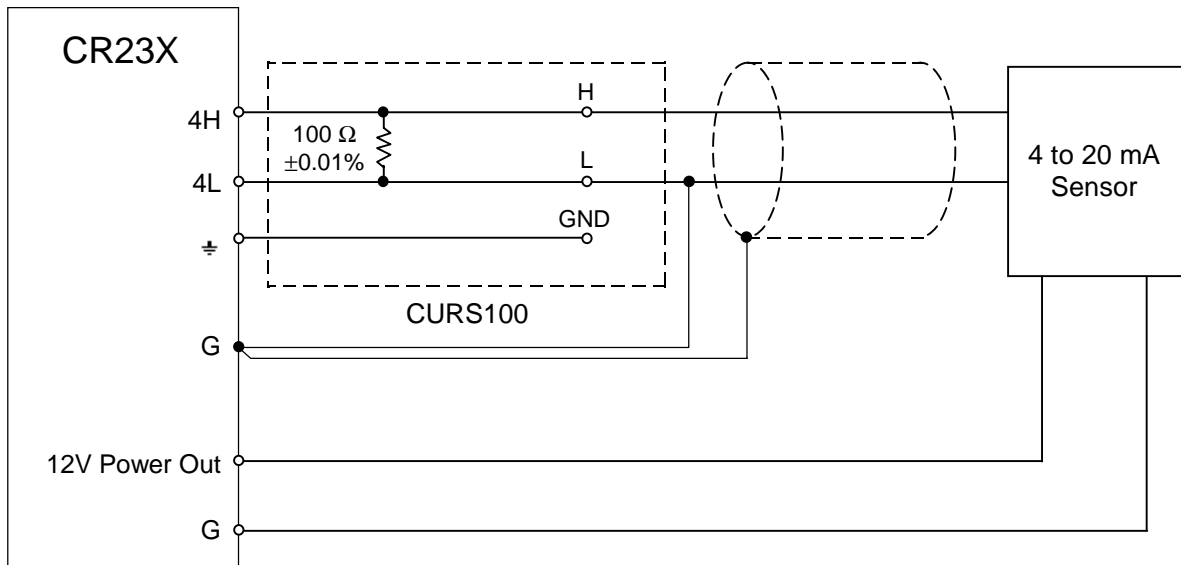
FIGURE 7.17-1 Wiring Diagram for CURS100 Terminal Input Module and 4 to 20 mA Sensor.

SECTION 8. PROCESSING AND PROGRAM CONTROL EXAMPLES

The following examples are intended to illustrate the use of Processing and Program Control Instructions, flags, dual Final Storage, and the capability to direct the results of Output Processing Instructions to Input Storage.

The specific examples may not be as important as some of the techniques employed, for example:

Directing Output Processing to Input Storage is used in the Running Average and Rainfall Intensity examples (8.1 and 8.2).

Flag tests are used in the Running Average, Interrupt Subroutine, Converting Wind Direction, and Saving Data Prior to Event examples (8.1, 8.5, 8.7 and 8.8).

Control ports and the Loop are illustrated in the AM32 example (8.3).

An algorithm for a down counter is used in the Saving Data Prior to Event example (8.8).

Powering 12 V sensors with the switched 12 V port (8.12).

As in Section 7 these examples are not complete programs to be taken verbatim. They need to be altered to fit specific needs.

8.1 COMPUTATION OF RUNNING MAXIMUM

It is sometimes necessary to compute a running maximum (i.e., the maximum covers a fixed number of samples and is continuously updated as new samples are taken). Because the output interval is shorter than the period over which the maximum is determined, Instruction 71 cannot be used; the algorithm for computing this maximum must be programmed by the user. The following example demonstrates a program for computing a running maximum.

In this example, each time a new measurement is made (in this case a thermocouple temperature) a maximum is determined and computed for the 10 most recent samples. This is done by saving all 10 temperatures in contiguous input locations and using the Spatial Maximum Instruction (49) to determine the maximum. The temperatures are stored in locations 11 through 20. Each time the table is executed, the new measurement is stored in location 20 and the maximum is stored in location 2. The Block Move Instruction (54) is then used to move the temperatures from locations 12 through 20 down by 1 location; the oldest measurement (in location 11) is lost when the temperature from location 12 is written over it.

PROGRAM

```
*       Table 1 Program
  01:     10.0        Execution Interval (seconds)

01:   Panel Temperature (P17)
      1:   1          Loc [ Panl_Temp ]

02:   Thermocouple Temp (DIFF) (P14)
      1:   1          Reps
      2:   1          ±10 mV Slow Range
      3:   1          DIFF Channel
      4:   1          Type T (Copper-Constantan)
      5:   1          Ref Temp Loc [ Panl_Temp ]
      6:  20          Loc [ Temp_i   ]
      7:   1          Mult
      8:   0          Offset

03:   Spatial Maximum (P49)
      1:  10          Swath
      2:  11          First Loc [ Temp_i_9  ]
      3:   2          Avg Loc [ 10smpl_av ]

04:   Block Move (P54)
      1:   9          No. of Values
      2:  12          First Source Loc [ Temp_i_8  ]
      3:   1          Source Step
      4:  11          First Destination Loc [ Temp_i_9  ]
      5:   1          Destination Step

05:   Do (P86)
      1:  10          Set Output Flag High
```

06: Sample (P70)
     1: 1   Reps
     2: 2   Loc [10smpl_mx]

INPUT LOCATIONS

1 Panl_Temp   15 Temp_i_5
2 10smpl_mx   16 Temp_i_4
11 Temp_i_#1   17 Temp_i_3
12 Temp_i_8   18 Temp_i_2
13 Temp_i_7   19 Temp_i_1
14 Temp_i_6   20 Temp_i

In the above example, all samples for the maximum are stored in input locations. This is necessary when an maximum must be output with each new sample. In most cases, maximums are desired less frequently than sampling. For example, it may be necessary to sample some parameter every 5 seconds and output every hour a maximum of the previous three hours' readings. If all samples were saved, this would require 2160 input locations. The same value can be obtained by determining an hourly maximum and determining the maximum of the hourly maximums for the past three hours. To do this requires that hourly maximums be stored in input locations.

Instruction 80 is used to send the 1 hour maximum to Input Storage and again to send the 3 hour maximum to Final Storage.

PROGRAM

*   Table 1 Program
  01:      5.0   Execution Interval (seconds)

01: Volt (Diff) (P2)
     1: 1   Reps
     2: 25   ±5000 mV Slow 60 Hz Rejection Range
     3: 3   DIFF Channel
     4: 5   Loc [XX_mg_M3]
     5: 10   Mult
     6: 0   Offset

02: If time is (P92)
     1: 0   Minutes (Seconds --) into a
     2: 60   Interval (same units as above)
     3: 10   Set Output Flag High

03: Set Active Storage Area (P80)
     1: 3   Input Storage Area
     2: 3   Array ID or Loc [max_i   ]

04: Maximum (P73)
     1: 1   Reps
     2: 5   Loc [XX_mg_M3]

05: Spatial Maximum (P49)
     1: 3   Swath
     2: 1   First Loc [max_i_2]
     3: 4   Avg Loc [3_Hr_max]

06: Set Active Storage Area (P80)
     1: 1   Final Storage Area 1
     2: 25   Array ID or Loc [_____]

07: Real Time (P77)
     1: 0110   Day,Hour/Minute

08: Sample (P70)
     1: 1   Reps
     2: 4   Loc [3_Hr_max]

09: If Flag/Port (P91)
     1: 10   Do if Output Flag is High (Flag 0)
     2: 30   Then Do

10: Block Move (P54)
     1: 2   No. of Values
     2: 2   First Source Loc [max_i_1]
     3: 1   Source Step
     4: 1   First Destination Loc [max_i_2]
     5: 1   Destination Step

11: End (P95)

INPUT LOCATIONS

1 max_i_2
2 max _i_1
3 max _i
4 3_Hr_ max
5 XX_mg_M3

8.2 RAINFALL INTENSITY

In this example, the total rain for the last 15 minutes is output only if any rain has occurred. The program makes use of the capability to direct the output of Output Processing Instructions to Input Storage.

Every 15 minutes, the total rain is sent to Input Storage. If the total is not equal to 0, output is redirected to Final Storage Area 1, the time is output and the total is sampled.

PROGRAM

```
*       Table 1 Program
  01:        60.0        Execution Interval (seconds)

01:   Pulse (P3)
       1:     1          Reps
       2:     1          Pulse Input Channel
       3:     2          Switch Closure
       4:     1          Loc [ Precip_mm ]
       5:      .254      Mult
       6:     0          Offset

02:   If time is (P92)
       1:     0          Minutes (Seconds --) into a
       2:     15         Interval (same units as above)
       3:     10         Set Output Flag High

03:   Set Active Storage Area (P80)
       1:     3          Input Storage Area
       2:     2          Array ID or Loc [ 15min_tot ]

04:   Totalize (P72)
       1:     1          Reps
       2:     1          Loc [ Precip_mm ]

05:   IF (X<=>F) (P89)
       1:     2          X Loc [ 15min_tot ]
       2:     2          <>
       3:     0          F
       4:     30         Then Do

06:   Set Active Storage Area (P80)
       1:     1          Final Storage Area 1
       2:     25         Array ID or Loc
                         [ _____ ]

07:   Real Time (P77)
       1: 0110           Day,Hour/Minute

08:   Sample (P70)
       1:     1          Reps
       2:     2          Loc [ 15min_tot ]

09:   End (P95)
```

INPUT LOCATIONS

1 Precip_mm
2 15min_tot

8.3 USING CONTROL PORTS AND LOOP TO RUN AM416 MULTIPLEXER

This example uses an AM416 to measure 16 copper-constantan thermocouples and 16 Model 223 soil moisture blocks. The sensors are read every ten minutes and the average value output once an hour. The multiplexer is housed in an AM-ENCT enclosure to minimize thermocouple errors created by thermal gradients. A 107 Temperature Probe is centrally located on the multiplexer board and used as a thermocouple temperature reference.

The AM416 switches the 223 moisture block out of the circuit when it is not being measured. This eliminates the need for the blocking capacitors used in the model 227 soil moisture block. The 223 blocks are about one fifth the cost of the 227 blocks.

Control ports are used to reset the AM416 and clock it through its channels. The sequence of the program is:

- Measure the 107 probe located at the AM416 for TC temperature reference.
- CR23X sets the port high which resets the AM416.
- A loop is entered; within each pass:
- The port clocking the AM416 is pulsed.
- Delay to ensure relay is closed.
- The connected TCs and moisture blocks are measured.
- CR23X sets the port controlling AM416 reset low.
- Soil moisture measurements are converted to block resistances.

The input location in which the temperature and soil moisture measurements are stored is indexed to the loop counter (Instruction 87, Section 12). An indexed location is incremented by one with each pass through the loop. For example, on the first pass temperature is stored in Location 2, and soil moisture in Location 18. On the second pass temperature is stored in Location 3, and soil moisture in Location 19. After 16 loop passes, temperature and soil moisture measurements occupy Locations 2 through 17 and 18 through 33, respectively.

Connections are shown in Figure 8.3-1.

CR23X   AM416
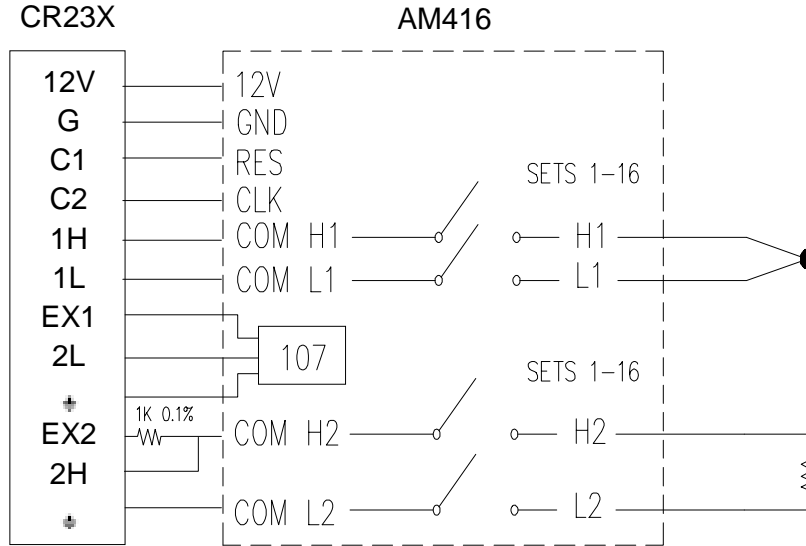
FIGURE 8.3-1. AM416 Wiring Diagram For Thermocouple and Soil Moisture Block Measurements

PROGRAM

```
*       Table 1 Program
 01:     600.0      Execution Interval (seconds)

01:   Temp (107) (P11)
      1:    1        Reps
      2:    4        SE Channel
      3:    1        Excite all reps w/Exchan 1
      4:    1        Loc [ Ref_Temp ]
      5:    1        Mult
      6:    0        Offset

02:   Do (P86)
      1:    41       Set Port 1 High

03:   Beginning of Loop (P87)
      1:    0        Delay
      2:    16       Loop Count

04:   Do (P86)
      1:    72       Pulse Port 2

05:   Delay w/Opt Excite (P22)
      1:    1        Ex Channel
      2:    0        Delay W/Ex (units = 0.01
                     sec)
      3:    1        Delay After Ex (units =
                     0.01 sec)
      4:    0        mV Excitation

06:   Thermocouple Temp (DIFF) (P14)
      1:    1        Reps
      2:    21       ±10 mV Slow 60 Hz
                     Rejection Range
      3:    1        DIFF Channel
      4:    1        Type T (Copper-Constantan)
```

```
      5:    1        Ref Temp Loc
                     [ Ref_Temp ]
      6:    2--      Loc [ TC_#1    ]
      7:    1        Mult
      8:    0        Offset

07:   AC Half Bridge (P5)
      1:    1        Reps
      2:    13       ±200 mV Fast Range
      3:    3        SE Channel
      4:    1        Ex Channel Option
      5:    200      mV Excitation
      6:    18--     Loc [ Soil_#1   ]
      7:    1        Mult
      8:    0        Offset

08:   End (P95)

09:   Do (P86)
      1:    51       Set Port 1 Low

10:   BR Transform Rf[X/(1-X)] (P59)
      1:    16       Reps
      2:    18       Loc [ Soil_#1   ]
      3:    .1       Multiplier (Rf)

11:   If time is (P92)
      1:    0        Minutes (Seconds --) into a
      2:    60       Interval (same units as above)
      3:    10       Set Output Flag High

12:   Real Time (P77)
      1: 0110        Day,Hour/Minute

13:   Average (P71)
      1:    33       Reps
      2:    1        Loc [ Ref_Temp ]
```

INPUT LOCATIONS

1 Ref_Temp	12 TC_#11	23 Soil_#6
2 TC_#1	13 TC_#12	24 Soil_#7
3 TC_#2	14 TC_#13	25 Soil_#8
4 TC_#3	15 TC_#14	26 Soil_#9
5 TC_#4	16 TC_#15	27 Soil_#10
6 TC_#5	17 TC_#16	28 Soil_#11
7 TC_#6	18 Soil_#1	29 Soil_#12
8 TC_#7	19 Soil_#2	30 Soil_#13
9 TC_#8	20 Soil_#3	31 Soil_#14
10 TC_#9	21 Soil_#4	32 Soil_#15
11 TC_#10	22 Soil_#5	33 Soil_#16

8.4 SUB 1 MINUTE OUTPUT INTERVAL SYNCHED TO REAL TIME

Output can be synchronized to seconds by pressing "-" or "C" while entering the first parameter in Instruction 92. If a counter, incremented within the program, was used to determine when to set the Output Flag, output would depend on the number of times the table was executed. The actual time of output would depend on when the program was actually compiled and started running. If the table overran its execution interval (Section 1.1.1), the output interval would not be the count multiplied by the execution interval, but some longer interval.

In this example a temperature (107 Temperature Probe) is measured every 0.5 seconds and the average output every 30 seconds.

PROGRAM

```
*       Table 1 Program
  01:       0.5       Execution Interval (seconds)

01:     Reference Temperature (P17)
        1:   1          Loc [ Ref_Temp  ]

02:     Thermocouple Temp (DIFF) (P14)
        1:   1          Reps
        2:  11          ±10 mV Fast Range
        3:   2          DIFF Channel
        4:   2          Type E (Chromel-Constantan)
        5:   1          Ref Temp Loc [ Ref_Temp ]
        6:   2          Loc [ TC_Temp   ]
        7:   1          Mult
        8:   0          Offset
```

```
03:     If time is (P92)
        1:    0--        Minutes (Seconds --) into a
        2:   30          Interval (same units as above)
        3:   10          Set Output Flag High

04:     Average (P71)
        1:    1          Reps
        2:    2          Loc [ TC_Temp   ]
```

INPUT LOCATIONS

1 Ref_Temp
2 TC_Temp

8.5 SWITCH CLOSURES ON CONTROL PORTS (RAIN GAGE)

Control ports 6, 7, and 8 can be used to measure switch closures up to 40 Hz. Instruction 3, pulse, is used to measure two rain gages on pulse inputs 1 and 2, and a third rain gage with control port 8. This is done as a comparison. In a real application the pulse channels would be used for wind speed and a control port for a rain gage. The rain gage is connected as diagrammed below.
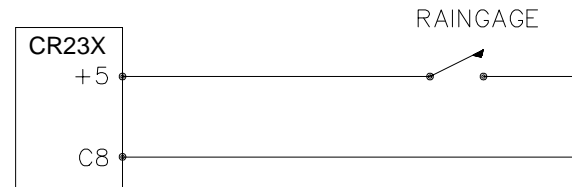
FIGURE 8.5-1. Connections for Rain Gage

PROGRAM

```
*       Table 1 Program
  01:      10.0       Execution Interval (seconds)

01:     Pulse (P3)
        1:    2          Reps
        2:    1          Pulse Input Channel
        3:    2          Switch Closure
        4:   10          Loc [ Precip_1  ]
        5:    .254       Mult
        6:    0          Offset

02:     Pulse (P3)
        1:    1          Reps
        2:    8          Control Port
        3:    2          Switch Closure
        4:   12          Loc [ Precip_3  ]
        5:    .254       Mult
        6:    0          Offset
```

03:     If time is (P92)
        1:      0   Minutes (Seconds --) into a
        2:      60   Interval (same units as above)
        3:      10   Set Output Flag High

04:     Real Time (P77)
        1: 0110   Day,Hour/Minute

05:     Totalize (P72)
        1:      3   Reps
        2:      10   Loc [Precip_1]

INPUT LOCATIONS

10 Precip_1
11 Precip_2
12 Precip_3

8.6 SDM-A04 ANALOG OUTPUT MULTIPLEXER TO STRIP CHART

This example illustrates the use of the SDM-A04 4 Channel Analog Output Multiplexer to output 4 analog voltages to a strip chart.

While of questionable value because of current requirements and strip chart reliability, some archaic regulations require strip chart backup on weather data. The SDM-A04 may be used with the CR23X to provide an additional four analog outputs to strip charts. The output values in this example are wind speed, wind direction, air temperature, and solar radiation.

Instruction 103 is used to activate the SDM-A04. The (4) millivolt values to output must be stored in adjacent Input Storage locations, the first of which is referenced in Instruction 103.

The following program measures the sensors every 5 seconds. The readings are moved to another 4 locations and scaled to a 0 to 1000 millivolt output for the SDM-A04. Wind direction is changed from a 0-360 degree input to output representing 0 to 540 degrees. This conversion is done in a subroutine which is described in the next example.

The example also includes instructions to output wind vector and average temperature and solar radiation every hour.

PROGRAM

*   Table 1 Program
  01:         5   Execution Interval (seconds)

01:     Pulse (P3)
        1:      1   Reps
        2:      1   Pulse Input Channel
        3:      22   Switch Closure, Output Hz
        4:      1   Loc [WS   ]
        5:      1.789   Mult
        6:      1   Offset

02:     Excite-Delay (SE) (P4)
        1:      1   Reps
        2:      14   ±1000 mV Fast Range
        3:      1   SE Channel
        4:      1   Excite all reps w/Exchan 1
        5:      5   Delay (units 0.01 sec)
        6: 1000   mV Excitation
        7:      2   Loc [0_360_WD]
        8:      .7273   Mult
        9:      0   Offset

03:     Temp (107) (P11)
        1:      1   Reps
        2:      2   SE Channel
        3:      2   Excite all reps w/Exchan 2
        4:      3   Loc [Ta   ]
        5:      1   Mult
        6:      0   Offset

04:     Volts (SE) (P1)
        1:      1   Reps
        2:      1   ±10 mV Slow Range
        3:      3   SE Channel
        4:      4   Loc [Rs   ]
        5:      .14493 Mult
        6:      0   Offset

05:     If time is (P92)
        1:      0   Minutes (Seconds --) into a
        2:      60   Interval (same units as above)
        3:      10   Set Output Flag High

06:     Block Move (P54)
        1:      4   No. of Values
        2:      1   First Source Loc [WS   ]
        3:      1   Source Step
        4:      5   First Destination Loc [WS_out   ]
        5:      1   Destination Step

07:     Do (P86)
        1:      1   Call Subroutine 1

08:     Scaling Array (A*Loc+B) (P53)
    1:     5   Start Loc [WS_out   ]
    2:     10   A1 ;Scale WS, 0 to 100
                 mph = 0 to 1000 mV
    3:     0   B1
    4:     1.8519   A2 ;Scale WD, 0 to 540
                 deg = 0 to 1000 mV
    5:     0   B2
    6:     25   A3 ;Scale Temp, 0 to 40
                 C = 0 to 1000 mV
    7:     0   B3
    8: 1000   A4 ;Scale Rad, 0 to 1 KW
                 = 1 to 1000 mV
    9:     0   B4

09:     SDM-A04 (P103)
    1:     4   Reps
    2:     30   Address
    3:     5   Loc [WS_out   ]

10:     If time is (P92)
    1:     0   Minutes (Seconds --) into a
    2:     60   Interval (same units as above)
    3:     10   Set Output Flag High

11:     Wind Vector (P69)
    1:     1   Reps
    2:     180   Samples per Sub-Interval
    3:     00   S, é1, & å(é1) Polar
    4:     1   Wind Speed/East Loc
                 [WS   ]
    5:     2   Wind Direction/North Loc
                 [0_360_WD]

12:     Average (P71)
    1:     2   Reps
    2:     3   Loc [Ta   ]

INPUT LOCATIONS

1 WS
2 0_360_WD
3 Ta
4 Rs
5 WS_out
6 WD_out
7 Ta_out
8 Rs_out

8.7 CONVERTING 0-360 WIND DIRECTION OUTPUT TO 0-540 FOR STRIP CHART

If 0-360 degree wind direction is output to a strip chart the discontinuity at 0/360 will cause the pen to jump back and forth full scale when the winds are varying from the north. In the days of strip charts this was solved with a 0-540 degree pot on the wind vane (direction changes from 540 to 180 and from 0 to 360 so the pen only jumps once when the wind is out of the north or south).

When faced with the necessity of strip chart output (see previous example), the following algorithm can be used to change a 0-360 degree input to 0-540. (If you have a 0-540 pot, it can be used with the CR23X since the Wind Vector Instruction, 69, will work with this output.)

To change 0-360 degrees to the 0-540 degrees, 360 degrees must sometimes be added to the reading when it is in the range of 0 to 180. The following algorithm does this by assuming that if the previous reading was less than 270, the vane has shifted through 180 degrees and does not need to be altered. If the previous 0-540 reading was greater than 270, 360 degrees is added.

This example is written as a subroutine which is used by the previous example to output an analog voltage to a strip chart.

*   Table 3 Subroutines

01:     Beginning of Subroutine (P85)
    1:     1   Subroutine 1

02:     IF (X<=>F) (P89)
    1: 10   X Loc [0_540_WD]
    2:     3   >=
    3: 270   F
    4: 30   Then Do

03:     Do (P86)
    1: 11   Set Flag 1 High

04:     Else (P94)

05:     Do (P86)
    1: 21   Set Flag 1 Low

06:     End (P95)

07:    Z=X (P31)
    1:    2   X Loc [0_360_WD]
    2: 10   Z Loc [0_540_WD]

08:    IF (X<=>F) (P89)
    1: 10   X Loc [0_540_WD]
    2:    4   <
    3: 180   F
    4: 30   Then Do

09:    If Flag/Port (P91)
    1: 11   Do if Flag 1 is High
    2: 30   Then Do

10:    Z=X+F (P34)
    1: 10   X Loc [0_540_WD]
    2: 360   F
    3: 10   Z Loc [0_540_WD]

11:    Z=X (P31)
    1: 10   X Loc [0_540_WD]
    2:    6   Z Loc [0_540_out]

12:    End (P95)

13:    End (P95)

14:    End (P95)

INPUT LOCATIONS

2 0_360_WD
6 0_540_out
10 0_540_WD

8.8 USE OF 2 FINAL STORAGE AREAS - SAVING DATA PRIOR TO EVENT

One of the uses of 2 Final Storage Areas is to save a fixed amount of data before and after some event.

In this example, a load cell is measured every second. It is assumed that at some random interval the load will exceed 25 pounds for less than 10 seconds. Exceeding 25 pounds is the event to be captured. The data from the 10 seconds before the event and 10 seconds after the event is to be saved (21 seconds including the scan in which the load first exceeds 25 pounds).

Every second the load cell is measured; hours-minutes, seconds, and the load are output to Final Storage Area 2 (4 values with the Array ID). 84 locations are allocated to Final Storage

Area 2. Thus, Area 2 holds 21 seconds (4 values/second x 21 seconds = 84 locations).

When 25 pounds is exceeded, 10 is loaded into an input location and flag 1 is set high.   The input location is used as a down counter. The flag indicates an event has occurred and prevents the input location from being reloaded until 11 seconds have passed.

The down counter is decremented by 1 each time the table is executed. When it equals 0 all the data in Final Storage Area 2 is transferred to Final Storage Area 1 (using Instruction 96) and Flag 1 is set low.

The down counter is set to 10 instead of 11 because it is decremented after checking to see if it is 0.

PROGRAM

*   Table 1 Program
  01:          1   Execution Interval (seconds)

01:    Full Bridge (P6)
    1:      1   Reps
    2:     21   ±10 mV Slow 60 Hz Rejection Range
    3:      1   DIFF Channel
    4:      1   Excite all reps w/Exchan 1
    5: 2500   mV Excitation
    6:      1   Loc [Force_kg]
    7: 15.120   Mult
    8:      0   Offset

02:    Do (P86)
    1:     10   Set Output Flag High

03:    Set Active Storage Area (P80)
    1:      2   Final Storage Area 2
    2:     10   Array ID or Loc
              [_____]

04:    Real Time (P77)
    1:     11   Hour/Minute,Seconds

05:    Sample (P70)
    1:      1   Reps
    2:      1   Loc [Force_kg]

06:    IF (X<=>F) (P89)
    1:      1   X Loc [Force_kg]
    2:      3   >=
    3:     25   F
    4:     30   Then Do

07:   If Flag/Port (P91)
        1:    21   Do if Flag 1 is Low
        2:    30   Then Do

08:   Do (P86)
        1:    11   Set Flag 1 High

09:   Z=F (P30)
        1:    10   F
        2:    0   Exponent of 10
        3:    2   Z Loc [Down_cnt]

10:   End (P95)

11:   End (P95)

12:   IF (X<=>F) (P89)
        1:    2   X Loc [Down_cnt]
        2:    1   =
        3:    0   F
        4:    30   Then Do

13:   If Flag/Port (P91)
        1:    11   Do if Flag 1 is High
        2:    30   Then Do

14:   Serial Out (P96)
        1:    81   All Data to other FS Area

15:   Do (P86)
        1:    21   Set Flag 1 Low

16:   End (P95)

17:   Else (P94)

18:   Z=X+F (P34)
        1:    2   X Loc [Down_cnt]
        2:    -1   F
        3:    2   Z Loc [Down_cnt]

19:   End (P95)

*   A   Mode 10 Memory Allocation
  01:      28   Input Locations
  02:      64   Intermediate Locations
  03:      84   Final Storage Area 2

INPUT LOCATIONS

1 Force_kg
2 Down_cnt

8.9 LOGARITHMIC SAMPLING USING LOOPS

A ground water pump test requires that water level be measured and recorded according to the following schedule.

Time into Test, min				Output Interval Loop #
00	to	10	10 sec.	1
10	to	30	30 sec.	2
30	to	100	1 min.	3
100	to	300	2 min.	4
300	to	1000	5 min.	5
1000	and greater		10 min.	6

This is accomplished with a series of loops (Instruction 87), where the delay and count parameters are used to implement the frequency of measurement (and output) and the duration of the that frequency. The unit of delay is the execution interval. A delay of 1 with a 10 second execution interval and a count of 60 means the instructions in the loop, in this case measure and output water level, are executed every 10 seconds for 10 minutes.

The drawdown portion of the test is completed at some time greater than 1000 minutes. To enter the recharge phase of the test, the operator enters the ✳ 6 A D Mode and sets Flag 1 high. At the next 10 minute pass through loop 6 the loop is exited. Program execution returns to the top of the program table and the measurement schedule starts over again for the recharge test.

The sensor is a 50 PSI Druck, model 930/ti with a calibration of 49.93 mV/10V of excitation or 4.993 mV/V. Your calibration will be different. An excitation voltage of 1500 mV yields a maximum signal of 7.489 mV at 50 PSI, fully utilizing the 7.5 mV Input Range to provide the best resolution.

The multiplier, m, is calculated to provide depth of water in feet:

m = (50 psi/4.993 mV/V) * (2.3067 ft/psi)

m = 23.099 ft/mV/V

The offset is calculated to provide a final value that represents the distance from the lip of the well to the water surface. Similar to Figure 7.16-2, the offset equals the initial distance of 47.23 feet plus the initial reading of 54.77, or 102 feet.

PROGRAM

```
*       Table 1 Program
  01:       10            Execution Interval (seconds)
```

;User must toggle Flag 1 to start measurements.
;

```
01:     If Flag/Port (P91)
        1:    21          Do if Flag 1 is Low
        2:     0          Go to end of Program Table
```

;Loop 1, Output every 10 seconds for 10 minutes.
;

```
02:     Beginning of Loop (P87)
        1:     1          Delay
        2:    60          Loop Count

03:     Do (P86)
        1:     1          Call Subroutine 1

04:     End (P95)
```

;Loop 2, Output every 30 seconds for 20 minutes.
;

```
05:     Beginning of Loop (P87)
        1:     3          Delay
        2:    40          Loop Count

06:     Do (P86)
        1:     1          Call Subroutine 1

07:     End (P95)
```

;Loop 3, Output every 1 minute for 70 minutes.
;

```
08:     Beginning of Loop (P87)
        1:     6          Delay
        2:    70          Loop Count

09:     Do (P86)
        1:     1          Call Subroutine 1

10:     End (P95)
```

;Loop 4, Output every 2 minutes for 200 minutes.
;

```
11:     Beginning of Loop (P87)
        1:    12          Delay
        2:   100          Loop Count

12:     Do (P86)
        1:     1          Call Subroutine 1

13:     End (P95)
```

;Loop 5, Output every 5 minutes for 700 minutes.
;

```
14:     Beginning of Loop (P87)
        1:    30          Delay
        2:   140          Loop Count

15:     Do (P86)
        1:     1          Call Subroutine 1

16:     End (P95)
```

;Loop 6, Output every 10 minutes until stopped by user.
;

```
17:     Beginning of Loop (P87)
        1:    60          Delay
        2:     0          Loop Count

18:     Do (P86)
        1:     1          Call Subroutine 1

19:     If Flag/Port (P91)
        1:    21          Do if Flag 1 is Low
        2:    31          Exit Loop if True

20:     End (P95)

*       Table 3 Subroutines

01:     Beginning of Subroutine (P85)
        1:     1          Subroutine 1

02:     Full Bridge (P6)
        1:     1          Reps
        2:    21          ±10 mV Slow 60 Hz Rejection
                          Range
        3:     1          DIFF Channel
        4:     1          Excite all reps w/Exchan 1
        5:  1500          mV Excitation
        6:     1          Loc [ Level_Ft  ]
        7:      .46199 Mult
        8:   102          Offset

03:     Do (P86)
        1:    10          Set Output Flag High

04:     Real Time (P77)
        1:  0111          Day,Hour/Minute,Seconds

05:     Sample (P70)
        1:     1          Reps
        2:     1          Loc [ Level_Ft  ]

06:     End (P95)
```

INPUT LOCATIONS

1 Level_Ft

8.10 COVARIANCE CORRELATION PROGRAMMING EXAMPLE

The example is a 2 level meteorological tower with 5 sensors at each level. The three components of the wind are measured using prop anemometers. Two thermocouples (TC) are used to measure ambient and wet-bulb temperatures and calculate water vapor pressure on-line. All sensors are scanned once per second (1 Hz) and a 5 minute subinterval averaging period with a 30 minute Output Interval is specified. The example optimizes the input measurement sequence for speed and shows the instructions necessary to provide calibrated inputs, properly ordered to produce the desired outputs from the Covariance Correlation (COV/CORR) Instruction. Table 8.10-1 groups the sensors according to measurement type and gives the CR23X multiplier and offset.

The props can all be measured as single-ended voltages. The vertical wind prop calibration differs from the U and V prop calibration. The fastest input sequence is to measure both levels (6 props) with a single instruction using the U and V calibration and correct the W measurements with the Fixed Multiply, Instruction 37.

An AM25T Solid State Thermocouple Multiplexer is used to measure the four Type E thermocouples. The AM25T has a built in 1000 ohm PRT that is used as a reference temperature for the thermocouples. The thermocouples are measured on the most sensitive input range, 2.5 mV, accommodating a $\pm 40^oC$ range between the measurement and AM25T TC reference junction. The resolution is ($.33 \mu V/(60 \mu V/^oC)$ or about 0.006^oC. Measuring absolute temperature with TCs requires a reference junction temperature measurement. The reference PRT is measured with Instruction 6. Temperature is computed with Instructions 59 and 16.

The specified outputs determine the input order required by the COV/CORR Instruction. Table 8.10-2 lists the desired outputs from the two levels along with the Input Storage locations for the processed results.

TABLE 8.10-1. Example Sensor Description and CR23X Multiplier and Offset

DESCRIPTION	SYMBOL	SENSOR	CALIB	MEAS TYPE	MULT	OFFSET
Horiz. Wind	U	prop	18m/s/V	S.E.V.	.018m/s/mV	0.0
Horiz. Wind	V	prop	18m/s/V	S.E.V.	.018	0.0
Vert. Wind	W	prop	22m/s/V	S.E.V.	.022	0.0
Air Temp.	Ta	TC	-	TC DIFF.	1.0^oC	0.0
Wet-bulb Temp.	Tw	TC	-	TC DIFF.	1.0^oC	0.0
Vap. Pressure	e	derived	-	-	-	-

TABLE 8.10-2. Example Outputs and Input Storage Locations

LEVEL 1 OUTPUTS

MEANS	LOC	VARIANCES	LOC	COVARIANCES	LOC	CORRELATIONS	LOC
M(W1)	20	V(W1)	25	CV(W1,U1)	30	CR(W1,U1)	34
M(U1)	21	V(U1)	26	CV(W1,V1)	31	CR(W1,V1)	35
M(V1)	22	V(V1)	27	CV(W1,Ta1)	32		
M(Ta1)	23	V(Ta1)	28	CV(W1,e1)	33		
M(e1)	24	V(e1)	29				

LEVEL 2 OUTPUTS

MEANS	LOC	VARIANCES	LOC	COVARIANCES	LOC
M(W2)	36	V(W2)	41	CV(W2,U2)	46
M(U2)	37	V(U2)	42	CV(W2,V2)	47
M(V2)	38	V(V2)	43	CV(W2,Ta2)	48
M(Ta2)	39	V(Ta2)	44	CV(W2,e2)	49
M(e2)	40	V(e2)	45	CV(U2,V2)	50
				CV(U2,Ta2)	51
				CV(U2,e2)	52
				CV(V2,Ta2)	53
				CV(V2,e2)	54

Table 8.10-3 lists the input channel configuration and Input Storage allocation for the measured values. After reading the new input samples, the Level 2 measurements are relocated using the Block Move Instruction 54, then Ta1 is relocated through a separate move and e1 is positioned by specifying the destination location in the Wet/Dry-Bulb Instruction. The COV/CORR Instruction must be entered twice, once for each level.

In addition to ordering Level 1 and Level 2 in locations 1-5 and 11-15 respectively, 2 more locations are required. Converting the wet-/dry-bulb measurements to vapor pressure using Instruction 57 requires atmospheric pressure. We'll use the standard atmosphere for the site elevation and key the value into Location 17 using the C command in the ∗ 6 Mode.

This example requires that 54 locations be allotted to Input Storage and 79 to Intermediate Storage (35 for the 1st COV/CORR Instruction, 43 for the second, and 1 for Instruction 92).

TABLE 8.10-3. Example Input Channel and Location Assignments

PARAM	INPUT CHAN	INPUT LOC		PARAM	INPUT LOC		PARA	INPUT LOC
W1	1	1		W1	1		W1	1
U1	2	2		U1	2		U1	2
V1	3	3		V1	3		V1	3
W2	4	4		Ta1	9	----------------------	Ta1	4
U2	5	5		Tw1	10	Separate moves	e1	5
V2	6	6		W2	11		W2	11
Ta2	7	7	->Block->	U2	12		U2	12
Tw2	8	8	move	V2	13		V2	13
Ta1	9	9		Ta2	14		Ta2	14
Tw1	10	10		Tw2	15		e2	15

PROGRAM

```
*       Table 1 Program
  01:          1           Execution Interval (seconds)

01:    Set Port(s) (P20)
       1:   9999           C8..C5 = nc/nc/nc/nc
       2:   9933           C4..C1 = nc/nc/1ms/1ms

02:    Volts (SE) (P1)
       1:      6           Reps
       2:      5           ±5000 mV Slow Range
       3:      1           SE Channel
       4:      1           Loc [ W1      ]
       5:    .018          Mult
       6:      0           Offset
```

```
03:    AM25TMultiplexer (P134)
       1:      4       Reps
       2:      11      10 mV, Fast Range
       3:      1       Channel
       4:      4       DIFF Channel
       5:      21      Exchan 1, 60 Hz Reject
       6:      1       Clock Control
       7:      2       Reset Control
       8:      2       Type E (Chromel-Constantan)
       9:      16      Ref Temp (Deg. C) Loc [ Ref_Temp ]
       10:     7       Loc [ Ta2_1    ]
       11:     1.0     Mult
       12:     0.0     Offset

04:    Z=X*F (P37)
       1:      1       X Loc [ W1     ]
       2:      1.2222  F
       3:      1       Z Loc [ W1     ]

05:    Z=X*F (P37)
       1:      4       X Loc [ W2     ]
       2:      1.2222  F
       3:      4       Z Loc [ W2     ]

06:    Block Move (P54)
       1:      5       No. of Values
       2:      4       First Source Loc [ W2      ]
       3:      1       Source Step
       4:      11      First Destination Loc
                       [ W2_i     ]
       5:      1       Destination Step

07:    Z=X (P31)
       1:      9       X Loc [ Ta1    ]
       2:      4       Z Loc [ W2     ]

08:    Wet/Dry Bulb Temp to VP (P57)
       1:      17      Pressure Loc [ Pressure ]
       2:      9       Dry Bulb Loc [ Ta1     ]
       3:      10      Wet Bulb Loc [ Tw1     ]
       4:      5       Loc [ U2       ]

09:    Wet/Dry Bulb Temp to VP (P57)
       1:      17      Pressure Loc [ Pressure ]
       2:      14      Dry Bulb Loc [ Ta2_i   ]
       3:      15      Wet Bulb Loc [ Tw2_i   ]
       4:      15      Loc [ Tw2_i    ]

10:    If time is (P92)
       1:      0       Minutes (Seconds --) into a
       2:      30      Interval (same units as above)
       3:      10      Set Output Flag High
```

SECTION 8. PROCESSING AND PROGRAM CONTROL EXAMPLES

11:     Covariance/Correlation (P62)
        1:      5   No. of Input Locations
        2:      5   No. of Means
        3:      5   No. of Variances
        4:      0   No. of Std. Dev.
        5:      4   No. of Covariance
        6:      2   No. of Correlations
        7:    300   Samples per Average
        8:      1   First Sample Loc [W1   ]
        9:     20   Loc [MEAN_W1   ]

12:     Covariance/Correlation (P62)
        1:      5   No. of Input Locations
        2:      5   No. of Means
        3:      5   No. of Variances
        4:      0   No. of Std. Dev.
        5:      9   No. of Covariance
        6:      2   No. of Correlations
        7:    300   Samples per Average
        8:     11   First Sample Loc [W2_i   ]
        9:     36   Loc [mean_W2   ]

13:     Real Time (P77)
        1:   0110   Day,Hour/Minute

14:     Sample (P70)
        1:     35   Reps
        2:     20   Loc [mean_W1   ]

TABLE 8.10-4. Thirty Minute Output From Example

01 119	02 DAY	03 HRMIN	04 M(W1)	05 M(U1)	06 M(V1)	07 M(Ta1)	08 M(e1)
09 V(W1)	10 V(U1)	11 V(V1)	12 V(Ta1)	13 V(e1)	14 CV(W1,U1)	15 CV(W1,V1)	16 CV(W1,Ta1)
17 CV(W1,e1)	18 CR(W1,U1)	19 CR(W1,V1)	20 M(W2)	21 M(U2)	22 M(V2)	23 M(Ta2)	24 M(e2)
25 V(W2)	26 V(U2)	27 V(V2)	28 V(Ta2)	29 V(e2)	30 CV(W2,U2)	31 CV(W2,V2)	32 CV(W2,Ta2)
33 CV(W2,e2)	34 CV(U2,V2)	35 CV(U2,Ta2)	36 CV(U2,e2)	37 CV(V2,Ta2)	38 CV(V2,e2)		

8.11 FAST FOURIER TRANSFORM EXAMPLES

8.11.1. EXAMPLE WITHOUT BIN AVERAGING

The CR23X was used to generate data representing two superimposed sine wave signals, one at 1.25 Hz (amplitude = 1) and the other at 0.25 Hz (amplitude = 2). The 1024 generated samples simulate a sampling rate of 10 Hz or a 0.1 second scan rate. Figure 8.11-1 shows a plot of the simulated signal. The FFT was applied to the data and the real and imaginary, phase and magnitude, and the power spectra results are shown in Tables 8.11-1, 8.11-2, and 8.11-3

respectively. A portion of the power spectra results are illustrated in Figure 8.11-2.

The phase of the cosine wave that describes the signal at the *beginning* of the first interval and the *end* of the last interval can be determined by looking at the CR23X program that generated the "original time series data". The 1.25 Hz signal began and ended at 270 degrees [cos 270 = cos(0 - 90) = sin 0]. The 0.25 Hz signal began at 270 degrees and ended at 126 degrees. The phases of the 1.25 and 0.25 signals are 270 and 198 respectively (Table 8.11-2).
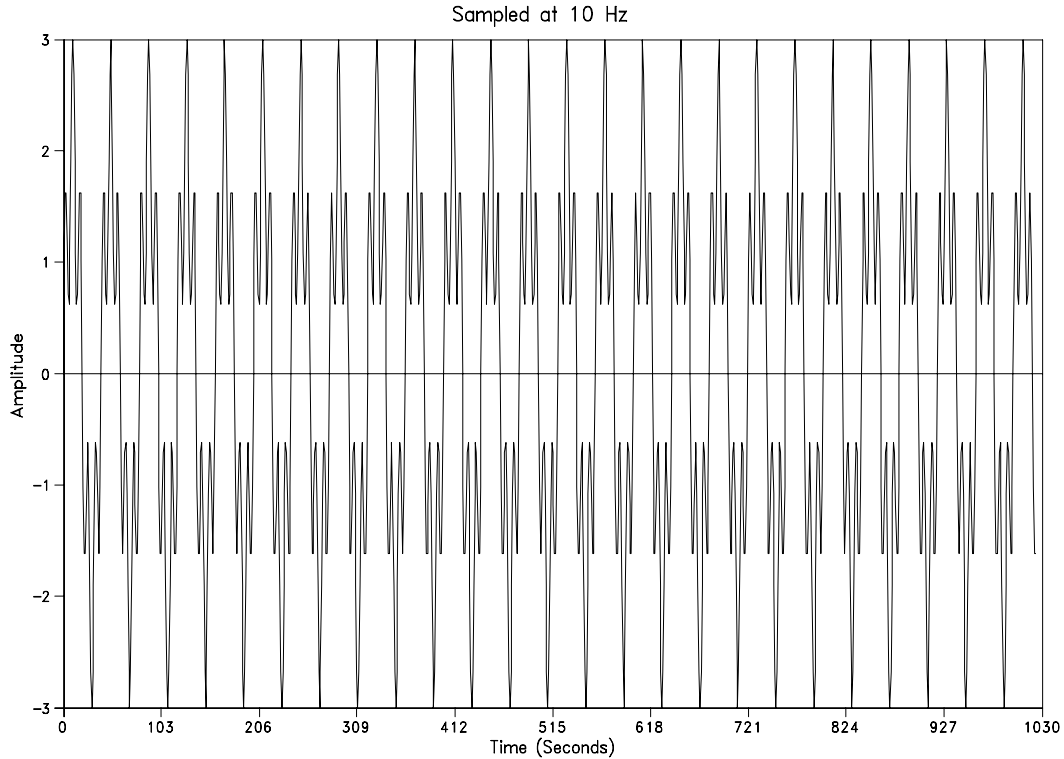
FIGURE 8.11-1. Simulated 1.25 and 0.25 Hz Signals
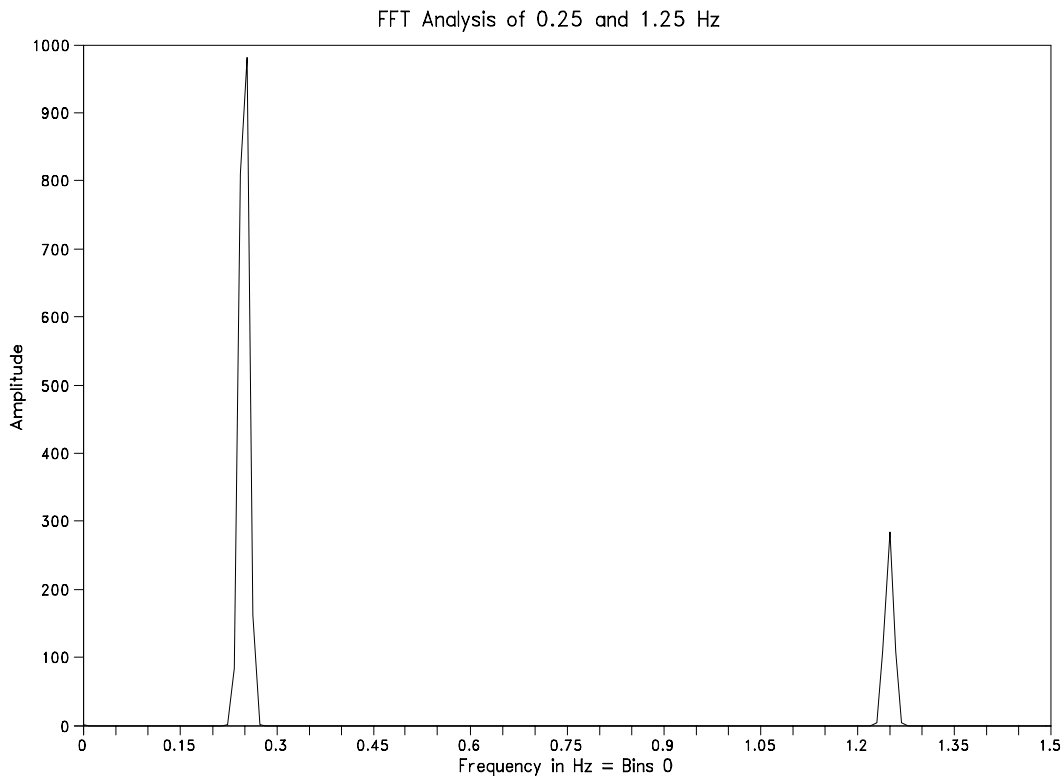
FIGURE 8.11-2. FFT Power Spectra Analysis of 0.25 and 1.25 Hz Signal

TABLE 8.11-1.   FFT Real and Imaginary Results 0.25 and 1.25 Hz Signal

BIN #	Hz	FFT Ri	FFT Ii
0	0	0.02303	0
1:	0.009766	0.01036	0
2	0.019532	-0.00206	0
3	0.029298	0	0
22	0.214852	-0.00086	-0.00009
23	0.224618	0.01096	0.0036
24	0.234384	-0.19328	-0.06277
25	0.24415	0.59858	0.19439
26	0.253916	-0.65827*	-0.21391*
27:	0.263682	0.26778	0.08709
28	0.273448	-0.02466	-0.00796
29	0.283214	0.00086	0.00034
125	1.22075	0	0.00069
126	1.230516	0.00009	-0.04342
127	1.240282	0.00009	0.23044
128	1.250048	0*	-0.37302*
129:	1.259814	0	0.23053
130	1.26958	-0.00009	-0.0435
131	1.279346	0.00009	0.0006
511	4.990426	-0.00009	0

TABLE 8.11-2. FFT Magnitude and Phase Results 0.25 and 1.25 Hz Signal

BIN #	Hz	FFT Mi	FFT Pi
0	0	0.02303	0
1	0.009766	0.01036	0
2	0.019532	0.00206	180
3:	0.029298	0	-99999
22	0.214852	0.00086	185.58
23	0.224618	0.01154	17.952
24	0.234384	0.20321	197.78
25	0.24415	0.62935	17.776
26	0.253916	0.69215*	197.79*
27	0.263682	0.28158	17.801
28	0.273448	0.02592	197.68
29:	0.283214	0.00092	21.646
125	1.22075	0.00069	90
126	1.230516	0.04342	270.11
127	1.240282	0.23044	89.979
128	1.250048	0.37302*	270*
129	1.259814	0.23053	90
130	1.26958	0.0435	269.89
131:	1.279346	0.00061	82.042
511	4.990426	0.00009	180

TABLE 8.11-3. FFT Power Spectra Results 0.25 and 1.25 Hz Signal

BIN #	Hz	FFT PSi
0:	0	1.0859
22	0.214852	0
23	0.224618	0.49212
24	0.234384	84.152
25	0.24415	811.01
26	0.253916	980.79*
27	0.263682	162.4
28	0.273448	1.4764
29:	0.283214	0
125	1.22075	0
126	1.230516	3.9369
127	1.240282	108.76
128	1.250048	284.94*
129	1.259814	108.76
130	1.26958	3.9369
131:	1.279346	0
511	4.990426	0

PROGRAM

;The first part of this program generates a
;dummy data set that represents two signals
;superimposed on top of each other. One signal
;has a frequency of 1.25 Hz and a zero to peak
;amplitude of 1 while the other signal has a
;frequency of .25 Hz and a zero to peak
;amplitude of 2. The data simulates a sampling
;rate of 10 Hz or a .1 sec scan rate.

;This program generates power spectra FFT
;results with out bin averaging.

;Flag 1 is automatically set once the inst. have
;been executed once.

*Table 1 Program
 01:          1   Execution Interval (seconds)

01:   If Flag/Port (P91)
          1:     11   Do if Flag 1 is High
          2:      0   Go to end of Program Table

;The 2nd through 11th instructions are used to
;generate and combine the 1.25 and 0.25 Hz
;signals for the FFT.

02:    Z=F (P30)
   1:      0   F
   2:      0   Exponent of 10
   3: 1025   Z Loc [_____]

03:    Z=F (P30)
   1:      0   F
   2:      0   Exponent of 10
   3: 1026   Z Loc [_____]

04:    Beginning of Loop (P87)
   1:      0   Delay
   2: 1024   Loop Count

05:    Z=SIN(X) (P48)
   1: 1025   X Loc [_____]
   2: 1027   Z Loc [_____]

06:    Z=SIN(X) (P48)
   1: 1026   X Loc [_____]
   2: 1028   Z Loc [_____]

07:    Z=X*F (P37)
   1: 1028   X Loc [_____]
   2:      2   F
   3: 1028   Z Loc [_____]

08:    Z=X+Y (P33)
   1: 1027   X Loc [_____]
   2: 1028   Y Loc [_____]
   3:    1--   Z Loc [#1   ]

09:    Z=X+F (P34)
   1: 1025   X Loc [_____]
   2:    45   F
   3: 1025   Z Loc [_____]

10:    Z=X+F (P34)
   1: 1026   X Loc [_____]
   2:     9   F
   3: 1026   Z Loc [_____]

11:    End (P95)

;The FFT is now computed and the power
;spectra results sent to Final Storage.

12:    FFT (P60)
   1:    10   Log (base 2) of Samples
   2:     1   Power Spectra/Taper
   3:     0   Log (base 2) of Bins
   4:     1   First Sample Loc [#1   ]
   5:     1   Mult

13:    Beginning of Loop (P87)
   1:      0   Delay
   2:   512   Loop Count

14:    Do (P86)
   1:    10   Set Output Flag High

15:    Resolution (P78)
   1:     1   high resolution

16:    Sample (P70)
   1:     1   Reps
   2:    1--   Loc [#1   ]

17:    End (P95)

18:    Do (P86)
   1:    11   Set Flag 1 High

*Table 2 Program
  02:        0.0000   Execution Interval (seconds)

*Table 3 Subroutines

End Program

*   A   Mode 10 Memory Allocation
  01: 1030   Input Locations
  02:   260   Intermediate Locations

8.11.2 EXAMPLE WITH BIN AVERAGING

The CR23X was used to generate data simulating wave data from an ocean buoy with four superimposed sine wave signals, 0.1, 0.125, 0.14, and 0.2 Hz. The 2048 generated samples simulate a sampling rate of 0.5 Hz or a 2.0 second scan rate. Figure 8.11-3 shows a plot of part of the simulated signal. A FFT with 8 bin averaging was performed on the data. A multiplier of 0.1 was used to keep the FFT results smaller than the +6999 upper limit of low resolution Final Storage. The results of the FFT are shown Table 8.11-4 and are illustrated in Figure 8.11-4.

In the example program, a multiplier of 0.1 is used in the FFT Instruction. By reducing the FFT results by a factor of 10, the Low Resolution output format can be used, thus maximizing the Final Storage capacity. A Low Resolution data point requires 2 bytes of Final Storage memory, while a High Resolution data point requires 4 bytes. When memory is a limiting factor, the data should be scaled to be less than 6999, so the Low Resolution format can be used.

Simulated Ocean Wave Buoy Data
Freq/Ampl .1/.11, .125/9, .14/6, .2/4
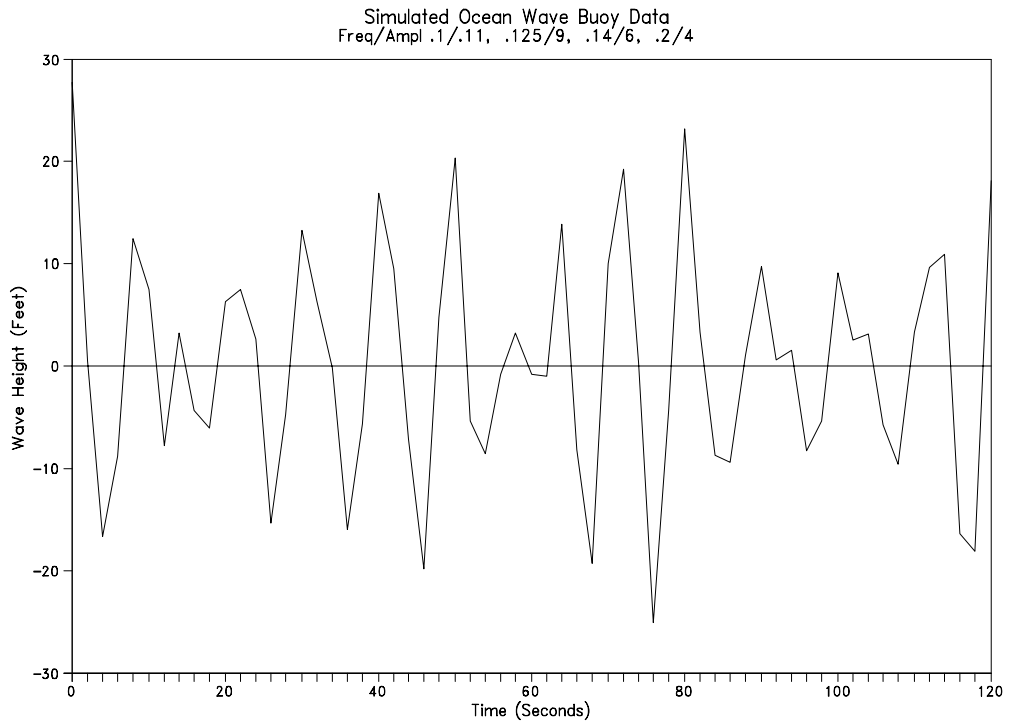
FIGURE 8.11-3. Simulated Ocean Buoy Wave Data

Simulated Ocean Wave Buoy FFT Results
Freq/Ampl .1/.11, .125/9, .14/6, .2/4
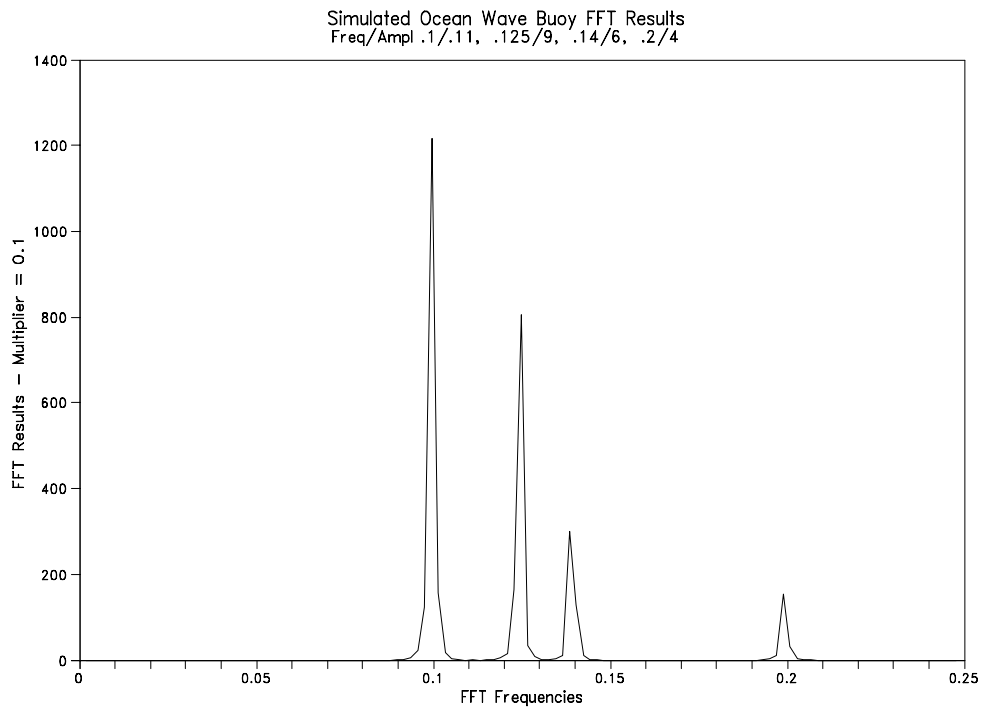
FIGURE 8.11-4. Simulated Ocean Buoy FFT Results

TABLE 8.11-4. FFT Bin Averaging Results from Simulated Ocean Buoy Wave Data

BIN #	FREQUENCY	FFT*0.1	BIN #	FREQUENCY	FFT*0.1
1	0.00195	0	26	0.0507	0
2	0.0039	0	27	0.05265	0
3	0.00585	0	28	0.0546	0
4	0.0078	0	29	0.05655	0
5	0.00975	0	30	0.0585	0
6	0.0117	0	31	0.06045	0
7	0.01365	0	32	0.0624	0
8	0.0156	0	33	0.06435	0
9	0.01755	0	34	0.0663	0
10	0.0195	0	35	0.06825	0
11	0.02145	0	36	0.0702	0
12	0.0234	0	37	0.07215	0
13	0.02535	0	38	0.0741	0
14	0.0273	0	39	0.07605	0
15	0.02925	0	40	0.078	0
16	0.0312	0	41	0.07995	0
17	0.03315	0	42	0.0819	0
18	0.0351	0	43	0.08385	0
19	0.03705	0	44	0.0858	0
20	0.039	0	45	0.08775	0
21	0.04095	0	46	0.0897	0.226
22	0.0429	0	47	0.09165	0.527
23	0.04485	0	48	0.0936	6.246
24	0.0468	0	49	0.09555	21.52
25	0.04875	0	50	0.0975	123.1

PROGRAM

;Simulate ocean wave data with 4 super
;imposed sine wave signals (Hz/zero to peak
;amplitude: .1/11, .125/9,

;Flag 1 high = generate and store "original time
;series data". Flag 2 high = calculate and store
;FFT results.

;Output Array Definitions:

;   104 = "ORIGINAL TIME SERIES DATA"

;   113= FFT POWER SPECTRA RESULTS

*   Table 1 Program
  01:          2   Execution Interval (seconds)

;If flag 1 is set, the simulated ocean wave data
;is calculated and sent to Final Storage.

01:    If Flag/Port (P91)
       1:     11   Do if Flag 1 is High
       2:     30   Then Do

02:    Do (P86)
       1:      1   Call Subroutine 1

03:    Beginning of Loop (P87)
       1:      0   Delay
       2:    2048   Loop Count

04:    Do (P86)
       1:     10   Set Output Flag High

05:    Real Time (P77)
       1:      1   Seconds

06:    Sample (P70)
       1:      1   Reps
       2:    284--   Loc [_____]

07:    End (P95)

08:    End (P95)

09:    Do (P86)
       1:     21   Set Flag 1 Low

;When flag 2 is set the FFT is computed and the
;results are sent to Final Storage.

10: If Flag/Port (P91)
    1:     12   Do if Flag 2 is High
    2:     30   Then Do

11: FFT (P60)
    1:     11   Log (base 2) of Samples
    2:      1   Power Spectra/Taper
    3:      3   Log (base 2) of Bins
    4:    284   First Sample Loc
                    [_____]
    5:      .1   Mult

12: Beginning of Loop (P87)
    1:      0   Delay
    2:   127   Loop Count

13: Do (P86)
    1:     10   Set Output Flag High

14: Sample (P70)
    1:      1   Reps
    2: 284--   Loc [_____]

15: End (P95)

16: End (P95)

17: Do (P86)
    1:     22   Set Flag 2 Low

*   Table 2 Program
  02:        0.0000   Execution Interval
                    (seconds)

*   Table 3 Subroutines

;Subroutine 1 creates the simulated ocean wave
;data. The 3rd through 7th instructions are used
;to slightly randomize the wave signal.

01: Beginning of Subroutine (P85)
    1:      1   Subroutine 1

02: Beginning of Loop (P87)
    1:      0   Delay
    2: 2048   Loop Count

03: Polynomial (P55)
    1:      1   Reps
    2:      1   X Loc [#1   ]
    3:      1   F(X) Loc [#1   ]
    4:     .1   C0
    5:     99   C1
    6:      0   C2
    7:      0   C3
    8:      0   C4
    9:      0   C5

04: Z=FRAC(X) (P44)
    1:      1   X Loc [#1   ]
    2:      1   Z Loc [#1   ]

05: Z=X+F (P34)
    1:      1   X Loc [#1   ]
    2:   -.5   F
    3:      2   Z Loc [_____]

06: Z=X*F (P37)
    1:      2   X Loc [_____]
    2:     30   F
    3:      2   Z Loc [_____]

07: Low Pass Filter (P58)
    1:      1   Reps
    2:      2   Sample Loc [_____]
    3:      3   Loc [_____]
    4:    .02   Weighting Factor

08: Scaling Array (A*Loc+B) (P53)
    1:      4   Start Loc [_____]
    2:      1   A1
    3:   144   B1
    4:      1   A2
    5: 100.77   B2
    6:      1   A3
    7:     90   B3
    8:      1   A4
    9:     72   B4

09: Beginning of Loop (P87)
    1:      0   Delay
    2:      4   Loop Count

10: Z=X+Y (P33)
    1:      3   X Loc [_____]
    2:    4--   Y Loc [_____]
    3:    4--   Z Loc [_____]

11: Z=X MOD F (P46)
    1:    4--   X Loc [_____]
    2:   360   F
    3:    4--   Z Loc [_____]

12: Z=SIN(X) (P48)
    1:    4--   X Loc [_____]
    2:    8--   Z Loc [_____]

13: End (P95)

14:    Scaling Array (A*Loc+B) (P53)
       1:    8   Start Loc [_____]
       2:    4   A1
       3:    0   B1
       4:    6   A2
       5:    0   B2
       6:    9   A3
       7:    0   B3
       8:    11   A4
       9:    0   B4

15:    Z=F (P30)
       1:    0   F
       2:    0   Exponent of 10
       3:    12   Z Loc [_____]

16:    Beginning of Loop (P87)
       1:    0   Delay
       2:    4   Loop Count

17:    Z=X+Y (P33)
       1:    8--   X Loc [_____]
       2:    12   Y Loc [_____]
       3:    12   Z Loc [_____]

18:    End (P95)

19:    Z=X*F (P37)
       1:    12   X Loc [_____]
       2:    1   F
       3:    284--   Z Loc [_____]

20:    End (P95)

21:    End (P95)

End Program

22:       P   End Table 3

*   A   Mode 10 Memory Allocation
  01:   2500   Input Locations
  02:    270   Intermediate Locations

8.12 USING THE SWITCHED 12 V TO POWER SENSORS

In this example, it is required to measure air temperature at two levels at a remote site where AC power is not available. The station is powered with a 12 V battery that is trickle charged by a solar panel. Two 12 VDC aspirated thermocouples (ASPTC) are used measure air temperature. To minimize the current drain of the ASPTCs, the fans are turned on 20 seconds before the CR23X measures the thermocouples. Immediately after the thermocouple measurements are made the fans are turned off.

The Switched 12 V terminal on the CR23X wiring panel is used to power the ASPTCs. The maximum amount of current that the Switched 12 V terminal can source is 600 mA at 50°C (360 mA at 80°C). Each ASPTC draws 140 mA, for a total ASPTC current requirement of 280 mA (see Figure 8.12-1).

FIGURE 8.12-1. Connections to Power Two 12 V ASPTCs with the Switched 12 V on the CR23X Wiring Panel

PROGRAM

```
*       Table 1 Program
  01:        20        Execution Interval (seconds)

01:    If time is (P92)
       1:    40--      Minutes (Seconds --) into a
       2:    60        Interval (same units as above)
       3:    49        Set Port 9 High

02:    If time is (P92)
       1:    0         Minutes (Seconds --) into a
       2:    1         Interval (same units as above)
       3:    30        Then Do

03:    If Flag/Port (P91)
       1:    49        Do if Port 9 is High
       2:    30        Then Do

04:    Reference Temperature (P17)
       1:    1         Loc [ Ref_Temp  ]

05:    Thermocouple Temp (DIFF) (P14)
       1:    2         Reps
       2:    1         ±2.5 mV Slow Range
       3:    2         DIFF Channel
       4:    2         Type E (Chromel-Constantan)
       5:    1         Ref Temp Loc [ Ref_Temp  ]
       6:    2         Loc [ Lower_TC  ]
       7:    1         Mult
       8:    0         Offset
```

```
06:    Do (P86)
       1:    59        Set Port 9 Low

07:    Z=X-Y (P35)
       1:    3         X Loc [ Upper_TC  ]
       2:    2         Y Loc [ Lower_TC  ]
       3:    4         Z Loc [ del_TC    ]

08:    If time is (P92)
       1:    0         Minutes (Seconds --) into a
       2:    60        Interval (same units as above)
       3:    10        Set Output Flag High

09:    Real Time (P77)
       1:  0110        Day,Hour/Minute

10:    Average (P71)
       1:    4         Reps
       2:    1         Loc [ Ref_Temp  ]

11:    End (P95)

12:    End (P95)

Input Locations

1 Ref_Temp
2 Lower_TC
3 Upper_TC
4 del_TC
```

SECTION 9. INPUT/OUTPUT INSTRUCTIONS

TABLE 9-1. Input Voltage Ranges and Codes

Range Codes*			Full Scale Range		Resolution Differential**	
Fast 250 µs Integ.	60 Hz Reject.	50 Hz Reject.				
10	20	30	Autorange***			
11	21	31	±10	mV	0.33	µV
12	22	32	±50	mV	1.67	µV
13	23	33	±200	mV	6.66	µV
14	24	34	±1000	mV	33.3	µV
15	25	35	±5000	mV	166	µV

* See "Measurements" section for a full list and explanation of possible range codes.
** Differential measurement, resolution for single-ended measurement is twice value shown.
*** See autoranging precaution in Table 3.5-1.

> **NOTE:** When a voltage input exceeds the range programmed, the value which is stored is set to the maximum negative number and displayed as -99999 in high resolution or -6999 in low resolution.

*** 1 SINGLE-ENDED VOLTS ***

FUNCTION
This Instruction is used to measure voltage at a single-ended input with respect to ground. Output is in millivolts.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Range code (Table 9-1)
03:	2	Single-ended channel number for first measurement
04:	4	Input location for first measurement
05:	FP	Multiplier
06:	FP	Offset

Input locations altered: 1 per repetition

*** 2 DIFFERENTIAL VOLTS ***

FUNCTION
This Instruction reads the voltage difference between the high and low inputs of a differential channel. Table 9-1 contains all valid voltage ranges and their codes. Both the high and low inputs must be within ±5.0 V of the CR23X's ground (see Common Mode Range in Section 14.7.2). Pyranometer and thermopile sensors require a jumper between L and ⏚ to keep them in Common Mode Range. Output is in millivolts.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Range code (Table 9-1)
03:	2	Differential channel number for first measurement
04:	4	Input location for first measurement
05:	FP	Multiplier
06:	FP	Offset

Input locations altered: 1 per repetition

*** 3 PULSE COUNT ***

P3 Pulse Count is used to measure high frequency and switch closure inputs on input channels P1..P4 and C5..C8. Low level AC inputs can also be measured on P1..P4. Table 9-2 shows how to select measurement options discussed in this segment.

SECTION 9. INPUT/OUTPUT INSTRUCTIONS
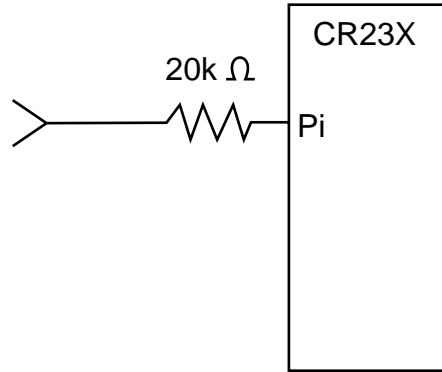
FIGURE 9-1. Conditioning Large Voltage Pulses

Use separate Pulse Count Instructions when measuring both pulse channels and control ports. All Pulse Count instructions must be kept in the same program table. If the Pulse Count Instruction is contained within a subroutine, that subroutine must be called from Table 2.

The use of control ports for pulse measurement causes the CR23X to use a continuous 10 mA of power.

Input Voltage

Excessive pulse voltage inputs can damage the CR23X. Refer to Figure 9-1 if reducing input voltage is required.

- Pulse Channels
  Maximum Input Voltage: ±20 V
- Control Ports
  Maximum Input Voltage: 5.0 V

High Frequency Input

- Pulse Channels
  Minimum Pulse Width: 1.2 microsecond
  Maximum Frequency: 25 kHz
    (8 Bit Counter, 50% Duty Cycle,
    0.015 execution interval)
  Maximum Frequency: 400 kHz
    (16 Bit Counter 50% Duty Cycle)
  Lower Threshold: 1.5 V*
  Upper Threshold: 3.5 V*

When a pulse channel is configured for high-frequency pulse, there is an internal 100 kohm pull-up resistor to 5 V on the pulse channel. This pull-up resistor accommodates open-collector output devices for high-frequency input.

- Control Ports (C5-C8)
  Minimum Pulse Width: 150 ηs
  Maximum Frequency: 2.5 kHz
    Triggered off Rising Edge
  -99999 displayed if maximum exceeded
  Lower Threshold: 1.5 V
  Upper Threshold: 3.5 V

*Larger input transitions are required at high frequencies because of the input RC filter with 1.2 microsecond time constant. Signals up to 400 kHz will be counted if centered around +2.5 V with deviations $\geq \pm$ 2.5 V for $\geq$ 1.2 microseconds.

Low Level AC (Pulse Channels Only)

Input Hysteresis: 15 mV
Maximum Input Voltage: 20 V peak-to-peak
Input Voltage and Frequency Range
  (16 bit counter required above 2.56 kHz)

20 mV	1.0 Hz to 1 kHz
200 mV	0.5 Hz to 10 kHz
1000 mV	0.3 Hz to 16 kHz

Switch Closure

- Pulse Channels
  A switch closure is connected between P1..P4 and analog ground. When the switch is open, the CR23X pulls the pulse channel to 5 V through a 100 kOhm impedance. When the switch is closed, the pulse channel is pulled to ground. The count is incremented when the switch opens.

  Minimum Switch Closed Time: 5 ms
  Minimum Switch Open Time: 6 ms
  Maximum Bounce Time: 1 ms open
    without being counted

- Control Ports

  The switch closure is connected between channels C5..C8 and the 5 V terminal. When the switch is open, the control port is pulled to ground through an internal 100 kOhm resistor. When the switch is closed, the control port is at 5 V. The count is incremented when the switch closes.

  Maximum Frequency: 40 Hz
  Minimum Switch Closed Time: 6 ms
  Minimum Switch Open Time: 6 ms
  High Precision Measurements
    Maximum Program Execution
      Interval: < 4.0 s
    Minimum Frequency: > 0.25 Hz

Pulse Channel Details: Counters, Resets, and Accumulators

Each of the four pulse channels (P1..P4) have an 8 bit counter. The 8 bit counters each have a limit of 255 counts per reset. The 8 bit counter reset will be every 0.01 s (100 Hz), 0.02 s (50 Hz), 0.05 s (20 Hz), or 0.1 s (10 Hz). Table 9-2 shows the maximum input frequency allowed at each reset interval.

TABLE 9-2. 8 Bit Counter Reset Interval and Maximum Input Frequency

Counter Reset Interval (s)	Counter Reset Frequency (Hz)	Maximum Input Frequency (kHz)
0.01	100	25.50
0.02	50	12.75
0.05	20	5.10
0.10	10	2.55

By default, the CR23X will determine which reset interval to use based on the datalogger's programmed execution interval. The CR23X sets the reset interval to the largest reset interval (0.01, 0.02, 0.05, or 0.10) that evenly divides into the execution interval. For clarification, Table 9-3 shows some examples of execution intervals and corresponding counter reset intervals.

As counter reset frequency increases, quiescent current drain of the CR23X increases. If power consumption is less of a concern, parameter 3 of the Pulse Counter (P3) instruction can be indexed (e.g. 03: 0--). Indexing the 3rd parameter in any P3 Pulse Counter instruction will invoke a 100 Hz reset frequency on all pulse channels (P1..P4).

Higher input frequencies can be counted by combining two counters on one input channel. The resulting 16 bit counter allows input frequencies up to 400 kHz. This option is valid only on P1 or P3; the subsequent channel, P2 or P4, becomes inactive. So, maximum number of Reps (parameter 1 in Instruction 3) is 4 when using 8 bit counters, and 2 when using 16 bit counters.

CR23X power usage increases as maximum input frequency increases.

At the reset interval, the processor transfers the values from the counters into 16 bit accumulators. The counters are then hardware reset to zero. The accumulators accumulate up to 65, 535 counts. Counts accumulate in the accumulators until the program table containing Pulse Count Instruction 3 is executed. **The execution interval of the table must be short enough that the accumulator does not overflow.** At the beginning of the execution of the program table (or before each pass through a P87 loop with a delay), totals in the accumulators are transferred to a temporary RAM buffer. The accumulator is then zeroed. When the execution reaches Pulse Count Instruction 3, the value in the RAM buffer is multiplied by the multiplier, added to the offset, then placed into the designated input location.

TABLE 9-3. Execution and Counter Reset Intervals and Maximum Input Frequencies

User Programmed Execution Interval (s)	Counter Reset Interval (s)	Counter Reset Frequency (Hz)	Maximum Input Frequency (kHz)
0.01	0.01	100	25.5
0.02	0.02	50	12.75
0.03	0.01	100	25.5
0.04	0.02	50	12.75
0.05	0.05	20	5.10
0.10	0.10	10	2.55
0.15	0.05	20	5.10
0.80	0.10	10	2.55
≥1.0	0.10	10	2.55

Control Port Details: Coprocessor and Accumulators

Counts on Control Ports C5..C8 are measured by a coprocessor. The coprocessor transfers counts into 16 bit accumulators. The accumulators accumulate up to 65,535 counts. Counts accumulate in the accumulators until the program table containing Pulse Count Instruction 3 is executed. **The execution interval of the table must be short enough that the accumulator does not overflow.** At the beginning of the execution, totals in the accumulators are transferred to a temporary RAM buffer. The accumulator is then zeroed. When the execution reaches Pulse Count Instruction 3, the value in the RAM buffer is multiplied by the multiplier, added to the offset, then placed into the designated input location.

CAUTION: The RAM buffer does not accumulate counts; it is zeroed each time the table is executed regardless of whether or not the pulse instruction is executed. If all counts are necessary, it is imperative that the Pulse Count Instruction be executed (not branched around) every time the table is executed.

When the execution interval is < 4.0 s, ports C5..C8 measure the actual time between edges of the pulse with a 0.5 μs resolution. The resulting frequency resolution is (0.5)*(Frequency). Therefore, for execution intervals < 4.0 second, ports C5..C8 yield more accurate frequency measurements, especially for low-frequency signals, than do pulse counters P1..P4.

For execution intervals ≥ 4.0 s, ports C5..C8 simply accumulate pulses similar to pulse counter channels P1..P4.

Using ports C5..C8 to count pulses increases the quiescent current of the CR23X by 8 mA. Maximum input frequency on ports C5..C8 is 2.5 kHz.

Table Overruns / Executions

Table overruns occur when a program is too long for its programmed execution interval, the CR23X clock is reset, or a table of higher priority is still running. When a table overrun occurs, the value in the accumulator is the result of a longer than normal interval. This value can either be used or it can be discarded. If pulse counts are being totalized, a missing count could be significant and the value from the erroneously long interval should NOT be discarded. If the pulse count is being processed in a way in which the resultant value is dependent upon the sampling interval (e.g., speed, RPM), the value from the excessive interval should be discarded. If the value is discarded the value in the RAM buffer from the previous measurement will be used.

Frequency Result

An option to output the count as a frequency (Hz or Hertz) is also available. At Execution Intervals less than four seconds, control ports C5..C8 measure the time element in the frequency calculation, whereas pulse channels P1..P4 simply use the programmed Execution Interval. Consequently, at Execution Intervals

less than 4 seconds, control ports C5..C8 measure frequency much more precisely than do pulse channels.

- Pulse Channels
  Hz = counts / Execution Interval
  Maximum Error (Hz) = $\pm[1/($Execution Interval $- 50\,\mu s)]$
- Control Ports
  Hz = counts / measured time ($\pm 0.5\,\mu s$) since previous measurement
  Error if Execution Interval < 4.0 s
  Maximum Error (Hz) = $\pm(0.5\,\mu s)*($Frequency$)^2$
  Error if Execution Interval $\geq$ 4.0 s
  Maximum Error (Hz) = $\pm[1/$Execution Interval$]$

If there have been no new counts during the scan interval (since last monitored), the previous frequency is returned to the input location. If it has been 4 seconds since a new count has occurred, the 0 is returned to the input location.

TABLE 9-2 Pulse Count Configuration Codes

Code	Configuration
x0	High Frequency Pulse (8 bit counter)
x1*	Low Level AC
x2	Switch Closure
x3*	High Frequency (16 bit counter)
x4*	Low Level AC (16 bit counter)
0y	No Change
1y	Long interval counts discarded
2y	Long interval counts discarded/ Frequency (Hz) output
xy*	10 Hz reset
xy--*†	100 Hz reset

* Options Applicable to Channels P1 to P4 Only
† 100 Hz reset only used with 8 bit counter

Example: 24-- Selects 16 bit, 100 Hz reset, Low Level AC with a frequency output.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Pulse channel or Control Port number for first measurement
03:	2	Configuration code (from above table)
04:	4	Input location for first measurement
05:	FP	Multiplier
06:	FP	Offset

Input locations altered: 1 per repetition

TABLE 9-2A. Excitation Channel Options

0X Excite all reps with channel X
1X Increment channel with each rep

*** 4 EXCITE, DELAY, AND MEASURE ***

FUNCTION
This instruction is used to apply an excitation voltage, delay a specified time, and then make a single-ended voltage measurement. A 1 before the excitation channel number (1X) causes the channel to be incremented with each repetition.

Ranges 41-45 and 51-55 (Section 13.1) do not have enough time between integrations to allow a delay.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Range code (Table 9-1)
03:	2	Single-ended channel number for first measurement
04:	2	Excitation channel number (Table 9-2A)
05:	4	Delay in hundredths of a second
06:	4	Excitation voltage (millivolts)
07:	4	Input location number for first measurement
08:	FP	Multiplier
09:	FP	Offset

Input locations altered: 1 per repetition

*** 5 AC HALF BRIDGE ***

FUNCTION
This instruction is used to apply an excitation voltage to a half bridge (Figure 13.5-1), make a single-ended voltage measurement of the bridge output, reverse the excitation voltage, then repeat the measurement. The difference between the two measurements is used to calculate the resulting value which is the ratio of the measurement to the excitation voltage. A 1 before the excitation channel number (1X) causes the channel to be incremented with each repetition.

The excitation "on time" for each polarity is exactly the same to insure that ionic sensors do not polarize with repetitive measurements. The range should be selected to be a fast measurement (range 11-15), limiting the excitation on time to less than 800 microseconds at each polarity. A slow integration time should not be used with ionic sensors because of polarization error.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Range Code (Table 9-1)
03:	2	Single-ended channel number
04:	2	Excitation channel number (Table 9-2A)
05:	4	Excitation voltage (millivolts)
06:	4	Input location number for first measurement
07:	FP	Multiplier
08:	FP	Offset

Input locations altered: 1 per repetition

*** 6 FULL BRIDGE WITH SINGLE ***
DIFFERENTIAL MEASUREMENT

FUNCTION
This Instruction is used to apply an excitation voltage to a full bridge and make a differential voltage measurement of the bridge output. The measurement is made with the polarity of the excitation voltage both positive and negative (Figure 13.5-1). The result is 1000 times the ratio of the measurement to the excitation voltage. A 1 before the excitation channel

number (1X) causes the channel to be incremented with each repetition.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Range code (Table 9-1)
03:	2	Differential channel number for first measurement
04:	2	Excitation channel number (Table 9-2A)
05:	4	Excitation voltage (millivolts)
06:	4	Input location number for first measurement
07:	FP	Multiplier
08:	FP	Offset

Input locations altered: 1 per repetition

*** 7 THREE WIRE HALF BRIDGE ***

FUNCTION
This Instruction is used to determine the ratio of the sensor resistance to a known resistance using a second voltage sensing wire from the sensor to compensate for lead wire resistance.

The measurement sequence is to apply an excitation voltage, make a single-ended measurement on the first channel, then repeat the measurement with an excitation of the opposite polarity. The same sequence is then applied to the second single-ended channel. This sequence is used for offset voltage removal. The combined results for the two single-ended measurement channels are then used to calculate the resulting value, which is the ratio of the voltage across the sensor to the voltage across the reference resistor (Figure 13.5-1). A 1 before the excitation channel number (1X) causes the channel to be incremented with each repetition.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Range code for both measurements (Table 9-1)
03:	2	Single-ended channel number for first measurement
04:	2	Excitation channel (Table 9-2A)
05:	4	Excitation voltage (millivolts)
06:	4	Input location number for first measurement
07:	FP	Multiplier
08:	FP	Offset

Input locations altered: 1 per repetition

*** 8 DIFFERENTIAL VOLTAGE WITH *** EXCITATION AND DELAY

FUNCTION
This measurement consists of applying a single excitation voltage, delaying a specified time, and making a differential voltage measurement. The result stored is the voltage measured.

"Delay" (Parameter 5) refers to increasing the signal settling time by increasing the time between the start of excitation and the start of signal integration (Section 13.2). If a delay of 0 is specified, the inputs for the differential measurement are not switched for a second integration as is normally the case. With the 0 delay, Instruction 8 does not have as good resolution or common mode rejection as other differential measurements. It does provide a very rapid means of making bridge measurements. This instruction does not reverse excitation. A 1 before the excitation channel number (1X) causes the channel to be incremented with each repetition.

The 50 and 60 Hz rejection ranges, 41-45 and 51-55 (Section 13.1), do not have enough time between integrations to allow a delay.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Range code (Table 9-1)
03:	2	Differential channel number for first measurement
04:	2	Excitation channel number (Table 9-2A)
05:	4	Delay (0.01s)
06:	4	Excitation voltage (millivolts)
07:	4	Input location number for first measurement
08:	FP	Multiplier
09:	FP	Offset

Input locations altered: 1 per repetition

*** 9 FULL BRIDGE WITH EXCITATION *** COMPENSATION

FUNCTION
This instruction is used to apply an excitation voltage and make two differential voltage measurements. The measurements are made with both positive and negative excitation voltage. The measurements are made on sequential channels. The result is the voltage measured on the second channel (V_2) divided by the voltage measured on the first (V_1). If V_1 is measured on the 5 V range (code 5,15, 25 or 35 in Parameter 2), then the result is 1000 times V_2/V_1. A 1 before the excitation channel number (1X) causes the channel to be incremented with each repetition.

When used as a 6 wire full bridge (Figure 13.5-1), the connections are made so that V_1 is the measurement of the voltage drop across the full bridge, and V_2 is the measurement of the bridge output. Because the excitation voltage for a full bridge measurement is usually in the 5 V range, the output is usually 1000 V_2/V_1 or millivolts output per volt excitation.

> **NOTE:** Do not use autoranging in Parameter 2.

When used to measure a 4 wire half bridge, the connections are made so that V_1 is the voltage drop across the fixed resistor (R_f), and V_2 is the drop across the sensor (R_s). As long as V_1 is not measured on the 2.5V range, the result is V_2/V_1 which equals R_s/R_f.

SECTION 9. INPUT/OUTPUT INSTRUCTIONS

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Range code for excitation measurement (V_1) (Do not use autorange) (Table 9-1)
03:	2	Range code for bridge measurement (V_2)
04:	2	Differential channel number for first measurement
05:	2	Excitation channel number (Table 9-2A)
06:	4	Excitation voltage (millivolts)
07:	4	Input location number for first measurement
08:	FP	Multiplier
09:	FP	Offset

Input locations altered: 1 per repetition

*** 10 BATTERY VOLTAGE ***

FUNCTION
This instruction measures the CR23X input voltage and writes it to an input location. The units for voltage are volts. The measurement is .2 to .3 volts less than actual battery voltage measurement. See Section 14.2 for CR23X power requirements.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location

Input locations altered: 1

*** 11 107 THERMISTOR PROBE ***

FUNCTION
This Instruction applies a 2.5 VAC excitation voltage to Campbell Scientific's Model 107 Thermistor Probe, makes a slow single-ended voltage measurement across a resistor in series with the thermistor, and calculates the temperature in °C with a polynomial. A multiplier of 1 and an offset of zero yields temperature in degrees Celsius. The maximum polynomial error from -40°C to +56°C is given here:

Curve Fit Error --

Range (°C)	Error (°C)
-40 to +56	<±1.0
-24 to +48	<±0.1

TABLE 9-3. Excitation/Integration Codes

Code	Result
0x	excite all rep with channel x
1x	increment chan x with each rep
2x	excite all reps with channel x, slow 60 Hz rejection, 10 ms delay
3x	excite all reps with channel x, slow 50 Hz rejection, 10 ms delay
4x	increment chan x with each rep, slow 60 Hz rejection, 10 ms delay
5x	increment chan x with each rep, slow 50 Hz rejection, 10 ms delay

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Single-ended channel number of first measurement
03:	2	Excitation channel/ Integration (Table 9-3)
04:	4	Input location for first measurement
05:	FP	Multiplier
06:	FP	Offset

Input locations altered: 1 per repetition

*** 12 207 RELATIVE HUMIDITY PROBE ***

FUNCTION
This instruction applies a 3.0 VAC excitation across Campbell Scientific's Model 207 Temperature and RH Probe, makes a fast single-ended measurement across a series resistor, calculates the result with a 5th order polynomial, and performs the required temperature compensation before outputting the result in % RH.

When measuring several probes, all the RH elements should be connected sequentially. The temperature values used to correct the RH measurements should also be stored sequentially to make use of the REP feature in Instruction 11.

NOTE: The temperature value used in compensating the RH value (Parameter 5) must be obtained (see Instruction 11) prior to executing Instruction 12 and must be in Celsius.

The RH results are placed sequentially into the input locations beginning with the first RH value.

In the 207 probe, the RH and temperature elements use a common excitation line.

CAUTION: Never excite the 207 probe with DC excitation because the RH chip will be damaged.

A 1 before the excitation channel number (1X) causes the channel to be incremented with each repetition.

The maximum RH polynomial error is given here:

Curve Fit Error --

Range (%RH)	Error (%RH)
10 - 100	±4
15 - 94	±1

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	First single-ended channel for RH measurement
03:	2	Excitation channel number
04:	4	Input location for first compensating temperature measurement (°C)
05:	4	Input location for first R.H. measurement
06:	FP	Multiplier
07:	FP	Offset

Input locations altered: 1 per repetition

*** 13 THERMOCOUPLE TEMPERATURE, *** SINGLE-ENDED MEASUREMENT

FUNCTION

This instruction uses the selected thermocouple calibration to calculate the thermocouple output voltage at the reference temperature, then it makes a SINGLE-ENDED VOLTAGE MEASUREMENT (Section 13.2) on the thermocouple and adds the measured voltage to the calculated reference voltage, then converts the voltage to temperature in °C. A multiplier of 1 and an offset of zero yields temperature in degrees Celsius. When using the ±10, ±50, and ±200 mV ranges, the thermocouple inputs are tested to ensure that the thermocouple is not open.

NOTE: Because of the open thermocouple test, high output impedance sensors should not be measured with P13 or P14.

To skip the measurement portion of Instruction 13 and apply the CSI polynomials to a millivolt value in an input location, index parameter three, input channel. The input location where the thermocouple output (mV) is located is now defined by parameter three. The thermocouple must be measured with Instruction 1.

TABLE 9-4. Thermocouple Type Codes

Code	Thermocouple Type		
X1	T (copper - constantan)	X = 0	Normal Measurement
X2	E (chromel - constantan)	X = 8	TC input from A5B40 isolation
X3	K (chromel - alumel)		(uses 5 V range)
X4	J (iron - constantan)	X = 9	Output -99999 if out of common
X5	B (platinum - rhodium)		mode range (Inst. 14 only)
X6	R (platinum - rhodium)		
X7	S (platinum - rhodium)		
X8	N (nickel - chromium)		

**TABLE 9-5. Voltage and Temperature Ranges for Thermocouples
if the Reference is 20°C**

Voltage Range	Type T	Type E	Type K	Type J
±10 mV	-200 to 227	-199 to 169	-56 to 264	-150 to 205
±50 mV	-200 to 400	-240 to 675	-56 to 1372	-150 to 422
±200 mV	-----	-240 to 1000	-----	-----

Voltage Range	Type B	Type R	Type S	Type N
±10 mV	50 to 1260	0 to 970	-50 to 1045	-200 to 333
±50 mV	50 to 1700	0 to 1450	-50 to 1450	-200 to 1240

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Range code (Table 9-1)
03:	2	Single-ended channel number for first TC (when indexed (--), becomes an input location containing voltage measurement)
04:	2	Thermocouple type code (Table 9-4)
05:	4	Reference temperature location
06:	4	Destination input location
07:	FP	Multiplier
08:	FP	Offset

Input locations altered: 1 per repetition

*** 14 THERMOCOUPLE TEMPERATURE, ***
DIFFERENTIAL MEASUREMENT

FUNCTION
This instruction calculates the thermocouple temperature for the thermocouple type selected. The instruction specifies a DIFFERENTIAL VOLTAGE MEASUREMENT (Section 13.2) on the thermocouple, adds the measured voltage to the voltage calculated for the reference temperature relative to 0°C, and converts the combined voltage to temperature in °C. A multiplier of 1 and an offset of zero yields temperature in degrees Celsius.

When using the ±10, ±50, or ±200 mV ranges, the differential inputs are briefly tested to insure that the thermocouple is not open. If the

thermocouple is open, -99999 or -6999 is output.

> **NOTE:** Because of the open thermocouple test, high output impedance sensors should not be measured with P13 or P14.

Table 9-4 gives the thermocouple type codes for Parameter 4. Enter a 9 in front of the code and the CR23X will make a check on common mode range; -99999 is output for temperature if common mode range is exceeded.

To skip the measurement portion of Instruction 14 and apply the CSI polynomials to a millivolt value in an input location, index parameter three, input channel. The input location where the thermocouple output (mV) is located is now defined by parameter three. The thermocouple must be measured with Instruction 2.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Range code (Table 9-5, 9-1)
03:	2	Differential channel number for first TC (when indexed (--), becomes an input location containing voltage measurement)
04:	2	Thermocouple type code (Table 9-4)
05:	4	Reference temperature location
06:	4	Destination input location
07:	FP	Multiplier
08:	FP	Offset

Input locations altered: 1 per repetition

*** 15 CONTROL PORT SERIAL I/O ***

FUNCTION
Send and receive serial data through the CR23X control ports, see Appendix B for details on using Instruction 15.

*** 16 TEMPERATURE FROM *** PLATINUM R.T.D.

FUNCTION
This instruction uses the result of a previous RTD bridge measurement to calculate the temperature according to the DIN 43760 specification adjusted (1980) to conform to the International Electrotechnical Commission standard. The range of linearization is -200° to 850°C. The error in the linearization is less than 0.001°C between -100° and +300°C, and is less than 0.003°C between -180° and +830°C. The error (T calculated - T standard) is +0.006° at -200°C and -0.006° at +850°C. The input must be the ratio R/Ro, where R is the RTD resistance and Ro the resistance of the RTD at 0°C (Sections 7.9 and 7.10). A multiplier of 1 and an offset of zero yields temperature in degrees Celsius.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	4	Input location of R/Ro
03:	4	Input location of result
04:	FP	Multiplier
05:	FP	Offset

Input locations altered: 1 per repetition

*** 17 PANEL TEMPERATURE ***

FUNCTION
This instruction measures the temperature (°C) of a thermistor that resides under the CR23X wiring panel. This temperature is generally adequate for use as the thermocouple reference temperature. Refer to Section 13.4.1 for accuracy information.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location number for temperature

Input locations altered: 1

*** 18 MOVE TIME TO INPUT LOCATION ***

FUNCTION
This instruction takes the current time in seconds into the minute, minutes into the day, or hours into the year and does a modulo divide (see Instruction 46) on the time value with the number specified in the second parameter. The result is stored in the specified input location. Entering 0 or a number which is greater than the maximum value of the time for the second parameter will result in the actual time value being stored.

PARAMETER 1 CODES

Code	Time Units
0	Seconds into minute (maximum 60)
1	Minutes into current day (maximum 1440)
2	Hours into current year (maximum 8784)
3	Store yr,day,hr,min,sec into 5 input locations (modulo divide not used)

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Time Code
02:	4	Number to modulo divide by
03:	4	Input location number

Input locations altered: 1 or 5

*** 19 WRITE SIGNATURE TO INPUT *** LOCATION

FUNCTION
This instruction stores the signature of the Read Only Memory (ROM) and user program memory (SRAM) into an input location. The signature is a result of the CR23X PROM, the size of SRAM, and the entries in the ⁕ 1 , ⁕ 2 , ⁕ 3 , ⁕ # , ⁕ A , and ⁕ C Modes. This signature is not the same as the signatures given in the ⁕ B Mode. Recording the signature allows detection of any program change or ROM failure.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location number

Input locations altered: 1

*** 20 SET PORT ***

FUNCTION
This instruction sets or configures specified control ports (C1-C8). On power-up, ports default to input configuration (i.e., they are not driven high or low by the CR23X, and can be used to read the status of an external signal using Instruction 25). When a port is set high, low, pulsed, or toggled by this instruction or a program control command, the port is automatically configured as an output.

> **NOTE**: Do not apply voltages greater than 16 VDC to control ports. 5.0 VDC is preferred.

Ports can also be set using the ⌑ 6 Mode or the J and K telecommunications commands. However, the ports MUST be configured as outputs before these means of setting them will work. The option to configure the port as an output is used when a port must be configured as an output without changing the state of the port.

Pulse duration, initiated by a program control instruction, can be set for each control port (Table 12-2). Instruction 20 does not pulse the port, it only sets the duration. If Instruction 20 is not used to set the duration, the pulse command will result in a 10 ms pulse.

Instruction 20 has two 4 digit parameters. Each digit represents one control port. The code (0-9) entered as the digit determines what effect command 20 has on the corresponding port.

TABLE 9-6. Port Configuration Option Codes

Code	Function
0	Set port low
1	Set port high
2	Toggle port
3	Pulse duration 1 ms
4	Pulse duration 10 ms
5	Pulse duration 100 ms
6	Pulse duration 1 s
7	Configure as output
8	Configure as input
9	Leave unchanged

Duration of pulse on subsequent pulse port command in Program Control Instruction.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	C8, C7, C6, C5 option codes
02:	4	C4, C3, C2, C1 option codes

Input locations altered: 0

*** 21 PULSE PORT WITH DURATION ***

FUNCTION
Instruction 21 pulses a control port for a specified amount of time in hundredths of seconds (0.01 seconds).

The pulse works as a toggle; if the port is high before the instruction is executed, it will pulse low and vice versa. Any value less than 1, including 0, gives a pulse of 10 milliseconds. The maximum input value is limited to 65,000, which gives a pulse length of 650 sec.

Parameter 1 is the port number to be pulsed. Parameter 2 is the input location containing the pulse length.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Control port
02:	4	Input location of pulse length in hundredths of a second

Input location altered: 0
Input locations read: 1

*** 22 DELAY WITH OPTIONAL *** EXCITATION

FUNCTION
This instruction is used to delay program execution. The excitation on time (parameter 2) should be set to zero and the off time delay (parameter 3) set to the desired delay. This instruction can also be used in conjunction with others for measuring a response to a timed excitation using the excitation outputs. It sets the selected excitation output to a specific value, waits for the specified time, then turns off the excitation and waits an additional specified time before continuing execution of the following instruction. Analog power is turned off during delay after excitation to drop power to 3 mA.

If the excitation channel is indexed, parameter 4 becomes an input location. The excitation voltage must be loaded into the specified input location before Instruction 22 is executed.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Excitation channel number (Table 9-2A)
02:	4	Delay time in hundredths of a second that excitation is on
03:	4	Delay time in hundredths of a second after excitation is turned off
04:	4	Excitation voltage in millivolts/Input location

Input locations altered: 0

*** 23 BURST MEASUREMENT ***

FUNCTION
Instruction 23 will repeatedly make a voltage measurement on a series of single-ended or differential channels, applying excitation if desired. The measurement units are millivolts prior to scaling. The measurements saved can be those made immediately upon execution of the instruction or grouped around a specified trigger condition. The results of the measurements may be stored in Input Storage or the raw A/D data can be sent out the serial port. The minimum sample interval per channel is .667 ms (i.e., one channel can be sampled at a maximum rate of 1.5 kHz).

> **CAUTION:** Measurement accuracy specifications will not be met on the ±10 mV input range when the sample interval per channel is set lower than 1 ms because of insufficient settling time.

MEASUREMENT
The voltage measurement uses the fast integration time of 250 µs. Differential measurements are made with a single integration. The input settling time before an integration is 370 µs for intervals greater than or equal to 1 ms per channel and 100 µs for intervals less than 1 ms per channel. The short 100 µs input settling time was chosen to allow burst measurements up to 1.5 kHz, at the expense of adequate input settling time on the ±10 mV input range. **The ±10 mV input range will not meet the specified measurement accuracy for intervals less than 1 ms per channel.** Settling errors on the order of ±100 µV were measured with a 1 kΩ source impedance when using less than 1 ms per channel in the burst mode. Settling errors are worsened for sensors with long leads and/or high output impedance. See Section 13.3 for discussion of settling time and time constants.

Excitation is always supplied from excitation channel 1. The excitation voltage in millivolts is entered in Parameter 9. If excitation is not desired, enter 0 for Parameter 9.

Three options are available for the first digit in Parameter 4.

0 - Trigger on first measurement channel.
1 - Trigger on Digital Control Port 1.
2 - Trigger on first measurement channel, set Digital Control Port high when trigger is met and low when measurements have finished.

When triggering on options 0 or 2, the measurement on the first specified channel (Parameter 3) is compared to the limit specified in Parameter 8. The user's multiplier and offset are not applied before the comparison: the limit must be entered in units of millivolts. If a digital trigger (low < 1.5V, 3.5 V < HIGH < 5 V) is used, it must be input to Digital Control Port #1. Option 2 is useful when 2 or more CR23X's are required to start "Bursting" at the same time. For example, Digital Control Port #1 of all the CR23X's involved are connected in parallel. The master CR23X sets its Digital Control Port #1 high when the trigger condition is met. The remaining dataloggers, programmed with option 1, trigger on the digital signal. The master CR23X sets Port #1 low when its measurements are completed.

Five trigger options are available for the second digit in Parameter 4: **0** trigger immediately, start saving or sending data with the first measurement made; **1** trigger if the measurement is greater than the limit entered or if the digital trigger is high; **2** trigger if the measurement is less than the limit or if the digital trigger is low; **3** trigger on rising edge; i.e., when the measurement goes from below to above the limit or when the digital trigger goes from low to high; **4** trigger on falling edge; i.e., trigger when measurement goes from above the limit to below it or when the digital trigger goes from high to low. When triggering on the rising or falling edge, the input must make the

specified transition to trigger. For example, when triggering on the rising edge, if the input starts out high, it must go low and then high again to trigger.

DATA SENT TO INPUT STORAGE
When the measurements are sent to Input Storage, Parameter 6 is used to specify the number of scans made on the channels being measured (the CR23X multiplies the number entered by 1000). The measurements from each channel are stored contiguously. For example: Parameter 1 specifies 4 channels are to be measured, Parameter 6 specifies 250 scans (0.250 entered), and Parameter 10 specifies 1 as the first input location to store data. The measurements from the first channel will be stored in input locations 1-250, those from the second channel in locations 251-500, etc. If insufficient locations are allocated to Input Storage ($\boxed{*}$ $\boxed{A}$) to accommodate the number of locations called for by Parameter 6 multiplied by Parameter 1 (e.g., 250 x 4 = 1000), an error code, E 60, will be displayed when the program is compiled.

The number of scans determines how many samples will be saved, and hence, when Instruction 23 will be completed and execution will pass on to the next instruction. Measurements before and/or after the trigger can be saved. Parameter 7 determines how many scans that occurred prior to the trigger are saved. For example, if 250 scans are specified, and an offset of 20 is entered for Parameter 7, then the trigger measurement will be stored in location 21. Locations 1-20 will contain the measurements that preceded the trigger and 22-250 will contain the measurements following the trigger. If only 10 measurements are made before the trigger, then they will be stored in locations 11-20 while the value -99999. will be stored in locations 1-10 for which no measurements were made.

DATA SENT TO SERIAL PORT
When the raw A/D data is sent out the serial port, the measurement data is not buffered and hence, only the trigger and subsequent measurements can be sent. The number of measurements is determined by Parameters 1 and 6. Because the total number of measurements are limited only by the storage of the receiving computer, this can be a very large number. Parameter 6 is the number of measurements per channel to send in units of

1000 (e.g., 0.001 represents 1 measurement). If 0 is entered for Parameter 6, the CR23X will continue to send data until the Instruction is aborted by pressing any key on the keyboard.

Raw A/D data may be sent to a maximum of 8 SM192 or SM716 Solid State Storage Modules (SM's) at 76.8K baud (Parameter 4, Destination Option C = 3). All SM's connected should be set to Fill and Stop and have consecutive addresses. The Burst data will be sent to the first available (lowest address) Storage Module, followed by the next lowest addressed, and so on. If the data is retrieved from the SM's using SMCOM.COM, a Storage Module communications program contained in the PC208 Datalogger Support Software, the data collection format must be "A as stored 8 bit". This will transfer the raw A/D data to a computer file. The raw A/D data can be converted to ASCII using SPLIT, a general purpose data reduction program also contained in PC208.

If SPLIT is not available for converting the raw A/D, the following A/D format information is provided for decoding purposes. At the start of the series of measurements, the CR23X makes a self-calibration measurement. The calibration data is sent at the start of the measurement data. The serial data is sent as a series of signed 2 byte integers (most significant byte sent first; i.e., Integer = 256 $*$ byte 1 + byte 2): I1...In. The first integer, I1 is a start of output identifier, FCxx (hex), where the first byte is always FC (never seen in the data), and the second byte is a number less than 100 (decimal, 64 hex), which is the Instruction Location Number of Instruction 23 in the program table. I2 divided by I3 is the multiplier and I4 the offset (to the raw data) determined by the first calibration. I2 is a fixed value determined by the input range selected. I5 through In are the raw measurement data. Thus, the value of the first measurement sent (M1) in millivolts is:

$$M_1 = I_2/I_3 (I_5 - I_4)$$

The measurement data are sent in the order that the measurements are made (i.e., the first measurement for each channel, then the second measurement for each channel, etc.).

> **NOTE**: When the raw serial data option is selected, the calibration values are for conversion to millivolts only. Parameters 11 and 12 are ignored.

SCAN INTERVAL
Instruction 23 has its own scan interval independent of the execution interval of the program table in which it resides. The resolution of the clock timing the execution interval is 407 nanoseconds. This scan interval, entered in Parameter 5 (in milliseconds), is the time between each scan of the specified channels (i.e., if 4 channels are specified in Parameter 1, and the scan interval is 5 ms, then the 4 measurements will be repeated every 5 ms). The minimum time that is allowed per measurement is 0.6666 ms. The maximum time that is allowed per measurement is 25 ms. If the scan interval entered does not allow this much time per measurement (e.g., if with 4 reps, an interval less than 2.666 ms is entered), an error code, E 61, will be displayed when the program is compiled. When sending data to the serial port, the rate at which the data can be transferred may limit the scan interval (e.g., at 9600 baud the minimum time per measurement is 2.2 ms).

Burst/Telecommunications Considerations

If a Burst measurement sequence is in progress, raising the datalogger's ring line will abort the Burst sequence. Peripherals which raise the ring line are modems (i.e., RF, Telephone, Short Haul, MD9, SC32A).

If the Burst Measurement Instruction is encountered while telecommunications is in progress, the destination of the data determines whether or not the instruction is executed:

Burst data sent to input locations

If a CR23X already in Telecommunications mode executes a Burst instruction specifying that Burst data be sent to input locations, all telecommunication activity will be suspended. After the Burst trigger condition is met and all Burst measurements made, telecommunications activity can resume.

Burst data sent to CS I/O or Computer RS-232 Port

If the Burst instruction specifies that Burst data be sent to the serial port (i.e., Storage Module),

CR23X program execution will pause until the Telecommunication mode is exited. During this pause telecommunications (i.e., view input locations in the Monitor Mode, etc.) can continue. No Burst measurements are made while in telecommunications and no Burst data is sent to the serial port. After telecommunications has ended, datalogger program execution will resume as if the Burst instruction were just executed.

> **NOTE:** Instruction 23 can be aborted by pressing any key on the CR23X Keyboard.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Input channels per scan (reps)
02:	2	Range code
03:	2	Single-ended or differential channel for first analog measurements
04:	4	Option, 4 digit code ABCD

    A Trigger
       0 - Trigger on 1st analog channel
       1 - Digital trigger on Control Port #1
       2 Same as 0, but set Digital Control Port #1 high when trigger is met, low when done measuring

    B Trigger option
       0 - Trigger immediately
       1 - Trigger if above limit (high)
       2 - Trigger if below limit (low)
       3 - Trigger on rising edge
       4 - Trigger on falling edge

    C Destination
       0 - Input Storage
       1 - CS I/O port 9600 baud
       2 - CS I/O port 76,800 baud

3 - CS I/O port
76,800 baud to
SM192/716
5 - CS I/O port
38.4 K
6 - RS-232 port
38.4 K

D   Measurement
0 - Differential
measurement
1 - Single-ended
measurement

05:	FP	Scan interval (ms, minimum 0.6666 x reps, limited to 25 ms)
06:	FP	Number of scans (units of 1000)
07:	4	Number of samples saved before trigger (not used with serial output)
08:	FP	Trigger limit (mV, unscaled measurement)
09:	4	Excitation voltage (mV)
10:	4	1st input location in which to store data
11:	FP	Multiplier (not used with serial output)
12:	FP	Offset (not used with serial output)

Input locations altered: number of scans plus number of samples saved before trigger

*** 24 CALIBRATION ***

Calibrate and put calibration values into input locations if desired. Calibration takes place only when Instruction 24 is executed. Automatic calibration is disabled if Instruction 24 is in a program table. See Section 13.7 for help on when to use Instruction 24.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Option (0 for most applications)
02:	4	Beginning input location (Option 0 does not alter any input locations)

Input Locations Used: 0 to 31 depending on option and what is being used in the program.

TABLE 9-7

Option	Description
0	Calibrate only the active components; do not store the results
1	Calibrate only the active components; store the active components
2	Calibrate all components; store the results
3	Do not calibrate; store active components

*** 25 READ PORTS ***

FUNCTION
The status of a group of ports selected by a mask is read and placed in an input location. The status is a base 2 representation of the ports converted to base 10. Port 1 is the least significant bit. For example, if all ports are read, and the port status is as follows:

PORT	C8	C7	C6	C5	C4	C3	C2	C1
VALUE	128	64	32	16	8	4	2	1
STATUS (0=low, 1=high)	0	0	1	1	0	0	1	0

Base 10 equivalent: 32 + 16 + 2 = 50

50 will be stored in the input location.

The mask is also base 2 representation; 1 indicates the port is to be read, 0 results in a 0 for the port regardless of the status of the port (AND operation). For example, if 50 (see above example) is entered for the mask, ports 2, 5, and 6 are read. If only ports 4 and 5 are high, the status will be 16 (port 4 is not read).

> **NOTE**: Voltages in excess of 16 volts applied to a control port can cause the CR23X to malfunction.

PARAM. NUM.	DATA TYPE	DESCRIPTION
01:	4	MASK (0-255)
02:	4	INPUT LOCATION TO STORE RESULT

Input locations altered: 1

*** 26 TIMER ***

FUNCTION

This instruction will reset a timer or store the elapsed time registered by the timer in seconds in an Input Storage location. Instruction 26 can be used with Program Control Instructions to measure the elapsed time between specific input conditions. There is only one timer and it is common to all tables (e.g., if the timer is reset in Table 1 and later in Table 2, a subsequent instruction in Table 1 to read the timer will store the elapsed time since the timer was reset in Table 2).

Elapsed time is tracked in 0.1* second increments. The maximum interval that can be timed is 46.6 days.

The timer is also reset in response to certain keyboard entries:

1. When tables are changed and compiled with the ⁎ 0 Mode, the timer is reset automatically.
2. Entering "*6" after changing the program compiles the programs, but does NOT reset the timer.

PARAM. NUM.	DATA TYPE	DESCRIPTION
01:	4	Input location no. of elapsed time in seconds (or enter 0 to reset)

Input locations altered: 1
(0 if timer is being reset)

*** 27 PERIOD AVERAGE (SE) ***

FUNCTION

Instruction 27 measures the period (microseconds) of a signal on a single-ended input channel. As an option, the frequency of the signal in kHz may be output instead of the period.

The number of cycles to measure should be chosen so that at least 1.1 milliseconds transpires while counting those cycles (e.g., if the maximum input frequency is 10 kHz, choose at least 11 cycles to be measured). If the time for the number of cycles is less than 1.1 milliseconds, an overrange value (displayed as -99999) will be stored for the measurement. The specified number of cycles are timed with a resolution of 12 nanoseconds, making the resolution on the period 12 nanoseconds divided by the number of cycles measured.

The "Time out" parameter specifies the maximum length of time the instruction will wait on each repetition for the specified number of cycles. If the cycles have not been counted within this time, -99999 will be loaded into the input location.

TABLE 9-8. Period Average Output Options/Range Codes

Range Code	Voltage Gain	Minimum Signal (pk-pk)[1]	Maximum Signal (pk-pk)[1]	Minimum Pulse Width	Maximum Frequency[2]
x4	1.0	500 mV	10.0 V	2.5 µs	200 kHz
x3	5.0	40 mV	2.0 V	10 µs	50 kHz
x2	20	5 mV	2.0 V	62 µs	8 kHz
x1	100	2 mV	2.0 V	100 µs	5 kHz

x = 0 Output Period in µs   x = 1 Output frequency in Hz

1 - Signals must be centered around datalogger ground to trigger zero-crossing detector.
2 - Maximum frequency equals 1/(Twice Minimum Pulse Width) for 50% duty cycle signals.

SECTION 9. INPUT/OUTPUT INSTRUCTIONS
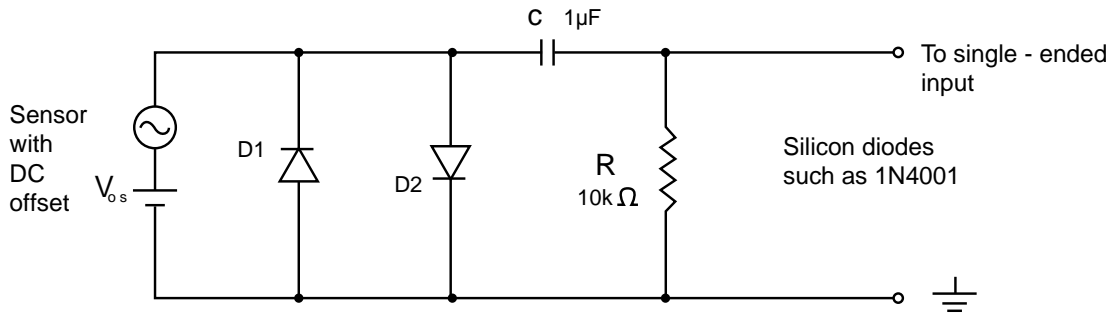
Figure 9-2. Recommended input conditioning circuit for period averaging.

An internal gain stage is used to amplify low-level ac signals prior to a zero-crossing detector for the period averaging measurement. The minimum pulse width requirements increase (maximum frequency decreases) with increasing gain as shown in Table 9-8. Signals larger than the specified maximum for a range will saturate the gain stage and prevent operation up to the maximum specified frequency. A recommended input conditioning circuit for period averaging measurements is shown in Figure 9-2.

> **CAUTION:** Noisy signals with slow transitions through the zero voltage threshold are problematic for period measurements, because of the potential for extraneous counts and the zero crossing point. A zero-crossing detector with 16 mV of hysteresis follows the voltage gain stages. The effective input referred hysteresis equals 16 mV divided by the selected voltage gain. The effective input referred hysteresis for range code X1 is only 0.16 mV. Consequently, 0.16 mV of noise on the input signal could cause extraneous counts for range code X1. For best results, select the smallest gain that will meet the minimum input signal requirements.

In Figure 9-2, the capacitor C is a dc blocking capacitor for offset voltage removal. Resistor R1 is used to bias the datalogger side of the input circuit to ground. Low-level sensor outputs can have dc offset voltages that prevent the internal voltage comparator from ever crossing its switch point. The reactance of the dc blocking capacitor ($Xc = (2 \pi f C)^{-1}$) and resistor R1 form a voltage divider at low frequencies ($R1/(R1 + Xc)$) that attenuates the applied input signal. This attenuation sets a lower limit on low-frequency operation and the minimum size of R1. The circuit shown in Figure 9-2 attenuates the input signal by a factor of 2 at 16 Hz.

The back-to-back silicon diodes D1 and D2 shown in Figure 9-2 provide ESD protection for capacitor C and the sensor, and also limit large amplitude sensor signals. These diodes clip large amplitude signals to approximately 1.4 V pk-pk, which is within the recommended input signal ranges for all range codes. Diodes D1 and D2 along with resistor R1 are recommended to limit large amplitude sensor signals, even when dc blocking capacitor C is not used. Sensors outputting large voltages may cause large currents to flow through these back-to-back diodes. A current limiting resistor may be desirable to minimize these currents in some situations.

The current flow through these clipping diodes may also induce single-ended offset voltages if it returns into the ÷ ground terminals because of an alternate return path (ground loop) back to the sensor. Single-ended offset voltages of up to 1 µV/mA of current that flows into the ÷ ground terminals can be induced across the front panel. The back-to-back diodes can be tied into the G ground terminals, rather than ÷ ground terminals, if this is a problem.

PARAM. NUM.	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Output option/Range code (Table 9-8)
03:	2	Single-ended input channel

04:	4	# Cycles to measure
05:	4	Time out (0.01 sec, at least the maximum duration of the number of cycles specified + 1 1/2 cycles.)
06:	4	Destination input location
07:	FP	Multiplier
08:	FP	Offset

Input locations altered: 1 per repetition

*** 28 VIBRATING WIRE *** MEASUREMENT

FUNCTION
Excites a vibrating wire sensor with a swept frequency (from low frequency to high), then measures the response period and calculates $1/T^2$, where T is the period in ms. Excitation is normally provided before each repetition. As an option, a single excitation can be made prior to all repetitions of the measurement. An AVW1 or AVW4 Vibrating Wire Interface is usually required for these sensors.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions Hit C (--) to skip repeat of excitation
02:	2	Single-ended channel for first measurement
03:	2	Excitation Channel
04:	2	Start frequency of sweep (100'S of Hz)
05:	2	End frequency of sweep (100'S of Hz)
06:	4	# Cycles to measure (0 means none)
07:	4	Delay before excitation applied (0.01 sec units)
08:	4	Input location ($1/T^2$), T in ms
09:	FP	Multiplier
10:	FP	Offset

Input locations altered: 1 per repetition

*** 29 INW PS9105E ***

FUNCTION
The Instrumentation Northwest PS9105 Pressure Transducer is used to measure water level. This instruction excites the PS9105 with

a single excitation channel and measures the sensor's output on two consecutive differential analog input channels. The pressure and sensor temperature are written into two consecutive input locations starting at the location specified in parameter 3.

Each PS9105 has 20 enhanced measurement parameters that are documented on the calibration sheet. Enter these parameters into Instruction 29.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Differential channel number for first measurement
02:	2	Excitation channel number
03:	4	Input location number
04:	FP	Pressure Range (psig) (enh. par. 1)
05:	FP	Enhanced Parameter 2
06:	FP	Enhanced Parameter 3
07:	FP	Enhanced Parameter 4
08:	FP	Enhanced Parameter 5
09:	FP	Enhanced Parameter 6
10:	FP	Enhanced Parameter 7
11:	FP	Enhanced Parameter 8
12:	FP	Enhanced Parameter 9
13:	FP	Enhanced Parameter 10
14:	FP	Enhanced Parameter 11
15:	FP	Enhanced Parameter 12
16:	FP	Enhanced Parameter 13
17:	FP	Enhanced Parameter 14
18:	FP	Enhanced Parameter 15
19:	FP	Enhanced Parameter 16
20:	FP	Enhanced Parameter 17
21:	FP	Enhanced Parameter 18
22:	FP	Enhanced Parameter 19
23:	FP	Enhanced Parameter 20

Input location altered: 2

*** 100 SDM TDR ***

FUNCTION
Instruction 100 is used to control and measure the CSI TDR Soil Moisture Measurement System with control ports C1, C2, and C3. See the TDR manual for information on Instruction 100.

SECTION 9. INPUT/OUTPUT INSTRUCTIONS

*** 101 SDM-INT8 ***

FUNCTION

The 8 channel Interval Timer (INT8) is a measurement module which provides processed timing information to the datalogger. Each of the 8 input channels may be independently configured to detect either rising or falling edges of either a low level AC signal or a 5 V logic signal. Each channel may be independently programmed. See the SDM-INT8 manual for detailed instructions and examples.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Address, Base 4 (00..33)
02:	4	*Input config; channels 8,7,6,5
03:	4	*Input config; channels 4,3,2,1
04:	4	**Function; channels 8,7,6,5
05:	4	**Function; channels 4,3,2,1
06:	4	***Averaging option
07:	4	Loc
08:	FP	Mult
09:	FP	Offset

Execution Time:	2.3 ms + 1.65 ms/value + averaging interval if used

Intermediate locations required: 1

Input locations altered: 1 per function

* Input configurations:
0 =	5V logic level, rising edge
1 =	5V logic level, falling edge
2 =	low level ac, rising edge
3 =	low level ac, falling edge

** Function:
0 =	none
1 =	period in ms
2 =	frequency in kHz
3 =	time since previous channel's edge in ms
4 =	time since channel 1 in ms
5 =	counts on channel 2 since channel 1, linear interpolation
6 =	frequency in kHz (low resolution)
7 =	counts
8 =	counts on Channel 2 since Channel 1, no interpolation

*** Averaging option:
0	Average over execution interval
0--	Continuous averaging
XXXX	Averaging interval in ms, XXXX>0
XXXX-	Capture all events until XXXX edges of channel 1 (0<XXXX, 9999)
9999-	Test memory

*** 102 SDM-SW8A ***

FUNCTION

The 8 channel SDM-SW8A Switch Closure Input Module is a peripheral for measuring up to 8 channels of switch closure or voltage pulse inputs. Each channel may be configured to read single-pole double-throw (SPDT) switch closure, single-pole single-throw (SPST) switch closure, or voltage pulse. Output options include counts, duty cycle, or state.

The SW8A is addressed by the datalogger, allowing multiple SW8A's to be connected to one datalogger. 16 addresses are available, but for most applications Campbell Scientific recommends no more than 4 SW8A's be connected to one datalogger.

If more channels are requested than exist in one module, the datalogger automatically increments the address and continues to the next SW8A. The address settings for multiple SW8A's must sequentially increase. For example, assume 2 SW8A's with an address of 22 and 23 are connected, and 12 Reps are requested. Eight channels from the first SW8A and the first 4 channels from the second SW8A will be read.

Only one Function Option (Parameter 3) may be specified per P102 Instruction. If all 4 functions are desired, 4 P102 Instructions are required.

Function Option 0 provides the state of the signal at the time P102 is executed. A 1 or 0 corresponds to high or low states, respectively.

Function Option 1 provides signal duty cycle. The result is the percentage of time the signal is high during the sample interval.

Function Option 2 provides a count of the number of positive transitions of the signal.

Function Option 3 provides the signature of the SW8A PROM. A positive number (signature) indicates the PROM and RAM are good, a zero (0) indicates bad PROM, and a negative number

indicates bad RAM. Function Option 3 is not used routinely, but is helpful in "debugging". Only one Rep is required for Option 3.

Parameter 4 specifies the first SW8A channel to be read (1..8). One or more sequential channels are read depending on the Reps. To optimize program efficiency, the sensors should be wired sequentially.

Data are stored in sequential input locations, starting at the location specified in Parameter 5. The number of input locations consumed is equal to the number of Reps.

The scaling multiplier and offset (Parameters 6 and 7) are applied to all readings. If a multiplier is not entered, all readings are set to 0.

If the SW8A does not respond, -99999 will be loaded into input locations. Modules which do not respond when addressed by the datalogger are possibly wired or addressed incorrectly. Verify the address specified in Parameter 2 corresponds to the jumper setting and all connections are correct and secure. See the SDM-SW8 Manual for examples.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Reps (# of channels)
02:	2	Starting Address, Base 4 (00..33)
03:	2	Function Option (0=State, 1=Duty, 2=Counts, 3=Signature)
04:	2	SDM-SW8A Starting Channel (1..8)
05:	4	Starting input location
06:	FP	Mult
07:	FP	Offset

Input locations altered: 1 per channel

*** 103 SDM-AO4 ***

FUNCTION
Instruction 103 is used to activate a SDM-AO4 connected to ports C1, C2, and C3.

There are 4 analog voltage outputs per SDM-AO4. The output voltages in millivolts must be stored in 4 adjacent input locations starting with the location entered in parameter 3. Four reps are required for each SDM-A04. Every 4 reps another device at the next higher address is selected.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01	2	Reps (# of outputs)
02	2	Starting Address, Base 4 (00..33)
03	4	Starting input location

C1 is Data line
C2 is Clk/Hand Shake line
C3 is SDE (Enable) line

Input locations reads: 1 per repetitions

*** 104 SDM-CD16AC ***

FUNCTION
Instruction 104 is used to control the SDM-CD16AC or the SDM-CD16. The SDM-CD16AC Control Port Expansion Module has 16 digital control ports with drivers. Each port can be controlled by a datalogger or controlled manually with an override toggle switch. Each port can be thought of as a switch to ground; closed when active, open when inactive. The primary function is to activate AC powered external relays, solenoids, or resistive loads under datalogger control.

The SDM-CD16AC is a synchronously addressed datalogger peripheral. Datalogger control ports 1, 2 and 3 are used to address the SDM-CD16AC then clock out the desired state of each of the 16 control ports. Up to 16 SDM-CD16AC's may be addressed, making it possible to control a maximum of 256 ports from the first three datalogger control ports.

For each Rep, the 16 ports of the addressed SDM-CD16AC are sent according to 16 sequential input locations starting at the input location specified in parameter 3. Any non-zero value stored in an input location activates (connects to ground) the associated SDM-CD16AC port. A value of zero (0) deactivates the port (open circuit). For example, assuming 2 Reps and a starting input location of 33, OUTPUT 1 through 16 of the first SDM-CD16AC are set according to input locations 33 through 48, and OUTPUT 1 through 16 of the second SDM-CD16AC are set according to input locations 49 through 64. See the SDM-CD16AC manual for detailed instructions and examples.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Reps (# of CD16AC modules sequentially addressed)
02:	2	Starting Address, Base 4 (00..33)
03:	4	Starting input location

Input locations read: 16 per repetition

*** 105 SDI-12 RECORDER ***

> **NOTE:** Version 1.2 of the SDI-12 specification has been implemented in the CR23X. Features added in this version are marked with a dagger (†) and may not be supported by older SDI-12 sensors.

Instruction 105 allows data to be collected from SDI-12 sensors. The sensor's SDI-12 data line is connected to a control port (C5-C8). The SDI-12 ground should be connected to a "G" terminal on the CR23X wiring panel. SDI-12 power may be connected to 12V.

If multiple SDI-12 sensors are used, up to ten sensors may be connected to a single control port, but each sensor must have a unique address and requires a separate Instruction 105.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Address (0-9, †10-126)
02:	2	Command (Table 9-8)
03:	2	Control Port (C5-C8)
04:	4	Input Loc.
05:	FP	Mult
06:	FP	Offset

Input locations altered: 1-9 (†1-99), depending on the SDI-12 sensor

Intermediate locations required: 21

PARAMETER 1. ADDRESS

Enter the address of the SDI-12 sensor (0-9). †Extended addresses (10-126) may be used by entering the decimal equivalent for the appropriate ASCII character (see Appendix E).

For example, address 'A' would be entered as 65.

PARAMETER 2. COMMAND

Enter a number to select the command to be sent to the SDI-12 sensor. Usually 0 is entered to select the M command. The following Standard SDI-12 commands are supported by the CR23X:

TABLE 9-9. SDI-12 Command Codes

ENTRY	COMMAND	DESCRIPTION
0	M	Initiate Measurement
0 --	C	†Initiate Concurrent Measurement
1..9	M1..M9	Additional Measurement commands specified by the SDI-12 sensor
10	V	Initiate Verify sequence
11	I	Send Identification

Command 0: the CR23X will issue the 'M' SDI-12 measurement command and wait for the sensor to complete its measurement before requesting the data and proceeding to the next instruction in the program table. If Instruction 105 is placed in Table 1, program execution will be suspended during this delay. If it is placed in Table 2, instructions in Table 1 may be executed during this delay.

†Command 0--: this command enables concurrent measurements with SDI-12 sensors that support this feature. With concurrent measurements, the CR23X can initiate measurements with multiple SDI-12 sensors without having to wait for each individual sensor to complete its sequence before proceeding to the next sensor.   The CR23X will issue the 'C' SDI-12 concurrent measurement command and wait for the sensor response, which includes the amount of time in seconds it will take for the sensor to make the measurement. The CR23X will not wait for the data: rather, it will continue executing the table. The next time the instruction is executed, the CR23X will check the elapsed time. If the elapsed time is equal to or greater than that given by the sensor, the CR23X will get the data from the SDI-12 sensor. In the following

execution of the instruction, the CR23X will again issue the 'C' command.

The results of an M, C, M1-M9, or V command sequence is numerical data, stored in input location(s). The response to the I command is text information, which is written directly to Final Storage regardless of the Output Flag's state. The ⁎ 7 mode of the CR23X cannot be used to view text data.

†In addition to the Standard SDI-12 commands, the CR23X can issue 'Extended' SDI-12 commands. Instruction 68, Extended Parameters 4 Digit, is used to supply the characters and values to be transmitted. Multiple Instructions 68 can be used sequentially to extend the command string. The

Command parameter 2 in Instruction 105 must be 0, not 0--. Parameter 4 in Instruction 105 should refer to the first Input Location, if any, to be sent as part of the command string. The parameters placed in P68 are the decimal ASCII equivalent of literal characters to be sent, or '128' if the value in an input location is to be sent. Enter a parameter of zero to end the string. The address prefix and '!' suffix are automatically sent in addition to the information listed in Instruction(s) 68. The CR23X keeps no data from the sensor response.

Example: To send the command 1A0+2.3456-87.654! where 1 is the SDI-12 sensor address, +2.3456 is the value in Input Location 5 and -87.654 is the value in Input Location 6, use the following instructions:

```
01:   SDI-12 Recorder (P105)
        1:      1         SDI-12 Address
        2:      0         Start Measurement (aM0!)
        3:      5         Port
        4:      5         Loc [ SendVal_1 ]
        5:      1         Mult
        6:      0         Offset

02:   Extended Parameters 4 Digit (P68)
        1:      65        Option            ;ASCII character A
        2:      48        Option            ;ASCII character 0
        3:      128       Option            ;Send first value
        4:      128       Option            ;Send second value
        5:      0         Option            ;End of command string
        6:      0         Option
        7:      0         Option
        8:      0         Option
```

PARAMETER 3. PORT

Enter the CR23X control port (C5-C8) connected to the SDI-12 sensor data line. The default port is C8.

PARAMETER 4. INPUT LOCATION

Input location where the returned data is stored. If multiple values are returned from the SDI-12 sensor they are stored in sequential input locations beginning at the specified location.

ERRORS

If the CR23X receives either an incorrect response or no response from an SDI-12 sensor, the CR23X will retry the operation. If after retries a valid response has not been received, the CR23X will store a -99999 in the input location specified in Parameter 4. Only the first location will be altered. Sequential locations will contain values from previous measurements.

TRANSPARENT MODE

The SDI-12 transparent mode is used to communicate directly with an SDI-12 sensor. A common application of the transparent mode is to verify proper operation of the SDI-12 sensor.

A computer or terminal is required to use the transparent mode; the keyboard cannot be used. Transparent mode is entered while the computer is in telecommunications with the SDI-12 recorder CR23X (at the asterisk '∗' prompt). Enter 'pX' at the asterisk prompt, where 'p' is the Control Port number (5-8) attached to the SDI-12 data line. The CR23X responds with 'entering SDI-12.' Any SDI-12 command preceded with the sensor address and followed with an exclamation point '!' may then be entered. For example, entering '0I!' would request identification from an SDI-12 sensor addressed at 0.

The SDI-12 prompt will not appear until the CR23X finishes executing all program tables. While in transparent mode, scheduled tables in the CR23X will not execute. Transparent mode ends and the '*' prompt is returned when an invalid SDI-12 command (e.g., a blank line) is entered, if the SDI-12 sensor doesn't respond within the time-out period following a valid command (approximately 1/3 second), or if the user does not enter a command before the mode times out (approximately 35 seconds). Security must be unlocked to level 2 before the Transparent mode is enabled.

*** 106 SDI-12 SENSOR ***

Instruction 106 allows a CR23X to be used as an SDI-12 sensor. The CR23X can make measurements and transfer data using SDI-12 commands in response to another SDI-12 recorder.

Instruction 106 supports the Standard SDI-12 commands as listed in the Parameter 2 description for Instruction 105.

The SDI-12 data line is attached to Control Port 8 and Instruction 106 must be the first instruction in Subroutine 98 located in Table 3. An SDI-12 recorder addresses the SDI-12 sensor CR23X by sending a Break and the sensor's address. The sensor CR23X will call subroutine 98 whenever it detects activity on the SDI-12 data line attached to Port 8, but if the break and the specified address are not received by Instruction 106, the remainder of the subroutine is not executed.

Two programming techniques exist for obtaining measurement values to be transferred by the sensor Instruction 106. The first technique makes the requested measurements "on demand" in response to the recorders request. The measurement instructions are located in Subroutine 98 and are executed only when the SDI-12 recorder requests measurements. This technique is preferred when measurements are to be made at the recorder's command.

The second technique transfers measurement values previously obtained by instructions in Table 1 or Table 2. Subroutine 98 contains only Instructions 106 (SDI-12 sensor) and 95 (End). When the recorder requests measurements, values already in the specified input locations are used. The advantage of this technique is that the sensor CR23X can be making and storing measurements independent of the SDI-12 recorder. The data is also returned slightly faster since the sensor CR23X does not make measurements when the recorder requests data, but rather uses measurements made at the last regular table execution.

These two techniques can be combined allowing the sensor CR23X to function as an SDI-12 sensor and to make independent measurements. While Subroutine 98 is being executed, normal Table 1 or 2 execution scheduling may be altered or missed since Subroutine 98 is not interrupted.

This is likely to occur if Subroutine 98 execution takes longer than the scan interval programmed for Table 1 or 2. It is also possible for instructions in Table 1 or 2 to prevent Subroutine 98 from being called in time for Instruction 106 to receive the address information from the recorder. This is likely to occur only if Table 1 or 2 is executed often and has instructions that take longer than 1/3 second to execute. For example, Instruction 4 (Excite-Delay-SE) with a 1/2 second delay could cause Subroutine 98 to miss the SDI-12 address information if it were executing when the SDI-12 data line became active. If this occurs the sensor CR23X will not respond to the SDI-12 recorder. Most instructions execute fast enough that when Instruction 106 misses the initial SDI-12 address, a subsequent retry by the recorder will work.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	ADDRESS (0-9, †10-126)
02:	4	TIME/VALUES ttnn: tt=time(sec) nn=no. values
03:	4	LOCATION starting loc. for values

Input locations altered: 0

Intermediate locations required: 182. To accommodate this number, increase the value in *A window 2, if keying program steps in by hand.

PARAMETER 1. ADDRESS

Enter the address for the CR23X acting as an SDI-12 sensor (0-9, †10-126 decimal value for ASCII character, see Appendix E). Each SDI-12 sensor connected to a given control port must have a unique address.

PARAMETER 2. TIME/VALUES

Enter the time in seconds required for the sensor CR23X to complete subroutine 98 followed by the number of input locations to be returned to the SDI-12 recorder. The format is *ttnn* where *tt* specifies the time in seconds and *nn* (maximum 63) is the number of values from Input Locations to be sent.

Enter a time of 0 to transfer the values already stored in input locations. With a time of zero, the remaining instructions in Subroutine 98 are not executed.

The actual time to complete subroutine 98 is the time required to execute all instructions from Instruction 106 (SDI-12 sensor) to the final Instruction 95 (End).

For response to the 'M' command, the entered time may be longer than the actual time without slowing the data exchange because the sensor CR23X signals the SDI-12 data recorder when the data is ready for transfer.

For response to either the 'M' or the 'C' command, if the entered time is too short the sensor CR23X will not respond and the data values will not be transferred. Similarly, no response occurs if the SDI-12 recorder queries the sensor CR23X before the entered time has elapsed.

PARAMETER 3. LOCATION

This parameter determines the starting input location for the 'n' values to be returned to the recorder. The 'M' or 'M1-M9' command issued by the SDI-12 recorder determines if the starting location is actually that specified in Parameter 3 or a multiple of 'n' past Parameter 3.

Starting input location = Parameter 3 + (n∗x), where n is specified in Parameter 2, and, x is the number following the 'M' sent by the SDI-12 recorder (1-9).If the 'M' command is sent by the recorder x = 0.

Results of Instruction 106

The sensor CR23X will return a set of input locations in response to the M or M1..M9 command sequence. The set of Locations returned is determined by Parameters 2 and 3 of Instruction 106.

The three values, sent in response to a V command sequence, indicate the status of the sensor CR23X. The first and second values are from the ⁎ B mode of the sensor CR23X, giving the number of watchdog errors (E08) and the number of table overruns that have occurred. The third is a signature of the sensor CR23X memory. This signature is created by the same technique that the Instruction 19 (Signature) uses.

In response to an I command, the CR23X sends the string 'a 12CAMPBELL CR23X 001 bbbb' where a is the sensor address and bbbb is the number from the eighth window of the ⁎ B mode.

SECTION 9. INPUT/OUTPUT INSTRUCTIONS

*** 107 SDM-CSAT3 ***

FUNCTION
This instruction controls and receives data from CSI's three-dimensional sonic anemometer (CSAT3). See the CSAT3 manual for information on Instruction 107.

Input locations altered: 5 per repetition

*** 108 SDM-UDG01 ***

FUNCTION
This instruction is used to measure the UDG01 Ultrasonic Snow Depth Sensor using control ports C1, C2, and C3. Air temperature is a required input for Instruction 108.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	UDG01 SDM address (Base 4)
02:	4	Input location number for the air temperature
03:	4	Input location number for the distance
04:	FP	Multiplier
05:	FP	Offset

Input locations altered: 1

*** 109 SDMX50 CHANNEL SELECT ***

FUNCTION
This instruction is used to manually select probes on the CSI TDR Soil Moisture Measurement System.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	SDMX50 Address (Base 4)
02:	2	SDMX50 channel number

Input locations altered: 0

*** 110 SDM-GROUP TRIGGER ***

FUNCTION
This instruction is used to synchronize the measurements of up to 15 SDM sensors that support the group trigger protocol. The data is retrieved with the appropriate device specific instruction. Instruction 110 does not have any parameters.

*** 113 SDM-SI04 ***

FUNCTION
Instruction 113 communicates with the SDM-SI04 serial input multiplexer. See the SDM-SI04 manual for directions.

*** 114 SET TIME ***

FUNCTION
Instruction 114 can be used to set the CR23X clock from values in input locations.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Option code:
		0 set time with hr,min,sec with values in 3 input locations.
		1 set time with day,hr,min,sec using 4 input locations.
		2 set time with yr,day,hr,min,sec using 5 input locations.
02:	4	First Input location number

Input locations read: 3 to 5, depending on option

*** 115 SET SDM BAUD ***

FUNCTION
Instruction 115 may be used to set the SDM communication rate. This may be necessary when communicating over longer cable lengths. The default bit period is 10 microseconds (entering either 0 or 1 will result in this period).

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Bit period, 10µs units

Normally this parameter represents the bit period. If the parameter is indexed (--), the value entered is an Input Location that contains the bit period to use.

> **NOTE:** The TDR Instruction 100 and the SDM-SIO4 Instruction 113 automatically adjusts the SDM communication rate to the fastest that will work.

*** 117 READ DATALOGGER ID ***

FUNCTION
Instruction 117 stores the datalogger ID into an input location. The datalogger ID is set in the ⁎ D mode with Command 8 (Section 1.8.4).

PARAM NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location number

Input locations altered: 1

*** 118 SDM-CAN ***

Consult Campbell Scientific.

*** 130 STATUS MONITOR ***

FUNCTION
Stores an error count or status information from ⁎ B into an input location. Indexing the option code will cause the instruction to store the value into an input location and then clear the ⁎ B value. This is designed to assist with troubleshooting and to alert users of possible problems.

PARAM NUMBER	DATA TYPE	DESCRIPTION
01:	2	Option Code (Index (--) to reset error count after reading)
02:	4	Input Location

Input Locations Altered: 1

Option Codes:

0	Read Watchdog or E08s Errors
1	Read Table Overruns
2	Read Low 12V Detection
3	Read Lithium Battery Voltage Level (Indexing does nothing)
4	Read Flash Errors
5	5V Error Counter
6	Program time of Table 1 (0.407 µs resolution, range is 6.8265)
7	Time required of previous instruction
8	Internal Panel temperature

Using option 8 instead of P17, thermocouple measurements run about five times faster, but the measurement may be as old as 2.8 minutes. This option should only be used over P17 when timing is more important than how old the reading is.

*** 131 EXTENDED VIBRATING WIRE *** MEASUREMENT

FUNCTION
Excites a vibrating wire sensor with a swept frequency (from low frequency to high), then measures the response period and calculates $1/T^2$, where T is the period in ms. Excitation is normally provided for each repetition. As an option, a single excitation can be made prior to all repetitions of the measurement. A AVW1 or AVW4 Vibrating Wire Interface is required for these sensors, but it may have trouble reading the low frequencies.

> T - Amount of Time to sweep between the specified frequencies
>
> N - Number of steps to use when sweeping from the Starting Frequency to the Ending Frequency

The $1/T^2$ value stored in the first Input Location specified in Parameter 11 is the result after the # of Cycles specified in Parameter 10. The $1/T^2$ value stored in the sequential Input Location is the result after 5 times the # of Cycles specified in Parameter 10. The value in the second Input Location is the correct value to use unless the reading falls into a certain range. This instruction is most often used with the vibrating strip. Vibrating wire/strip manufacturers have datalogger programming instructions they recommend using to decide which of the 2 readings to use.

TABLE 9-10. Extended Vibrating Wire Range Codes

Range Code	Peak to Peak Volts Required @ Max. Freq.*	Maximum Frequency
1	2 mV	10 kHz
2	5 mV	20 kHz
3	20 mV	30 kHz
4	500 mV	100 kHz

PARAM NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions (Index (- -) to skip repeat of excitation)
02:	2	Range Code (Table 9-10)
03:	2	Single-ended channel for first measurement
04:	2	EX CHAN
05:	4	Starting Frequency (HZ)
06:	4	Ending Frequency (HZ)
07:	4	T (sweep, Units = 1 msec)
08:	4	N (Number of steps)
09:	4	Delay after excitation before measurement (Units = 0.01 sec)
10:	4	CYCLES to measure
11:	4	DELAY between reps (Units = 0.01 sec)
12:	4	First Input Location
13:	FP	Multiplier
14:	FP	Offset

Input Locations Altered: 2 per repetition

*** 132 INPUT SETTLING TIME ***

Instruction P132 allows a user to change the default 450 µs input voltage settling time for Instructions P1, P2, P5, P6, P7, P9, P11, P13, and P14. Measurement errors will result if insufficient input settling time exists for a given sensor and cable length (see Section 13.3). The minimum allowable settling time for P132 is 100 µs for all input ranges except the ± 10 mV range, in which 200 µs is the minimum. The maximum allowable settling time is 6 ms. An entry of 0.0000 ms results in the default 450 µs input settling time. High-output impedance (> 1 kΩ) sensors and / or long cable lengths may require more than the 450 µs default settling time. Low-output impedance sensors and / or short cable lengths may require less settling time. A single P132

instruction in the datalogger program will change all applicable default settling times. Multiple P132 instructions in a datalogger program will cause settling time changes for the analog measurement instructions following each P132. Shortened settling times for the P10 and P17 instruction will degrade the measurement accuracy, while longer times will increase the time required to perform the instructions. A P132 within a scan does not over-ride the settling times in the P4 and P8 instruction; instead it adds to this time.   If P132 is used in the program, **P4, P8, P10, and P17 Instructions should be preceeded by a P132** to set the settling time back to the desired 450 µs (time 0.0000). Performing P132 instructions within a scan increases the time to execute the scan by 0.13 ms per P132 instruction. This needs to be considered when using the P132 instruction to reduce the time required to complete a scan. P132 in Table 1 will not affect measurements in Table 2. The reverse is also true.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	FP	milliseconds Minimum = 0.2 for X1 Range codes, else Minimum range is 0.1. Maximum = 6.0 ms. 0 = default = 450 milliseconds

Input Locations Altered = 0

*** 133 Continuous Analog Out ***

Sets a continuous analog output port to a voltage level specified in an input location. The analog output degrades approximately 0.17mV every 7 seconds requiring the instruction to be periodically repeated to maintain a given output accuracy.

Port Option	Description	mA Source (+V)	mA Sink (-V)
01 - 02	CAO1 & CAO2	15	5
11 - 12	CAO1 & CAO2 with Boost	15	15

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Option / CAO Channel Number
02:	4	Input Location number of analog output magnitude (mV)

Input location read: 1

*** 134 AM25T ***

This instruction controls the AM25T Solid State Multiplexer for Thermocouples. P134 can measure thermocouples temperatures or millivoltage.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repititions
		0 Measure Reference Temperature Only
		1..25 Measure AM25T Channels
02:	2	TC Range Code (Table 9-1)
03:	2	First AM25T Channel
04:	2	Differential Channel
05:	2	RTD Integration/ Excitation Channel (x)
		0 Reference Temperature not updated
		1x 250 usec integration (no AC rejection)
		2x 60 Hz rejection
		3x 50 Hz rejection

06:	2	Clock Control Port
07:	2	Reset Control Port
08:	2	Thermocouple Type (x) (Table 9-4)
		x=0 mV result
		x=1..8 Type TEKJBRSN
		9x Common Mode Range Check (See P14)
09:	4	Reference Temperature Location
10:	4	First Destination Input Location
11:	FP	Multiplier
12:	FP	Offset

Input location altered: 1 per repetition

*** 189 SDM-CS7500 ***

FUNCTION
This instruction controls and receives data from the CS7500 LI-COR Open Path Gas Analyzer. See the CS7500 manual for information on Instruction 189.

SECTION 10. PROCESSING INSTRUCTIONS

To facilitate cross referencing, parameter descriptions are keyed [ ] to the values given on the PROMPT SHEET. These values are defined as follows:

    [Z] = Destination input location for result
    [X] = Input location of X
    [Y] = Input location of Y
    [F] = Fixed Data (user specified floating
            point number)

*** 30 $Z = F * 10^n$ ***

FUNCTION
Store a fixed value into an input location. The value is entered in scientific notation; the absolute value of the number may range from 1×10^{-19} to 9×10^{18}. A value smaller than the minimum is set to 0, while a larger value is set to the maximum.

PARAM. NUMBER	DATA TYPE	DESCRIPTION	
01:	FP	Mantissa	[F]
02:	2	n (Exponent of 10)	
03:	4	Destination for input location	[Z]

Input locations altered: 1

*** 31 Z = X ***

FUNCTION
Copy data from one input location to another.

PARAM. NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Source input location number	[X]
02:	4	Destination input location	[Z]

Input locations altered: 1

*** 32 Z = Z + 1 ***

FUNCTION
Add 1 to the current value in the specified input location.

PARAM. NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Destination input location	[Z]

Input locations altered: 1

*** 33 X + Y ***

FUNCTION
Add X to Y and place result in a third input location.

PARAM. NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Input location of X	[X]
02:	4	Input location of Y	[Y]
03:	4	Dest. input location of X+Y	[Z]

Input locations altered: 1

*** 34 X + F ***

FUNCTION
Add F to X (where F is a fixed floating point number) and place the result in an input location.

PARAM. NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Input location of X	[X]
02:	FP	Fixed value	[F]
03:	4	Dest. input location of X+F	[Z]

Input locations altered: 1

*** 35 X - Y ***

FUNCTION
Subtract Y from X and place the result in an input location.

PARAM. NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Input location of X	[X]
02:	4	Input location of Y	[Y]
03:	4	Dest. input location for X-Y	[Z]

Input locations altered: 1

SECTION 10. PROCESSING INSTRUCTIONS

*** 36 X * Y ***

FUNCTION
Multiply X by Y and place the result in an input location (Z).

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location of X [X]
02:	4	Input location of Y [Y]
03:	4	Dest. input location for X*Y [Z]

Input locations altered: 1

*** 37 X * F ***

FUNCTION
Multiply X by F (where F is a fixed multiplier) and place the result in an input location.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location of X [X]
02:	FP	Fixed value [F]
03:	4	Dest. input location for X*F [Z]

Input locations altered: 1

*** 38 X / Y ***

FUNCTION
Divide X by Y and place the result in an input location. Division by 0 will cause the result to be set to the maximum CR23X number (99999).

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location of X [X]
02:	4	Input location of Y [Y]
03:	4	Dest. input location for X/Y [Z]

Input locations altered: 1

*** 39 SQUARE ROOT ***

FUNCTION
Take the square root of X and place the result in an input location. If X is negative, 0 will be stored as the result.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location of X [X]
02:	4	Dest. input location for $X^{1/2}$ [Z]

Input locations altered: 1

*** 40 LN(X) ***

FUNCTION
Take the natural logarithm of X and place the result in an input location. If X is 0 or negative, -99999 will be stored as the result.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location of X [X]
02:	4	Dest. input location for LN(X) [Z]

Input locations altered: 1

*** 41 EXP(X) ***

FUNCTION
Raise the exponential (EXP) base e to the X power and place it in an input location.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location of X [X]
02:	4	Dest. input for EXP(X) [Z]

Input locations altered: 1

*** 42 1/X ***

FUNCTION
Take the inverse of X and place the result in an input location. If X=0, 99999 will be given as the result.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location of X [X]
02:	4	Dest. input location for 1/X [Z]

Input locations altered: 1

*** 43 ABS(X) ***

FUNCTION
Take the absolute (ABS) value of X and place the result in an input location.

PARAM. NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Input location of X	[X]
02:	4	Dest. input location for ABS(X)	[Z]

Input locations altered: 1

*** 44 FRACTIONAL VALUE OF X ***

FUNCTION
Take the fractional (FRAC) value (i.e., the non-integer portion) of X and place the result in an input location.

PARAM. NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Input location of X	[X]
02:	4	Dest. input location for FRAC(X)	[Z]

Input locations altered: 1

*** 45 INTEGER VALUE OF X ***

FUNCTION
Take the integer (INT) value of X and place the result in an input location.

PARAM. NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Input location of X	[X]
02:	4	Dest. input location for INT(X)	[Z]

Input locations altered: 1

*** 46 X MOD F ***

FUNCTION
Do a modulo divide of X by F and place the result in an input location. X MOD F is defined as the REMAINDER obtained when X is divided by F (e.g., 3 MOD 2 = 1). X MOD 0 returns X.

PARAM. NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Input location of X	[X]
02:	FP	Fixed divisor	[F]
03:	4	Dest. input loc. For X MOD F	[Z]

Input locations altered: 1

*** 47 X^Y ***

FUNCTION
Raise X to the Y power and place the result in an input location.

PARAM. NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Input location of X	[X]
02:	4	Input location of Y	[Y]
03:	4	Dest. input location for X^Y	[Z]

Input locations altered: 1

*** 48 SIN(X) ***

FUNCTION
Calculate the sine of X (X is assumed to be in degrees) and place the result in an input location. The cosine of a number can be obtained by adding 90 to the number and taking the sine (COSX = SIN (X + 90)).

PARAM. NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Input location of X	[X]
02:	4	Dest. input location for SIN(X)	[Z]

Input locations altered: 1

*** 49 SPATIAL MAXIMUM ***

FUNCTION
Find the spatial maximum (SPA MAX) value of the given set or SWATH of input locations and place the result in an input location.

To find the input location where the maximum value occurs, add 1000 to the input location number destination selected [Z] and enter this modified location number as Parameter 03. The input location ID of the maximum value observed will then be stored in destination [Z] plus 1.

SECTION 10. PROCESSING INSTRUCTIONS

Parameter 3 cannot be entered as an indexed location within a loop (Instruction 87). To use Instruction 49 within a loop, enter Parameter 3 as a fixed location and follow 49 with the Instruction 31 (Move Data). In Instruction 31, enter the location in which 49 stores its result as the source (fixed) and enter the destination as an indexed location.

PARAM. NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Swath	[SWATH]
02:	4	Starting input location [1ST LOC]	
03:	4	Dest. input location for maximum	[MAX or Z]

Input locations altered: 1 or 2

*** 50 SPATIAL MINIMUM ***

FUNCTION
Find the spatial minimum (SPA MIN) value of the given set or SWATH of input locations and place the result in an input location. To find the input location where the minimum value occurs, follow the instructions given above for SPATIAL MAXIMUM.

Parameter 3 cannot be entered as an indexed location in a loop. Within a loop, Instruction 50 must be used in conjunction with Instruction 31 as described for Instruction 49.

PARAM. NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Swath	[SWATH]
02:	4	Starting input location [1ST LOC]	
03:	4	Dest. input location for minimum	[MIN or Z]

Input locations altered: 1 or 2

*** 51 SPATIAL AVERAGE ***

FUNCTION
Take the spatial average (SPA AVG) over the given set or SWATH of input locations and place the result in an input location.

PARAM. NUMBER	DATA TYPE	DESCRIPTION	
01:	4(--)*	Swath	[SWATH]
02:	4	Starting input location [1ST LOC]	
03:	4	Dest. input location of average	[AVG or Z]

Input locations altered: 1

*If Swath is indexed (--) the zeros are not included in the average.

*** 52 RUNNING AVERAGE ***

FUNCTION
This instruction calculates the running average of a value in an input location. The most recent n values (where n is the number specified in parameter 4) are kept in intermediate storage. When Instruction 52 is executed, the current value is written over the oldest value and the average of the values is calculated and stored in the destination location. Out of range values (displayed as -99999.) are not included in the average.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Reps (REPS)
02:	4	Input location of source data
03:	4	Destination input location
04:	4	Number of values (n) in running average window (NUMAVG)

Input locations altered: 1 per repetition

Intermediate locations required: Reps *(2+n)

*** 53 SCALING ARRAY WITH ***
MULTIPLIER AND OFFSET

FUNCTION
Take 4 input location values, multiply each by a floating point constant, then add another floating point constant to the resulting products and place the final results back into each of the original 4 input locations.

PARAM. NUMBER	DATA TYPE	DESCRIPTION	
01:	4	First input location	[STRT LOC]
02:	FP	Multiplier 1	[A1]
03:	FP	Offset 1	[B1]
04:	FP	Multiplier 2	[A2]

05:	FP	Offset 2	[B2]
06:	FP	Multiplier 3	[A3]
07:	FP	Offset 3	[B3]
08:	FP	Multiplier 4	[A4]
09:	FP	Offset 4	[B4]

Input locations altered: 4

*** 54 BLOCK MOVE ***

FUNCTION
Executes a "block move" of data in input locations. Parameters specify the number of values to move, the source, source step, destination, and destination step. The "step" parameters designate the increment of the source and destination input locations for each value that is moved. For example, a "source step" of 2 and a "destination step" of 1 will move data from every other input location to a contiguous block of input locations.

PARAM. NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Number of values to move	
02:	4	1st source location	
03:	2	Step of source	
04:	4	1st destination location	
05:	2	Step of destination	

Input locations altered: Number of values of move

*** 55 5TH ORDER POLYNOMIAL ***

FUNCTION
Evaluate a 5th order polynomial of the form

$$F(X)=C0+C1X+C2X^2+C3X^3+C4X^4+C5X^5$$

where C0 through C5 are the coefficients for the argument X raised to the zero through fifth power, respectively. The magnitude of the user entered coefficient is limited to a range of $\pm.00001$ to ±99999. Polynomials with coefficients outside this range can be modified by pre-scaling the X value by an appropriate factor to place the coefficients within the entry range. Pre-scaling can also be used to modify coefficients which are very close to 0 to increase the number of significant digits.

PARAM. NUMBER	DATA TYPE	DESCRIPTION	
01:	2	Repetitions	[REPS]
02:	4	Starting input location for X	[X]
03:	4	Dest. input location for F(X)	[F(X) or Z]
04:	FP	C0 coefficient	[C0]
05:	FP	C1 coefficient	[C1]
06:	FP	C2 coefficient	[C2]
07:	FP	C3 coefficient	[C3]
08:	FP	C4 coefficient	[C4]
09:	FP	C5 coefficient	[C5]

Input locations altered: 1 per repetition

*** 56 SATURATION VAPOR PRESSURE ***

FUNCTION
Calculate saturation vapor pressure (over water SVPW) in kilopascals from the air temperature ($\circ$C) and place it in an input location. The algorithm for obtaining SVPW from air temperature ($\circ$C) is taken from: Lowe, Paul R.: 1977, "An approximating polynomial for computation of saturation vapor pressure," *J. Appl. Meteor,* **16**, 100-103.

Saturation vapor pressure over ice (SVPI) in kilopascals for a 0$\circ$C to -50$\circ$C range can be obtained using Instruction 55 and the relationship

$$SVPI = -.00486 + .85471 X + .2441 X^2$$

where X is the SVPW derived by Instruction 56. This relationship was derived by Campbell Scientific from the equations for the SVPW and the SVPI given in Lowe's paper.

PARAM. NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Input location of air temperature $\circ$C	[TEMP.]
02:	4	Destination input location for saturated vapor pressure	[VP or Z]

Input locations altered: 1

*** 57 VAPOR PRESSURE FROM ***
WET-/DRY-BULB TEMPERATURES

FUNCTION
Calculate vapor pressure in kilopascals from wet and dry-bulb temperatures in °C. This algorithm type is used by the National Weather Service:

$$VP = VPW - A(1 + B*TW)(TA - TW) P$$

VP = ambient vapor pressure in kilopascals
VPW = saturation vapor pressure at the wet-bulb temperature in kilopascals
TW = wet-bulb temperature, °C
TA = ambient air temperature, °C
P = air pressure in kilopascals
A = 0.000660
B = 0.00115

Although the algorithm requires an air pressure entry, the daily fluctuations are small enough that for most applications a fixed entry of the standard pressure at the site elevation will suffice. If a pressure sensor is employed, the current pressure can be used.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location of atmospheric pressure in kilopascals [PRESSURE]
02:	4	Input location of dry-bulb temp. [DB TEMP.]
03:	4	Input location of wet-bulb temp. [WB TEMP.]
04:	4	Destination input location for vapor pressure [VP or Z]

Input locations altered: 1

*** 58 LOW PASS FILTER ***

FUNCTION
Apply a numerical approximation to an analog resistor capacitor (RC) low pass (LP) filter using the following algorithm.

$$F(Xi) = W*Xi + F(Xi-1) * (1-W)$$

Where X = input sample, W = user entered weighting function (O< W <1). If W=O,

$F(Xi)=X_1$; if W=1, F(Xi)=X and F(Xi-1) = output calculated for previous sample

The equivalent RC time constant is given by T/W, where T is the sampling time in seconds. For values of W less than 0.25, the analogous "cut off" frequency (the frequency where the ratio of output to input is .707) is accurately represented by $W/(2\pi T)$. For larger values of W, this "analog" estimate of the cutoff frequency becomes less representative.

On the first execution after compiling, F(X) is set equal to X.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions [REPS]
02:	4	First input location for input data [X]
03:	4	Dest. input location for first filtered result[F(X) or Z]
04:	FP	Weighting function, W [W]

Input locations altered: 1 per repetition
Intermediate locations required: 1 per repetition

*** 59 BRIDGE TRANSFORM ***

FUNCTION
This instruction is used to aid in the conversion of a ratiometric Bridge measurement by obtaining the value for R_s which is equivalent to $R_f[X/(1-X)]$, where X is the value derived by the standard CR23X Bridge Measurement Instructions (with appropriate multiplier and offset, Section 13.5) and R_f represents the MULTIPLIER value. The result of Instruction 59 is stored in the same location that X was.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions [REPS]
02:	4	Starting input location & result destination [X]
03:	FP	Multiplier (Rf) [MULT]

Input locations altered: 1 per repetition

*** 60 FAST FOURIER TRANSFORM ***

THEORY

Instruction 60 performs a Fast Fourier Transform (FFT) on a set of data contained in contiguous locations in Input Storage. The FFT is used to obtain information on the relative magnitudes and phases of the various frequency components in a time varying signal. FFT theory requires that the signal be sampled at a frequency that is at least two times faster than the highest frequency component in the signal. For example, a signal representing ocean waves with a maximum frequency of 0.125 Hz would need to be sampled at a rate of 0.25 Hz or greater. The measurements must be made at the appropriate sampling rate and stored in contiguous input locations before the FFT can be applied. The measured data stored in sequential input locations is also referred to as the "original time series data".

The results of the FFT can be expressed as: 1) the power spectrum, 2) the real and imaginary components, or 3) the magnitude and phase components. The power spectrum results indicate the amount of power at the different frequencies but do not contain any phase information. The real and imaginary results are analogous to the orthogonal (east and north) representation of a wind vector. The magnitude and phase results are analogous to the polar (speed and direction) representation of a wind vector.

If desired, the original time varying signal can be reconstructed by taking the Inverse Fourier Transform of either the real and imaginary or the magnitude and phase results.

PROGRAMMING

The FFT Instruction is a Processing Instruction and will not output data to final memory or any data storage device. FFT results are transferred to Final Memory by setting the Output Flag and sampling the input locations which contain the FFT results.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Log base 2 of N, where N is the number of samples in the original data; e.g., this parameter is 11 for 2048 samples or 12 for 4096 samples (maximum).
02:	2	FFT Option 0X = Power Spectrum 1X = Real & Imaginary 2X = Magnitude & Phase X0 = No Taper X1 = Taper
03:	2	Log base 2 of the number of adjacent bins to average; e.g., this parameter is 3 for averaging 8 adjacent bins.
04:	4	Input location containing the first value in the original time series. After this program has executed, this location will contain the first spectral bin or bin average of the result.
05:	FP	Multiplier for scaling FFT results.

Input storage locations: Equal to the number of values in the original time series data.

Intermediate locations required: N/8+2

Execution time in seconds:

POWER SPECTRUM
$$= 0.00035*N*LOG_{10}(N)/LOG_{10}(2)+0.01$$
MAGNITUDE AND PHASE
$$= 0.00035*N*LOG_{10}(N)/LOG_{10}(2)+0.14$$
REAL AND IMAGINARY
$$= 0.00072*N*LOG_{10}(N)/LOG_{10}(2)+1.44$$

PARAMETER 1 is an integer from 1 to 12. The number 2, raised to the power of Parameter 1, is equal to the number of data values in the original time series data, N. For example, if Parameter 1=12, then N=4096. Maximum value for Parameter 1 is 12 (4096).

PARAMETER 2, the FFT Option, is a two digit parameter. The first digit specifies which results are to be generated and the second digit specifies whether or not a "taper" is applied.

First Digit: A "**0**" in the first digit specifies that the FFT result be expressed as the power spectra. Bin averaging is allowed with this option. A "**1**" in the first digit specifies that the FFT result be expressed as the real and imaginary components. Bin averaging is not allowed with this option. A "**2**" in the first digit specifies that the FFT result be expressed as

10-7

the magnitude and phase components. Bin averaging is not allowed with this option.

Second Digit: A "**0**" in the second digit specifies that no taper be applied, whereas a "**1**" specifies that a taper be applied. If the original data set is not known to be periodic with an integral number of periods in the data set, then it is necessary to apply a taper to the beginning and end of the data. The taper that is applied is a four term Blackman-Harris as specified by Fredric J. Harris, Proceedings of the IEEE, Vol. 66, No. 1, January 1978. When the taper is used, the CR23X applies a correction by multiplying the results by the ratio of the variance before the taper to the variance after the taper.

The variance ratio is saved only when the power spectra option is selected. The variance ratio is saved in input location $(N/2)+B$, where B is the location (Parameter 4) that contained the first value of the original time series data.

PARAMETER 3 specifies how many of the resulting spectral bins (if any) are to be averaged. Bin averaging is only allowed when power spectral is output. Parameter 3 is equal to the log base 2 of A where A is the number of bins to be averaged. For example, if there are 1024 samples in the original time series data and the resulting 512 spectral bins are averaged in groups of 8 (Parameter 3 = 3 = log base 2 of 8) then 63 (=N/2A-1) averaged bins will be produced.

PARAMETER 4 defines which input location will contain the first value at the original time series data. Once the FFT program has executed, this location will contain the first resulting value.

PARAMETER 5 is a scaling multiplier that is applied to the FFT results (i.e., after the FFT computation has been done). The multiplier is not applied to the DC component.

MEMORY REQUIREMENTS
The number of Input Memory Locations allocated must be enough to accommodate the N sampled values, where N is defined under Parameter 1. The number of Intermediate Memory Locations allocated must be greater than or equal to $(N/8 + 2)$. If insufficient Input or Intermediate Memory Locations have been allocated, the datalogger will flag an E60 or E04, respectively.

> **NOTE:** The CR23X has 1 Intermediate Location that is not available for use by Processing or Output Instructions so the number of Intermediate Locations allocated must exceed the indicated total requirement by at least 1.

FFT RESULTS WITHOUT BIN AVERAGING

When no bin averaging is specified, the FFT results may be calculated in terms of the Power Spectrum, Real and Imaginary, or Magnitude and Phase. The rest of this section deals with the DC component, bin frequency, and the FFT results just mentioned. An example showing each of the possible results is given in Section 8.11.1.

DC COMPONENT
Before the FFT is applied, the average of the original time series data is subtracted from each value. This is done to maintain the resolution of the math in the rest of the FFT calculations.

If the Real and Imaginary or the Magnitude and Phase results are specified by Parameter 2 the DC component is the average of the original time series data. If Power Spectrum results are specified, the DC component is equal to the square of the average of the original time series data times 2N.

The DC component is stored in the first input location specified by Parameter 4 which corresponds with the frequency at 0 Hz (bin 0).

BIN FREQUENCY
The band width or the frequency covered by each bin is equal to F/N where F is the sample frequency in Hz (1/scan interval in seconds).

The frequency (f_i) of any given bin i where i ranges from 0 to (N/2)-1 is given by the following equation:

$$(i\text{-}1 * F) / N \; < \; f_i < \; (i * F) / N \qquad [1]$$

For example, given that the Power Spectrum result shows that the energy peak of a signal falls in bin 128 when it is sampled at a frequency of 10 Hz for 1024 samples, the frequency of the signal is:

$$127 * 10 / 1024 < f_i < 128 * 10 / 1024$$

$$1.24 \text{ Hz} < f_i < 1.25 \text{ Hz}$$

REAL AND IMAGINARY COMPONENTS
The result of the FFT when the Real and Imaginary option is selected is N/2 input locations containing the Real components (R_i) followed by N/2 input locations containing the Imaginary components (I_i). There is a real and an imaginary component for each bin. The value of i varies from 1 to N/2. The real and imaginary results at each frequency i, are related to the magnitude (M_i) and phase (P_i) as shown below:

$$R_i = M_i * \cos P_i \quad [2]$$

$$I_i = M_i * \sin P_i \quad [3]$$

where M_i is the magnitude and P_i is the phase of the signal in degrees. Magnitude is half of the zero to peak amplitude or one quarter of the peak to peak value of the sinusoidal signal.

MAGNITUDE AND PHASE COMPONENTS
The result of the FFT when the magnitude and phase option is selected is N/2 input locations containing the magnitude components (Mi) followed by N/2 input locations containing the phase components (P_i). Magnitude is half of the zero to peak amplitude or one quarter of the peak to peak value of the sinusoidal signal. There is a magnitude and a phase component for each bin. The value of i varies from 1 to N/2. The magnitude and phase components are related to the real (R_i) and imaginary (I_i) components as shown below:

$$M_i = SQRT[(R_i*R_i) + (I_i*I_i)] \quad [4]$$

$$P_i = \arctan (I_i/R_i) \quad [5]$$

To calculate the magnitude and phase the CR23X's FFT algorithm must first compute the real and imaginary components. Conversion from real and imaginary to the magnitude and phase requires quite a bit more datalogger execution time and no new information is gained. If datalogger execution time is limiting, program the datalogger to store the real and imaginary results and have a computer do the conversion to magnitude and phase during the data reduction phase. The FFT assumes the signal was sampled at the beginning of each of N intervals. Since the FFT assumes the signal is periodic with a period equal to the total sampling period, the result of its phase calculation at each frequency component is the average of the phase at the *beginning* of the first interval with

the phase at the *end* of the last interval. The phase is the angle (0 to 360 degrees) of the cosine wave that describes the signal at a particular point in time.

POWER SPECTRUM
The result of the FFT when the power spectrum option is selected is N/2 bins of spectral energy (PS_i) representing frequencies from 0 Hz to 1/2 the sampling frequency. The value of i varies from 1 to N/2. The result in each bin i, is related to the magnitude (M_i) of the wave in the following manner:

$$PS_i = 2*N*(M_i*M_i) \quad [6]$$

where the magnitude is half of the zero to peak amplitude or one quarter of the peak to peak value of the sinusoidal signal.

The power spectrum can also be expressed as either of the following:

$$PS_i = N*(U_i*U_i) \quad [7]$$

$$PS_i = F*T*(U_i*U_i) \quad [8]$$

U_i is defined as the root mean square (RMS) value of the sine component of frequency i (f_i) (U_i = magnitude (M_i) of the sine wave multiplied by the square root of 2) in units of the input signal multiplied by the scaling multiplier. In equation 8, F is the sampling frequency (Hz) and T is the duration of the original time series data (seconds).

When the FFT results are expressed in terms of the power spectrum, a multiplier of 1 will cause the average of all the bins to be very nearly equal to twice the variance of the original data.

FFT RESULTS WITH BIN AVERAGING

When bin averaging is specified, the FFT results can only be calculated in terms of the power spectrum. The rest of this section deals with the DC component, bin frequency, and the power spectrum results. An example showing bin averaging FFT results is given in Section 8.11.2.

DC COMPONENT
Before the FFT is applied, the average of the original time series data is subtracted from each value. This is done to maintain the resolution of the math in the rest of the FFT calculations. When bin averaging is specified then the DC component is not output.

SECTION 10. PROCESSING INSTRUCTIONS

BIN FREQUENCY
The band width or the frequency covered by each averaged bin is equal to FA/N where F is the sample frequency in Hz (1/scan interval in seconds) and A is the number of bins being averaged.

The frequency (f_i) of any given averaged bin i where i ranges from 1 to (N/2A)-1 is given by the following equation:

$$i-1 * F * A / N < fi < i * F * A / N \qquad [9]$$

For example, given that the power spectrum result shows that the energy peak of a signal falls in bin 32 when it is sampled at a frequency of 10 Hz for 1024 samples and that the bin averaging specified is 4, the frequency of the signal in bin i is:

$$31 * 10 * 4 / 1024 < f_i < 32 * 10 * 4 / 1024$$

$$1.21 \text{ Hz} < f_i < 1.25 \text{ Hz}$$

POWER SPECTRUM
The result of the FFT with A bins averaged, are (N/2A)-1 bins of average spectral energy (APSn) representing frequencies from 0 Hz to 1/2 the sampling frequency. The value of i varies from 1 to (N/2A)-1. The results are found in consecutive input locations starting with the first one specified by Parameter 4. The value for average bin n (APS_n) is related to the spectral bin values (PS_i see previous section) by the following equation:

$$APS_n = (\Sigma PS_i + 0.5(PS_{nA-A/2} + PS_{nA+A/2}))/A \qquad [10]$$

where i goes from nA-(A/2-1) to nA+(A/2-1)

The following table illustrates how bin averaging is done given a time series of 1024 values taken at one per second with the resulting 512 spectral bins averaged in groups of 4 (Parameter 3 = log base 2 of 4 = 2) to produce 127 averaged bins.

TABLE 10-1. Example of FFT Power Spectrum Bin Averaging (Assuming 1024 time series values starting in Location 1)

No Bin Averaging			Averaged in Groups of 4		
BIN NO.	LOC.	REPRESENTATIVE FREQUENCY	AVERAGED BIN NO.	LOC. FREQUENCY	REPRESENTATIVE
0	1	DC			
1	2	1/1024			
2	3	2/1024			
3	4	3/1024			
4	5	4/1024	------------------------1	1	4/1024 or 1/256
5	6	5/1024			
6	7	6/1024			
7	8	7/1024			
8	9	8/1024	------------------------2	2	8/1024 or 2/256
9	10	9/1024			
10	11	10/1024			
11	12	11/1024			
		. . .			
502		502/1024			
503		503/1024			
504		504/1024	-------------------- 126	126	504/1024 or 126/256
505		505/1024			
506		506/1024			
507		507/1024			
508		508/1024	-------------------- 127	127	508/1024 or 127/256
509		509/1024			
510		510/1024			
511		511/1024			

Examples of the use of the FFT are given in Section 8.

*** 61 INDIRECT INDEXED MOVE ***

FUNCTION

Moves input data from location X to location Y, where X and/or Y are indirectly addressed (X and Y are stored in the locations specified by Parameters 1 and 2). If a location parameter is specified as "indexed" (xxxx--), then the actual input location referenced is calculated by adding the current index counter to the value in the specified input location. When used outside a loop, the addressing is simply indirect because the index counter is zero.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location containing source location X
02:	4	Input location containing destination location Y

Input locations altered: 1

*** 62 COVARIANCE/CORRELATION ***

FUNCTION

The Covariance/Correlation Instruction (COV/CORR) for the CR23X calculates: 1) means (M), 2) variances (V), 3) standard deviations (SD), 4) covariances (COV), and 5) correlation coefficients (CORR) for a set of input values and stores the results in Input Storage. The instruction requires the set of input values to be located contiguously in Input Storage. The user specifies the location of the first value and how many total values exist. The number of input values processed by each type of calculation (means, variances, etc.) is independently specified for each type. The order of the input values determines which inputs are processed for each type of calculation.

The instruction does not conform to the CR23X's four instruction types. Data located in Input Storage is processed, and the results returned to Input Storage whenever an averaging period is completed (Parameter 7) or the Output Flag is set. The instruction controlling the Output Flag must precede the COV/CORR Instruction. The reason the calculated results are returned to Input Storage is to allow the user access for additional processing before storing the values in Final Storage. Sample Instruction 70 must be used to transfer final results from Input to Final Storage.

To accommodate cases where it is desirable to calculate the statistical quantities over time periods shorter than the Output Interval, an averaging period shorter than the Output Interval may be specified. The final values obtained at the Output Interval are the properly weighted average of the values calculated at the subinterval averaging periods. This feature allows the recording of statistical data over longer time periods by removing the effect of longer period frequencies in the input signals; i.e., it provides a high pass filter. For example, assume the variance of an input is desired. It is determined that the averaging period should not exceed 5 minutes due to variation in the mean over longer time intervals. One approach is to calculate and record the variance every 5 minutes. By specifying the subinterval averaging period as 5 minutes and the Output Interval as 1 hour, however, the average of the 5 minute variances are recorded every hour. The averaging period is entered as the number of input samples in Parameter 7 of the COV/CORR Instruction. The number of samples for a given period is given by:

$$\text{Number of Samples} = \frac{\text{Averaging period in seconds}}{\text{Table execution interval in seconds}}$$

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Number of input values located sequentially in input memory
02:	2	Number of means desired
03:	2	Number of variances desired
04:	2	Number of standard deviations desired
05:	2	Number of covariances desired
06:	2	Number of correlations desired
07:	FP	Number of input samples in averaging period
08:	4	Input storage location of first value in sequential input string
09:	4	First Input Storage location to store string of final results

If the specified number of samples in the averaging period (Parameter 7) exceeds the actual number of samples occurring in the Output Interval, the Output Interval becomes the averaging period.

TABLE 10-2. Maximum Number of Outputs and Output Order for K Input Values.
(The output order flows from left to right and from top to bottom)

	INPUTS:	X1	X2	X3	X4	.....	XK
TYPE	MAX NO. OUTPUTS	(1st)	(2nd)	(3rd)	OUTPUTS (4th)		(Kth)
Means	K	M(X1)	M(X2)	M(X3)	M(X4)	.....	M(XK)
Variances	K	V(X1)	V(X2)	V(X3)	V(X4)	.....	V(XK)
Std. Deviation	K	SD(X1)	SD(X2)	SD(X3)	SD(X4)	.....	SD(XK)
Covariance	(K/2)(K-1)		CV(X1,X2)	CV(X1,X3) CV(X2,X3)	CV(X1,X4) CV(X2,X4) CV(X3,X4)	....	CV(X1,XK) CV(X2,XK) CV(X3,XK) . . . CV(XK-1,XK)
Correlation Coefficient	(K/2)(K-1)		CR(X1,X2)	CR(X1,X3) CR(X2,X3)	CR(X1,X4) CR(X2,X4) CR(X3,X4)	....	CR(X1,XK) CR(X2,XK) CR(X3,XK) . . . CR(XK-1,XK)

SYMBOL	DEFINITION
M(XK)	Mean of Kth value
V(XK)	Variance of Kth value
SD(XK)	Standard deviation of Kth value
CV(XK,X1)	Covariance of Kth and Ith value
CR(XK,X1)	Correlation of Kth and Ith value

MAXIMUM NUMBER OF POSSIBLE OUTPUTS

No limitation exists on the number of inputs that can be processed by the COV/CORR Instruction, but the processing time and Intermediate Storage requirements increase rapidly. The instruction requires that the input values reside sequentially in Input Storage. Since the number of outputs is specified for each type of statistical calculation, the instruction starts with the first value, working sequentially through the input values. For this reason, the order of the input values determines which values are processed.

Table 10-2 shows the maximum number of outputs which can be generated and the output order for K input values located sequentially in Input Storage. The output order shown in Table 10-2 flows from left to right and top to bottom.

INSTRUCTION PROCESSING
The CV/CR Instruction contains 3 separate processing phases:

1. Input Processing
2. Averaging Period Processing
3. Output Processing

The Input Processing phase is where new input values are received, the necessary squares or cross products formed, and the appropriate summations calculated as required by the desired final output. The rate at which the measurements can be made, the input values ordered, and the input processing phase completed without interruption determines the maximum rate of execution (see Execution Time).

The Averaging Period Processing occurs whenever the number of input samples entered in Parameter 7 is satisfied or whenever an Output Interval occurs (i.e., whenever the Output Flag is set). Results from these calculations are stored sequentially in Input Storage locations starting with the location specified in Parameter 9. The calculations performed are shown below, where N is the number of input samples in the averaging period:

1. Means:
$$M(X) = \Sigma X/N$$
2. Variances:
$$V(X) = \Sigma X^2/N - (\Sigma X/N)^2$$
3. Standard Deviations:
$$SD(X) = V(X)^{1/2}$$
4. Covariances:
$$CV(X,Y) = \Sigma XY/N - \Sigma X \ \Sigma Y/N^2$$
5. Correlations:
$$CR(X) = CV(X,Y)/(SD(X)SD(Y))$$

NOTE:

1. The square root algorithm in the CR23X returns a result of 0 for negative arguments.

2. The divide algorithm returns the largest floating point number possible ($\pm 10^{18}$ displayed as ± 99999) for a divide by 0.

3. When computing the variance of a constant signal, round off error produces a small negative result. The CR23X returns a 0 for the square root of a negative number; therefore, the standard deviation is set to 0. If the signal is also used in a correlation calculation, division by 0 returns an overrange value for the correlation result.

If a fast execution interval is specified, it is possible that the combined execution times of the input and averaging period processing may exceed the program table execution interval. The occurrence of an execution interval Overrun (see Section 1.1.1 of CR23X Operator's Manual) is noted by a "$^T{}_o$" in the lower right of the display in the Running Table mode ($\boxed{*}$ $\boxed{0}$ mode). This results in the omission of one input sample. The calculations are not affected, however, since the number of input samples is incremented only when valid input processing occurs. Averaging period processing occurs only when the number of input samples specified in Parameter 7 is accumulated.

Regardless of whether all of the input samples for the averaging period (specified in Parameter 7) have occurred or not, averaging period processing occurs whenever the Output Flag is set. This accommodates situations where the Output Interval may not be an integer multiple of the averaging period. If for example a 30 minute Output Interval is set by Instruction 92 and an 8 minute averaging period is specified by Parameter 7, then three 8 minute and one 6 minute calculations will occur. The proper weighting of these values in producing the final output is described below.

The Output Processing occurs only at the Output Interval and involves averaging the statistical results obtained at the averaging periods. These final results are then stored sequentially in Input Storage beginning with the location specified in Parameter 9. The Sample Instruction 70 must be used to transfer the data to Final Storage. All but the last averaging period in the Output Interval will contain the same number of input scans as specified by Parameter 7. To insure that results from the averaging periods contribute to the final result proportional to their averaging periods, the output processing uses the following equation:

$$RF = (NR1 + NR2 + ... + N'RL)/NT$$

RF is the final result
R1 R2, etc., are the results from the averaging period processing
RL is the result from the last averaging period in the Output Interval
N   is the number of input samples in the specified averaging period (Parameter 7)
N' is the number of input scans in the last averaging period
NT is the total number of input samples processed in the Output Interval

INTERMEDIATE STORAGE REQUIREMENTS

The number of Intermediate locations will depend upon the number of input values and outputs desired:

1. Define K as the number of input values.
2. Define S as the maximum of either the variances, standard deviations, or C, where

C = K if K < the number of correlations requested, or
C = number of correlations + 1 if K > the number of correlations requested.
3.   Define Q as the maximum of either the covariances or correlations desired.
4.   Define P as the total number of outputs desired.

The amount of intermediate memory locations (IML) required, is then given by:

$$IML = K + S + Q + P + 2$$

EXECUTION TIME
If K, S, and Q are defined as in the previous section, the execution time of the COV/CORR Instruction in milliseconds can be approximated by:

$$T(ms) = 1.1K + 0.5S + 0.9Q + 1.8$$

When evaluating how frequently input samples can be processed by the COV/CORR Instruction (i.e., determining the minimum program table execution interval), the time required to make the measurements and order the input values must be added to the COV/CORR execution time. Two alternatives exist for the measurement portion of the programming. The fastest method is to group as many sensors as possible into the fewest measurement instructions, ignoring the Input location order required by the COV/CORR Instruction. After the measurements are made, use "move" instructions (i.e., 31 and 54) to obtain the proper input order. The slower alternative is to order separate measurement instructions directly as required by the COV/CORR Instruction. While avoiding "move" instructions, this approach uses more measurement instructions. The reason the first method is in general faster is that less overhead time is required in going from one measurement to another within a single instruction (using the "repetitions" feature) than in going from one measurement instruction to another.

In many situations, the CR23X must perform measurement and processing tasks in addition to those associated with the COV/CORR Instruction. Uninterrupted operation of the COV/CORR Instruction is assured by entering it in Program Table 1 (highest priority) and placing the additional tasks in Program Table 2.

A covariance correlation example is given in Section 8.

*** 63 PARAMETER EXTENSION ***

Instruction 63 is used immediately following Instructions 97 or 98 to allow the entry of a variable number of parameters. Instruction 63 can be entered several times in sequence if the number of parameters requires it. There are 8 two digit parameters. Refer to instruction being extended (97, 98) for specifics on the use of Instruction 63.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:-08:	2	Depends upon preceding instruction. Following Instruction 97 RF IDs and Phone No. 1 digit at a time, 32 Between RF IDs, 70 after the last RF ID, 32 and 84 Between RF and DC112 Phone, and 13 To END. Following Instruction 98 (255 character limit) Base 10 value of ASCII character (Appendix E) 00 TO END.

Input locations altered: 0

*** 64 PAROSCIENTIFIC "T" SERIES PROCESSING ***

FUNCTION
Instruction 64 processes measurements made on the Paroscientific "T" Series pressure transducer. The transducer outputs a temperature frequency and a pressure frequency. The frequencies are measured using the Period Averaging Instruction, 27, which returns the period of the temperature and pressure signals in microseconds.
Instruction 64 converts period to temperature and pressure according to the following relationships provided by Paroscientific.

Pressure (psi) =
$$C * (1 - T_0^2/Tau^2) * (1 - (D * (1 - T_0^2/Tau^2)))$$

Temperature (°C) =
$$Y_1U + Y_2U^2 + Y_3U^3$$

where, $C = C_1 + C_2U + C_3U^2$ (psi),

$T_0 = T_1 + T_2U + T_3U^2 + T_4U^3 + T_5U^4$ (microsecond),

$D = D_1 + D_2U$ (microsecond),

$U = U(t) - U_0$ (microsecond),

Tau = measured pressure (microsecond),

U(t) = measured temperature (microsecond).

Values for the calibration coefficients (U_0, Y_1, Y_2, Y_3, C_1, C_2, C_3, D_1, D_2, T_1, T_2, T_3, T_4, T_5) are provided by Paroscientific.

Instruction 64 has two parameters as shown below.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Start Loc: U(t), Tau, U0..T5
02:	4	Destination Loc: Temp($^{\circ}$C), Pressure(psi), Signature

PARAMETER 1 - U(t), Tau, and 14 coefficients provided by Paroscientific must reside in contiguous input locations starting at the location entered in Parameter 1.

PARAMETER 2 - The results of Instruction 64: temperature in $^{\circ}$C, pressure in psi, and the signature of the 14 coefficients are stored in 3 contiguous input locations starting at the location entered in Parameter 2.

Only the first 5 digits of each 7 digit coefficient are entered in the datalogger. The maximum error that occurs due to using five digit coefficients is 0.001 psi throughout a 15 psi range at any temperature between -30° and +60°C. The maximum error in temperature calculations is 0.2°C over the same temperature range.

Coefficients are entered using CR23X Instruction 30 (Z=F) or Instruction 65 (Bulk Load). A calibration sheet from Paroscientific lists the 14 coefficients. Coefficients are entered in the same order as they appear on the calibration sheet. Coefficient U_0 is entered first and T_5 last. $Y_3/1000$ and D_1*1000 are entered for the Y3 and D1 coefficients, respectively. The example below shows the correct format for entering the coefficients.

Example: The 14 coefficients shown below are for Paroscientific "T" Series transducer Serial Number 30135. Your coefficients will be different.

Coeff.	Value	Entry
U_0	5.860253	5.8603
Y_1	-3970.348	-3970.3
Y_2	-7114.265	-7114.3
* Y_3	102779.1	102.78
C_1	70.29398	70.294
C_2	6.610141	6.6101
C_3	-119.2867	-119.29
* D_1	0.0308837	30.884
D_2	0.0	0.0
T_1	26.33703	26.337
T_2	0.8516985	0.85170
T_3	21.80118	21.801
T_4	0.0	0.0
T_5	0.0	0.0

* Y_3 and D_1 coefficients are entered as $Y_3/1000$ and D_1*1000.

The large number of coefficients which must be entered into the program often leads to entry error. As a check on coefficient entry, the third value output by Instruction 64 is the signature of the coefficients. The signature is a calculated number unique to the value and order of the 14 coefficients. For future comparisons, note this signature after entering the coefficients correctly. Signatures will be identical if and only if exactly the same coefficients are entered.

Input locations read: 16
Input locations altered: 3

*** 65 BULK LOAD ***

FUNCTION
Instruction 65 inputs given values in up to eight Input Storage locations.

The Bulk Load instruction has 9 parameters. The first eight are the values to be entered in input storage locations. The ninth is the input location for the first data value; subsequent data values are placed in sequential input locations.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	FP	Param. to be entered
02:	FP	"
03:	FP	"
04:	FP	"
05:	FP	"
06:	FP	"
07:	FP	"
08:	FP	"
09:	4	Starting input location

Input locations altered: 8

*** 66 ARCTAN ***

FUNCTION

Calculate the angle in degrees whose tangent is X/Y. The polarity of X and Y must be known to determine the quadrant of the angle, as shown here. If 0 is entered for Parameter 2, the Arctangent of X is the result (limits of ARCTAN(X) are $-90° <$ ARCTAN $< 90°$).

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location of X [X]
02:	4	Input location of Y [Y]
03:	4	Destination input location for ARCTAN(X/Y) = [Z]

Input locations altered: 1
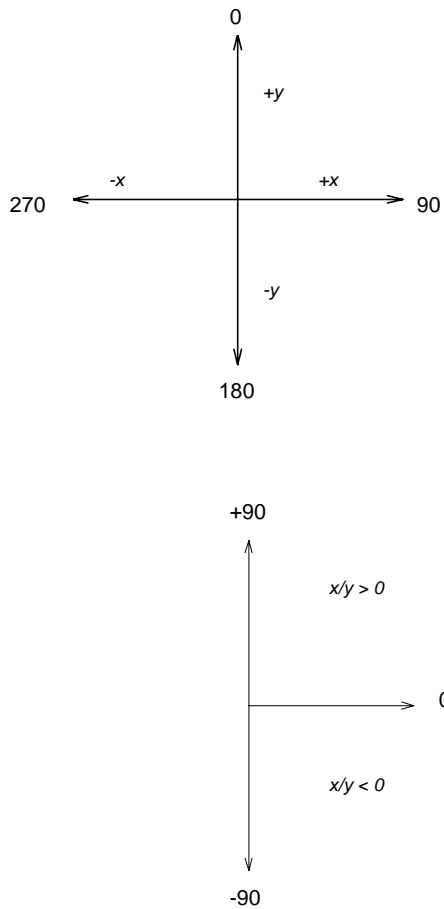
FIGURE 10-1. Quadrant that the Angle Falls in is Defined by the Sign of x and y.

*** 67 DYNAGAGE SAP-FLOW ***

FUNCTION

This instruction is used in conjunction with a Dynamax, Inc. Stem Flow Gage. It processes four voltages measured by instruction 2. See Appendix F for information on Instruction 67.

Input locations altered: 2 or 6

*** 68 EXTENDED PARAMETERS 4 DIGIT ***

FUNCTION

This instruction is used to give other instructions additional parameters. Each of the eight parameters in Instruction 68 is defined by the instruction it follows. Refer to the specific instruction that uses extended parameters.

Input location altered: 0

SECTION 11. OUTPUT PROCESSING INSTRUCTIONS

*** 69 WIND VECTOR ***

FUNCTION
Instruction 69 processes the primary variables of wind speed and direction from either polar (wind speed and direction) or orthogonal (fixed East and North propellers) sensors. It uses the raw data to generate the mean wind speed, the mean wind vector magnitude, and the mean wind vector direction over an output interval. Two different calculations of wind vector direction (and standard deviation of wind vector direction) are available, one of which is weighted for wind speed.

When used with polar sensors, the instruction does a modulo divide by 360 on wind direction, which allows the wind direction (in degrees) to be 0 to 360, 0 to 540, less than 0, or greater than 540. The ability to handle a negative reading is useful in an example where a difficult to reach wind vane is improperly oriented and outputs 0 degrees at a true reading of 340 degrees. The simplest solution is to enter an offset of -20 in the instruction measuring the wind vane, which results in 0 to 360 degrees following the modulo divide.

When a wind speed sample is 0, the instruction uses 0 to process scalar or resultant vector wind speed and standard deviation, but the sample is not used in the computation of wind direction. The user may not want a sample less than the sensor threshold used in the standard deviation. If this is the case instruction 89 can be used to check wind speed, and if less than the threshold, Instruction 30 can set the input location equal to 0.

Standard deviation can be processed one of two ways: 1) using every sample taken during the output period (enter 0 for parameter 2), or, 2) by averaging standard deviations processed from shorter sub-intervals of the output period. Averaging sub-interval standard deviations minimizes the effects of meander under light wind conditions, and it provides more complete information for periods of transition[1].

---

[1] EPA On-site Meteorological Program Guidance for Regulatory Modeling Applications.

Standard deviation of horizontal wind fluctuations from sub-intervals is calculated as follows:

$$\sigma(\Theta)=[((\sigma\Theta_1)^2+(\sigma\Theta_2)^2 ...+(\sigma\Theta_M)^2)/M]^{1/2}$$

where $\sigma(\Theta)$ is the standard deviation over the output interval, and $\sigma\Theta_1 ... \sigma\Theta_M$ are sub-interval standard deviations.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	4	Samples per sub-interval (number of scans, enter 0 for no sub-interval)

03:	2	Sensor/Output 2 digits: AB
		A Sensor type:
		    0 = Speed and Direction
		    1 = East and North
		B Output option:
		    0 S, $\Theta 1$, $\sigma(\Theta 1)$
		    1 S, $\Theta 1$
		    2 S, U, Θu, $\sigma(\Theta u)$

04:	4	First wind speed input location no. (East wind speed)
05:	4	First wind direction input location no. (North wind speed)

Outputs Generated: 2-4 (depending on output option) per repetition

A sub-interval is specified as a number of scans. The number of scans for a sub-interval is given by:

  Desired sub-interval (secs) / scan rate (secs)

In an example where the scan rate is 1 second and the Output Flag is set every 60 minutes, the standard deviation is calculated from all 3600 scans when the sub-interval is 0. With a sub-interval of 900 scans (15 minutes) the standard deviation is the average of the four sub-interval standard deviations. The last sub-interval is weighted if it does not contain the specified number of scans.

11-1

There are three Output Options that specify the values calculated.

Option 0:

Mean horizontal wind speed, S.
Unit vector mean wind direction, $\Theta 1$.
Standard deviation of wind direction, $\sigma(\Theta 1)$.

Standard deviation is calculated using the Yamartino algorithm. This option complies with EPA guidelines for use with straight-line Gaussian dispersion models to model plume transport.

Option 1:

Mean horizontal wind speed, S.
Unit vector mean wind direction, $\Theta 1$.

Option 2:

Mean horizontal wind speed, S.
Resultant mean wind speed, U.
Resultant mean wind direction, Θu.
Standard deviation of wind direction, $\sigma(\Theta u)$.

This standard deviation is calculated using Campbell Scientific's wind speed weighted algorithm.

Use of the Resultant mean horizontal wind direction is not recommended for straight-line Gaussian dispersion models, but may be used to model transport direction in a variable-trajectory model.

Measured raw data:

S_i = horizontal wind speed
Θ_i = horizontal wind direction
Ue_i = east-west component of wind
Un_i = north-south component of wind
N = number of samples

Calculations:

Scalar mean horizontal wind speed, S:
$$S=(\Sigma S_i)/N$$
where in the case of orthogonal sensors:
$$S_i=(Ue_i^2+Un_i^2)^{1/2}$$

Unit vector mean wind direction, $\Theta 1$:
$$\Theta 1=\text{Arctan }(Ux/Uy)$$

where

$$Ux=(\Sigma \sin \Theta_i)/N$$
$$Uy=(\Sigma \cos \Theta_i)/N$$

or, in the case of orthogonal sensors

$$Ux=(\Sigma(Ue_i/U_i))/N$$
$$Uy=(\Sigma(Un_i/U_i))/N$$
$$\text{where } U_i=(Ue_i^2+Un_i^2)^{1/2}$$

Standard deviation of wind direction, $\sigma(\Theta 1)$, using Yamartino algorithm:

$$\sigma(\Theta 1)=\text{arc sin}(\varepsilon)[1+0.1547\ \varepsilon^3]$$

where,

$$\varepsilon=[1-((Ux)^2+(Uy)^2)]^{1/2}$$

and Ux and Uy are as defined above.

Resultant mean horizontal wind speed, U:

$$U=(Ue^2+Un^2)^{1/2}$$

where for polar sensors:

$$Ue=(\Sigma S_i \text{ Sin } \Theta_i)/N$$
$$Un=(\Sigma S_i \text{ Cos } \Theta_i)/N$$

or, in the case of orthogonal sensors:

$$Ue=(\Sigma Ue_i)/N$$
$$Un=(\Sigma Un_i)/N$$

Resultant mean wind direction, Θu:

$$\Theta u=\text{Arctan }(Ue/Un)$$

Standard deviation of wind direction, $\sigma(\Theta u)$, using Campbell Scientific algorithm:

$$\sigma(\Theta u)=81(1-U/S)^{1/2}$$

*** 70 SAMPLE ***

FUNCTION
This instruction stores the value from each specified input location. The value(s) stored are those in the input location(s) when Instruction 70 is executed with the Output Flag set high.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Repetitions
02:	4	Starting input location no.
Outputs Generated:		1 for each repetition

*** 71 AVERAGE ***

FUNCTION
This instruction stores the average value over
the given output interval for each input location
specified.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	4	Starting input location no.

Outputs Generated: 1 per repetition

*** 72 TOTALIZE ***

FUNCTION
This instruction stores totalized value over the given
output interval for each input location specified.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	4	Starting input location no.

Outputs Generated: 1 per repetition

*** 73 MAXIMUM ***

FUNCTION
This instruction stores the MAXIMUM value taken
(for each input location specified) over a given
output interval. An internal FLAG is set whenever a
new maximum value is seen. This FLAG may be
tested by Instruction 79. Time of occurrence
maximum value(s) is OPTIONAL output
information, which is formatted and activated by
entering one of the following CODES for Param. 2.

Code	Options
00	Output value ONLY
01	Output value with SECONDS
10	Output value with HOUR-MINUTE
11	Output value with HR-MIN,SEC

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Time of maximum (optional)
03:	4	Starting input location no.

Outputs Generated: 1 per repetition
(1 or 2 additional outputs per repetition with time
option)

*** 74 MINIMUM ***

FUNCTION
Operating in the same manner as Program 73,
this instruction is used for storing the MINIMUM
value (for each input location specified) over a
given output interval.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Time of minimum (optional)
03:	4	Starting input location no.

Outputs Generated: 1 per repetition
(1 or 2 additional outputs per repetition with time
option)

*** 75 STANDARD AND WEIGHTED ***
VALUE HISTOGRAM

FUNCTION
Processes input data as either a standard
histogram (frequency distribution) or a weighted
value histogram.

The standard histogram outputs the fraction of
the Output Interval that the value in a specified
input location (defined as the bin select value) is
within a particular sub-range of the total
specified range. A counter in the bin associated
with each sub-range is incremented whenever
the value falls within that sub-range. The value
which is output to Final Storage for each bin is
computed by dividing the accumulated total in
each bin by the total number of scans. This
form of output is also referred to as a frequency
distribution.

The weighted value histogram uses data from 2
input locations. One location contains the bin
select value; the other contains the weighted
value. Each time the instruction is executed,
the weighted value is added to a bin. The sub-
range that the bin select value is in determines
the bin to which the weighted value is added.
When the Output Flag is set, the value
accumulated in each bin is divided by the
TOTAL number of input scans to obtain the
values that are output to Final Storage. These
values are the contributions of the sub-ranges
to the overall weighted value. To obtain the
average of the weighted values that occurred

while the bin select value was within a particular sub-range, the value output to Final Storage must be divided by the fraction of time that the bin select value was within that particular sub-range (i.e., a standard histogram of the bin select value must also be output).

For either histogram, the user must specify: 1) the number of repetitions, 2) the number of bins, 3) a form code specifying whether a closed or open form histogram is desired (see below), 4) the bin select value input location, 5) the weighted value input location (see below), 6) the lower range limit, and 7) the upper range limit.

The standard histogram (frequency distribution) is specified by entering "0" in the weighted value input location parameter. Otherwise, this parameter specifies the input location of the weighted value. When more than one repetition is called for, the bin select value location will be incremented each repetition and the weighted value location will remain the same (same weighted value sorted on the basis of different bin select values). The weighted value location will be incremented if it is entered as an indexed location (key "C" at some point while keying in Parameter 5; two dashes, --, will appear on the right of the display). At the user's option, the histogram may be either closed or open. The open form includes all values below the lower range limit in the first bin and all values above the upper range limit in the last bin. The closed form excludes any values falling outside of the histogram range.

The difference between the closed and open form is shown in the following example for temperature values:

Lower range limit	10°C
Upper range limit	30°C
Number of bins	10

	Closed Form	Open Form
Range of first bin	10 to 11.99 deg.	<12 deg.
Range of last bin	28 to 29.99 deg.	>28 deg.

A common use of a closed form weighted value histogram is the wind speed rose. Wind speed values (the weighted value input) are accumulated into corresponding direction sectors (bin select input).

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	4	Number of bins
03:	2	Form code (0=open form, 1=closed form)
04:	4	Bin select value input location no.
05:	4	Weighted value input location no. (0 = frequency distribution option)
06:	FP	Lower limit of range
07:	FP	Upper limit of range

Outputs Generated: Number of Bins * Repetitions

*** 77 RECORD REAL TIME ***

FUNCTION
This Instruction stores the current time in Final Storage. At midnight the clock rolls over from 23:59 to 00:00. The day also changes.

If hourly or daily summary data is output, it may be desirable to have the previous day output, since that is when the measurements were made. Entering a 2 for the day code causes the previous day to be output if it is the first minute of the day. Similarly, entering 2 for the hour-minute code causes 2400 instead of 0000 to be output (the next minute is still 0001). When day and hour-minute are both output, a 2 for either code results in the previous day at 2400.

If year is output along with a 2 option in day or hour-minute, the previous year will be output during the first minute of the new year.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Enter appropriate TIME option code

Outputs Generated: 1 for each time parameter selected

Code	Result
xxx1	SECONDS (with resolution of 0.125 sec.)
xx1x	HOUR-MINUTE
xx2x	HOUR-MINUTE, 2400 instead of 0000
x1xx	JULIAN DAY
x2xx	JULIAN DAY, previous day during first minute of new day
1xxx	YEAR

Any combination of Year, Day, HR-MIN, and seconds is possible (e.g., 1011: YEAR, HR-MIN, SEC).

*** 78 SET HIGH OR LOW RESOLUTION *** DATA STORAGE FORMAT

FUNCTION

This instruction enables high resolution (5 character) or low resolution (4 character) final data storage format. Instruction 78 should be entered ahead of the output instructions for which the specified resolution is desired. The default format is low resolution. At the beginning of each program table execution, the low resolution format is automatically enabled. See Table 2.2-1 for details.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	0 = low resolution; 1 = high resolution

Outputs Generated: 0

*** 79 SAMPLE ON MAXIMUM OR *** MINIMUM

FUNCTION

Instruction 79 samples specified input location values at the time a new maximum or minimum value is detected by a previous Maximum (73) or Minimum (74) Instruction. When the Output Flag is set, the values copied to Intermediate Storage are transferred to Final Storage.

Instruction 79 must directly follow the maximum or minimum Instruction to which it refers. If the previous Instruction 73 or 74 has more than 1 repetition, Instruction 79 samples whenever a new maximum or minimum is detected in any of the locations. If sampling is to occur only when a specific input location shows a new maximum or minimum, the previous Maximize or Minimize Instruction should have one rep referring to that input location.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions (number of sequential locations to sample)
02:	4	Starting input location no.

Outputs Generated: 1 per repetition

*** 80 SET ACTIVE STORAGE AREA ***

FUNCTION

Instruction 80 is used to redirect Output data to either of the Final Storage areas or to Input Storage and to set the array ID for Final Storage. At the beginning of each table the Active Output area is set to Final Storage Area 1.

When directed to Final Storage, the second parameter can be used to set the output array ID. Instruction 80 should follow the instruction setting flag 0. If parameter 2 is 0, the array ID is determined by the instruction location number of Instruction 80 or by the instruction that set the Output Flag, whichever comes last.

> **NOTE:** When a P80 is used to set any array ID, all array IDs should be set to unique numbers.

When data are sent to Input Storage, no array ID is sent. Output array ID's can be indexed within a loop.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Storage area option 00 or 01 = Final Storage Area 1 02 = Final Storage Area 2 03 = Input Storage Area
02:	4	Starting input location destination if option 03 Output Array ID if options 0-2 (1-511 are valid IDs)

*** 81 RAINFLOW HISTOGRAM ***

FUNCTION

Instruction 81 implements the rainflow counting algorithm, essential to estimating cumulative damage fatigue to components undergoing stress/strain cycles. Data can be provided by making measurements in either the standard or

the burst mode. The Rainflow Instruction can process either a swath of data following the burst mode, or it can process "on line" similar to other processing instructions.

The output is a two dimensional Rainflow Histogram for each sensor or repetition. One dimension is the amplitude of the closed loop cycle (i.e., the distance between peak and valley); the other dimension is the mean of the cycle (i.e.,[peak value + valley value]/2). The value of each element (bin) of the histogram can be either the actual number of closed loop cycles that had the amplitude and average value associated with that bin or the fraction of the total number of cycles counted that were associated with that bin (i.e., number of cycles in bin divided by total number of cycles counted).

The user enters the number of mean bins, the number of amplitude bins, and the upper and lower limits of the input data.

The values for the amplitude bins are determined by difference between the upper and lower limits on the input data and by the number of bins. For example, if the lower limit is 10 and the upper limit is 150, and there are 5 amplitude bins, the maximum amplitude is 150 - 100 = 50. The amplitude change between bins and the upper limit of the smallest amplitude bin is 50/5 = 10. Cycles with an amplitude, A, less than 10 will be counted in the first bin. The second bin is for $10 \leq A < 20$, the third for $20 \leq A < 30$, etc.

In determining the ranges for mean bins, the actual values of the limits as well as their difference are important. The lower limit of the input data is also the lower limit of the first mean bin. Assume once again that the lower limit is 100, the upper limit 150, and that there are 5 mean bins. In this case the first bin is for cycles which have a mean value M, $100 \leq M < 110$, the second bin $110 \leq M < 120$, etc.

If $C_{m,a}$ is the count for mean range m and amplitude range a, and M and N are the number of mean and amplitude bins respectively, then the output of one repetition is arranged sequentially as ($C_{1,1}$, $C_{1,2}$, ... $C_{1,N}$, $C_{2,1}$, $C_{2,2}$, ... $C_{M,N}$). Multiple repetitions are sequential in memory. Shown in two dimensions, the output is:

$$
\begin{array}{cccc}
C_{1,1} & C_{1,2} & \cdots & C_{1,N} \\
C_{2,1} & C_{2,2} & \cdots & C_{2,N} \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot \\
C_{M,1} & C_{M,2} & \cdots & C_{M,N}
\end{array}
$$

The histogram can have either open or closed form. In the open form, a cycle that has an amplitude larger than the maximum bin is counted in the maximum bin; a cycle that has a mean value less than the lower limit or greater than the upper limit is counted in the minimum or maximum mean bin. In the closed form, a cycle that is beyond the amplitude or mean limits is not counted.

The minimum distance between peak and valley, parameter 8, determines the smallest amplitude cycle that will be counted. The distance should be less than the amplitude bin width ([high limit - low limit]/no. amplitude bins) or cycles with the amplitude of the first bin will not be counted. However, if the value is too small, processing time will be consumed counting "cycles" which are in reality just noise.

More than one Rainflow Histogram can be calculated using the Repetitions parameter. The swath of input data, the size of the mean and amplitude dimensions, the low and high limits of the input data, and minimum distance between peaks and valley are all selectable by the user with parameters.

Data are output to Final Storage or to Input Storage for further processing when the datalogger's Output Flag is set. Partial accumulations are kept in Intermediate Storage.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	4	Input location of input data
03:	4	Swath (per rep) of data at input location; 1 = on line processing
04:	4	No. of Mean bins
05:	4	No. of Amplitude Bins
06:	FP	Low limit of input data
07:	FP	High limit of input data
08:	FP	Minimum distance between peak and valley

09:	2	Option (AB)
	A:	Form 0 = closed, 1 = open form
	B:	Output 0 = fraction, 1 = counts
10:	4	Input location to start storing histogram. Enter 0 to send output directly to Final Storage.

Execution time: 6.5 - 7.0 ms, with 60 Amplitude Bins and one Mean Bin.

Intermediate Storage locations required: Reps x (Bins+2 x [No. of Amplitude Bins] + 4), where Bins = No. Mean Bins x No. Amplitude Bins.

Outputs Generated: No. Mean Bins x No. Amplitude Bins x Reps

*** 82 STANDARD DEVIATION IN TIME ***

FUNCTION
Calculate the standard deviation (STD DEV) of a given input location. The standard deviation is calculated using the formula:

$$S = ((\Sigma X_i^2 - (\Sigma X_i)^2/N)/N)^{1/2}$$

where X_i is the ith measurement and N is the number of samples.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	4	Starting input location no.

Outputs Generated: 1 per repetition

SECTION 12. PROGRAM CONTROL INSTRUCTIONS

TABLE 12-1. Flag Description

Flag 0	Output Flag
Flag 1 to 8	User Flags
Flag 11 to 18	User Flags
Flag 9	Intermediate Processing Disable Flag

TABLE 12-2. Command Codes

0	Go to end of program table[3]
1-9, 79-99	Call Subroutine 1-9, 79-99[1]
10-19	Set Flag 0-9 high
111-118	Set Flag 11 to 18 high
20-29	Set Flag 0-9 low
211-218	Set Flag 11 to 18 low
30	Then Do
31	Exit loop if true
32	Exit loop if false
40	RS232 Port On
41-48	Set Port 1-8 high[2]
49	Set Switched 12 V high
50	RS232 Port Off
51-58	Set Port 1-8 low[2]
59	Set Switched 12 V low
61-68	Toggle Port 1-8[2]
71-78	Pulse Port 1-8[2]

[1]   95, 96, 97, and 98 are special subroutines which can be called by Control ports 5, 6, 7, and 8 going high; see Instruction 85 for details.

[2]   The ports can be indexed to the loop counter.

[3]   If this command is executed while in a subroutine, execution jumps directly to the end of the table that called the subroutine.

*** 83 IF CASE X < F ***

FUNCTION
If the value in the location specified in the Begin Case Instruction 93 is less than the fixed value entered as parameter 1 then execute the command in parameter 2 then go to the end of the case statement when the next Instruction 83 occurs. Else, continue to next instruction. See Instruction 93 for an example.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	FP	Fixed value
02:	2	Command

*** 85 LABEL SUBROUTINE ***

FUNCTION
This instruction marks the start of a subroutine. Subroutines are a series of instructions beginning with Instruction 85 and terminated with Instruction 95, END. All subroutines must be placed in Table 3 (Subroutine Table). When a subroutine is called by a command in a Program Control Instruction, the subroutine is executed, then program flow continues with the instruction following that which called the subroutine.

Subroutines may be called from within other subroutines (nested). The maximum nesting level for subroutines is 7 deep. Attempts to nest more than 7 deep will not be detected at compilation, but will result in a run time error. When the sixth subroutine attempts to call the seventh, error 31 will be displayed. Execution will not branch to the seventh subroutine; it will continue with the Instruction following that calling the subroutine.

98, 97, 96, 95 PORT INTERRUPT SUBROUTINES. If subroutine 98 (97, 96, 95) is included in Table 3 then Port 8 (7, 6, 5) will cause an interrupt on the rising edge and the subroutine will be executed. These subroutines can also be called from any table.

These subroutines can interrupt Table 1 or 2 or can occur when neither Table is being executed. When the port activating 95, 96, 97, or 98 goes high during the execution of a table, the instruction being executed is completed before the subroutine is run (i.e., it is as if the subroutine was called by the next instruction).

The priority is 98, 97, 96, 95, Table 1, Table 2. If all four interrupt subroutines are pending (ports go high at the same time or go high during the execution of the same instruction in one of the tables), 98 will be executed first, then 97, then 96, and finally 95.

While 95, 96, 97, or 98 is being executed as a result of the respective port going high, that port interrupt is disabled (i.e., the subroutine must be completed before the port going high will have any effect).

SECTION 12. PROGRAM CONTROL INSTRUCTIONS

> **NOTE:** Do not set the output flag in an interrupt subroutine unless it is the only place the output flag is set and data are output.

> **NOTE:** If Control Ports 5, 6, 7, or 8 are used for pulse measurements or interrupt subroutines, the CR23X will go into a 10 mA power drain state.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Subroutine number (1-9, 79-99)

*** 86 DO ***

FUNCTION
This Instruction unconditionally executes the specified command.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Command (Table 12-2)

*** 87 LOOP ***

FUNCTION
Instructions included between the Loop Instruction and the End Instruction (95) are repeated the number of times specified by the iteration count (Parameter 2), or until an Exit Loop command (31,32) is executed by a Program Control Instruction within the Loop. If 0 is entered for the count, the loop is repeated until an Exit Loop command is executed.

The first parameter, delay, controls how frequently passes through the loop are made. Its units are multiples of the table execution interval. A delay of 0 means that there is no delay between passes through the loop. Each time the table is executed all iterations of the loop will be completed and execution will pass on to the following instructions. If the delay is 5, every fifth time that the execution interval comes up, one pass through the loop is made; only those instructions in the loop will be executed and other portions of the table are not executed in the interim. When a loop with delay is executed, the next execution starts at the loop, skipping over any previous and following instructions in the table.

When a fixed number of iterations are executed, the time spent in the loop is equal to the product of the execution interval, delay, and the number of iterations. For example, a loop with a delay of 1 and a count of 5 will take 5 seconds if the execution interval is 1 second. When the loop is first entered, one pass through the loop is made, then the CR23X delays until the next execution interval and makes the second pass through the loop. After making the fifth pass through the loop, there is the fifth delay, after which execution passes to the instruction following the END instruction which goes with the loop.

While in a loop with delay, the table will not be initiated at each execution interval. (However, the overrun decimals will not be displayed.) Some consequences of this are: The Output Flag will not be automatically cleared between passes through the loop. Because Table 2 cannot interrupt Table 1, Table 2 will not be executed while Table 1 is in a loop with delay. Table 1 will not interrupt Table 2 in the middle of an output array. Thus, if the Output Flag is set in Table 2 prior to entering the loop or within the loop, the flag must be specifically cleared before the end of the pass if Table 1 is to be executed.

Input locations for Processing Instructions within a loop can be entered as Indexed locations. An Indexed location causes the input location to be incremented by 1 with each pass through the loop. (The Index counter is added to the location number in the program table.) Input locations which are not indexed will remain constant.

To specify an Indexed location, depress the C key at some point while keying in the digits for the input location and before entering the location with the A key. Two dashes, --, appear in the two right most characters of the display, indicating the entry is Indexed.

When the same output processing is required on values in sequential input locations, it must be accomplished by using the repetitions parameter of the Output Instruction, not by indexing the input location within a loop.

An Output Instruction within a loop is allotted the same number of Intermediate Storage locations as it would receive if it were not in the loop. For example, the average instruction with a single repetition is allotted only two Intermediate

locations: one for the number of samples and one for the running total. Each time through the loop the sample counter is incremented and the value in the referenced input location is added to the total. If the input location is indexed, the values from all input locations are added to the same total.

Note that if the Output Flag is set prior to entering the loop in the above example, 10 values will be output. The first will be the average of all the readings in locations 1-10 since the previous output. Because the Intermediate locations are zeroed each time an output occurs, the next nine values will be the current values (samples at the time of output) of locations 2-10.

Loops can be nested. Indexed locations within nested loops are indexed to the inner most loop that they are within. The maximum nesting level in the CR23X is 11 deep. This applies to If Then/Else comparisons and Loops or any combination thereof. An If Then/Else comparison which uses the Else Instruction (94) counts as being nested 2 deep.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Delay
02:	4	Iteration count

The following example involves the use of the Loop Instruction, without a delay, to perform a block data transformation.

The user wants 1 hour averages of the vapor pressure calculated from the wet- and dry-bulb temperatures of 5 psychrometers. One pressure transducer measurement is also available for use in the vapor pressure calculation.

1. The input locations are assigned as follows:

   a) pressure
      Location 10

   b) dry-bulb temperatures
      Location 11-15

   c) wet-bulb temperatures
      Location 16-20

   d) calculated vapor pressure
      Location 21-25
      (Vapor pressure is written over the wet-bulb temperatures.)

2. The program flow is as follows:

   a) Enter the Loop Instruction (#87) with Delay=0 and iteration count=5.

   b) Calculate the vapor pressure with Instruction 57 using a normal location entry of 10 for atmospheric pressure and Indexed locations of 11, 16 and 16 for the dry-bulb, wet-bulb, and calculated vapor pressure, respectively.

   c) End loop with Instruction 95.

   d) Use the If Time Instruction (#92) to set the Output Flag every hour.

   e) Use the Average Instruction (#71) with 5 repetitions starting at input location 21 to average the vapor pressure over the hour.

The actual keyboard entries for the examples are shown below with the first example Instruction location equal to 10. The Input Instructions to make the pressure and temperature measurements are assumed.

TABLE 12-3. Loop Example: Block Data Transform

1: Beginning of Loop (P87)
| 1: | 0 | Delay |
| 2: | 5 | Loop Count |

2: Wet/Dry Bulb Temp to VP (P57)
1:	10	Pressure Loc [Pressure]
2:	11--	Dry Bulb Loc [DryBulb1]
3:	16--	Wet Bulb Loc [WetBulb1]
4:	21--	Loc [VP1]

3: End (P95)

4: If time is (P92)
1:	0	Minutes (Seconds --) into a
2:	60	Interval (same units as above)
3:	10	Set Output Flag High (Flag 0)

5: Average (P71)
| 1: | 5 | Reps |
| 2: | 21 | Loc [VP1] |

The Loop with a delay may be used so that only those instructions within the Loop are executed while certain conditions are met. As a simple example, suppose it is desired to execute one set of instructions from midnight until 6 AM, another set between 6 AM and 4 PM, and a third set between 4 PM and midnight. Between 6 AM and 4 PM, samples are desired every 10

seconds; the rest of the time one minute between samples is sufficient. The execution interval is set to 10 seconds; when a one minute sample rate is desired, a delay of 6 (6 x 10s = 60s) is used in the loop.

TABLE 12-4. Example: Loop with Delay

1: Beginning of Loop (P87)
    1:      6   Delay
    2:      0   Loop Count

2: Do (P86)
    1:      1   Call Subroutine 1

3: If (X<=>F) (P89)
    1:     25   X Loc [Day   ]
    2:      3   >=
    3:      6   F
    4:     31   Exit Loop if True

4: End (P95)

5: Beginning of Loop (P87)
    1:      1   Delay
    2:      0   Loop Count

6: Do (P86)
    1:      1   Call Subroutine 1

7: If (X<=>F) (P89)
    1:     25   X Loc [Day   ]
    2:      3   >=
    3:     16   F
    4:     31   Exit Loop if True

8: End (P95)

9: Beginning of Loop (P87)
    1:      6   Delay
    2:      0   Loop Count

10: Do (P86)
    1:      1   Call Subroutine 1

11: If (X<=>F) (P89)
    1:     25   X Loc [Day   ]
    2:      3   >=
    3:      5   F
    4:     32   Exit Loop if False

12: End (P95)

*Table 3 Subroutines

1: Beginning of Subroutine (P85)
    1:      1   Subroutine 1

2: Time (P18)
    1:      2   Hours into current year
                   {maximum 8784)
    2:     24   Mod/By
    3:     25   Loc [Day   ]

3: End (P95)

*** 88 IF X COMPARED TO Y ***

FUNCTION
This Instruction compares two input locations and, if the result is true, executes the specified Command. The comparison codes are given in Table 12-5.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location for X
02:	2	Comparison code (Table 12-5)
03:	4	Input location for Y
04:	2	Command (Table 12-2)

TABLE 12-5. Comparison Codes

Parameter 1	Function
1	IF X = Y
2	IF X $\neq$ Y
3	IF X $\geq$ Y
4	IF X < Y

*** 89 IF X COMPARED TO F ***

FUNCTION
This Instruction compares an input location to a fixed value and, if the result is true, performs the specified Command. The comparison codes are given in Table 12-5.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location for X
02:	2	Comparison code (Table 12-5)
03:	FP	Fixed value
04:	2	Command (Table 12-2)

*** 90 STEP LOOP INDEX ***

FUNCTION
When used within a Loop (Instruction 87), Instruction 90 will increment the index counter by a specified amount after the first time through the loop, thus affecting all indexed input

location parameters in subsequent instructions. For example, if 4 is specified, the index counter will count up by 4 (0,4,8,12,...) inside the loop. Instruction 90 does not affect the loop counter which still counts by 1.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Increment for the loop index counter

*** 91 IF FLAG / PORT / MODEM ***

FUNCTION
This Instruction checks the status of one of the eighteen Flags, one of the eight ports, or the serial I/O port modem enable and conditionally performs the specified Command. Ports can be indexed (--) with the C key.

The first Parameter specifies the condition to check:

1X	Execute command if Flag X is high
2X	Execute command if Flag X is low
4Y	Execute command if port Y is high
5Y	Execute command if port Y is low
40	Execute command if CS I/O modem is on
50	Execute command if CS I/O modem is off
49	Execute command if SW12V is off
59	Execute command if SW12V is off

X = 0-9, 11-18   Y = 1-8

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Flag or Port condition to check
02:	2	Command (Table 12-2)

*** 92 IF TIME IS ***

FUNCTION
The user specifies the number of minutes or seconds into an interval, the duration of the interval, and a command. The command is executed each time the real time is the specified time into the interval. The "If" condition will always be false if 0000 is entered as the time interval.

The time interval is synchronized with real time; if a 60 minute time interval is specified with 0 minutes into the interval, the Command will be executed each hour on the hour. The time interval is synchronized internally by making a modulo divide (Instruction 46) of the number of minutes since midnight by the specified real time interval. If the result is 0, the interval is up. Thus, the first interval of the day always starts at midnight (0 minutes). The maximum interval is 1440 minutes.

The time into an interval is only true the first time Instruction 92 is executed within a given minute (or second). For example, if the command is to set the Output Flag at 0 minutes into a 10 minute interval, and the execution interval of the table is 10 seconds, every 10 minutes there will only be one output generated by this instruction, not five.

The time into interval and the interval may be entered in seconds for intervals less than 60 seconds. Error 92 will occur if these parameters are too large.

To enter the times in seconds, press "C" after keying in the number of seconds into the interval for Parameter 1; two dashes will appear to the right of the number (XXXX--). When the time into interval is entered as seconds (XXXX--), the time interval will also be interpreted as seconds.

The maximum number of seconds that can be entered is 59 for Parameter 1 and 60 for Parameter 2.

The Output Flag (Flag 0) is a special case in that it will automatically be cleared if it is not time to set it.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	4	Time into interval (minutes or seconds if entered XXXX--)
02:	4	Time interval (units same as above)
03:	2	Command (Table 12-2)

*** 93 BEGIN CASE STATEMENT ***

The value in the specified input location is compared against parameters in following If Case instructions (83). When a comparison is true, the command in the If Case instruction is executed and the program flow goes to the End instruction (95) associated with the Begin Case instruction.

```
PARAM.   DATA
NUMBER   TYPE      DESCRIPTION

01:        4        Input location for
                    subsequent comparisons
```

EXAMPLE:

```
1:  CASE (P93)
  1:       2        Case Loc [ ValueX   ]

2:  If Case Location < F (P83)
  1:     69.4       F
  2:       3        Call Subroutine 3
;else
3:  If Case Location < F (P83)
  1:      72        F
  2:      10        Set Output Flag High (Flag 0)
;else
4:  If Case Location < F (P83)
  1:     77.3       F
  2:      30        Then Do

5:  Z=F (P30)
  1:      0.0       F
  2:       0        Exponent of 10
  3:      25        Z Loc [ ValueY   ]

6:  End (P95) ;of Then Do

7:  End (P95) ;of Case Statement
```

*** 94 ELSE ***

FUNCTION
When Command 30 (Then/Else) is used with an If Instruction, the Else Instruction is used to mark the start of the instructions to execute if the test condition is false (Figure 3.8-1). The Else Instruction is optional; when it is omitted, a false comparison will result in execution branching directly to the End Instruction. Instruction 94 has no parameters.

*** 95 END ***

FUNCTION
Instruction 95 is used to indicate the end/return of a subroutine (Instruction 85), the end of a loop (Instruction 87), the end of an If Then/Else sequence (Instructions 88-92 when used with command 30), or the end of a Case sequence (Instruction 93). The End Instruction has no parameters.

*** 96 ACTIVATE SERIAL DATA OUTPUT ***

FUNCTION
Instruction 96 is used to activate Storage Module (SM192/SM716 or Card Storage Module) or serial data (printer) output. Some applications require the Computer RS232 port be left on continuously for proper operation. The Computer RS232 port can be configured to stay on continuously in the ∗ D mode as described in Section 1.8.7.

Normally Instruction 96 is placed in the program table after all Output Instructions have been entered and is executed each time the Table is executed. In this situation any data sent to Final Storage is output as soon as possible. However, by using Program Control Instructions to allow execution of Instruction 96 only at certain times, the user can control when the output device(s) are active. Instruction 96 allows a choice of serial data format and the selection of Addressed or Pin Enabled device for the serial print output.

A single parameter is used to select whether the instruction is to control the "printer", Storage Module, or Card Storage Module output, and if the printer is selected, the format and baud rate. The Instruction must be entered separately for each device that is to receive output.

If both Final Storage areas are in use, Instruction 96 will send data from the area which is currently active. Final Storage Area 1 is active at the start of each Table. Instruction 80 can be used to change the active area. The Area set by Instruction 80 remains active until changed by another Instruction 80 or the Table ends (at which time Area 1 becomes the active Area 1). Instruction 80 can also direct output to Input Storage, in which case Instruction 96 assumes Final Storage Area 1.

If the CR23X is already communicating on a 9-pin serial port when Instruction 96 is executed, the output request is put in a queue and program execution continues. As a 9-pin serial port becomes available, each device in the queue will get its turn.

The request is not put in the queue if the same device is already in the queue. The data contained in the queue (and which determine a unique entry) are baud rate (if applicable), and

the Final Storage Area. Instruction 98 to send characters also uses this queue.

When an entry reaches the top of the queue, the CR23X sends all data accumulated since the last transfer to the device up to the location of the DSP at the time the device became active (this allows everything in the queue to get a turn even if data is being stored faster than it can be transferred to a particular device).

The "other Final Storage Area" device option (the non-active area) allows a "fast" Final Storage area to be transferred to the main area on some trigger condition so there is some history recorded prior to the trigger condition (Section 8.8). The source of data is the currently active Final Storage Area set by Instruction 80 (default = 0 or 1).

> **NOTE:** All memory pointers are positioned to the DSP location when the datalogger compiles a program. For this reason, always retrieve uncollected data before making program changes.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Option Device

TABLE 12-6. Output Device and Baud Rate Codes

Code	Baud Rate
0	300
1	1200
2	9600
3	76800
4	2400
5	4800
6	19200
7	38400

ADDRESSED PRINT DEVICE, y = Baud code
1y = Printable ASCII
2y = Comma Separated ASCII
3y = Binary Final Storage Format
7N = Storage Module N (N=1-8; Section 4.4.2) (Stored in Binary Format)
7N-- = Output File Mark to Storage Module N

SERIAL PRINTER, COMPUTER, OR PIN-ENABLED PRINT DEVICE, y = Baud code (SDE pulled high)
4y = Printable ASCII (CS I/O)
4y-- = Printable ASCII (RS-232)
5y = Comma Separated ASCII (CS I/O)
5y-- = Comma Separated ASCII (RS-232)
6y = Binary Final Storage Format (CS I/O)
6y-- = Binary Final Storage Format (RS-232)

TRANSFER DATA TO OTHER FINAL STORAGE AREA
80 = New data only
81 = All data

*** 97 INITIATE ***
TELECOMMUNICATIONS

Instruction P97 enables the CR23X to initiate telecommunications with

1) a PC via telephone, radio, or generic modem ("Callback")
2) a pager system
3) a human via synthesized voice
4) another datalogger

Several documents supplement information in this section.

1) Appendix G of the CR23X manual covers telephone and generic modem applications
2) Appendix G of the CR23X manual covers pager applications
3) COM300 Voice Modem Manual covers voice applications
4) Appendix H of the CR23X Manual covers calling applications wherein another datalogger is called

Telephone support is also available from Campbell Scientific.

Theory of Operation

When the CR23X executes P97, it looks to see if the interrupt disable flag is low. The interrupt disable flag is one of User Flags 1-8 dedicated in P97 for this use. If the interrupt disable flag is low, the CR23X initiates a call according to protocols programmed into P97. If a computer is answering the call, the CR23X sends a unique 3 digit code. If the computer returns the same 3 digit code, the CR23X keeps the link open and waits for subsequent commands from the computer. If the code is not returned, the datalogger hangs up. P97 can be programmed to try the call again later. The number of failed calls is recorded in a CR23X input location. The failure counter is reset to zero when P97 is executed if the prior call was successful on its first try. The value in the failure input location

can be output to CR23X final storage. Doing so will help evaluate the success of P97.

P97 should not be placed in a conditional statement or subroutine, but rather controlled by controlling the interrupt disable flag. P97 must be executed twice for each call. Once to initiate the call and once more to reset after a successful call or after retries have expired.

Parameter 1 Modem / Baud Rate

Consult Table 12-7 for options. Match the option to the modem connected to the CR23X serial port. Options 20 - 47 (telephone modems) automatically send out the following commands:

ATV0 <CR><LF>   Request numeric
                     responses

ATS7=180 <CR><LF> Wait 180 s for carrier
                     after dialing

ATDT   Dial phone number with
                     tones

Phone numbers, RF paths, or generic modem dialing strings are entered with one or more P68 instructions immediately following the P97. See discussion below for details on the use of P68.

Parameter 2 Interrupt Disable Flag

Specify which User Flag (1 - 8) to use to interrupt and disable calls. When set low, calling is enabled. When set high, calling is interrupted and / or disabled. Care must be taken in programming logic to control this flag to ensure that nuisance calls are minimized. This flag is automatically set high after a successful call.

Parameter 3 Time Limit on Call Attempt

One second units. A call attempt is timed from execution of P97 until a valid answering identification code is received by the CR23X. If the code is not received before the time limit, the CR23X hangs up and waits for the next attempt or retry.

Parameter 4 Delay Between Fast Retries

One second units.

Parameter 5 Number of Fast Retries

Number of retries at the fast rate entered in parameter 4.

Parameter 6 Delay Between Slow Retries

One minute units. After fast retries have expired, P97 will continue to try at the slow rate entered here.

> **NOTE:** Actual delays between fast and slow retries include a random offset that ranges between 0 and 0.5 of the delay specified. The offset prevents calls from different stations occurring at the same time. The resolution of the offset is the execution interval of the program table wherein P97 resides. The offset is divided by the execution interval to determine how many times P97 must be executed before it retries. For this to work properly, P97 must be executed each pass through the program table. P97 should not be placed in conditional loops or subroutines.

Parameter 7 Failures Location

Input location where the number of failures is to be recorded

Parameter 8 Identification Code

Three digit identification code. The code will be sent every 4 seconds until the CR23X receives an identical response, or until the time limit on the call expires. No more than 4 seconds can elapse between each character of the reply.

Additional Programming Requirements

Radio, telephone, and generic modem applications require the use of one or more Instruction P68 immediately following P97.

P68 Hints:

Digits, such as telephone numbers and RF addresses, are entered directly, one character per line (0 - 9). All other characters must be entered as ASCII decimal equivalents. A list of ASCII decimal equivalents is shown in Appendix E.

X (ASCII equivalent = 88). If X is entered in the first parameter in the first (01:88), the second parameter represents a delay of 0 to 9 seconds (02:9). The delay is placed between when the call is answered and when the identification code is sent. This command is useful in compensating for delays or audio queues, as may occur in pager or other systems.

Space (32). Separate RF station addresses with a space.

F (70). Enter an F to indicate the end of an RF path.

T (84). Indicate a switch from RF to telephone with a space (32) followed by a T.

<CR> (13). A carriage return (<CR>) must be used to terminate the character sequence. Any unused P68 parameters are left at 0.

Table 12-8 lists special characters that are active only when a generic modem / baud rate option is used.

Table 12-7. Option Codes for P97 Initiate Telecommunications

Code	Device	Baud
00	RF95	300
01	RF95	1200
02	RF95	9600
10	**Direct CS I/O**	**300**
11	**Direct CS I/O**	**1200**
14	**Direct CS I/O**	**2400**
15	**Direct CS I/O**	**4800**
12	**Direct CS I/O**	**9600**
16	**Direct CS I/O**	**19.2 k**
17	**Direct CS I/O**	**38.4 k**
20	COM200, DC112	300
21	COM200, DC112	1200
24	COM200	2400
25	COM200	4800
22	COM200	9600
26	Telephone	19.2 k
27	Telephone	38.4 k
31	**VS1, COM300**	**Voice**
40	VS1, COM300	300
41	VS1, COM300	1200
44	COM300	2400
45	COM300	4800
42	COM300	9600
46	COM300	19.2 k
47	COM300	38.4 k
50	**RF (SDC)**	**300**
51	**RF (SDC)**	**1200**
52	**RF (SDC)**	**9600**
60	Direct Computer RS232	300
61	Direct Computer RS232	1200
64	Direct Computer RS232	2400
65	Direct Computer RS232	4800
62	Direct Computer RS232	9600
66	Direct Computer RS232	19.2 k
67	Direct Computer RS232	38.4 k
70	Generic Modem CS I/O	300
71	Generic Modem CS I/O	1200
74	Generic Modem CS I/O	2400
75	Generic Modem CS I/O	4800
72	Generic Modem CS I/O	1600
76	Generic Modem CS I/O	19.2 k
77	Generic Modem CS I/O	38.4 k
80	Generic Modem RS232	300
81	Generic Modem RS232	1200
84	Generic Modem RS232	2400
85	Generic Modem RS232	4800
82	Generic Modem RS232	9600
86	Generic Modem RS232	19.2 k
87	Generic Modem RS232	38.4 k

TABLE 12-8. Instruction P68 Parameters Specific to Generic Modem Applications

Parameter   Description

Tn   Transmit n bytes (n = ASCII digit, bytes = ASCII decimal equivalent)
En   Transmit n bytes, wait for echo (n = ASCII digit, bytes = ASCII decimal equivalent)
Rn   Receive n bytes (n = ASCII digit)
Dnn   Delay (nn = 1/10ths of seconds)
Wnn   Wait before retry (nn = 1/10ths of seconds)
            Retries = the last byte sent from the E command or,
            Retries = the last sequence sent with the T command
Lnn   Transmit the data value at Input Location nn
Inn   Transmit the data value at Input Location nn and wait for an echo of each character

---

> **CAUTION:** Instruction 97 with the voice option and the SDI-12 instructions may not be in same table. Instruction 97 must be in Table 1 and the SDI-12 instructions must be in Table 2.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Modem option and baud rate code. Left digit specifies the modem being used and the right, the baud rate.
02:	2	Interrupt disable flag
03:	4	Time limit on call, 1 sec. units
04:	4	Delay between fast retries, 1 sec. units
05:	2	No. of retries at fast rate
06:	4	Delay between slow retries, 1 min. units
07:	4	Input location to store no. of tries
08:	4	ID to send

*** 98 SEND CHARACTER ***

Instruction 98 is used with Instruction 63 to send a character or string of characters (up to 15) out a serial port to a device. The Computer RS232 port can be configured to stay on continuously in the $\boxed{*}$ $\boxed{D}$ mode as described in Section 1.8.7.

The device may be either addressed or pin-enabled (Section 6.2). Instruction 63 must immediately follow 98. The character or characters to send are entered in Instruction 63 as the decimal equivalents (99 is the maximum number allowed) of the 7 bit ASCII character (sent as 8 bits, no parity). For example, to send the ASCII character control R, 18 would be entered. Enter a null (0) to terminate the string. Appendix E contains a listing of the ASCII characters.

If the 9 pin connector is already active when Instruction 98 is executed, the output request is put in a queue (see Instruction 96).

This instruction can be used to send a control character to activate some listing device. The specified character(s) is sent at the time Instruction 98 and 63 are executed.

PARAM. NUMBER	DATA TYPE	DESCRIPTION	
01:	2	1y	Addressed Print Device
		4y	Pin-enabled Print Device (CS I/O)
		4y--	Pin-enabled Print Device (RS-232)
		y is baud rate code (Table 12-6)	

*** 111 RUN PROGRAM FROM FLASH ***

FUNCTION
This instruction is used to load a program stored in FLASH into RAM. The program in RAM is replaced by the Flash program specified in Parameter 1. If Parameter 1 is indexed, the CR23X will compile the program like the $\boxed{*}$ $\boxed{6}$ mode. Use this instruction with caution.

NOTE: The memory allocation ([*] [A]) must be the same between the program in RAM and the program that is loaded from Flash. If the memory allocations are not the same, the CR23X will reallocate memory and the data in Final Storage will be lost. Do not use the auto program allocation (0 for Parameter 5 in the [*] [A] mode), because data will be lost.

PARAM. NUMBER	DATE TYPE	DESCRIPTION
01:	2	Flash program number (name)

*** 114 SET TIME ***

Datalogger time can be set to the nearest second using P114. Time values are read from sequential input locations. Time is set each time P114 is executed.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Option 0 = Hr, Min, Sec 1 = Day*, Hour, Minutes, Seconds 2 = Year, Day*, Hour, Minutes, Seconds
02:	4	Beginning Input Location

Input Locations Used = 3 to 5, depending on option selected

*Day must be entered as day-of-year (1 to 366)

*** 120 TGT1 TELONICS GOES ***

FUNCTION
This instruction is used to transmit data from CR23X Final Storage via a GOES satellite to a central ground station. See the TGT1 manual for information on Instruction 120.

*** 121 ARGOS ***

FUNCTION
This instruction is used to transmit data from CR23X Final Storage via an ARGOS satellite. See the ARGOS Interface Notes for information on Instruction 121.

*** 122 INMARSAT-C ***

FUNCTION
This instruction is used to transmit data from CR23X Final Storage and receive data via the Galaxy INMARSAT - C satellite transceiver. See the INMARSAT-C interface notes for information on Instruction 122.

*** 123 AUTOMATIC PROGRAMMING OF A *** TGT1

FUNCTION
Instruction 123 performs Automatic Programming of TGT1 GOES Transmitter. See the TGT1 manual for information on Instruction 123.

*** 220 DISPLAY CHARACTERS ***

Displays specified information to the display while in the [*] [0] or Running Table mode. Each parameter represents the ASCII code of a character to display or the input location containing a value to display. Table 12-9 shows character to ASCII conversions. Executing this instruction without character 16 (see below) will leave the display on, increasing power usage by 4.5 mA.

NOTE: Use of P220 may disrupt telecommunications with the datalogger.

PARAM. NUMBER	DATE TYPE	DESCRIPTION
01:	4	ASCII code
02:	4	ASCII code
03:	4	ASCII code
04:	4	ASCII code
05:	4	ASCII code
06:	4	ASCII code
07:	4	ASCII code
08:	4	ASCII code

Program Memory bytes = 18

Input Locations Read = 0-8 depending on number of parameters using the XXX -- option.

8 Parameters of type 4.

Valid Character Codes

XXX-- Displays value stored in input location XXX

SECTION 12. PROGRAM CONTROL INSTRUCTIONS

Special Movement codes possible to use

8, 129	Backspace
10	Clears bottom line
13	Blank Display
16	Turn off display
128	Beginning of Top line
192	Beginning of Bottom line
128-151	Skip # Spaces (please note that not all "characters" are displayed on the screen). For example 131 skips 1

space, 132 skips 2 spaces, etc. It does not write over any characters written in the spaces it skips.

192-215 Move to the first Character on the Bottom Line & Skip # Spaces (note that not all "characters" are displayed on the screen). For example 193 skips 1 space, 194 skips 2 spaces, etc. It does not write over any characters written in the spaces it skips.

TABLE 12-9. P220 ASCII Conversion

1	T O	55	7	80	P	105	i
9	T O	56	8	81	Q	106	j
32	SPACE	57	9	82	R	107	k
33	!	58	:	83	S	108	l
34	"	59	;	84	T	109	m
35	#	60	<	85	U	110	n
36	$	61	=	86	V	111	o
37	%	62	>	87	W	112	p
38	&	63	?	88	X	113	q
39	'	64	@	89	Y	114	r
40	(	65	A	90	Z	115	s
41	)	66	B	91	[	116	t
42	*	67	C	92	¥	117	u
43	+	68	D	93	]	118	v
44	,	69	E	94	^	119	w
45	-	70	F	95	_	120	x
46	.	71	G	96	`	121	y
47	/	72	H	97	a	122	z
48	0	73	I	98	b	123	{
49	1	74	J	99	c	124	|
50	2	75	K	100	d	125	}
51	3	76	L	101	e	126	→
52	4	77	M	102	f	127	←
53	5	78	N	103	g		
54	6	79	O	104	h		

SECTION 13. CR23X MEASUREMENTS

> **NOTE:** Highlighted portions of this section have not been updated for the CR23X.

13.1 FAST AND SLOW MEASUREMENT SEQUENCE

The CR23X makes voltage measurements by integrating the input signal for a fixed time and then holding the integrated value for the analog to digital (A/D) conversion. The A/D conversion is made with a 15 bit successive approximation technique which resolves the signal voltage to approximately one part in 60,600 of the full scale range on a differential measurement (e.g., 1/60,600 x 10 V = 165 µV). The resolution of a single-ended measurement is one part in 30,300. Table 13-9 lists possible input voltage ranges.

Integrating the signal removes noise that could create an error if the signal were instantaneously sampled and held for the A/D conversion. Slow integration times provides a more noise-free reading than fast integration times. One of the most common sources of noise is 60 Hz from AC power lines. For all input ranges, except the ±5000 mV range, the 60 Hz integration time of 16.67 milliseconds is equal to one 60 Hz cycle. During the integration time, the AC noise will integrate to 0. The 50 Hz integration options lengthen the integration time to 20 ms for countries that have 50 Hz

power. This is the preferred method of AC rejection.

The CR23X also has options for faster AC line rejection. Two integrations are made spaced 1/2 cycle apart (Figure 13.1-1), which results in the AC noise integrating to 0. This is the method used by the ±5000 mV range for AC noise rejection. This method is also available on other ranges by preceding the range code with a 4 (60 Hz) or 5 (50 Hz), i.e., 41, 42..45.

There are several situations where the fast integration time of 250 µsec is preferred. The fast integration time minimizes time skew between measurements and increases the throughput rate. The current drain on the CR23X batteries is lower when fast integration time is used. The fast integration time should ALWAYS be used with the AC half bridge (Instruction 5) when measuring AC resistance or the output of an LVDT. An AC resistive sensor will polarize if a DC voltage is applied, causing erroneous readings and sensor decay. The induced voltage in an LVDT decays with time as current in the primary coil shifts from the inductor to the series resistance; a long integration time would result in most of the integration taking place after the signal had disappeared.

TABLE 13.1-1. Input Voltage Ranges and Codes

Range Code					Full Scale Range	Differential** Resolution
Fast 250 µs Integ.	60 Hz Reject.	50 Hz Reject.	Faster 60 Hz Reject.	Faster 50 Hz Reject.		
10	20	30	40	50	Autorange ***	
11	21	31	41	51	±10 mV	0.33 µV
12	22	32	42	52	±50 mV	1.67 µV
13	23	33	43	53	±200 mV	6.66 µV
14	24	34	44	54	±1000 mV	33.3 µV
15			45	55	±5000 mV	166. µV

Same as 2X 60Hz Rejection Range. Made available for compatibility with 21X Micrologger programs.
***Differential measurement, resolution for single-ended measurement is twice value shown.*
****See autoranging precaution in Table 3.5-1.*
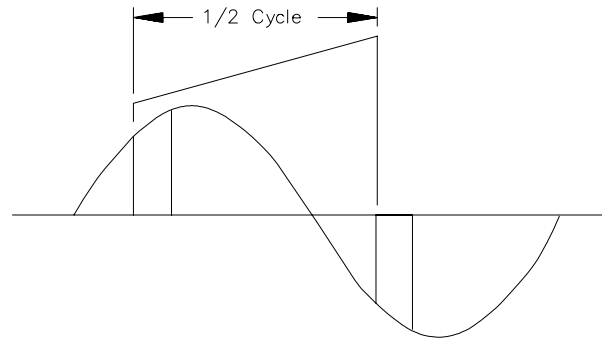
FIGURE 13.1-1. Fast 50 and 60 Hz Noise Rejection

Setup   Amplifier   Integration   A/D Conversion
         Settling

450 uS   250 uS fast   260 uS   Reset Integrator
               16.67 mS 60 Hz Reject
               20.00 mS 50 Hz Reject
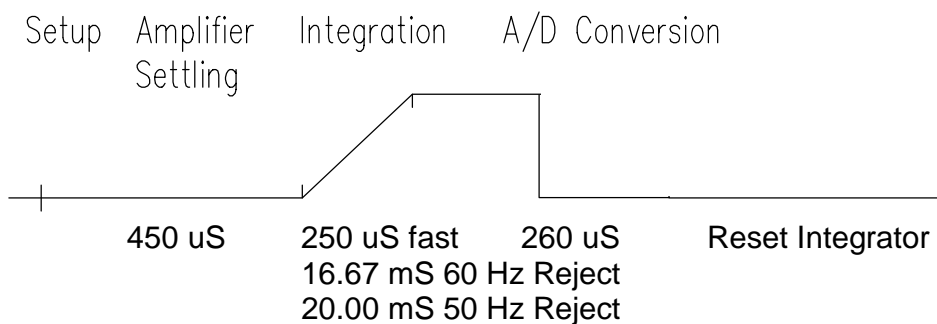
FIGURE 13.1-2. Timing of Single-Ended Measurement

13.1.1 OFFSET VOLTAGE MEASUREMENT

A single-ended measurement on the ±10 mV and ±50 mV ranges takes longer than on other input ranges because an offset measurement prior to every scan is performed. Offset measurements are performed in background on all other input ranges. Measurement time can be reduced on the ±10 and ±50 mV ranges by using the "--" option (e.g., 11--, 12--, 21--, 22--, 31--, 32--). The "--" option causes offset measurements for the ±10 mV and ±50 mV ranges to be performed in the background. The "--" option, however, is used at the expense of compensation for rapid changes in offset voltage. The "--" option is selected by pressing the $\boxed{C}$ key after entering the numeric value on the CR23X key pad.

13.2 SINGLE-ENDED AND DIFFERENTIAL VOLTAGE MEASUREMENTS

The timing and sequence of a single-ended measurement is shown in Figure 13.2-1. A single-ended measurement is made on a single input which is referenced to ground. A single integration is performed for each measurement.

A differential measurement measures the difference in voltage between two inputs. The measurement sequence on a differential measurement involves two integrations. First with the high input referenced to the low, then with the inputs reversed (Figure 13.2-2).

The CR23X computes the differential voltage by averaging the magnitude of the results from the two integrations and using the polarity from the first. An exception to this is the differential measurement in Instruction 8 which makes only one integration.

Because a single-ended measurement is referenced to CR23X ground, any difference in ground potential between the sensor and the CR23X will result in an error in the measurement. For example, if the measuring junction of a copper-constantan thermocouple, used to measure soil temperature, is not insulated and the potential of earth ground is 1 mV greater at the sensor than at the point where the CR23X is grounded, the measured voltage would be 1 mV greater than the thermocouple output or approximately 25°C high.

```
Setup  &  Integ.     A/D Conv. Switch Inpt.   Integ.    A/D Conv.
Settle                         & Settle
```

250 us fast   260 us   500 us   250 us fast   260 us
16.67 ms 60 Hz Reject   16.67 ms 60 Hz Reject
20.00 ms 50 Hz Reject   20.00 ms 50 Hz Reject
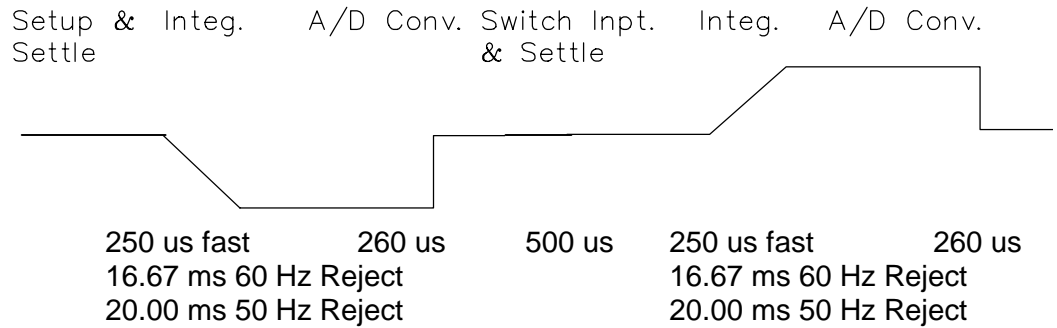
FIGURE 13.2-2. Differential Voltage Measurement Sequence

Another instance where a ground potential difference creates a problem is in a case such as described in Section 7.2, where external signal conditioning circuitry is powered from the same external source as the CR23X. Despite being tied to the same ground, differences in current drain and lead resistance result in different ground potential at the two instruments. For this reason a differential measurement should be made on an analog output from the external signal conditioner. Differential measurements MUST be used where the inputs are known to be different from ground, such as is the case with the output from a full bridge.

In order to make a differential measurement, the inputs must be within the CR23X common mode range of ±5.0 V. The common mode range is the voltage range, relative to CR23X ground, within which both inputs of a differential measurement must lie in order for the differential measurement to be made.

For example, if the high side of a differential input is at 4 V and the low side is at 3 V relative to CR23X ground, there is no problem; a measurement made on the ±5.0 V range would indicate a signal of 1 V. However, if the high input is at 5.8 V and the low input is at 4.8 V, the measurement cannot be made because the high input is outside of the common mode range. The CR23X will indicate the overrange with the maximum negative number (Section 3.5.)

Problems with exceeding common mode range may be encountered when the CR23X is used to read the output of external signal conditioning circuitry if a good ground connection does not exist between the external circuitry and the CR23X. When operating where AC power is available, it is not always safe to assume that a good ground connection exists through the AC wiring. If a CR23X is used to measure the output from a laboratory instrument (both plugged into AC power and referencing ground to outlet ground), it is best to run a ground wire between the CR23X and the external circuitry. Even with this ground connection, the ground potential of the two instruments may not be at exactly the same level, which is why a differential measurement is desired (Section 7.2).

If a differential measurement is used on a sensor that is not referenced to CR23X ground through a separate connection (e.g., a net radiometer), a jumper wire should be connected between the low side of the differential input and analog ground to hold the sensor in common mode range.

A differential measurement has better noise rejection than a single-ended measurement. Integrating the signal in both directions also reduces input offset voltage due to thermal effects in the amplifier section of the CR23X. Input offset voltage on a single-ended measurement is less than ±5 microvolts; the input offset voltage on a differential measurement is less than ±1 microvolt.

A single-ended measurement is quite satisfactory in cases where noise is not a problem and care is taken to avoid ground potential problems. Channels are available for twice as many single-ended measurements. A single-ended measurement takes about half the time of a differential measurement, which is valuable in cases where rapid sampling is a requirement.

NOTE: Sustained voltages in excess of +16 VDC applied to the analog inputs will damage the CR23X input circuitry.

13.3 THE EFFECT OF SENSOR LEAD LENGTH ON THE SIGNAL SETTLING TIME

Whenever an analog input is switched into the CR23X measurement circuitry prior to making a measurement, a finite amount of time is required for the signal to stabilize at its correct value. The rate at which the signal settles is determined by the input settling time constant which is a function of both the source resistance, and input capacitance (explained below). The CR23X allows a 450 μs settling time before initiating the measurement. In most applications this settling time is adequate, but the additional wire capacitance associated with long sensor leads can increase the settling time constant to the point that measurement errors may occur. There are three potential sources of error which must settle before the measurement is made:

1. The signal must rise to its correct value.

2. A transient (~30 mV on an open channel) due to charge injection of the multiplexed analog inputs, must settle.

3. A larger transient, usually about 40 mV/V, caused by the switched, precision excitation voltage used in resistive bridge measurements must settle.

The purpose of this section is to bring attention to potential measurement errors caused when the input settling time constant gets too large and to discuss procedures whereby the effects of lead length on the measurement can be estimated. In addition, physical values are given for three types of wire used in CSI sensors, and error estimates for given lead lengths are provided. Finally, techniques are discussed for minimizing input settling error when long leads are mandatory.
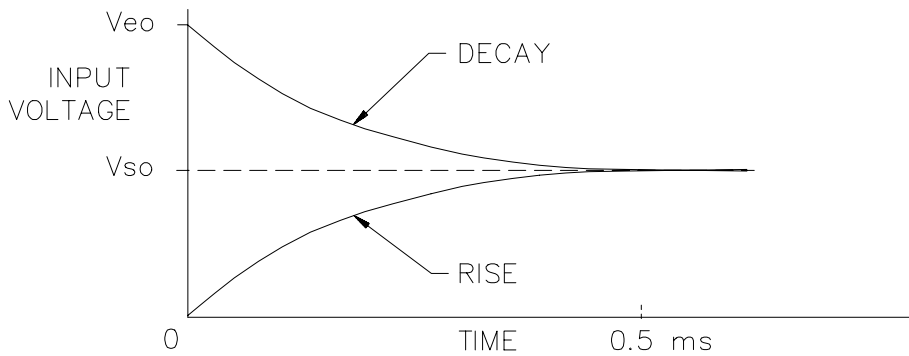
FIGURE 13.3-1. Input Voltage Rise and Transient Decay

13.3.1 THE INPUT SETTLING TIME CONSTANT

The rate at which an input voltage rises to its full value or that a transient decays to the correct input level are both determined by the input settling time constant. In both cases the waveform is an exponential. Figure 13.3-1 shows both a rising and decaying waveform settling to the signal level, Vso. The rising input voltage is described by Equation 13.3-1 and the decaying input voltage by Equation 13.3-2.

$$V_s = V_{so} (1-e^{-t/R_oC_T}), \text{ rise} \qquad [13.3\text{-}1]$$

$$V_s = V_{so} + (V_{eo}-V_{so})\, e^{-t/R_oC_T}, \text{ decay} \qquad [13.3\text{-}2]$$

where V_s is the input voltage, V_{so} the true signal voltage, V_{eo} the peak transient voltage, t is time in seconds, R_o the source resistance in ohms, and C_T is the total capacitance between the

signal lead and ground (or some other fixed reference value) in farads.

The settling time constant, τ in seconds, and the capacitance relationships are given in Equations 13.3-3 through 13.3-5,

$$\tau = R_oC_T \qquad [13.3\text{-}3]$$

$$C_T = C_f + C_wL \qquad [13.3\text{-}4]$$

$$C_f = 3.3 \text{ nfd} \qquad [13.3\text{-}5]$$

where C_f is the fixed CR23X input capacitance in farads, C_w is the wire capacitance in farads/foot, and L is the wire length in feet.

Equations 13.3-1 and 13.3-2 can be used to estimate the input settling error, V_e, directly.

For the rising case, $V_s = V_{so}-V_e$, whereas for the decaying transient, $V_s = V_{so}+V_e$. Substituting these relationships for V_s in Equations 13.3-1 and 13.3-2, respectively, yields expressions in V_e, the input settling error:

$$V_e = V_{so}\, e^{-t/R_oC_T}, \text{ rise} \qquad [13.3\text{-}6]$$

$$V_e = V_{e'o}\, e^{-t/R_oC_T}, \text{ decay} \qquad [13.3\text{-}7]$$

Where $V_{e'o} = V_{eo}-V_{so}$, the difference between the peak transient voltage and the true signal voltage.

> **NOTE:** Since the peak transient, V_{eo}, causes significant error only if it is several times larger than the signal, V_{so}, error calculations made in this section approximate $V_{e'o}$ by V_{eo}; i.e., $V_{eo} = V_{eo}-V_{so}$.

If the input settling time constant, τ, is known, a quick estimation of the settling error as a percentage of the maximum error (V_{so} for rising, $V_{e'o}$ for decaying) is obtained by knowing how many time constants (t/τ) are contained in the 450 μs CR23X input settling interval (t). The familiar exponential decay relationship is given in Table 13.3-1 for reference.

TABLE 13.3-1. Exponential Decay, Percent of Maximum Error vs. Time in Units of τ

Time Constants	% Max. Error	Time Constants	% Max. Error
0	100.0	5	0.7
1	36.8	7	0.1
3	5.0	10	0.004

Before proceeding with examples of the effect of long lead lengths on the measurement, a discussion on obtaining the source resistance, R_o, and lead capacitance, C_wL, is necessary.
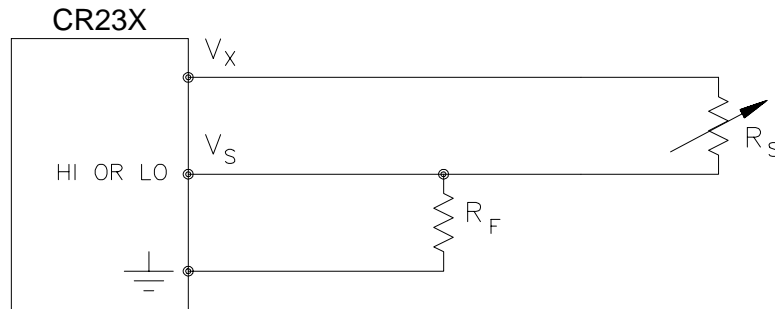
FIGURE 13.3-2. Typical Resistive Half Bridge
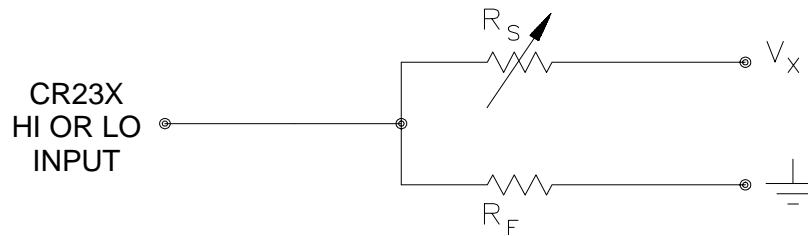
FIGURE 13.3-3. Source Resistance Model for Half Bridge Connected to the CR23X

DETERMINING SOURCE RESISTANCE

The source resistance used to estimate the settling time constant is the resistance the CR23X input "sees" looking out at the sensor. For our purposes the source resistance can be defined as the resistance from the CR23X input through all external paths back to the CR23X. Figure 13.3-2 shows a typical resistive sensor, (e.g., a thermistor) configured as a half bridge. Figure 13.3-3 shows Figure 13.3-2 re-drawn in terms of the resistive paths determining the source resistance Ro, is given by the parallel resistance of Rs and Rf, as shown in Equation 13.3-8.

$$R_0 = R_S R_f / (R_S + R_f) \qquad [13.3\text{-}8]$$

If R_f is much smaller, equal to or much greater than R_S, the source resistance can be approximated by Equations 13.3-9 through 13.3-11, respectively.

$$R_0 \sim R_f, R_f << R_S \qquad [13.3\text{-}9]$$

$$R_0 = R_f / 2, R_f = R_S \qquad [13.3\text{-}10]$$

$$R_0 \sim R_S, R_f >> R_S \qquad [13.3\text{-}11]$$

The source resistance for several Campbell Scientific sensors are given in column 3 of Table 13.3-5.

DETERMINING LEAD CAPACITANCE

Wire manufacturers typically provide two capacitance specifications: 1) the capacitance between the two leads with the shield floating, and 2) the capacitance between the two leads with the shield tied to one lead. Since the input lead and the shield are tied to ground (often through a bridge resistor, R_f) in single-ended measurements such as Figure 13.3-2, the second specification is used in determining lead capacitance. Figure 13.3-4 is a representation of this capacitance, C_w, usually specified as pfd/ft. C_w is actually the sum of capacitance between the two conductors and the capacitance between the top conductor and the shield. Capacitance for 3 lead wires used in Campbell Scientific sensors is shown in column 6 of Table 13.3-2.
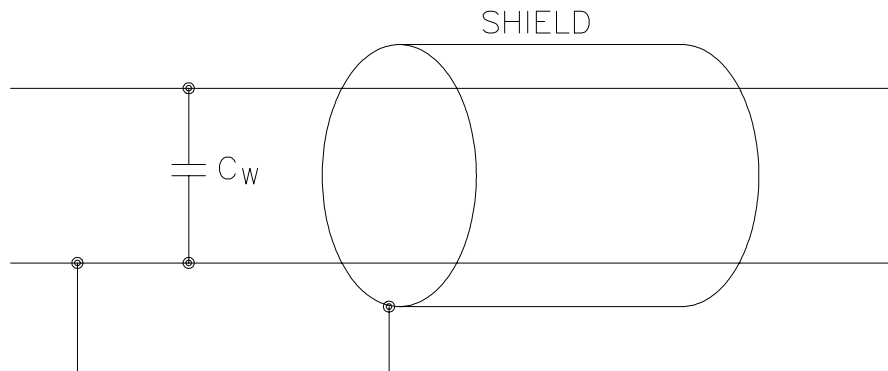
FIGURE 13.3-4. Wire Manufacturers Capacitance Specifications, C_w

TABLE 13.3-2. Properties of Lead Wires Used by Campbell Scientific

CSI Part #	Conductors	Insulation	AWG	RI (ohms/1000ft.)	C_w (pfd/ft.)
9721	3 shld. 3 pair	polypropylene	24	28	41
9661	1 shld. pair	polypropylene	22	18	42
7359	1 shld. 4 cond.	polyethylene	22	16	32

DIELECTRIC ABSORPTION

The dielectric absorption of insulation surrounding individual conductors can seriously affect the settling waveform by increasing the time required to settle as compared to a simple exponential. Dielectric absorption is difficult to quantify, but it can have a serious effect on low level measurements (i.e., 50 mV or less). The primary rule to follow in minimizing dielectric absorption is: Avoid PVC insulation around conductors. PVC cable jackets are permissible since the jackets don't contribute to the lead capacitance because the jacket is outside the shield. Campbell Scientific uses only polyethylene and polypropylene insulated conductors in CR23X sensors (see Table 13.3-2) since these materials have negligible dielectric absorption. Teflon insulation is also very good but quite expensive.

13.3.2 EFFECT OF LEAD LENGTH ON SIGNAL RISE TIME

In the 034A Wind Vane, a potentiometer sensor, the peak transient voltage is much less than the true signal voltage (see Table 13.3-5). This means the signal rise time is the major source of error and the time constant is the same as if C_w were between the signal lead and ground as represented below.
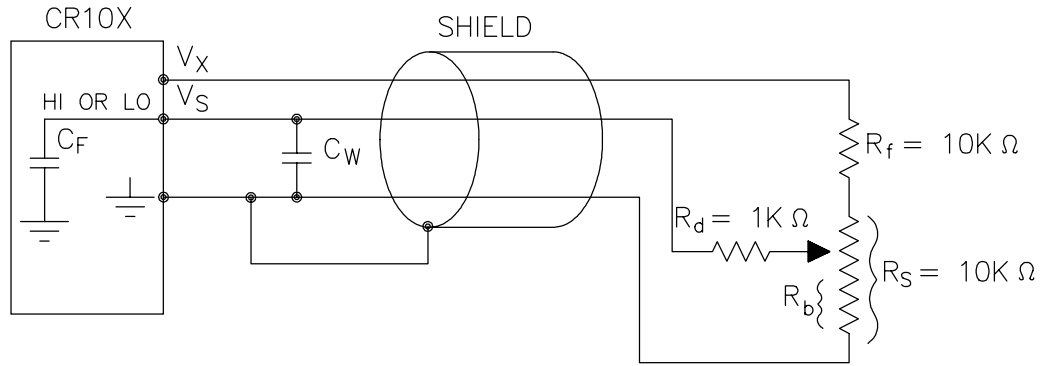
FIGURE 13.3-5. Model 034A Wind Direction Sensor
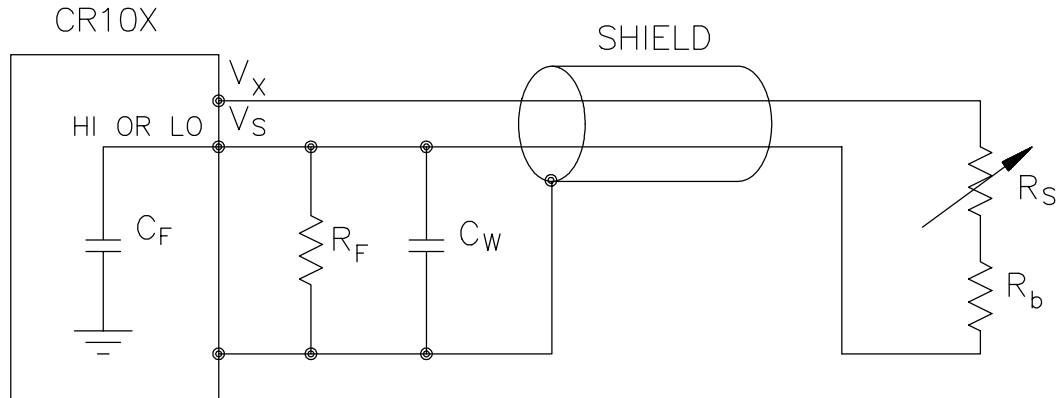
FIGURE 13.3-6. Resistive Half Bridge Connected to Single-Ended CR23X Input

R_o, the source resistance, is not constant because R_b varies from 0 to 10 kohms over the 0 to 360 degree wind direction range. The source resistance is given by:

$$R_o = R_d+(R_b(R_s-R_b+R_f)/(R_s+R_f)) =$$

$$R_d+(R_b(20k-R_b)/20k) \qquad [13.3\text{-}12]$$

Note that at 360 degrees, R_o is at a maximum of 6k (R_b=10k) and at 0 degrees, R_o is 1k (R_b=0). It follows that settling errors are less at lower direction values.

The value of R_b for any direction D (degrees) is given by:

$$R_b(kohms) = (10k)(D)/360 \qquad [13.3\text{-}13]$$

Equation 13.3-6 can be rewritten to yield the settling error of a rising signal directly in units of degrees.

$$\text{Error (degrees)} = De^{-t/(R_o(C_f+C_wL))} \qquad [13.3\text{-}14]$$

Equation 13.3-12, -13 and -14 can be combined to estimate the error directly in degrees at various directions and lead lengths (Table 13.3-3). Constants used in the calculations are given below:

C_f = 3.3nfd

C_w = 41 pfd/ft., p/n 9721 wire

t = 450µs

Example calculation at 360° with 1000 ft lead: (all variables converted to ohms, farads, seconds, or feet)

error =
$$360° * e^{(-450*10^{-6}\,s\,/\,(6*10^3\,\Omega*(3.3*10^{-9}\,fd+(41*10^{-12}\,fd\,ft^{-1}*1000ft))))}$$

error = 66°

TABLE 13.3-3. Settling Error, in Degrees, for 024A Wind Direction Sensor vs. Lead Length

Wind Direction	Error	
	L=1000 ft.	L=500 ft.
360°	66°	15°
270°	45°	9°
180°	21°	3°
90°	4°	0°

The values in Table 13.3-3 show that significant error occurs at large direction values for leads in excess of 500 feet. Instruction 4, Excite, Delay, and Measure, should be used to eliminate errors in these types of situations. Using a 10 ms delay, settling errors are eliminated up to lengths that exceed the drive capability of the excitation channel (~ 2000 ft.).

13.3.3 TRANSIENTS INDUCED BY SWITCHED EXCITATION

Figure 13.3-6 shows a typical half bridge resistive sensor, such as Campbell Scientific's Model 107 Temperature Probe, connected to the CR23X. The lead wire is a single-shielded pair, used for conducting the excitation (V_x) and signal (V_s) voltages. When V_x is switched on, a transient is capacitively induced in V_s, the signal voltage. If the peak transient level, V_{eo}, is less than the true signal, V_{so}, the transient has no effect on the measurement. If V_{eo} is greater than V_{so}, it must settle to the correct signal voltage to avoid errors.

TABLE 13.3-4. Measured Peak Excitation Transients for 1000 Foot Lengths of Lead Wires Used by Campbell Scientific

Vx(mV)	------------------------V_{eo}(mV) -----------------------					
	R_f=1 kohm			R_f=10 kohm		
	# 9661	# 7359	# 9721	# 9661	# 7359	# 9721
2000	50	100	60	100	140	80
1000	25	65	40	60	90	40

NOTE: Excitation transients are eliminated if excitation leads are contained in a shield independent from the signal leads.

The size of the peak transient is linearly related to the excitation voltage and increases as the bridge resistor, R_f, increases. Table 13.3-4 shows measured levels of V_{eo} for 1000 foot lengths of three Belden wires used in Campbell Scientific sensors. Values are given for R_f equal to 1 kohm and 10 kohm. Table 13.3-4 is meant only to provide estimates of the size of excitation transients encountered; the exact level will depend upon the specific sensor configuration.

Equation 13.3-7 can be solved for the maximum lead length, L, permitted to maintain a specified error limit. Combining Equations 13.3-7 and 13.3-4 and solving for L gives:

$$L = -(R_o C_f + (t/\ln(V_e/V_{eo})))/R_o C_w \qquad [13.3\text{-}15]$$

where V_e is the measurement error limit.

EXAMPLE LEAD LENGTH CALCULATION FOR 107 TEMPERATURE SENSOR

Assume a limit of 0.05°C over a 0°C to +40°C range is established for the transient settling error. This limit is a reasonable choice since it approximates the linearization error over that range. The output signal from the thermistor bridge varies nonlinearly with temperature ranging from about 100 µV/°C at 0°C to 50 µV/°C at 40°C. Taking the most conservative figure yields an error limit of V_e = 2.5 µV. The other values needed to calculate the maximum lead length are summarized in Table 13.3-5 and listed below:

1) V_{eo} ~ 50 mV, peak transient at 2 V excitation

2) V_e ~ 2.5 µV, allowable measurement error

3) t = 450 µs, CR23X input settling time

4) R_o = 1 kohm, 107 probe source resistance

5) C_f = 3.3 nfd, CR23X input capacitance

6) C_w ~ 42 pfd/ft., lead wire capacitance

Solving Equation 13.3-15 gives a maximum lead length of:

L ~ 1003 ft., error ~ 0.05°C

Setting the allowable error at 0.1°C or approximately 5 µV, the maximum lead length increases to:

L ~ 1085 ft., error ~ 0.1°C

13.3.4 SUMMARY OF SETTLING ERRORS FOR CAMPBELL SCIENTIFIC RESISTIVE SENSORS

Table 13.3-5 summarizes the data required to estimate the effect of lead length on settling errors for Campbell Scientific's resistive sensors. Comparing the transient level, V_{eo}, to the input range, one suspects that transient errors are the most likely limitation for the 107 sensor. The sensors in the WVU-7 are the same as in the Model 107 (the lead wire is different), but the signal leads for the WVU-7 wet- and dry-bulbs are not subject to excitation transients because they are shielded independently from the excitation.

TABLE 13.3-5. Summary of Input Settling Data For Campbell Scientific Resistive Sensors

Sensor Model #	Belden Wire #	Ro (kohms)	Cw (pfd/ft.)	τ^* (us)	Input Range(mV)	V_x(mV)	V_{eo}(mV)**
107	9661	1	42	45	10	2500	50
227	9661	0.1-1	42	5-45	1000	500	0
237	9661	1	42	45	50	5000	65
034A	9721	1-6	41	1-222	5000	5000	0-90

*   Estimated time constants are for 1000 foot lead lengths and include 3.3nfd CR23X input capacitance.

**   Measured peak transients for 1000 foot lead lengths at corresponding excitation, V_x.

TABLE 13.3-6. Maximum Lead Length vs. Error for Campbell Scientific Resistive Sensors

Sensor Model #	Error	Range	$V_e(\mu V)$	Maximum Length(ft.)
107	0.05°C	0°C to 40°C	5	1000[1]
034A	3°	@ 360°	2083	380[2]
227	-	-	-	2000[3]
237	10 kohm	20k to 300k	1000	2000[3]

[1] based on transient settling
[2] based on signal rise time
[3] limit of excitation drive

The comparatively small transient yet large source resistance of the 034A sensor indicates that signal rise time may be the most important limitation. The analysis in Section 13.3.2 confirms this.

The Model 227 Soil Moisture Block has a relatively short time constant and essentially no transient. Lead lengths in excess of 2000 feet produce less than a 0.1 bar (0-10 bar range) input settling error. With this sensor, the drive capability of the excitation channel limits the lead length. If the capacitive load 0.1 µfd and the resistive load is negligible, V_x will oscillate about its control point. If the capacitive load is 0.1 or less, V_x will settle to within 0.1% of its correct value 150 µs. A lead length of 2000 feet is permitted for the Model 227 before approaching the drive limitation.

Table 13.3-6 summarizes maximum lead lengths for corresponding error limits in six Campbell Scientific sensors. Since the first three sensors are nonlinear, the voltage error, V_e, is the most conservative value corresponding to the error over the range shown.

MINIMIZING SETTLING ERRORS IN NON-CAMPBELL SCIENTIFIC SENSORS

When long lead lengths are mandatory in sensors configured by the user, the following general practices can be used to minimize or measure settling errors:

1. When measurement speed is not a prime consideration, Instruction 4, Excite, Delay, and Measure, can be used to insure ample settling time for half bridge, single-ended sensors.

2. An additional low value bridge resistor can be added to decrease the source resistance, R_o. For example, assume a YSI nonlinear thermistor such as the model 44032 is used

with a 30 kohm bridge resistor, R'_f. A typical configuration is shown in Figure 13.3-7A. The disadvantage with this configuration is the high source resistance shown in column 3 of Table 13.3-7. Adding another 1 K resistor, R_f, as shown in Figure 13.3-7B, lowers the source resistance of the CR23X input. This offers no improvement over configuration A because R'_f still combines with the lead capacitance to slow the signal response at point P. The source resistance at point P (column 5) is essentially the same as the input source resistance of configuration A. Moving $R_{f'}$ out to the thermistor as shown in Figure 13.3-7C optimizes the signal settling time because it becomes a function of R_f and C_w only.

Columns 4 and 7 list the signal voltages as a function of temperature using a 5000 mV excitation for configurations A and C, respectively. Although configuration A has a higher output signal (5000 mV input range), it does not yield any higher resolution than configuration C which uses the ±1000 mV input range.

> **NOTE:** Since R_f' attenuates the signal in configuration B and C, one might consider eliminating it altogether. However, its inclusion "flattens" the non-linearity of the thermistor, allowing more accurate curve fitting over a broader temperature range.

3. Where possible, run excitation leads and signal leads in separate shields to minimize transients.

4. Avoid PVC-insulated conductors to minimize the effect of dielectric absorption on input settling time.

5. Use the CR23X to measure the input settling error associated with a given configuration. For example, assume long leads are required but the lead capacitance, Cw, is unknown. Configure Rf on a length of cable similar to the measurement. Leave the sensor end open as shown in Figure 13.3-8 and measure the result using the same instruction parameters to be used with the sensor. The measured deviation from 0V is the input settling error.

6. Most Campbell Scientific sensors are configured with a small bridge resistor, R_f, (typically 1 kohm) to minimize the source resistance. If the lead length of a Campbell Scientific sensor is extended by connecting to the pigtails directly, the effect of the lead resistance, R_l, on the signal must be considered. Figure 13.3-9 shows a Campbell Scientific Model 107 sensor with 500 feet of extension lead connected directly to the pigtails. Normally the signal voltage is proportional to $R_f/(R_s+R_b+R_f)$, but when the pigtails are extended, the signal is proportional to $(R_f+R_l)/(R_s+R_b+R_f+R_l)$. R_l is much smaller than the other terms in the denominator and can be discarded. The effect on the signal can be analyzed by taking the ratio of the signal with extended leads, V_{sl} to the normal signal, V_s:

$$V_{sl}/V_s = (R_f+R_l)/R_f$$

Plugging in values of R_f=1k and R_l=.012k (500' at 23 ohms/1000', Table 13.3-2) gives an approximate 1% error in the signal with extended leads. Converting the error to °C gives approximately a 0.33=°C error at 0°C, 0.53°C error at 20°C, and a 0.66°C error at 40°C. The error can be avoided by maintaining the pigtails on the CR23X end of the extended leads because R_l does not add to the bridge completion resistor, R_f, and its influence on the thermistor resistance is negligible.

TABLE 13.3-7. Source Resistances and Signal Levels for YSI #44032 Thermistor Configurations Shown in Figure 13.3-7 (2V Excitation)

| T | R_s (kohms) | --------A-------- | | -----B----- | -------C------- | |
		R_o (kohms)	V_s(mV)	R_o@P (kohms)	R_o (kohms)	V_s(mV)
-40	884.6	29.0	164	30.0	1	5.5
-20	271.2	27	498	27.8	1	16.5
0	94.98	22.8	1200	23.4	1	39.5
+25	30.00	15.0	2500	15.2	1	82.0
+40	16.15	10.5	3250	10.6	1	106.0
+60	7.60	6.1	3989	6.1	1	129.5
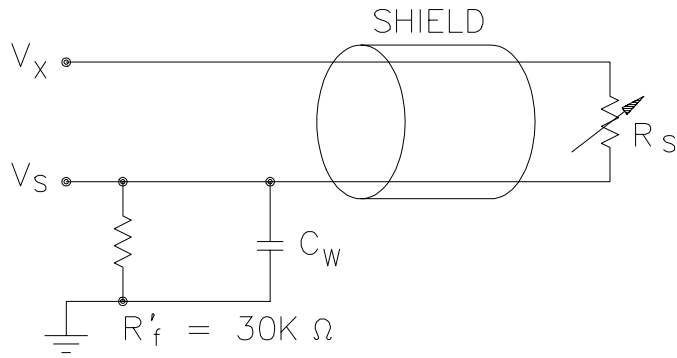
$$A) \quad R_O = R_S R'_f / (R_S + R'_f), \qquad V_S = V_X R'_f / (R_S + R'_f)$$
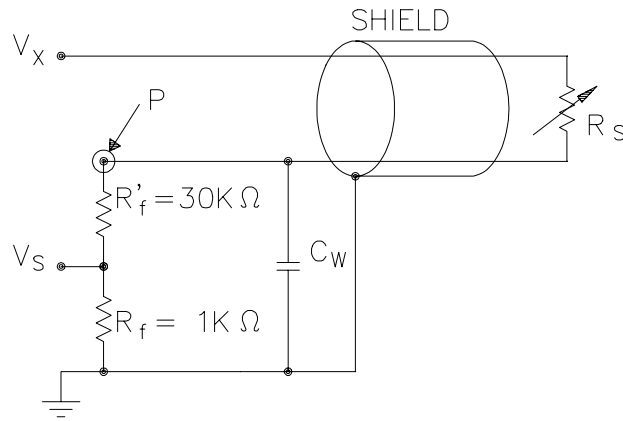
$$B) \quad R_O @P = R_S (R'_f + R_f) / (R_S + R'_f + R_f)$$
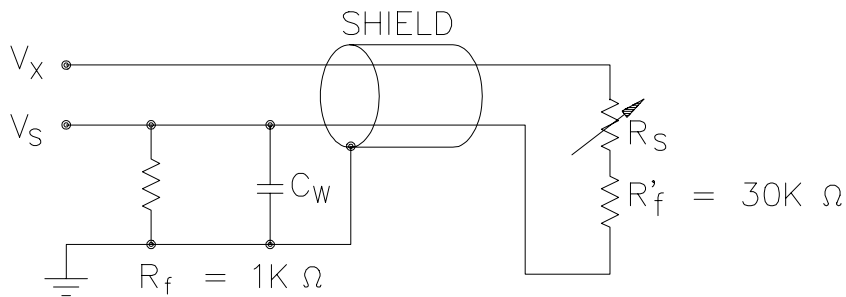
$$C) \quad R_O = R_f, \qquad V_S = V_X R_f / (R_S + R'_f + R_f)$$

FIGURE 13.3-7. Half Bridge Configuration for YSI #44032 Thermistor Connected to CR23X
Showing: A) large source resistance,
B) large source resistance at point P, and
C) configuration optimized for input settling
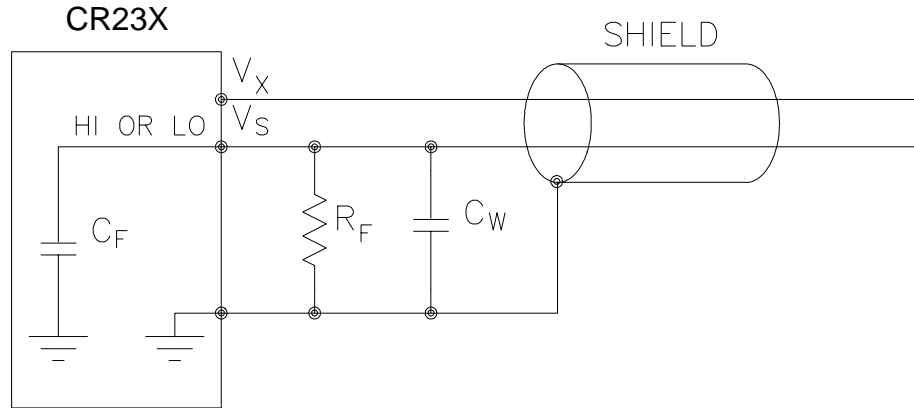
FIGURE 13.3-8. Measuring Input Settling Error with the CR23X
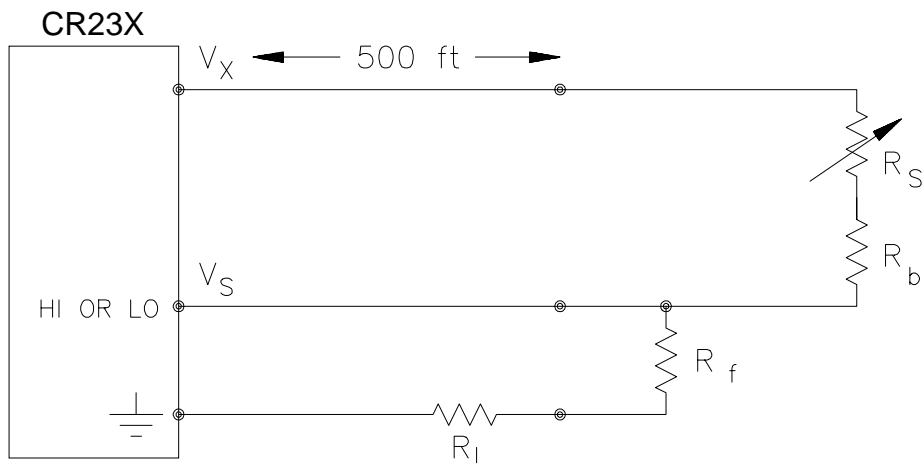
FIGURE 13.3-9. Incorrect Lead Wire Extension on Model 107 Temperature Sensor
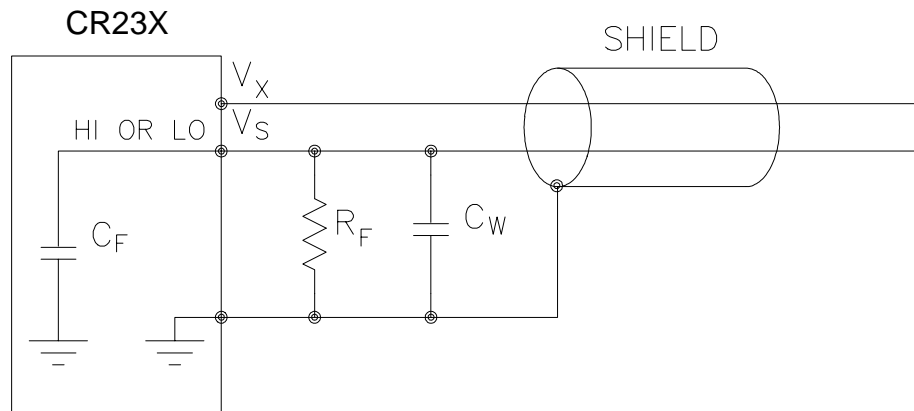
FIGURE 13.3-8. Measuring Input Settling Error with the CR23X

CR23X
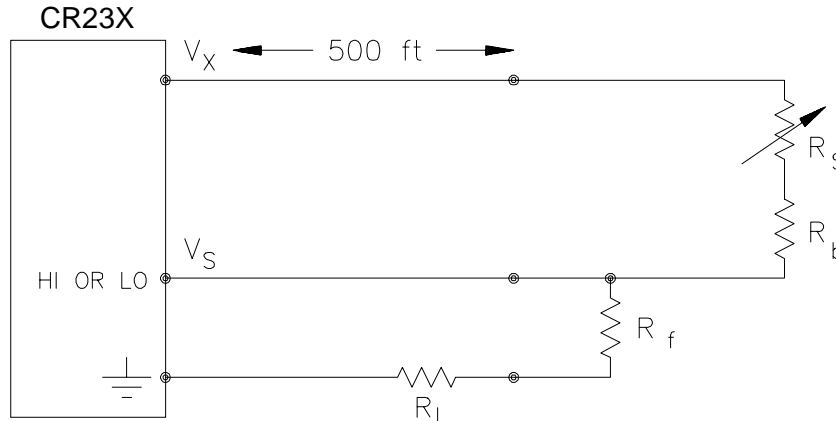
FIGURE 13.3-9. Incorrect Lead Wire Extension on Model 107 Temperature Sensor

13.4 THERMOCOUPLE MEASUREMENTS

A thermocouple consists of two wires, each of a different metal or alloy, which are joined together at each end. If the two junctions are at different temperatures, a voltage proportional to the difference in temperatures is induced in the wires. When a thermocouple is used for temperature measurement, the wires are soldered or welded together at the measuring junction. The second junction, which becomes the reference junction, is formed where the other ends of the wires are connected to the measuring device. (With the connectors at the same temperature, the chemical dissimilarity between the thermocouple wire and the connector does not induce any voltage.) When the temperature of the reference junction is known, the temperature of the measuring junction can be determined by measuring the thermocouple voltage and adding the corresponding temperature difference to the reference temperature.

The CR23X determines thermocouple temperatures using the following sequence. First, the temperature of the reference junction is measured. If the reference junction is the CR23X panel, the temperature is measured with the built-in thermistor (Instruction 17). The reference junction temperature in °C is stored in an input location which is accessed by the thermocouple measurement instruction (Instruction 13 or 14). The CR23X calculates the voltage that a thermocouple of the type specified would output at the reference junction temperature if its reference junction were at 0°C, and adds this voltage to the measured thermocouple voltage. The temperature of the measuring junction is then calculated from a polynomial approximation of the National Institute of Standards and Technology (NIST Monograph 175) TC calibrations.

13.4.1 ERROR ANALYSIS

The error in the measurement of a thermocouple temperature is the sum of the errors in the reference junction temperature, the thermocouple output (deviation from standards published in NIST Monograph 175), the thermocouple voltage measurement, and the polynomial error (difference between NIST standard and CR23X polynomial approximations). The discussion of errors which follows is limited to these errors in calibration and measurement and does not include errors in installation or matching the sensor to the environment being measured.

REFERENCE JUNCTION TEMPERATURE WITH CR23X PANEL

The CR23X panel temperature is measured with a BetaTherm 100K6A1 thermistor mounted under the analog input terminal strip. The accuracy of this measurement is a combination of the thermistor's interchangeability specification and the precision of the bridge resistors. In a "worst case" example, all errors add to yield a ±0.3°C error in the range of -10°C to +60°C. Figure 13.4-1 shows the thermistor's interchangeabiity.

Any difference in temperature between the thermistor and the actual reference junction (the channel to which the thermocouple is attached) will become an error in the measurement. The terminal strip cover is designed to reduce the gradient across the terminal strip. In field applications, with the terminal cover installed and the CR23X in one of Campbell Scientific's enclosures, this error will generally be less than 0.3°C.

100K6A1 Interchangeability Error (deg C)
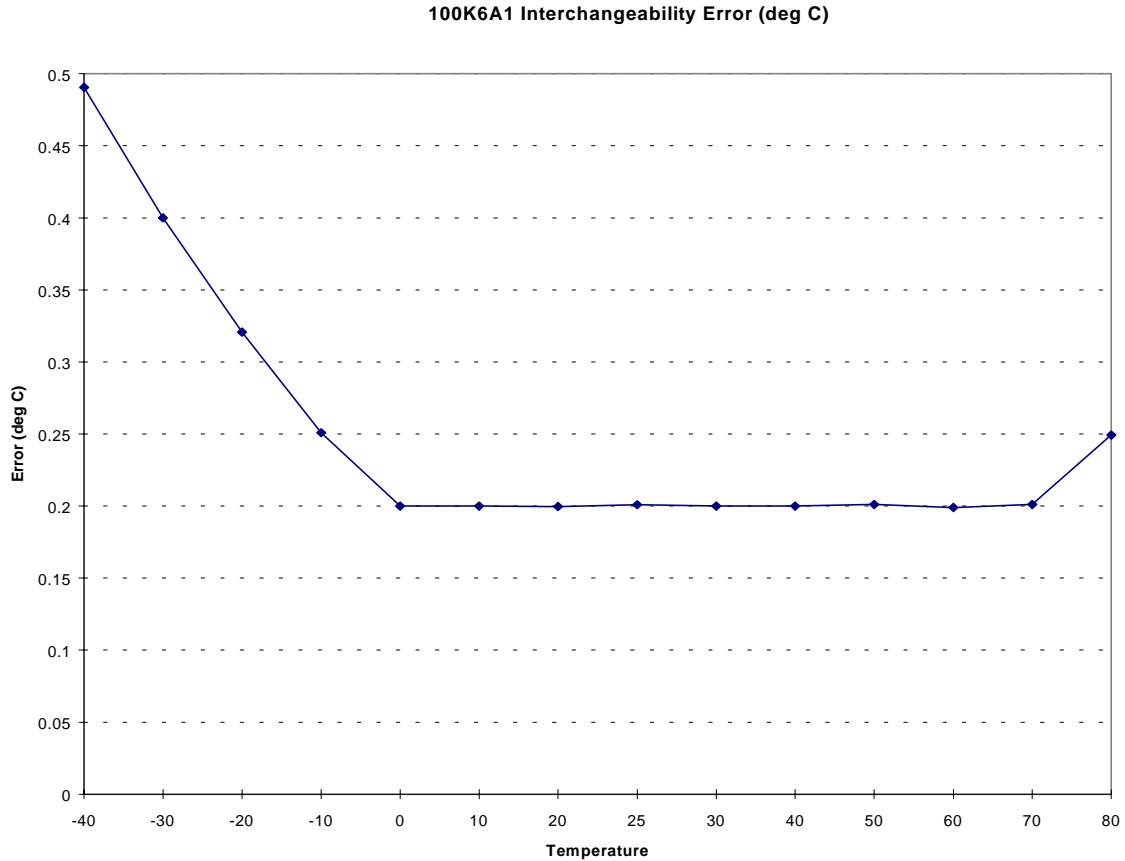
FIGURE 13.4-1. BetaTherm 100K6A1 Interchangeability Error

When the CR23X is subjected to large temperature gradients or rapid temperature changes, The reference temperature error can be much greater. Figure 13.4-2 shows the errors on three different channels while the CR23X was in a chamber that had the temperature changed from -40 to +60°C in about 12 minutes. The initial error while the temperature is stable at -40°C (-10 to 0 minutes) is largely thermistor interchangeablity error. During the temperature change, the differences in temperature between the input terminals and the reference thermistor dominate the error. Wire routing, the battery base, and the source of the temperature gradient can affect which channels have the greatest error. When making thermocouple measurements, ALWAYS USE THE TERMINAL COVER; the above test without the cover results in errors more than three times worse than shown in Figure 13.4-2.
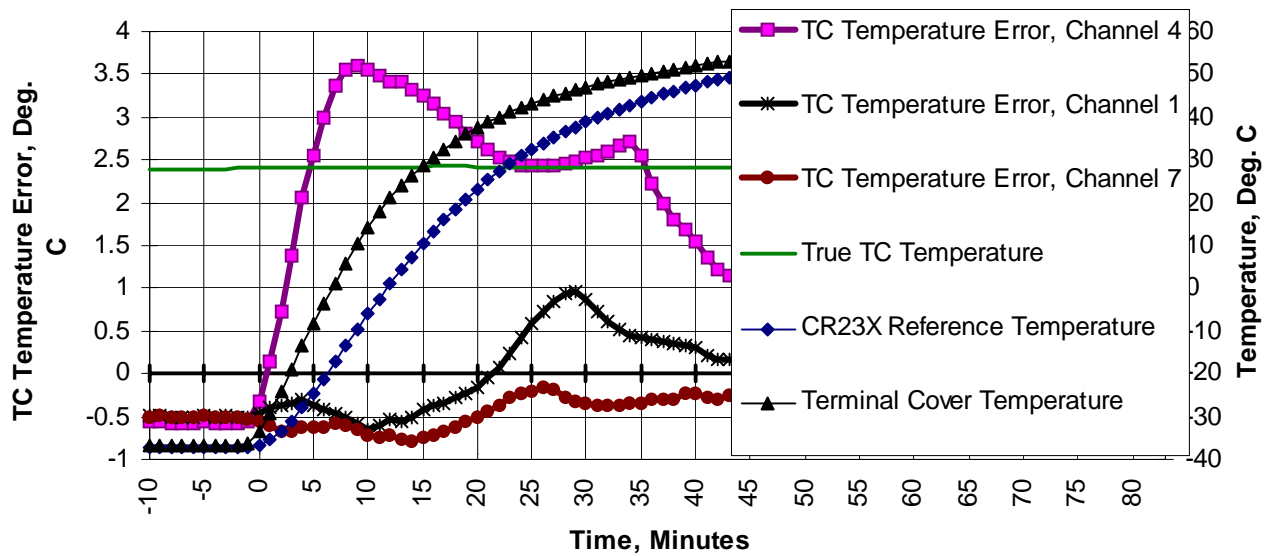
FIGURE 13.4-2. Thermocouple Temperature Error During Rapid Temperature Change Base w/o Battery, Terminal Cover in Place

TABLE 13.4-1. Limits of Error for Thermocouple Wire (Reference Junction at 0°C)

Thermocouple Type	Temperature Range °C	Limits of Error (Whichever is greater)	
		Standard	Special
T	-200 to 0	± 1.0°C or 1.5%	
	0 to 350	± 1.0°C or 0.75%	± 0.5°C or 0.4%
J	0 to 750	± 2.2°C or 0.75%	± 1.1°C or 0.4%
E	-200 to 0	± 1.7°C or 1.0%	
	0 to 900	± 1.7°C or 0.5%	± 1.0°C or 0.4%
K	-200 to 0	± 2.2°C or 2.0%	
	0 to 1250	± 2.2°C or 0.75%	± 1.1°C or 0.4%

THERMOCOUPLE LIMITS OF ERROR

The standard reference which lists thermocouple output voltage as a function of temperature (reference junction at 0°C) is the National Institute of Standards and Technology Monograph 175 (1974). The American National Standards Institute has established limits of error on thermocouple wire which is accepted as an industry standard (ANSI MC 96.1, 1975). Table 13.4-1 gives the ANSI limits of error for standard and special grade thermocouple wire of the types accommodated by the CR23X.

When both junctions of a thermocouple are at the same temperature, there is no voltage produced (law of intermediate metals). A consequence of this is that a thermocouple cannot have an offset error; any deviation from a standard (assuming the wires are each homogeneous and no secondary junctions exist) is due to a deviation in slope. In light of this, the fixed temperature limits of error (e.g., ±1.0°C for type T as opposed to the slope error of 0.75% of the temperature) in the table above are probably greater than one would experience when considering temperatures in the environmental range. In other words, the reference junction, at 0°C, is relatively close to the temperature being measured, so the absolute error (the product of the temperature difference and the slope error) should be closer to the percentage error than the fixed error. Likewise, because thermocouple calibration error is a slope error, accuracy can be increased when the reference junction temperature is close to the measurement temperature. For the same reason, differential temperature measurements, over a small temperature gradient, can be extremely accurate.

In order to quantitatively evaluate thermocouple error when the reference junction is not fixed at 0°C, one needs limits of error for the Seebeck coefficient (slope of thermocouple voltage vs. temperature curve) for the various thermocouples. Lacking this information, a reasonable approach is to apply the percentage errors, with perhaps 0.25% added on, to the difference in temperature being measured by the thermocouple.

ACCURACY OF THE THERMOCOUPLE VOLTAGE MEASUREMENT

The accuracy of a CR23X voltage measurement is specified as ±0.05% (±0.025% 0 to 40°C) of the full scale range being used to make the measurement. The actual accuracy may be better than this as it involves a slope error (the error is proportional to the measurement being made though limited by the resolution). The error in the temperature due to inaccuracy in the measurement of the thermocouple voltage is worst at temperature extremes, where a relatively large scale is necessary to read the thermocouple output. For example, assume type K (chromel-alumel) thermocouples are used to measure temperatures at 1000°C. The TC output is on the order of 40 mV, requiring the ±50 mV input range. The accuracy specification of 0.025% FSR is 25 µV which is a temperature error of about 0.6°C (±(.00025)(100 mV)=25 µV). In the environmental temperature range with voltage measured on an appropriate scale, error in temperature due to the voltage measurements is a few hundredths of a degree.

THERMOCOUPLE POLYNOMIALS: Voltage to Temperature

NIST Monograph 175 gives high order polynomials for computing the output voltage of a given thermocouple type over a broad range of temperatures. In order to speed processing and accommodate the CR23X's math and storage capabilities, 4 separate 6th order polynomials are used to convert from volts to temperature over the range covered by each thermocouple type. Table 13.4-2 gives error limits for these thermocouple polynomials.

TABLE 13.4-2. Limits of Error on Datalogger
Thermocouple Output Linearization
(Relative to ITS-90 Standard in NIST
Monograph 175)

TC Type	Range °C			Limits of Error °C
T	-200	to	87	± 0.022
	87	to	316	± 0.042
	316	to	400	± 0.060
E	-240	to	-130	± 0.40
	-130	to	479	± 0.05
	479	to	1000	± 0.21
K	-56	to	413	± 0.05
	413	to	1372	± 0.40
J	-150	to	92	± 0.02
	92	to	412	± 0.04
B	50	to	1007	± 0.20
	1007	to	1700	± 0.50
R	0	to	579	± 0.10
	579	to	1450	± 0.40
S	0	to	972	± 0.17
	972	to	1450	± 0.40
N	-200	to	1000	± 0.0075
	1000	to	1240	± 0.01

REFERENCE JUNCTION COMPENSATION: Temperature to Voltage

The polynomials used for reference junction compensation (converting reference temperature to equivalent TC output voltage) do not cover the entire thermocouple range. Substantial errors will result if the reference junction temperature is outside of the polynomial range. The ranges covered by these polynomials include the CR23X environmental operating range, so there is no problem when the CR23X is used as the reference junction. External reference junction boxes, however, must also be within these temperature ranges. Temperature difference measurements made outside of the reference temperature range should be made by obtaining the actual temperatures referenced to a junction within the reference temperature range and subtracting. Table 13.4-3 gives the reference

temperature ranges covered and the limits of error in the polynomials within these ranges.

Two sources of error arise when the reference temperature is out of range. The most significant error is in the calculated compensation voltage; however, error is also created in the temperature difference calculated from the thermocouple output. For example, suppose the reference temperature for a measurement on a type T thermocouple is 300°C. The compensation voltage calculated by the CR23X corresponds to a temperature of 272.6°C, a -27.4°C error. The type T thermocouple with the measuring junction at 290°C and reference at 300°C would output -578.7 µV; using the reference temperature of 272.6°C, the CR23X calculates a temperature difference of -10.2°C, a -0.2°C error. The temperature calculated by the CR23X would be 262.4°C, 27.6°C low.

TABLE 13.4-3. Reference Temperature
Compensation Range and Linearization
Error (Relative to ITS-90 Standard in NIST
Monograph 175)

TC Type	Range °C			Limits of Error °C
T	-100	to	100	± 0.025
E	-150	to	206	± 0.040
K	-50	to	100	± 0.030
J	-150	to	296	± 0.045
B	25	to	80	± 0.100
R	-50	to	100	± 0.020
S	-50	to	100	± 0.080
N	-100	to	100	± 0.001

ERROR SUMMARY

The magnitude of the errors described in the previous sections illustrate that the greatest sources of error in a thermocouple temperature measurement are likely to be due to the limits of error on the thermocouple wire and in the reference temperature determined with the built-in thermistor. Errors in the thermocouple and reference temperature polynomials are extremely small, and error in the voltage measurement is negligible.

To illustrate the relative magnitude of these errors in the environmental range, we will take a worst case situation where all errors are

maximum and additive. A temperature of 45°C is measured with a type T (copper-constantan) thermocouple, using the ±5 mV range. The nominal accuracy on this range is 2.5 µV (0.05% of 5 mV), which at 45°C changes the temperature by 0.06°C. The RTD is 25°C but is indicating 25.3°C, and the terminal that the thermocouple is connected to is 0.3°C cooler than the RTD.

TABLE 13.4-4. Example of Errors in Thermocouple Temperature

Source	Error °C	% of Total Error	
		$1^{o}C$ Error	1% Slope Error
Reference Temp.	0.6	36.1	69.6
TC Output			
ANSI	1.0	60.1	
0.01 x 20°C	0.2		23.2
Voltage Measurement	0.06	3.6	7.0
Reference Linearization	0.001	0.1	0.1
Output Linearization	0.001	0.1	0.1
Total Error			
With ANSI error	1.662	100	
Assuming 1% slope error	0.862		100

13.4.2 USE OF EXTERNAL REFERENCE JUNCTION OR JUNCTION BOX

An external junction box is often used to facilitate connections and to reduce the expense of thermocouple wire when the temperature measurements are to be made at a distance from the CR23X. In most situations, it is preferable to make the box the reference junction, in which case its temperature is measured and used as the reference for the thermocouples; copper wires are run from the box to the CR23X (Section 7.4). Alternatively, the junction box can be used to couple extension grade thermocouple wire to the thermocouples being used for measurement, and the CR23X panel used as the reference junction. Extension grade thermocouple wire has a smaller temperature range than standard thermocouple wire, but meets the same limits of error within that range. The only situation where it would be necessary to use extension grade wire instead of a external measuring junction is where the junction box temperature is outside the range of reference junction compensation provided by the CR23X. This is only a factor when using type K thermocouples, where the upper limit of the reference compensation linearization is 100°C and the upper limit of the extension grade wire is 200°C. With the other types of thermocouples, the reference compensation range equals or is greater than the extension wire range. In any case, errors can arise if temperature gradients exist within the junction box.

Figure 13.4-3 illustrates a typical junction box. Terminal strips will be a different metal than the thermocouple wire. Thus, if a temperature gradient exists between A and A' or B and B', the junction box will act as another thermocouple in series, creating an error in the voltage measured by the CR23X. This thermoelectric offset voltage is a factor whether or not the junction box is used for the reference. It can be minimized by making the thermal conduction between the two points large and the distance small. The best solution in the case where extension grade wire is being connected to thermocouple wire would be to use connectors which clamped the two wires in contact with each other.
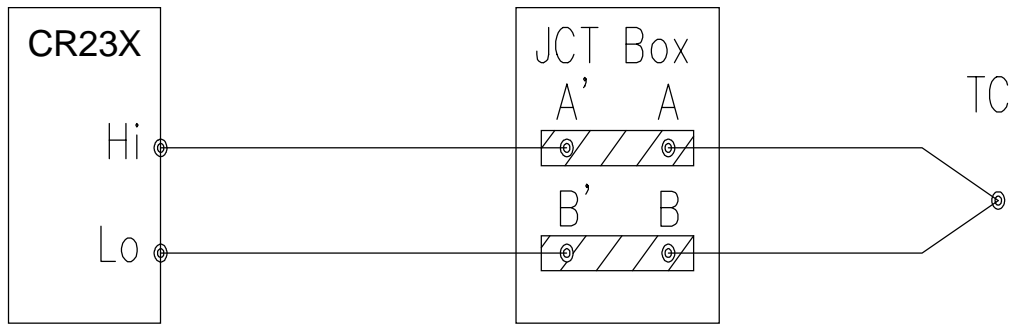
FIGURE 13.4-3. Diagram of Junction Box

An external reference junction box must be constructed so that the entire terminal area is very close to the same temperature. This is necessary so that a valid reference temperature can be measured, and to avoid a thermoelectric offset voltage which will be induced if the terminals at which the thermocouple leads are connected (points A and B in Figure 13.4-3) are at different temperatures. The box should contain elements of high thermal conductivity, which will act to rapidly remove any thermal gradients to which the box is subjected. It is not necessary to design a constant temperature box; it is desirable that the box respond slowly to external temperature fluctuations. Radiation shielding must be provided when a junction box is installed in the field. Care must also be taken that a thermal gradient is not induced by conduction through the incoming wires. The CR23X can be used to measure the temperature gradients within the junction box.

13.5 BRIDGE RESISTANCE MEASUREMENTS

There are 6 bridge measurement instructions included in the standard CR23X software. Figure 13.5-1 shows the circuits that would typically be measured with these instructions. In the diagrams, the resistors labeled R_s would normally be the sensors and those labeled R_f would normally be fixed resistors. Circuits other than those diagrammed could be measured, provided the excitation and type of measurements were appropriate.

With the exception of Instructions 4 and 8, which apply an excitation voltage then wait a specified time before making a measurement, all of the bridge measurements make one set of measurements with the excitation as programmed and another set of measurements with the excitation polarity reversed. The error

in the two measurements due to thermal emfs is then accounted for in the processing of the measurement instruction. The excitation is switched on 450µs before the integration portion of the measurement starts and is grounded as soon as the integration is completed. Figure 13.5-2 shows the excitation and measurement sequence for Instruction 6, a 4 wire full bridge. When more than one measurement per sensor is necessary (Instructions 7 and 9), excitation is applied separately for each measurement. (For example, in Instruction 9 (6 wire full bridge), the differential measurement of the voltage drop across the sensor is made with the excitation at both polarities; excitation is again applied and reversed for the measurement of the output voltage.)

Instruction 8 applies an excitation voltage, delays a specified time, and makes a differential voltage measurement. If a delay of 0 is specified, the inputs for the differential measurement are not switched for a second integration as is normally the case (Section 13.2). The result stored is the voltage measured. Instruction 8 does not have as good resolution or common mode rejection as the ratiometric bridge measurement instructions. It does provide a very rapid means of making bridge measurements as well as supplying excitation to circuitry requiring differential measurements. This instruction does not reverse excitation. A 1 before the excitation channel number (1X) causes the channel to be incremented with each repetition. The output of Instruction 8 is simply the voltage measurement. When 8 is used to measure a full bridge (same connections as Instruction 6 in Figure 13.5-1), the result is V_1 which equals V_x ($R_3/(R_3+R_4)$ $R_2/(R_1+R_2)$). (In other words, to make the output the same as Instruction 6, use a factor of $1000/V_x$ in the multiplier.)

INSTR. #	DIAGRAM	DESCRIPTION	RESULT = X
4		DC HALF BRIDGE WITH USER ENTERED SETTLING TIME	$X = V_1 = V_x \dfrac{R_s}{R_s + R_f}$
5		AC HALF BRIDGE EXCITATION ALTERNATES POLARITY FOR ION DEPOLARIZATION	$X = \dfrac{V_1}{V_x} = \dfrac{R_s}{R_s + R_f}$
6		4 WIRE FULL BRIDGE	$X = 1000 \dfrac{V_1}{V_x} = 1000 \left(\dfrac{R_3}{R_3 + R_4} - \dfrac{R_2}{R_1 + R_2} \right)$
7		3 WIRE HALF BRIDGE	$X = \dfrac{2V_2 - V_1}{V_x - V_1} = \dfrac{R_s}{R_f}$
9		6 WIRE FULL BRIDGE WITH EXCITATION LEAD COMPENSATION	$X = 1000 \dfrac{V_2}{V_1} = 1000 \left(\dfrac{R_3}{R_3 + R_4} - \dfrac{R_2}{R_1 + R_2} \right)$ (V_1 ON 5 V RANGE)
9		4 WIRE HALF BRIDGE	$X = \dfrac{V_2}{V_1} = \dfrac{R_s}{R_f}$ (V_1 NOT ON 5 V RANGE)

FIGURE 13.5-1. Circuits Used with Instructions 4-9
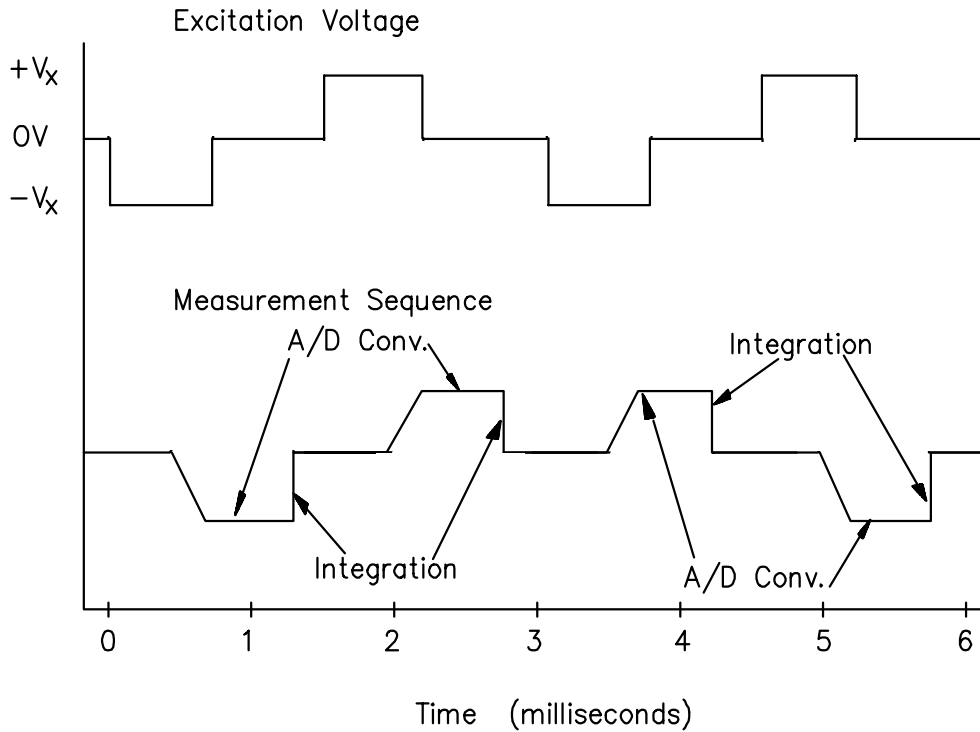
FIGURE 13.5-2. Excitation and Measurement Sequence for 4 Wire Full Bridge

---

TABLE 13.5-1. Comparison of Bridge Measurement Instructions

Instr.	Circuit	Description
4	DC Half Bridge	User entered settling time allows compensation for capacitance in long lead lengths. No polarity reversal. One single-ended measurement. Measured voltage output.
5	AC Half Bridge	Rapid reversal of excitation polarity for ion depolarization. One single-ended measurement at each excitation polarity. Ratiometric output.
6	4 Wire Full Bridge	Slightly lower noise than 9. One differential measurement at each excitation polarity. Ratiometric output.
7	3 Wire Half Bridge	Compensates for lead wire resistance, assuming resistance is same in both wires. Two single-ended measurements at each excitation polarity. Ratiometric output.
8	Differential Measurement with Excitation	Makes a differential measurement without reversing excitation polarity or switching inputs. Used for fast measurements on load cells, PRTs etc. Resolution and common mode rejection worse than 6 if used with delay=0. Measured voltage output.
9	6 Wire Full Bridge or 4 Wire Half Bridge	Compensates for lead wire resistance. Two differential measurements at each excitation polarity. Ratiometric output.

Calculating the actual resistance of a sensor which is one of the legs of a resistive bridge usually requires the use of one or two Processing Instructions in addition to the bridge measurement instruction. Instruction 59 takes a value, X, in a specified input location and computes the value MX/(1-X), where M is the multiplier and stores the result in the original location. Instruction 42 computes the reciprocal of a value in an input location. Table 13.5-2 lists the instructions used to compute the resistance of any single resistor shown in the diagrams in Figure 13.5-1, provided the values of the other resistors in the bridge circuit are known.

TABLE 13.5-2. Calculating Resistance Values from Bridge Measurement

Instr.	Result		Instr.#	Multiplier;	Offset
4	$X = V_x(R_s/(R_s+R_f))$				
	$R_s = R_f \dfrac{X/V_x}{1-X/V_x}$		4 59	$1/V_x$ R_f	0
	$R_f = \dfrac{1}{((X/V_x)/(1-X/V_x))/R_s}$		4 59 42	$1/V_x$ $1/R_s$	0
5	$X = R_s/(R_s+R_f)$				
	$R_s = R_f \dfrac{X}{1-X}$		5 59	1 R_f	0
	$R_f = \dfrac{1}{(X/(1-X))/R_s}$		5 59 42	1 $1/R_s$	0
6,8,9*	$X = 1000 [R_3/(R_3+R_4)-R_2/(R_1+R_2)]$		*used for full bridge		
	$R_1 = \dfrac{1}{(X_1/(1-X_1))/R_2}$ where $X_1 = -X/1000 + R_3/(R_3+R_4)$		6 or **9** 8 **6,8**, or **9** 59 42	-0.001; $1/V_x$ $1/R_2$	$R_3/(R_3+R_4)$
	$R_2 = R_1(X_2/(1-X_2))$ where $X_2 = X_1$		6 or **9** 59	-0.001 R_1	$R_3/(R_3+R_4)$
	$R_3 = R_4(X_3/(1-X_3))$ where $X_3 = X/1000 + R_2/(R_1+R_2)$		6 or **9** 59	0.001 R_4	$R_2/(R_1+R_2)$
	$R_4 = \dfrac{1}{(X_4/(1-X_4))/R_3}$ where $X_4 = X_3$		6 or **9** 59 42	0.001 $1/R_3$	$R_2/(R_1+R_2)$
7&9*	$X = R_s/R_f$		* used as half bridge		
	$R_s = R_f X$		**7** or **9**	R_f	0
	$R_f = R_s/X$		**7** or **9** 42	$1/R_s$	0

13.6 RESISTANCE MEASUREMENTS REQUIRING AC EXCITATION

Some resistive sensors require AC excitation. These include the 207 relative humidity probe, soil moisture blocks, water conductivity sensors, and wetness sensing grids. The use of DC excitation with these sensors can result in polarization, which will cause an erroneous measurement, and may shift the calibration of the sensor and/or lead to its rapid decay.

The AC half bridge Instruction 5 (incorporated into the 207 relative humidity measurement Instruction 12) reverses excitation polarity to provide ion depolarization and, in order to minimize the time excitation is on, grounds the excitation as soon as the signal is integrated (Figure 13.6-1). The slow integration time should never be used with a sensor requiring AC excitation because it results in the excitation lasting about 20 times as long, allowing polarization to affect the measurement.
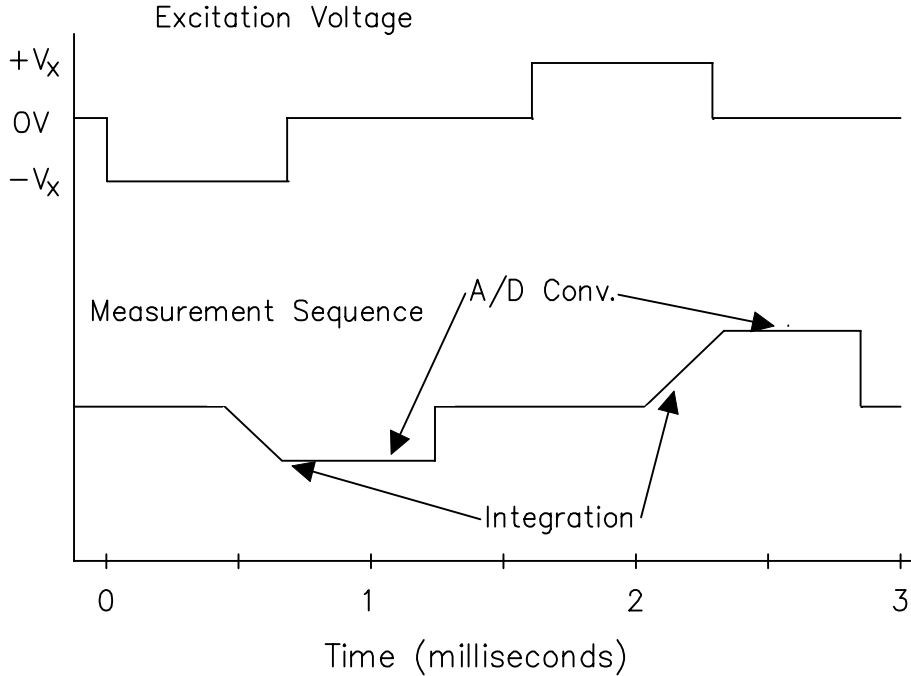
FIGURE 13.6-1. AC Excitation and Measurement Sequence for AC Half Bridge
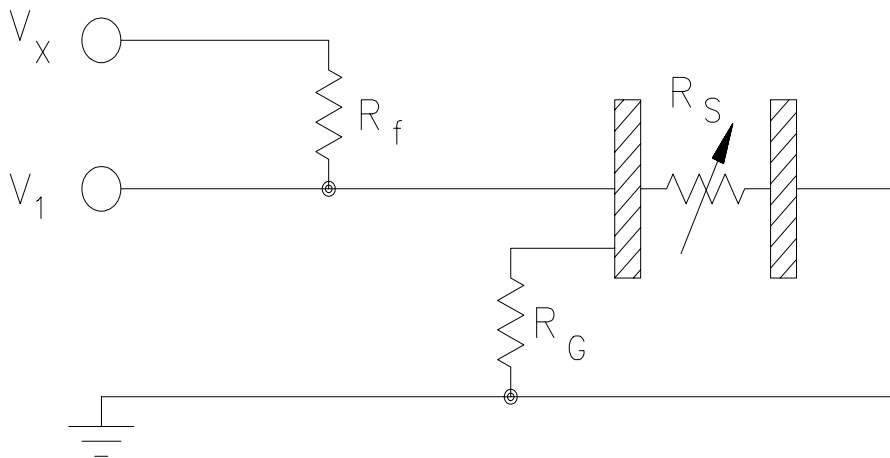
FIGURE 13.6-2. Model of Resistive Sensor with Ground Loop

INFLUENCE OF GROUND LOOP ON MEASUREMENTS

When measuring soil moisture blocks or water conductivity, the potential exists for a ground loop which can adversely affect the measurement. This ground loop arises because the soil and water provide an alternate path for the excitation to return to CR23X ground, and can be represented by the model diagrammed in Figure 13.6-2.

In Figure 13.6-2, V_x is the excitation voltage, R_f is a fixed resistor, R_s is the sensor resistance, and R_G is the resistance between the excited electrode and CR23X earth ground. With R_G in the network, the measured signal is:

$$V_1 = V_x \frac{R_s}{(R_s + R_f) + R_s R_f / R_G} \qquad [13.6\text{-}1]$$

$R_s R_f/R_G$ is the source of error due to the ground loop. When R_G is large the equation reduces to the ideal. The geometry of the electrodes has a great effect on the magnitude of this error. The Delmhorst gypsum block used in the 227 probe has two concentric cylindrical electrodes. The center electrode is used for excitation; because it is encircled by the ground electrode, the path for a ground loop through the soil is greatly reduced. Moisture blocks which consist of two parallel plate electrodes are particularly susceptible to ground loop problems. Similar considerations apply to the geometry of the electrodes in water conductivity sensors.

The ground electrode of the conductivity or soil moisture probe and the CR23X earth ground form a galvanic cell, with the water/soil solution acting as the electrolyte. If current was allowed to flow, the resulting oxidation or reduction would soon damage the electrode, just as if DC excitation was used to make the measurement. Campbell Scientific probes are built with series capacitors in the leads to block this DC current. In addition to preventing sensor deterioration, the capacitors block any DC component from affecting the measurement.

13.7 CALIBRATION PROCESS

The CR23X makes voltage measurements by integrating the input signal for a fixed time and then holding the integrated value for the analog to digital (A/D) conversion. The A/D conversion is made by a 15 bit successive approximation

using a digital to analog converter (DAC). The result from the approximation is DAC counts, which are multiplied by coefficients to obtain millivolts (mV).

The CR23X has an internal calibration function that feeds positive and negative voltages through the amplifiers and integrator and calculates new calibration coefficients. By adjusting the calibration coefficients the accuracy of the voltage measurements is maintained over the -25 to +50°C operating range of the CR23X. Calibration is executed under four conditions:

1. When the CR23X is powered up.

2. Automatically when Instruction 24 is not contained in a program table.

3. When the watchdog resets the processor.

4. When the calibration instruction, Instruction 24, is executed.

AUTOMATIC CALIBRATION SEQUENCE

The primary advantage of automatic calibration is that the CR23X is constantly calibrated without user programming. The CR23X defaults to automatic calibration when Instruction 24 is not contained in a program table.

The CR23X performs an automatic self-calibration in background to compensate for device changes due to aging along with more rapid changes due to fluctuations in environmental temperature.

The CR23X only calibrates those parameters that are necessary for the users datalogger program. As a result, the time required for a complete self-calibration in background depends upon the number of ranges selected by the user. The CR23X performs a segment of the background calibration every 4 seconds, and each unique voltage input range (13 total) consists of four segments. The background self-calibration always contains at least nine segments because the $\pm$ 5000 mV input range is calibrated and then used for a background internal and panel temperature measure. As an example, if the $\pm$ 5000 mV, $\pm$ 200 mV, and $\pm$ 50 mV input ranges are selected in a program, then the total self-calibration would consist of 9 + 8 = 17 segments

and would complete in background every (17 segments) * (4 sec/segment) = 68 seconds. If a user selects all possible input range codes, with the exception of either 60 Hz or 50 Hz rejection, but not both, along with period averaging, the background calibration would consist of 44 total segments and would require (44 segments) * (4 sec / segment) = 176 seconds (2.9 minutes) to complete in background. If this rate of update for self-calibration is too slow, because of possible rapid thermal fluctuations, then a user can force a self-calibration to be done every scan with the P24 (calibrate) instruction.

INSTRUCTION 24 CALIBRATION

The alternative to automatic calibration is the use of Instruction 24, the calibration instruction. Instruction 24 implements a complete calibration which occurs ONLY when EXECUTED by a program table. Instruction 24 calibration is the average of 10 calibrations, and takes approximately 2.8 seconds to complete. Automatic calibration is disabled when a program is compiled that contains Instruction 24.

Instruction 24 calibration, as opposed to automatic calibration, may be advantageous in applications where: 1) the CR23X is exposed to extreme thermal gradients, or 2) automatic calibration would interfere with the desired sampling rate, and the ambient temperature is stable enough to allow calibration at specific points during program execution.

Calibration coefficients are replaced each time that Instruction 24 is executed. Unlike

automatic calibration, there is no time constant for the coefficients to respond in changes to calibration. Instruction 24 calibration ensures that the coefficients are optimum at the time that the instruction is executed. For example, consider a CR23X mounted under the dash of an automobile, where temperature could easily change 50 degrees. Temperature changes affect the measurement circuitry which must be compensated for by calculating new coefficients. Each time Instruction 24 is executed a new set of calibration coefficients is calculated based on the measurements made at that time.

Calibration at a certain point during program execution may be advantageous for some applications. For example, suppose Table 2 has an execution time of 15.6 ms, but only executes when flag 1 is set. Table 1 has a 5 minute execution time which makes a temperature measurement, and sets flag 1 if the temperature exceeds a fixed value. To prevent overrun errors which would occur in Table 2 if the automatic calibration was used, Instruction 24 could be executed before the temperature measurement was made by Table 1.

Instruction 24 also has an option to store the results of the automatic calibration in Input Storage. This can be used to detect hardware problems. If -99999 appears in any of the input locations reporting calibration values, the CR23X has a hardware problem or needs factory calibration.

SECTION 14. INSTALLATION AND MAINTENANCE

14.1 PROTECTION FROM THE ENVIRONMENT

The normal environmental variables of concern are temperature and moisture. The standard CR23X is designed to operate reliably from -25 to +50°C (-40°C to +80°C, optional) in noncondensing humidity. When humidity tolerances are exceeded, damage to IC chips, microprocessor failure, and/or measurement inaccuracies due to condensation on the various PC board runners may result. Effective humidity control is the responsibility of the user.

The CR23X is not hermetically sealed. Two half unit packets of DESI PAK desiccant are located by the batteries. This desiccant should, in most cases, keep the interior components dry between battery changes. A dry package weighs approximately 19 grams and will absorb a maximum of six grams of water at 40% humidity and 11 grams at 80%. Desiccant packets can be dried out by placing the packets in an oven at 120°C for 16 hours (desiccant only, not the CR23X).

Campbell Scientific offers two enclosures with back plates for housing the CR23X datalogger and associated peripherals. The back plates have a grid of square holes punched on one inch centers. The black CR23X datalogger base has four holes at the corners used to mount the datalogger. Small rubber mounting feet are installed in the four holes when shipped from the factory. The four rubber feet are removed to access the four mounting holes.

The holes are on a 6 by 8 inch spacing. An enclosure supply kit is shipped with the enclosures. In the supply kit, are plastic grommets and compatible screws. Place the grommets in appropriate square holes then attach the datalogger base to the enclosure back plate with the supplied screws.

The fiberglass enclosures are classified as NEMA 4X (water-tight, dust-tight, corrosion-resistant, indoor and outdoor use). A 1.25" diameter entry/exit port is located at the bottom of the enclosure for routing cables and wires. The enclosure door can be fastened with the hasp for easy access, or with the two supplied screws for more permanent applications. The white plastic inserts at the corners of the enclosure must be removed to insert the screws. Both enclosures are white for reflecting solar radiation, thus reducing the internal enclosure temperature.

The Model ENC 12/14 fiberglass enclosure houses the CR23X and one or more peripherals. Inside dimensions of the ENC 12/14 are 14"x12"x5.5", outside dimensions are 18"x13.5"x8.13" (with brackets); weight is 11.16 lbs.

The Model ENC 16/18 fiberglass enclosure houses the CR23X and several peripherals. Inside dimensions of the ENC 16/18 are 18"x16"x8¾", outside dimensions are 18½"x18¾"x10½" (with brackets); weight is 18 lbs.

TABLE 14.8-1. Typical Current Drain for Common CR23X Peripherals

Peripheral	Typical Current Drain (mA)	
	Quiescent	Active
AM25T	.5	1
AM416	.1	17
COM100	.5	1.8
COM200 Phone Modem	0.0012	140
CSM1	.5	18
MD9	1.2	80
RAD Modem and SC932 Interface	2.2	10-15
RF100-VHF 5 Watt Radio	30	1700
RF200-UHF 4 Watt Radio	30	1500
RF95 RF Modem	1.4	30
SDM-A04	10.5	--
SDM-CD16	6.0	11/LED lit
SDM-INT8	0.4	6.5
SDM-SW8A	3	6
SM192/SM716 Storage Module	0.25	3
VS1	.05	110

14.2 POWER REQUIREMENTS

The CR23X operates at a nominal 12 VDC. Below 11.0 V or above 16 volts the CR23X does not operate properly.

The CR23X is diode protected against accidental reversal of the positive and ground leads from the battery. Input voltages in excess of 18 V may damage the CR23X and/or power supply. A transzorb provides transient protection by limiting voltage at approximately 20 V.

System operating time for the batteries can be determined by dividing the battery capacity (amp-hours) by the average system current drain. The CR23X typically draws 2.0 mA (2.5 mA maximum) in the quiescent state (with display off), 7.0 mA with display on, 45 mA while processing, and 70 mA during an analog measurement; the length of operating time for each datalogger instruction is listed in Section 3.
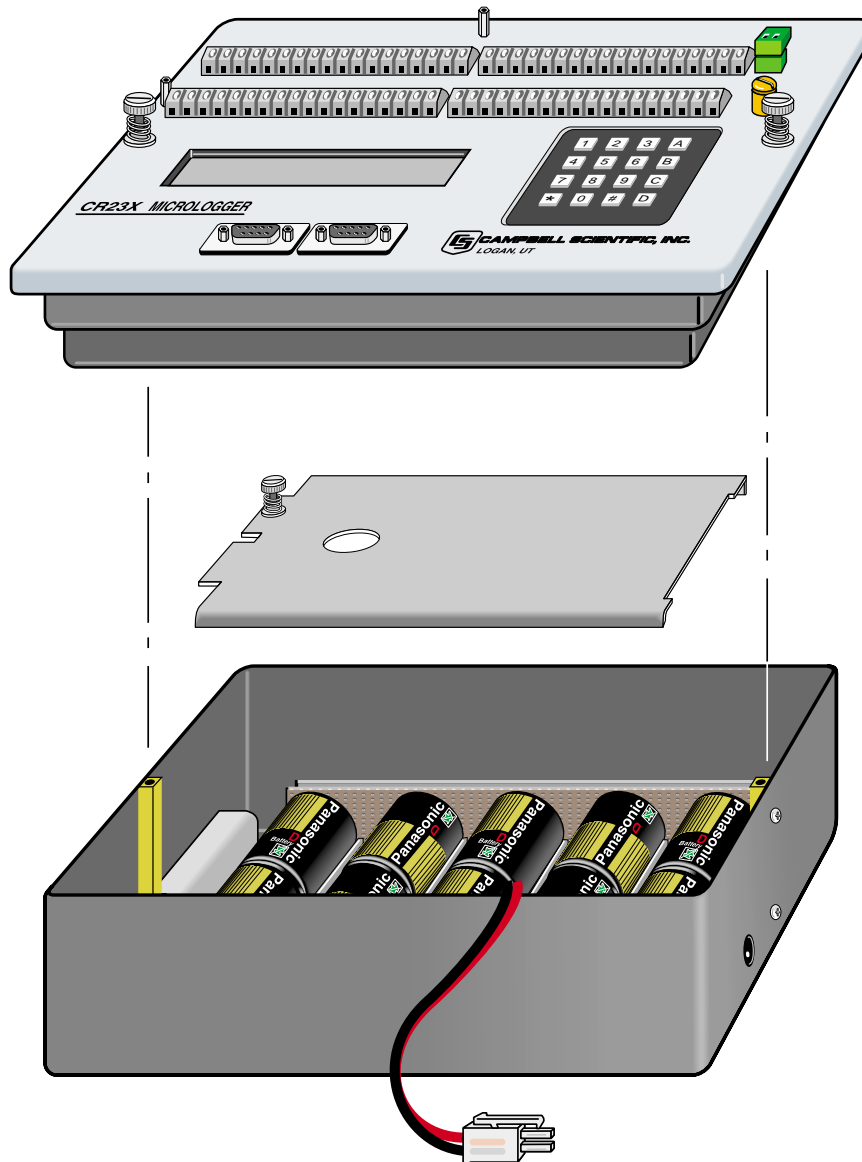
FIGURE 14.3-1. CR23X Battery Pack and Panel

14.3 CR23X POWER SUPPLIES

The CR23X is available with either alkaline or lead acid battery options. It may also be purchased without a battery option.
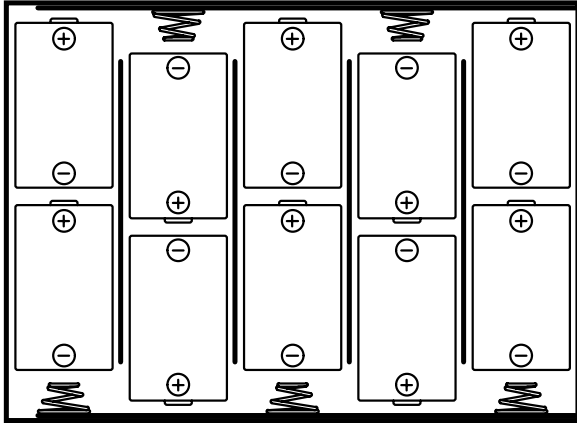
10 D-TYPE ALKALINE BATTERIES

FIGURE 14.3-1. Alkaline Battery Orientation

While the CR23X has a wide operating temperature range (-40 TO +80°C), the battery bases have more limited operating ranges. **Exceeding these ranges will degrade battery capacity and lifetime and could also cause permanent damage.**

14.3.1 CR23X ALKALINE BATTERY BASE

Install alkaline batteries as shown in Figure 14.3-1.

Temperature range: -25° to +50°C (Table 14.3-2).

The CR23X utilizes 10 alkaline D cells. After battery installation, the power switch on the side of the CR23X will turn the datalogger on and off.

A fresh set of ten high-quality alkaline D cells can be expected to have about 10 amp hours at 20°C.

Voltage with ten new D cells is about 15.5 V. If external power is connected to the CR23X with the alkaline power supply, power will be drawn from the source of highest voltage. In a typical configuration using an external lead-acid battery, the D cell voltage will discharge from 15 V to about 12 V before the lead-acid battery starts to supply current. This means the D cells

will lose approximately 35% (15 Ah - 9.8 Ahr = 5.2 Ah) of their capacity before the external power takes over.

The amp-hour rating decreases with temperature as shown in Table 14.3-2. Datalogger Instruction 10 can be used to monitor battery voltage. Replace the alkaline cells before the CR23X battery voltage drops below 11.0 V.

A thermal fuse in the alkaline base will allow a current drain of 1.85 A at 20°C (1.3 A at 50°C). In event of a short, the thermal fuse will cut power after about 40 seconds.

TABLE 14.3-2. Typical Alkaline Battery Service and Temperature

Temperature (°C)	% of 20°C Service
20 - 50	100
15	98
10	94
5	90
0	86
-10	70
-20	50
-30	30

NOTE: This data is based on one "D" cell under conditions of 50 mA current drain with a 30 ohm load. As the current drain decreases, the percent service improves for a given temperature.

14.3.2 CR23X LEAD ACID BATTERY BASE

Temperature range: -40° to +60°C (Table 14.3-3).

Charging voltage: 17 to 24 VDC or 18 V RMS AC.

The CR23X includes a 12V, 7.0 amp-hour lead acid battery, an AC transformer (18 V RMS AC), and a temperature compensated charging circuit with a charge indicating LED (Light Emitting Diode). An AC transformer or solar panel should be connected to the base at all times. The charging source powers the CR23X while float charging the lead acid batteries. The internal lead acid battery powers the datalogger if the charging source is interrupted. The lead

acid battery specifications are given in Table 14.3-3.

The leads from the charging source connect to a wiring terminal plug on the side of the base. Polarity of the leads to the connector does not matter. A transzorb provides transient protection to the charging circuit. A sustained input voltage in excess of 40V will cause the transzorb to limit voltage.

The red light (LED) on the base is on during charging with 17 to 24 VDC or 18 V RMS AC. The switch turns power to the CR23X on or off. Battery charging still occurs when the switch is off.

Should the lead acid batteries require replacement, consult Figure 14.3-2 for wiring.
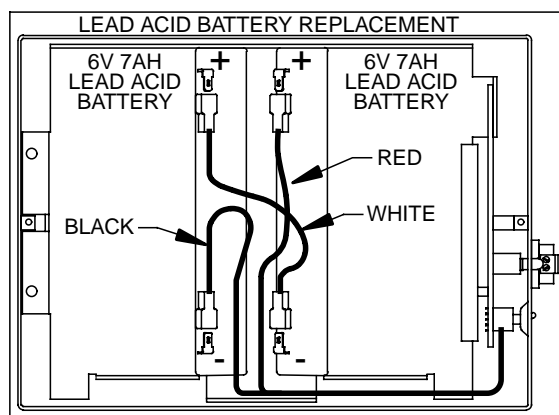
FIGURE 14.3-2. Lead Acid Battery Wiring

Monitor the power supply using datalogger Instruction 10. Users are strongly advised to incorporate this instruction into their data acquisition programs to keep track of the state of the power supply. If the system voltage level consistently decreases through time, some element(s) of the charging system has failed. Instruction 10 measures the voltage internal to the CR23X, not the voltage of the lead acid battery. The measured voltage will normally be about 0.3 V less than the voltage at the internal or external 12 V input. This voltage drop is on account of a Schottkey diode. External power sources must be disconnected from the CR23X to measure the actual lead acid battery voltage.

TABLE 14.3-3. CR23X Rechargeable Battery and AC Transformer Specifications

Lead Acid Battery

Battery Type	Yuasa NP7-6
Float Life @ 25°C	3 years minimum
Capacity	7.0 amp-hour
Shelf Life, full charge	6 months
Charge Time (AC Source)	40 hr full charge, 20 hr 95% charge
Operating temperature	-15°C to 50°C charge -20°C to 60°C discharge

AC Transformer

Input:	120 VAC, 60 Hz
Isolated Output:	18 VAC 1.11 Amp

There are inherent hazards associated with the use of sealed lead acid batteries. Under normal operation, lead acid batteries generate a small amount of hydrogen gas. This gaseous by-product is generally insignificant because the hydrogen dissipates naturally before build-up to an explosive level (4%) occurs. However, if the batteries are shorted or overcharging takes place, hydrogen gas may be generated at a rate sufficient to create a hazard. Campbell Scientific makes the following recommendations:

1. A CR23X equipped with standard lead acid batteries should NEVER be used in applications requiring INTRINSICALLY SAFE equipment.

2. A lead acid battery should not be housed in a gas-tight enclosure.

14.3.3 LOW PROFILE CR23X

The low profile CR23X option is not supplied with a battery base. See Section 14.5 and 14.6 for external power connection considerations.

14.4 SOLAR PANELS

Auxiliary photovoltaic power sources may be used to maintain charge on lead acid batteries.

When selecting a solar panel, a rule-of-thumb is that on a stormy overcast day the panel should provide enough charge to meet the system current drain (assume 10% of average annual global radiation, kW/m^2). Specific site information, if available, could strongly influence

the solar panel selection. For example, local effects such as mountain shadows, fog from valley inversion, snow, ice, leaves, birds, etc. shading the panel should be considered.

Guidelines are available from the Solarex Corporation for solar panel selection called "DESIGN AIDS FOR SMALL PV POWER SYSTEMS". It provides a method for calculating solar panel size based on general site location and system power requirements. If you need help in determining your system power requirements contact Campbell Scientific's Marketing Department.

Connect solar panel leads to the green connector at the side of the CR23X next to the power switch (rechargeable base only). Polarity does not matter.

14.5 DIRECT BATTERY CONNECTION TO THE CR23X WIRING PANEL

For some applications, size restrictions or other operational considerations may preclude the use of Campbell Scientific power supply options. In these cases the power supply may be connected directly to the CR23X. Any clean, battery backed 11 to 18 VDC supply may be connected to the 12 V and G connector terminals on the front panel. When connecting external power to the CR23X, first, remove the green power connector from the CR23X front panel. Insert the positive 12 V lead into the right-most terminal of the green connector. Insert the ground lead in the left terminal. Double check polarity before plugging the green connector into the panel.

Diode protection exists so that an external battery can be connected to the green G and 12 V power input connector, without loading or charging the internal batteries. The CR23X will draw current from the source with the largest voltage. When power is connected through the front panel, switch control on the standard CR23X power supplies is by-passed.

14.6 VEHICLE POWER SUPPLY CONNECTIONS

14.6.1 CR23X WITH BATTERY BASE

Some vehicle testing applications use the vehicle's 12 Volt power system as the primary supply for a CR23X with a battery base. However, when a vehicle's starting motor is engaged, the system voltage drops considerably below the nominal 12 volts needed for uninterrupted datalogger function. To counter this problem, diodes have been placed in the CR23X in series with the 12 V Power In connector. This allows the battery base to supply the needed voltage during motor start. The diodes also prevent the separate power systems of the CR23X and vehicle from attempting to charge each other.
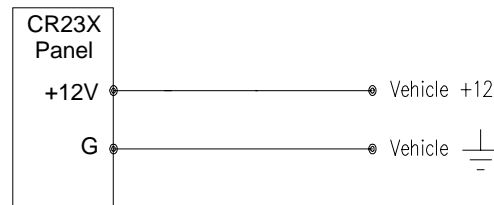
FIGURE 14.6-1. Connecting Vehicle Power Supply

If a CR23X without batteries is to be powered from the 12 Volts of a motor vehicle, a second 12 V supply is also required at the time of vehicle start-up. When the starting motor of a vehicle with a 12 V electrical system is engaged, the voltage drops considerably below the nominal 12 V, which would cause the CR23X to stop measurement every time the vehicle is started. The second 12 V supply prevents this malfunction. Figure 14.6-2 shows the general case for connecting the two supplies to the CR23X. The diode allows the vehicle to power the CR23X without the second supply attempting to power the vehicle. To reduce the potential for ground reference errors in measurements, the ground lead should be 16 AWG or larger.
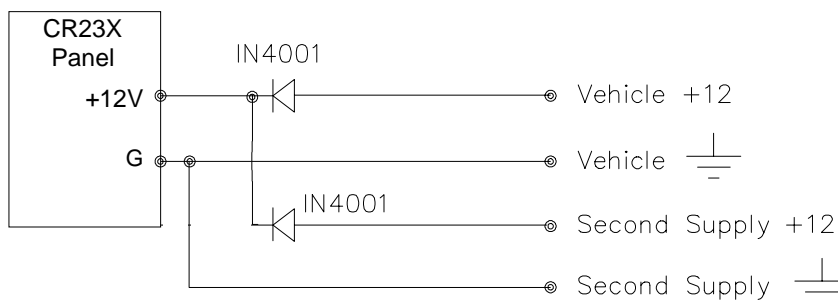
FIGURE 14.6-2. Connecting to Vehicle Power Supply

14.7 CR23X Grounding

Grounding of the CR23X and its peripheral devices and sensors is critical in all applications. Proper grounding will ensure the maximum ESD (electrostatic discharge) protection and the higher measurement accuracy.

14.7.1 ESD PROTECTION

An ESD (electrostatic discharge) can originate from several sources. However, the most common, and by far potentially the most destructive, are primary and secondary lightning strikes. Primary lightning strikes hit the datalogger or sensors directly. Secondary strikes induce a voltage in power lines or sensor wires.

The primary devices for protection against ESD are gas-discharge tubes (GDT). All critical inputs and outputs on the CR23X are protected with GDTs (RS-232, CS I/O and 12 Vin do not have GDTs). The GDTs fire at 150 V to allow current to be diverted to the earth ground lug. To be effective, the earth ground lug must be properly connected to earth (chassis) ground. As shown in Figure 14.7-1, the power ground and signal ground are independent lines until joined inside the CR23X.

Tie analog signal shields and returns to grounds (⏚) located in analog input terminal strips.

Tie CAO and pulse-counter returns into grounds (⏚) in CAO and pulse-counter terminal strip. Large excitation return currents may also be tied into this ground in order to minimize induced single-ended offset voltages in half bridge measurements.

External Power Input

Star Ground at Ground Lug ⏚

Tie 5 V, SW12, 12 V and C1-C8 returns into power grounds (G).

FIGURE 14.7-1. Schematic of CR23X Grounds

The 9-pin serial I/O ports on the CR23X are another path for transients to enter and damage the CR23X. Communications devices such a telephone or short-haul modem lines should have spark gap protection. Spark gap protection is often an option with these products, so it should always be requested when ordering. The spark gaps for these devices must be connected to either the CR23X earth ground lug, the enclosure ground, or to the earth (chassis) ground.

A good earth (chassis) ground will minimize damage to the datalogger and sensors by providing a low resistance path around the

system to a point of low potential. Campbell Scientific recommends that all dataloggers in use be earth (chassis) grounded. All components of the system (dataloggers, sensors, external power supplies, mounts, housings, etc.) should be referenced to one common earth (chassis) ground.

In the field, at a minimum, a proper earth ground will consist of a 6 to 8 foot copper sheathed grounding rod connected to the CR23X Ground Lug with a 12 AWG wire. In low conductive substrates, such as sand, very dry soil, ice, or rock, a single ground rod will probably not provide an adequate earth ground.

For these situations, consult the literature on lightning protection or contact a qualified lightning protection consultant. An excellent source of information on lightning protection can be located via the web at http://www.polyphaser.com.

In vehicle applications, the earth ground lug should be firmly attached to the vehicle chassis with 12 AWG wire or larger.

In laboratory applications, locating a stable earth ground is not always obvious. In older buildings, new cover plates on old AC sockets may indicate that a safety ground exists when in fact the socket is not grounded. If a safety ground does exist, it is good practice to verify that it carries no current. If the integrity of the AC power ground is in doubt, also ground the system through the buildings, plumbing or another connection to earth ground.

14.7.2 EFFECT OF GROUNDING ON MEASUREMENTS: COMMON MODE RANGE

The common mode range is the voltage range, relative to the CR23X ground, within which both inputs of a differential measurement must lie in order for the differential measurement to be made. Common mode range for the CR23X is ±5.0 V. For example, if the high side of a differential input is at 2 V and the low side is at 0.5 V relative to CR23X ground, a measurement made on the ±5.0 V range would indicate a signal of 1.5 V. However, if the high input changed to 6 V, the common mode range is exceeded and the measurement may be in error.

Common mode range may be exceeded when the CR23X is measuring the output from a sensor which has its own grounded power supply and the low side of the signal is referenced to the sensors power supply ground. If the CR23X ground and the sensor ground are at sufficiently different potentials, the signal will exceed the common mode range. To solve this problem, the sensor power ground and the CR23X ground should be connected, creating one ground for the system.

In a laboratory application, where more than one AC socket may be used to power various sensors, it is not safe to assume that the power grounds are at the same potential. To be safe, the ground of all the AC sockets in use should be tied together with a 12 AWG wire.

14.7.3 EFFECT OF GROUNDING ON SINGLE-ENDED MEASUREMENTS

Single-ended analog voltage measurements (Instructions P1, P4, P5, P7, P11, P12, and P13) are referenced to the CR23X ground. This makes single-ended measurements particularly sensitive to fluctuations in the CR23X ground potential. The ground potential will fluctuate proportional to how much power is being used by, and being sourced from, the CR23X. Peripheral devices such as multiplexers and many sensors source power from the CR23X. The grounding scheme in the CR23X has been designed to minimize the effects of ground potential fluctuations. To take advantage of this design, observe the following grounding rule:

> **NOTE:** Always connect a device's ground next to the active terminal associated with that ground.

Examples:

1. Connect 5 Volt, 12 Volt, and control grounds to G terminals.

2. Connect excitation grounds to the closest ⏚ terminal on the upper right CR23X terminal block.

3. Connect the low side of single-ended sensors to the nearest ⏚ terminal on the upper or lower left terminal blocks.

4. Connect shield wires to the nearest ⏚ terminal on the upper or lower left terminal blocks.

If offset problems occur because of shield or ground leads with large current flow, tying the problem leads into the ⏚ terminals next to the excitation, CAO, and pulse-counter channels should help. Problem leads can also be tied directly to the ground lug to minimize induced single-ended offset voltages.

14.8 POWERING SENSORS AND PERIPHERALS

The CR23X is a convenient source of power for sensors and peripherals requiring a continuous or semi-continuous 5 VDC or 12 VDC source. The CR23X has 2 continuous 12 Volt (12V) supply terminals, 1 switched 12 Volt (SW12)

supply terminal, and 1 continuous 5 Volt (5V) supply terminal. Voltage on the 12V and SW12 terminals will change with the CR23X supply voltage. The 5V terminal is regulated and will always remain near 5 Volts (±4%)so long as the CR23X supply voltage remains above 11 Volts. The 5V terminal is not suitable for resistive bridge sensor excitation, however. Table 14.8-1 shows current sourcing limitations of the 12 Volt and 5 Volt ports. Table 14.8-2 shows current requirements for several CSI peripheral. Other devices normally have current requirements listed in their specifications. Current drain of all peripherals and sensors combined should not exceed current sourcing limits of the CR23X.

Table 14.8-1 Current Sourcing Limits

Terminals	Current Source Limit
SW12	< 900 mA @ 20°C
	< 729 mA @ 40°C
	< 600 mA @ 50°C
	< 567 mA @ 60°C
	< 400 mA @ 80°C
12V + SW12	< 1.85 A @ 20°C
	< 1.50 A @ 40°C
	< 1.30 A @ 50°C
	< 1.17 A @ 60°C
	< 0.85 A @ 80°C
5V + CSI/O	< 200 mA

Make certain that the primary source of power for the CR23X can sustain the current drain for the period of time required. Contact a CSI applications engineer for help in determining a power budget for applications that approach the limits of a given power supply's capabilities. Be particularly cautious about any application using alkaline batteries and some form of telemetry, applications using solar panels and cellular telephone or radio, applications requiring long periods of time between site visits, or applications at extreme temperatures.

14.9 CONTROLLING POWER TO SENSORS AND PERIPHERALS

Controlling power to an external device is a common function of the CR23X. Whether control is required as part of the function of a system function or to conserve power, the principles of power control are the same.

Many devices can conveniently be controlled with the SW12 (Switched 12 Volt) port on the CR23X. Table 14.8-1 shows the current sourcing limits of the SW12 port. Applications requiring more control channels or greater power sourcing capacity can usually be satisfied with the use of Campbell Scientific's A21REL-12 Four Channel Relay Driver, A6REL-12 Six Channel Relay Driver, SDM-CD16AC 16 Channel AC/DC Relay Module, or by using the control (C1-C8) ports as described in Section 14.9.1

14.9.1 USE OF DIGITAL I/O PORTS FOR SWITCHING RELAYS

Each of the eight digital I/O ports can be configured as an output port and set low or high (0 V low, 5 V high) using I/O Instruction 20, Port Set, or commands 41 - 68 associated with Program Control Instructions 83 through 93. A digital output port is normally used to operate an external relay driver circuit because the port itself has a limited drive capability (2.0 mA minimum at 3.5 V).

Figure 14.9-1 shows a typical relay driver circuit in conjunction with a coil driven relay which may be used to switch external power to some device. In this example, when the control port is set high, 12 V from the datalogger passes through the relay coil, closing the relay which completes the power circuit to a fan, turning the fan on.

In other applications it may be desirable to simply switch power to a device without going through a relay. Figure 14.9-2 illustrates a circuit for switching external power to a device without going through a relay. If the peripheral to be powered draws in excess of 75 mA at room temperature (limit of the 2N2907A medium power transistor), the use of a relay (Figure 14.9-1) would be required.

Other control port activated circuits are possible for applications with greater current/voltage demands than shown in Figures 14.9-1 and 2. For more information contact a Campbell Scientific applications engineer.
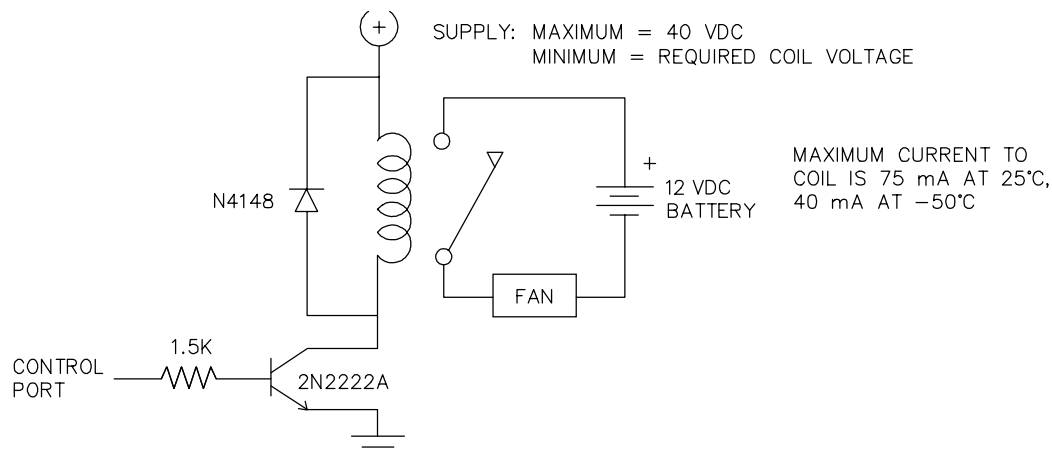
FIGURE 14.9-1. Relay Driver Circuit with Relay
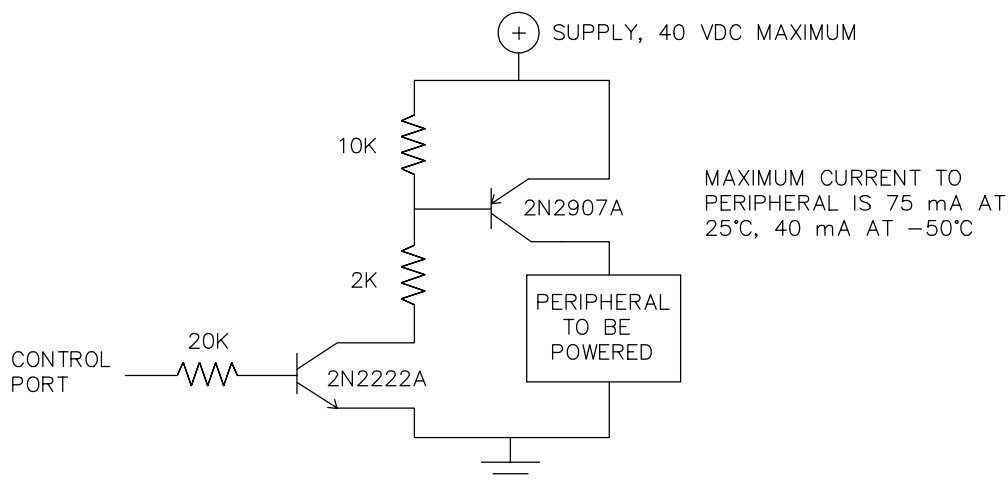
FIGURE 14.9-2. Power Switching without Relay

14.10 MAINTENANCE

The CR23X power supplies require a minimum of routine maintenance.

When not in use, the rechargeable supply should be stored in a cool, dry environment with the AC charging circuit activated.

The alkaline supply should not drop below 11.0 V before replacement. When not in use, remove the ten cells to eliminate potential corrosion of contact points and store in a cool dry place.

14.10.1 DESICCANT

The CR23X is shipped with desiccant to reduce humidity. Desiccant should be changed periodically. To prevent corrosion in uncontrolled or condensing atmospheres, the CR23X must be placed inside a weather tight instrument enclosure with desiccant. If only alkaline batteries are used, the enclosure may be sealed to inhibit vapor transfer into the enclosure. Do not seal the enclosure if lead acid batteries are present. Hydrogen gas generated by the batteries may build up to an explosive concentration.

14.10.2 REPLACING THE INTERNAL BATTERY

> **CAUTION:** Misuse of the lithium battery or installing it improperly can cause severe injury. Fire, explosion, and severe burn hazard! Do not recharge, disassemble, heat above 100°C (212°F), solder directly to the cell, incinerate, nor expose contents to water.

The CR23X contains a lithium battery that operates the clock and SRAM when the CR23X is not powered. The CR23X does not draw any power from the lithium battery while it is powered by a 12 VDC supply. In a CR23X stored at room temperature, the lithium battery should last approximately 10 years (less at temperature extremes). Where the CR23X is powered most or all of the time the lithium cell should last much longer.

While powered from an external source, the CR23X measures the voltage of the lithium battery daily. This voltage is displayed in the B Mode window 8 (Section 1.6) and in response to the telecommunications Status command (Section 5). A new battery will have approximately 3 volts. The battery should be replaced when the voltage drops below 2.4 volts. If the lithium cell is removed or allowed to discharge below the safe level, the CR23X will still operate correctly while powered. Without the lithium battery, the clock will reset and data will be lost when power is removed.

A replacement lithium battery can be purchased from Campbell Scientific or from an Electronics store (e.g., Radio Shack). Table 14.10-1 lists the specifications of the battery.

Table 14.10-1 CR23X Lithium Battery Specifications

Model	Sanyo	CR17335 SE (3V)
Capacity		1800 mAh
self discharge rate		1%/year @ 25°C
Diameter		17.0 mm
Length		33.5 mm
Temperature range		
Operating		-40°C to 85°C
Storage		-40°C to 60°C

The CR23X must be partially disassembled to replace the lithium cell.

The battery is replaced as shown in Figures 14.10-4 to 14.10-6. The battery is held in place by a band clamp. It can be removed by gently prying the band from the sides of the battery holder.

FIGURE 14.10-4. Removal of CR23X back plate and lithium battery location.

The new cell is placed into the battery holder, observing the polarity markings on the holder. Replace the band clamp, ensuring that both ends snap securely into the battery holder.
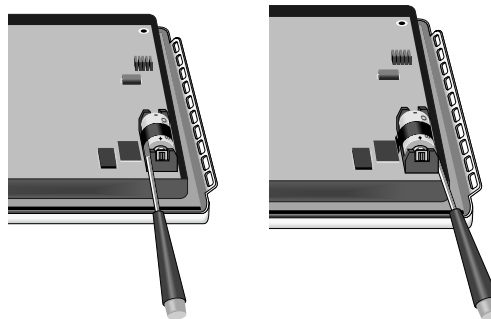
FIGURE 14.10-5. Loosening of band clamp.
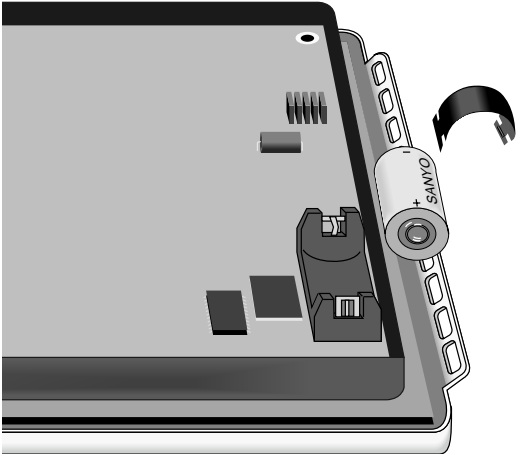
FIGURE 14.11-6. Removal of band clamp and battery.

APPENDIX A. GLOSSARY

ASCII: Abbreviation for American Standard Code for Information Interchange (pronounced "askee"). A specific binary code of 128 characters represented by 7 bit binary numbers.

ASYNCHRONOUS: The transmission of data between a transmitting and a receiving device occurs as a series of zeros and ones. For the data to be "read" correctly, the receiving device must begin reading at the proper point in the series. In asynchronous communication, this coordination is accomplished by having each character surrounded by one or more start and stop bits which designate the beginning and ending points of the information (see Synchronous).

BAUD RATE: The speed of transmission of information across a serial interface, expressed in units of bits per second. For example, 9600 baud refers to bits being transmitted (or received) from one piece of equipment to another at a rate of 9600 bits per second. Thus, a 7 bit ASCII character plus parity bit plus 1 stop bit (total 9 bits) would be transmitted in 9/9600 sec. = .94 ms or about 1000 characters/sec. When communicating via a serial interface, the baud rate settings of two pieces of equipment must match each other.

DATA POINT: A data value which is sent to Final Storage as the result of an Output Instruction. Strings of data points output at the same time make up Output Arrays.

EXECUTION INTERVAL: The time interval between initiating each execution of a given Program Table. If the Execution Interval is evenly divisible into 24 hours (86,400 seconds), the Execution Interval will be synchronized with 24 hour time, so that the table is executed at midnight and every execution interval thereafter. The table will be executed for the first time at the first occurrence of the Execution Interval after compilation. If the Execution Interval does not divide evenly into 24 hours, execution will start on the first even second after compilation. See Section OV4.3.1 for information on the choice of an Execution Interval.

EXECUTION TIME: The time required to execute an instruction or group of instructions. If the execution time of a Program Table exceeds the table's Execution Interval, the Program Table will be executed less frequently than programmed (Section OV4.3.1 and 8.9).

FINAL STORAGE: That portion of memory allocated for storing Output Arrays. Final Storage may be viewed as a ring memory, with the newest data being written over the oldest. Data in Final Storage may be displayed using the ⁎ 7 Mode or sent to various peripherals (Sections 2, 3, and OV4.1).

GARBAGE: The refuse of the data communication world. When data are sent or received incorrectly (and there are numerous reasons this happens) a string of invalid, meaningless characters (garbage) results. Two common causes are: 1) a baud rate mismatch and 2) synchronous data being sent to an asynchronous device and vice versa.

HANDSHAKE, HANDSHAKING: The exchange of predetermined information between two devices to assure each that it is connected to the other. When not used as a clock line, the CLK/HS (pin 7) line in the CR23X is primarily used to detect the presence or absence of peripherals such as the Storage Module.

HIGH RESOLUTION: A high resolution data value has 5 significant digits and may range in magnitude from +.00001 to +99999. A high resolution data value requires 2 Final Storage locations (4 bytes). All Input and Intermediate Storage locations are high resolution. Output to Final Storage defaults to low resolution; high resolution output must be specified by Instruction 78.

INDEXED INPUT LOCATION: An Input location entered as an instruction parameter may be indexed by keying "C" before it is entered by keying "A"; two dashes (--) will appear at the right of the display. Within a loop (Instruction 87, Section 12), this will cause the location to be incremented with each pass through the loop. Indexing is also used with Instruction 75 to cause an Input location, which normally remains constant, to be incremented with each repetition.

INPUT STORAGE: That portion of memory allocated for the storage of results of Input and Processing Instructions. The values in Input Storage can be displayed and altered in the * 6 Mode.

INPUT/OUTPUT INSTRUCTIONS: Used to initiate measurements and store the results in Input Storage or to set or read Control/Logic Ports.

INSTRUCTION LOCATION NUMBER: As instructions are entered in a Program Table, they are numbered sequentially. The instruction location number gives the location of that instruction in the program sequence. When programming a table, the instruction location number and a P (e.g., 04: P00) prompts the user when it is time to enter an instruction.

INTERMEDIATE STORAGE: That portion of memory allocated for the storage of results of intermediate calculations necessary for operations such as averages or standard deviations. Intermediate storage is not accessible to the user.

LOOP: In a program, a series of instructions which are repeated a prescribed number of times, followed by an "end" instruction which exists the program from the loop.

LOOP COUNTER: Increments by 1 with each pass through a loop.

LOW RESOLUTION: The default output resolution. A low resolution data value has 4 significant decimal digits and may range in magnitude from +0.001 to +6999. A low resolution data value requires 1 Final Storage location (2 bytes).

MANUALLY INITIATED: Initiated by the user, usually with a keyboard, as opposed to occurring under program control.

MODEM/TERMINAL: Any device which: 1) has the ability to raise the CR23X's ring line or be used with the SC32A to raise the ring line and put the CR23X in the Telecommunications Command State and 2) has an asynchronous serial communication port which can be configured to communicate with the CR23X.

ON-LINE DATA TRANSFER: Routine transfer of data to a peripheral left on-site. Transfer is controlled by the program entered in the datalogger.

OUTPUT ARRAY: A string of data points output to Final Storage. Output occurs only when the Output Flag (Flag 0) is set. The first point of an Output Array is the Output Array ID, which gives the Program Table Number and the Instruction Location Number of the Instruction which sets the Output Flag. The data points which complete the Array are the result of the Output Processing Instructions which are executed while the Output Flag is set. The Array ends when the Output Flag is reset at the end of the table or when another Instruction acts upon the Output Flag. Output occurs only when the output flag is set. (Section 2.1)

OUTPUT INTERVAL: The time interval between initiations of a particular Output Array. Output occurs only when the Output Flag is set. The flag may be set at fixed intervals or in response to certain conditions (Sections OV4 and 1.2.1).

OUTPUT PROCESSING INSTRUCTIONS: Process data values and generate Output Arrays. Examples of Output Processing Instructions include Totalize, Maximize, Minimize, Average, etc. The data sources for these Instructions are values in Input Storage. The results of intermediate calculations are stored in Intermediate Storage. The ultimate destination of data generated by Output Processing Instructions is usually Final Storage but may be Input Storage for further processing. The transfer of processed summaries to Final Storage takes place when the Output Flag has been set by a Program Control Instruction.

PARAMETER: Used in conjunction with CR23X Program Instructions, parameters are numbers or codes which are entered to specify exactly what a given instruction is to do. Once the instruction number has been entered in a Program Table, the CR23X will prompt for the parameters by displaying the parameter number in the ID Field of the display.

PRINT DEVICE: Any device capable of receiving output over pin 6 (the PE line) in a receive-only mode. Printers, "dumb" terminals, and computers in a terminal mode fall in this category.

PRINT PERIPHERAL: See Print Device.

PROCESSING INSTRUCTIONS: These Instructions allow the user to further process input data values and return the result to Input Storage where it can be accessed for output processing.   Arithmetic and transcendental functions are included in these Instructions.

PROGRAM CONTROL INSTRUCTIONS: Used to modify the sequence of execution of Instructions contained in Program Tables; also used to set or clear flags.

PROGRAM TABLE: That portion of memory allocated for storing programs consisting of a sequence of user instructions which control data acquisition, processing, and output to Final Storage. Programming can be separated into 2 tables, each having its own user-entered Execution Interval. A third table is available for programming subroutines which may be called by instructions in Tables 1 or 2. The $*$ 1 and $*$ 2 Modes are used to access Tables 1 and 2. The $*$ 3 Mode is used to access Subroutine Table 3. The length of the tables is constrained only by the total memory available for programming (Section 1.5). Tables 1 and 2 have independent execution intervals. Table 1 execution has the higher priority; it may interrupt Table 2.

RING LINE (PIN 3): Line pulled high by an external device to "awaken" the CR23X.

SAMPLE RATE: The rate at which measurements are made. The measurement sample rate is primarily of interest when considering the effect of time skew (i.e., how close in time are a series of measurements). The maximum sample rates are the rates at which measurements are made when initiated by a single instruction with multiple repetitions.

SIGNATURE: A number which is a function of the data and the sequence of data in memory. It is derived using an algorithm which assures a 99.998% probability that if either the data or its sequence changes, the signature changes.

SYNCHRONOUS: The transmission of data between a transmitting and receiving device occurs as a series of zeros and ones. For the data to be "read" correctly, the receiving device must begin reading at the proper point in the series. In synchronous communication, this coordination is accomplished by synchronizing the transmitting and receiving devices to a common clock signal (see Asynchronous).

TABLE OVERRUNS: Skipped scans occurring when the CR23X program is too long for the execution interval. Table overruns can cause errors in pulse measurements.

THROUGHPUT: The throughput rate is the rate at which a measurement can be made, scaled to engineering units, and the reading stored in Final Storage. The CR23X has the ability to scan sensors at a rate exceeding the throughput rate (see SAMPLE RATE). The primary factor affecting throughput rate is the amount of processing specified by the user. In normal operation, all processing called for by an instruction must be completed before moving on the next instruction. The maximum throughput rate for a fast single-ended measurement is approximately 192 measurements per second (12 measurements, repeated 16 times per second). This rate is possible if the CR23X's self-calibration function is suspended (this is accomplished by entering Instruction 24 into Program Table 2 while leaving the Execution Interval 0 so Program Table 2 never executes).

When the self-calibration function is operating, the maximum throughput rate for a fast, single-ended measurement is 192 measurements per second (12 measurements, 16 times per second).

APPENDIX B. CR23X CONTROL PORT SERIAL I/O INSTRUCTION 15

B.1 SPECIFICATIONS

FUNCTION

Send/receive full duplex serial data through the CR23X control ports and Computer RS-232 9-pin serial port. Received serial data can be buffered. This prevents data from being lost from sensors that output data without hardware of software command prompting from the datalogger. Received data can also have simple ASCII filters applied to locate the beginning of the actual data set. This is especially useful when reading data from serial sensors such as Global Positioning System (GPS) Receivers.

> **NOTE:** When the CR23X RS-232 9-Pin port is used to read serial data from a sensor (i.e., P15 parameter 4 = 9), connection to a PC for normal PC208W communications is no longer available. To re-establish PC communications on the CR23X RS-232
> 9-pin port P15 parameter 4 must be set to a value other than 9 or P15 must be removed entirely from the program. The program must be re-compiled with these changes to have normal communications on the CR23X 9-pin RS-232 port.

INPUT

Data sent by the sensor can be ASCII data values, hexadecimal pairs in ASCII representation, or decimal encoded binary bytes (Section B3, Parameter 2).

OUTPUT

Output can be a preamble set of commands to the sensor or data from the CR23X's input locations (Section B3, Parameter 6).

BAUD RATES

Control ports (C5,C6,C7,C8 ONLY):
300, 1200, 2400, 4800

9-pin RS-232 Port on CR23X:
300, 1200, 2400, 4800, 9600, 19200, 38400, 76800

HANDSHAKE CONTROL LINES

(applies to control ports only)

REQUEST TO SEND or DATA TERMINAL READY (RTS/DTR)

The CR23X signals the sensor that it is ready to send or receive data. This line is always asserted at the start of the instruction when parameter 4, digit A is greater than zero. If no output is specified then the line acts like RTS and remains asserted until all input is received. If output is specified, the line acts like DTR, asserted during output (Section B3, Parameter 4).

CLEAR-TO-SEND (CTS)

The sensor signals the CR23X that it is ready to receive the preamble or data (Section B3, Parameters 3 and 9).

LOGIC LEVELS

RS-232 - Logical 1 is low voltage. RTS/DTR asserted is high voltage.
TTL   - Logical 1 is high voltage. RTS/DTR asserted is low voltage.

PARITY AND STOP BITS

RECEIVE

- ASCII and Hexadecimal Pairs in ASCII Representation:
      1 start bit, 8 data bits, no parity, 1 stop bit 8th bit ignored

- Binary to decimal equivalent:
      1 start bit, 8 data bits, no parity, 1 stop bit

TRANSMIT

1 start bit, 8 data bits, no parity, 1 stop bit

B.2 SELECTED OPERATING DETAILS

Standard RS-232 logic levels range from 3 to 25 volts and -3 to -25 volts. The 9-pin RS-232 Port on the CR23X will support this range of logic levels. The maximum and minimum input to CR23X control ports is 5 and 0 volts, respectively. **Inputs exceeding these limits may damage the CR23X**. Figure B1 gives a circuit which may be used to limit voltage levels to 0 to 5 volts. Each input (data and handshake) that is out of the 0 to 5 volt range requires one of these circuits, or equivalent.
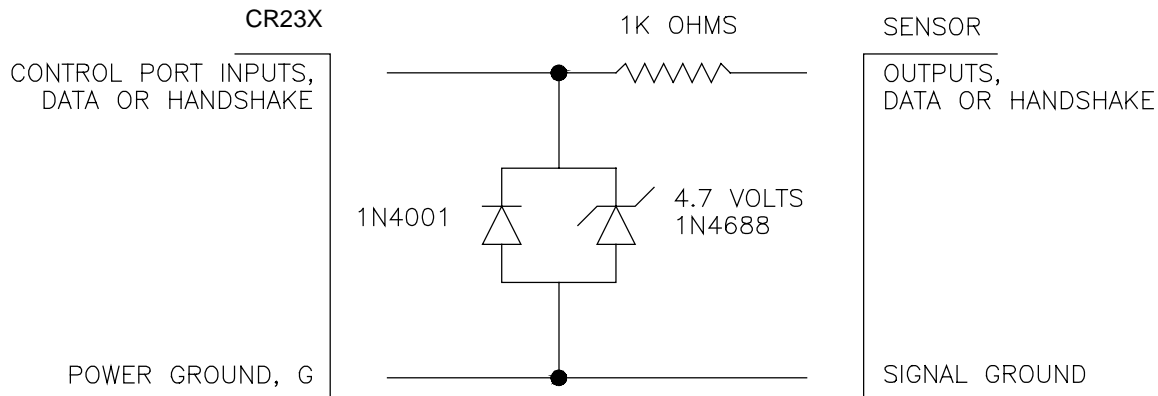
FIGURE B-1. Circuit To Limit Input to 0 to 5 Volts

B.3 INSTRUCTION 15 AND PARAMETER DESCRIPTIONS

PAR. NO.	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Configuration code (xy)
03:	4	CTS / Delay before send 0 = Wait for Clear to Send >0 = delay (.01 secs)
04:	2	Control ports (AB)
05:	4	Output start Loc
06:	4	Number of locations to send 0 = send nothing xxxx = locations; preamble xxxx-- = locations; data
07:	4	Input termination character (0..255)
08:	4	Maximum number of input characters
09:	4	Delay for CTS and/or Input (.01 secs)
10:	4	Input start location
11:	FP	Multiplier
12:	FP	Offset

INPUT LOCATIONS ALTERED determined by input

INTERMEDIATE LOCATIONS USED preamble size/4 + 1

EXECUTION TIME

If Parameters 3 and 6 are zero (0), maximum execution time is Parameter 9 times 0.01 seconds (input time out).

If Parameter 3 is 0 and 6 is not 0, maximum time is Parameter 9 times 0.01 secs times 2 (CTS and input time outs).

If Parameter 8 is 0 (no input), maximum time is Parameter 3 times .01 secs, or Parameter 9 times .01 secs if Parameter 3 = 0.

Add [10 bits/X bits/second] per byte output

X can equal 300, 1200, 2400, 4800, 9600, 19200, 38400, 76800

> **NOTE**: Times are shorter if CTS and/or data input is done before Parameter 9 time out.

PARAMETER 1 - REPETITIONS

Parameter 1 specifies the number of sensors that can be read using the same Instruction 15 parameter configuration. For example, a REP of 3 is used for 3 identical sensors or 3 sensors that are satisfied by the same parameter configuration. Instruction 15 must be entered separately for each sensor or group of sensors requiring a different configuration. A maximum of either 2 or 4 repetitions are possible depending on the configuration.

The CR23X sequentially increments sets of control ports and input locations with each repetition. The number of control ports used for each repetition (2, 3 or 4) depends on whether data is to be output and/or input, and the number of lines used to control the timing (Refer to Section B4, "Control Port Configurations and Sensor Wiring").

The starting output location specified in Parameter 5 is used for all repetitions.

PARAMETER 2 - CONFIGURATION CODE

The configuration code is a two-digit number specifying the input format, logic level, baud rate, buffering option, and optional decimal delimiter. Index (--) this parameter to indicate that you do not want to buffer received data.

Configuration Code

The configuration code is a two digit parameter in the form of *XY*, which specifies the input and output format. For detailed information about the configurations, see Configuration Codes Description.

X	*Description*
0	RS-232 ASCII
1	RS-232 ASCII Hex pair
2	RS-232 Binary
3	TTL ASCII
4	TTL ASCII Hex pair
5	TTL Binary

Y	*Description*
0	300 baud
1	1200 baud
2	2400 baud
3	4800 baud
4	9600 baud*
5	19200 baud*
6	38400 baud*
7	76800 baud*

*CSI 9-Pin RS-232 Port Only

> **NOTE:** Instruction 15 control port configurations can only allow combinations of 1200 and 2400 baud when used with 300 baud in the same datalogger program. The baud rate of 4800 cannot be used with 300 baud. Combinations of 1200, 2400, and 4800 baud are allowed.

ASCII

This option causes the CR23X to receive and decode an ASCII string of numbers into one or more data values. Data are assumed to consist of an optional sign ("+" or "-") followed by digits and an optional decimal point. Any other character that is not the termination character (Parameter 7) is assumed to be a delimiter between data values; however, the decimal point itself can be configured as a delimiter by a 6X or 7X configuration code enabling user to read higher resolution sensor data. Any digits before a "+" or "-" are discarded. The data is input assuming 1 start bit, 8 data bits, no parity and 1 stop bit, with the 8th (parity) bit ignored. All 8 bits are checked for the terminator character (Parameter 7).

Example: The input string

"-123.456,+1000,0000,2333,.0001*"

is converted to

-123.456, 1000.0, 0.0, 2333.0 and 0.0001

and placed in input locations starting at the location specified in Parameter 10. The termination character is "*" (ASCII equivalent 42).

Scientific notation is not valid. The string "+1.23E-12" is converted into two numbers: 1.23 and -12.0. The "E" is interpreted as a data point delimiter.

Hexadecimal Pairs in ASCII Representation

This option decodes ASCII representation of hexadecimal code into decimal values. The value from each pair is stored in separate input locations starting with the location specified in Parameter 10. The data is input assuming 1 start bit, 8 data bits, no parity and 1 stop bit, with the 8th (parity) bit ignored. All 8 bits are checked for the terminator character (Parameter 7).

Example: The input string

"7F7E0A0B0C1E<cr><lf>"

is translated into

127, 126, 10, 11, 12, 30.

The pair "7F" is equal to $7*16+15 = 112 + 15 = 127$. Any character with a decimal value below 48 (ASCII character 0) terminates the string. <cr> terminates the string in this example.

Binary to Decimal Equivalent

This option receives and converts each 8 bit byte to its decimal equivalent (0-255). The data is input assuming 8 data bits, no parity and 1 stop bit.

Example: The ASCII input string

"7F7E0A0B0C1E<cr><lf>"

is translated into

55, 70, 55, 69, 48, 65, 48, 66, 48, 67, 49, 69, 13, 10

and stored in successive input locations. Note this is the only option that does not discard the 8th (parity) bit.

PARAMETER 3 - CTS / DELAY BEFORE SEND

(applies to control port configuration only)

If Parameter 3 is zero (0), the CR23X waits for the Clear To Send to come high before sending output. If Clear To Send does not come high within the time specified in Parameter 9, output does not occur and -99999 is placed in the Input Memory Location specified in Parameter 10.

If Parameter 3 is non-zero, this parameter specifies the length of a delay in units of 0.01 seconds between asserting Request To Send and sending output. A CTS line is not used.

PARAMETER 4 - CONTROL PORT CONFIGURATION

A two-digit parameter (AB) indicates the control port(s) that will be used by the first repetition (A+1 and/or B+1 may also be used depending on the P15 CONFIGURATION selected). Control Ports 1, 2, 3, and 4 support hardware flow control (DTR/RTS and CTS lines). Control ports 5, 6, 7, and 8 support transmit and receive (TXD and RXD) data lines.

A is the control port used for DTR/CTS:
0   Don't use DTR/RTS and CTS
1   Control Port 1
2   Control Port 2
3   Control Port 3
4   Control Port 4

B is the control port used for TXD/RXD:
5   Control Port 5
6   Control Port 6
7   Control Port 7
8   Control Port 8

9   CR23X RS-232 Port

Example: An entry of 15 specifies Port 1 for DTR and Port 5 for either RXD using CONFIGURATION 1.

More information on control ports is given in Section B4, "Control Port Configuration and Sensor Wiring."

PARAMETER 5 - OUTPUT START LOCATION

Parameter 5 specifies the first input location containing data to be sent, if any.

PARAMETER 6 - NUMBER OF LOCATIONS TO SEND; PREAMBLE OR DATA

Parameter 6 specifies the number of input locations to send. A "--" following the number indicates that the datalogger needs to convert the values to ASCII. Entering only the number of input locations (i.e., no "--") implies that the values are in ASCII decimal equivalent code.

A "0" in Parameter 6 means to send nothing.

If outputting a preamble or command characters to a sensor, the characters must be in the ASCII decimal equivalent code (Appendix E). The preamble is stored in sequential input locations with one ASCII decimal equivalent code (0 to 127) in each location. The CR23X sends the preamble one ASCII character per input location. The decimal codes may be loaded into input locations using Instruction 30. Any negative value indicates that a "BREAK" should be sent. A "BREAK" is a spacing condition of more than 150 milliseconds.

If the number of locations to send is followed by "--", the CR23X sends an ASCII representation of the value in the input location. Seven ASCII characters are sent for each input location

(+ or -, five digits, and a decimal point). This feature is used most often when outputting data to another CR23X. Data points are separated by a space, and the string is terminated with a Carriage Return and Line Feed.

The data are transmitted with 1 start bit, 8 data bits, no parity, and one stop bit. For sensors requiring 7 data bits with 2 stop bits, add 128 to the last input location.

PARAMETER 7 - INPUT TERMINATION CHARACTER

The termination character is used to mark the end of the ASCII string received by the CR23X. Regardless of the configuration code, all 8 bits are used in determining the termination character. Thus, if the sensor is sending parity, it may be necessary to add 128 to the decimal value of the character.

If the termination character IS NOT encountered, input will continue until time out (Parameter 9) or until the maximum number of characters (Parameter 8) is received.

Characters received following the termination character are discarded.

PARAMETER 8 - MAXIMUM NUMBER OF CHARACTERS TO RECEIVE

Parameter 8 defines the total number of characters to expect per input, including numeric, non-numeric, polarity, decimal, space, and carriage return characters. For proper operation of the (ring) buffer pointer, you should specify the number of characters expected + 1. When the number of characters specified has been reached, any additional characters are discarded as if the termination character had been received.

Parameter 8 should be 0 if no input is expected.

PARAMETER 9 - DELAY FOR CLEAR-TO-SEND OR SERIAL INPUT

Parameter 9 is a 4 digit number that specifies a time out delay in units of 0.01 seconds. The CTS time out only applies to control port option 7. The time out specifies one of the following:

- The amount of time to wait for the CTS port to come high (Parameter 3=0 and Parameter 6>0). If the delay lapses, -99999 is stored in the input location specified by Parameter 10 and no data is output.

- The amount of time to wait for serial input (Parameter 8>0). If the delay lapses and some characters have been received, it acts like a termination character. If no characters have been received, a -99999 is stored to indicate a fault in communication.

- Both of the above (Parameter 3=0 and Parameter 6>0 and Parameter 8>0).

If output and input are completed before the end of the delay, program execution immediately advances to the next instruction. Thus, the delay may be over estimated without slowing down table execution.

If the delay expires, CR23X execution passes on to the next instruction.

The maximum delay is 99.99 seconds. If a longer delay is required, enter (-1) in Parameter 9. Under this condition, the program execution passes to the next instruction only when one of the other completion conditions (Parameters 7 or 8) are met. Consideration should be given to the power supply and current drain (45 mA) when completion of Instruction 15 is delayed for an extended period.

PARAMETER 10 - INPUT LOCATION

This parameter defines the location in the CR23X's Input Storage where the first received data value is stored. Multiple data values are stored in sequential input locations starting with the location specified by Parameter 10.

PARAMETERS 11 AND 12 - MULTIPLIER AND OFFSET

The data received by the CR23X can be scaled using the multiplier and offset parameters. A multiplier of 1 and offset of 0 are required to preserve the data in the same form that it was received.

B.4 CONTROL PORT CONFIGURATIONS AND SENSOR WIRING

The table below lists all possible input/output configurations.

Glossary:

DTR	Data Terminal Ready	Output (5V)
RTS	Request to Send	Output (5V)
TXD	Transmit Data	Output
CTS	Clear to Send	Input
RXD	Receive Data	Input

DTR/RTS are the same control port and are asserted high (5V) when digit A of Parameter 4 is greater than zero.

CONFIGURATION 1

Serial data are received as input only by the CR23X. No data or preamble are transmitted as output; the CTS line is not used. Parameter 6 must be 0 and Parameter 8 must be greater than zero. Parameter 4 is a 2-digit parameter (AB) and specifies the control port used for RTS output (A=1,2,3 or 4) and also the control port used as the serial input line (B=5,6,7 or 8). RTS in this example, when asserted (5V) tells the sensor that the CR23X is ready to accept characters. A total of 2 control ports are being used per repetition. A maximum of 4 repetitions can be used in this configuration.

TABLE B-1. Input/Output Configurations

Configuration Number	Parameters			Port Function				Ports Used
	3	6	8	DTR/RTS	CTS	TXD	RXD	
1 DTR,RX	x	0	nz	A	nc	nc	B	2
2 DTR,TX	nz	nz	0	A	nc	B	nc	2
3 DTR,CTS,TX	0	nz	0	A	A+1	B	B+1	3
4 DTR,TX, RX	nz	nz	nz	A	nc	B	B+1	3
5 DTR,CTS,TX,RX	0	nz	nz	A	A+1	B	B+1	4

RTS/DTR = Request to Send
            /Data Terminal Ready
    CTS = Clear to Send
      x = don't care
      A = control port to use for DTR/RTS and CTS

TXD = serial output
RXD = serial input
nz = non-zero number
nc = not connected
B   = control port to use for TXD and RXD

Parameter 3 = CTS/Delay before send
Parameter 6 = Number of Locations to Send
Parameter 8 = Maximum number of characters to receive

"A" and "B" are incremented by the number of "Ports Used" for each repetition.

CONFIGURATION 2

Serial data are transmitted from the CR23X without using the CTS line. There is no receive data input in this configuration. Parameters 3 and 6 must be greater than zero and Parameter 8 must be zero. Parameter 4 is a 2-digit parameter (AB) and specifies the control port used for the DTR output line (A = 1, 2, 3 or 4) and also the control port used for the serial output line (B = 5, 6, 7 or 8). When DTR output is asserted (5V), the CR23X is telling the sensor that it is getting ready to transmit characters immediately. A total of 2 control ports are being used per repetition. A maximum of 4 repetitions can be used in this configuration.

CONFIGURATION 3

Serial data are transmitted from the CR23X with the aid of the CTS Line. There is no received data input line used. Parameters 3 and 8 must be zero and Parameter 6 must be greater than zero. When DTR is asserted (5V) the CR23X is telling the sensor that it is getting ready to transmit characters immediately but it first checks the state of CTS. If the CTS input line is low (0V) then the Datalogger will not transmit characters to the sensor. Data are transmitted immediately after the delay in specified in Parameter 3.   Parameter 4 is a 2-digit parameter (AB) and specifies the control port used for the DTR line (A = 1, 2, 3 or 4) and also

the control port used for the serial output line (B = 5, 6, 7 or 8). A total of 3 control ports are being used per repetition. A maximum of 2 repetitions can be used in this configuration.

CONFIGURATION 4

Serial data are transmitted and received by the CR23X. The CTS line is not used. Parameters 3, 6 and 8 must be greater than zero. Parameter 4 is a 2-digit parameter (AB) and specifies the control port used for the RTS line (A = 1, 2, 3 or 4) and also the starting control port used for the serial output and serial input pairs (B = 5 and 6 or 7 and 8). Control ports 6 and 7 (TX/RX pairs) can also be used but this limits the repetitions to 1. RTS output in this example, when asserted (5V), tells the sensor that the CR23X is ready to receive characters. Data are transmitted immediately after the delay in specified in Parameter 3. A total of 3 control ports are being used per repetition. A maximum of 2 repetitions can be used in this configuration.

CONFIGURATION 5

Serial data are transmitted and received by the CR23X. The CTS line is used. Parameter 3 must be zero while Parameters 6 and 8 must be greater than zero. Parameter 4 is a 2-digit parameter (AB) and specifies the control port used for the RTS output and CTS input line pair

(A = 1 and 2 or 3 and 4). The starting control port used for the serial output and serial input pairs is specified as the second digit (B = 5 and 6 or 7 and 8). Control ports 2 and 3 (RTS/CTS) and 6 and 7 (TX/RX) pairs can also be used but this limits the repetitions to 1. RTS output in this example, when asserted (5V), tells the sensor that the CR23X is ready to receive characters. The CR23X will not transmit characters unless it sees the CTS input line high (5V). A total of 4 control ports are being used per repetition. A maximum of 2 repetitions can be used in this configuration.

B.5 RS-232 SERIAL DATA CONFIGURATION AND DATA BUFFERING

B.5.1 DATA INPUT VIA RS-232 9-PIN CONNECTOR

Serial data can be received through the CR23X 9-pin RS-232 port rather than through control ports C5 - C8 if you select for parameter 4 the single-digit option code of "9". The advantages of using the RS-232 port:

a. you can read in higher baud rate data streams from 9600 to 76800 baud (300 to 4800 also supported)

b. it allows RS-232 level sensors to be connected to the port without additional clamping circuitry

c. it provides optical isolation which can improve data integrity probability (P15 contains no error checking algorithms).

The RS-232 port is configured as a Data Communications Equipment (DCE) device. The original intention of this port was to provide direct connection to a PC serial port which is normally a Data Terminal Device or DTE. Since most serial sensors are also configured as a DCE device a null modem interface will be required. This can be accomplished through a simple cable as shown below or purchased from a local computer store.

Null Modem Connection Example

9-pin D to 9 pin D

SIGNAL	PIN No.		PIN No.
RXD	2	to	3
TXD	3	to	2
DTR	4	to	6
GND	5	to	5
DTR	6	to	4
RTS	7	to	8
CTS	8	to	7

25 pin D to 9 pin D

SIGNAL	25 PIN No.		9 PIN No.	
TXD	2	to	RXD	2
RXD	3	to	TXD	3
RTS	4	to	CTS	8
CTS	5	to	RTS	7
DSR	6	to	DTR	4
SG	7	to	SG	5
DTR	20	to	DSR	6

GND (SG)	- Ground (Signal GND)
TXD	- Transmit Data
RXD	- Receive Data
RTS	- Request to send
CTS	- Clear to send
DSR	- Data Send Ready
DCD	- Carrier Detect
DTR	- Data Terminal Ready

Null Modem Signal Connections

Connector 1 to   Connector 2

DTR	DSR/DCD
DSR/DCD	DTR
RTS	CTS
CTS	RTS
TXD	RXD
RXD	TXD
GND	GND

B.5.2 DATA BUFFERING

Since P15 is executed in a program table at fixed intervals (i.e. 60 seconds), it is possible to miss data transmitted from a sensor if the sensor outputs data automatically at a different interval (i.e., 10 seconds). P15 will support buffering of this data up to the maximum number of characters specified in parameter 8 minus 1 data byte. For example: if your application requires that 15 data bytes be buffered, make Parameter 8 equal to 16. If parameter 2 is indexed (--) no data buffering will occur.

B.6 INPUT DATA FILTERS.

P15 supports search filters strings that can be applied to a received data set. It is typically used locate the beginning of the desired data set. Multiple filters can be applied if the data is imbedded in a long string. To implement a filter string follow P15 immediately with either instructions P63 or P68 (both of which are extended parameter instructions). Multiple use of these instructions can be used if the search string is longer than 8 characters. Separate different types of filter strings with a null character (00). Once the first string has been found and the data decoded and loaded into input locations the next sequential string loaded into P63 or P68 (after the 00) will be used for any other incoming data. If the filter string is not found -99999 will be stored in the input location. Once the last string has been found and the data decoded and loaded into input locations, the first string loaded will be used again.

Enter the actual filter/search string by keying the decimal equivalent of the ASCII character (see Appendix E. of the CR23X Operators Manual).

Example: GPS receiver program

A common application with P15 is to read in data from a GPS receiver. The GPS receivers typically have a defined serial output protocol called NMEA 0183. NMEA 0183 is an interface protocol created by the National Marine Electronics Association (http://www.nmea.org/). The NMEA 0183 Standard defines electrical signal requirements, data transmission protocol, timing and specific sentence formats for a 4800 baud serial data bus.

The primary advantage of this data string when used with P15 is that data is separated with commas. Since P15 only interprets and decodes numerical values the commas are treated as data delimiters resulting in each navigation value residing in a unique input location.

Two common message strings that come out of GPS receivers are the GGA and VTG messages. The GGA message includes time, position and fix related data for the GPS receiver. The VTG message conveys the actual track made good and the speed relative to the ground. Example are listed below.

EXAMPLES:

GGA Position String - 63 characters including CRLF

$GPGGA,174548,4146.008,N,11151.240,W,1,6,001.9,1113,M,-017,M,,*40CRLF

Field	Data	Description
1.	$GPGGA	message identification
2.	174548	UTC of position (i.e., Greenwich mean Time)
3.	4146.008	Latitude
4.	N	N (North) or S (South)
5.	11151.240	Longitude
6.	W	W (West) or E (East)
7.	1	GPS Quality Indicator: 0=No GPS Available, 1=GPS Available, 2=DGPS
8.	6	Number of Satellites in use
9.	001.9	Horizontal Dilution of Precision (HDOP).
10.	1113	Antenna Altitude
11.	M	M = Meters
12.	-017	Geoidal Separation in Meters.
13.	M	M = Meters
14.	*40	Checksum
15.	CRLF	Carriage Return Line Feed

VTG Direction and Speed - 37 characters including CRLF

$GPVTG,010,T,356,M,5.39,N,10.0,K*4FCRLF

Field	Data	Description
1.	$GPVTG	message identification
2.	010	Track made good in degrees True
3.	T	True
4.	356	Track made good in degrees Magnetic
5.	M	Magnetic
6.	5.39	Speed over ground in Knots
7.	N	Knots
8.	10.0	Speed over ground in Kilometers per Hour
9.	K	Kilometers per Hour
10.	*4F	Checksum
11.	CRLF	Carriage Return Line Feed

The following program example will read-in the GGA and VTG GPS strings. Since this program is unique to the GPS application, *additional* data processing will be required to decode and format these strings. See Program Example #3 for a complete listing.

---

*Table 1 Program

01:	1	Execution Interval (seconds)

; READ IN SERIAL GPS DATA VIA NULL MODEM AND
; CR23X 9-PIN RS-232 PORT

1: Port Serial I/O (P15)

1:	1	Reps	
2:	03	Configuration Code	;RS-232 ASCII, 4800 BAUD
3:	1	Delay (0.01 seconds)	
4:	9	RS-232 Port	;GET GPS DATA VIA CSI 9-PIN RS-232 PORT
5:	0	Output Loc [_____]	
6:	0	No. of Locs to Send	
7:	42	Termination Character	;ASTERISK *
8:	100	Maximum Characters	;GGA + VTG CHARACTERS
9:	100	CTS/Input Wait	
10:	27	Loc [RAW_TIME]	;Input Location for first byte of received data
11:	1.0	Mult	
12:	0.0	Offset	

; CONFIGURE AND LOAD INPUT FILTER STRINGS

2: Extended Parameters (P63)

1:	36	Option	;$
2:	71	Option	;G
3:	80	Option	;P
4:	71	Option	;G
5:	71	Option	;G
6:	65	Option	;A
7:	0	Option	;END OF STRING 1
8:	36	Option	;$

3: Extended Parameters (P63)

1:	71	Option	;G
2:	80	Option	;P
3:	86	Option	;V
4:	84	Option	;T
5:	71	Option	;G
6:	00	Option	;END OF STRING 2
7:	00	Option	
8:	00	Option	

B.7 PROGRAM EXAMPLES

The following examples represent portions of a larger, more complete application program, and **SHOULD NOT BE USED VERBATIM**.

B.7.1 EXAMPLE 1: ATMOSPHERIC INSTRUMENTS RESEARCH AIR-DB-1A BAROMETER

Various barometer functions (mode, units of output, baud rate, etc.) are determined by jumpers internal to the barometer. A summary of the jumper configurations is found in Section B8. Detailed instructions for jumper placement are found in the AIR Barometer Operations Manual provided by AIR, Inc.

B.7.1.1 CR23X-BAROMETER CONFIGURATION LIMITATIONS

The CR23X is limited to reading only certain barometer jumper configurations as described below:

MODES of OPERATION

The CR23X can read all Modes of Operation except the Serial ID and Signature Analysis portion of the Test Mode. Serial input is required. The CR23X will NOT read a parallel input.

UNITS of OUTPUT

The CR23X reads all Units of Output.

NUMBER of MEASUREMENTS AVERAGED

Any number of "measurements averaged" may be read by the CR23X. Instruction 15 execution time is prolonged when the barometer makes several measurements to compute an average.

BAUD RATE

300 or 1200 baud

RS-232 JUMPERS

For typical field use where power is limiting, the RS-232 Jumpers should be removed to enable a TTL (0-5V) level output.

B.7.1.2 CR23X/BAROMETER HOOK-UP

Table 2 provides connection information for two TTL and one RS-232 version of data transfer. Each hook-up has specific advantages:

HOOK-UP #1
This hook-up demands the least power consumption (typically 1.0 mA quiescent, 6.0 mA active). If the barometer is jumpered to output the average of one measurement, the execution time of Instruction 15 is 0.5 to 0.6 seconds. The barometer's "shut down" and data transmit lines connect to the RTS and RX control ports (Parameter 4). TTL logic is specified in Parameter 2. When the instruction is run, the shut down line connected to RTS goes from 5 to zero volts. The barometer "wakes up", makes the measurement(s), and transmits the data. Upon receipt of the data, the RTS line goes to 5 volts, returning the barometer a low current drain "sleep" mode.

HOOK-UP #2
This hook-up results in greater power consumption than Configuration 1 (typically 5.5 mA quiescent, 6.0 mA active), but executes faster. If the barometer is jumpered to output the average of one measurement, Instruction 15 execution time is 0.2 to 0.3 seconds. The only difference between HOOK-UPs #1 and #2 is that the barometer's DTR line replaces the shut down line. The barometer is always "awake", leading to faster instruction execution at the expense of higher current drain.

HOOK-UP #3
The purpose of this hook-up is to read the barometer on a standard RS-232 device, such as a printer or terminal instead of the CR23X. If failure occurs, this hook-up allows using a third device to check if the problem is with the barometer or with the CR23X.

TABLE B-2. CR23X/Barometer Connection Details

	HOOK—UP #1	HOOK—UP #2	HOOK—UP #3	BAROMETER CONDUCTORS	
CONNECTIONS	CONTROL PORT 5	CONTROL PORT 5	INSULATE	BLK	DATA (TTL)
	INSULATE	INSULATE	PIN 3	BRN	DATA (RS232)
	⏚	FIRST CONTROL PORT	⏚	RED	DTR (TTL)
	INSULATE	INSULATE	PIN 20	ORG	DTR (RS232)
	FIRST CONTROL PORT	⏚	INSULATE	GRN	SHUT DOWN
	⏚	⏚	PIN 7	YEL	⏚
	+12VDC	+12VDC	+12VDC	BLU	+12VDC
	INSULATE	INSULATE	−12VDC	PRP	−12VDC
DEVICE	CR23X	CR23X	TERMINAL/PRINTER		
POWER SUPPLY	+12VDC	+12VDC	+12VDC		
SECONDS/ MEASUREMENT	0.5 TO 0.6	0.1 TO 0.2			
TYPICAL SYSTEM CURRENT DRAIN ACTIVE QUIESCENT	6.0mA 1.0mA	6.0mA 5.5mA			
BAROMETER OUTPUT	LSTTL	LSTTL	RS232		
RS232 DRIVER CHIP	DISABLE	DISABLE	ENABLE		

> **CAUTION:** To avoid damage to the barometer CMOS chips, connect ground first and power last. All sensor leads not used in a particular hook-up configuration should be insulated to avoid possible shorting.

B.7.1.3 INSTRUCTION 15 PARAMETER CONSIDERATIONS

The AIR Barometer requires only the CR23X's RTS line connected to its shut down or DTR line to transfer data. No CTS or output is required. It therefore uses 2 CR23X ports, as described by "CONFIGURATION 1" (Section B4).

If one AIR barometer is connected according to Hook-up #1 with only one sample output, the parameters are as follows:

01:	1	Repetitions
02:	00	Configuration code of ASCII/TTL/1200 bd
03:	0	CTS/Delay before send (does not matter)
04:	AB	Control ports used (User's choice). A = sensor Shut Down
05:	0	Output start location (no output from CR23X)
06:	0	Number of locations to send (no output from CR23X)
07:	13	Input termination character is Carriage Return

08:    9   Max characters to receive
09: 100   Delay for input is 1.0 second
10:    ?   First input location, User's choice
11:    1   Multiplier
12:    0.0000   Offset

The following paragraphs contain more detailed information on some of the parameters.

    Parameter 2 - Set to 00 to select the TTL logic levels.

    Parameter 4 - The A control port is connected to the "shut down" line (green); the B to the TTL dataline (black).

    Parameter 7 - The input termination character is carriage return (13).

    Parameter 8 - Total number of characters to expect per input for 1 barometric reading is 9. One value from the AIR barometer typically consists of an occasional garbage character, a sign, 5 data characters, 1 decimal point, and a carriage return.

When reading a serial input, Instruction 15 uses intermediate and input locations. One input location is required per data point. The number of intermediate locations consumed is dependent on the value of Parameter 8. To minimize the number of intermediate locations used, Parameter 8 should be equal or one greater than the number of characters per barometer output.

Table 3 provides information on the number of characters to expect from various barometer modes.

If the barometer output contains multiple data points, the input location specified in Parameter 10 is sequentially incremented for each data point.

    Parameter 9 - The time out parameter limits the amount of time that Instruction 15 waits to receive a input. The execution time of Instruction 15 is dependent on both the hook-up configuration and the barometer jumper configuration. Exact execution time must be verified, but an initial estimate may be calculated as follows:

HOOK-UP CONFIGURATION #1:
$ET = 0.1 \times MA + 0.9$

HOOK-UP CONFIGURATION #2:
$ET = 0.1 \times MA + 0.6$

WHERE: ET = Execution time in seconds

MA = Number of measurements averaged by the sensor

Parameter 9 Time out = ET/0.01

TABLE B-3. Number of Characters/Output and Memory Requirements for Various Barometer Output Modes

SERIAL OUTPUT MODE	CHARACTERS PER OUTPUT	DATA POINTS PER OUTPUT	INPUT LOC. REQUIRED	INTERM. LOC. REQUIRED
BAROMETER	8	1	1	3

ALTIMETER
  1 DATA POINT
  2 DATA POINTS
	
	8
	22
		
		1
		2
			
			1
			2
				
				3
				6

ALTIMETER SETTING	22	2	2	6

TEST
  RAW/PRESS
  RAW
	
	41
	33
		
		5
		4
			
			5
			4
				
				11
				9

Example: 100 samples are averaged by the barometer connected to the CR23X via hook-up #1.

ET = 0.1 * 100 + 0.9

ET = 10.9 seconds

Time out = 10.9/0.01

Time out = 1090

In general, the above equations overestimate the execution time. When execution speed is critical, the minimum time out delay must be determined through trial and error.

B.7.1.4 BAROMETER PROGRAM

The following example program assumes the barometer is jumpered for TTL output and the average of one measurement. The following considerations are accounted for in the program.

- To preserve the barometer measurement resolution, barometer data must be output to the CR23X Final Storage in High Resolution (OUTPUT PROCESSING INSTRUCTION 78).

- A communication failure shows up as a partial value or -99999. It is generally desirable to exclude this value from any processing and log the incidence of failure. A failure may be detected by comparing it to expected limits. A bad value may be excluded from processing by temporarily setting the Intermediate Processing Disable flag (flag 9). To log a failure the reading could be output immediately with the time of occurrence.

01:	P15	Control Port Serial I/O	
01:	1	Repetitions	
02:	31	Configuration code	*ASCII;TTL*
03:	1	CTS/Delay before send	
04:	15	C1*RTS/DTR* , C5 *TXD/RXD*	
05:	0	Output start location	
06:	0	Number of locations to send	
07:	13	Input termination character	*13 = CR*
08:	20	Max characters to receive	
09:	200	Delay for CTS/input	*Wait 1 sec for Input*
10:	1	First input location	
11:	1	Multiplier	*barometer jumpered*
12:	0	Offset	*for millibar output*
02:	P89	If X<=>F	*START: Limit testing*
01:	1	X Loc AIR, mb	
02:	3	>=	*If reading is ≥ 1300 mb*
03:	1300	F	
04:	11	Set high Flag 1	*then set Flag 1 high*
03:	P89	If X<=>F	
01:	1	X Loc AIR, mb	
02:	4	<	*If reading is < 800 mb*
03:	800	F	
04:	11	Set high Flag 1	*then set Flag 1*
04:	P91	If Flag/Port	*If Flag 1 is set,*
01:	11	Do if flag 1 is high	
02:	30	Then Do	*do the following.*
05:	P86	Do	*Output time of failure*
01:	10	Set high Flag 0 (output)	

06:	P77	Real Time	
01:	110	Day,Hour-Minute	
07:	P70	Sample	*Output out of range reading*
01:	1	Rep	
02:	1	Loc AIR, mb	
08:	P86	Do	*Set Intermediate Processing*
01:	19	Set high Flag 9	*Disable Flag*
09:	P86	Do	*Set Flag 1 low*
01:	21	Set low Flag 1	
10:	P95	End	*End of If Flag Then Do*
11:	P92	If time is	*Output mb twice/day*
01:	0	minutes into a	
02:	720	minute interval	
03:	10	Set high Flag 0 (output)	
12:	P77	Real Time	
01:	110	Day,Hour-Minute	
13:	P71	Average	
01:	1	Reps	
02:	1	Loc AIR, mb	
14:	P	End Table 1	

Input Location Labels:
1:AIR, mb

B.7.2 EXAMPLE 2: CR23X TO CR23X COMMUNICATION

The following example shows how data can be sent between CR23X's. The program for the CR23X sending the data, CR23X#1, includes a loop which creates data to send; the numbers are generated and stored in Input Locations 1 through 10. Instruction 15 then transfers the contents of Locations 1 through 10 to CR23X#2. Instruction 15 in CR23X#2 receives and stores the values in Input Locations 11 through 20.

As shown in Figure 2, the DTR from #1 is connected to control port 8 of CR23X#2. RS-232 logic is used so DTR is +5V when asserted. The DTR going high is connected to port 8 of CR23X#2, causing subroutine 98 to be executed (refer to CR23X manual, Instruction 85, Section 12). Subroutine 98 contains Instruction 15, programmed to receive input. CR23X#2's RTS is connected to CR23X#1's CTS. When CR23X#2 executes Instruction 15, the RTS going high signals CR23X#1 to send data.

CR23X#1 SEND DATA			CR23X#2 RECEIVE DATA
FUNCTION	PORT	PORT	FUNCTION
DTR	C1 ----------->C8		Interrupt; SBR 98
CTS	C2<----------- C1		RTS
TX	C5 ----------->C5		RX

FIGURE B-2. Wiring Diagram for CR23X to CR23X Communication Example

CR23X#1 PROGRAM - SEND DATA

*		1	Table 1 Programs	
	01:	5	Sec. Execution Interval	
01:		P87	Beginning of Loop	*Start loop to generate values*
	01:	0	Delay	
	02:	10	Loop Count	
02:		P32	Z=Z+1	
	01:	11	Z Loc [:COUNTER]	
03:		P31	Z=X	
	01:	11	X Loc COUNTER	
	02:	1--	Z Loc [:VALUE #1]	
04:		P95	End	*End Loop to generate values*
05:		P15	Control Port Serial I/O	
	01:	1	Repetitions	
	02:	0	Configuration code	*RS-232, 1200 baud*
	03:	0	CTS/Delay before send	
	04:	15	C1 RTS/DTR, C5 TXD/RXD	*DTR = C1, TX = C5*
	05:	1	Output start location	
	06:	10--	Number of locations to send	*Start at Loc 10 and (--) convert*
	07:	0	Input termination character	
	08:	0	Max characters to receive	
	09:	100	Delay for CTS/input	*wait 1 sec for CTS (C2) to go high*
	10:	28	First input location	*If 1 sec delay expires before CTS goes high, store -99999 in Loc 28*
	11:	1	Multiplier	
	12:	0	Offset	
06:		P	End Table 1	

Input Location Labels:

1:VALUE #1	4:VALUE #4	7:VALUE #7	10:VALUE #10
2:VALUE #2	5:VALUE #5	8:VALUE #8	11:COUNTER
3:VALUE #3	6:VALUE #6	9:VALUE #9	12:_____

CR23X#2 PROGRAM - RECEIVE DATA

*		3	Table 3 Subroutines	
01:		P85	Beginning of Subroutine	
	01:	98	Subroutine Number	
02:		P15	Control Port Serial I/O	
	01:	1	Repetitions	
	02:	0	Configuration code	*RS-232, 1200 bd*
	03:	0	CTS/Delay before send	
	04:	6	No RTS/DTR, C6 TXD/ RXD	
	05:	0	Output start location	
	06:	0	Number of locations to send	

07:	13	Input termination character	*Terminate on carriage return*
08:	75	Max characters to receive	
09:	200	Delay for CTS/input	*Wait 2 seconds for input*
10:	11	First input location	*Store VALUE #1 in Location 11, #10 in 20*
11:	1	Multiplier	
12:	0.0000	Offset	

03:       P95   End

Input Location Labels:

11:VALUE #1	16:VALUE #6
12:VALUE #2	17:VALUE #7
13:VALUE #3	18:VALUE #8
14:VALUE #4	19:VALUE #9
15:VALUE #5	20:VALUE #10

Program Example # 3

This program will decode the $GPGGA and $GPVTG strings, format them in input locations, and set the CR23X clock according to the GPS time. The primary reason for formatting the data strings is to parse long data strings into multiple input locations. The CR23X can display 5 significant digits. The maximum value that can be displayed in an input location is 99999. Data in the GPS string is anywhere from 1 to 8 characters. Data fields such as time has 6 digits (i.e. HHMMSS) and will exceed the 99999 input location value limit. To get around this problem, data will be parsed and separated into individual input locations.

If the GPS receiver is to be used to keep the CR23X clock updated (via P114) caution should be used when pulse count type measurements (i.e. wind speed, velocity) are made. This can increase or decrease the total number of counts based on if the datalogger clock was slow or fast. The program code below can be put into a conditional type statement (P92, P95) to be executed on an less frequent interval rather than once per second (i.e. once per day).

CR23X#3 PROGRAM EXAMPLE

*Table 1 Program
  01:            1   Execution Interval (seconds)

; LOOK FOR $GPGGA AND $GPVTG STRINGS

; READ IN SERIAL GPS DATA VIA NULL MODEM AND
; CR23X 9-PIN RS-232 PORT

01:    Port Serial I/O (P15)

1:	1	Reps	
2:	03	Configuration Code	;RS-232 ASCII, 4800 BAUD
3:	1	Delay (0.01 seconds)	
4:	9	RS-232 Port	;GET GPS DATA VIA CSI 9-PIN RS-232 PORT
5:	1	Output Loc [HOUR_MIN]	
6:	0	No. of Locs to Send	
7:	42	Termination Character	;ASTERISK *
8:	100	Maximum Characters	;GGA + VTG CHARACTERS
9:	100	CTS/Input Wait	
10:	27	Loc [RAW_TIME]	
11:	1.0	Mult	

```
        12:     0.0     Offset
```

; CONFIGURE AND LOAD INPUT FILTER STRINGS
; $GPGGA and $GPVTG

```
02:     Extended Parameters (P63)
        1:     36      Option                          ;$
        2:     71      Option                          ;G
        3:     80      Option                          ;P
        4:     71      Option                          ;G
        5:     71      Option                          ;G
        6:     65      Option                          ;A
        7:      0      Option                          ;END OF STRING 1
        8:     36      Option                          ;$
```

```
03:     Extended Parameters (P63)
        1:     71      Option                          ;G
        2:     80      Option                          ;P
        3:     86      Option                          ;V
        4:     84      Option                          ;T
        5:     71      Option                          ;G
        6:     00      Option                          ;END OF STRING 2
        7:     00      Option
        8:     00      Option
```

; PERFORM DATA FORMATTING

```
04:     If (X<=>F) (P89)
        1:     27      X Loc [ RAW_TIME  ]
        2:      3      >=
        3:      0      F
        4:     30      Then Do
```

; ONLY PROCESS DATA IF GPS DATA IS AVAILABLE

```
05:     If (X<=>F) (P89)
        1:     30      X Loc [ RAW_AVAIL ]
        2:      1      =
        3:      1      F
        4:     30      Then Do
```

; PARSE TIME INTO (2) INPUT LOCATIONS

```
06:     Z=X*F (P37)
        1:     27      X Loc [ RAW_TIME  ]
        2:     .01     F
        3:     27      Z Loc [ RAW_TIME  ]
```

```
07:     Z=INT(X) (P45)
        1:     27      X Loc [ RAW_TIME  ]
        2:      1      Z Loc [ HOUR_MIN  ]
```

```
08:     Z=FRAC(X) (P44)
        1:     27      X Loc [ RAW_TIME  ]
        2:      2      Z Loc [ SECONDS   ]
```

```
09:    Z=X*F (P37)
       1:        2        X Loc [ SECONDS  ]
       2:      100        F
       3:        2        Z Loc [ SECONDS  ]

10:    Z=INT(X) (P45)
       1:        2        X Loc [ SECONDS  ]
       2:        2        Z Loc [ SECONDS  ]
```

; ADD OFFSET TO ALLOW UPDATED TIME TO BE CORRECTLY UPLOADED INTO CR23X
; TYPICALLY THE SCAN RATE + 1 SECOND

```
11:    If (X<=>F) (P89)
       1:        2        X Loc [ SECONDS  ]
       2:        4        <
       3:       59        F
       4:       30        Then Do

12:    Z=X+F (P34)
       1:        2        X Loc [ SECONDS  ]
       2:        2        F
       3:        2        Z Loc [ SECONDS  ]

13:    End (P95)
```

; AT 60 SECONDS SET SECONDS TO ZERO

```
14:    If (X<=>F) (P89)
       1:        2        X Loc [ SECONDS  ]
       2:        1        =
       3:       60        F
       4:       30        Then Do

15:    Z=F (P30)
       1:        0        F
       2:        0        Exponent of 10
       3:        2        Z Loc [ SECONDS  ]

16:    End (P95)
```

; ADJUST OFFSET WHEN ROLLOVER TIME IS ZERO

```
17:    If (X<=>F) (P89)
       1:        2        X Loc [ SECONDS  ]
       2:        1        =
       3:       99        F
       4:       30        Then Do

18:    Z=F (P30)
       1:        1        F
       2:        0        Exponent of 10
       3:        2        Z Loc [ SECONDS  ]
```

```
19:   Z=X+F (P34)
      1:      1        X Loc [ HOUR_MIN  ]
      2:      1        F
      3:      1        Z Loc [ HOUR_MIN  ]

20:   End (P95)
```

; PARSE LONG NAVIGATION STRINGS.

```
21:   Z=INT(X) (P45)
      1:     28        X Loc [ RAW_LAT   ]
      2:      3        Z Loc [ LAT_DEG   ]

22:   Z=FRAC(X) (P44)
      1:     28        X Loc [ RAW_LAT   ]
      2:      4        Z Loc [ LAT_DDD   ]

23:   Z=INT(X) (P45)
      1:     29        X Loc [ RAW_LONG  ]
      2:      5        Z Loc [ LONG_DEG  ]

24:   Z=FRAC(X) (P44)
      1:     29        X Loc [ RAW_LONG  ]
      2:      6        Z Loc [ LONG_DDD  ]

25:   Z=X (P31)
      1:     30        X Loc [ RAW_AVAIL ]
      2:      7        Z Loc [ GPS_AVAIL ]

26:   Z=X (P31)
      1:     31        X Loc [ RAW_SATS  ]
      2:      8        Z Loc [ NO_SATS   ]

27:   Z=X (P31)
      1:     32        X Loc [ RAW_HDOP  ]
      2:      9        Z Loc [ HDOP      ]
```

; CONVERT ALTITUDE TO FEET

```
28:   Z=X*F (P37)
      1:     33        X Loc [ RAW_ALT   ]
      2:    3.281      F
      3:     10        Z Loc [ ALTITUDE  ]

29:   Z=X (P31)
      1:     35        X Loc [ RAW_TRUE  ]
      2:     11        Z Loc [ DEG_TRUE  ]

30:   Z=X (P31)
      1:     36        X Loc [ RAW_MAG   ]
      2:     12        Z Loc [ DEG_MAG   ]
```

; CONVERT SPEED FROM KPH TO MPH

```
31:    Z=X*F (P37)
       1:      38       X Loc [ RAW_SPDK ]
       2:   0.6214   F
       3:      13       Z Loc [ SPD_MPH   ]
```

; OUTPUT DATA EVERY INTERVAL

```
32:    Do (P86)
       1:      10       Set Output Flag High (Flag 0)

33:    Real Time (P77)
       1:    1220       Year,Day,Hour/Minute (midnight = 2400)

34:    Resolution (P78)
       1:       1       High Resolution

35:    Sample (P70)
       1:      13       Reps
       2:       1       Loc [ HOUR_MIN  ]
```

; SET DATALOGGER CLOCK WHEN VALID TIME AND GPS IS AVAILABLE EVERY INTERVAL

```
36:    If Flag/Port (P91)
       1:      28       Do if Flag 8 is Low
       2:      30       Then Do

37:    Do (P86)
       1:      18       Set Flag 8 High

38:    Z=X*F (P37)
       1:       1       X Loc [ HOUR_MIN  ]
       2:    0.01       F
       3:      39       Z Loc [ HOURS     ]

39:    Z=INT(X) (P45)
       1:      39       X Loc [ HOURS     ]
       2:      39       Z Loc [ HOURS     ]

40:    Z=X*F (P37)
       1:       1       X Loc [ HOUR_MIN  ]
       2:    0.01       F
       3:      40       Z Loc [ MINUTES__ ]

41:    Z=FRAC(X) (P44)
       1:      40       X Loc [ MINUTES__ ]
       2:      40       Z Loc [ MINUTES__ ]

42:    Z=X*F (P37)
       1:      40       X Loc [ MINUTES__ ]
       2:     100       F
       3:      40       Z Loc [ MINUTES__ ]

43:    Z=X (P31)
       1:       2       X Loc [ SECONDS   ]
       2:      41       Z Loc [ SEC       ]
```

; PERFORM RANGE CHECKING TO ENSURE GPS TIME DATA FALLS WITHIN A REASONABLE RANGE.

```
1: If (X<=>F) (P89)
   1:         39        X Loc [ HOURS    ]
   2:          4          <
   3:         24        F
   4:         30        Then Do

2: If (X<=>F) (P89)
   1:         39        X Loc [ HOURS    ]
   2:          3        >=
   3:          0        F
   4:         30        Then Do

3: If (X<=>F) (P89)
   1:         40        X Loc [ MINUTES   ]
   2:          4          <
   3:         60        F
   4:         30        Then Do

4: If (X<=>F) (P89)
   1:         40        X Loc [ MINUTES   ]
   2:          3        >=
   3:          0        F
   4:         30        Then Do

5: If (X<=>F) (P89)
   1:         41        X Loc [ SEC      ]
   2:          4          <
   3:         60        F
   4:         30        Then Do

6: If (X<=>F) (P89)
   1:         41        X Loc [ SEC      ]
   2:          3        >=
   3:          0        F
   4:         30        Then Do

7: Set Real Time Clock (P114)
   1:          0        Set Hr,Min,Sec from locations
   2:         39        Loc [ HOURS    ]

8: End (P95)

9: End (P95)

10: End (P95)

11: End (P95)

12: End (P95)

13: End (P95)
```

*Table 2 Program
  02:           0.0000   Execution Interval (seconds)

*Table 3 Subroutines

End Program

```
1   [ HOUR_MIN ]     RW--    5       2
2   [ SECONDS  ]     RW--    7       6
3   [ LAT_DEG  ]     -W--    0       1
4   [ LAT_DDD  ]     -W--    0       1
5   [ LONG_DEG ]     -W--    0       1
6   [ LONG_DDD ]     -W--    0       1
7   [ GPS_AVAIL ]    -W--    0       1
8   [ NO_SATS  ]     -W--    0       1
9   [ HDOP    ]      -W--    0       1
10  [ ALTITUDE ]     -W--    0       1
11  [ DEG_TRUE ]     -W--    0       1
12  [ DEG_MAG  ]     -W--    0       1
13  [ SPD_MPH  ]     -W--    0       1
14  [ _____ ]     ----    0       0
15  [ _____ ]     ----    0       0
16  [ _____ ]     ----    0       0
17  [ _____ ]     ----    0       0
18  [ _____ ]     ----    0       0
19  [ _____ ]     ----    0       0
20  [ _____ ]     ----    0       0
21  [ _____ ]     ----    0       0
22  [ _____ ]     ----    0       0
23  [ _____ ]     ----    0       0
24  [ _____ ]     ----    0       0
25  [ _____ ]     ----    0       0
26  [ _____ ]     ----    0       0
27  [ RAW_TIME ]     RW--    4       2
28  [ RAW_LAT  ]     R---    2       0
29  [ RAW_LONG ]     R---    2       0
30  [ RAW_AVAIL ]    R---    2       0
31  [ RAW_SATS ]     R---    1       0
32  [ RAW_HDOP ]     R---    1       0
33  [ RAW_ALT  ]     R---    1       0
34  [ RAW_GEOID ]    ----    0       0
35  [ RAW_TRUE ]     R---    1       0
36  [ RAW_MAG  ]     R---    1       0
37  [ RAW_SPDN ]     ----    0       0
38  [ RAW_SPDK ]     R---    1       0
39  [ HOURS    ]     RW--    4       2
40  [ MINUTES__ ]    RW--    5       3
41  [ SEC     ]      RW--    3       1
42  [ _____ ]     ----    0       0
43  [ _____ ]     ----    0       0
44  [ _____ ]     ----    0       0
```

B.8 SUMMARY OF BAROMETER JUMPER CONFIGURATIONS

	Jumper #	7	6	5	4	3	2	1	0
Barometer mode (serial output)		u	u	r	r	n	n	0	0
Barometer mode (parallel output)		u	u	0	b	n	n	0	1
Altimeter mode (serial output)		u	u	r	r	n	n	1	0
Altimeter mode (parallel output)		u	u	0	b	n	n	1	0
Altimeter setting mode (serial output)		r	r	1	0	n	n	1	1
Altimeter setting mode (parallel output)		*	*	1	1	n	n	1	1
Test mode (serial output)		r	r	1	1	t	t	0	1

Units of output - Barometer modes		7	6	5	4	3	2	1	0
	mb	0	0	-	-	-	-	-	-
	in(Hg)	0	1	-	-	-	-	-	-
	mm(Hg)	1	0	-	-	-	-	-	-
	PSIA	1	1	-	-	-	-	-	-

Units of output - Altimeter modes		7	6	5	4	3	2	1	0
	ft	0	0	-	-	-	-	-	-
	m	0	1	-	-	-	-	-	-
	ft/in(Hg)	1	0	-	-	-	-	-	-
	m/mb	1	1	-	-	-	-	-	-

Number of measurements averaged - All modes		7	6	5	4	3	2	1	0
	1	-	-	-	-	0	0	-	-
	10	-	-	-	-	0	1	-	-
	100	-	-	-	-	1	0	-	-
	1000	-	-	-	-	1	1	-	-

Serial output baud rate - Barometer mode and Altimeter mode		7	6	5	4	3	2	1	0
	9600	-	-	0	0	-	-	-	-
	1200	-	-	0	1	-	-	-	-
	300	-	-	1	0	-	-	-	-
	110	-	-	1	1	-	-	-	-

Serial output baud rate - Altimeter setting mode and Test mode		7	6	5	4	3	2	1	0
	9600	0	0	-	-	-	-	-	-
	1200	0	1	-	-	-	-	-	-
	300	1	0	-	-	-	-	-	-
	110	1	1	-	-	-	-	-	-

Parallel output data format - Bar parallel mode and Alt parallel mode		7	6	5	4	3	2	1	0
	ASCII coded output	-	-	0	0	-	-	-	-
	Binary coded output	-	-	0	1	-	-	-	-

Test mode tests -		7	6	5	4	3	2	1	0
	Raw data/press (mb)	-	-	-	-	0	0	-	-
	Serial ID test	-	-	-	-	0	1	-	-
	SA test	-	-	-	-	1	0	-	-
	Raw data	-	-	-	-	1	1	-	-

Key to table abbreviations -

r = Serial output baud rate
n = Number of measurements to be averaged
b = Format of parallel output - binary/ASCII coded
u = Units of output
* = Don't care - jumpers must be in place however
t = Test type

APPENDIX C. ADDITIONAL TELECOMMUNICATIONS INFORMATION

C.1 TELECOMMUNICATIONS COMMAND WITH BINARY RESPONSES

Command	Description
[nnnnn]**F**	BINARY DUMP - CR23X sends, in Final Storage Format (binary), the number of Final Storage locations specified (nnnnn) from current MPTR locations, then Signature (no prompt). A 40 second time out timer is reset each F command. nnnnn must be ≤ 65535.

DATALOGGER J AND K COMMANDS

3142J and 2413J

The older 3142J command is used to toggle datalogger user flags and ports, request final storage data, and to establish the input locations returned by the K command. The newer 2413J was added to later operating systems and is identical to the 3412J command except that when the datalogger receives the 2413J command, it ignores the bytes that toggle flags and ports. Setting ports and flags in the new operating systems is done with the U command which adds checksums for improved reliability.

A note on datalogger OS versions:
Beginning with Version 1.7 of the CR23X operating system (released October 1999), the datalogger can use either the new 2413J command or the older 3412J command. Since the J command is not sent with a checksum, the older 3142J could, over a noisy communications link, be misinterpreted and cause unintended changes to flags or ports. The advantage of the new version of the J command is that it does not attempt to set flags or ports; instead, software can use the new U command which is sent with a checksum to accomplish setting of flags, ports and input locations. Telecommunications software can determine whether the new J command is available in a particular datalogger by issuing an A command to request the datalogger's status (Section 5, Telecommunications). If the datalogger version

number is reported as V4, the datalogger OS can use the new 2413J, as well as the new S, T and U commands. See Section 5, Telecommunications, for a description of these commands.

The format of the J command is as follows:

$$2413J<CR>abcd…nNULL$$

or

$$3142J<CR>abcd...nNULL$$

where

1) either "2413J<CR>" or "3142J<CR>" is the command.

2) "a" is a 1 byte value representing the user flags to be toggled. The most significant bit (MSB), if set, will toggle datalogger user flag 8. Likewise, the 2nd most significant bit, if set, will toggle user flag 7, and so on to the least significant bit which, if set, toggles user flag 1. Toggle means that if a flag is set, it will be then reset, or if it is reset, it will be set.

3) "b" is a 1 byte value whose MSB will determine whether Final Storage Data is returned after the K command. If the bit is set, Final Storage Data, if any, will be returned after the next K command. The datalogger initially has this bit reset upon entering telecommunications, but once set by a J command, it will remain set until reset by another J command or telecommunications is terminated.

The 2nd MSB set means a port toggle byte will follow and port status is to be returned with the K command. Like the MSB, this bit is reset upon entering telecommunications, but remains set once set until reset by another J command or telecommunications is terminated.

The 3rd MSB set means a flags 11..18 byte will follow the port toggle byte (if present) and flags 11..18 byte is to be returned with the K command. Like the MSB, this bit is reset upon entering telecommunications, but remains set once set until reset by

another J command or telecommunications is terminated.

The 4th MSB indicates if the input locations requested by the J command are 2 byte (B4 set) or 1 byte. In the case of 2 byte locations (most significant byte first), the terminating location is a 2 byte NULL. FF in the most significant byte will still abort the command.

If "b" = FF or 11111111, then the J command aborts.

The remaining bits are reserved.

4) If the 2nd MSB in "b" was set then "c" is a port toggle byte, otherwise "c,d,...,n" are each 1 byte binary values each representing a datalogger input storage location. The data at those locations will be returned after the next K command. ASCII code 1 (0000001 binary) represents input location 1. ASCII codes 2 (00000010 binary) represents input location 2, and so on. The order of the location requests is not important. The list is limited, however, to 62 total location requests.

5) "Null" or ASCII code 0 (00000000 binary) terminates the J command. Alternately, 11111111 binary aborts the J command. If aborted, flags will not be toggled and location requests will not be saved.

For example:

User Enters	Datalogger Echo
2	2
4	4
1	1
3	3
J	J
CR	CR
	LF
	<
a	a
b	b
c	c
d	d
n	n
Null	Null

K The K command returns datalogger time, user flag status, port status if requested, the

data at the input locations requested in the J command, and Final Storage Data if requested by the J command. The format of the command is K<CR> (K Return). The datalogger will echo the K and Return and send a Line Feed. The amount of data that follows depends on the J command previously executed; four time bytes, a user flags byte, four bytes for each input location requested in the J command, Final Storage data in CSI's binary format if requested by the J command, and terminating in 7F 00 HEX and two signature bytes.

User Enters	Datalogger Echo
K	K
CR	CR
	LF
	Time Minutes byte 1
	Time Minutes byte 2
	Time Tenths byte 1
	Time Tenths byte 2
	Flags 1..8 byte
	Ports byte (if requested)
	Flags 11..18 byte (if requested)
	Data1 byte 1
	Data1 byte 2
	Data1 byte 3
	Data1 byte 4
	Data2 byte 1
	Data2 byte 2
	Data2 byte 3
	Data2 byte 4
	DataN byte 1
	DataN byte 2
	DataN byte 3
	DataN byte 4
	Final Storage Data bytes
	01111111 binary byte
	00000000 binary byte
	Signature byte 1
	Signature byte 2

Time Minutes byte 1 is most significant. Convert from binary to decimal. Divide by 60 to get hours, the remainder is minutes. For example, 00000001 01011001 (01 59 HEX) is 345 decimal minutes or 5:45.

Time Tenths byte 1 is most significant. Convert from binary to decimal. Divide by 10 to get seconds and tenths of seconds. For example, 00000001 11000110 (01 C6 HEX) is 454 decimal or 45.4 seconds. Thus the datalogger time for 01 59 01 C6 HEX is 5:45:45.4.

The Flags byte expresses datalogger user flag status. The most significant bit represents Flag 8, and so on to the least significant bit which represents Flag 1. If a bit is set, the user flag is set in the datalogger.

The optional ports byte (currently on return if requested by a CR23X J command) expresses the datalogger port status. The most significant bit represents Port 8, and so on to the least significant bit which represents Port 1.

For each input location requested by the J command, four bytes of data are returned. The bytes are coded in Campbell Scientific, Inc. Floating Point Format. The format is decoded to the following:

$$Sign(Mantissa*2^{(Exponent)})$$

Data byte 1 contains the Sign and the Exponent. The most significant bit represents the Sign; if zero, the Sign is positive, if one, the Sign is negative. The signed exponent is obtained by subtracting 40 HEX (or 64 decimal) from the 7 remaining least significant bits.

Data bytes 2 to 4 are a binary representation of the mantissa with byte 2 the most significant and 4 the least. The mantissa ranges in value from 80 00 00 hex (0.5 decimal) to FF FF FF HEX ($1-2^{-24}$ decimal). The one exception is for zero which is 00 00 00 00 HEX.

The Mantissa is calculated by converting Data bytes 2 to 4 into binary. Each bit represents some fractional value which is summed for all 24 bits. The bits are arranged from MSB to LSB with the most significant as bit_{23} and least significant as bit_0. The value that each bit represents = 2^{n-24}; where n=bit location. For example, if there was a 1 at bit_{20}, it's value would be $2^{(20-24)}$ or 2^{-4}.

A simple method for programming this is as follows:

Set Mantissa = 0.

Set Bit Value = 0.5.

For loop count = 1 to 24 do the following:

> If the MSB is one, then add Bit Value to the Mantissa.

> Shift the 24 bit binary value obtained from Data bytes 2 to 4 one bit to the left.

> Multiply Bit Value by 0.5.

End of loop.

Another method that can be used as an estimate is to convert Data bytes 2 to 4 from a long integer to floating point and dividing this value by 16777216.

As an example of a negative value, the datalogger returns BF 82 0C 49 HEX.

Data byte 1 = BF HEX.

Data byte 2 to 4 = 82 0C 49 HEX (or 8522825 decimal).

Data byte 1 is converted to binary to find the Sign. BF HEX = 10111111 BINARY.

The most significant bit is 1 so the Sign is NEGATIVE.

The exponent is found by subtracting 40 HEX from the remaining least significant bits. Converting the binary to hexadecimal, 111111 BINARY = 3F HEX (or 63 decimal).

3F - 40 HEX = FF FF FF FF FF HEX. Or in decimal: 63 - 64 = -1.

Exponent is -1 decimal.

The binary representation of Data bytes 2 to 4 is: 10000010 00001100 01001001.

Summing all the fractional values: $2^{-1} + 2^{-7} + 2^{-13} + 2^{-14} + 2^{-18} + 2^{-21} + 2^{-24} = 0.50800$.

Using the estimate method to find the Mantissa = 82 0C 49 HEX / 1 00 00 00 HEX (or 8522825 / 16777216) which is 0.50800 decimal.

The value is then $(-)0.508*2^{-1}$ which equals -0.254.

As an example of a positive value, the datalogger returns 44 D9 99 9A HEX.

Data byte 1 = 44 HEX.

Data byte 2 to 4 = D9 99 9A HEX (or 891290 decimal).

Data byte 1 is converted to binary to find the Sign. 44 HEX = 01000100 BINARY.

NOTE: Don't lose the leading zero!

The most significant bit is 0 so the Sign is POSITIVE.

The exponent is found by subtracting 40 HEX from the remaining least significant bits. Converting the binary to hexadecimal, 1000100 BINARY = 44 HEX (or 68 decimal).

44 - 40 HEX = 4 HEX. Or in decimal: 68 - 64 = 4.

Exponent is 4 decimal.

The binary equivalent of Data bytes 2 to 4 is: 11011001 10011001 10011010.

Summing all the fractional values:

$$2^{-1} + 2^{-2} + 2^{-4} + 2^{-5} + 2^{-8} + 2^{-9} + 2^{-12} + 2^{-13} + 2^{-16} + 2^{-17} + 2^{-20} + 2^{-21} + 2^{-23} = 0.85000.$$

Using the estimate method to find the Mantissa = D9 99 9A HEX / 1 00 00 00 HEX (or 14260634 / 16777216) which is 0.85000 decimal.

The value is then $(+)0.85*2^4$ which equals 13.60.

If appropriately requested by a J command, Final Storage data, if any, will immediately follow the input location data. Refer to the datalogger manual for a description of how to decode Final Storage data in CSI's binary data format. Final Storage data will be limited to not more than 1024 bytes per K command.

The K command data is terminated with 7F 00 HEX (a unique binary format code) followed by two signature bytes. Refer to the datalogger manual for the meaning and calculation of the signature bytes. The signature in this case is a function of the first time byte through the 7F 00 HEX bytes. Calculate the signature of the bytes received and compare with the signature received to determine the validity of the transmission.

C.2 FINAL STORAGE FORMAT

CR23X data is formatted as either 2 byte LO Resolution or 4 byte HI Resolution values. The first two bytes of an Output Array contain a unique code (FC Hex) noting the start of the Output Array and the Output Array ID, followed by the 2 or 4 byte data values. At the end of the data sent in response to the telecommunications F command a 2 byte signature is sent (see below).

Representing the bits in the first byte of each two byte pair as ABCD EFGH (A is the most significant bit, MSB), the byte pairs are described here.

**LO RESOLUTION FORMAT - D,E,F,
NOT ALL ONES**

Bits	Description
A	Polarity, 0 = +, 1 = -.
B, C	Decimal locators as defined below.
D-H plus	13 bit binary value (D=MSB).
second	Largest possible number without D, E, and F all 1 is 7167,
byte	but CSI defines the largest allowable range as 6999.

The decimal locators can be viewed as a negative base 10 exponent with decimal locations as follows:

B	C	Decimal Location
0	0	XXXX.
0	1	XXX.X
1	0	XX.XX
1	1	X.XXX

DATA TYPE WHEN D,E,F, ALL EQUAL ONE

If D, E, and F are all ones, the data type is determined by the other bits as shown below. X implies a "don't care" condition; i.e., the bit can be either 1 or 0 and is not used in the decode decision.

A	B	C	D	E	F	G	H	DATA TYPE AND SECOND BYTE FORMAT
1	1	1	1	1	1	X	X	A,B,C, = 1 - Start of Output Array, G & H are the most significant bits of the Output Array ID. All 8 bits of the 2nd byte are also included in the ID.
X	X	0	1	1	1	X	X	C = 0 - First byte of a 4 byte value.
0	0	1	1	1	1	X	X	A,B = 0; C = 1 - Third byte of a 4 byte value.
0	1	1	1	1	1	1	1	A = 0; remaining bits = 1 - First byte of a 2 byte "dummy" word. The CR23X always transmits a 0 for the 2nd byte, but the word can be decoded on the basis of the 1st byte only.

HIGH RESOLUTION FORMAT

Continuing to use the A-H bit representation, the four byte number is shown below as two two byte pairs.

AB0111GH   XXXXXXXX   001111GH   XXXXXXXX

BITS, 1ST BYTE, 1ST PAIR	DESCRIPTION
CDEF = 0111	Code designating 1st byte pair of four byte number.
B	Polarity , 0 = +, 1 = -.
G,H,A,	Decimal locator as defined below.
2nd byte	16th - 9th bit (left to right) of 17 bit binary value.
ABCDEF = 001111	Code designating 2nd byte pair of four byte number.
G	Unused bit.
H	17th and MSB of 17 bit binary value.
2nd byte	8th - 1st bit (left to right) of 17 bit binary value.

CSI defines the largest allowable range of a high resolution number to be 99999.

Interpretation of the decimal locator for a 4 byte data value is given below. The decimal equivalent of bits GH is the negative exponent to the base 10.

BITS	DECIMAL FORMAT
G H A	**5 digits**
0 0 0	XXXXX.
0 0 1	XXXX.X
0 1 0	XXX.XX
0 1 1	XX.XXX
1 0 0	X.XXXX
1 0 1	.XXXXX

C.3 GENERATION OF SIGNATURE

At the end of a binary transmission, a signature is sent. The signature is a 2 byte integer value which is a function of the data and the sequence of data in the Output Array. It is derived with an algorithm that assures a 99.998% probability of detecting a change in the data or its sequence. The CR23X calculates the signature using each transmitted byte beginning with the Final Storage format data (for K command, echo and carriage return line feed are not included) until the 2 byte signature itself. By calculating the signature of the received data and comparing it to the transmitted signature, it can be determined whether the data was received correctly.

SIGNATURE ALGORITHM

- S1,S0 - represent the high and low bytes of the signature, respectively

- M - represents a transmitted data byte

- n - represents the existing byte

- n+1 - represents the new byte

- T - represents a temporary location

- C - represents the carry bit from a shift operation

1. The signature is initialized with both bytes set to hexadecimal AA.

$$S_1(n) = S_0(n) = AA$$

2. When a transmitted byte, M(n+1), is received, form a new high signature byte by setting it equal to the existing low byte. Save the old high byte for later use.

$$T_1 = S_1(n)$$

$$S_1(n+1) = S_0(n)$$

3. Form a temporary byte by shifting the old low signature byte one bit to the left and adding any carry bit which results from the shift operation. A "shift left" is identical to a multiply by 2. Ignore any carry bit resulting from the add.

$$T_2 = \text{shift left } (S_0(n)) + \text{carry}$$

4. Form the new low signature byte by adding the results of operation 3 to the old high signature byte and the transmitted byte. Ignore any carry bits resulting from these add operations.

$$S_0(n+1) = T_2 + S_1(n) + M(n+1)$$

As each new transmitted byte is received, the procedure is repeated.

C.4 [*] [D] COMMANDS TO TRANSFER PROGRAM WITH COMPUTER

[*] [D] Commands 1 and 2 (when entered from the Keyboard/Display) and 7 have an additional 2 digit option parameters (7 is entered with the Storage Module address, e.g., 71). The CR23X

will display the command number and prompt for the option. If the keyboard display is not being used, the CR23X will have already set the baud rate to that of the device it is communicating with and will be ready to send or receive the file as soon as command 1 or 2 is entered.

TABLE C.4-1. [*] [D] Command 1 and 2 Options		
Command		**Option Code Description**
1 & 2	1x 4x	Synchronously addressed Hardware enabled
		x = Baud Rate Codes 0 - 300
		1 - 1200
		2 - 9600
		3 - 76,800 4 - 2400 5 - 4800 6 - 19,200 7 - 38,400

After the option code is keyed in, key "A" to execute the command. Command 2 will be aborted if no data is received within 40 seconds.

WHEN COMMAND 2 IS EXECUTED ALL DATA IN INPUT AND INTERMEDIATE STORAGE ARE ERASED. This section describes commands 1 and 2.

SENDING ASCII PROGRAM INFORMATION

Program listings are sent in ASCII. At the end of the listing, the CR23X sends control E (5 hex or decimal) twice.

Table 1.8-4 is an example of the program listing sent in response to command. Note that the listing uses numbers for each mode: The numbers for [*] [A] , [*] [B] , and [*] [C] modes are 10, 11, and 12, respectively.

**TABLE C.4-2. Example Program Listing
From ⸬ ⎕ Command 1**

```
MODE 1
SCAN RATE 5
1:P17
1:1

2:P86
1:10

3:P70
1:1
2:1

4:P0

MODE 2
SCAN RATE 0

MODE 3
1:P0

MODE 10
1:28
2:64
3:0
4:5332
5:1971

MODE 12
1:0
2:0

MODE 11
1:6597
2:30351
3:48
4:0
5:0
^E ^E
```

LOAD PROGRAM FROM ASCII FILE

Command 2 sets up the CR23X to load a program which is input as serial ASCII data in the same form as sent in response to command 1.

A download file need not follow exactly the same format that is used when listing a program (i.e., some of the characters sent in the listing are not really used when a program is loaded). Some rules which must be followed are:

1. "**M**" must be the first character other than a carriage return, line feed, semicolon, or 7D Hex. The "**M**" serves the same function as "∗" does from the keyboard. The order in which the Modes are sent in does not matter (i.e., the information for Mode 3 could be sent before that for Mode 1).

2. "**S**" is necessary prior to the **S**can Rate (execution interval).

3. The colons (**:**) are used to mark the start of actual data.

4. A semicolon (**;**) tells the CR23X to ignore the rest of the line and can be used after an entry so that a comment can be added.

There are 4 two-character control codes which may be used to verify that the CR23X receives a file correctly:

^B ^B   (2hex, 2hex)--Discard current buffer and reset signature

^C ^C   (3hex, 3hex)--Send signature for current buffer

^D ^D   (4hex, 4hex)--Load current buffer and reset signature

^E ^E   (5hex, 5hex)--Load current buffer, Exit and compile program

As a download file is received, the CR23X buffers the data in memory; the data is not loaded into the editor or compiled until the CR23X receives a command to do so. The maximum size of the buffer is 1.5K. The minimum file that could be sent is the program listing, then ^E ^E. ^C ^C tells the CR23X to send the signature (Appendix C.3) for the current buffer of data. If this signature does not match that calculated by the sending device, ^B ^B can be sent to discard the current buffer and reset the signature. If the signature is correct, ^D ^D can be sent to tell the CR23X to load the buffer into the editor and reset the signature. Once the complete file has been sent and verified, send ^E ^E to compile the program and exit the load command.

APPENDIX E. ASCII TABLE

American Standard Code for Information Interchange
Decimal Values and Characters

(X3.4-1968)

Dec.	Char.	Dec.	Char.	Dec.	Char.	Dec.	Char.
0	CONTROL @	32	SPACE	64	@	96	`
1	CONTROL A	33	!	65	A	97	a
2	CONTROL B	34	"	66	B	98	b
3	CONTROL C	35	#	67	C	99	c
4	CONTROL D	36	$	68	D	100	d
5	CONTROL E	37	%	69	E	101	e
6	CONTROL F	38	&	70	F	102	f
7	CONTROL G	39	'	71	G	103	g
8	CONTROL H	40	(	72	H	104	h
9	CONTROL I	41	)	73	I	105	i
10	CONTROL J	42	*	74	J	106	j
11	CONTROL K	43	+	75	K	107	k
12	CONTROL L	44	,	76	L	108	l
13	CONTROL M	45	-	77	M	109	m
14	CONTROL N	46	.	78	N	110	n
15	CONTROL O	47	/	79	O	111	o
16	CONTROL P	48	0	80	P	112	p
17	CONTROL Q	49	1	81	Q	113	q
18	CONTROL R	50	2	82	R	114	r
19	CONTROL S	51	3	83	S	115	s
20	CONTROL T	52	4	84	T	116	t
21	CONTROL U	53	5	85	U	117	u
22	CONTROL V	54	6	86	V	118	v
23	CONTROL W	55	7	87	W	119	w
24	CONTROL X	56	8	88	X	120	x
25	CONTROL Y	57	9	89	Y	121	y
26	CONTROL Z	58	:	90	Z	122	z
27	CONTROL [	59	;	91	[	123	{
28	CONTROL \	60	<	92	\	124		
29	CONTROL]	61	=	93	]	125	}
30	CONTROL ^	62	>	94	^	126	~
31	CONTROL _	63	?	95	_	127	DEL

APPENDIX F. DYNAGAGE SAP-FLOW (P67)

F1. FUNCTION

The Dynagage Processing Instruction, P67, is used in conjunction with a Dynamax, Inc. Dynagage stem flow gage. Instruction 67 calculates the sap flow rate within a plant stem by using an energy balance equation and inputs from a Dynagage sensor.

Briefly, the energy input into the stem is a known quantity (Q_h), conductive radial and vertical energy losses are calculated (Q_r and Q_v, respectively). The residual energy is assumed lost through sap flow (Q_f) and is given as:

$$Q_f = Q_h - (Q_r + Q_v).$$

By measuring the change in temperature (dT) of the stem by heating and assuming 4.186 Joules of energy are required to heat each gram of sap $1°$ C, the flow rate can be calculated.

Accumulated flow can be calculated using datalogger Instruction 72 (Totalize).

F2. INSTRUCTION DETAILS

Dynagage Sap-Flow (P67)

1:	0000	Begin Input Location [_____]
2:	0.0	K_{sh} (W/mV)
3:	0.0	Heater Resistance(Ohm)
4:	0.0	Stem Area (sq cm)
5:	0.0	Thermal Conductivity (W/m∗K)
6:	0.0	Thermocouple Gap (cm)
7:	0.0	Low-Flow Cutoff Temperature (C)
8:	0.0	High-Flow Cutoff Velocity (cm/s)
9:	00	Output
10:	0000	Destination Input Location [_____]
11:	0.0	Mult
12:	0.0	Offset

F2.1 INPUT VOLTAGES

PARAMETER 1

Four voltages must be stored sequentially beginning with the Input Loc option stated in Parameter 1.

INPUT LOCATION
Loc 1: Ch (Thermopile (mV))
Loc 2: Ah (Upper TC (mV))
Loc 3: Bh (Lower TC (mV))
Loc 4: Voltage Input

These voltages are measured using Instruction 2. Voltages for locations 1-3 are measured on the 2.5 mV slow range. The voltage for location 4 is measured on the 2500 mV slow range.

F2.2 CONSTANTS

Parameters 2 through 6 are constants that must be entered for each Dynagage.

PARAMETER 2
K_{sh}(W/mV), a constant required to relate the thermopile output proportionally to the radial heat transfer.

PARAMETER 3
The heater resistance (Ohms), measured with an Ohm meter.

PARAMETER 4
The cross sectional area of the stem (sq cm).

PARAMETER 5
Thermal conductivity (W/m∗K). This number is generally 0.54 W/m∗K for herbaceous plants or 0.42 W/m∗K for woody plants.

PARAMETER 6
Thermocouple gap (cm) is the distance between the thermocouples in the Dynagage, measured in cm. This value is determined by Dynamax and is found in the Dynagage documentation.

F2.3 FILTERS

Two predictable conditions may occur when the output from the Dynagage is not valid. In these cases, the output must be "filtered" so only reasonable flow rates are processed. These conditions are very low flow rates and high flow rates, both of which cause dT to approach zero. The values generally placed in Parameters 7

and 8 are 0.5 °C and 0.042 cm s^{-1}, respectively. These values should be used unless conditions determined by the user indicate otherwise.

PARAMETER 7
Low-flow filter (0.5°C). This filter sets the reported flow rate (F) to zero if Q_f is greater than or equal to 0 and less than 20% of Q_h, and if dT is less than 0.5°C.

When there is a zero flow rate in a very small stem, dT approaches zero. In this situation the flow rate (F) will be highly exaggerated if a minor residual Q_f exists. To avoid this, F is set to zero. Negative flow rates, resulting from negative dT readings, are also filtered out. This may occur in small stems at night.

The reported flow rate is set to -0.036 g h^{-1}. This alerts the user to the possible need of adjusting K_{sh}. K_{sh} may not be perfectly set, and it is possible that a large negative flow will occur, which could adversely affect the flow accumulator. The small negative number (-0.036) will not adversely affect the flow accumulator.

PARAMETER 8
High Flow Filter (0.042 cm s^{-1}). This filter sets F equal to a calculated theoretical maximum flow (F_{max}) if F is greater than F_{max}. F_{max} equals the theoretical maximum velocity (V_{max}) multiplied by the cross sectional area of the stem ($F_{max} = V_{max} * A$).

This filter is used to protect the integrity of the flow accumulator when the gage capacity to measure F has been exceeded. When F is exceptionally high, nearly all of the heat is absorbed by the sap. As F continues to increase, dT approaches zero asymptotically. As dT becomes infinitesimal, previously insignificant thermal noise from radiation or other sources can cause a major exaggeration of F. The value 0.042 cm s^{-1} for V_{max} was determined by Dynamax to be a typical maximum. At the users option, V_{max} (Parameter 8) can be increased or decreased when the species under study is verified to be accurately measured.

F2.4 INSTRUCTION OUTPUT

PARAMETER 9
Two output options are available.

Option 0	Option 1
Sapflow (g h^{-1})	Sapflow (g h^{-1})
K_{shapp} (W mV^{-1})	K_{shapp} (W mV^{-1})
	dT (°C)
	power input (W)
	Q_v (Vertical Flux, W)
	Q_r (Radial Flux, W)
	Q_f (Sapflow Flux, W)

F2.4.1 Sapflow (g h^{-1})

The calculated sap flow rate. If total sapflow is desired, Output Processing Instruction 72 can be used in conjunction with Instruction 67 to totalize sap flow.

F2.4.2 Q_f and dT

Q_f is the residual energy that is assumed lost through sap flow and is defined as:

$$Q_f = Q_h - (Q_r + Q_v)$$

The temperature difference, dT, is defined as:

$$dT = \text{Upper TC - Lower TC}$$

where the Upper TC is the thermocouple above the heater and the Lower TC is the thermocouple below the heater.

F2.4.3 Q_r, K_{sh}, K_{shapp}

Q_r, radial energy loss, is the loss of energy through the cork and foam sheath on the sensor. K_{sh}, the thermal conductivity constant, is used to calculate Q_r. Since Q_r can be a large percentage of the total energy loss, K_{sh} must be determined accurately as outlined in the Dynagage manual. The essence of the procedure is to record K_{sh} (the second output of Instruction 67, K_{shapp}) when there is no stem flow such as at dawn. This value is then placed in Instruction 67, Parameter 2, K_{sh} (W mV^{-1}). K_{shapp} and K_{sh} (W mV^{-1}) have no effect on each other in Instruction 67. K_{sh} must be determined for each installation.

F2.4.4 Power Input (W)

Power input or Q_h is calculated from the input voltage and heater resistance: $Q_h = V/R$.

F2.4.5 Q_V (W)

Qv, or vertical energy loss, is the loss of energy through the wood at the ends of the sensor not associated with the heating of the sap.

Appendix A

Instruction P67 Processing

Loc = input location assuming input locations 1, 2, 3, and 4 are used.

Par = Instruction P67 parameter.

P_{in} = (Loc 4)*(Loc 4)/(Par 3)

Q_v = (((Loc 3)-(Loc 2))/(4.0*(Par 6)))*(Par 4)*(Par 5)

Q_r = (Loc 1)*(Par 2)

Q_f = P_{in} - Q_v - Q_r

K_{shapp} = (P_{in} - Q_v)/(Loc 1)

dT = ((Loc 2+Loc 3)/2.0)*25.0

Sapflow = Q_f/(dT*4.186)

If Par 7 μ 0.0 then go to XXXXXX

If Q_f < (0.2*P_{in}) and If dT < Par 7, then Sapflow = 0.0

If Q_f < (0.2*P_{in}) and If Q_f < 0.0, then Sapflow = - 0.00001

XXXXXX

If Par 8 μ 0.0 then go to YYYYYY

F_{max} = (Par 8)*(Par 4)

If Sapflow > F_{max}, the Sapflow = F_{max}

YYYYYY

Sapflow = (Sapflow*(Par 11))+(Par 12)

Transfer to Input Locations beginning with Parameter 10, either short output or long output, depending on Parameter 9.

APPENDIX F. DYNAGAGE SAP-FLOW (P67)

APPENDIX G. CALLBACK (CR23X INITIATED TELECOMMUNICATIONS)

Callback is a joint feature of the CR23X and PC208W Datalogger Support Software. In common datalogging applications, the PC calls the CR23X; with Callback, the CR23X is enabled to call the PC. Callback is usually used to immediately notify operators via PC that an alarm condition exists. This appendix also covers using the CR23Xs portion of Callback (Instruction P97) to initiate calls using a voice synthesizer modem or calls to a pager service.

G.1 INTRODUCTION

Successful use of Callback requires precision placement of hardware and software elements. Success is usually achieved by first becoming proficient with the simplest case, then incrementally adding levels of complexity. If possible, become familiar with Callback over a simple RS232 connection (using the SC32A Interface or SC929 Cable) before incorporating telephone or radio options.

G.2 DEVELOPING A CALLBACK APPLICATION

Suggested procedure:

1) Establish telecommunications with the CR23X with the PC208W Datalogger Support Software. For help in doing this,

consult the PC208W manual, PC208W on-screen help, and the CR23X manual. Limited free telephone support is also available from Campbell Scientific applications engineers.

2) Modify the PC208W software to answer a call. Three simple modifications are required:

a) As shown in Figure G.2-1 go to *PC208W | Setup | COM(Port#) | Hardware* and check the box labeled "Allow Call-back On this Port." Also, set the Callback baud rate equal to that used in step 1 above. If in doubt, use a baud rate of 1200. Setting the port to allow callback may cause PC208W storage module support to malfunction.
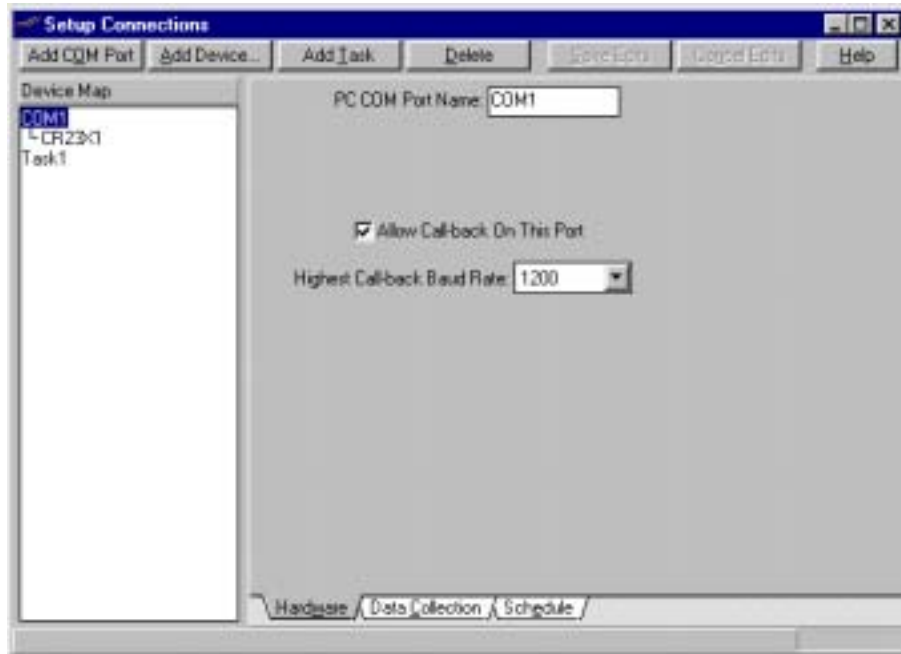
FIGURE G.2-1. Configuring COM Port for Callback

b) As shown in Figure G.2-2 go to *PC208W | Setup | (StationName) | Hardware* and enter a 3 digit identification code. The number you choose should be unique to the station. Make a note of the baud rate and 3 digit code since these values will be used in programming the CR23X.
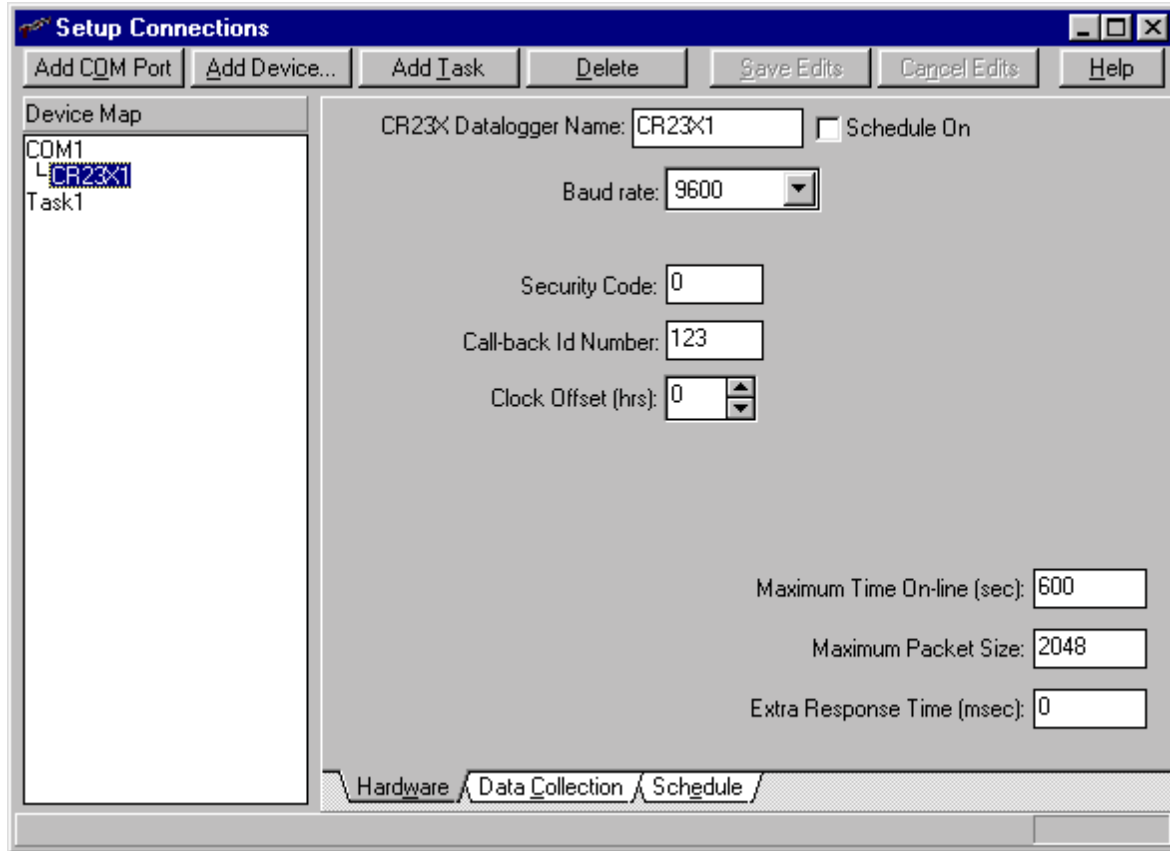
FIGURE G.2-2. Entering Callback Identification Code
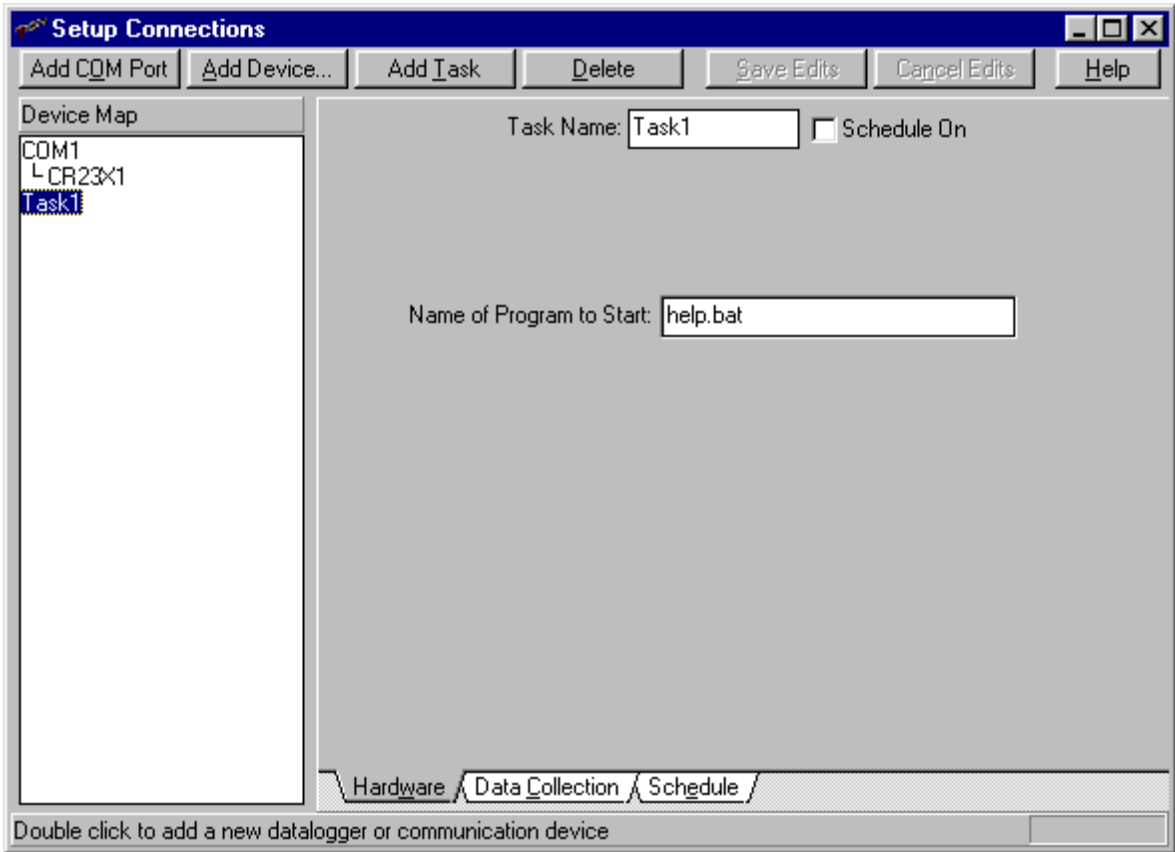
FIGURE G.2-3. Configuring a Task

c) As shown in Figure G.2-3, go to *PC208W | Setup* and add Task1 to the device map. Also go *to PC208W | Setup | Task1 | Hardware* and enter the name of the executable file that will indicate an alarm.

3) Program the CR23X to make the call. Section G.3 addresses CR23X programming.

G.3 CR23X PROGRAMMING

The CR23X is programmed for Callback by the use of Instruction P97 Initiate Telecommunications. A detailed description of each P97 parameter is given in Section 12. Initially, set parameter 3 to 60 seconds; set parameters 4, 5, and 6 to 0. Doing otherwise increases the risk of large long distance charges, dead datalogger batteries, and phone calls in the middle of the night. After initial success, parameters 3 through 6 can be adjusted to refine the application.

In most examples below, the program measures a 107 temperature probe once per 10 second interval and stores an average temperature in final storage once per 1 minute interval. If the air temperature exceeds 32°C, the computer is called, data saved to disk, and a batch file executed to show a visual alarm (batch file is not supplied by Campbell Scientific).

G.3.1 RS232 APPLICATION

Following is an example program for use with a CR23X connected to a computer via an SC32A Optically Isolated RS232 Interface or an SC929 Interface Cable.

APPENDIX G. CALLBACK (CR23X INITIATED TELECOMMUNICATIONS)

```
*Table 1 Program
  01:          10            Execution Interval (seconds)

1: Panel Temperature (P17)
  1:           1             Loc [ RefTemp   ]

2: Thermocouple Temp (DIFF) (P14)
  1:           1             Reps
  2:          11             10 mV, Fast Range
  3:           5             DIFF Channel
  4:           1             Type T (Copper-Constantan)
  5:           1             Ref Temp (Deg. C) Loc [ RefTemp   ]
  6:           2             Loc [ TempDegC  ]
  7:          1.0            Mult
  8:           0             Offset

3: If time is (P92)
  1:           0             Minutes (Seconds --) into a
  2:           1             Interval (same units as above)
  3:          10             Set Output Flag High (Flag 0)

4: Real Time (P77)
  1:        1220             Year,Day,Hour/Minute (midnight = 2400)

5: Average (P71)
  1:           1             Reps
  2:           2             Loc [ TempDegC  ]

6: If (X<=>F) (P89)
  1:           2             X Loc [ TempDegC  ]
  2:           4             <
  3:          31             F
  4:          11             Set Flag 1 High

7: If (X<=>F) (P89)
  1:           2             X Loc [ TempDegC  ]
  2:           3             >=
  3:          32             F
  4:          21             Set Flag 1 Low

8: Initiate Telecommunications (P97)
  1:          11             Direct CS IO/1200 Baud<<<<<<<<<<<Note Device / Baud Code
  2:           1             Disabled when User Flag 1 is High
  3:         120             Seconds Call Time Limit
  4:           0             Seconds Before Fast Retry
  5:           0             Fast Retries
  6:           0             Minutes before Slow Retry
  7:           3             Failures Loc [ Failures  ]
  8:         123             Call-back ID
```

G-4

G.3.2 TELEPHONE / CELLULAR TELEPHONE APPLICATION

Following is an example program for use with a CR23X connected to a computer via a COM200 Datalogger Telephone Modem, a COM300 Datalogger Telephone Modem in data mode, or COM100 Cellular Transceiver. A modification may be needed to the modem string in the PC208W Setup screen. If the string contains an &Q setting, it must be set to &Q0 for 1200 baud (&Q5 for 9600 baud).

```
*Table 1 Program
  01:        10        Execution Interval (seconds)

1: Panel Temperature (P17)
  1:         1         Loc [ RefTemp   ]

2: Thermocouple Temp (DIFF) (P14)
  1:         1         Reps
  2:         11        10 mV, Fast Range
  3:         5         DIFF Channel
  4:         1         Type T (Copper-Constantan)
  5:         1         Ref Temp (Deg. C) Loc [ RefTemp   ]
  6:         2         Loc [ TempDegC  ]
  7:         1.0       Mult
  8:         0         Offset

3: If time is (P92)
  1:         0         Minutes (Seconds --) into a
  2:         1         Interval (same units as above)
  3:         10        Set Output Flag High (Flag 0)

4: Real Time (P77)
  1:         1220      Year,Day,Hour/Minute (midnight = 2400)

5: Average (P71)
  1:         1         Reps
  2:         2         Loc [ TempDegC  ]

6: If (X<=>F) (P89)
  1:         2         X Loc [ TempDegC  ]
  2:         4         <
  3:         31        F
  4:         11        Set Flag 1 High

7: If (X<=>F) (P89)
  1:         2         X Loc [ TempDegC  ]
  2:         3         >=
  3:         32        F
  4:         21        Set Flag 1 Low

;*********Cell Transceiver Control Segment***********
;Include this segment only if a cell transceiver is
;used and controlled by the CR23X.  Segment enables
;transceiver for 10 minutes, at the top of each hour,
;during routine operation, but turns cell phone on
;immediately if an alarm condition occurs.
```

```
8: If Flag/Port (P91) ;Enable Cell Transceiver if No Alarm
   1:         11          Do if Flag 1 is High
   2:         30          Then Do

9: If time is (P92)
   1:          0          Minutes (Seconds --) into a
   2:         60          Interval (same units as above)
   3:         41          Set Port 1 High

10: If time is (P92)
   1:         10          Minutes (Seconds --) into a
   2:         60          Interval (same units as above)
   3:         51          Set Port 1 Low

11: End (P95)

12: If Flag/Port (P91) ;Enable Cell Transceiver if Alarm
   1:         21          Do if Flag 1 is Low
   2:         30          Then Do

13: Do (P86)
   1:         41          Set Port 1 High

14: End (P95)

;****************************************************

15: Initiate Telecommunications (P97)
   1:         21          Phone Modem/1200 Baud<<<<<<<<<<<Note Device / Baud Code
   2:          1          Disabled when User Flag 1 is High
   3:        120          Seconds Call Time Limit
   4:          0          Seconds Before Fast Retry
   5:          0          Fast Retries
   6:          0          Minutes before Slow Retry
   7:          3          Failures Loc [ Failures  ]
   8:        123          Call-back ID

16: Extended Parameters 4 Digit (P68)
   1:          1          Option                  ; First Digit of Computer's Phone Number
   2:          4          Option
   3:          3          Option
   4:          5          Option
   5:          1          Option
   6:          2          Option
   7:          3          Option
   8:          4          Option

17: Extended Parameters 4 Digit (P68)
   1:          5          Option
   2:          6          Option
   3:          7          Option
   4:         13          Option                  ; ASCII Decimal Termination Character
   5:          0          Option
   6:          0          Option
   7:          0          Option
   8:          0          Option
```

G.3.4 PAGER APPLICATION

Following is an example program that causes a CR23X to call a pager in response to an alarm condition. No data are transferred.

The first program segment (instructions 1 through 5) presets the P97 interrupt disable flag on the first program execution only.

Input location [3] is used as a master control register. If Flag 4 is low, then input location [3] increments by 1. If input location [3] is <> 0, no further callbacks will be attempted. This way the

number of attempts / pages from one alarm event can be controlled. Location [3] must manually be cleared to reset the pager alarm.

CR23X to pager communications are strictly one-way, so the CR23X will not retry based on the success or failure of the page. Commas create a delay between the number being dialed and page numbers being transmitted. Each comma causes a 2-second delay. Specific applications require the number of retries and the delay to be determined by trial and error. Four or five successful tests before deployment are recommended.

```
*Table 1 Program
  01:           10.0          Execution Interval (seconds)
;
;****************************************
;
;Interrupt disable flag is set high to prevent callback.
;Error location cleared.
;Flag transition states are preset.
;****************************************
;
;
1:  If Flag/Port (P91)
  1:           21            Do if Flag 1 is Low
  2:           30            Then Do

2:  Do (P86)
  1:           11            Set Flag 1 High

3:  Do (P86)
  1:           14            Set Flag 4 High

4:  Z=F (P30)
  1:           0             F
  2:           0             Exponent of 10
  3:           4             Z Loc [ ErrorLoc  ]

5:  Z=F (P30)
  1:           1             F
  2:           0             Exponent of 10
  3:           5             Z Loc [ new_flag  ]

6:  Z=F (P30)
  1:           1             F
  2:           0             Exponent of 10
  3:           6             Z Loc [ old_flag  ]

7:  End (P95)

8:  Panel Temperature (P17)
  1:           1             Loc [ RefTemp   ]
```

```
9:  Thermocouple Temp (DIFF) (P14)
    1:          1           Reps
    2:          11          10 mV, Fast Range
    3:          5           DIFF Channel
    4:          1           Type T (Copper-Constantan)
    5:          1           Ref Temp (Deg. C) Loc [ RefTemp   ]
    6:          2           Loc [ TempDegC  ]
    7:          1.0         Mult
    8:          0           Offset
```

```
;************************************************
;
;If the error location has been incremented to 1 (CR23X
;has called pager twice), set Flag 4 high to disable page.
;************************************************
;
```

```
10:  IF (X<=>F) (P89)
    1:          4           X Loc [ ErrorLoc  ]
    2:          1            =
    3:          1           F
    4:          14          Set Flag 4 High
```

```
11:  If Flag/Port (P91)
    1:          14          Do if Flag 4 is High
    2:          30          Then Do
```

```
12:  Z=F (P30)
    1:          0           F
    2:          0           Exponent of 10
    3:          4           Z Loc [ ErrorLoc  ]
```

```
13:  Z=F (P30)
    1:          1           F
    2:          0           Exponent of 10
    3:          5           Z Loc [ new_flag  ]
```

```
14:  End (P95)
```

```
;*******************************************
;
;Flag 8 is used as a page system override. If flag 8
;is high, the CR23X will not call the pager. This allows
;for the use of a timer to prevent call out after the
; initial attempt is made for a specified length of time.
;*******************************************
;
;
;*******************************************
;
;Old_flag / new_flag are used to catch the transition
;from a call out condition to a reset the flag condition.
;*******************************************
;
```

```
15:  IF (X<=>F) (P89)
    1:          6           X Loc [ old_flag  ]
    2:          1           =
    3:          0           F
    4:          30          Then Do
```

```
16: IF (X<=>F) (P89)
  1:           5        X Loc [ new_flag  ]
  2:           1        =
  3:           1        F
  4:          30        Then Do

17: Z=F (P30)
  1:           1        F
  2:           0        Exponent of 10
  3:           8        Z Loc [ cntr_set  ]

18: Do (P86)
  1:          18        Set Flag 8 High

19: End (P95)

20: End (P95)

21: If Flag/Port (P91)
  1:          28        Do if Flag 8 is Low
  2:          30        Then Do

22: IF (X<=>F) (P89)
  1:           2        X Loc [ TempDegC  ]
  2:           3        >=
  3:          32        F
  4:          30        Then Do

23: Do (P86)
  1:          24        Set Flag 4 Low

24: Z=F (P30)
  1:           0        F
  2:           0        Exponent of 10
  3:           5        Z Loc [ new_flag  ]

25: End (P95)

26: Else (P94)

27: Do (P86)
  1:          14        Set Flag 4 High

28: End (P95)
```

```
.***************************************
;
;Call counter disables the call out for 4 hours.
;This is based on a 10 second program scan rate.
.***************************************
;

29: IF (X<=>F) (P89)
  1:           9        X Loc [ CallCntr  ]
  2:           1        =
  3:        1440        F
  4:          30        Then Do
```

```
30: Z=F (P30)
   1:          0        F
   2:          0        Exponent of 10
   3:          8        Z Loc [ cntr_set ]

31: Z=F (P30)
   1:          0        F
   2:          0        Exponent of 10
   3:          9        Z Loc [ CallCntr ]

32: Do (P86)
   1:         28        Set Flag 8 Low

33: End (P95)

34: IF (X<=>F) (P89)
   1:          8        X Loc [ cntr_set ]
   2:          1        =
   3:          1        F
   4:         30        Then Do

35: Z=Z+1 (P32)
   1:          9        Z Loc [ CallCntr ]

36: End (P95)

37: Initiate Telecommunications (P97)
   1:         21        Phone Modem/1200 Baud
   2:          4        Disabled when User Flag 4 is High
   3:         30        Seconds Call Time Limit
   4:          0        Seconds Before Fast Retry
   5:          0        Fast Retries
   6:         15        Minutes before Slow Retry
   7:          4        Failures Loc [ ErrorLoc ]
   8:          0        Data Logger ID

38: Extended Parameters (P63)
   1:          1        Option        ; ASCII digit, first digit in pager number
   2:          8        Option
   3:          0        Option
   4:          1        Option
   5:          7        Option
   6:          5        Option
   7:          0        Option
   8:          9        Option

39: Extended Parameters (P63)
   1:          5        Option
   2:          3        Option
   3:          4        Option        ; Last digit in pager number
   4:         44        Option        ; ASCII Decimal Equivalent Comma (2 second
                                        delay)
   5:         44        Option
   6:         44        Option
   7:          9        Option        ; ASCII digit, first digit in pager message
   8:          8        Option
```

```
40: Extended Parameters (P63)
   1:          7          Option
   2:          3          Option
   3:          4          Option
   4:         13          Option                    ; Termination Character
   5:         00          Option
   6:         00          Option
   7:         00          Option
   8:         00          Option

41: Z=X (P31)
   1:          5          X Loc [ new_flag  ]
   2:          6          Z Loc [ old_flag  ]
```

G.3.5 GENERIC MODEM (CR23X OS 1.9 OR HIGHER)

Following is an example program for use with a CR23X connected to a computer via a non-standard modem link. Although other P97 modem options have pre-configured telecommunications protocols, the generic modem option requires that all transmit, receive, and delay requirements be explicitly programmed into the P68 instructions following P97 in the CR23X program. The example below requires the CR23X to wait 2 seconds after an alarm is triggered, then transmit an "RS" to the receiving modem. The "RS" must be echoed by the receiving modem. After the "RS" echo is received, the CR23X must transmit an "A," which must be echoed, then a "B," which must be echoed, and then a "C" which must be echoed. The CR23X then transmits the station identifying code, 123<CR>, which must be echoed. Upon receiving the echoed 123<CR>, the interrupt disable flag is set high and data retrieval can begin.

```
1: Panel Temperature (P17)
   1:          2          Loc [ RefTemp   ]

2: Thermocouple Temp (DIFF) (P14)
   1:          1          Reps
   2:         11          10 mV, Fast Range
   3:          5          DIFF Channel
   4:          1          Type T (Copper-Constantan)
   5:          2          Ref Temp (Deg. C) Loc [ RefTemp   ]
   6:          3          Loc [ TempDegC  ]
   7:        1.0          Mult
   8:          0          Offset

3: If time is (P92)
   1:          0          Minutes (Seconds --) into a
   2:          1          Interval (same units as above)
   3:         10          Set Output Flag High (Flag 0)

4: Real Time (P77)
   1:       1220          Year,Day,Hour/Minute (midnight = 2400)

5: Average (P71)
   1:          1          Reps
   2:          3          Loc [ TempDegC  ]
```

```
6:  If (X<=>F) (P89)
    1:          3        X Loc [ TempDegC ]
    2:          4        <
    3:         31        F
    4:         11        Set Flag 1 High

7:  If (X<=>F) (P89)
    1:          3        X Loc [ TempDegC ]
    2:          3        >=
    3:         32        F
    4:         21        Set Flag 1 Low

8:  Initiate Telecommunications (P97)
    1:         71        Generic Modem CS IO/1200 Baud
    2:          1        Disabled when User Flag 1 is High
    3:        120        Seconds Call Time Limit
    4:          0        Seconds Before Fast Retry
    5:          0        Fast Retries
    6:          0        Minutes before Slow Retry
    7:          1        Failures Loc [ Failures  ]
    8:        123        Call-back ID

9:  Extended Parameters 4 Digit (P68)
    1:         87        Option ; W |Wait...
    2:         20        Option ; 20 |...2 Seconds
    3:         84        Option ; T |TXMT following...
    4:          2        Option ; 2 |...2 bytes
    5:         82        Option ; R |byte #1
    6:         83        Option ; S |byte #2
    7:         82        Option ; R |RCV following...
    8:          2        Option ; 2 |...2 bytes

10: Extended Parameters 4 Digit (P68)
    1:         82        Option ; R |byte #1
    2:         83        Option ; S |byte #2
    3:         69        Option ; E |TXMT/wait for echo...
    4:          3        Option ; 3 |...of following 3 bytes
    5:         65        Option ; A |byte #1
    6:         66        Option ; B |byte #2
    7:         67        Option ; C |byte #3
    8:         13        Option ;   |Termination
```

APPENDIX H
CALL ANOTHER DATALOGGER VIA PHONE OR RF

H.1 INTRODUCTION

Instructions 97, Initiate Telecommunications, and 63, Extended Parameters can be used to call another datalogger and collect data in input locations. This function can only be accomplished via phone or radio modems.

H.2 PROGRAMMING

Instruction 97 initiates the call and Instruction 63 specifies the dialing path and special options. More than one Instruction 63 may be required. The calling datalogger's program requires Instructions 97 and 63 for each datalogger it calls. Only one datalogger can be called with each set of instructions.

When the calling datalogger executes Instruction 97, it establishes communication and then toggles a flag in the remote datalogger. After a specified delay, the calling datalogger monitors that same flag (in the remote datalogger). When the calling datalogger detects the flag is low, it collects the specified input locations. The collected values are stored in the calling datalogger's input locations beginning at the location after the "failure location" (Parameter 7 of Instruction 97).

In the example below, the remote datalogger is programmed to make measurements when its flag 1 is high and then set flag 1 low. The calling datalogger toggles flag 1 and then waits until flag 1 is low to collect the data, insuring that the input location values are transferred after the desired measurements and processing are complete.

Section H.2 covers programming requirements for the calling CR23X. Section H.3 shows the programming requirements for the remote datalogger.

H.3 PROGRAMMING FOR THE CALLING CR23X

Programming Example 2.1 uses a COM200 phone modem while Programming Example 2.2 uses a RF modem.

H.3.1 INSTRUCTION 97

Many of the parameters in Instruction 97 don't apply when being used to call another datalogger. See Section 12 for more detailed information on Instruction 97's parameters.

PAR. NO.	DATA TYPE	DESCRIPTION
01:	2	Modem/Baud Option
02:	2	Flag No. to Disable Instruction 97
03:	4	Seconds Call Time Limit
04:	4	Seconds Before Fast Retry
05:	2	Fast Retries
06:	4	Minutes Before Slow Retry
07:	4	Failures Loc :
08:	4	Datalogger ID

Parameter 1
This specifies the first modem type to call out through and the appropriate baud rate. See Table H.3-1 for valid options. An RF modem is in the SDC state when the 9th dip switch is closed.

TABLE H.3-1. Option Code for Modem Type and Baud Rate

Modem	Baud Rate		
	300	1200	9600
RF, ME State	00	01	02
Phone (DC112/COM200)	20	21	22
Voice Data	40	41	42
RF, SDC State	50	51	52

Parameter 2
This specifies the flag lowered in the calling datalogger to initiate the call to the other datalogger. This should not be confused with the flag that the calling datalogger sets in the remote datalogger (specified in Instruction 63).

Parameter 3
This is the total length of time the calling datalogger will try to collect input locations from the remote datalogger. The timer begins when a call is initiated and includes the dialing time. This timer must be long enough for the remote datalogger to detect that a flag has been set (i.e., long enough to allow for the execution

APPENDIX H. CALL ANOTHER DATALOGGER VIA PHONE OR RF

interval of the remote datalogger), make the appropriate measurements, lower the flag, and allow for the input location to be transferred.

Parameters 4, 5, and 8
These parameters don't apply when calling a datalogger. Leave these options as 0.

Parameter 6
Normally this option is not used and should be left as 0.

Standard datalogger to datalogger calls have a delay of 2 seconds for the calling datalogger to wait for all the input locations to be returned from the remote datalogger. For long links such as RF to RF through multiple repeaters, this delay may not be sufficient. Parameter 6 can be used as an extra user specified delay in units of 0.01 seconds.

Parameter 7
This parameter specifies the location to store the number of times the call fails. The collected values are stored in the calling datalogger's input locations beginning at the location after the Failure Location.

H.3.2 INSTRUCTION 63

Instruction 63 is used to specify the dialing path and special options for Instruction 97. More than one Instruction 63 may be required. Instruction 68 may be used instead of instruction 63 when a 3 or 4 digit parameter needs to be entered instead of the 2 digit parameter allowed by Instruction 63.

The first set of the parameters are the phone number or RF path. Each digit of the phone number or an RF ID# goes into a separate parameter.

Separate each RF ID# by a 32 (Space). After the last RF ID#, enter a 70 ("F").

If the phone modem at the remote site is a VS1, after the last digit of the phone number you need several 44s (",") then a 42 ("∗"), and then a 9. Enough commas are needed so the VS1 has enough time to pickup the phone line and to start talking. Normally 2 or 3 commas is sufficient. These characters are required to tell the remote VS1 to go into the data/computer mode instead of voice.

After the last digit of the phone number (including any voice codes) or RF path (after the 70), enter a 68 ("D") as one parameter. This

tells the calling datalogger that it is calling another datalogger.

The next two parameters indicate the number of locations to retrieve and the beginning input location to collect respectively. You can not retrieve from input locations 255 or higher. You are limited on the number of locations you may collect per call.

The next parameter is used to specify the flag to toggle and monitor in the remote datalogger.

The second to last parameter is the delay (in units of 0.1 seconds) the calling datalogger should wait before checking to see if the flag has been reset. Once the calling datalogger determines the flag has been reset, it collects the data.

The last parameter needs to be a 13 to terminate the call.

Programming Example 2.1: Calling CR23X Using Phone Modem

Program: This program fragment calls a datalogger at phone number '"539" every 2 minutes. The CR23X toggles Flag 1 in the remote datalogger to trigger measuring and data collection. It collects 3 input locations beginning at the remote datalogger's location 1. These values are stored in locations 2, 3, and 4.

Flag Usage: Flag 2 is used to control Instruction 97 (when to make the call). The program need only set the flag low; Instruction 97 will set it high after a successful transfer.

*Table 1 Program
  01:      1.0   Execution Interval (seconds)

1: If time is (P92)
  1:      0   Minutes (Seconds --) into a
  2:      2   Interval (same units as above)
  3:      22   Set low Flag 2

2: Initiate Telecommunications (P97)
  1:      22   Phone Modem/9600 Baud
  2:      2   Disabled when User Flag 2 is High
  3:      45   Seconds Call Time Limit
  4:      0   Seconds Before Fast Retry
  5:      0   Fast Retries
  6:      0   Minutes before Slow Retry
  7:      1   Failures Loc [Failures]
  8:   0000   Data Logger ID

H-2

3: Extended Parameters (P63)

1:	5	Phone # = 539
2:	3	
3:	9	
4:	68	"D" to call datalogger
5:	3	# of Locs to collect
6:	1	1st Loc to collect
7:	1	Flag to toggle in Remote Datalogger
8:	0	Delay

4: Extended Parameters (P63)

1:	13	Terminate character
2:	00	
3:	00	
4:	00	
5:	00	
6:	00	
7:	00	
8:	00	

3: Extended Parameters (P63)

1:	4	RF ID# of repeater site = 4
2:	32	Space
3:	1	RF ID# of 2nd site = 10
4:	0	
5:	70	"F"
6:	68	"D" to call datalogger
7:	3	# of Locs to Collect
8:	1	1st Loc to Collect

4: Extended Parameters (P63)

1:	1	Flag to toggle in Remote Datalogger
2:	0	Delay
3:	13	Terminate character
4:	0	
5:	0	
6:	0	
7:	0	
8:	0	

Programming Example 2.2: Calling CR23X using RF modems

Program: This program fragment calls a datalogger every 2 minutes at using the RF Path of "4 10F" (that is from the calling CR23X to RF ID# 4 to RF ID# 10 at the remote site). The CR23X toggles Flag 1 in the remote datalogger to trigger measuring and data collection. It collects 3 input locations beginning at the remote datalogger's location 1. These values are stored in locations 2, 3, and 4.

Flag Usage: Flag 2 is used to control Instruction 97 (when to make the call). The program need only set the flag low; Instruction 97 will set it high after a successful transfer.

*Table 1 Program

01:	1.0	Execution Interval (seconds)

1: If time is (P92)

1:	0	Minutes (Seconds --) into a
2:	2	Interval (same units as above)
3:	22	Set low Flag 2

2: Initiate Telecommunications (P97)

1:	02	RF Modem/9600 Baud
2:	2	Disabled when User Flag 2 is High
3:	45	Seconds Call Time Limit
4:	0	Seconds Before Fast Retry
5:	0	Fast Retries
6:	0	Minutes before Slow Retry
7:	1	Failures Loc [Failures]
8:	0000	Data Logger ID

H.4 REMOTE DATALOGGER PROGRAMMING

The remote datalogger should be programmed to detect when the specified flag is set high. When the flag is set high, measurements and processing are done; then the flag is set low. Once the calling datalogger detects that the flag is low, it collects the specified input locations.

Since the calling datalogger sets the flag by toggling it, the remote datalogger should be programmed so the flag is always low except when the calling datalogger sets it. If the flag is already high when the call occurs, the flag will be toggled (setting it low), and the input locations will be transferred immediately.

The calling datalogger may have to wait an entire remote datalogger execution interval before the flag set is detected. The execution interval in the datalogger being called should be set to allow the call to be completed within the time limit set in Parameter 3 of Instruction 97 of the calling datalogger.

Program example for a remote CR23X

Program: Measures battery voltage, and internal and external temperatures in response to a call from the calling datalogger.

Flag Usage: Flag 1 will be set high by the calling datalogger. The flag is lowered after the measurements are made.

APPENDIX H. CALL ANOTHER DATALOGGER VIA PHONE OR RF

```
*  1    Table 1 Programs
   01:     1          Sec. Execution Interval

01:     P91         If Flag
   01:    11          1 is set
   02:    30          Then Do

02:     P10         Battery Voltage
   01:     1          Loc :

03:     P17         Panel Temperature
   01:     2          Loc :

04:     P11         Temp 107 Probe
   01:     1          Rep
   02:     1          IN Chan
   03:     1          Excite all reps w/EXchan 1
   04:     3          Loc :
   05:     1          Mult
   06:     0          Offset

05:     P86         Do
   01:    21          Set low Flag 1

06:     P95         End

07:      P          End Table 1
```

APPENDIX I. TD OPERATING SYSTEM ADDENDUM FOR CR510, CR10X, AND CR23X MANUALS

TD OPERATING SYSTEM ADDENDUM FOR CR510, CR10X, AND CR23X MANUALS

REVISION: 1/03

TD and PakBus Operating System Addendum for CR510, CR10X, and CR23X Manuals

AD1 Major Differences

Table Data (TD) operating systems have two major differences from the standard operating systems: First - the namesake - in the way data are stored internally and second, in the options available for transferring that data to external devices. The standard operating systems support both on site external storage (i.e., storage modules) that may be manually retrieved and telecommunications. The TD operating systems have more advanced telecommunication and networking capabilities but do not support storage modules. There are two versions of the TD operating system: TD and PakBus. The PakBus operating system includes the PakBus communications protocol that allows some additional communications options (Section 12); other features are the same as the TD operating system.

The datalogger hardware and direct measurement capabilities are the same in either case.

Feature	Table Data OS	Standard OS
Internal Data Storage	Multiple Data Tables, at least one Table for each output interval.	One or two Final Storage Areas. Data Arrays output at different intervals may share the same area and are identified by ID.
Term for a set of values output together	Record (a row of the table)	Array
Term for individual values within the array or record	Field (a column of the table)	Element
Number of elements or fields in an array or record	A fixed number, determined by the program	Generally fixed but conditional elements possible. Determined by program
Conditional Output	Yes	Yes
TimeStamps	Automatic	Optional
At Site / Manual Data Transfer	**No**	**Yes**
Storage Module Support	**No**	**Yes**
Instruction 96 – Serial Output	**No**	**Yes**
Instruction 98 – Send Character	**No**	**Yes**
Telecommunications	Commands protocol with error checking on all commands and responses. Feedback to confirm commands have been accepted.	Simple commands with error checking on data sent from datalogger to computer
Error Checking for data	Yes	Yes
Confirmation for Commands	**Yes**	**No**
Data Advise – data automatically sent (speeds multi-hop RF)	**Yes**	**No**
PakBus packet switching.	**Yes – in PakBus**	**No**
Supports Wireless Sensor	**Yes – in PakBus**	**No**
Support for Satellite Transmitters	**No**	**Yes**
Analog Measurements	No Difference	No Difference
Pulse Measurements	No Difference	No Difference

AD2 Overview of Data Storage Tables

Within a data table, data is organized in records and fields. Each row in a table represents a record and each column represents a field. To understand the concept of tables it may be helpful to consider an example. A CR10-TD is to be used to monitor 3 thermocouples (TC). Each hour a temperature for each of the three TC is to be stored. The table has 4 fields : "DATE_TIME TEMP1 TEMP2 TEMP3". Each hour a new "record" would be added. The "hourly" table would then be organized as follows:

DATE_TIME	TEMP1	TEMP2	TEMP3
01/27/91 10:00:00	23.5	24.6	28.2
01/27/91 11:00:00	24.2	22.4	23.4

Only the hourly data is stored in the hourly table, Each output interval has its own table. Data tables can also be "event driven" rather than interval driven, that is a new record is stored when a specified event occurs rather than based on time. Each table is completely independent of any other tables and all records in a given table have the same number of fields.

The TD operating system supports naming of tables and fields, so any data value can be referenced by the table and field names. For example, the temperature data for the first thermocouple is referenced as "HOURLY.TEMP1". Computer software also allows the station to be named. When multiple dataloggers are in use, this can be used to reference specific data in the network. If, in the previous example, the CR10T site was named DALLAS, the first thermocouple's data values would be referenced by "DALLAS.HOURLY.TEMP1".

AD3 Converting an existing program to Table Based OS

This section is intended for those familiar with programming an Array based datalogger.

AD3.1 Programming changes

- Remove all Record Real Time instructions (Instruction 77).

- Remove all Serial Data Output and Serial Print instructions (Instructions 96 and 98).

- Remove all Initiate Telecommunication (callback) instructions (Instruction 97).

- Check all instructions which set the Output Flag (Flag 0). These should be replaced with the Data Table Instruction (Instruction 84). If the Set Active Storage Area instruction (Instruction 80) is used, it should be removed as Instruction 84 provides this functionality.

- Check all If Time Instructions (Instruction 92) as the units may change from minutes to seconds. Any instruction 92 that sets the output flag (Flag 0) is replaced by Instruction 84.

- Check the Move Time To Input Location Instruction (Instruction 18) as some parameters have changed.

- Check the Maximum and Minimum Instructions (Instructions 73 and 74) as there is only one option to store time with the value.

- Edit Input Location labels removing all spaces and special characters. Only letters, numbers, and the "_" characters are allowed. Labels should start with a letter.

- Add labels for the Final Storage values. Use the same character as are allowed for Input Location labels. See Section 2.1

AD3.2 Making the Changes with Edlog

Programs for Array based logger can be converted to Table Based using EDLOG for most of the editing by doing the following:

1. Make a copy of the original program with the name you want the new program to have: Load the original into Edlog and "Save As" the new name.

2. Remove or comment out all Instructions 77, 96, 97, and 98. (first three points in AD3.1, these instructions are not in the Table OS)

3. Save the edited program and close it in Edlog.

4. Edit the CSI file with a text editor (e.g., "Notepad" - Edlog will not allow you to make and save this change) and add –TD to the datalogger type on the first line, for example, change:
   ;{CR10X}
   to:
   ;{CR10X-TD}.
   Save the CSI file and close the editor.

5. Open the file with Edlog. Edlog should now recognize that the program is for a table data OS.

6. Add Instructions 84 where necessary and make the other necessary changes.

AD4 Summary of Differences from the Datalogger Manual:

Section	Differences
Section	**Differences**
Overview	**Figure OV2.1-2: See Figure 1.5-1 in Addendum.** **Table OV3.2-1: See Table OV4.1-1 in TD Addendum.** **OV4, OV5, OV6 :See TD Addendum.**
Section 1	**Section 1.5 A Mode** is replaced by addendum – the TD loggers allocate memory differently. **Section 1.8 - *D Mode** is replaced by TD Addendum – TD loggers do not support storeing multiple programs or Storage Modules. PakBus Settings are added to the *D Mode.
Section 2	**Replaced entirely** by TD Addendum.
Section 3	**Section 3.7.1** does not apply to the **TD operating system** which **does not use Output Flag 0.** **Table 3.8-1** Valid Flag Commands are 11 – 19 to set high and 21- 29 to set Low. Because the TD operating system **does not use Flag 0, Commands 10 and 20 are not valid with the TD operating system**. **Table 3.10-1** TD Addendum. has a corrected version
Section 4	**Does not apply:** The TD operating system does not support External Storage Peripherals.
Section 5	**Does not apply:** The communications commands and protocol of the TD operating system is different than that of the standard operating systems. Campbell Scientific provides software for communications; a description of the protocol is beyond the scope of this addendum.
Section 6	Many of the peripherals discussed in section 6 are not supported by the TD operating system.
Section 7	**No Change**
Section 8	**Replaced entirely** by TD Addendum.
Section 9	**Instruction 18** has some differences in the time options, see addendum.
Section 10	**No Change**
Section 11	**Instructions 73 and 74** have only one option for storing the time of max or min (time is output as a quoted string). **Instruction 80** – Set Active Storage Area, is not in the TD operating system. Its functions are included in Instruction 84 – Data Table. **Instruction 84** – Data Table, sets the conditions and destination for output data. This instruction is only in the TD operating system (see TD Addendum..)

Section 12	**The TD operating system does not use the output Flag 0**. Commands dealing with it are not valid.
	Instruction 92 – There is no option for minutes, **time is in seconds only**.
	Instructions Not In TD OS: **Instruction 96 – Serial Output** **Instruction 98 – Send Character** **Instruction 111 – Load Program from Flash**
	New Instructions for PakBus:
	Instruction 190 – Send or Get Input Locations **Instruction 191 – One way Final Storage Data Transfer** **Instruction 192 – PakBus Message** **Instruction 193 – Wireless Network Master Control** **Instruction 194 – Time Until Transmit** **Instruction 195 – Set Clock from Address** **Instruction 196 – Wireless Remote** **Instruction 197 – Force Route Through Address**
Section 13	**No Change**
Section 14	**No Change**

TABLE DATA ADDENDUM

MEASUREMENT AND CONTROL MODULE OVERVIEW

While this section of the addendum references the CR10X, everything but the measurement instructions in the example programs applies to the other dataloggers as well.

Table OV3.2-1 in the CR10X Manual is incorrect for the TD operating system. See Table OV4.1-1 below.

The following sections OV4, OV5, and OV6 replace those in the CR10X Manual.

OV4. PROGRAMMING THE CR10X

A program is created by entering it directly into the datalogger or into a computer using the LOGGERNET program EDLOG. This manual describes direct interaction with the CR10X. Work through the direct programming examples in this overview before using EDLOG and you will know the basics of CR10X operation as well as an appreciation for the help provided by the software. Section OV4.5 describes options for loading the program into the CR10X.

OV4.1 FUNCTIONAL MODES

CR10X/User interaction is broken into different functional MODES (e.g., programming the measurements and output, setting time, etc.). The modes are referred to as Star (*) Modes since they are accessed by first keying *, then the mode number or letter. Table OV4.1-1 lists the CR10X Modes.

OV4.2 KEY DEFINITION

Keys and key sequences have specific functions when using the CR10KD keyboard or a computer in the remote keyboard state (Section 5). Table OV4-2 lists these functions. In some cases, the exact action of a key depends on the mode the CR10X is in and is described with the mode in the manual.

When using a computer/terminal to communicate with the CR10X (Telecommunications) there are some keys available in addition to those found on the CR10KD. Table OV4.2-2 lists these keys.

TABLE OV4.1-1. * Mode Summary

Key	Mode
*0	LOG data and indicate active Tables
*1	Program Table 1
*2	Program Table 2
*3	Program Table 3, subroutines only
*5	Display/set real time clock
*6	Display/alter Input Storage data, toggle flags and ports
*7	Display Data Storage Table data
*9	Display Data Storage Table sizes
*A	Memory allocation/reset
*B	Signature/status
*C	Security

TABLE OV4.2-1. Key Description/Editing Functions

Key	Action
0-9	Key numeric entries into display
*	Enter Mode (followed by Mode Number)
A	Enter/Advance
B	Back up
C	Change the sign of a number or index an input location to loop counter
D	Enter the decimal point
#	Clear the rightmost digit keyed into the display
#A	Advance to next instruction in program table (*1, *2, *3)
#B	Back up to previous instruction in program table.
#D	Delete entire instruction

TABLE OV4.2-2. Additional Keys Allowed in Telecommunications

Key	Action
-	Change Sign, Index (same as C)
CR	Enter/advance (same as A)

OV4.3 PROGRAMMING SEQUENCE

In routine applications, the CR10X measures sensor output signals, processes the measurements over some time interval and stores the processed results. A generalized programming sequence is:

1. Enter the execution interval. In most cases, the execution interval is determined by the desired sensor scan rate.

2. Enter the Input/Output instructions required to measure the sensors.

3. If processing in addition to that provided by the Output Processing Instructions (step 5) is required, enter the appropriate Processing Instructions.

4. Enter the Data Table Instruction 84 to test the output condition and output when the condition is met. For example, use

    Instruction 84 to output based on time.

    Instruction 84 to output every execution interval.

    Instruction 84 to output based on a Program Flag.

    This instruction must precede the Output Processing Instructions which store data in a Data Storage Table. Instructions are described in Sections 9 through 12.

5. Enter the Output Processing Instructions to store processed data in the Data Storage Table. The order in which data are stored is determined by the order of the Output Processing Instructions in the table.

6. Repeat steps 4 through 6 for additional outputs on different intervals or conditions.

> **NOTE**: The program must be executed for output to occur. Therefore, the interval specified with the Data Table Instruction is set must be evenly divisible by the execution interval. For example, with a 2 minute execution interval and a 5 minute output interval, the program will only be executed on the even multiples of the 5 minute intervals, not on the odd. Data will be output every 10 minutes instead of every 5 minutes.

Execution intervals are synchronized with midnight. Output intervals set with Instruction 84 are synchronized with real time starting at midnight, January 1, 1990.

OV4.4 INSTRUCTION FORMAT

Instructions are identified by an instruction number. Each instruction has a number of parameters that give the CR10X the information it needs to execute the instruction.

The CR10X Prompt Sheet has the instruction numbers in red, with the parameters briefly listed in columns following the description. Some parameters are footnoted with further description under the "Instruction Option Codes" heading.

For example, Instruction 73 stores the maximum value that occurred in an Input Storage location over the output interval. The instruction has three parameters (1) REPetitionS, the number of sequential Input Storage locations on which to find maxima, (2) TIME, an option of storing the time of occurrence with the maximum value, and (3) LOC the first Input Storage location operated on by the Maximum Instruction. The codes for the TIME parameter are listed in the "Instruction Option Codes".

The repetitions parameter specifies how many times an instruction's function is to be repeated. For example, four 107 thermistor probes may be measured with a single Instruction 11, Temp-107, with four repetitions. Parameter 2 specifies the input channel of the first thermistor (the probes must be connected to sequential channels). Parameter 4 specifies the Input Storage location in which to store measurements from the first thermistor. If location 5 were used and the first probe was on channel 1, the temperature of the thermistor on channel 1 would be stored in input

location 5, the temperature from channel 2 in input location 6, etc.

Detailed descriptions of the instructions are given in Sections 9-12. Entering an instruction into a program table is described in OV5.

OV4.5 ENTERING A PROGRAM

Programs are entered into the CR10X in one of two ways:

1. Keyed in using the CR10X keyboard
2. Stored on disk/seat from computer

A program is created by keying it directly into the datalogger as described in Section OV5, or on a PC using EDLOG.

EDLOG is used to develop programs for Campbell Scientific CR10X dataloggers. EDLOG is a prompting editor for writing and documenting programs for Campbell Scientific CR10X dataloggers. Program files developed with EDLOG can be downloaded directly to the CR10X using NetAdmin. NetAdmin supports communication via direct wire, telephone, or Radio Frequency (RF).

OV5. PROGRAMMING EXAMPLES

We will start with a simple programming example. There is a brief explanation of each step to help you follow the logic. When the example uses an instruction, find it on the Prompt Sheet and follow through the

description of the parameters. Using the Prompt Sheet while going through these examples will help you become familiar with its format. Sections 9-12 have more detailed descriptions of the instructions.

With the Wiring Panel connected to the CR10X, hook up the power leads as described in Section OV1.2. Next, connect the CR10X to either a CR10KD Keyboard/Display or the computer (Section OV3). The programming steps in the following examples use the keystrokes possible on the keyboard/display. With a terminal, some responses will be slightly different.

If the CR10KD is connected to the CR10X when it is powered up, the display will show:

Display	Explanation
HELLO	On power-up, the CR10X displays "HELLO" while it checks the memory (this display occurs only with the CR10KD).

after a few seconds delay

:96	The size of the machine's total memory (RAM plus 32 K of ROM), in this case 96K

OV5.1 SAMPLE PROGRAM 1

In this example the CR10X is programmed to read its own internal temperature (using a built in thermistor) every 5 seconds and to send the results to Final Storage.

Key	Display	Explanation
*	00:00	Enter mode.
1	01:00	Enter Program Table 1.
A	01:0.0000	Advance to execution interval (In seconds)
5	01:5	Key in an execution interval of 5 seconds.
A	01:P00	Enter the 5 second execution interval and advance to the first program instruction location.
17	01:P17	Key in Instruction 17 which directs the CR10X to measure the internal temperature in degrees C. This is an Input/Output Instruction.
A	01:0000	Enter Instruction 17 and advance to the first parameter.
1	01:1	The input location to store the measurement, location 1.
A	02:P00	Enter the location # and advance to the second program instruction.

The CR10X is now programmed to read the internal temperature every 5 seconds and place the reading in Input Storage Location 1. The program can be compiled and the temperature displayed.

Key	(ID:Data)	Explanation
*0	LOG 1	Exit Table 1, enter *0 Mode, compile table and begin logging.
*6	06:0000	Enter *6 Mode (to view Input Storage).
A	01:21.234	Advance to first storage location. Panel temperature is 21.234°C (the display will show the actual temperature).

Wait a few seconds:

	(ID:Data)	Explanation
	01:21.423	The CR10X has read the sensor and stored the result again. The internal temp is now 21.423°C. The value is updated every 5 seconds when the table is executed. At this point the CR10X is measuring the temperature every 5 seconds and sending the value to Input Storage. No data are being saved. The next step is to have the CR10X send each reading to Final Storage.
*1	01:00	Exit *6 Mode. Enter program table 1.
2A	02:P00	Advance to 2nd instruction location (this is where we left off).
84	02:P84	This is the Data Table Instruction.
A	01:0.0000	Enter 84 and advance to the first parameter (which is the time into the interval).
0	01:0	This parameter determines when in the output internal data is stored. 0 stores data on the even interval.
A	02:0.000	Enter 0 and advance to the second parameter.
0	02:0	This parameter specifies the output interval. 0 stores data each execution.
A	03:0.0000	Enter 0 and advance to third parameter.
1000	03:1000.00	This parameter specifies how many records to store in the table before overwriting the oldest. For this example, keep 1000 records.
A	03:P00	Enter 1000 and advance to the third program instruction.
70	03:P70	The SAMPLE instruction. It directs the CR10X to take a reading from an Input Storage location and send it to Final Storage (an Output Processing Instruction).
A	01:0000	Enter 70 and advance to the first parameter (repetitions).
1	01:1	There is only one input location to sample; repetitions = 1.
A	02:0000	Enter 1 and advance to second parameter (Input Storage location to sample).
1	02:1	Input Storage Location 1, where the temperature is stored.
A	04:P00	Enter 1 and advance to fourth program instruction.
*	00:00	Exit Table 1.
0	LOG 1	Enter *0 Mode, compile program, log data.

OV5.2 SAMPLE PROGRAM 2

This second example is more representative of a real-life data collection situation. Once again the internal temperature is measured, but it is used as a reference temperature for the differential voltage measurement of a type T (copper-constantan) thermocouple; the CR10X should have arrived with a short type T thermocouple connected to differential channel 5.

When using a type T thermocouple, the copper lead (blue) is connected to the high input of the differential channel, and the constantan lead (red) is connected to the low input.

A thermocouple produces a voltage that is proportional to the difference in temperature between the measurement and the reference junctions.

To make a thermocouple (TC) temperature measurement, the temperature of the reference junction (in this example, the approximate panel temperature) must be measured. The CR10X takes the reference temperature, converts it to the equivalent TC voltage relative to 0°C, adds the measured TC voltage, and converts the sum to temperature through a polynomial fit to the TC output curve (Section 13.4).

The internal temperature of the CR10X is not a suitable reference temperature for precision thermocouple measurements. It is used here for the purpose of training only. To make thermocouple measurements with the CR10X, purchase the Campbell Scientific Thermocouple Reference, Model CR10XCR (Section 13.4) and make the reference temperature measurement with Instruction 11.

Instruction 14 directs the CR10X to make a differential TC temperature measurement. The first parameter in Instruction 14 is the number of times to repeat the measurement. Enter 1, because in this example there is only one thermocouple. If there were more than 1 TC, they could be wired to sequential channels, and the number of thermocouples entered for repetitions. The CR10X would automatically advance through the channels sequentially and measure all of the thermocouples.

Parameter 2 is the voltage range to use when making the measurement. The output of a type T thermocouple is approximately 40 microvolts

per degree C difference in temperature between the two junctions. The ±2.5 mV scale will provide a range of ±2500/40 = ±62.5°C (i.e., this scale will not overrange as long as the measuring junction is within 62.5°C of the panel temperature). The resolution of the ±2.5 mV range is 0.33 µV or 0.008°C.

Parameter 3 is the analog input channel on which to make the first, and in this case only, measurement. Parameter 4 is the code for the type of thermocouple used. This information is located on the Prompt Sheet or in the description of Instruction 14 in Section 9. The code for a type T (copper-constantan) thermocouple is 1.

Parameter 5 is the Input Storage location in which the reference temperature is stored. Parameter 6 is the Input Storage location in which to store the measurement (or the first measurement; e.g., if there are 5 repetitions and the first measurement is stored in location 3, the final measurement will be stored in location 7). Parameters 7 and 8 are the multiplier and offset. A multiplier of 1 and an offset of 0 outputs the reading in degrees C. A multiplier of 1.8 and an offset of 32 converts the reading to degrees F.

In this example, the sensor is measured once a minute, and the average temperature is output every hour. Once a day the maximum and minimum temperatures and the times they occur will be output.

The first example described program entry one keystroke at a time. This example does not show the "A" key. Remember, "A" is used to enter and/or advance (i.e., between each line in the example below). This format is similar to the format used in EDLOG.

It's a good idea to have both the manual and the Prompt Sheet handy when going through this example. You can find the program instructions and parameters on the Prompt Sheet and can read their complete definitions in the manual.

To obtain daily output, the Data Table instruction is followed by the Output Instructions to store the daily maximum and minimum temperatures and the time each occurs.

SAMPLE PROGRAM 2

Instruction # (Loc:Entry)	Parameter (Par#:Entry)	Description
*1		Enter Program Table 1
01:60		60 second (1 minute) execution interval
Key "#D" repeatedly until is displayed	01:P00	Erase previous Program before continuing.
01:P17		Measure internal temperature
	01:1	Store temp in Location 1
02:P14		Measure thermocouple temperature (differential)
	01:1	1 repetition
	02:1	Range code (2.5 mV, slow)
	03:5	Input channel of TC
	04:1	TC type: copper-constantan
	05:1	Reference temp is stored in Location 1
	06:2	Store TC temp in Location 2
	07:1	Multiplier of 1
	08:0	No offset
03:P84		Data Table Instruction
	01:0	0 seconds into the interval
	02:3600	3600 second (60 min.) internal
	03: 0	Automatically allocate # of records

The CR10X is programmed to measure the thermocouple temperature every sixty seconds. The CR10X automatically allocates the number of records. Time information is automatically stored. Next the output instruction for the average is added.

Instruction # (Loc.:Entry)	Parameter (Par.#:Entry)	Description
04:P71		Average instruction
	01:1	One repetition
	02:2	Location 2 - source of TC temps. to be averaged
05:P84		Data Table Instruction
	01:0	0 seconds into the interval
	02:86400	86400 second interval (24 hrs.)
	03:0	Automatically allocate # of records
06: P73		Maximize instruction
	01:1	One repetition
	02:1	Output the time of the daily maximum
	03:2	Data source is Input Storage Location 2.
07: P74		Minimize instruction
	01:1	One repetition
	02:1	Output the time of the daily minimum
	03:2	Data source is Input Storage Location 2.

The program to make the measurements and send the desired data to Final Storage has been entered. The program is complete. The clock must now be set so that the date and time tags are correct. (Here the example reverts back to the key by key format.)

Key	Display	Explanation
*5	00:21:32	Enter *5 Mode. Clock running but not set correctly.
A	05:01.01	Advance to month-day (MMDD).
1004	05:1004	Key in MMDD (Oct 4 in this example).
A	05:1990	Enter and advance to location for year.
1994	05:1994	Key in year.
A	05:00:21	Enter and advance to location for hours and minutes (24 hr. time).
1324	05:1324	Key in hrs.:min. (1:24 PM in this example).
A	05:27.250	Key in seconds
30	05:30	
A	13:24:30	Clock set and running. Changes made when A was pressed.
*0	LOG 1	Exit *5, compile Table 1, commence logging data.

OV5.3 EDITING AN EXISTING PROGRAM

When editing an existing program in the CR10X, entering a new instruction inserts the instruction; entering a new parameter replaces the previous value.

To insert an instruction, enter the program table and advance to the position where the instruction is to be inserted (i.e., P in the data portion of the display) key in the instruction number, and then key A. The new instruction will be inserted at that point in the table, advance through and enter the parameters. The instruction that was at that point and all instructions following it will be pushed down to follow the inserted instruction.

An instruction is deleted by advancing to the instruction number (P in display) and keying #D (Table 4.2-1).

To change the value entered for a parameter, advance to the parameter and key in the correct value then press A. Note that the new value is not entered until A is keyed.

OV6. DATA RETRIEVAL OPTIONS

Data is retrieved over some form of telecommunications link, whether it be RF (radio), telephone, short haul modem, coaxial cable (mulitdrop) , or direct link. The table data operating system does not support on-line output to peripheral storage devices (see Figure OV 6.1-1).

The retrieval of data does NOT erase those data from the Data Storage Tables in Final Storage. The data remains in the table ring memory until:

The are written over by new records of data. (Section 2.1)

Input Storage memory is reallocated (Section 1.5)

The datalogger program is changed and compiled.

Power to the datalogger is turned off.
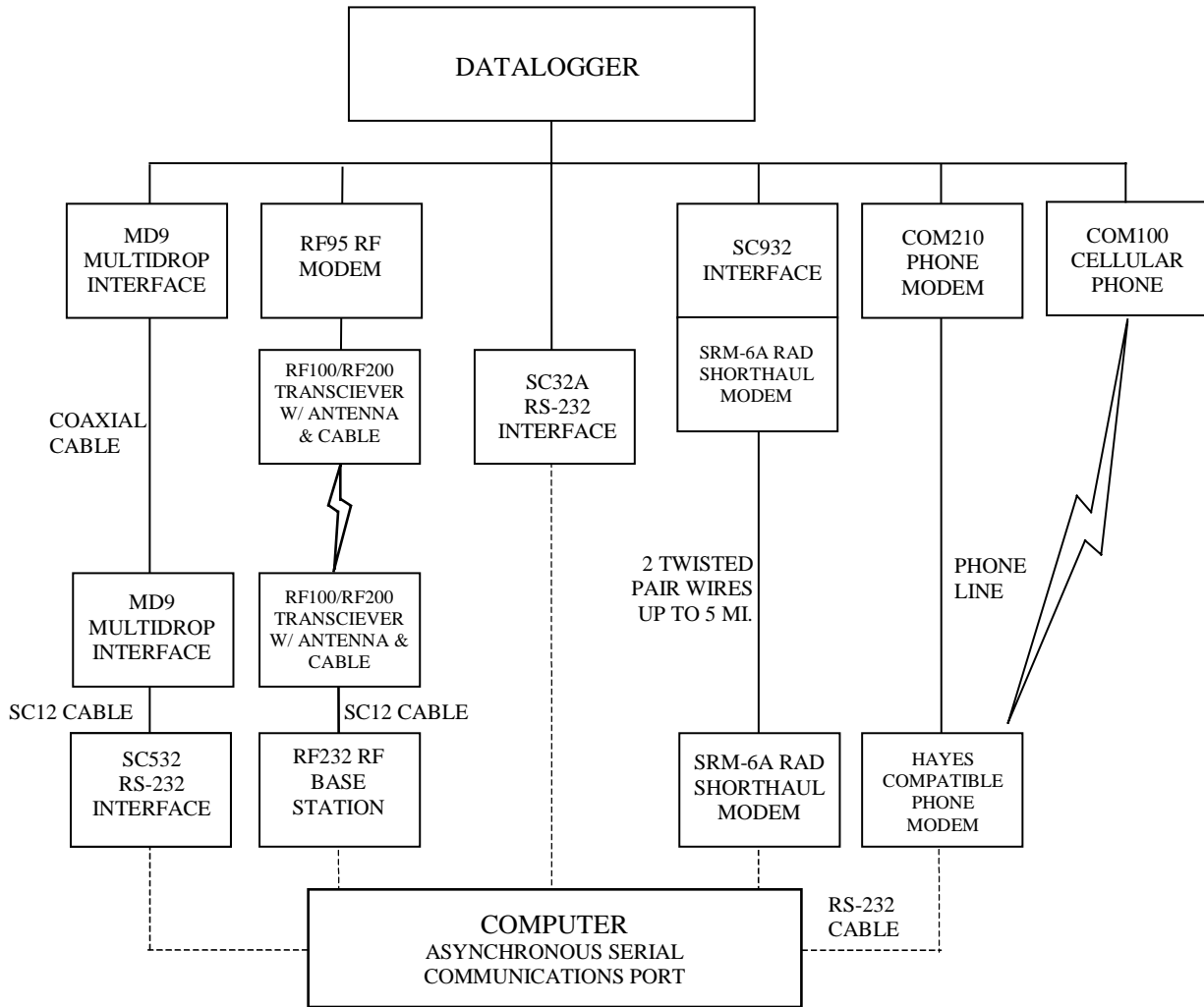
NOTES:
1. ADDITIONAL METHODS OF DATA RETRIEVAL ARE:
   A. SATELLITE TRANSMISSION
   B. DIRECT DUMP TO PRINTER
   C. VOICE PHONE MODEM TO VOICE PHONE OR PC WITH HAYES COMPATIBLE PHONE MODEM

2. THE DSP4 HEADS UP DISPLAY ALLOWS THE USER TO VIEW DATA IN INPUT STORAGE. ALSO BUFFERS FINAL STORAGE DATA AND WRITES IT TO CASSETTE TAPE, PRINTER OR STORAGE MODULE.

3. ALL CAMPBELL SCIENTIFIC RS-232 INTERFACES HAVE A FEMALE 25 PIN RS-232 CONNECTOR.

FIGURE OV6.1-1. Data Retrieval Hardware Options

SECTION 1. FUNCTIONAL MODES

Sections 1.5 and 1.8 are replaced by the following sections.

1.5 MEMORY ALLOCATION - ∗A

1.5.1 INTERNAL MEMORY

When powered up with the keyboard display attached, the CR10KD displays HELLO while performing a self check. The total system memory is then displayed in K bytes. The size of memory can be displayed in the ∗B mode.

Input Storage is used to store the results of Input/Output and Processing Instructions. The values stored in input locations may be displayed using the ∗6 Mode (Section 1.3).

Final Storage holds stored data for a permanent record. Output Instructions store data in Final Storage Data Tables. The data in Final Storage can be monitored using the ∗7 Mode (Section 2.3).

Intermediate Storage is a scratch pad for Output Processing Instructions. It is used to store the results of intermediate calculations necessary for averages, standard deviations, histograms, etc. Intermediate Storage is not accessible by the user.

Each Input or Intermediate Storage location requires 4 bytes of memory. Each Final Storage location requires 2 bytes of memory. Low resolution data points require 1 Final Storage location and high resolution data points require 2. Section 2 describes Final Storage and data retrieval in detail.

Figure 1.5-1 lists the basic memory functions and the amount of memory allotted to them.

**Flash Memory
(EEPROM)**

**SRAM
Total 128 Kbytes**

How it works:
The **Operating System** is loaded into Flash Memory at the factory. **System Memory** is used while the CR10X is running for calculations, buffering data and general operating tasks.

Any time a user loads a program into the datalogger, the program is compiled in SRAM and stored in the **Active Program** areas. If the datalogger is powered off and then on, the Active Program is loaded from Flash and run.

The Active Program is run in SRAM to maximize speed. The program accesses **Input Storage** and **Intermediate Storage** and stores data into **Final Storage** for later retrieval by the user.

Table Data Operating Systems Save Input Storage and Final Storage Lables.

Flash Memory, increases Final Storage by 524,288 data values per Mbyte.

(Memory Areas separated by dashed lines:
can be re-sized.)

Operating System
(96 Kbytes-CR10X)
(128 Kbytes-CR23X)

Active Program
(16 Kbytes)

Input and Final Storage Lables
(16 Kbytes)

**32K SRAM
Main Memory**

System Memory

Active Program
(automatically allocated)

Input Storage
default: 28 locations
(Size Set with *A)

Intermediate Storage
(automatically allocated)

PakBus Routing Table
(Size Set with *D 15)

PakBus Settings
(Size Set with *A)

96K SRAM

Final Storage Data Tables

Flash EEPROM
Optional in CR10X

Final Storage
(*Additional* 524,288 locations per Mbyte)

FIGURE 1.5-1. Datalogger Memory

1.5.2 *A MODE

The *A Mode is used to 1) check the size of Input Storage, Intermediate Storage, Final Storage, Program Memory; PakBus and user Settings memory 2) check the number of bytes remaining in Flash Program memory; Main Memory, and Label Memory 3) change the memory allotted to Input Locations and Settings; and 5) to completely reset the datalogger.

When *A is entered, the first number displayed is the number of memory locations allocated to Input Storage. The "A" key is used to advance through the next 6 windows. Table 1.5-2 describes what the values in the *A Mode represent.

The sizes of Input Storage and Settings Memory may be altered by keying in the desired value and entering it by keying "A".

TABLE 1.5-2. Description of ∗A Mode Data

Keyboard Entry	Display ID: Data	Description of Data
∗ A	01: XXXX	**Input Storage Locations** (minimum of 28, maximum of 6655, but the usable maximum is less than this because intermediate and program storage require some of this memory). This value can be changed by keying in the desired number.
A	02: XXXX	**Intermediate Storage Locations** (maximum limited by available memory and constraints on Input and Final Storage). The CR10X-TD will assign the exact number needed for the active program. The CR10X-TD erases all data whenever the program is changed and compiled.
A	03: XXXXX	**Final Storage Locations** (minimum of 0, maximum limited by available memory). Changing this number automatically reallocates Final Storage Area 1.
A	04: XXXXX	**Bytes allocated for user program**. The CR10X-TD will assign the exact number needed. The CR10X-TD erases all data whenever the program is changed and compiled. **Key in 98765 to completely reset datalogger.**
A	05:	**Bytes free in Flash Memory for active program.** The user cannot change this window. It is a function of window 5 and the program.
A	06:	**PakBus and user Settings memory**
A	07:	Main Memory Free
A	06:	Label Bytes Free

The maximum size of of Final Storage is determined by the memory installed (Table 1.5-1). The size of Final Storage and the rate at which data are stored determines how long it will take for Final Storage to fill, at which point new data will write over old.

Twenty-eight is the minimum number of Input locations allowed. Intermediate Storage and Final Storage are erased when the number of Input locations is changed. This feature may be used to clear memory without altering programming. The number of locations does not actually need to be changed; the same value can be keyed in and entered.

Intermediate Storage and Program Memory are automatically allocated. All data are erased any time the program is changed and compiled. If there is not enough memory available in the 32K Main Memory for the Intermediate Storage required by the current program, the "E:04" ERROR CODE will be displayed in the ∗0, ∗6, and ∗B Modes.

After repartitioning memory, the program must be recompiled. Compiling erases Intermediate Storage. Compiling with ∗0 erases Input Storage; compiling with ∗6 leaves Input Storage unaltered (If its size was unchanged).

ENTERING 98765 in the program memory window 6 COMPLETELY RESETS THE CR10X. All memory is erased including the program and memory is checked. Memory allocation returns to the default. The reset operation requires approximately 1 minute for a CR10X, 5 minutes for a CR10X-1M, and 10 minutes for a CR10X-2M. Please be patient while the reset takes place; if the CR10X is turned off in the middle of a reset, it will perform the reset the next time it is powered up.

1.6 MEMORY TESTING AND SYSTEM STATUS - ∗B

No changes from standard operating system, see datalogger manual.

1.7 ∗C MODE -- SECURITY

No changes from standard operating system, see datalogger manual.

1.8 ∗D MODE – TRANSFER PROGRAMS, GENERAL SETTINGS

The ∗D Mode is used to transfer datalogger programs between a datalogger and a computer, to erase a program, to set the degree

to which memory is cleared on powerup, to set the PakBus ID, and to set communication to full or half duplex.

CSI datalogger support software makes use of the ∗D Mode to upload and download programs from a computer. Appendix C gives some additional information on Commands 1 and 2 that are used for these operations.

When "∗D" is keyed in, the CR10X will display "13:00". A command (Table 1.8-1) is entered by keying the command number and "A".

TABLE 1.8-1. ∗D Mode Commands

Command	Description
1	Send (Print) ASCII Program
2	Load ASCII Program, ∗0 Compile
2--	Load ASCII Program, ∗6 Compile
3	# Rings Before Answering Phone
7	Erase Current Program
10	Set Powerup Options
12	Set Initial Baud
15	PakBus Address/Routing Table
16	Memory for General Purpose File
17	PakBus Routing Table
18	PakBus Beacon Interval
19	PakBus Neighbor List

If the CR10X program has not been compiled when the command to save a program is entered, it will be compiled before the program is saved. When a program is loaded, it is immediately compiled and run. When a command is complete, "13:0000" is displayed; ∗D must be entered again before another command can be given.

TABLE 1.8-2. Program Load Error Codes

E 94	Program Storage Area full
E 95	Program does not exist in flash
E 96	Storage Module not connected or wrong address
E 97	Data not encountered within 30 sec.
E 98	Uncorrectable errors detected
E 99	Wrong type of file or Editor Error

1.8.1 ERASING CURRENT PROGRAM

The 7 command may be used to delete the current program as show in Table 1.8-3.

TABLE 1.8-3 Deleting Current Datalogger Program

Key entry	Display
∗D	13:00
7A	07:00

You may now enter:

0A	Erase active program (i.e., load a blank program; memory allocation and Final Storage are reset).

1.8.2 PROGRAM TRANSFER WITH STORAGE MODULE

Not supported in Table Data Operating Systems.

1.8.3 FULL/HALF DUPLEX

Not supported in Table Data Operating Systems.

1.8.4 SET DATALOGGER ID

Command 8 not supported in Table Data Operating Systems.

1.8.5 SETTING POWERUP OPTIONS

Setting options for the Program on Powerup allows the user to specify what information to retain from when the datalogger was last on. This allows Flag/Port status, the User Timer, and the Input/Intermediate Storage to be cleared or not cleared.

Table 1.8-8. Setting Powerup Options

Key entry	Display
∗D	13:00
10A	10:0X

Where X is the powerup option currently selected. You may now change the option:

0A	Clears input locations, ports, flags, user timer, and intermediate storage locations.
1A	Clears intermediate storage only (leaves Input Storage, Flags/Ports, and User Timer as is).
2A	Doesn't clear anything.

1.8.6 SET INITIAL BAUD

Table 1.8-10 shows the option codes available for setting the initial baud rate. Setting the initial baud rate forces the CR10X to try the selected baud rate first when connecting with a device.

TABLE 1.8-9. Set Initial Baud Rate / Set RS232 Power

Key Entry	Display	Comments
*D	13:00	Enter Command
12A	12:00	Connect Baud Rate Enter Baud Rate Code X (Table 1.8-11).

TABLE 1.8-10. Baud Rate Codes

X = 0   300 Baud
X = 1   1200 Baud
X = 2   9600 Baud
X = 3   76.8 K Baud

1.8.7 SET PROGRAM COMPILE OPTION

Command 13 is not supported in Table Data operating systems.

1.8.8 SET PAKBUS ADDRESS

*D 15 allows the user to set the PakBus Address of the datalogger and to set the maximum size for its routing table.

TABLE 1.8-11. PakBus Address and Routing Table

Key Entry	Display	Comments
*D	13:00	Enter Command
15A	15:xxxx	PakBus Address , Enter zero if the datalogger is not to be used as a PakBus device (1..4094 is legal, the default is 1)
A	01:xxxx	If the datalogger is to be used a a router, enter the maximum number of nodes (PakBus Addresses) to allocate space for in the pakbus network. 0 = leafnode, <>0 = router
A	02:xxxx	Enter the maximum number of neighbors in the pakbus network to allocate space for. This parameter is used only if datalogger is used as a router (01: is non-zero).
A	03:xxxx	Enter maximum number of routers in the pakbus network to allocate space for. This parameter is used only if datalogger is used as a router (01: is non-zero).
A	04:xxxx	Enter the PakBus address for a default router (1..4094, 0 for no default router). The default router is used for a message if the destination PakBus address is not in the routing table. A router discovering new routes will not explore beyond its own default router.

The memory for a routing table comes out of the pool for program, input locations, and intermediate storage.

The total number of bytes used for the routing table =

Nodes x 12
+ Neighbors x 8
+ Routers x 6
+ (Routers x (Nodes – Routers) + (Routers x (Routers – 1))/2) x 4

The *D15 entries are sent when the program is retrieved. They can also be set like other *D settings via the DLD file.

1.8.9 ALLOCATE MEMORY FOR GENERAL PURPOSE FILES

*D16:xx   ;allocate xx 64K byte chunks of memory for general purpose files. The area comes out of final storage space. Files are stored in a circular buffer (ring memory) in this space.

1.8.10 VIEW ROUTING TABLE

*D17 allows viewing the current routing table information. This is view only.

TABLE 1.8-12. Values in Routing Table

Key Entry	Display	Comments
*D	13:00	Enter Command
17A		Enter the view routing table command
	01:xxxx	;pakbus address of destination node
	02:xxxx	;via neighbor with xxxx pakbus address
	03:xxxx	;a worst case response time metric (seconds)

(Repeats for next destination node.)

1.8.11 SET PAKBUS ROUTER BEACON INTERVAL

TABLE 1.8-13. Set Beacon interval

Key Entry	Display	Comments
*D	13:00	Enter Command
18A		Enter the beacon interval settings
A	01:xxxx	Enter the Interval (seconds) for SDC7
A	02:xxxx	Enter the Interval (seconds) for SDC8
A	03:xxxx	Enter the Interval (seconds) for CS I/O Pin Enabled, 9600 baud
A	04:xxxx	Enter the Interval (seconds) for RS232, 9600 baud (CR23X only)

1.8.12 PAKBUS NEIGHBOR FILTER

In some networks, sending beacons can be disruptive. Entering values in the *D19 mode disables the beacon. A PakBus datalogger with nonzero *D19 settings will not send beacons and will only respond to beacons from nodes with addresses in the neighbor list.

Instead of sending beacons, the datalogger will send "hello" messages to neighbors in the list to determine if it can communicate with them. Neighbors (or potential neighbors) should be nodes that communicate directly with the datalogger without going through a router.

Note that this list is not automatically cleared by compiling a new program. (It may be changed if the new program contains *D19 entries.) It can be edited by changing entries. Once 0 is entered for a neighbor address, all entries beyond the 0 entry are cleared.

TABLE 1.8-14. Set PakBus Neighbors

Key Entry	Display	Comments
*D	13:00	Enter Command
19A	19:00	Port (17- SDC7, 18 – SDC8, 02 – CSI/O, 02—CR23X RS232 port, 9600 baud
A	19:0000	Interval in seconds of the expected rate of communication. A neighbor is aged after 2.5 times this interval and the Hello attempt will be reinitiated.
A	01:xxxx	PakBus Address of neighbor
A	01:xx	Swath of neighbors with sequential addresses starting with above address.
A	02:xxxx	PakBus Address of neighbors
A	02:xx	Swath of neighbors with sequential addresses starting with above address.
... etc.		
A	nn:0000	Terminates the list.

1.9 *9 DATA TABLES SIZES.

The *9 Mode is used to view the sizes of the Data Storage Tables (section 2.1) created by the datalogger program (*1, *2, or *3). The *9 Mode will also display how long until any automatically allocated Data Storage Tables fill. All Data Storage Tables are in a ring configuration such that the oldest records are overwritten by new records once the table is full. The sizes are given as the number of records. A record can be thought of as a row of data where each field (i.e., column) is a data value associated with an Output Processing Instruction. The order and number of fields in a Data Table are determined by the Output Processing Instructions following the Data Table Instruction. The tables are numbered in the order the Data Table Instruction appear in the Program Tables, *1 first, *2 second, and *3 last.

TABLE 1.9-1. Description of * 9 Data

Keyboard Entry	Display ID: Data	Description
*9	09:xx	Number of tables/ Enter table number to jump to that table or 0 to see next parameter.
0A	00:x.xxxx	Days before any automatically allocated table fills.
A	01:xxxxx.	Number of records in table 01
A	02:xxxxx	Number of records in table 02
A	nn:xxxxx.	Number of records in table nn

SECTION 2. INTERNAL DATA STORAGE

2.1 FINAL STORAGE AND DATA TABLES

Final Storage is that portion of memory where final processed data are stored. It is from Final Storage that data is transferred to your computer.   With the TD datalogger, Final Storage is organized into Data Storage Tables. These data tables should not be confused with the program tables *1, *2, and *3 that contain the datalogger program.

Within each data table, data is organized in records and fields. Each row in a table represents a record and each column represents a field. To understand the concept of tables it may be helpful to consider an example. A CR10X is to be used to monitor 3 thermocouples (TC). Each hour a temperature for each of the three TC is to be stored. The table has 4 fields: "DATE_TIME TEMP1 TEMP2 TEMP3." Each hour a new "record" would be added. The "hourly" table would then be organized as follows:

DATE_TIME	TEMP1	TEMP2	TEMP3
01/27/91 10:00:00	23.5	24.6	28.2
01/27/91 11:00:00	24.2	22.4	23.4

Only the hourly data is stored in the hourly table, Each output interval has its own table. Data tables can also be "event driven" rather than interval driven, that is a new record is stored when a specified event occurs rather than based on time. Each table is completely independent of any other tables and all records in a given table have the same number of fields.

Each table is allocated (manually by the user or automatically by the datalogger) a number of records. Different tables have different numbers of records. Each data table is in a ring memory configuration such that when the allocated number of records has been stored, each subsequent new record will overwrite the oldest stored record. The *9 Mode may be used to view the size of the Data Storage Tables. (Section 1.9)

The TD datalogger supports naming of tables and fields, so any data value can be referenced by the table and field names. For example, the temperature data for the first thermocouple is referenced as "HOURLY.TEMP1." As Data Tables are allocated in the datalogger program, some Final Storage Memory is reallocated for the storage of these labels and other data table overhead.

> **NOTE:** All Data Storage Tables are reallocated and erased whenever the datalogger program is recompiled (*0, *6, *B), when Input Storage Memory is reallocated (*A), or when a new datalogger program is transferred from the computer to the datalogger. ALWAYS RETRIEVE UNCOLLECTED DATA BEFORE MAKING ANY CHANGES.

A time stamp and record number are automatically included with the each record in each table. These are used as part of the data collection protocol.

2.1.1 TIME AND TIMESTAMPS

Each record in a table has a time stamp associated with it. With Instruction 84 set for interval output (a interval in seconds is specified as the second parameter), time is not actually stored with each record. Using the timestamp of the last record stored and the table interval, the datalogger can calculate the timestamp for any previous record. When retrieved, each record in the data file will have a timestamp. This saves 6 bytes per record by not storing time with each record. A consequence of not storing time is that if output does not occur at a scheduled time, the datalogger must keep track of the discontinuity in the timestamps so it can correctly calculate timestamps for records older than the missing record. The datalogger will keep track of the 10 most recent discontinuities in each table. If more than ten discontinuities occur, records with timestamps older than the oldest discontinuity cannot be reliably timestamped when collected. For this reason interval tables should not be used if outputs will be routinely missed. Outputs can be missed (discontinuities can occur) when:

- The datalogger clock is changed such that is passes an output interval.

- The output interval is not an even multiple of the scan rate (table execution interval).

- Table execution is such that Instruction 84 is not executed each scan.

- Table overruns occur.

- Watchdog errors (E08) occur.

2.1.2 RECORD NUMBERS

In addition to a timestamp, each record has record number. The record numbers are unique within a data table. Record numbers are 4 byte unsigned numbers ranging from 0 to 4,294,967,296. When the datalogger program is compiled the next record number for each table is set to zero.

2.1.3 TABLES AND FIELDS.

There are four "built in" tables. These tables are present when the datalogger is powered on. These tables are named INLOCS, TIMESET, ERRORLOG, and STATUS. The fields within these tables are also already defined with the exception of the INLOCS. The field names for each of the fields is given below. TmStamp is the label for the timestamp and RecNbr is the label for the record number.

TimeSet
TmStamp, RecNbr, OldTime

ErrorLog
TmStamp, RecNbr, Code

Status
TmStamp, RecNbr, Battery, WatchDog, OverRuns, InLocs, PrgmFree, Storage, Tables, DaysFull, Holes, PrgmSig, PromSig, PromID, ObjSrlNo

Where the labels are defined as follows:

Battery – Indicates the datalogger battery voltage.

WatchDog – The number of Watchdog Errors (E08). (Maximum 99). Section 3.10

OverRuns – Program table overruns that have occurred (Maximum 99). Section 1.1.1.

InLocs – Number of Input Location that have been allocated. Section 1.5.2

PrgmFree – Amount (bytes) of program memory remaining. Section 1.5.2

Storage – Number of Final Storage Locations available for Data Storage Tables. Section 1.5.2

Tables – Number of user created Data Tables.

DaysFull – Size (in days) of the Data Storage Tables using automatic record allocation. See Instruction 84.

Holes – Number of missed records or holes in all Data Storage Tables. Section 2.1.1

PrgmSig – Signature of program memory program. Same as *B mode first window. Section 1.6.

PromSig – Signature of datalogger PROM. Same as *B mode second window. Section 1.6.

Prom ID – Item number of PROM. Same as *B mode seventh window. Section 1.6.

ObjSrlNo – Object code serial number of PROM. Same as *B mode eighth window. Section 1.6.

Like the InLocs table, a new Status record is always available when the datalogger is checked for data. For this reason the Status table is usually collected for display rather than archive.

InLocs
TmStamp, RecNbr, Flags, Ports, Loc1, Loc2, Loc3, Loc4, Loc5, Loc6, Loc7, Loc8, Loc9

Like the Status table, a new InLocs record is always available when the datalogger is checked for data. For this reason these tables are usually collected for display rather than archive.

By default only nine Input Location are labeled. The default InLocs field labels can be replaced with user created labels. These labels are created with EDLOG technique for Input Location labels. Press CTRL-L when editing a LOC field.

When additional data tables are created with Instruction 84, these tables and the fields they contain can also be named by the user. The datalogger will supply a default name for tables and fields if they are not named. The default names are T01, T02, for the tables and F01, F02, etc. for the fields, numbered in the order they appear. Edlog allows the programmer to name the tables and fields.

The Timestamp and record number labels are added automatically.

2.2 DATA OUTPUT FORMAT AND RANGE LIMITS

Data is stored internally in Campbell Scientific's Binary Final Storage Format (Appendix C.2). Data may be sent to Final Storage in either LOW RESOLUTION or HIGH RESOLUTION format.

2.2.1 RESOLUTION AND RANGE LIMITS

Low resolution data is a 2 byte format with 4 significant digits and a maximum magnitude of +7999. High resolution data is a 4 byte format (see Section 2.2.2).

TABLE 2.2-1. Resolution Range Limits of CR10 Data

Resolution	Zero	Minimum Magnitude	Maximum Magnitude
Low	0.000	+ 0.001	+7999.
High	0.0000	$1x10^{-19}$	$+9x10^{+18}$

The resolution of the low resolution format is reduced to 3 significant digits when the first (left most) digit is 8 or greater. Thus, it may be necessary to use high resolution output or an offset to maintain the desired resolution of a measurement. For example, if water level is to be measured and output to the nearest 0.01 ft., the level must be less than 80 ft. for low resolution output to display the 0.01 ft. increment. If the water level was expected to range from 50 to 90 feet the data could either be output in high resolution or could be offset by 20 ft. (transforming the range to 30 to 60 ft.).

2.2.2 HIGH RESOLUTION FINAL STORAGE DATA, INPUT, AND INTERMEDIATE STORAGE DATA FORMAT

While low resolution output data have the limits described above, computations are done in floating point arithmetic. In high resolution Final Storage Input and Intermediate Storage, the numbers are stored and processed in a binary format with a 23 bit binary mantissa and a 6 bit binary exponent. The largest and smallest numbers that can be stored and processed are $9 x 10^{18}$ and $1 x 10^{-19}$, respectively. The size of the number determines the resolution of the

arithmetic. A rough approximation of the resolution is that it is better than 1 in the seventh digit. For example, the resolution of 97,386,9$\underline{2}$4 is better than 10. The resolution of 0.00867319$\underline{2}$4 is better than 0.000000001.

A precise calculation of the resolution of a number may be determined by representing the number as a mantissa between .5 and 1 multiplied by 2 raised to some integer power. The resolution is the product of that power of 2 and 2^{-24}. For example, representing 478 as $.9336 * 2^9$, the resolution is $2^9 * 2^{-24} = 2^{-15} =$ 0.0000305. A description of Campbell Scientific's floating point format may be found in Appendix C.

2.3 DISPLAYING STORED DATA ON KEYBOARD/DISPLAY *7 MODE.

The keyboard display (or the computer in Keyboard/Display mode) can be used to examine Data Storage Tables in Final Storage table data.

Key *7. The display will show: 07:nn where nn is the number of Data Storage Tables defined. Enter a table number (followed by the "A" key) to view that table. Tables are numbered in the order of the appearance of the Data Storage Table Instruction (84) in the datalogger program. Tables in the *1 program area are numbered first, followed by *2 and those in the *3 subroutines numbered last.

The display will then show the first field of the newest record in the table. If the display does not change, the select table has not had any data stored in it yet.

When a table is visualized the newest data record is at the bottom and the oldest is at the top. When the table is full and a new record is stored, the records shift up pushing the oldest record off the top and storing the new record at the bottom.

The display (or value displayed on the computer) can be thought of as a cursor, which can be moved up and down or right and left through the data. The display shows the field number to the left of the colon and the data value to the right. The keys used to move the display/cursor are summarized in the following table:

TABLE 2.3-1. *7 Mode Command Summary

KEY	ACTION
A	"Advances" along a record, when the end of the record is reached the 'cursor' advances to the first field in the next record.
B	"Backs" up along a record, wraps to the last element in the previous record
C	"Climbs" up the table, toward the oldest data, stops on oldest record.
D	"Drops" down the table, toward the newest data, stops on newest record.
#	Enter Time Mode to display timestamp. Enter new time values to jump to record.

Each record of data in a table has a time associated with it. Event data stores the time as part of the data, interval tables calculate the time based on the interval and a reference time. The time associated with each record consist of a year, month/day, hour/minute, and seconds value. Julian dates are not used. When viewing data in the *7 mode, event data records have time as part of the record. The field number on the keyboard/display does not change while viewing the four time values since time is considered a single field. Interval data does not contain time and it is not displayed as part of the record. To see the time values for a given record, press the # key while viewing any of the fields within the record.

The A and B key are used move through the four time values for the record. The C and D keys can be used to move to newer or older records and view the same time value of the new record. Pressing the # key again while viewing time or using the A or B keys to advance or back beyond the time values will return the display to the same field as was displayed when the time mode was entered.

The time field is displayed as: day.month, year, hr:min, seconds

While viewing time, entering new time values will allow the display to jump to the record with time values closest to those entered. The jump takes place when the time mode is left.

Interval tables do not store time as part of the record, but calculate the time. A table 10

records long is maintained inside Intermediate Storage to keep track of "holes" in the recorded data, due to watch dog errors or clock changes, so that the time of each record can be implicitly maintained.

This "hole" table rings around, and any recorded data that cannot be time stamped using this "hole" table cannot be displayed.

To view the "hole" table for a given table, Key *7. The display will show: 07:nn, where nn is the number of Data Storage Tables defined. Enter the number of table where you want to view the "holes."   Then press the C key followed by the A key. The Hole table fields are as follows:

01:time of hole;
02:number of holes.

SECTION 3. INSTRUCTION SET BASICS

Section 3.7.1 does not apply to the TD operating system which does not use Output Flag 0.

Table 3.8-1 Valid Flag Commands are 11 – 19 to set high and 21- 29 to set low. Because the TD operating system does not use Flag 0, Commands 10 and 20 are not valid with the TD operating system.

The following table replaces Table 3.10-1 for the TD operating system.

TABLE 3.10-1. Error Codes

Code	Type	Description
03	Editor	Program table full
04	Compile	Intermediate Storage full
05	Compile	Storage Area #2 not allocated
08	Run Time	CR10X reset by watchdog timer
09	Run Time	Insufficient Input Storage
10	Run Time	Low Battery Voltage
11	Editor	Attempt to allocate more Input or Intermediate Storage than is available
20	Compile	SUBROUTINE encountered before END of previous subroutine
21	Compile	END without IF, LOOP or SUBROUTINE
22	Compile	Missing END
23	Compile	Nonexistent SUBROUTINE
24	Compile	ELSE in SUBROUTINE without IF
25	Compile	ELSE without IF
26	Compile	EXIT LOOP without LOOP
27	Compile	IF CASE without BEGIN CASE
28	Compile	At compile time, no output specified after P84 or unable to automatically allocate at least one record per P84
29	Compile	Output table requests more memory than available
30	Compile	IF and/or LOOP nested too deep
31	Run Time	SUBROUTINES nested too deep
32	Compile	Instruction 3 and interrupt subroutine use same port
33	Compile	Cannot use Control Port 6 as counter or interrupt subroutine with Instruction 15 or SDM or SDI-12 input/output
40	Editor	Instruction does not exist
41	Editor	Incorrect execution interval
43	Compile	Time in Instruction 84 or 92 is not a multiple of execution interval (note: time is in seconds)
44	Compile	Loop (P87) cannot contain Data Table (P84)
60	Compile	Insufficient Input Storage
61	Compile	Burst Measurement Scan Rate too short
62	Compile	N<2 in FFT
68	Compile	Instruction 118 without enough Instructions 68 or 63
80	Compile	Illegal Interval in P193
81	Compile	Illegal Node ID in P193
82	Compile	Illegal Reps in P193
92	Compile	Instruction 92, intervals in seconds: Time into Interval > 59 or Interval > 60

94	Program Transfer	Program Storage Area full
95	Program Transfer	Program does not exist in Flash memory
96	Program Transfer	Addressed device not connected
97	Program Transfer	Data not received within 30 seconds
98	Program Transfer	Uncorrectable errors detected
99	Program Transfer	Wrong file type or editor error

THIS SECTION ENTIRELY REPLACES THE CR10X MANUAL SECTION 8.

SECTION 8. PROCESSING AND PROGRAM CONTROL EXAMPLES

This section contains examples for the CR10X. The appropriate voltage range codes would have to be selected for the CR23X (see CR23X Manual Section 8 for the measurement instructions). The CR510-TD may not support all the examples.

The following examples are intended to illustrate the use of Processing and Program Control Instructions, flags, and the capability to direct the results of Output Processing Instructions to Input Storage.

The specific examples may not be as important as some of the techniques employed, for example:

> *Directing Output Processing to Input Storage is used in the Running Average and Rainfall Intensity examples (8.1 and 8.2).*

> *Flag tests are used in the Running Average, Interrupt Subroutine, and Converting Wind Direction (8.1, 8.5, and 8.7).*

> *Control ports and the Loop are illustrated in the AM32 example (8.3).*

As in Section 7 these examples are not complete programs to be taken verbatim. They need to be altered to fit specific needs.

8.1 COMPUTATION OF RUNNING AVERAGE

It is sometimes necessary to compute a running average (i.e., the average covers a fixed number of samples and is continuously updated as new samples are taken). Because the output interval is shorter than the averaging period, Instruction 71 cannot be used; the algorithm for computing this average must be programmed by the user. The following example demonstrates a program for computing a running average.

In this example, each time a new measurement is made (in this case a thermocouple temperature) an average is computed for the 10 most recent samples. This is done by saving all 10 temperatures in contiguous input locations and using the Spatial Average Instruction (51) to compute the average. The temperatures are stored in locations 11 through 20. Each time the table is executed, the new measurement is stored in location 20 and the average is stored in location 2. The Block Move Instruction (54) is then used to move the temperatures from locations 12 through 20 down by 1 location; the oldest measurement (in location 11) is lost when the temperature from location 12 is written over it.

Input Location Labels:

1:Panl Temp	15:Temp_i5	11:Temp_i9	17:Temp_i3
2:smpl10av	16:Temp_i4	12:Temp_i8	18:Temp_i2
		13:Temp_i7	19:Temp_i1
		14:Temp_i6	20:Temp_i

Where i is current reading, i1 is previous reading, etc.

*	1	Table 1 Programs
01:	10	Sec. Execution Interval
01:	P17	Module Temperature
01:	1	Loc [:Panl_Temp]
02:	P14	Thermocouple Temp (DIFF)
01:	1	Rep
02:	1	2.5 mV slow Range
03:	1	IN Chan
04:	1	Type T (Copper-Constantan)
05:	1	Ref Temp Loc Panl_Temp
06:	20	Loc [:Tempi]
07:	1	Mult
08:	0	Offset
03:	P51	Spatial Average
01:	10	Swath
02:	11	First Loc Temp_i9
03:	2	Avg Loc [:smpl10avg]
04:	P54	Block Move
01:	9	No. of Values
02:	12	First Source Loc Temp_i8
03:	1	Source Step
04:	11	First Destination Loc [:Temp_i9]
05:	1	Destination Step

05:	P84	Data Table
01:	0	Seconds into interval
02:	0	Every time
03:	0	Records (0=auto; -=redirect)
06:	P70	Sample
01:	1	Reps
02:	2	Loc smpl10avg
07:	P	End Table 1

In the above example, all samples for the average are stored in input locations. This is necessary when an average must be output with each new sample. In most cases, averages are desired less frequently than sampling. For example, it may be necessary to sample some parameter every 5 seconds and output every hour an average of the previous three hours' readings. If all samples were saved, this would require 2160 input locations. The same value can be obtained by computing an hourly average and averaging the hourly averages for the past three hours. To do this requires that hourly averages be stored in input locations.

Instruction 84 is used to send the 1 hour average to Input Storage and again to send the 3 hour average to Final Storage.

Input Location Labels:

    1:AVG_i2
    2:AVG_i1
    3:AVG_i
    4: AVG_3_HR
    5:XX_mg_M3

*	1	Table 1 Programs
01:	5	Sec. Execution Interval
01:	P2	Volt (DIFF)
01:	1	Rep
02:	25	2500 mV 60 Hz rejection Range
03:	3	IN Chan
04:	5	Loc [:XX_mg_M3]
05:	10	Mult
06:	0	Offset
02:	P84	Data Table
01:	0	Seconds into interval
02:	3600	Seconds interval
03:	-3	Records (0=auto; -=redirect)
03:	P71	Average
01:	1	Rep
02:	5	Loc XX_mg_M3
04:	P51	Spatial Average
01:	3	Swath

02:	1	First Loc AVG_i2
03:	4	Avg Loc [:Avg_3_HR]
05:	P84	Data Table
01:	0	Seconds into interval
02:	3600	Seconds interval
03:	0	Records (0=auto; -=redirect)
06:	P70	Sample
01:	1	Reps
02:	4	Loc Avg_3_HR
07:	P92	If time is
01:	0	seconds into a
02:	3600	second interval
03:	30	Then Do
08:	P54	Block Move
01:	2	No. of Values
02:	2	First Source Loc AVG_i1
03:	1	Source Step
04:	1	First Destination Loc [:AVG_i2]
05:	1	Destination Step
09:	P95	End
10:	P	End Table 1

8.2 RAINFALL INTENSITY

In this example, the total rain for the last 15 minutes is output only if any rain has occurred. The program makes use of the capability to direct the output of Output Processing Instructions to Input Storage.

Every 15 minutes, the total rain is sent to Input Storage. If the total is not equal to 0, output is redirected to Final Storage Area 1, the time is output and the total is sampled.

Input Location Labels:

    1:Rain_mm
    2: TOT_15mm

*	1	Table 1 Programs
01:	60	Sec. Execution Interval
01:	P3	Pulse
01:	1	Rep
02:	1	Pulse Input Chan
03:	2	Switch closure
04:	1	Loc [:Rain_mm]
05:	.254	Mult
06:	0	Offset
02:	P86	Do
01:	21	Set low Flag 1

03:	P92	If time is
01:	0	seconds into a
02:	900	second interval
03:	11	Set high Flag 1
04:	P84	Data Table
01:	0	Seconds into interval
02:	-1	When flag 1 is high
03:	-2	Records (0=auto; -=redirect)
05:	P72	Totalize
01:	1	Rep
02:	1	Loc Rain_mm
06:	P89	If X<=>F
01:	2	X Loc TOT_15min
02:	1	=
03:	0	F
04:	21	Set low Flag 1
07:	P84	Data Table
01:	0	Seconds into interval
02:	-1	When flag 1 is high
03:	0	Records (0=auto; -=redirect)
08:	P70	Sample
01:	1	Reps
02:	2	Loc TOT_15min
09:	P	End Table 1

8.3 USING CONTROL PORTS AND LOOP TO RUN AM416 MULTIPLEXER

This example uses an AM416 to measure 16 copper-constantan thermocouples and 16 Model 223 soil moisture blocks. The sensors are read every ten minutes and the average value output once an hour. The multiplexer is housed in an AM-ENCT enclosure to minimize thermocouple errors created by thermal gradients. A 107 Temperature Probe is centrally located on the multiplexer board and used as a thermocouple temp. reference.

The AM416 switches the 223 moisture block out of the circuit when it is not being measured. This eliminates the need for the blocking capacitors used in the model 227 Soil Moisture Block. The 223 blocks are about one fifth the cost of the 227 blocks.

Control ports are used to reset the AM416 and clock it through its channels. The program sequence is:

Measure the 107 probe located at the AM416 for TC temperature reference.

CR10 sets the port high which resets the AM416.

A loop is entered; within each pass:
The port clocking the AM416 is pulsed.
The connected TCs and moisture blocks are measured.

CR10 sets the port controlling AM416 reset low.

Soil moisture measurements are converted to block resistances.

The input location in which the temperature and soil moisture measurements are stored is indexed to the loop counter (Instruction 87, Section 12). An indexed location is incremented by one with each pass through the loop. For example, on the first pass temperature is stored in Location 2, and soil moisture in Location 18. On the second pass temperature is stored in Location 3, and soil moisture in Location 18. After 16 loop passes, temperature and soil moisture measurements occupy Locations 2 through 17 and 18 through 33, respectively.   Connections are shown in Figure 8.3-1.
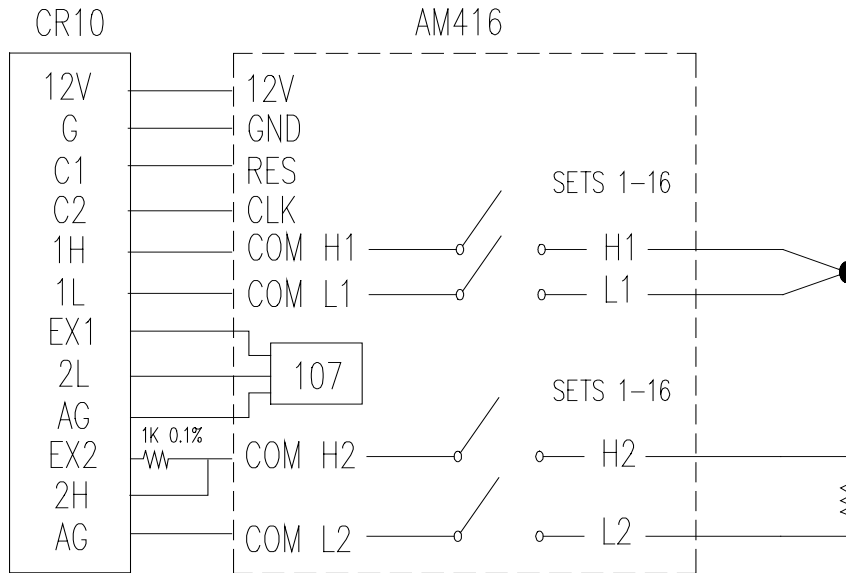
FIGURE 8.3-1. AM416 Wiring Diagram For Thermocouple and Soil Moisture Block Measurements

EXAMPLE PROGRAM MULTIPLEXING
THERMOCOUPLES AND SOIL MOISTURE
BLOCK

*	1	Table 1 Programs
01:	600	Sec. Execution Interval
01:	P11	Temp 107 Probe
01:	1	Rep
02:	4	IN Chan
03:	1	Excite all reps w/EXchan 1
04:	1	Loc [:REF_TEMP]
05:	1	Mult
06:	0	CR10X
02:	P86	Do
01:	41	Set high Port 1
03:	P87	Beginning of Loop
01:	0	Delay
02:	16	Loop Count
04:	P86	Do
01:	72	Pulse Port 2
05:	P14	Thermocouple Temp (DIFF)
01:	1	Rep
02:	21	2.5 mV 60 Hz rejection Range
03:	1	IN Chan
04:	1	Type T (Copper-Constantan)
05:	1	Ref Temp Loc REF_TEMP
06:	2--	Loc [:TC_TEMP_1]
07:	1	Mult
08:	0	Offset

06:	P5	AC Half Bridge
01:	1	Rep
02:	14	250 mV fast Range
03:	3	IN Chan
04:	1	Excite all reps w/EXchan 1
05:	250	mV Excitation
06:	18--	Loc [:SOIL_M_1]
07:	1	Mult
08:	0	Offset
07:	P95	End
08:	P86	Do
01:	51	Set low Port 1
09:	P59	BR Transform Rf[X/(1-X)]
01:	16	Reps
02:	18	Loc [:SOIL_M_1]
03:	.1	Multiplier (Rf)
10:	P84	Data Table
01:	0	Seconds into interval
02:	3600	Seconds interval
03:	0	Records (0=auto; -=redirect)
11:	P71	Average
01:	33	Reps
02:	1	Loc REF_TEMP
12:	P	End Table 1

8.4 INTERRUPT SUBROUTINE USED TO COUNT SWITCH CLOSURES (RAIN GAGE)

Subroutines given the label of 97 or 98 will be executed when control ports 7 or 8, respectively, go high (5 V, see Instruction 85, Section 12). In this example, Subroutine 98 and control port 8 are substituted for a pulse counting channel to count switch closures on a tipping bucket rain gage.

The subroutine adds 0.254 (mm, bucket calibrated for 0.01 inch tip) to an input location and uses Instruction 22 to delay 0.2 seconds.

The delay is to insure that any switch bouncing (when closing, the contacts actually bounce off each other, making and breaking the circuit several times) has died out before the subroutine is completed. (The pulse count inputs do this automatically.) Without the delay, the subroutine could be completed and called again by a bounce, causing false counts. The interrupt has no effect while the subroutine is still being executed.

Subroutine 98 is in effect keeping a running total in Input Storage. On the output interval, this total is sampled to Final Storage and zeroed by the program in Program Table 1.

To provide comparison, this example has the 2 pulse inputs also reading rain gages. (In a real situation, it is more likely that the pulse counters would be used for 2 wind speeds.) In Program Table 1, the 2 normal pulse inputs are read and the hourly totals output to Final Storage with Instruction 72.

The rain gage is connected as diagrammed below. When the switch closes, 5 volts is applied to port 8 which causes the subroutine to be executed.
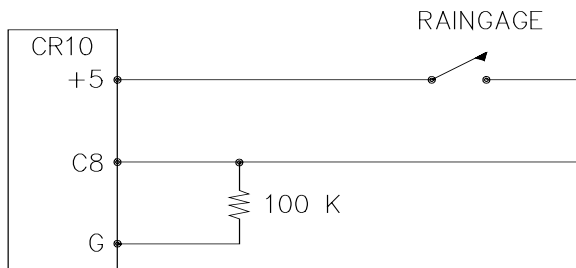
FIGURE 8.4-1. Connections for Rain Gage

Input location Assignments:

10:Rain_1	(from Pulse count)	
11:Rain_2	(from Pulse count)	
12:Rain_3	(from subroutine 98 while Output Flag is low)	

*	1	Table 1 Programs
01:	10	Sec. Execution Interval
01:	P3	Pulse
01:	2	Reps
02:	1	Pulse Input Chan
03:	2	Switch closure
04:	10	Loc [:Rain_1]
05:	.254	Mult
06:	0	Offset
02:	P84	Data Table
01:	0	Seconds into interval
02:	3600	Seconds interval
03:	0	Records (0=auto; -=redirect)
03:	P72	Totalize
01:	2	Reps
02:	10	Loc Rain_1
04:	P70	Sample
01:	1	Reps
02:	12	Loc Rain_3
05:	P92	If time is
01:	0	seconds into a
02:	3600	second interval
03:	30	Then Do
06:	P30	Z=F
01:	0	F
02:	0	Exponent of 10
03:	12	Z Loc [:Rain_3]
07:	P95	End
08:	P	End Table 1
*	3	Table 3 Subroutines
01:	P85	Beginning of Subroutine
01:	98	Subroutine Number
02:	P	34 Z=X+F
01:	12	X Loc Rain_3
02:	.254	F
03:	12	Z Loc [:Rain_3]
03:	P22	Excitation with Delay
01:	1	EX Chan
02:	0	Delay w/EX (units=.01sec)
03:	20	Delay after EX (units=.01sec)
04:	0	mV Excitation
04:	P95	End
05:	P	End Table 3

8.5 SDM-A04 ANALOG OUTPUT MULTIPLEXER TO STRIP CHART

This example illustrates the use of the SDM-A04 4 Channel Analog Output Multiplexer to output 4 analog voltages to strip chart.

While of questionable value because of current requirements and strip chart reliability, some archaic regulations require strip chart backup on weather data. The SDM-A04 may be used with the CR10 to provide analog outputs to strip charts. The output values in this example are wind speed, wind direction, air temperature, and solar radiation.

Instruction 103 is used to activate the SDM-A04. The 4 millivolt values to output must be stored in adjacent Input Storage locations, the first of which is referenced in Instruction 103.

The following program measures the sensors every 5 seconds. The readings are moved to another 4 locations and scaled to a 0 to 1000 millivolt output for the SDM-A04. Wind direction is changed from a 0-360 degree input to output representing 0 to 540 degrees. This conversion is done in a subroutine which is described in the next example.

The example also includes instructions to output wind vector and average temperature and solar radiation every hour.

Input Location Labels:

    1:WS
    2: WD_360
    3:TEMP_F
    4:Solar_Rad
    5:WS_output
    6:WD540_out
    7:TEMP_out
    8:SR_out
    10:WD_540

*	1	Table 1 Programs
01:	5	Sec. Execution Interval

01:	P3	Pulse
01:	1	Rep
02:	1	Pulse Input Chan
03:	22	Switch closure; Output Hz.
04:	1	Loc [:WS]
05:	1.789	Mult
06:	1	Offset

02:	P4	Excite,Delay,Volt(SE)
01:	1	Rep
02:	14	250 mV fast Range
03:	1	IN Chan
04:	1	Excite all reps w/EXchan 1
05:	5	Delay (units .01sec)
06:	1000	mV Excitation
07:	2	Loc [:WD_360]
08:	.7273	Mult
09:	0	Offset

03:	P11	Temp 107 Probe
01:	1	Rep
02:	2	IN Chan
03:	2	Excite all reps w/EXchan 2
04:	3	Loc [:Temp_F]
05:	1.8	Mult
06:	32	Offset

04:	P1	Volt (SE)
01:	1	Rep
02:	2	7.5 mV slow Range
03:	3	IN Chan
04:	4	Loc [:Solar_Rad]
05:	.14493	Mult
06:	0	Offset

05:	P54	Block Move
01:	4	No. of Values
02:	1	First Source Loc WS
03:	1	Source Step
04:	5	First Destination Loc [:WS_output]
05:	1	Destination Step

06:	P86	Do
01:	1	Call Subroutine 1

07:	P53	Scaling Array (A*loc +B)
01:	5	Start Loc [:WS_output]
02:	10	A1 Scale WS, 0-100 MPH = 0-1000 MV
03:	0	B1
04:	1.8519	A2 Scale WD, 0-540 DEG = 0-1000 MV
05:	0	B2
06:	5.7143	A3 Scale TEMP, -25 - 100 F = 0-1000 MV
07:	25	B3
08:	1000	A4 Scale RADIATION, 0-1KW/M^2 = 0-1000 MV
09:	0	B4

08:	P103	SDM-A04
01:	4	Reps
02:	30	Address
03:	5	Loc WS_output

09:	P84	Data Table
01:	0	Seconds into interval
02:	3600	Seconds interval
03:	0	Records (0=auto; -=redirect)

10:	P69	Wind Vector
01:	1	Rep
02:	180	Samples per sub-interval
03:	0	Polar Sensor/(S, D1, SD1)
04:	1	Wind Speed/East Loc WS
05:	2	Wind Direction/North Loc WD_360
11:	P71	Average
01:	2	Reps
02:	3	Loc Temp_F
12:	P	End Table 1

8.6 CONVERTING 0-360 WIND DIRECTION OUTPUT TO 0-540 FOR STRIP CHART

If 0-360 degree wind direction is output to a strip chart the discontinuity at 0/360 will cause the pen to jump back and forth full scale when the winds are varying from the north. In the days of strip charts this was solved with a 0-540 degree pot on the wind vane (direction changes from 540 to 180 and from 0 to 360 so the pen only jumps once when the wind is out of the north or south).

When faced with the necessity of strip chart output (see previous example), the following algorithm can be used to change a 0-360 degree input to 0-540. (If you have a 0-540 pot, it can be used with the CR10 since the Wind Vector Instruction, 69, will work with this output.)

To change 0-360 degrees to the 0-540 degrees, 360 degrees must sometimes be added to the reading when it is in the range of 0 to 180. The following algorithm does this by assuming that if the previous reading was less than 270, the vane has shifted through 180 degrees and does not need to be altered. If the previous 0-540 reading was greater than 270, 360 degrees is added.

This example is written as a subroutine, used by the previous example to output an analog voltage to a strip chart.

Input Location Labels:

    2:WD_360
    6:WD540_out
    10:WD_540

| * | 3 | Table 3 Subroutines |
| * | 3 | Table 3 Subroutines |

| 01: | P85 | Beginning of Subroutine |
| 01: | 1 | Subroutine Number |

02:	P89	If X<=>F
01:	10	X Loc WD_540
02:	3	>=
03:	270	F
04:	30	Then Do
03:	P86	Do
01:	11	Set high Flag 1
04:	P94	Else
05:	P86	Do
01:	21	Set low Flag 1
06:	P95	End
07:	P31	Z=X
01:	2	X Loc WD_360
02:	10	Z Loc [:WD_540]
08:	P89	If X<=>F
01:	10	X Loc WD_540
02:	4	<
03:	180	F
04:	30	Then Do
09:	P91	If Flag/Port
01:	11	Do if flag 1 is high
02:	30	Then Do
10:	P34	Z=X+F
01:	10	X Loc WD_540
02:	360	F
03:	10	Z Loc [:WD_540]
11:	P31	Z=X
01:	10	X Loc WD_540
02:	6	Z Loc [:WD540_out]
12:	P95	End
13:	P95	End
14:	P95	End
15:	P	End Table 3

8.7 LOGARITHMIC SAMPLING USING LOOPS

A ground water pump test requires that water level be measured and recorded according to the following schedule.

Time into Test, min			Output Interval	Loop #
00	to	10	10 sec.	1
10	to	30	30 sec.	2
30	to	100	1 min.	3
100	to	300	2 min.	4
300	to	1000	5 min.	5
1000	and greater		10 min.	6

This is accomplished with a series of loops (Instruction 87), where the delay and count parameters are used to implement the frequency of measurement (and output) and the duration of the that frequency. The unit of delay is the execution interval. A delay of 1 with a 10 second execution interval and a count of 60 means the instructions in the loop, in this case measure and output water level, are executed every 10 seconds for 10 minutes.

The drawdown portion of the test is completed at some time greater than 1000 minutes. To enter the recharge phase of the test, the operator enters the *6AD Mode and sets Flag 1 high. At the next 10 minute pass through loop 6 the loop is exited. Program execution returns to the top of the program table and the measurement schedule starts over again for the recharge test.

The sensor is a 50 PSI Druck, model 930/ti with a calibration of 49.93 mV/10V of excitation or 4.993mV/V. Your calibration will be different. An excitation voltage of 1500 mV yields a maximum signal of 7.489 mV at 50 PSI, fully utilizing the 7.5 mV Input Range to provide the best resolution.

The multiplier, m, is calculated to provide depth of water in feet:

$m = (50 \text{ psi}/4.993 \text{ mV/V}) * (2.3067 \text{ ft/psi})$

$m = 23.099 \text{ ft/mV/V}$

The offset is calculated to provide a final value that represents the distance from the lip of the well to the water surface. Similar to Figure 7.16-2, the offset equals the initial distance of 47.23 feet plus the initial reading of 54.77, or 102 feet.

*		1	Table 1 Programs
01:		10	Sec. Execution Interval

User must toggle Flag 1 to start measurements

01:	P91	If Flag/Port
01:	21	Do if flag 1 is low
02:	0	Go to end of Program Table

Loop 1, Output every 10 seconds for 10 minutes

02:	P87	Beginning of Loop
01:	1	Delay
02:	60	Loop Count

03:	P86	Do
01:	1	Call Subroutine 1

04:	P95	End

Loop2, Output every 30 seconds for 20 minutes

05:	P87	Beginning of Loop
01:	3	Delay
02:	40	Loop Count

06:	P86	Do
01:	1	Call Subroutine 1

07:	P95	End

Loop 3, Output every 1 minute for 70 minutes

08:	P87	Beginning of Loop
01:	6	Delay
02:	70	Loop Count

09:	P86	Do
01:	1	Call Subroutine 1

10:	P95	End

Loop 4, Output every 2 minutes for 200 minutes

11:	P87	Beginning of Loop
01:	12	Delay
02:	100	Loop Count

12:	P86	Do
01:	1	Call Subroutine 1

13:	P95	End

Loop 5, Output every 5 minutes for 700 minutes

14:	P87	Beginning of Loop
01:	30	Delay
02:	140	Loop Count

15:	P86	Do
01:	1	Call Subroutine 1

16:	P95	End

*Loop 6, Output every 10 minutes until stopped
by user*

17:	P87	Beginning of Loop
01:	60	Delay
02:	0	Loop Count
18:	P86	Do
01:	1	Call Subroutine 1
19:	P91	If Flag/Port
01:	21	Do if flag 1 is low
02:	31	Exit Loop if true
20:	P95	End
21:	P	End Table 1

*	3	Table 3 Subroutines
01:	P85	Beginning of Subroutine
01:	1	Subroutine Number
02:	P6	Full Bridge
01:	1	Rep
02:	22	7.5 mV 60 Hz rejection Range
03:	1	IN Chan
04:	1	Excite all reps w/EXchan 1
05:	1500	mV Excitation
06:	1	Loc [:LEVEL_FT]
07:	.46199	Mult
08:	102	Offset
03:	P84	Data Table
01:	0	Seconds into interval
02:	0	Every time
03:	0	Records (0=auto; -=redirect)
04:	P70	Sample
01:	1	Reps
02:	1	Loc LEVEL_FT
05:	P95	End
06:	P	End Table 3

SECTION 9. INPUT/OUTPUT INSTRUCTIONS

*** 18 MOVE TIME TO INPUT LOCATION ****

FUNCTION

This instruction takes current time or date information and does a modulo divide (see Instruction 46) on the time/date value with the number specified in the second parameter. The result is stored in the specified Input Location.

Entering 0 or a number greater than the maximum value of the time/date for the modulo divide will result in the actual time/date value being stored.

PARAMETER 1 CODES

Code	Time/Date Units
00	Seconds into day (maximum 86400)
01	Minutes into day (maximum 1440)
02	Hours into year (maximum 8784)
03	Hours into day (maximum 24)
04	Day of month (maximum 31)
05	Month of year (maximum 12)
06	YR MO DAY HR MIN SEC

PARAM NUMBER	DATA TYPE	DESCRIPTION
01:	2	Time/Date Code
02:	4	Number to modulo divide by
03:	4	Input Location Number

Input Locations altered: 1

SECTION 11. OUTPUT PROCESSING INSTRUCTIONS

Instructions 73 – Maximum and 74 – Minimum have only one time option. (Time is output as a quoted string.) Instruction 80 – Set Active Storage Area, is not in the TD operating system. Its functions are included in Instruction 84 – Data Table. Instruction 84 is only in the TD operating system.

*** 73 MAXIMUM ***

FUNCTION
This instruction stores the MAXIMUM value taken (for each input location specified) over a given output interval. An internal FLAG is set whenever a new maximum value is seen. This FLAG may be tested by Instruction 79. Time of occurrence maximum value(s) is OPTIONAL output information, which is formatted and activated by entering one of the following CODES for Parameter no. 2.

CODE	OPTIONS
00	Output value ONLY
01	Output value with TIME

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Time of maximum (optional)
03:	4	Starting input location no.

Outputs Generated: 1 for each input location (plus 1 with time of max. option)

*** 74 MINIMUM ***

FUNCTION
Operating in the same manner as Program 73, this instruction is used for storing the MINIMUM value (for each input location specified) over a given output interval.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Time of minimum (optional)
03:	4	Starting input location no.

Outputs Generated: 1 for each input location (plus 1 with time of min. option)

*** 84 DATA TABLE ***

FUNCTION
Instruction 84 is used to define a table of final storage data. New records of data are stored in the table based on time (interval data) or when a user flag (CR10X flags 1-8) is set (event data). Time based output intervals are specified in seconds. Fractional values may be used following the same rules that apply for the table scan rate.

PARAM. NUMBER	DATA TYPE	DESCRIPTION
01:	FP	Time into interval (Seconds)
02:	FP	Time interval (Seconds) 0 = Store new record each time. -a = Store if user flag "a" is set (a=1..8)
03:	FP	Number of records in table 0 = automatically allocates number of records. If 0 and parameter 2 = -a (event driven) or if 0 and parameter 2 = 0 then allocates size as if a 1 second output table -x = redirect to input loc x

Parameter 1 specifies how many seconds into the interval, specified in parameter 2, data will be output to final storage. If the output is event driven, enter a 0 for parameter 1.

Parameter 2 specifies the output interval (seconds) for the table. If output is determined by a flag, -a should be used, ('a' is the flag to be tested). If Parameter 2 = 0, data will be output unconditionally each time Instruction 84 is executed.

Parameter 3 specifies how many records will be stored in the table. When the table is full, subsequent new records will overwrite the oldest

records. If 0 is entered, records will be automatically allocated such that all automatic tables will be filled at the same time. If some tables specify the number of records and some tables are automatically allocated, the specified records will be allocated first, and then the remaining memory will be divided among the automatically allocated tables such that they will be filled at the same time. The Star 9 mode gives the size of all tables and the period (in days) before any automatically allocated table fills. If -x is entered, the processed results are returned to input locations.

Section 12. Program Control Instructions

The TD operating system does not use the output Flag 0. Commands dealing with it are not valid.

Instructions 96 – Serial Output, 98 – Send Character, and 111 – Load Program from Flash, are NOT in the TD operating system.

The instructions described in this section are only in the PakBus operating system.

Wireless Networks

More recent CR10X, CR510, and CR23X dataloggers with the PakBus operating system use the PakBus communications protocol. In addition to providing a robust means of packet based communication, the protocol allows transfer of input location data from one datalogger to another, or from a Campbell Scientific wireless sensor/datalogger (CR200 series datalogger) to a "host" (or "master") TD datalogger.

The following is some general information on the requirements for successful communication in a PakBus network.

Datalogger Requirements

PakBus communication requires the current PakBus operating system in the CR10X, CR510, or CR23X datalogger. All CR200 Series dataloggers are capable of PakBus communication.

All dataloggers in the PakBus network require a unique address. The address for TD dataloggers is set in the *D15 mode. The address for the CR200 datalogger's is set using Pakcom software or LoggerNet version 2.1 or greater.

Communication Notes

PakBus dataloggers are also capable of Modbus communication. The Modbus packet can ride on top of the PakBus packet or can be send independently. Instruction 190 is used to set up a datalogger as a Modbus master device. The datalogger's Modbus address is set in the *D8 mode of the datalogger. The Modbus and Pakbus addresses for a datalogger cannot be the same.

In order to communicate to a datalogger via another datalogger from LoggerNet's Connect window (e.g., communicating with a CR205 via a CR10XTD-PB) the datalogger through which you will be communicating must be set as a router. This is done by entering the number of remote dataloggers that the router will be connecting to in the *D15 mode, parameter 2.

The PakBus instructions described in this section are often used in combination with a CR200 series datalogger and the RF400 series spread spectrum radio. Table 12-1 lists some of the advantages and disadvantages of different methods of transferring data.

TABLE 12-1. CR205/CR210/CR215 in PakBus Network			
	Stand Alone Datalogger	**SendGetData P190**	**Wireless Sensor P193**
General Description	CR205 is programmed as a stand alone datalogger. Data are stored in datalogger and retrieved by computer running Loggernet	May be used as either a wireless sensor interface (data stored in "Master" Logger) or to transfer data to another logger while still independently storing data.	"Master" datalogger stores all data and sets transmission schedule for remote CR205 w/ sensors.
PRO	• Data are stored in Datalogger, loggernet will automatically retry if communication fails. • No special datalogger programming required for data transfer	• Most flexible datalogger to datalogger data transfer. • Can be set to automatically retry sending data.	• Potential for lowest power consumption by remote CR205. • Data are collected only from "Master" datalogger. • Automatic retries if adequate time is allocated
CON/limitations	• Separate data files for each logger.	• Both the "Remote Sensor" CR205 and the Master Datalogger need to be programmed to handle the data. • Data collection should be scheduled to avoid conflicts with datalogger to datalogger communication.	• Both the "Remote Sensor" CR205 and the Master Datalogger need to be programmed to handle the data. • Without storing data in the sensor, data are lost if transmission fails. • Master radio must be on continuously. • For simplicity, all wireless sensor CR205s should have the same sensors. (or at least no more than 4 different sensor configurations) • Data collection should be scheduled to avoid conflicts with Sensor to Master communication.
PakBus Instructions used in Datalogger Programming	Normal programming to measure sensors, process, and store data.	SetValue, GetValue, or P190 in datalogger initiating communication. Programming to deal with any sent data or codes in the other datalogger.	Wireless Network Master (P193) in master datalogger. TimeUntilTransmit (or P194) and SendGetData (or P196) in remote.

		Stand Alone Datalogger	SendGetData P190	Wireless Sensor P193
Radio Settings		*Radio address, net address, and hop sequence must be the same in all CR2xxs and RF400s in the network. Because only one header length can be set for a radio, only one power cycling interval should be used in network; i.e., do not mix 8 second and 1 second interval radio power cycling. RF on (no cycling) is specified where necessary below.*		
	CR205 Power Mode and Header	Select power mode for appropriate response time and current drain.	Select power mode for appropriate response time and current drain. Header must match called station's cycle.	NO HEADER. RfpinEn (lowest power) or to allow communication with computer for checking station, downloading program: RF8_Sec or RF1_sec.
	RF400 on CR10X or CR23X	Router, if used, must send appropriate header	Router, if used, must send appropriate header	RF on, *header to match CR205 setting to allow contacting CR205*
	Beacon Interval	0	0	0
	RF400 on computer	RF on, *header to match neighbors' settings*	RF on, *header to match neighbors' settings*	RF on, *header to match neighbors' settings*
LoggerNet Settings *When RF400 with direct access to network is connected to computer.*		*For each datalogger:* Com -PbusPort -logger	*For each datalogger with stored data:* Com -PbusPort -logger	*For each master datalogger:* Com -PbusPort -logger

PakBus Get/Send Locations (P190)

A program control instruction that sends data to or retrieves data from another datalogger in a PakBus network. This instruction can also be used to issue commands to Modbus devices, where the datalogger acts as a Modbus master.

```
 1:  PakBus - Get/Send Locations (P190)
 1: 00      Port
 2: 0000    Address
 3: 00      Command
 4: 0000    Security
 5: 0000    Remote Loc/Coil/Register
 6: 0000    Swath
 7: 0000    Local Loc [ _____ ]
 8: 0000    Result Code Loc [ _____ ]
```

Notes:

Edlog allocates only one of the input locations used in parameters 5 and 7 of this instruction. The additional input locations must be inserted manually using the Input Location Editor.

If this instruction is used to retrieve a value or set a value in the remote datalogger's public (or input location) table (i.e., code 26 or 27 is used in parameter 3), Instruction 63 or 68 must follow this instruction to enter the variable name that will be accessed.

Index parameter 3 to delay execution of subsequent program instructions until the datalogger receives a valid response or error from the remote. Otherwise, the PakBus command is queued, the datalogger proceeds to the next instruction, and the communications are handled later when the remote replies.

Port

The communications port that will be used by the local PakBus datalogger during the execution of this instruction. Valid options are:

Code	Description
0	Modem Enabled Device, 300
1	Modem Enabled Device, 1200
2	Modem Enabled Device, 9600
3	Modem Enabled Device, 76800
4	Modem Enabled Device, 2400
5	Modem Enabled Device, 4800
6	Modem Enabled Device, 19200
7	Modem Enabled Device, 38400
16	SDC 6 (COM 310)
17	SDC 7
18	SDC 8

Note: The CR10X-TD and CR510-TD have limited communication rates and do not support options 4 through 7.

Address

When a PakBus command is being issued, this address refers to the PakBus address of the datalogger. When a Modbus command is being issued, this address refers to the Modbus address. The PakBus and Modbus addresses in the datalogger cannot be set to the same number.

If this instruction is used within a loop, index this parameter to automatically increment the address with each pass through the loop. When this parameter is indexed, program execution will be delayed until a response is received from the remote datalogger.

PakBus Communication

The unique address for the datalogger in the PakBus network that will be communicated with using this instruction.

The Pakbus address is set in the datalogger's *D15 mode.

Modbus Communication

The unique address for the datalogger in a Modbus network that will be communicated with using this instruction (the slave device).

The Modbus address is set in the datalogger's *D8 mode. The valid range of IDs for a Modbus slave device are 1 - 99. Setting the datalogger's Modbus address to 0 disables it as a Modbus slave.

Command

This parameter determines what type of communication should take place in the PakBus or Modbus network when the instruction is executed.

Command	Description
1	Read Coil Status (Modbus command)
2	Read Input Status (Modbus command)
3	Read Holding Registers (Modbus command)
4	Read Input Registers (Modbus command)
5	Force Single Coil (Modbus command)
15	Force Multiple Coils (Modbus command)
16	Preset Multiple Registers (Modbus command)
21	Receive input location data from another datalogger (Pakbus command)
22	Send input location data to another datalogger (Pakbus command)
26	Get Value
27	Set Value
61	Read Coil Status (Modbus command)
62	Read Input Status (Modbus command)
63	Read Holding Registers (Modbus command)
64	Read Input Registers (Modbus command)
65	Force Single Coil (Modbus command)
66	Force Multiple Coils (Modbus command)
67	Preset Multiple Registers (Modbus command)

Notes:

Codes 61 through 67 are used when the Modbus packet will ride on top of Pakbus as a datagram.

If Get Value or Set Value is used (26 or 27), parameter 5 is left blank and Instruction 63 or 68 is used following this instruction to enter the variable name in the datalogger's Public (or input locations) table that will be accessed.

If this parameter is indexed, the datalogger will proceed to the next instruction only after it receives a valid response or error from the remote. Otherwise, the PakBus command is queued, the datalogger proceeds to the next instruction, and the communications are handled later when the remote replies. It may be

desirable to delay execution of subsequent instructions if those instructions perform further processing on the response from the remote.

Security

Enter the level 2 security code for the remote datalogger in the PakBus network that will be communicated with using this instruction when Command 22 is used for parameter 3 (send input location data to another datalogger).

If the security code in this instruction does not match the security code of the remote datalogger, the remote datalogger will discard the message, and the failure will be indicated in the local datalogger by an incremental change in the Result Code Location (parameter 8).

If security is not set in the remote datalogger or if command 21 is used for parameter 3, this parameter can be left at 0.

For additional information on security codes, see Program Security.

Enter the level 2 security code for the remote datalogger in the PakBus network that will be communicated with using this instruction when Command 22 is used for parameter 3 (send input location data to another datalogger).

If the security code in this instruction does not match the security code of the remote datalogger, the remote datalogger will discard the message, and the failure will be indicated in the local datalogger by an incremental change in the Result Code Location (parameter 8).

If security is not set in the remote datalogger or if command 21 is used for parameter 3, this parameter can be left at 0.

Remote Location/Coil/Register

PakBus Communication

If data is being received from another datalogger in the PakBus network (Parameter 3 set to 21), this is the first input location in the remote datalogger from which to retrieve the data.

If data is being sent to another datalogger in the PakBus network (Parameter 3 set to 22), this is the first input location in the remote datalogger in which to store the first data value.

If a variable in the Public (or Input Locations) table is being accessed using Get Value or Set Value (Parameter 3 set to 26 or 27), this parameter is left blank and Instruction 63 or 68 is used following this instruction to enter the variable name.

Modbus Communication

This is the first coil or register to be acted upon when the instruction is executed.

For general information on input locations, see Input Locations.

Remote Location

PakBus Communication

If data is being received from another datalogger in the PakBus network (Parameter 3 set to 21), this is the first input location in the remote datalogger from which to retrieve the data.

If data is being sent to another datalogger in the PakBus network (Parameter 3 set to 22), this is the first input location in the remote datalogger in which to store the first data value.

Modbus Communication

This is the first coil or register to be acted upon when the instruction is executed.

Swath

PakBus Communication

The number of input locations that will be sent to or retrieved from the remote datalogger.

Modbus Communication

The number of subsequent coils or registers that will be acted upon when the instruction is executed.

Local Location

PakBus Communication

If data is being received from another datalogger in the PakBus network (Parameter 3 set to 21), this is the first input location in which to store the data.

If data is being sent to another datalogger in the PakBus network (Parameter 3 set to 22), this is the first input location for the swath of input locations that will be sent to the remote datalogger.

Notes:

If Command code 21 is chosen, the number of input locations required for the transferred data must be allocated manually.

If this instruction is used within a loop, index this parameter to automatically increment the input locations with each pass through the loop. The input locations for this parameter are calculated as the number of passes through the loop * the swath of locations, plus one location (for the response).

Modbus Communication

The input location that is the source or the destination of the information that will be transferred when this instruction is executed. Discrete values are packed

or unpacked with the least significant bit of the first byte, starting at this location. Incoming discrete values are set to -1.0 for ON and 0 for OFF. Outgoing discrete values are translated as 0.0 to OFF and non-zero to ON.

For general information on input locations, see Input Locations.

Result Code Location

The input location in which to store the results of the data transfer.

Result	Description
0	Successful
>0	Initial attempt failed (value indicates the number of retries)

Up to 2 retries will be attempted if data transfer fails. The retry interval is 1 second, plus 1/2 * number of hops for the node. The instruction runs in the background after initiated.

Note: If this instruction is used within a loop, index this parameter to automatically increment the input location in which the result is stored with each pass through the loop.

Send Final Storage Data (P191)

A program control instruction that transfers final storage data from one or more tables in a PakBus datalogger to a computer.

```
2:  PakBus - Send Final Storage Data (P191)
 1: 00       Port
 2: 0000     Address
 3: 0000     Table ID
 4: 00       Flag
```

Table ID

The ID for the data table that should be sent to the computer using this instruction. If the ID is set to 0, then all final storage tables will be transferred.

Flag

The user flag that will determine if the table definitions are transferred along with the data table(s). When the flag is high, the table definitions for the specified table(s) will be output as a separate data gram.

Send Message (P192)

A program control instruction that sends a message to another datalogger in the PakBus network.

This instruction can be used in a network with several Master dataloggers to synchronize all Master datalogger's clocks. One datalogger would use this instruction to periodically broadcast a clock report, and the remaining dataloggers would use instruction 195, Use Remote Clock Report, to set their clocks by the transmitted value.

This instruction is not necessary in networks with wireless sensors and only one Master datalogger, because the Wireless Network Master (P193) and Wireless Network Remote (P196) instructions perform these functions automatically.

This instruction can also be used to remove a datalogger from the PakBus network.

```
3:  PakBus - Send Message (P192)
 1: 00       Port
 2: 0000     Address
 3: 2        Clock Report
```

Message Type

Entry   **Description**
2   Clock report; sends the current time.
13   The datalogger that receives this message will remove the sending datalogger from its neighbor list (and therefore all links to the sending datalogger).

Wireless Network Master (P193)

A program control instruction that is used to prepare the local datalogger to send data to or receive data from one or more dataloggers/wireless sensors in a PakBus network. The instruction also assigns a transmission time to the remote dataloggers/wireless sensors. Instruction 193 does not actually initiate the transfer of data. Data transfer is initiated by the wireless sensor.

Multiple Instruction 193s can be used in a program to configure up to four different groups of dataloggers/wireless sensors. A "group" is determined by the First Remote Address and the Number of Remotes. A datalogger/wireless sensor can only belong to one group. An error message will occur (E81) if a datalogger/wireless sensor is assigned to more than one group.

```
 4:  PakBus - Wireless Network Master (P193)
 1: 00       Number of Remotes
 2: 0000     First Remote Address
 3: 0000     Time Into Transmit Interval (sec)
 4: 0000     Transmit Interval (sec, 0 = use execution interval)
 5: 00       Transmit Delay Between Remotes (sec)
 6: 00       Swath to Receive
 7: 0000     First Loc for Data Received [ _____ ]
 8: 00       Swath to Send
 9: 0000     First Loc to Send [ _____ ]
10: 0000     Result Code Loc [ _____ ]
```

Notes:

The wireless sensors will actually begin transmitting before the specified transmission time (based on Time Into Transmit Interval and Transmit Interval) so that transmission is complete when the specified transmission time occurs. The Transmit Delay Between Remotes is factored into to the transmit time assigned to each remote.

Edlog allocates only one of the input locations used in parameters 7, 9, and 10 of this instruction. The additional input locations must be inserted manually using the Input Location Editor. For information on manually inserting input locations, refer to Manually Inserting Input Locations Into Edlog.

Number of Remotes

The number of remote dataloggers/wireless sensors in the PakBus network that will be communicated with using this instruction.

First Remote Address

The unique address for the first remote datalogger/wireless sensor in the PakBus network that will be communicated with using this instruction. All of the remotes that will be communicated with using this P193 should have sequential addresses.

The address is set in the datalogger's *D15 mode (refer to the datalogger user's manual for additional information). It can be any number between 1 and 4095.

Time into Transmit Interval

An offset, in seconds, into the Transmit Interval. The valid range is 0 through 9998.

Example

To set up the remotes for an hourly transmission at 15 minutes past the hour, the Time into Transmit Interval would be set at 900 and the Transmit Interval would be set at 3600.

Transmit Interval

The transmission interval, in seconds, that will be assigned to the group of dataloggers/remote sensors being set up with this instruction. The valid range is 1 through 9999.

The wireless sensors will actually begin transmitting before the specified transmission time (based on Time Into Transmit Interval and Transmit Interval) so that transmission is complete when the specified transmission time occurs. The Transmit Delay Between Remotes, parameter 5, is factored into to the transmit time assigned to each remote.

The datalogger program can be written to execute P193 on every program scan, or within a P92 (If Time) instruction. The Transmit Interval must equal the interval on which Instruction 193 is being executed.

Note: The Transmit Interval must be sufficiently long so that all of the remotes have a chance to respond before the next transmit interval occurs. Therefore, Transmit Interval must be equal to or greater than Number of Remotes * Transmit Delay Between Remotes. You will have to estimate the Transmit Delay Between Remotes if that value (parameter 5) is set to 0 (which means use the estimated value from the datalogger's routing table).

Example

To set up the remotes for an hourly transmission at 15 minutes past the hour, the Time into Transmit Interval would be set at 900 and the Transmit Interval would be set at 3600.

Transmit Delay Between Remotes

The amount of delay, in seconds, between transmission from each remote. If this parameter is left at 0, the master datalogger will automatically assign the delay based on the routing table (usually about 3 seconds between remotes). Otherwise, a specific delay can be entered. A specific delay may be necessary for slow communication links.

Some communications links do not require a delay (such as when communicating over an NL100). In this instance, the parameter can be left at 0 and indexed.

Code	Description
0	Use the default, as determined by the routing table
>0	Use the value entered
--0	Use no delay (dashes are accomplished by indexing the parameter)

Note: The wireless sensors will actually begin transmitting before the specified transmission time (based on Time Into Transmit Interval and Transmit Interval) so that transmission is complete when the specified transmission time occurs. The Transmit Delay Between Remotes is factored into to the transmit time assigned to each remote.

Example

Assume 4 wireless remotes in a network, with the first having an address of 1 and the remainder of the remotes addressed consecutively. The transmission time is set at 900 seconds into a 3600 second interval (15 minutes past each hour). If Transmit Delay Between Remotes is set at 5, Remote 4 will transmit at about 15 seconds before the transmit time, Remote 3 at about 10 seconds before, Remote 2 at about 5 seconds, and Remote 1 at the transmit time.

Swath to Receive

The number of data values that will be received from each remote when data is transferred. If a remote sends less than the number of values indicated by the swath, the remaining locations will be filled with an overrange value (-99999). If a remote sends more than the number of data values indicated by the swath, the extra values will be discarded by the local datalogger.

First Location for Data Received

The first input location in which the first data value received from the first remote should be stored. Subsequent data values from the group of remotes will be stored in consecutive input locations.

Note: The number of input locations required for the transferred data (Number of Remotes * Swath to Receive) must be allocated manually.

For general information on input locations, see Input Locations.

Swath to Send

The number of data values that will be sent to each remote when data is transferred.

First Location to Send

The input location which holds the first value that should be sent to the dataloggers/wireless sensors in the group. The range of values sent to the remote(s) is determined by the Swath to Send parameter (parameter 8).

For general information on input locations, see Input Locations.

Result Code Location

The input location in which a code is stored to indicate the result of the data transfer. A 0 indicates the data transfer was successful; any number greater than 0 indicates a failure.

Note: One Result Code Location is required for each remote (specified by Number of Remotes, parameter 1). These input locations must be allocated manually.

When the datalogger receives a wireless message from a remote, the corresponding Result Code Location is set to -1. When Instruction 193 is executed, the Result Code Location is incremented by 1. Therefore, if communication is successful, the Result Code Location will be 0 after the execution of Instruction 193. If data transfer is unsuccessful, the Result Code Location for the remote that failed will be incremented, and will continue to increment with each failed attempt.

Seconds Until Transmit (P194)

A program control instruction that places in an input location the number of seconds until it is time to transmit data to the host datalogger. This instruction is used in conjunction with a conditional statement to determine when the Wireless Network Remote instruction (P196) is executed to initiate communication with the host. The communication schedule is determined by the host (or master) datalogger and is set in the remote when it first initiates communication. If no communication has taken place and the value has not been set, or if scheduled communication was not successful, Seconds Until Transmit will return a random offset into a one minute interval.

```
5:  PakBus - Seconds Until Transmit (P194)
 1: 0000     Loc with Seconds Until Transmit [ _____ ]
```

Note: If the datalogger is not being used as a wireless sensor (i.e., Instruction 196 is not in the program), this instruction can be used to place a random number of seconds into a minute interval in the specified input location. The random seed is based on the datalogger's PakBus address.

Location with Seconds Until Transmit

The input location in which to store the number of seconds until it is time to transmit to the host datalogger.

Use Remote Clock Report (P195)

A program control instruction that sets a remote datalogger's clock based on the clock value transmitted from the host (or master) datalogger specified by the address provided in parameter 1.

```
6:  PakBus - Use Remote Clock Report (P195)
 1: 0000      Address
```

Note: This instruction is not used if the datalogger is configured as a wireless sensor using instruction 196 (Wireless Network Remote). When instruction 196 is used, the remote datalogger will automatically adjust its clock to match the host (master) datalogger's clock whenever communication is successful.

Address

The unique address for the host (master) datalogger in the PakBus network, whose clock value will be used to set the clock in the remote datalogger.

The address is set in the datalogger's *D15 mode (refer to the datalogger user's manual for additional information). It can be any number between 1 and 4095.

Wireless Network Remote (P196)

A program control instruction that is used to set up a remote datalogger to act as a wireless sensor/controller in a PakBus network.

Communication with the host (master) datalogger is dictated by the host datalogger. A communication time is assigned to the remote datalogger when communication is first accomplished with the host. The remote datalogger uses Instruction 194, Seconds Until Transmit, in conjunction with a conditional statement to determine when P196 is executed, and therefore, when data is transferred to the host.

When first installed or when communication is not successful, P194 will indicate a random interval of up to one second. The remote will try to contact the master on this interval until communication is successful and it is programmed with a transmit time.

The remote datalogger's clock is synchronized with the host datalogger's clock, each time communication between the two dataloggers is successful.

```
7:  PakBus - Wireless Network Remote (P196)
 1: 00        Port
 2: 0000      Master Address
 3: 0000      Security
 4: 00        Swath to Receive from Master
 5: 0000      First Loc for Data Received [ _____ ]
 6: 00        Swath to Send to Master
 7: 0000      First Loc to Send [ _____ ]
 8: 0000      Result Code Loc [ _____ ]
```

Swath to Receive From Master

The number of data values that will be received from the host (master) datalogger when data is transferred. If the host sends less than the number of values indicated by the swath, the remaining locations will be filled with an overrange value (-99999). If the host sends more than the number of data values indicated by the swath, the extra values will be discarded by the local datalogger.

First Location for Data Received

The first input location in which the first data value received from the host (master) datalogger should be stored. Subsequent data values from the host will be stored in consecutive input locations.

Notes:

The number of input locations required for the transferred data must be allocated manually.

Security

Enter the level 2 security code for the master datalogger in the PakBus network that will be communicated with using this instruction .

If the security code in this instruction does not match the security code of the master datalogger, the master datalogger will discard the message, and the failure will be indicated in the local datalogger by an incremental change in the Result Code Location (parameter 8).

If security is not set in the master datalogger, this parameter can be left at 0.

For additional information on security codes, see Program Security.

Enter the level 2 security code for the master datalogger in the PakBus network that will be communicated with using this instruction .

If the security code in this instruction does not match the security code of the master datalogger, the master datalogger will discard the message, and the failure will be indicated in the local datalogger by an incremental change in the Result Code Location (parameter 8).

If security is not set in the master datalogger, this parameter can be left at 0.

Swath to Send

The number of data values that will be sent to the host (master) datalogger when data is transferred.

First Location to Send

The input location which holds the first value that should be sent to the host (master) datalogger. The range of values sent is determined by the Swath to Send parameter (parameter 6).

For general information on input locations, see Input Locations.

Result Code Location

The input location in which a code is stored to indicate the result of the data transfer. A 0 indicates the data transfer was successful; any number greater than 0 indicates a failure. A -2 indicates that communication was established with the datalogger at the specified address, but the datalogger was not programmed as a host (master) datalogger using Instruction 193. In this instance, a 0 is stored in parameter 5, First Location for Data Received.

For general information on input locations, see Input Locations.

Force Route (P197)

A program control instruction that is used to force the datalogger to use a specific route in the PakBus network to communicate with the destination datalogger. This information is set in the datalogger's routing table.

```
8:  PakBus - Force Route (P197)
 1: 00       Port
 2: 0000     Neighbor's Address
 3: 0000     Address
 4: 00       Hops
```

Note: For communications paths where there are multiple hops, this instruction fixes only the first hop.

Neighbor's Address

The address of the first hop (or repeater) in the PakBus network that the datalogger should use in communicating with the destination datalogger.

Hops

The number of hops (or repeaters) in the communications path to the destination datalogger.

Set Setting (P198)

A program control instruction that is used to set a setting in a PakBus datalogger. This instruction should be followed by instruction 63 or instruction 68 with the values for the setting that should be changed.

If the address in this instruction is set to the address of the datalogger executing the instruction, the datalogger will change its own setting.

```
9:  PakBus - Set Setting (P198)
 1: 00       Port
 2: 0000     Address
 3: 0000     Result Code Loc [ _____ ]
```

Result Location

Result Code	Description
-1001	The attempted setting is a read-only setting
-1002	Out of space in the remote
-1003	Syntax error
0	Success
>1	Number of communication failures

Routing Table Information (P199)

A program control instruction that is used to store the datalogger's routing table information in a series of input locations. This instruction is used most often as a trouble-shooting tool.

```
10:  PakBus - Routing Table Information (P199)
 1: 0000    First Loc [ _____ ]
```

Parameter 1 specifies the first input location in which to begin storing the information. For each route, there are 3 pieces of information returned:

1. The PakBus Address of the Destination datalogger.
2. The PakBus address of any datalogger used as a hop to the Destination datalogger.
3. The Response Metric, in seconds (each hop takes 1 second).

A -1 in an input location indicates the end of the routing table information.

Note: The input locations required for this instruction must be allocated manually.

Example

Consider the following routing table information:
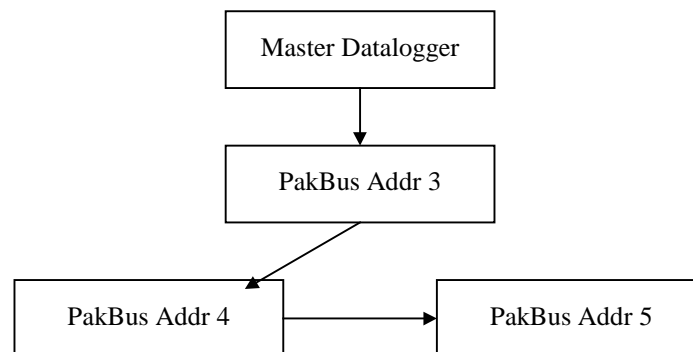
The information returned using this instruction would be similar to:

Input Location Used	Value Stored	Description
1	3	Address of destination datalogger
2	3	Address of repeater datalogger
3	1	Response metric, 1 second (1 hop)
4	4	Address of destination datalogger
5	3	Address of repeater datalogger
6	2	Response metric, 2 seconds (2 hops, 3 & 4)
7	5	Address of destination datalogger
8	3	Address of repeater datalogger (first hop only)
9	3	Response metric, 3 seconds (3 hops, 3, 4 & 5)
10	-1	End of string

PakBus Settings (Options | PakBus Settings)

This dialog box is used to configure some of the PakBus settings, that are normally set in the datalogger's *D mode, when a program is downloaded to the datalogger.

For all of the options below, if the check box Do Not Change Current Settings is enabled, then those settings will not be changed when the program is downloaded to the datalogger.

Network

The Network option is used to set the PakBus address in the datalogger and to configure the datalogger as a router if required. This option is the same as the datalogger's *D19 mode.

Address - Enter the PakBus address that should be assigned to the datalogger.

Maximum number of nodes - Enter the total number of dataloggers in the PakBus network.

Maximum number of neighbors - Enter the number of dataloggers in the PakBus network that the datalogger can communicate with directly (i.e., without going through another datalogger).

Maximum number of routers - Enter the number of neighbors to the datalogger that act as routers to one or more other dataloggers in the PakBus network.

Beacon Intervals

This option is used to set the interval on which the datalogger will transmit a beacon out a particular port to the PakBus network. Use the drop-down list box to select the port over which the beacon will be transmitted, and enter the

desired interval in the Communications Interval field. This option is the same as the datalogger's *D18 mode.

In some networks, a beacon interval might interfere with regular communication in the PakBus network (such as in an RF network), since the beacon is broadcast to all devices within range. In such cases, it may be more appropriate to use the Neighbor Filter instead, which broadcasts a beacon only to those dataloggers which it has not received communication from within a specified interval.

Neighbor Filter

This option allows you to list potential neighbors that are available to the datalogger in the PakBus network. The datalogger will attempt to issue a "hello" command to all the dataloggers listed in the neighbors filter list, and will transmit an expected communication interval. The communication interval is the interval on which the datalogger expects to receive communication from the neighbors. If communication is not received from a neighbor within 2.5 times this interval, then the datalogger will attempt to issue another "hello" command to that datalogger only (thus, creating less network traffic than the Beacon Interval).

The expected interval is entered into the Communication Interval field in seconds. The neighbors are defined by entering their addresses into the table. A range of addresses can be entered by using the Swath field. For example, entering 1 for the address and 5 for the swath will set up dataloggers with PakBus addresses 1, 2, 3, 4, and 5 as neighbors to the current datalogger.

This option is the same as the datalogger's *D19 mode.

Allocate General Purpose File Memory

PakBus dataloggers have the ability to store files transmitted from an NL100 in a general purpose memory area. This memory area is configured as ring memory. A value can be entered to specify the number of 64K blocks of memory that should be used for this purpose. Final storage memory will be reduce by the amount of memory specified in this option. This option is the same as the datalogger's *D16 mode.

CR23X INDEX

NOTE: The pages listed in this index will get you in the right section; however, the exact page number may be off.

Campbell Scientific Companies

Campbell Scientific, Inc. (CSI)
815 West 1800 North
Logan, Utah 84321
UNITED STATES
www.campbellsci.com
info@campbellsci.com

Campbell Scientific Africa Pty. Ltd. (CSAf)
PO Box 2450
Somerset West 7129
SOUTH AFRICA
www.csafrica.co.za
cleroux@csafrica.co.za

Campbell Scientific Australia Pty. Ltd. (CSA)
PO Box 444
Thuringowa Central
QLD 4812 AUSTRALIA
www.campbellsci.com.au
info@campbellsci.com.au

Campbell Scientific do Brazil Ltda. (CSB)
Rua Luisa Crapsi Orsi, 15 Butantã
CEP: 005543-000 São Paulo SP BRAZIL
www.campbellsci.com.br
suporte@campbellsci.com.br

Campbell Scientific Canada Corp. (CSC)
11564 - 149th Street NW
Edmonton, Alberta T5M 1W7
CANADA
www.campbellsci.ca
dataloggers@campbellsci.ca

Campbell Scientific Ltd. (CSL)
Campbell Park
80 Hathern Road
Shepshed, Loughborough LE12 9GX
UNITED KINGDOM
www.campbellsci.co.uk
sales@campbellsci.co.uk

Campbell Scientific Ltd. (France)
Miniparc du Verger - Bat. H
1, rue de Terre Neuve - Les Ulis
91967 COURTABOEUF CEDEX
FRANCE
www.campbellsci.fr
campbell.scientific@wanadoo.fr

Campbell Scientific Spain, S. L.
Psg. Font 14, local 8
08013 Barcelona
SPAIN
www.campbellsci.es
info@campbellsci.es

Please visit www.campbellsci.com to obtain contact information for your local US or International representative.