BDR320 BASIC DATA RECORDER
OPERATOR'S MANUAL

REVISION: 6/96

WARRANTY AND ASSISTANCE

The **BDR320 BASIC DATA RECORDER** 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. To obtain a Returned Materials Authorization (RMA), contact CAMPBELL SCIENTIFIC, INC., phone (801) 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.

Non-warranty products returned for repair should be accompanied by a purchase order to cover the repair.

CAMPBELL SCIENTIFIC, INC.

815 W. 1800 N.
Logan, UT 84321-1784
USA
Phone (801) 753-2342
FAX (801) 750-9540

Campbell Scientific Canada Corp.
11564 -149th Street
Edmonton, Alberta T5M 1W7
CANADA
Phone (403) 454-2505
FAX (403) 454-2655

Campbell Scientific Ltd.
14-20 Field Street
Shepshed, Leics. LE12 9AL
ENGLAND
Phone (44)-50960-1141
FAX (44)-50960-1091

LICENSE FOR USE

The PC 300 support software included with the BDR 320 is protected by both United States copyright law and international copyright treaty provisions. You may copy it onto a computer to be used, and you may make archive copies of the software for the sole purpose of backing-up Campbell Scientific, Inc. software and protecting your investment from loss. All copyright notices and labelling must be left intact.

This software may be used by any number of people, and may be freely moved from one computer location to another so long as there is no possibility of the software being used at one location while it's being used at another. The software, under the terms of this license, cannot be used by two different people in two different places at the same time.

LIMITED WARRANTY

Campbell Scientific, Inc. warrants that the magnetic diskette on which the accompanying computer software is recorded and the documentation provided with it are free from physical defects in materials and workmanship under normal use. Campbell Scientific, Inc. warrants that the computer software itself will perform substantially in accordance with the specifications set forth in the Operator's Manual published by Campbell Scientific, Inc. Campbell Scientific, Inc. also warrants that the software is compatible with IBM PC/XT/AT and PS/2 microcomputers and 100% compatible computers only. Campbell Scientific, Inc. is not responsible for incompatibility of this software running under any operating system other than those specified in accompanying data sheets or operator's manuals.

The above warranties are made for ninety (90) days from the date of original shipment.

Campbell Scientific, Inc. will replace any magnetic diskette or documentation which proves defective in materials or workmanship without charge.

Campbell Scientific, Inc. will either replace or correct any software that does not perform substantially according to the specifications set forth in the Operator's Manual with a corrected copy of the software or corrective code. In the case of a significant error in the documentation, Campbell Scientific, Inc. will correct errors in the documentation without charge by providing addenda or substitute pages.

If Campbell Scientific, Inc. is unable to replace defective documentation or a defective diskette, or if Campbell Scientific, Inc. is unable to provide corrected software or corrected documentation within a reasonable time, Campbell Scientific, Inc. will either replace the software with a functionally similar program, or refund the purchase price paid for the software.

Campbell Scientific, Inc. does not warrant that the software will meet licensee's requirements or that the software or documentation are error free or that the operation of the software will be uninterrupted. The warranty does not cover any diskette or documentation which has been damaged or abused. The software warranty does not cover any software which has been altered or changed in any way by anyone other than Campbell Scientific, Inc. Campbell Scientific, Inc. is not responsible for problems caused by computer hardware, computer operating systems, or the use of Campbell Scientific, Inc.'s software with non-Campbell Scientific, Inc. software.

ALL WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED AND EXCLUDED. CAMPBELL SCIENTIFIC, INC. SHALL NOT IN ANY CASE BE LIABLE FOR SPECIAL, INCIDENTAL, CONSEQUENTIAL, INDIRECT, OR OTHER SIMILAR DAMAGES EVEN IF CAMPBELL SCIENTIFIC HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Campbell Scientific, Inc. is not responsible for any costs incurred as a result of lost profits or revenue, loss of use of the software, loss of data, cost of re-creating lost data, the cost of any substitute program, claims by any party other than licensee, or for other similar costs.

LICENSEE'S SOLE AND EXCLUSIVE REMEDY IS SET FORTH IN THIS LIMITED WARRANTY. CAMPBELL SCIENTIFIC, INC.'S AGGREGATE LIABILITY ARISING FROM OR RELATING TO THIS AGREEMENT OR THE SOFTWARE OR DOCUMENTATION (REGARDLESS OF THE FORM OF ACTION - E.G. CONTRACT, TORT, COMPUTER MALPRACTICE, FRAUD AND/OR OTHERWISE) IS LIMITED TO THE PURCHASE PRICE PAID BY THE LICENSEE.

BDR320 BASIC DATA RECORDER
TABLE OF CONTENTS

APPENDICES

TABLES

FIGURES

BDR320 OVERVIEW

OV1. INTRODUCTION

The BDR320 is a basic data recorder intended for remote data acquisition applications. The BDR320 includes:

- the CPU module
- the wiring panel
- either the PS12-LA or the PS12-ALK power supply
- a watertight 10 x 12 inch enclosure
- a copy of PC300 BDR Support Software (contains EDLOG3, TERM3, BTOA and SMCOM)
- the BDR320 operator's manual

This Overview introduces the BDR320, its hardware, software and operation. It includes a tutorial which takes the user through a simple example using Prompt Programming. Section 2 explains Prompt Programming in detail. This method of programming is quick, easy and sufficient for most applications.

Sections 3 and 4 describe data retrieval in the BDR320. For many users, this overview and the first three sections of the manual will be all they need.

For users with complex programming needs, including conditional outputs or control, Sections 5 and 7 describe Direct Programming and EDLOG3. Section 8 lists and explains the Instruction Set which can be used to create BDR320 programs for unique applications.

OV2. HARDWARE

Components consist of a 12 VDC power supply, wiring panel, datalogger and enclosure. The BDR320 system is shown in Figures OV2-1 and OV2-3. Figure OV2-2 contains an illustration of the terminal strip and an explanation of the function associated with each terminal.

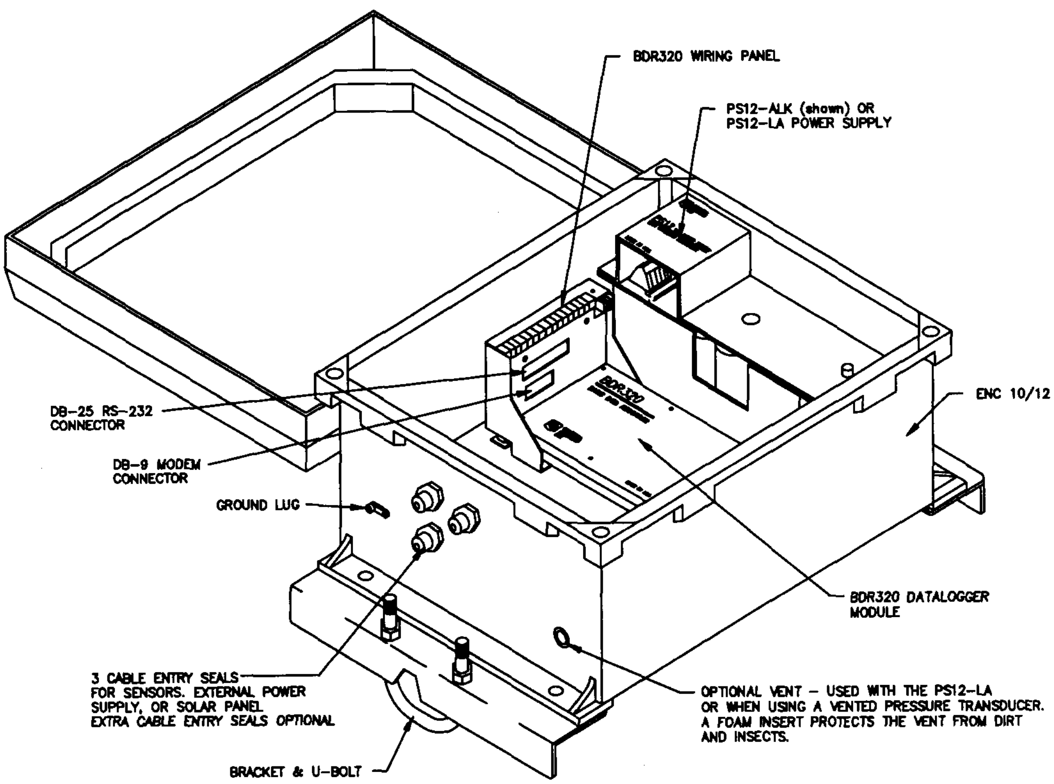
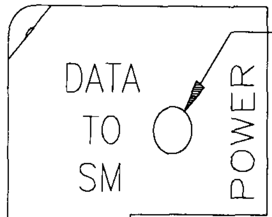
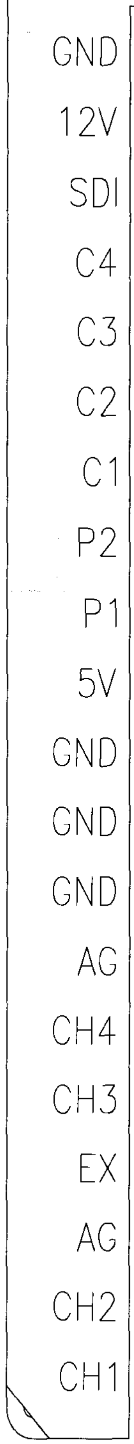


FIGURE OV2-1. The BDR320



An LED verifies storage module data transfer.

Two connectors for inputting 12 VDC to the BDR320.



SDI-12 COMMUNICATION LINES - SDI-12 sensors are measurement devices that include one or more sensing elements and a serial digital interface (SDI) for communicating with a datalogger and transferring results. Up to 10 SDI sensors can be connected to and addressed by the BDR320. The 12X command allows communication with SDI sensors (Section 9.14). (12V and GND can be used for other purposes if there are no SDI sensors in use.)

CONTROL PORTS - A port can be set high (5 VDC) or low (0 VDC) according to time or a measured event. Typically a control port is used to activate an external device (e.g. a sampler) through a relay. A relay driver circuit is usually required due to the small current drive of the control ports (1.5 mA at 3.5 V). C1 can be set high or low continuously. C2, C3, and C4 are active only during program execution (See Appendix C). C2, C3, and C4 can also be used as digital inputs to read 5V CMOS logic levels. (Instruction 91.)

PULSE COUNTERS - Available for measuring switch closure and voltage pulse type sensors. P1 is slow (20 Hz); a typical application is a tipping bucket rain gage. P2 is faster (150 Hz); a typical application is a contact closure flow meter.

5 VDC - Power for external devices (80 ma maximum.) This is not switched; it is always on.

POWER GROUND - To tie cable shields to ground and to provide a ground reference for pulse counters and binary inputs. For protection against transient voltage spikes, power ground should be tied to a good earth ground. The external grounding lug on the enclosure is provided for this purpose.

ANALOG GROUND - Reference for single-ended measurements and excitation return.

ANALOG INPUTS - CH3 and CH4 are single-ended inputs. When wired and programmed for a different measurement, CH3 is the HI side and CH4 is the LO side of Differential Channel 2. See below for definitions of single-ended and differential measurements.

EXCITATION - Provides a fixed excitation voltage for resistive bridge measurements. Bridge measurements have a ratiometric output.

ANALOG GROUND - See above.

ANALOG INPUTS - Terminals CH1 and CH2 are singled-ended analog input channels. In a single-ended measurement the voltage on an analog input is measured with respect to analog ground (AG). When wired and programmed for a differential measurement, CH1 is the HI side and CH2 is the LO side of Differential Channel 1. In a differential measurement, the voltage on the HI input is measured with respect to the voltage on the LO input.

FIGURE OV2-2. BDR320 Input Terminals

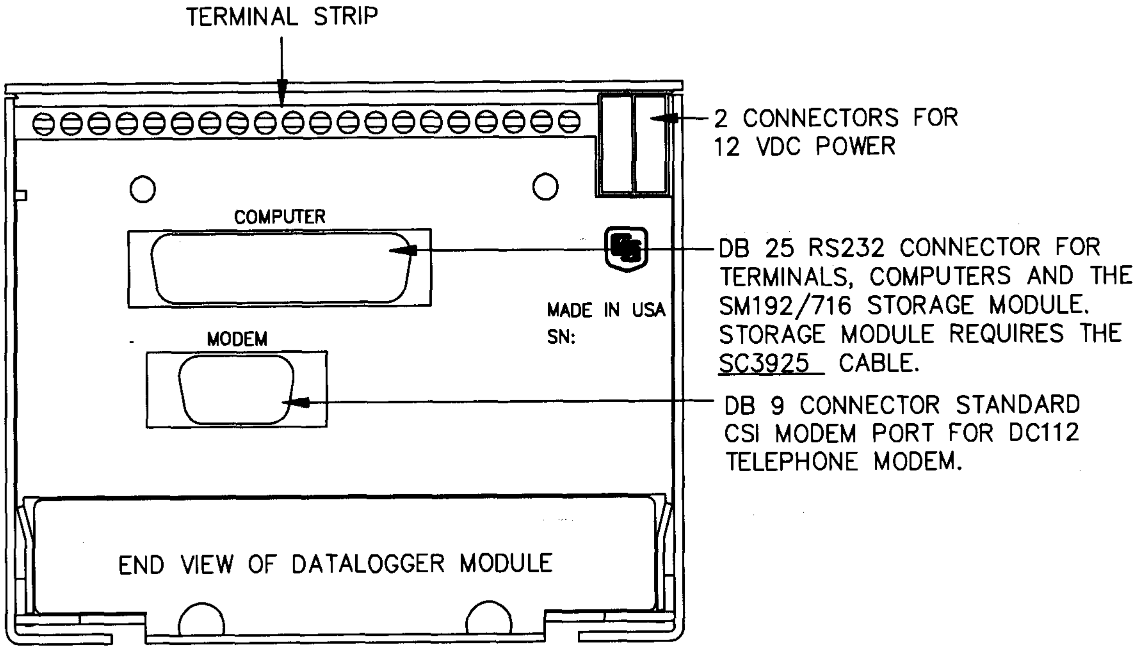


FIGURE OV2-3. BDR320 Wiring Panel

> **NOTE:** The excitation voltage on the BDR320 is 4000 mV ±50mV. During calibration at the factory, the **exact** excitation voltage vs the CPU module temperature is measured and burned into the PROM.
>
> At each execution of the program, the BDR320 measures its temperature and, based on a 3rd order polynomial, calculates the value of the excitation voltage. This method has an **accuracy of 0.02%** full scale. This calculated value is used in the BDR320's ratiometric bridge measurements.

Access to the BDR320 is through a computer. Functions such as programming, setting the BDR clock and changing the offset are accomplished by issuing commands, either directly or over a telecommunications link from the computer. Data are retrieved with the computer, the SM192/SM716 Storage Module, or via telecommunications.

OV3. SOFTWARE

The operating system of the BDR320 is burned into the system EPROM and is transparent to the user. The RAM in the BDR320 stores the user written program (Section OV4.2) and Final Storage data. This storage is dynamically allocated. The space for program storage is deducted from the total; the remainder is allocated to data storage. For most applications, the BDR320 can store in excess of 30,000 data points.

In the BDR320, data are stored in **output tables.** In most cases, an output table stores all data output at a certain time interval, e.g., Table 1 could be created to store 1 minute data, Table 2 could store 10 minute data and Table 3 could store daily data.

PC300 BDR Support Software supports the user in all BDR operations except telecommunications. Telecommunications is supported by PC208 Datalogger Support Software which may be purchased from Campbell Scientific, Inc. This manual does not deal with BDR telecommunications.

PC300 contains the following programs:

TERM3 - allows communication between the BDR320 and an MSDOS compatible computer. Other features support: BDR programming, data retrieval, and monitoring of inputs (Section 1).

EDLOG3 - a program editor for the 300 series dataloggers. It is used to create a BDR program in the direct programming method. (Section 7)

SMCOM - The SM192/716 storage module data retrieval program. (Section 4.1)

BTOA - converts binary data to ASCII. Storage module data is retrieved in binary. (Section 4.2)

OV4. OPERATION

There are three phases of the operation of the BDR320:

- Installation
- Programming
- Data Retrieval

OV4.1 INSTALLATION AND MAINTENANCE

OV4.1.1 POWER SUPPLIES

The BDR320 operates at a nominal 12 VDC. Below 9 or above 18 VDC, the BDR320 will not operate properly.

The BDR320 comes complete with either the PS12-ALK (BDR320 B) or PS12-LA (BDR320 C) Power Supply. The batteries for the power supplies are shipped separately. They must be installed. If you have the PS12 ALK, install the eight alkaline D cells according to the "map" inside the holder. Place the holder inside the PS12 with the connector to the wiring panel side. Make sure the PS12 switch is in the "off" position. Insert the power connector into the receptacle in the wiring panel. Connect all sensor leads, control lines, etc., and then turn the switch on.

If you have the PS12-LA, the procedure is the same except that you install the seven amp hour lead acid battery into the PS12.

Appendix B contains complete information on the PS12.

> **NOTE:** The PS12 contains two power connectors so that a fresh battery (ies) can be connected to the BDR Module before the old one is removed. Power continues uninterrupted.

If the power supply drops below 9.0 VDC, the datalogger enters a low power survival state where programming and data are maintained but program execution stops and communication ceases. Functions return to normal when adequate power is provided.

The BDR "wakes up" every five minutes to check if the supply voltage has risen. If the voltage is low, the datalogger returns to the subsistence level. When an adequate voltage is supplied, the BDR resumes program execution within five minutes or when it receives a command from the computer. An E05 and E06 error with time is recorded in the Error Log when the low power supply condition starts and stops, respectively.

OV4.1.2 PROTECTION FROM THE ENVIRONMENT

Moisture, dirt and extremes of temperature are of primary concern.

The BDR320 is production tested to operate from -35 to +55^{o}C. It has a 10 x 12 inch fiberglass enclosure that is classified as NEMA 6P before the entry holes are drilled. The cable entry fittings are watertight when properly used. Each BDR320 contains desiccant which should be changed regularly.

The BDR320 can be mounted to a wall, to a bench or to a 1 1/4 inch pipe. The enclosure has a small vent which serves two purposes. When a rechargeable battery is used for power, the vent furnishes an escape for hydrogen gas in the unlikely event of a charging malfunction. The vent also provides a dry chamber for venting a pressure transducer to atmosphere.

Unfortunately, even this small vent allows moisture to enter and deplete the desiccant. Vented enclosures should have their desiccant changed often. See the desiccant schedule, Appendix A. The BDR320 is shipped with a special plug which can be used to close the vent opening. IF the unit has NEITHER the rechargeable battery nor a vented pressure transducer, remove and save the vent hardware and install the plug with the wing nut on the inside of the enclosure.

> **CAUTION:** The vent must be used if the unit has rechargeable batteries.

OV4.1.3 WIRING

The standard version of the BDR320 contains three watertight cable entry seals. Each cable seal accepts a single cable. These cable seals have gasketed openings which constrict as the outer ring is screwed on tighter.

Each of the two cable entry seals closest to the lid of the enclosure (Figure OV4-1) can accept a cable with a diameter from .118 to .275 inches. The single cable seal beneath these two has a diameter range of .231 to .394 inches. The smaller cable seals accommodate the majority of CSI sensors as well as the cable from the solar panel. The larger seal will handle the cable from a vented pressure transducer.

Each cable entry seal is shipped with a plug which maintains the watertight seal. To wire a sensor to the BDR320, loosen the outer ring on the cable seal, remove the plug and save it inside the enclosure. Insert the wire and connect the conductors to the wiring panel. Tighten the outer ring so the cable is enclosed snugly. Over time the outer ring will need to be re-tightened to maintain the integrity of the seal.

The BDR320 can be ordered with six watertight cable entry seals (Figure OV4-1): four with the smaller diameter and two with the larger diameter. There is an extra charge for this option.

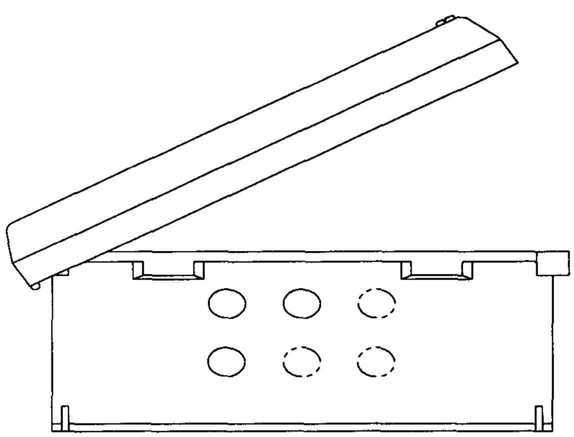


FIGURE OV4-1. Connector Locations on the Bottom of the BDR320

OV4.1.4 GROUNDING

A good earth ground establishes ground reference and reduces damage from lightning by providing a low resistance path around the system to a point of low potential.

ALL DATALOGGERS MUST BE EARTH GROUNDED.

It is the users responsibility to provide this earth ground.

All components of a system (datalogger, sensors, external power supplies, mounts, housings, etc.) should be referenced to one common earth ground. The grounding lug on the bottom side of the BDR320 enclosure is provided for this purpose.

EFFECT OF GROUNDING ON MEASUREMENTS: COMMON MODE RANGE

In order for a differential measurement to be made, both inputs must be within ±5 volts of the BDR ground. This ±5 volts is referred to as the common mode range. For example, if the high side of a differential input is at 5V and the low side is at 0.5V relative to BDR ground, a measurement made on the 5V input range is 4.5V. The same measurement results if the high side of the signal is 0V and the low side is -4.5V relative to the BDR ground. If the high input exceeds 5V, or the low input goes below -5V, the common mode range is exceeded and the measurement cannot be made.

Exceeding common mode range could be a problem when the BDR is measuring the output from a sensor which has its own grounded power supply and the low side of the signal is referenced to power ground. If the BDR 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 BDR ground should be connected, creating one ground for the system.

OV4.1.5 USE OF THE DIGITAL CONTROL PORT FOR SWITCHING RELAYS

The digital control ports can be set low or high (0V low, 5V high) using the port commands associated with Program Control Instructions 83 through 93. C1 can be set high or low continuously; C2, C3, C4 are active only during program execution. A digital output port is normally used to operate an external relay driver circuit because the port itself has a limited drive capability.

For details on this application see Appendix C.

OV4.1.6 MAINTENANCE

The BDR and power supply require minimal routine maintenance. Alkaline batteries should be changed and rechargeable batteries should be changed or charged before they drop below 9.1 volts. The voltage should be read by the BDR320. Use the stAtus command in TERM3.

When alkaline cells are not in use, remove them from the battery holder to prevent possible corrosion. Rechargeable lead acid batteries can be left in PS12 and float charged. A lead acid battery loses capability during storage. A battery which has been stored for a long period of time should be "top charged" before being put into service again (Appendix B).

Desiccant should be replaced routinely. See the desiccant schedule, Appendix A.

OV4.2 PROGRAMMING

In a basic application, the BDR measures the sensor(s) and stores the result for subsequent transfer to a computer or storage module. The BDR can also process the measurements over time and store the average, maximum, minimum, total, or standard deviation. For example, stage height can be measured and stored every 15 minutes. From the 15 minute readings, a daily average, maximum, and minimum can be calculated and stored.

Whether the application is simple or complex, the function of the BDR is controlled by a user written program.

There are two methods of programming the BDR:

- PROMPT programming - for simple applications
- DIRECT programming in EDLOG3

OV4.2.1 PROMPT PROGRAMMING

In PROMPT programming, the user "fills in the blanks" in an input table and an output table. The input table contains the information needed to make the measurement(s), e.g., scan interval, channel number, and the type of measurement. The output table contains the information needed to process the measurement results and generate the required output data, e.g. output interval, type of processing (average, maximum) etc. Once the tables have been completed, the BDR generates the program, loads it and displays the number of days until the final storage will be full.

All of the software required for prompt programming is in the datalogger. TERM3, or some other terminal emulator program, is required for the computer to communicate with the datalogger. The prompt programming method is designed for simple programs that measure, process and store data. Specialized programs requiring program branching, subroutines or intermittent output must be developed through direct programming.

An example using Prompt programming is given in OV5, and more details are in Section 2.

OV4.2.2 DIRECT PROGRAMMING

In DIRECT programming, the operator uses the BDR Instruction Set to "build" a program which the BDR executes. Direct programming is very flexible, allowing branching, subroutines and intermittent (conditional) output (Sections 5-8).

Direct programming can be accomplished in terminal emulator mode but the usual method is to develop the program in EDLOG3 (part of CSI PC300 BDR Support Software) and load the program into the BDR from the computer using TERM3 (also part of PC300). PC300 requires an IBM-PC or compatible computer.

> **NOTE:** Regardless of the programming method, when a program is entered or edited, the BDR320 must recompile and allocate memory for the program and data storage. Reallocating memory causes all stored data to be erased. **Always retrieve data from the BDR before making program changes or reallocating memory.**

OV4.3 DATA RETRIEVAL

Data stored in the BDR320 can be retrieved by a computer, the CSI SM192 or SM716 Storage Module or via telecommunications.

OV4.3.1 DATA RETRIEVAL BY A COMPUTER

TERM3, part of the PC300 software package, is used to facilitate data retrieval by a computer. Connect the computer's serial port to the DB25 port on the BDR wiring panel. This port is RS232; no interface is required.

Run TERM3 and select one of the data retrieval options. Data will be saved in ASCII (binary is optional).

The resultant file will be suitable for direct import by spreadsheet packages such as Lotus 123.

Complete details on this procedure are found in Section 3.

OV4.3.2 DATA RETRIEVAL BY THE SM192 OR SM716 STORAGE MODULE

In addition to the BDR and the storage module, this operation requires the SC3925 cable, SMCOM, and BTOA (Section 4).

> **NOTE:** The BDR cannot send data to the storage module in an "on line" mode. The storage module must be brought to the BDR and connected and allowed to "milk" the data under operator control.

OV4.3.3 TELECOMMUNICATIONS

Data retrieval via telecommunications (phone, RF, etc.,) is supported with a special (at this time) version of the PC208 Software. Contact the Campbell Scientific Water Resources group for details.

OV5. DEMONSTRATION TUTORIAL

For this tutorial, you need the BDR320, the sample "level sensor", a computer cable, and the PC300 software. If you received this manual with the BDR320, the "Level Sensor" should be inserted in the binder in front of this page.

As stated earlier, there are three main phases in the operation of the BDR320:

Installation

Programming

Data Retrieval

This tutorial takes you through each of these phases with simple examples. There are two problems for you to work out. Refer to the solutions provided or to the manual as needed.

This tutorial is not a substitute for reading the manual. It is intended to help you quickly learn the basics and understand the features and power of the BDR320.

Let's get started.

OV5.1 INSTALLATION

Attach your RS-232 ribbon cable to the 25 pin connecter on the BDR320 and to the serial port of your computer. Wire the "level sensor" to the terminal strip as shown on the diagram of the sensor. Turn the PS12 switch on.

Installing PC300 Software:

Hard Disk
Make a directory on your hard disk for PC300:

C:\>md pc300

Copy the contents of the PC300 disk to the hard disk directory:

C:\>cd pc300

C:\>PC300\copy a:*.*

Floppy Disk
Format a blank disk and make a working copy of the PC300 software disk. If PC300 is in drive A and the formatted disk in drive B, use the copy command:

A:\>copy *.* b:

OV5.2 EXECUTING TERM3

From the drive containing the working copy, or the PC300 directory on your hard disk, execute TERM3 by typing **TERM3** followed by the **Enter** key at the DOS prompt. TERM3 will find the BDR320 through the appropriate COM port. If everything is turned on and connected correctly you should soon see a screen that offers the following choices:

TERM3 OPTIONS

U - get Uncollected data
R - Rename station, collect all
D - Download program to datalogger
S - Save program from datalogger
M - Monitor Input Locations
P - Prompt Programming
T - Terminal emulator
K - clocK view/set
A - stAtus
Q - Quit
Option:_

First, check the status of the BDR320 by typing **A** on your computer. The status will be displayed on your computer screen as follows (specific values for some of the items will be different on your BDR):

Time 01/01/89 00:00:45
NextExe
Battery:+11.315
Input Locs:0028
Prgm Bytes:0058
Storage:+30919
Unused:+30919
Data Tables:0000
Prgm Sig:+1326.0
EE Sig:+7237.0
EE Ver:01
Cal: 03/07/91
Prom Sig:+20135
Errors:0000
Prom ID:+6661.0
OBJSrlNo:0002
SN:+1219.0
C1466
*****

Now set the BDR320 clock by selecting the K - clocK view/set option. Type **K**, and the following screen is displayed:

Datalogger time: 01/01/89 00:01:16
PC time: 10/22/90 1 10:42:51

Set datalogger time to PC time? (Y/N)_

Type **Y** to set the clock.

OV5.3 PROGRAMMING

This tutorial demonstrates Prompt Programming. Prompt programming is initiated by executing the P command. Type **P** to enter the prompt programming mode. The following screen will be displayed:

I - Input Tables
D - Data Storage Tables
E - Erase All Tables
ESC - Quit
Select I,D,E,ESC

Enter I and the following will appear:

Input Table Number 01
Measurement Interval mins 0005

We will use the default table 01, and measure once a second. Press **Enter** and the cursor moves to the minutes field; the space bar toggles the interval back and forth between minutes and seconds. With **secs** displayed, press **Enter**; the cursor moves to the next field where you enter **1** for a one second measurement interval.

Sensors are grouped in Input Tables according to the interval at which they are to be measured. Separate input tables are used if different measurement intervals are necessary. Press **Enter** again to see the screen:

LOC	NAME	UNITS	TYPE	CHN	MULT	OFFSET
01	_		OPT?			

We will name the sensor "Level" and label the units as inches. Type **Level** and press **Enter**; the cursor will move to the units column. Type **in** and press **Enter**. The sensor is a potentiometer; in the field **TYPE**, press the space bar repeatedly until **POT** appears and then press **Enter**. Type 3 for the channel number. Leave the **MULT** and **OFFSET** at 1.0 and 0 for now.

LOC	NAME	UNITS	TYPE	CHN	MULT	OFFSET
01	Level	in	POT	03	1.0	0.0
02			OPT?			

Press **ESC** three times to back out to the original menu screen. Select **M**onitor Input Locations and the current measurement is displayed. The level sensor is at location 1; the measurement is updated every second. Squeeze the level sensor in different places along its length, and watch the value change on the screen.

The sensor output is in volts per volt of excitation. The output you see will vary between 0, at no resistance, to 1, at maximum resistance. The sensor is five inches in length, so change the multiplier to 5 to convert the units to inches. To do this, press **ESC** to return to the main menu, and press **P** to enter the prompt programming mode. Go back into input table #1 and change the multiplier to 5 by using the cursor keys or **Enter** to move to the MULT column. The table should look like this when you are finished:

LOC	NAME	UNITS	TYPE	CHN	MULT	OFFSET
01	Level	in	POT	03	5	0.0
02			OPT?			

Again, escape out to the main menu and select **M**onitor Input Locations to observe the measurements.

At this point, we are making measurements, but not storing any data. For this example, record the average level every ten seconds. Enter the prompt programming mode by returning to the main menu, and typing **P**. This time type **D** to enter the Data Storage Tables. Enter table 1 and set the interval to ten seconds.

Data Table Number 01
Recording Interval secs 10

The following screen will then appear:

Item	Loc		Process
01	01	Level	OPT?

The Loc column contains the Input Storage locations in which the measurements programmed in the Input Tables are saved.

Our level measurement is in location 1; enter it. The name of the Loc is automatically displayed to the right. The Process column allows you to store the average, maximum, total, etc. Again, the space bar scrolls you through the possible choices. We want to record the average level every ten seconds; set the Process to AVG. The screen should appear as follows:

Item	Loc		Process
01	01	Level	AVG

Escape out to the main menu. As the program compiles, you are told the number of days of data the BDR320 can hold in memory. Once the memory fills, the oldest data are overwritten by new data.

OV5.4 DATA COLLECTION

To collect data to a disk file, exit to the main menu and select **U** for **get Uncollected data.** You are prompted for a site name. Call the site TUT. TERM3 will store the data in a file named TUT-1.001.

Exit TERM3 by typing **Q**. At the DOS prompt use the "type" command to view your data (i.e. c:\PC300>type TUT-1.001). The data from TUT-1.001 should look something like this:

```
"SN:+1219.0 Table:0001 Time:10/22/90
11:27:10 Interval:0010 s"
"M""D""Y""H""M""S""Level "
"M""D""Y""H""M""S""in__A"
10/22/90 11:27:10  3.66
10/22/90 11:27:20  3.56
10/22/90 11:27:30  3.46
10/22/90 11:27:40  3.35
10/22/90 11:27:50  3.24
```

NOTE: Your numbers will differ.

Congratulations, you have programmed the BDR320 and logged data! Now let's go on to a problem that is a little bit more complex.

Problem #1

We need to measure the water level every second and record an average every five seconds. Our reading must match a staff gage reading on site. Four inches of water on our sensor is equal to a reading of 24 inches on the staff gage. We also need to record the average, maximum, and minimum level every five minutes along with the internal temperature of the BDR320. A pH sensor requiring a 30 sec. warm up will also be recorded at the five min. interval.

Solution:
From the TERM3 menu, press **P** to enter the prompt programming mode.

INPUT TABLES

Because we want to read sensors and store data at two different time intervals, we must use two Input and two Data Storage Tables. Set up the first Input Table the same as in the previous example. The second Input Table also has some extra options for control and averaging. Let us assume that the internal battery voltage reading is from a sensor that requires a 30 second warm up (e.g., some pH sensors). The second table measures the battery voltage and internal temperature every five minutes. Turn on the C1 control port 30 seconds prior to taking the measurement. Place the cursor on the Input Table Number; change it to 2. Change the header values to the following:

Input Table Number 02
Measurement Interval secs 300
Ctrl Port Prior turn on secs 30
Duration of Average (s) 0000
# of Samples in Avg. 0001

The battery voltage and temperature are internal measurements and do not use any input channels. These measurements are selected in the TYPE column.

LOC	NAME	UNITS	TYPE	CHN	MULT	OFFSET
02	Batt	V	BATT			
03	Temp	C	TEMP			
04			OPT?			

Battery voltage is always output in volts, and internal temperature in °C; a multiplier and offset are not needed with these measurements.

OUTPUT TABLES

The first Data Storage Table is the same as the example above except that the interval is changed to five seconds.

The second table records every five minutes and has five outputs: three for the level sensor, one for the battery voltage, and one for the temperature. Change the recording interval of data table 1 and enter the data table number 2 to program data table 2. The item number is automatically incremented as each new output value is added to the table by typing cursor down or **Enter** at the end of an item entry. Lines with "OPT?" are merely places to add new output processing.

Table 2 should appear as follows:

Data Table Number 02
Recording Interval mins 5

Item	Loc		Process
01	01	Level	AVG
02	01	Level	MAX
03	01	Level	MIN
04	02	Batt	SMPL
05	03	Temp	SMPL

Escape out to the main menu again, and enter monitor mode to check the measurements. The pH (battery) and temperature are not measured until the end of the five minute interval.

ADJUST OFFSET

When a water level sensor is installed in the field, the offset applied to the measurement must be adjusted so the reading matches a fixed datum. Let's adjust the BDR320 reading to match the "staff gage." Type **T** to enter the terminal emulator (this time a small window appears above the monitor screen). The Y command enables us to easily adjust our measurement to the staff gage reading. Type **1Y** to adjust the offset for input location #1. The first number you see should be 0.0, which is the current offset. The second number, the sensor reading, should update once per second. Apply pressure to the sensor at the mark labeled 4. At this sensor reading, assume we would like the logger to record a stage of 24 inches. While continuing to squeeze the sensor at the 4 mark, type **24** and **Enter**. The screen should update with an offset of about 20 and a reading of about 24. Move your finger up and down the sensor and the reading should vary from 20 to 25 inches.

Escape back to the monitor screen. Locations 2 and 3 with the pH (battery voltage) and temperature are probably updated by now.

Data notes can be stored in the BDR320 memory. Enter the terminal emulator again (Type **T**), and type in the command **4321S** and press **Enter**. The date and time are displayed, followed by the cursor. A message up to 1000 characters in length may be entered. Type in "offset adjustment made". Escape back to the monitor screen.

COLLECT DATA

To unload data to a disk file, exit to the main menu and select **U** for **get Uncollected data.** You are prompted for a site name because the program was changed from the previous collection. Call the site TEST. A suffix has also been added, and the names of the files are TEST-1.001, TEST-2.001, and LTEST.001. Exit TERM3 by typing **Q**. At the DOS prompt, use the "type" command to view your data (i.e., c:\>type TEST-1.001). The data from TEST-1.001 should look like the output we reviewed on the screen in the earlier example. LTEST.001 contains the notes log, the time set log, and the error log. The data for TEST-2.001 should look something like this:

TABLE OV5-1. TEST-2.001 Data

"SN:+1219.0 Table:0002 Time:10/22/90 14:20:00 Interval:0005 min"

"M""D""Y"	"H""M""S"	"Level "	"Level "	"Level "	"pH "	"temp "
"M""D""Y"	"H""M""S"	"in___A"	"in___H"	"in___L"	"V____S"	"C____S"
10/22/90	14:20:00	16.75	24.87	20.36	11.3	24.56
10/22/90	14:25:00	22.64	24.91	20.25	11.34	24.71
10/22/90	14:30:00	23.4	20.53	20.2	11.29	24.83
10/22/90	14:35:00	21.74	23.28	20.51	11.35	24.9
10/22/90	14:40:00	23.9	24.45	23.23	11.35	24.86
10/22/90	14:45:00	24.36	24.56	24.06	11.36	24.85
10/22/90	14:50:00	24.1	24.5	23.09	11.36	24.85
10/22/90	14:55:00	23.54	24.2	23.08	11.36	24.87
10/22/90	15:00:00	24.5	24.67	24.17	11.36	24.89

OV6. BDR 320 SPECIFICATIONS

The following electrical specifications are valid for an ambient temperature range of -35 to +55°C, unless stated otherwise.

ANALOG INPUTS

NUMBER OF CHANNELS: 2 differential or up to 4 single-ended

MEASUREMENT TYPES: single-ended and differential voltage, ratiometric half bridge and full bridge

ACCURACY: Single-ended or differential voltage: 0.1% of full scale
Ratiometric bridge measurements: 0.02% of full scale

INPUT RANGE, INTEGRATION TIME, RESOLUTION:

Full Scale Range mV	Integration ms	Resolution µV
-20 to +5000 autoranging	up to 16.7	2.8 to 43
-20 to +80	50.0	0.7
-20 to +120	33.3	1.4
-20 to +250	16.7	2.8
-30 to +1000	5.3	8.6
-50 to +5000	1.08	43.0

INPUT NOISE VOLTAGE: 0.25 µV RMS on -20 to +80 mV range

COMMON MODE RANGE: ± 5 volts

INPUT CURRENT: 10 nanoamperes

INPUT RESISTANCE: 6 gigohms

PULSE COUNTERS

NUMBER OF CHANNELS: 2

INPUT SIGNAL: Switch closure

	Channel 1	Channel 2
Max input frequency (Hz)	20	150.0
Min switch closure time (µs)	100	200.0
Min voltage pulse low time (µs)	100	200.0
Max debounce filter time (ms)	10	5.5
Max voltage input (v)	5.0	5.0
Result	Counts	Frequency

NOTE: Pulse count channel 1 activates the processor on each count requiring 13mA for 20 ms.

SDI-12

Compatible with SDI-12 standard version 1.0, October, 1988

ANALOG OUTPUTS

EXCITATION: A single excitation output for resistive bridge measurements; switched to 4.0 volts ± 50mV at time of measurement. Maximum output current: 35 mA

NOTE: The precise value of the excitation voltage is measured during the BDR 320 calibration. Bridge measurements are ratiometric with an accuracy of 0.02% of full scale.

5V PORT: Continuous, 5VDC ± 0.2VDC
Current available: 72 mA maximum

DIGITAL CONTROL OUTPUTS

Four digital control outputs, enabled according to a programmed time or event. C1 can be set high or low continuously; C2, C3, C4 active only during program execution.

OUTPUT VOLTAGE (no load):
High, 5V ± 0.1V
Low, 0V ± 0.1V

OUTPUT RESISTANCE: 1000 ohms

RS-232 PORT - DB-25 CONNECTOR

FORMAT: ASCII, 8 bit, no parity, 1 start bit, 1 stop bit

TYPE: RS-232C

BAUD: 300, 1200, 9600

MODE: Full duplex, asynchronous

NOTE: This DB-25 connector functions as the Storage Module (SM192/716) port.

MODEM PORT

DB-9 connector for CSI modems

TRANSIENT PROTECTION

All input and output connections to the BDR320 are protected using RC filters and transzorbs or spark gaps.

POWER

POWER SUPPLY: 9VDC minimum; 18VDC maximum

CURRENT DRAIN: 80 microamps quiescent; 13 mA when communicating with a computer, or SDI-12 sensor; 22 mA during analog measurement.

DATA LOSS PROTECTION: If the power supply drops below 9.0 VDC, the datalogger enters a low power survival state where programming and data are maintained but program execution stops and communication ceases. Functions return to normal when adequate power is provided.

PROGRAMMING

PROMPT PROGRAMMING - program generation from prompted input

DIRECT PROGRAMMING - flexible instruction set; EDLOG3

MEASUREMENT INTERVAL - 1 second to 1440 minutes

PROCESSING - Numerical and transcendental operations for algorithm development

OUTPUT PROCESSING - Sample, average, totalize, maximize, minimize, histogram, wind vector, sample on max or min, standard deviation

COMPUTER COMMANDS SHORT LIST

STATUS - Listing of parameters critical to datalogger operation

DATA DIAGNOSTICS - Occurrences of run time errors are logged. Changes to real-time clock are also logged.

DATA RETRIEVAL - Retrieve all data, all since last retrieval, or time window; optional time tag, ASCII format.

DATA NOTES - Up to 1000 alpha-numeric characters.

DISPLAY INPUTS - Instantaneous measurements on command for on-site verification and calibration of sensors.

ADJUST OFFSET - Enter desired measurement value and offset is calculated automatically.

PHYSICAL

DATALOGGER & WIRING PANEL ONLY:
Size: 7.4 x 4.75 x 3.75 inches
Weight: 1.3 lbs.

WITH STANDARD ENCLOSURE:
Size: 15.5 x 11.4 x 6.9 inches
Weight: 11.0 lbs. (less battery)

SECTION 1. TERM3

TERM3 is a communications program specifically designed for the 300 Series (including the BDR320) dataloggers from Campbell Scientific, Inc. It has features for programming the BDR320 and for data retrieval with a computer. TERM3 runs on IBM PC, PC-XT, PC-AT and IBM PC compatible computers.

To run TERM3, connect the BDR to the PC with a direct cable and type 'TERM3' on the DOS command line. Press RETURN.

TERM3 automatically establishes communication at 9600 baud on COM1 or COM2. When done, it leaves the COM Port and its interrupt completely disabled. Subsequent programs must reinitialize the Port before using it (e.g., if the Port is used for a serial printer, the DOS MODE command must be issued before the PRINT command will address the COM Port).

If all connections are correct, the TERM3 OPTIONS menu will appear.

1.1 TERM3 OPTIONS

U - get Uncollected data
R - Rename station, collect all
D - Download program to datalogger
S - Save program from datalogger
M - Monitor input locations
P - Prompt programming
T - Terminal emulator
A - stAtus
K - clocK view/set
Q - Quit

1.1.1 OPTION U - GET UNCOLLECTED DATA

Collects all data since the last data retrieval. During the initial data retrieval, all data are retrieved. Data are retrieved in ASCII unless binary is requested. The file formats are discussed in Section 3.2.

The first time data are retrieved from a station, TERM3 creates a station file. It prompts the user for a site name for the station. If no name is given, TERM3 enters the serial number of the BDR as the site name.

In addition to the site name, the station file contains the serial number of the BDR and the date and time of the last data collection for each of the data tables in the BDR.

1.1.2 OPTION R - RENAME STATION, COLLECT ALL

The user may rename the station if desired. This is useful if the BDR is moved to a new location or if the user decides to use one of the parameter switches (/B, /N or /S, Section 3). If the name is changed, all data are collected.

1.1.3 OPTION D - DOWNLOAD PROGRAM TO DATALOGGER

The D option programs the BDR320 from a disk file. The type of file needed is a .DLD file saved with the S option (Section 1.1.4) after Prompt Programming or created with EDLOG3. If a file is not found, TERM3 terminates option D. Entering '[path]*' lists the .DLD files.

Pressing ESC at any point will break the TERM3 programming sequence. After datalogger receives the entire file, the program is compiled. If no compile errors are detected, the BDR runs the program.

After programming, TERM3 returns to the options menu. The user may monitor the datalogger operation (option M), or send the disconnect command (option Q), as desired.

If a datalogger compile error is found, TERM3 erases the program and returns to the options menu.

1.1.4 OPTION S - SAVE PROGRAM FROM DATALOGGER

The S option saves the selected BDR320 program on a disk file. This allows the user to program the BDR in the Prompt programming mode, save the program on disk and download it to another BDR320. The user specifies the filename. Enter '[path]*' for a list of existing .DLD files. If no extension is provided, TERM3 adds the extension .DLD to the filename. When selecting the S option, the datalogger continues logging while TERM3 enters the telecommunications command mode. The program is displayed on the monitor as it is received. The file is automatically closed when the program has been received.

Lastly, the monitor displays the status information, the error count and the message "Saving program completed." and returns to the options menu.

1.1.5 OPTION M - MONITOR INPUT LOCATIONS

The M option monitors up to 33 of the first 254 input locations and allows the user to see or change the status of datalogger's flags or control ports. The first 11 input locations and labels will be displayed on the screen. The displayed input locations can be changed by keying **L**. When TERM3 prompts "Locations:", the desired location numbers separated by commas can be specified. For example, 1..33 would display the first 33 input locations, or 1..11,22..32,89..99 would display locations 1 through 11, 22 through 32 and 89 through 99. Entering commas with no number between them (e.g., 1..5,,7..11) will leave a blank space on the screen.

The user flag status is shown below the location data. Set flags are displayed in reverse video. Keys F1 through F8 toggle user flags 1 through 8, respectively (i.e., If the flag is set, the function key will reset it).

The datalogger time is shown below the flag status.

The digital resolution of the location data displayed will default to 3 digits; by pressing the **D** key, an additional digit will be displayed. A maximum of six digits can be displayed.

While in the monitor option, a user can get into the terminal emulator option by pressing the **T** key. The top portion of the screen will have a window in which three lines of the terminal emulator are displayed. While in this mode, the input locations on the screen will not be updated until the Terminal Emulator option is exited by pressing **Esc** or **Ctrl_**.

TERM3 will remember the locations and number of digits displayed when the monitor mode is selected again, unless a new program has been downloaded.

The numbers 1 through 4 are displayed to the right of datalogger time port status. When a port is set high (+5 volts), the corresponding number will be displayed in reverse video; the user can set port 1 high, or low, by pressing **P** followed by **1**. When **P** is pressed, a **P** will be displayed on the screen indicating that TERM3 knows that portion of the command has been given. The C1 port is always an output. C2, C3 and C4 are inputs unless the program in the BDR320 changes them to outputs by executing one of the program control instructions with a command to set the port high or low (Section 8.4). TERM3 can only toggle ports that are outputs.

To load a fixed value into an input location, press the **I** key; a prompt at the bottom of the screen will read "Change Input Location value 11". Input location eleven will be displayed in reverse video when the monitor mode is entered. To enter a new value in location eleven, simply enter the number after the prompt and press **Return**. The new number is then entered into input location 11. To load other input locations, enter the input location then a colon and the value to be entered and then press return. Several input locations can be loaded with one command by separating each input location and value with a comma. Up to 100 characters can be entered in one command. To load input locations 12, 14 and 15 with values 55, 1 and 100 respectively, the following key strokes would be given: 12:55,14:1,100. Data values separated by commas will be loaded at consecutive locations. The last input location to be loaded will then be displayed in reverse video.

The rate at which displayed data is updated depends upon how many values are monitored, selected baud rate, and the computer's speed.

The M option is exited by pressing **ESC** or will exit automatically if communication with the datalogger is broken.

1.1.6 OPTION P - PROMPT PROGRAMMING

Prompt programming simplifies programming requirements for routine BDR320 applications. See the Overview and Section 2.

1.1.7 OPTION T - TERMINAL EMULATOR

Keyboard entries are transmitted out the COM port and characters received are displayed on the monitor. The BDR's computer commands (Section 9) are entered via the terminal emulator option. Direct programming (Section 5) can be accomplished in the terminal emulator option; however, it's much easier to use EDLOG3 for this purpose. **ESC** or **Ctrl_** returns to the options menu.

1.1.8 OPTION A - STATUS

When **A** is keyed, the BDR320 sends the following status information:

Time mm/dd/yy hh:mm:ss	= Real time
NextExe hh:mm:ss	= Time of next program or calibration execution
Battery:+xx.xxx	= Datalogger power supply voltage
Input Locs:xxxx	= Input Locations allocated
Prgm Bytes:xxxx	= Program bytes used
Storage:+xxxxx.	= Storage locations available for Data Tables
Unused:+xxxxx.	= Storage locations not allocated
Data Tables:xxxx	= Number of Data Tables in program
Prgm Sig:+xxxxx.	= User program signature
EE Sig:+xxxxx.	= EEPROM (calibration) signature
EE Ver:xx	= Calibration version
Cal:mm/dd/yy	= Date of calibration
PROM Sig:+xxxxx.	= PROM signature
Errors = xxxx	= Total number of runtime errors since erasing error log
PROM ID:+xxxxx	= PROM identification number
OBJSrlNo	= PROM revision
SN:+xxxx.x	= datalogger serial number
Cxxxx	= Checksum

1.1.9 OPTION K - CLOCK VIEW/SET

The K option allows the user to set the BDR320's clock using the computer's clock. The datalogger time and the PC time are displayed. TERM3 asks:

Set datalogger time to PC time? (Y/N):

If **Y** is keyed, the clock is set and TERM3 returns to the options menu.

If **N** is keyed, TERM3 returns to the options menu.

The clock is always assumed to be correct. Time tags for data records are calculated from the clock time at which the most recent record was recorded. Thus, the next record stored after changing the clock (by more than the recording interval) will change the times calculated for all records in the data table. The Time Set Log (Section 9.7) records the eight most recent times that the clock has been set.

Errors are logged if the time between records is different from the user set interval, and the time has not been changed by the user.

1.1.10 OPTION Q - QUIT

Quit TERM3 by typing **Q**. When TERM3 ends, the COM Port and its interrupt are completely disabled. Subsequent programs must reinitialize the Port before using it (e.g., if the Port is used for a serial printer, the MODE command must be issued before the DOS PRINT command will address the COM Port).

1.2 COMMAND LINE PARAMETERS

TERM3 allows command line parameters. Command line parameters can be entered in any meaningful order. For example, the command

TERM3 D d:\path\program M Q

would download the designated program file, then monitor its input locations with labels from the program file. When ESC or Ctrl_ is pressed or communication with the datalogger is broken, then TERM3 will Quit. Separate each command line parameter with a space. The command line can be up to 128 characters long. The command line can be saved in a DOS batch file (extension **.BAT**) for repeated execution. Commands like **C** can be spelled out to **CALL**, or **D** to **DOWNLOAD** on the command line. Only the first letter of the word is used by TERM3.

SECTION 2. PROMPT PROGRAMMING

Prompt programming simplifies programming requirements for routine BDR320 datalogger applications. The datalogger prompts for user input and generates the program. Prompt programming is designed for simple programs that measure, process, and store data. If more programming flexibility is required, i.e., subroutines, conditional output, set point controls, etc., refer to Direct Programming, Section 5.

> ***CAUTION:*** *The two methods of programming, i.e., prompt programming and direct programming (Sections 2 and 5), can not be mixed. A program generated in the direct method can not be edited in the prompting version. The reverse is also true.*

2.1 PROMPT PROGRAMMING OVERVIEW

The prompt programming main menu is displayed by pressing **P** in the TERM3 options menu.

```
I      -  Input Tables
D      -  Data Storage Tables
E      -  Erase all tables
ESC -  Quit
Select    I,D,E,ESC
```

Programming is divided into input and output programming. Input programming is done in Input Tables (option I) which define:

- The interval at which measurements are made.
- What measurements are made on which input channels.
- Where measurement results are stored.

Output programming is done in Data Storage Tables (option D) which define:

- Which processes (average, maximum, sample, etc.) to apply to the measurement results.
- The interval at which processed data are stored, i.e., the time period over which a process is applied.

The E option erases all programming, resulting in the loss of all stored data.

ESC exits the Prompt Programming mode, and returns operation to the terminal emulator command state. If a program is entered or edited, the datalogger will compile the program and erase all stored data.

2.2 INPUT TABLES

All inputs to the datalogger are programmed in Input Tables. Entries include: measurement type, channel, and where the results are stored.

Two Input Tables are available; either or both may be used. Table 2 offers the same programming capabilities as Table 1, with two additional features:

- Set the C1 control port high a programmable number of seconds or minutes prior to the start of a measurement interval.
- Reduce measurement noise by making multiple measurements, calculating the average and storing the result in the designated LOCation.

The main reason to have two Input Tables is to measure two groups of sensors at different intervals. If all sensors are to be measured at the same interval, and one of the unique features of Table 2 is needed, all programming should be done in Table 2.

2.2.1 TABLE NUMBER AND MEASUREMENT INTERVAL

Type **I** to program input tables.

The default display for the I option is the header for Input Table 1:

```
Input Table Number 01
Measurement Interval mins 0005
```

Input Table Number
If a program is present, the display advances to the header for the Input Table with the largest number. To edit the other table, enter the other table number.

Measurement Interval
The default interval is five minutes. This means the measurements in the table will be made every five minutes synchronized to the BDR clock (e.g., at 12:00, 12:05, 12:10, etc.,). With the cursor on "mins", strike the computer space bar to change the interval units to seconds, as shown below.

Input Table Number 01
Measurement Interval secs 0005

The maximum interval is 1440 minutes and the minimum interval is one second.

2.2.2 INPUT TABLE 2 HEADER

Table 2 is accessed by entering a **2** in the "Input Table Number" line. The default Table 2 header is shown below.

Input Table Number 02
Measurement Interval mins 0005
Ctrl Port prior turn on mins 0001
Duration of Average (s) 0000
# of Samples in Avg. 0001

In addition to the measurement interval, which has the same meaning as for Table 1, there are parameters to allow turning on control port C1 prior to the measurements and averaging several measurements for the reading. Table 2 is identical to Table 1 if the control port entry is set to zero and the values for duration of average and number of samples are default, as shown below.

Input Table Number 02
Measurement Interval mins 0005
Ctrl Port prior turn on mins 0000
Duration of Average (s) 0000
# of Samples in Avg. 0001

Cntrl Port prior turn on
This parameter specifies the number of seconds or minutes before the start of the measurement interval to set the control port C1 high. For example, the dissolved oxygen sensor in the USGS Mini-Monitor must be activated one minute before measurements to allow it to equilibrate. The interval must be less than the measurement interval. Note that units for the "Cntrl Port prior turn on" will change to remain the same as the "Measurement Interval" units. Control port C1 is automatically set low after program execution. Refer to Appendix C for relay driver information.

Duration of Average and # of Samples in Avg.
The average of several back to back measurements is often more representative of the measured parameter than one measurement, e.g., a stage measurement made without a stilling well.

The entry for "Duration of Average" specifies the time over which the average is taken. The entry for "# of Samples in Avg." specifies how many measurements to make within that time. For example, if the duration is 20 seconds and the number of samples is 10, the average is the result of 10 measurements with two seconds between each measurement.

The following constraints are placed on these parameters:

- The "Duration of Average" cannot exceed the Measurement Interval of either Table 1 or 2. (This short term averaging feature is to reduce noise in "one" measurement; for a long term average use the AVG processing function in the Data Storage Tables.)

- The time interval between measurements is in even seconds or, if less than one second, in tenths of a second. If the duration divided by the number of samples does not result in an even interval, the interval is rounded down and the duration of average will actually be shorter than the value entered.

  Example #1 - Assume the duration is 20 seconds and the number of samples is 11. Duration/samples is 1.82 seconds. The actual interval between measurements is one second, and the actual duration is 11 seconds.

  Example #2 - Assume the duration is two seconds and the number of samples is seven. Duration/samples is 0.28 seconds. The actual interval is 0.2 seconds and the actual duration is 1.4 seconds.

2.2.3 MEASUREMENT SELECTION

After completing the table header, the table of measurements is displayed.

Loc	Name	units	Type	Chn	Mult	Offset
01			OPT?			

This is where measurements are assigned to channels, and multipliers and offsets are entered to convert the measurements to engineering units. Labels and units may be assigned to the values measured.

Loc

The Loc column contains the Input Storage location in which the measurement is saved. As measurements are added, the location is automatically incremented.

Name

The name may be up to a six characters long and should be descriptive of the parameter being measured. The name entered here is displayed at the top of the data column when retrieving data, making the data point easy to identify. For example, "STAGE" can be used for a stage height measurement.

Units

Up to three characters may be used to describe measurement units. For example, "FT" may be the unit of the stage measurement.

Type

"Type" refers to the type of sensor to be measured or type of measurement to be made. Strike the space bar to scroll the measurement OPTions listed below. The "B" key backs up to the previous option.

SDI

SDI-12 Sensor

This measurement addresses, issues a command, and retrieves data from a SDI-12 sensor.

Some SDI sensors output more than one data point per output. For example, the Hydrolab SCOUT may be programmed to output nine datapoints each time the datalogger requests measurements. The SDI measurement type must be entered as many times as data points are desired. For example, assume the SCOUT is equipped to make temperature, conductivity, dissolved oxygen and pH measurements each time the datalogger calls it. The SDI measurement must be entered four times, once for each parameter. Refer to the SDI channel ("Chn") description below for further information.

POT

Potentiometer
Input Range = -90 to 5000 mV
Units of measurement = Ratiometric, Vs/Vx
Vs= signal voltage; Vx= excitation voltage

The most likely sensor to be used with this instruction is the 10 turn potentiometer commonly used for float pulley retrofits. POT is a half bridge measurement also used to measure linear thermistors or wind vanes. Single ended analog channels CH1 through CH4 can be used with this instruction.

BATT

Battery voltage
Units of measurement = Volts

The datalogger power supply with the highest potential, i.e., internal or external, is measured using this instruction. Units are in volts. None of the analog channels are used in this measurement.

TEMP

Datalogger Temperature
Units of measurement = °C

This instruction measures the datalogger temperature in degrees Centigrade. None of the analog channels are used in this measurement.

THRM

103, 103B Thermistor
Units of measurement = °C

Campbell's 103 and 103B temperature probes are measured using this instruction. Single ended analog channels 1 through 4 can be used with this instruction.

CNT

Pulse Count
Input Range = 5000 mV maximum
Units of measurement
Chnl 1 = counts
Chnl 2 = frequency; Hz

This instruction measures pulses on pulse count channels P1 or P2. Channel 1 should be limited to slow or infrequent pulse inputs such as a tipping bucket rain gage. Channel 2 is a fast pulse counter, designed for a contact closure flow meter or a tipping bucket rain gage. Channel 2 requires less power to operate than channel 1.

SE

Single-ended Voltage
Input Range = autorange; -20 to 5000 mV
Units of measurement = mV

The SE instruction makes single-ended voltage measurements. The analog output version of the Mini-Monitor or pressure transducers with single-ended outputs are typical sensors. Single-ended analog channels 1 through 4 can be used with this instruction.

DIFF

Differential Voltage
Input Range = autorange; -20 to 5000 mV
Units of measurement = mV

DIFF is a differential voltage measurement, applicable to differential channels 1 (CH1/CH2) and 2 (CH3/CH4). The same sensors measured using SE can be measured with less noise using DIFF.

FULL

Full Bridge
Input Range = -20 to 80 mV
Units of measurement = Ratiometric, 1000(Vs/Vx)
Vs= signal voltage; Vx = excitation voltage

The full bridge measurement applies to differential channels 1 (CH1/CH2) and 2 (CH3/CH4). A strain gage pressure transducer is the most likely sensor to be used with this instruction.

6 WIR

6 Wire Full Bridge
Excitation Input Range = autorange
Bridge signal input range = autorange
Units of measurement = Ratiometric 1000 (Vs/Vx)
Vs = signal voltage; Vx = excitation voltage

The 6 wire full bridge is used to measure a strain gage pressure transducer with a long cable from the transducer to the BDR320. Measuring the excitation voltage compensates for voltage drop in the cable. Differential input 1 (CH1/CH2) is used to measure the excitation, and differential input 2 (CH3/CH4) measures the output from the transducer.

Chn

For analog and pulse measurements, "Chn" is the abbreviation for channel. The sensor connected to the channel defined here is measured by the instruction specified under "Type". Single ended measurements which apply to single ended channels 1 through 4 are POT, THERM, and SE. Differential channels 1 (CH1/CH2) and 2 (CH3/CH4) are measured with measurement types DIFF, FULL, and 6 WIR. The pulse count channels are measured with CNT.

"Chn" for SDI measurements has a different meaning. Assume a three character entry, "ACV", where "A" is the sensor address, "C" is the command, and "V" is the value.

- The sensor address, A, may be any integer between 0 and 9, inclusive.

- The command, C, may be any whole number between 0 and 9. The command code is specific to the SDI sensor, but in all cases code 0 means start measuring.

- Value, V, defines which data point should be stored in the specified Location. Most SDI-12 sensors output only one or two data points. To store the first value returned, enter **1**. Enter **2** to store the second value returned. If both values are to be stored, the SDI measurement must be entered twice, with the same address and command, but changing V each time. The example below is for a Hydrolab Scout that is to output temperature, conductivity, dissolved oxygen, and pH when commanded to measure. The address and measurement command remain the same but the value to store (V) changes with each measurement line.

Loc	Name	units	Type	Chn	Mult	Offset
01	temp	C	SDI	001	+1.0000	+0.0000
02	d.o.	ppm	SDI	003	+1.0000	+0.0000
03	pH		SDI	004	+1.0000	+0.0000
04	cond	mmh	SDI	002	+1.0000	+0.0000

NOTE: The above example illustrates the use of "V", the value parameter. The order of output from the Scout is temperature, conductivity, dissolved oxygen, and pH, but the order of storage in the datalogger is temperature, dissolved oxygen, pH, and conductivity.

Mult
The multiplier is the slope of a straight line equation. The measurement is multiplied by this value to convert the measurement to the desired engineering unit. Before calculating the multiplier, determine what units the measurement type returns.

Multiplier Example #1
The sensor is a 10 turn, 10 kOhm potentiometer with a one foot circumference wheel. With a one foot wheel at 1000 Ohms per foot, the range goes from 0 to 10 feet. The measurement type is POT, which returns the ratio of the signal voltage to the excitation voltage, or Vs/Vx.

The multiplier, M, is (10-0)/(1-0), or 10. The program for this example is shown below.

Loc	Name	units	Type	Chn	Mult	Offset
01	stage	ft	POT	001	+10.000	+0.0000

Multiplier Example #2
The sensor is a strain gage pressure transducer with a calibration of 10 mV per volt of excitation at 50 feet of water. The measurement type is FULL, which returns 1000 times the ratio of the signal voltage to the excitation voltage (1000*Vs/Vx). Vs and Vx are in millivolts, making 1000*Vs/Vx equal to millivolts per volt of excitation.

The multiplier, M, is (50-0)/(10-0), or 5. The program for this example is shown below.

Loc	Name	units	Type	Chn	Mult	Offset
01	stage	ft	FULL	001	+5.0000	+0.0000

Offset
The offset is the Y-intercept. After the measurement is scaled according to the multiplier, the offset is added to adjust the measurement to a known reference, i.e., the outside staff gage. The easiest way to enter the correct offset is through the Y command (Section OV5.5, 9.10). The Y command does not work with the THERM or CNT measurement types.

If the offset must be entered directly into the Input Table, exit the Prompt Programming mode with the ESCape key to allow the datalogger to compile the program. With the offset equal to zero, view the measurement in the designated LOCation using the "n,m,l" command. Calculate the offset as follows:

Offset = Reference - Measurement

Re-enter the Prompt Programming mode, advance to the desired program line in the Input Table and enter the offset.

2.3 DATA STORAGE TABLES

Data Storage Tables (option D) are used to specify:

- How often to store data.
- What data to store.
- What process (average, maximum, sample, etc.) to apply to a particular measurement.

2.3.1 DATA STORAGE TABLE HEADER

The default display for the D option is shown below.

Data Table Number 01
Recording Interval mins 1440

Data Table Number
The display advances to the largest number Data Table, providing information on how many Data Tables are programmed. To display a different table, enter the number of the desired table. Up to nine Data Tables may be programmed.

Recording Interval
The recording interval is the period in which records are stored, i.e., a 60 minute recording interval stores records every hour on the hour. The default interval is one day (1440 minutes). With the cursor on "mins" strike the computer space bar to change the interval units to seconds, as shown below.

Data Table Number 01
Recording Interval secs 1440

The maximum interval is 1440 minutes and the minimum is one second. The recording interval should be an even multiple of the measurement interval(s) of corresponding sensors.

2.3.2 DATA TABLE INSTRUCTION SELECTION

Enter the Recording Interval (above) to display the next level of Data Table programming:

Item	Loc		Process
01	01	(name)	OPT?

Item
Item is a count of how many data points are in a record. Item automatically increments with each output selected.

Loc
The Loc column contains the Input Storage locations in which the measurements programmed in Input Tables are saved. Strike the space bar to display different input locations. The name of the measurement value programmed in Input Tables is automatically displayed to the right of input location.

OPT?
Strike the space bar to display the processing options which can be applied to the measurement value over the specified recording interval. The "B" key backs up to the previous option. Options include:

AVG (average)
SMPL (sample)
MAX (maximum)
MIN (minimum)
TOTAL (totalize)
SD (standard deviation)

Programming Example #1
Assume the sensor is a 10 turn potentiometer with float, and hourly samples of stage are to be recorded. The Input Table and Data Table programming are given below.

Input Table Number 01
Measurement Interval mins 0060

Loc	Name	units	Type	Chn	Mult	Offset
01	STAGE	ft	POT	001	+10.000	+0.0000

Data Table Number 01
Recording Interval mins 0060

Item	Loc		Process
01	01	STAGE	SMPL

Note that the offset is not entered in the Input Table. After exiting the Prompt Programming mode, use the "Y" command (see Section 4) to have the datalogger calculate the offset and enter it in the Input Table.

Following the entry of a processing option, another program line is added, as shown below.

Item	Loc		Process
01	01	STAGE	SMPL
01	01	STAGE	OPT?

If more than one measurement is programmed, i.e., more than one input Location requires processing, a different location may be displayed by striking the space bar when the cursor is under Loc.

Item	Loc		Process
01	01	STAGE	SMPL
01	02	(name)	OPT?

All of the locations programmed in Input Tables can be displayed by repeatedly striking the space bar.

Programming Example #2
Assume an SDI shaft encoder and Campbell's Temperature Probe are measured every 15 minutes. Two Data Tables are programmed to record:

- 15 minute sample of stage.
- 24 hour average, maximum, and minimum stage, average temperature and sample of battery voltage.

Input Table Number 01
Measurement Interval mins 0015

Loc	Name	units	Type	Chn	Mult	Offset
01	STAGE	ft	SDI	001	+1.0000	+0.0000
02	TEMP	C	THRM	004	+1.0000	+0.0000
03	BATT	V	BATT			

Data Table Number 01
Recording Interval mins 0015

Item	Loc		Process
01	01	STAGE	SMPL

Data Table Number 02
Recording Interval mins 1440

Item	Loc		Process
01	01	STAGE	AVG
02	01	STAGE	MAX
03	01	STAGE	MIN
04	02	TEMP	AVG
05	03	BATT	SMPL

Note in the Input Table under "Chn" that the SDI sensor has an address and command of zero and is storing the first measurement returned by the sensor. Note also that the temperature probe is measured on single ended channel 4.

2.3.3 MEMORY ALLOCATION

The datalogger allocates memory for data storage so that the Data Tables fill up their memory at about the same time. In this way, records from all tables cover the same time period. If only one Data Table is programmed, memory is allocated to store as many complete records as possible. The amount of time before memory is full is displayed after exiting the Prompt Programming Mode. For example, given Example Program #2, after exiting the W Mode the display reads

Storage filled in +304.80 days
Exiting programming mode
H = Help; errors:0000

2.4 EDITING

Enter advances to the next location requiring input.
Right arrow (Ctrl D) for forward.
Left arrow (Ctrl S) for back.
Up arrow (Ctrl E) for up.
Down arrow (Ctrl X) for down.
ESC backs out to previous programming level.

When using the ESC key to exit the Prompt Programming mode, and a programming error is detected, the following sequence of steps occur:

- The screen is cleared.
- The error is displayed.
- Strike any key to return to the Prompt Programming mode main menu.
- Select the correct option to go and fix the error or strike ESC return to TERM3's options menu.

If the wrong instruction Type is entered, it may be changed. It is not possible to delete an instruction from the middle of an Input Table, but no measurement will be made if "OPT" is selected for the Type.

SECTION 3. DATA RETRIEVAL WITH TERM3

Data are retrieved within TERM3 by selecting either the U - get Uncollected data, or the R - Rename station collect all option. Three types of files are created in your PC during the data collection process: data files, a log file , and a "STNNAMES" file. The data files contain the data transferred from the BDR320. The log file contains data notes, information on clock updates, and an error log. This section deals with the details of data format and management of the BDR320 and PC300 software.

3.1 COLLECTING DATA

The U option, **get Uncollected data**, is used to collect data from the datalogger. The first time data are collected, TERM3 prompts as follows:

Adding Serial No. XXXX to the file STNNAMES

Default collection parameters:
ASCII file
Append data to existing file
Output time as MM/DD/YY HH:MM:SS
Site name suffix parameter switches:
/B binary collection
/N start new collection file each time
/S output time as seconds since midnight, 1 Jan 89

Enter site name of 1 to 6 characters plus optional switches:_

The optional switches allow:

/B; Collection of data in binary format (more compact, but must eventually be converted to an ASCII file using the BTOA software; Section 4.2)

/N; A new file is created each time data is collected (suffix of file name is incremented; Section 3.2).

/S; Output time as seconds since midnight, 1 Jan 89. This option is useful if you wish to save a more compact representation of time (Section 3.2).

Subsequent data retrieved using the U option will be appended to these original files using the same format options. (See 3.3) The default collection parameters will retrieve date-time based data to ASCII files.

The R command is used to Rename a station, collect all data, and change the optional switches if desired.

3.2 DATA FILES

Assume there are four data tables created by the program in the datalogger with serial #1219. When TERM3 is run, the U option collects and stores data in the following files:

TEST-1.001
TEST-2.001
TEST-3.001
TEST-4.001

3.2.1 FILE NAMES

The data file names have the format

ssssst.nnn

where:

ssssss serial number or name for site with that serial number. If the name or serial number is less than six characters, a dash (-) is appended. If no site name is entered the serial number is used.

t data table number

nnn 001 to 499 for ASCII data, 500 to 999 for binary data. When the /N option is selected, nnn is incremented each time data are collected.

TABLE 3.2-1. Output Data File

"SN:+1219.0 Table:0002 Time:10/22/90 14:20:00 Interval:0005 min"

"M""D""Y"	"H""M""S"	"Level "	"Level "	"Level "	"batt"	"temp "
"M""D""Y"	"H""M""S"	"in___A"	"in___H"	"in___L"	"V____S"	"C____S"
10/22/90	14:20:00	16.75	24.87	20.36	11.3	24.56
10/22/90	14:25:00	22.64	24.91	20.25	11.34	24.71
10/22/90	14:30:00	23.4	20.53	20.2	11.29	24.83
10/22/90	14:35:00	21.74	23.28	20.51	11.35	24.9
10/22/90	14:40:00	23.9	24.45	23.23	11.35	24.86
10/22/90	14:45:00	24.36	24.56	24.06	11.36	24.85
10/22/90	14:50:00	24.1	24.5	23.09	11.36	24.85
10/22/90	14:55:00	23.54	24.2	23.08	11.36	24.87
10/22/90	15:00:00	24.5	24.67	24.17	11.36	24.89

3.2.2 ASCII FORMAT

Figure 3.2-1 is an example of the general format of data files. The information in the column headings consists of a name, units and the processing used.

The processing symbols are interpreted as follows:

Symbol	Processing
S	Sample
A	Average
T	Total
H	Maximum (High)
L	Minimum (Low)
D	Standard Deviation

3.2.3 /S SWITCH; LOTUS 123 R DATE FUNCTION

The /S switch can be used to output the time in seconds since midnight, January 1st 1989. To convert the seconds to Lotus DATE.TIME format, divide the seconds by 86400 and add to @date(89,1,1). Refer to your Lotus Manual for details in using the @DATE and @TIME functions.

From the default time format of MO/DD/YY HR:MM:SS, the Lotus DATE.TIME format can be derived with @DATE(YY,MO,DD) + @TIME(HH,MM,SS), where the function parameters are the cell addresses of the time elements.

3.3 THE LOG FILE

The Log file contains data notes, the Time Set log, and an Error log. The Log file is named by placing "L" in front of the station name. Table 3.3-1 is an example of a Log file.

Table 3.3-1 Log File

SN:1221
NotesLog:
04/19/91 10:41:42 THE STAGE READING WAS 0.05 FEET HIGH BEFORE THE OFFSET WAS ADJUSTED.

*

TimeSetLog:

01/01/89 02:20:28 04/19/91 10:39:33

*

Errorlog:

*

3.3.1 NOTES LOG

The Notes log consists of the data notes entered by users (4321S command, Section 9.8).

3.3.2 TIME SET LOG

The Time Set log lists the last eight times the clock was set. The format is:

Previous Time	Time Set To
01/01/89 02:20:28	04/19/91 10:39:33

3.3.3 ERROR LOG

The Error log lists the number of run time errors that have occurred and the first and most recent time at which each occurred. The format of the Error log is:

cc nn mo/day/yr hr:min:sec mo/day/yr hr:min:sec

where:
cc = error code

nn = number of occurrences of the error
mo/day/yr hr:min:sec = time of the first and the most recent occurrences of error

Compiling and editor errors are displayed on the computer screen, but are not stored in the error log. (Sections 3 and 9)

ERROR LOG CODE DEFINITION

01	EEPROM coefficients not programmed
02	Measurement hardware overranging
03	Unsuccessful datalogger calibration
04	Internal thermistor out of range
05	Start of low power supply condition
06	End of low power supply condition
07	Signature of PROM, program, and EEPROM does not match value stored. (New signature is then stored.)
08	Datalogger reset by watchdog timer
09	Insufficient storage space
10	Outer subroutine called from nested inner subroutine
11	Program table overrun
12	Output did not occur at programmed interval
13	Power-up test failed
14	1) Clock set back. 2) Watch dog or overrun where time difference is greater than 255 sec.

Once a day at midnight, the signature of the PROM, program, and EEPROM is calculated and compared with the value it has stored. If the calculated and stored value do not match, error 7 is logged and the newly calculated signature is stored.

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 hardware problem or a software bug and should be reported to Campbell Scientific.

When storing data, error 12 is logged if the time between records is different from the interval set in Instruction 84, and the time has not been changed with the C command. Resetting the datalogger clock does not cause error 12.

Error 13 indicates that a fault in RAM, PROM, or CPU memory was detected on power-up. DO NOT CONTINUE TO USE THE DATALOGGER IF THIS ERROR OCCURS. Contact Campbell Scientific to discuss the problem and arrange for the repairs.

An example error log is:

ErrorLog:
E02 99 04/20/89 09:45:00 05/16/89 13:15:00

The example shows that the measurement hardware, error 2, has overranged 99 times or more between April 20 at 9:45 AM and May 16 at 1:15 PM. The maximum number of occurrences that may be recorded is 99.

3.4 THE STNNAMES FILE

The STNNAMES file contains file management information for TERM3. On execution, TERM3 calls the connected BDR320 and looks at the STNNAMES file for the BDR's serial number and the associated site name. These are displayed on the right side of the upper window of TERM3. If the serial number is not in STNNAMES, it is added. TERM3 will create a STNNAMES file if it does not find one.

For example, on the first collection, the site name is entered as "TEST" (no switches set); TERM3 creates the file STNNAMES containing the following information for datalogger serial number 1219:

TABLE 3.4-1 First STNNAMES File

```
1219 13375 TEST
1 001 10/10/90 13:50:01
2 001 10/10/90 13:50:06
3 001 10/10/90 13:50:11
4 001 10/10/90 13:50:17
```

The STNNAMES file above contains information on the BDR320 Station and the files created for data tables one through four. The first line contains the serial number of the BDR (1219), the program signature (13375), the site name (TEST), and any data retrieval switches used. The remaining rows begin with the number of the data table retrieved. The second element is the number of files collected from the corresponding table. The third and fourth elements show the date and time from which TERM3 will collect data when the **get Uncollected data** option is specified.

3.4.1 APPENDING UNCOLLECTED DATA

If the STNNAMES file already exists with the serial number and the correct program signature, only data since the last retrieval are collected. Data are appended to the existing data files corresponding to that site. On the next collection, STNNAMES is updated to:

TABLE 3.4-2 STNNAMES File After Second Collection

```
1219 13375 TEST
1 001  10/10/90  16:03:12
2 001  10/10/90  16:03:21
3 001  10/10/90  16:03:23
4 001  10/10/90  16:03:30
```

As new data are *appended* to the existing files, the only thing that changes in STNNAMES is the time.

3.4.2 NEW FILES EACH COLLECTION /N

If the /N parameter switch is selected when the site is named (TEST/N), new data are stored in separate files, and STNNAMES updates the extension number each time data are retrieved:

TABLE 3.4-3 STNNAMES With /N Collection

```
1219 13375 TEST/N
1 002  10/10/90  16:23:12
2 002  10/10/90  16:23:12
3 002  10/10/90  16:23:13
4 002  10/10/90  16:23:13
```

The second element of line two now shows that 002 files of data have been collected from table one. The following is a list of data files collected from datalogger TEST:

TEST-1.001
TEST-2.001
TEST-3.001
TEST-4.001
TEST-1.002
TEST-2.002
TEST-3.002
TEST-4.002

Two files exist for each BDR output table. TEST-1.001 contains all the records from data table one, collected with the initial U command. TEST-1.002 is a continuation of the data from table one that was collected with the second U command. In the second collection, TERM3 used the Table 3.4-1 STNNAMES file and saw that it needed to get all the data in table one that was recorded after 10/10/90 13:50:01. TERM3 then wrote the Table 3.4-3 STNNAMES file.

3.4.3 RENAMING FILE/PROGRAM CHANGES

The R data collection option creates a new entry in STNNAMES. New data files are created which contain all data in the BDR320.

If the R command was given when the Table 3.4-1 STNNAMES existed, STNNAMES would be changed to:

TABLE 3.4-4 STNNAMES With Renamed File

```
1219 0 TEST
1 001  10/10/90  13:50:01
2 001  10/10/90  13:50:06
3 001  10/10/90  13:50:11
4 001  10/10/90  13:50:17
1219 13375 TEST
1 002  10/10/90  16:23:12
2 002  10/10/90  16:23:12
3 002  10/10/90  16:23:13
4 002  10/10/90  16:23:13
```

The number between the serial number and the site name (13375 in this case) is the signature of the BDR program. This signature will change if any changes are made to the program (data memory is reset), or if an offset is adjusted (data memory not reset). This number has been set to "0" by TERM3 for the first collection entry and the associated time and name is thus recognized by TERM3 as not being current.

> **NOTE:** Adjusting an offset changes the program signature. On the next collection, TERM3 will collect all data.

The **get Uncollected data** option will collect all data if a STNNAMES file does not exist, the serial number of the BDR320 is not listed in STNNAMES, or the program signature has changed. In any of these cases, it is usually desired that all data be collected. If you do not want all the data, i.e., you only adjusted an offset, then change the STNNAMES signature to the new signature with a text editor. The new signature is displayed with the status command or found in the log file.

3.4.4 MODIFYING STNNAMES WITH A TEXT EDITOR

With an ordinary text editor, the collection times may also be altered. Deleting a table from the list will prevent TERM3 from collecting data from that table. However, if all data tables are deleted from STNNAMES, option R or U will reinstate collection from all tables.

For example, if Table 3.4-3 STNNAMES is modified by deleting table 1 and changing the collection time for data table 4 to 16:30, data from table 1 will not be collected and data from table 4 will start at or after 16:30 (Table 3.4-5).

TABLE 3.4-5 Modified STNNAMES

```
1219 13375 TEST/N
2 002 10/10/90 16:03:21
3 002 10/10/90 16:03:23
4 002 10/10/90 16:30:00
```

Thus, using option U will result in the following data files (note: the /N switch is still valid):

TEST2.003
TEST3.003
TEST4.003

SECTION 4. DATA RETRIEVAL WITH STORAGE MODULE AND SMCOM

The SM192/716 Storage Modules, with the latest PROM, can collect data automatically from the BDR320. When the storage module is plugged into the datalogger, using the SC3925 cable, the Storage Module wakes up, recognizes the BDR320 interface, and sets itself to the fill and stop mode. The Storage Module then proceeds to retrieve the uncollected data. When data collection is complete, the L.E.D. flashes indicating the Storage Module has retrieved all the uncollected data from the Storage Module.

> ***NOTE:** The Storage Module can not collect data "on line" under BDR320 program control.*

When initiating data collection, the Storage Module looks for the file STNNAMES (stored as program 1). If STNNAMES is not present, the Storage Module collects all data from the datalogger and creates the file STNNAMES in program 1. The Storage Module inserts file marks between each data table. The Storage Module collects the notes, errorlog, and time set log and stores them in a single file.

SMCOM is used to retrieve data files, log files, and STNNAMES from the Storage Module. SMCOM communicates with the Storage Modules through an asynchronous communications adapter at 19,200 baud.

Either the SC532 Storage Module Interface or the PC201 Card with the SC209 cable is required to interface the Storage Module to the PC..

A typical data collection procedure would be as follows:

1. *The first time a Storage Module is put into service, reset it using SMCOM option E.*
2. *Collect data from the BDR320. The Storage Module creates the file STNNAMES in program 1.*
3. *Retrieve data from the Storage Module using the retrieve uncollected data option in SMCOM (option U), in final storage format, F option.*
4. *Use BTOA to rename and convert the binary data files to ASCII format (Section 4.2).*
5. *Clear the Storage Module data with SMCOM (Option C). The STNNAMES file (ROOT1.DLD, ROOT is the "root" filename given by the user) is saved and reinstalled automatically.*
6. *If a different Storage Module is being sent into the field, use SMCOM (option F) to store STNNAMES (ROOT1.DLD) as program 1 in the Storage Module(s) being sent to the field.*
7. *Collect data from the same sites (in any order). The Storage Module uses STNNAMES (program 1) to collect only the uncollected data. If new sites are visited, the site information is added to STNNAMES and all data are collected from the new sites.*

4.1 SMCOM

4.1.1 GETTING STARTED

Enter 'SMCOM' on the command line to load and execute the program. The PC must be logged onto the current drive and directory containing SMCOM, or the appropriate "path" must be specified. SMCOM will prompt:

SMCOM Ver X.X
Copyright (c) 1986, 1987, 1988, 1989, 1990
Campbell Scientific, Inc.
Logan, Utah 84321
Serial Port Options:
1 -- Com1
2 -- Com2
3 -- Com3
4 -- Com4
Option:

Enter in the appropriate number for the communication option that you are using. The computer will prompt:

Is the SM232 or SM232A interface being used? (Y/N):

If one of these interfaces is being used, key **Y** and follow SMCOM's directions. Otherwise, key **N**. SMCOM immediately establishes communication with the Storage Module and puts a status window at the top of the screen. The status will be updated automatically when changes are made. If the proper communication is not established, SMCOM aborts, and gives the message:

SM192/SM716 Storage Module does not respond! Reducing baud rate...

If this happens, verify that all cables are connected properly and that the proper COM port was specified.

4.1.2 SMCOM OPTIONS

The SMCOM OPTIONS menu appears as soon as SMCOM establishes communication with the Storage Module. The options will be displayed as follows:

> **NOTE:** Only options E, U, and C are needed when using the SM192/716 with the BDR320.

SMCOM Options:
T -- Terminal emulator
A -- Collect all data files
U -- Collect uncollected data files
N -- Collect newest data file
L -- Collect one data file starting at display pointer L
P -- Collect program files
D -- Store a .DLD program file
F -- Store a file
E -- Erase and Reset Storage Module
C -- Clear Data Area
S -- Switch Settings
Q -- Quit
Option:

To select an option, key the appropriate letter. When one of the "collect" options is selected, SMCOM will ask for a file name and storage format. In the following descriptions of the options, ROOT refers to the root collection filename which you enter (Section 4.1.3).

T -- TERMINAL EMULATOR
The T option allows the user to communicate directly with the Storage Module. This option allows users to manually issue Storage Module telecommunication commands. Keying F1 while in Terminal Emulator will display a list of Storage Module commands. This option is seldom used because all normal operations can be accomplished by other selections from the options menu. Keyboard entries are transmitted out of the COM port to the Storage Module and Storage Module responses are displayed on the computer monitor. **ESC**ape or **Ctrl_** exits this mode and returns the options menu.

A -- COLLECT ALL DATA FILES
The A option will collect all non-program files in the Storage Module. Normally, these files will contain data that the Storage Module collected when attached to a datalogger; however, the file could also be a text file stored in the Storage Module from another source. SMCOM detects the file separators, called file marks. The oldest file in the Storage Module is saved with the root filename with the next available number. The numbers are assigned sequentially until all data files are collected (Section 4.1.3).

The message "xxxxxx: Writing to file ROOTn.dat." will appear on the monitor for each file collected, where xxxxxx is the beginning location number of the file in the

Storage Module. The location information may be useful if the user wishes to directly examine the file in the Storage Module using the Terminal Emulator option.

U -- COLLECT UNCOLLECTED DATA FILES
The U option functions the same as the A option with the exception that the data is collected from the current position of the Storage Module Dump Pointer. The Dump Pointer is at the beginning of memory after the Storage Module has been reset. Only the A and U options advance the Dump Pointer; other options do not change its position. The U option will collect the files created since the last time the U or A option was executed.

N -- COLLECT NEWEST DATA FILE
Only the most recently stored file is collected with the N option.

L -- COLLECT ONE DATA FILE STARTING AT DISPLAY POINTER
If the user positions the Display Pointer (L) in the Terminal Emulator option, the SMCOM L option can be used to collect data from that location to the end of the file.

P -- COLLECT PROGRAM FILES
The P option collects the datalogger programs stored in the Storage Module. The Storage Module maintains the addresses of up to eight program files. The collected computer files will be named ROOT1.DLD to ROOT8.DLD the number for each corresponds to the program's address in the Storage Module.

D -- STORE A .DLD PROGRAM FILE
The D option is used to store a datalogger program in the Storage Module. SMCOM asks for the computer file name and adds the extension .DLD if not included. SMCOM then sends the file to the Storage Module. The prefix '#n' (n=1 to 8) in front of the file name will direct the file to one of eight program files in the Storage Module (e.g., #3TEST.DLD tells the SMCOM to store TEST.DLD as program 3. If the prefix is not included, the file will be stored as program number one.

F -- STORE A FILE
Non-program text or data files can be sent from the computer to the Storage Module using the F option. SMCOM asks for the file name, then sends the file. If the computer file is an odd number of bytes, the Storage Module will append a null character (ASCII 0 decimal) to the end of the file.

E -- ERASE AND RESET STORAGE MODULE
Key **E**, and after asking if you are sure you want to, SMCOM will send the commands to erase and reset the Storage Module. This option performs a write/read cycle to check all memory. During this test, a "+" is displayed for each RAM chip written to and a "-" is displayed when the chip is successfully read. "X" indicates a chip could not be read. All data and programs are erased and switches are set to the default settings (address = 1, baud = 76800/9600, fill & stop = off, and encode ASCII = on).

C -- CLEAR DATA AREA
The **C** option is used to clear the data storage area of the Storage Module. This option, after asking if you are sure, erases all data stored in the Storage Module. Programs stored in the Storage Module will be retrieved prior to resetting the data storage area. After clearing the data storage area, the programs are reinstalled into the Storage Module. Storage Module switch settings are unaffected by the C option. During this option, a "+" is displayed for each RAM chip cleared. Unlike the E option, the C option does not test memory.

S -- SWITCH SETTINGS
Key **S** to change the switch settings shown at the top of the screen. Starting with Address, the switch setting will be highlighted. Key the spacebar to scroll through the options. Key **Enter** to move to the next switch. Key **ESC**ape when done.

Q -- QUIT
Quit SMCOM and return to the system by typing Q.

4.1.3 NAMING OF DATA FILES ON DISK

When one of the "Collect" options is selected, SMCOM will prompt:

Root collection file name (6 characters max):

The root collection file name is the basis for the file names that will be created by the various collection options. In this manual, "root" refers to the file name entered. Enter '[path\]*' to list files with the extension .DAT.

Data files with a given root name are numbered sequentially. The first file will be root001.DAT. If there are already data files with the same root name on the directory in use, SMCOM will assign the next available numbers. For example, if the root name is TRAIN and files up to TRAIN009.DAT are already on the disk, when two new files are collected from the Storage Module, the files will be named TRAIN010.DAT and TRAIN011.DAT.

4.1.4 DATA COLLECTION FORMATS

SMCOM will ask for a data file format. The options are:

F -- Final Storage (FS) Format
D -- FS converted to ASCII arrays with IDs
C -- FS converted to comma delineated ASCII arrays
A -- As stored (8 bit data)
P -- As stored (strip parity)

NOTE: Use only Format F for BDR320 data.

4.1.5 SMCOM COMMAND LINE PARAMETERS

All options to be executed while SMCOM is active can be entered on the command line. The command line is limited to 128 characters. Each option is separated by a space. Option names on the command line can be entered either as a single letter or as a word starting with that letter. For example, the command line

SMCOM 3 NO UNCOLLECTED TEST FINAL CLEAR QUIT

is equivalent to

smcom 3 n u test f c q.

This command line will execute SMCOM through COM3, not using SM232 or SM232A. All uncollected data files will be collected and stored in Final Storage format using the root file name TEST. SMCOM will then clear the data area, restore STNNAMES, and quit.

For automatic execution, the command line can be executed within a batch file.

4.2 BTOA

The program BTOA is used to convert binary data collected from the BDR320 datalogger to an ASCII format. An ASCII file will import directly into Lotus 123[R] using the "numbers option" or similar spreadsheets.

BTOA is self prompting. Typing BTOA will bring the following up on the computer screen:

Option:Dates Seconds No Dates Headers Rename

BTOA Ver. 2.1
Copyright (c) 1990, 1991
Campbell Scientific, Inc.
Binary to ASCII Conversion
Use Cursor key to select:
(1)Option
(2) Station or file

For this example, the data files STAGE001.DAT and STAGE002.DAT were collected using SMCOM. The procedure to convert these files to ASCII is as follows:

1. Rename STAGE001 and STAGE002 by tagging each file with the space bar and selecting the command line switch **RENAME**. BTOA will ask for a site name (if desired). A six character site name may be specified.

For this example, STAGE was specified for the site name. BTOA renames the files to LSTAGE.001 (LOG FILE) and STAGE-1.501 (DATA FILE).

NOTE: Files with an extension greater than 500 are binary files, and data files with an extension less than 500 are ASCII files.

2. Using the space bar, tag the station file "STAGE". Select the command line "DATE" function, then press **ENTER**. BTOA converts the binary file STAGE-1.501 to the ASCII file

STAGE-1.001. The following is a portion of the converted ASCII file STAGE-1.001:

```
"SN:+1219.0  Table:0001  Time:06/23/92
11:31:00  Interval:0001  min"
"M""D""Y""H""M""S""BATT  ""T BDR ""STAGE "
"M""D""Y""H""M""S""V____S""C____A""FT___A"
06/23/92  11:31:00   11.2    23.37   .88
06/23/92  11:32:00   11.21   23.39   .96
06/23/92  11:33:00   11.22   23.39   .94
06/23/92  11:34:00   11.22   23.4    .9
06/23/92  11:35:00   11.2    23.4    .86
06/23/92  11:36:00   11.19   23.39   .83
```

The file LSTAGE-1.001 is the datalogger notes log. The following is the converted file LSTAGE-1.001:

```
SN:1219
NotesLog:

TimeSetLog:
01/01/89 00:04:26  06/23/92 10:56:35
ErrorLog:
*
```

BTOA has five optional command line switches: DATE, SECONDS, NO DATES, HEADER, and RENAME.

The DATE option results in date and time in the MM/DD/YY HH:MM:SS format as in the previous example.

The SECONDS option outputs time as seconds since midnight on January 1, 1989. This time format is useful if Lotus 123^R is used for data processing (Section 3.2.3).

The NO DATES option outputs data with no time information.

The HEADER option prints data file header information to the screen. This option allows a quick review of the datalogger's various data table collection times. No ASCII conversion takes place.

The RENAME option is used to change named files to their serial number equivalents or from serial numbers to site names. This option must be used on files collected from the Storage Module before attempting ASCII conversion.

If a binary file has already been converted to an ASCII file, BTOA will not reconvert the file. The ASCII file must be deleted prior to reconverting a file from binary to ASCII. This function allows additional binary data files to be added to a directory without BTOA reconverting existing files.

A command line can be entered at the DOS prompt to convert individual data files. For example, entering BTOA 1217-1.501, will cause BTOA to convert the first file collected from data table 1 of datalogger serial 1217.

SECTION 5. DIRECT PROGRAMMING

The term "direct" is used to differentiate this method of programming from "prompt" programming. Prompt programming is all that is required for the majority of Basic Data Recorder applications; the datalogger prompts for user input and generates the program. Direct programming is used to directly enter the Instructions which the datalogger executes. Direct programming is more flexible, allowing program branching, subroutines, and intermittent output.

Direct programming may be done in the datalogger following the 7H computer command. More often, a program is developed on a PC using EDLOG3 (Section 6) and downloaded to the BDR with TERM3.

5.1 *1, *A, AND *C MODES

The BDR has three programmable modes. ***1 Mode** is the Program Table where all data acquisition, processing, and storing instructions reside. The instructions are repeated at a user defined Program Table Interval. Subroutines are also programmed in the *1 Mode.

***A Mode** allows reallocation of Input Storage which is where the results of Input/Output, Processing, and redirected Output Processing Instructions are stored.

***C Mode** is used to block unauthorized access to the user's program and certain BDR functions.

Initial programming and subsequent changes to the *1, *A, and *C Modes cause the datalogger to recompile the program. Retrieve all data prior to making program changes; **RECOMPILING ERASES ALL STORED DATA.**

5.1.1 *1 MODE PROGRAM TABLE

The Program Table is accessed by entering the 7H computer command. When entered, the computer display shows the Program Table Interval. If there is an existing program in the table, entering ***1** followed by an instruction location number prior to A will advance the display directly to the instruction (e.g., ***15A** advances the display to the fifth instruction in the table).

The Table Interval is entered in whole number units of minutes or seconds as follows:

XXXX Minutes
XXXX-- Seconds

The dashes after the seconds is entered by keying **C**.

Execution of the table is repeated at the rate determined by this entry. The table will not execute if 0 is entered. If a previously entered execution interval is changed to 0, the table will not recompile and stored data will not be deleted (See COMPILING A PROGRAM).

If the specified interval is less than the time required to execute the program, the BDR finishes processing the table and waits for the next occurrence of the interval before initiating the table (i.e., when the execution interval is up and the table is still executing, that interval is skipped). Since no advantage is gained in the rate of execution with this situation, it should be avoided by specifying a Program Table Interval adequate for the table processing time.

NOTE: Whenever the processing time of the user's programs exceeds the Program Table Interval, Error 11 is logged in the Error Log (Section 4).

SUBROUTINES

Subroutines are entered in the programming table (*1 Mode) and are called with program control instructions in the *1 Mode. The group of instructions which form a subroutine starts with Instruction 85, Label Subroutine, and ends with Instruction 95, End.

COMPILING A PROGRAM

When a program is first entered, or if any changes are made in the *1, *A, or *C Modes, the program must be compiled before it starts running. The compile function checks for programming errors and initializes program information for use during program execution.

If errors are detected, the appropriate error codes are indicated on the computer display. The compile function is executed by entering ***0** or **ESC**ape.

On compiling, the output port and all flags are set low and data values contained in Input Storage are set to zero. **ALL STORED DATA ARE ERASED WHEN THE BDR RECOMPILES.**

5.1.2 *A MODE REALLOCATE INPUT STORAGE

The *A Mode is accessed using the 7H computer command, followed by ***A**. This Mode is used to 1) determine the number of locations allocated to Input Storage, and 2) change the number of locations.

If the number of locations allocated to Input Storage is changed, the **BDR RECOMPILES THE PROGRAM, ERASING ALL STORED DATA.**

TABLE 5.1-1. Description of *A Mode Data

Keyboard Entry	Display ID: Data	Description of Data
*A	01: XXXX	Input Storage Locations. This value can be changed by entering the desired number (minimum of 28)

5.1.3 *C MODE SECURITY

The *C Mode is accessed with the 7H computer command, followed by ***C**. This mode is used to block unauthorized access to the user's program information and certain BDR functions. There are three levels of security, each with its own four digit password. All passwords are set to 0000 on power-up which disables security. 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 stored in write protected memory and affect the program signature.

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 L computer command. 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.

Changing a password will cause the program to be recompiled, erasing all stored data.

TABLE 5.1-2. *C Mode Entries

SECURITY DISABLED

Keyboard Entry	Display ID: Data	Description
*C	01:XXXX	Level 1: Non-zero password blocks *1, *A, and 7W commands.
A	02:XXXX	Level 2: Non-zero password blocks C, I, 1W commands.
A	03:XXXX	Level 3: Non-zero password blocks all remaining commands except A, L, N, H, U, 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 unlocked. Password 1 entered unlocks levels 1,2, and 3 resulting in security level 0. (everything unlocked) Password 2 entered unlocks levels 2 and 3 resulting in security level 1. Password 3 entered unlocks level 3 resulting in security level 2.

The L computer command temporarily changes the security level. After terminating communications, security is reset.

5.2 PROGRAM INSTRUCTION TYPES

The instructions used to program the BDR are divided into four types: **Input/Output** (I/O), **Processing, Output Processing**, and **Program Control**. Instructions are identified by a number. Each instruction has a fixed number of parameters associated with it which give the datalogger the information it needs to execute the instruction.

- I/O Instructions (numbers 1-29) are used to make measurements and store the readings in input locations.
- Processing Instructions (numbers 30-61) perform numerical operations using data from Input Storage locations and place the results back into specified Input Storage locations.
- Output Processing Instructions (numbers 69-82) provide a method for generating time or event dependent data summaries from processed sensor readings residing in specified Input Storage locations.
- Program Control Instructions (numbers 83-100) are used to direct program execution based on time and/or conditional tests on input data.

5.3 PARAMETER DATA TYPES

There are three different data types used for Instruction parameters: Floating Point (FP), four digit integers (4), and two digit integers (2). In the listings of the instruction parameters in Section 8, the parameter data type is identified by its abbreviation.

Floating Point parameters are used to enter numeric constants for calibrations or arithmetic operations. While it is only possible to enter and display 5 digits (magnitude ñ.00001 to ñ99999.), the internal format has a much greater range ($\pm 1 \times 10^{-19}$ to $\pm 9 \times 10^{18}$). Instruction 30 can be used to enter a number in scientific notation to be loaded into an input location.

5.4 REPETITIONS

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 have two differential voltage measurements to make on the same voltage range, wire the inputs to sequential channels and instead of entering the Differential Voltage Measurement Instruction two times, enter it once with two repetitions. The instruction will make two measurements starting on the specified channel number and continuing through the next differential channel, with the results being stored in the specified input location and the succeeding input location. Averages for the two measurements can be calculated by entering the Average Instruction with two repetitions.

5.5 ENTERING NEGATIVE NUMBERS

The C key is used to enter a negative. It must be keyed after a number has been keyed in but before the number is entered. On floating point numbers a minus sign (-) will appear to the *left* of the number.

"--" is used to the *right* of the Program Table Interval to indicate seconds and indexed Input Locations in a loop (Section 5.6).

5.6 INDEXING INPUT LOCATIONS

When used within a Loop, Input Locations can be indexed to the loop counter. An input location is indexed by keying **C** after the location number is keyed but before the number is entered. The loop counter is added to the indexed value to determine the actual Input Location the instruction acts on. Normally, the loop counter is incremented by one after each pass through the loop. Instruction 90, Step Loop Index, allows the increment step to be changed. See Instructions 87 and 90, Section 8, for more details.

5.7 INPUT 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 5.7-1).

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.

Range code 00 can be used to make the voltage measurement in the proper range without selecting a specific voltage range.

Auto range should be used when a measured voltage varies widely over the 5 volt range, or when several varying voltage measurements are being made with one measurement instruction by using the REPS parameter.

Auto range is convenient to use since it can make any voltage measurement within the 5 volt limit. Two disadvantages exist:

- Some resolution is lost on voltage measurements less than 120 mV by using the auto range rather than the 80 mV or 120 mV range.
- The integration time of the auto range can be longer than that of the specific input range if the voltage is greater than 120 mV. Longer integration times increase current drain from the power supply.

TABLE 5.7-1. Input Voltage Ranges and Codes

Range Code	Integration time, ms	Range, mV
00 (auto range)	up to 16.7	-20 to +5000
01	50	-20 to +80
02	33.3	-20 to +120
03	16.7	-20 to +250
04	5.3	-30 to +1000
05	1.08	-50 to +5000

When a voltage input exceeds the specified range, the value stored is set to the maximum negative number and displayed as -99999 and stored as -6999.

An input voltage greater than ñ5 volts on one of the analog inputs will result in errors and possible overranging on other analog inputs. Analog input voltages greater than 16 volts may permanently damage the datalogger.

5.8 DATA STORAGE AND OUTPUT PROCESSING

Data are stored in Data Tables. The Data Table Interval Instruction (Instruction 84) defines how often and what data are stored. Data output at the specified interval constitute a record.

Output Processing Instructions (Instructions 70 through 82) following the Data Table Interval Instruction define what data are included in the Table. The first Output Processing Instruction outputs the first data point of the record.

The number of records to store in a Data Table is also specified in Instruction 84. When the program is compiled, memory is allocated for each Data Table based on record size and the number of records to store.

Up to nine Data Tables may be used.

Output Processing Instructions associated with an Instruction 84 store processed data values when and only when the interval specified for that Instruction 84 coincides with the real time clock.

Most Output Processing Instructions require both an intermediate data processing operation and a final data processing operation. For example, when the Totalize Instruction, 72, 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. At output time, the final processing operations store the resulting total and zero intermediate calculations.

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

5.9 USE OF FLAGS: PROGRAM CONTROL

There are 9 flags which may be used in datalogger programs (Table 5.9-1). Flag 9 is specific to disabling intermediate processing and Flags 1-8 may be used as desired in the program. Flag 9 is automatically set low at the beginning of measurement interval. Flags 1-8 remain unchanged until acted on by a Program Control Instruction or until toggled by an operator through the computer.

TABLE 5.9-1. Flag Description

Flag 1 to 8	-	User Flags
Flag 9	-	Intermediate Processing Disable Flag

Flags are set with Program Control Instructions.

5.9.1 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 5.9-2 lists these codes.

TABLE 5.9-2. Command Codes

0	-	Go to end of Program Table
1-9, 79-99	-	Call Subroutine 1-9, 79-99
11-19	-	Set Flag 1-9 high
21-29	-	Set Flag 1-9 low
30	-	Then Do
31	-	Exit loop if true
32	-	Exit loop if false
41	-	Set Port high
51	-	Set Port low
61	-	Toggle Port
71	-	Pulse Port

5.9.2 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) marks the beginning of the instructions which are to be executed regardless of the outcome of the comparison (Figure 5.9-1).

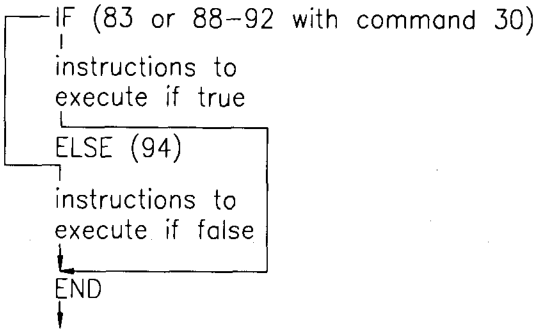


FIGURE 5.9-1. If Then/Else Execution Sequence

If Then/Else comparisons may be nested to form logical AND or OR branching. Figure 5.9-2 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.

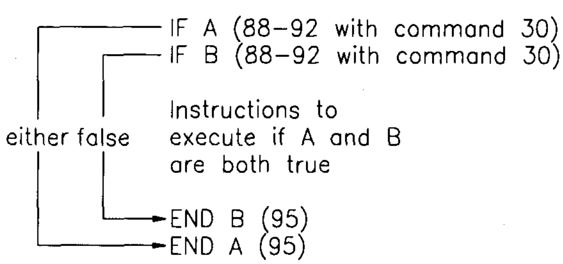


FIGURE 5.9-2. Logical AND Construction

Figure 5.9-3 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 5.9-3. Logical OR Construction

A logical OR can also be constructed by setting a flag if a comparison is true. (The flag must be cleared before making the 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 Instructions 83 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 8).

5.9.3 NESTING

A branching or loop instruction, which occurs before a previous branch or loop has been closed, is nested. The maximum nesting level for loops and branching instructions is 9 deep. Begin Case Instruction 93 counts as 1 level. Instructions 83, 86, 88, 89, 91, and 92 each count as one level when used with 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.

Parameter 1 of the Loop Instruction is pass duration, or the time required to make one pass through the loop. Pass duration must be less than one second in order to nest loops. If the pass duration is greater than or equal to one second, loops can not be nested.

Subroutine nesting, which has no limit, is when a subroutine is called from another subroutine. A subroutine cannot be embedded within other subroutines. A subroutine must end before another subroutine begins (error 20, Section 5.11). Any loops or IF/THEN DO sequences started within a subroutine must end before the subroutine.

5.9.4 FLAG 9 INTERMEDIATE 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 the program table.

Flag 9 is unique in that if it 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 the need to enter another instruction to specifically set Flag 9 low before proceeding to other instructions.

TABLE 5.11-1. Error Codes

Code	Type	Description
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	No Output Processing Instructions following P84
29	Compile	Flag 0 does not exist
30	Compile	IF and/or LOOP nested too deep
31	Compile	Storage area exceeded
95	Editor	Instruction does not exist
96	Editor	Attempt to allocate more Input than is available
97	Editor	Time-out on program download
98	Editor	Out of program memory
99	Editor	Wrong program file type

5.10 END, INSTRUCTION 95

The END instruction is required to mark the end of:

- A Subroutine (starts with Instruction 85).
- A Loop (starts with Instruction 87).
- An IF . . . THEN DO sequence (starts with one of Instructions 89-93 with the THEN DO command 30).

The IF instructions 89-93 require Instruction 95 only when the THEN DO command 30 is used.

If one of the above instructions is used without the corresponding END, error 22 is displayed when compiling the program. Error 21 is displayed if END is used without being preceded by one of these instructions (Section 5.11).

An END instruction is always paired with the most recent instruction that requires an END and does not already have one. A way of visualizing this is to draw lines between each instruction requiring an END and the END paired with it (as in Figure 5.9-2). The lines must not cross. To debug logic or find a missing or extra END error, list the program and draw the lines.

5.11 ERROR CODES

There are 3 types of errors flagged by the datalogger: compile, run time, and editor. The error log, 1U command (Section 4), is composed of run time errors only (errors 1 through 13).

The codes and descriptions of compile and editor errors is given in Table 5.11. These errors appear on the computer screen when the errors are detected.

5.12 PROGRAM SYNTAX AND DOWNLOAD RULES

This Section describes the syntax requirements for programs developed in a computer for the purpose of downloading to the datalogger. This section does not need to be read if the programs for download are developed in EDLOG3 (Section 6).

A program residing in a datalogger may be transmitted to the computer using the 2718,1Q command. A program residing on a computer may be loaded to a datalogger using the 2718Q command.

Table 5.12 is an example program which requests stage from an SDI-12 sensor (Instruction 29), measures datalogger battery (Instruction 10) and temperature (Instruction 17). The first Instruction 84 stores a sample stage reading every hour. The second Instruction 84 saves daily average stage, maximum stage, time of max stage, minimum stage and time of minimum stage. The primary purpose of this example is to show the syntax required to generate a program on a text editor, not programming per se.

TABLE 5.12-1. Example Program Listing

```
};300
;SAMPLE.DLD
;$
;:STAGE  FT:BATT   V :TEMP   C
;$

MODE 1
Program Table Interval 1

1:P29
1:101
2:1
3:1
4:0

2:P10
1:2

3:P17
1:3

4:P84
1:60
2:4000

5:P70
1:1
2:1

6:P84
1:1440
2:31

7:P71
1:1
2:1

8:P73
1:1
2:10
3:1

9:P74
1:1
2:10
3:1

MODE 10
1:28

MODE 12
1:0000
2:0000
3:0000
```

RULES FOR DOWNLOAD FILES

1. } means start of program and must be the first character of a program.

2. ; is the beginning of a comment line

3. Labels/Units for Input Locations begin and end with ";$". Each Input Location label/unit line begins with ";:". Each line contains five groups of nine characters each, separated by colons (:). The first six characters are used for the label; the last three for the units. The first label/unit entry corresponds to Input Location 1, the second to Input Location 2, etc.

4. "MODE" or "M" must be the first character of the line other than a carriage return or line feed when changing modes. A comment line starts with a semicolon, or "}" (7D Hex). The order that the Modes are sent does not matter (i.e., the information for Mode 12 (or C; security) could be sent before that for Mode 1).

5. "Program Table Interval" is required before entering the interval in minutes (XXXX) or seconds (XX-).

6. Colons (:) are used to mark the start of actual parameter data.

7. There are 4 two-character control codes which may be used to verify that the datalogger 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 datalogger buffers the data in memory; the data is not loaded into the editor or compiled until the BDR receives a command to do so. The maximum size of the buffer is 1.5K. The minimum file that could be sent, provided it is shorter than 1.5 Kbytes, is the program listing, then ^E ^E.

Larger programs can be loaded in blocks. ^C ^C tells the BDR to send the signature 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 BDR 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.

SECTION 6. DIRECT PROGRAMMING EXAMPLES

DIRECT PROGRAMMING

In some cases, you may need the flexibility of direct programming for branching, conditional output, etc. Direct programming utilizes the Instruction Set of the BDR320. The Instruction Set is familiar to you if you have used other Campbell Scientific dataloggers, although there are some differences. In the following example, EDLOG3 software is used to develop a program that will do about the same things we did in the sample problem in the Overview. Once the program is developed, it is downloaded to the BDR with TERM3.

Example #1

The level sensor is to be scanned every five seconds and an average recorded every minute.

Solution

Execute the EDLOG3 software by typing "EDLOG3" on the computer. You are prompted for a name; call it "test2". A starter file is loaded; fill in the file as we go through it below. You are prompted to enter the interval; enter **5--** . The -- denotes that the interval is in seconds:

```
*          1       Program Table
 01:       5--     Min. (Sec.--) interval
```

You are now at a line asking for the next program instruction:

```
01:        P       End Program Table
```

The level sensor is a potentiometer; Instruction 5 is used for half bridge measurements. Enter **5** after the P and the instruction and parameters will appear on the screen. The values for the parameters must be filled in. Set the Reps to 1 because we only have 1 level sensor. The Range Option of 0 is the auto range. The level sensor is wired to IN Chan 1, and the level is stored at Location 1. The Multiple of 5 scales the output to units of inches:

```
01:        P5      Half Bridge (Pot)
 01:       1       Reps
 02:       0       Autorange
 03:       1       IN Chan
 04:       1       Loc [:level in ]
 05:       5       Mult
 06:       0.      Offset
```

Note that as the parameters are filled in, EDLOG3 will change the parameter notation to document the selected option. If you need help on the options available, try pressing F1. If help is available for the option the cursor is on, it will be displayed on the right of the screen.

Instruction 84 sets the output interval for final storage. Enter 84 and fill in the parameters:

```
02:      P84       Output Record
 01:       1       Minutes interval
 02:   99999       No. of records
```

We want data every minute. The 99999 allocates the remainder of the data storage for this table.

The desired data is the average level. Instruction 71 is used to output the average level from Location 1:

```
03:      P71       Average
 01:       1       Reps
 02:       1       Loc level in

04:        P       End Program Table
```

That is the end of the program. Press **ESC**ape and then **S** to save the file, and **Q** to quit. We need to load the program to the BDR320 from TERM3. In TERM3 select D - Download program to datalogger. The name is "Test2".

That's it! It wasn't too bad was it? Go ahead with testing out the program and retrieve data like we did earlier.

Example #2

You are required to record the average daily level at a stream gauging site for a monthly report. You also need a continuous record of level during runoff events. However, you do not want to have to process hundreds of pages of data each month. You determine that your need would be satisfied if you have a record of level every time it changes 0.25 inches or more since the last recorded level. Under this scenario, you would have 30 readings of average daily level. If the total change in level both plus and minus were two feet, you would have 96 (2 ft/0.25 inches) records of continuous level change during the month.

Solution:

The following programs perform the task. Refer to the manual for an explanation of the instructions used.

*	1	Program Table	
01:	1--	Min. (Sec.--) interval	
01:	P5	Half Bridge (Pot)	*measure pot*
01:	1	Reps	
02:	00	Range Option	
03:	3	IN Chan	
04:	1	Loc [:level]	
05:	5	Mult	
06:	0.0000	Offset	
02:	P84	Output Record	
01:	1440	Minutes interval	*output once per day*
02:	99999	No. of records	
03:	P71	Average	*the average*
01:	1	Reps	
02:	1	Loc level	*level*
04:	P35	Z=X-Y	
01:	1	X Loc level	
02:	2	Y Loc Lst recrd	
03:	4	Z Loc [:change]	*Compute the change in level since the last value was recorded*
05:	P43	Z=ABS(X)	*Take the absolute value of the change*
01:	4	X Loc change	
02:	4	Z Loc [:change]	
06:	P89	If X<=>F	*If the change is greater than 0.25*
01:	4	X Loc change	
02:	3	>=	
03:	.25	F	
04:	30	Then Do	*then*
07:	P84	Output Record	*Output (with time)*
01:	0--	Each time	
02:	99999	No. of records	
08:	P70	Sample	
01:	1	Reps	
02:	1	Loc level	*the new level*
09:	P31	Z=X	*and set the last recorded value to the current level*
01:	1	X Loc level	
02:	2	Z Loc [:Lst recrd]	
10:	P95	End	*end*
11:	P	End Program Table	

SECTION 7. EDLOG3

7.1 OVERVIEW

EDLOG3 is used to develop and document programs for the BDR320. EDLOG3 is written for use on IBM personal computers (or compatible) running DOS with at least 256k of RAM memory and an 80 column by 25 line monitor.

Instructions and parameters are entered with the same characters that are used to program the datalogger directly. EDLOG3 automatically describes the instructions and prompts for the parameters. The user may add additional comments as desired. Cursor movement commands are listed in Table 7.2-1.

7.1.1 GETTING STARTED

Enter **EDLOG3** on the command line to load and execute the program. The PC must be logged onto the disk and directory containing EDLOG3 or the appropriate "path" must be specified per the PC-DOS Manual. EDLOG3 prompts for a ".DOC filename:". The filename is not entered with an extension, but may include drive and path specifications (e.g., [D:][\path\]filename). EDLOG3 adds the extension .DOC. EDLOG3 loads the file or, if the file does not already exist, loads a Starter program. The Starter program becomes the root of the new file to be edited. If * is entered in place of the filename, EDLOG3 will show the .DOC filenames for the specified drive and directory path.

7.1.2 INSTRUCTION AND COMMENT FIELDS

The document file is divided into instruction and comment fields. At the beginning of the file are comment lines where the user may define outputs and flag and channel usage. Additional comment lines may be added (Section 7.3.2) under any of the categories shown, or any of the comment lines except the first may be deleted.

Below these comments is the body of the program. This is divided into the Program Table, *A Memory allocation, and *C Security options. On the left of each of these lines is datalogger ID information which cannot be edited. This field includes Mode, Entry number and Parameter number information. The next field to the right is the Instruction field where the actual programming is done. These first two fields correspond to the ID and Data fields of the datalogger display.

On the right is the Comment field. Each Instruction, Mode, and Parameter line includes some protected comments about that line which cannot be edited. These comments may become more specific once entries have been made in the Instruction field. The user may add to the comment part of the line. The length of the user entered comment is limited. When the limit is reached, EDLOG3 will not accept additional text and another comment line must be added (Ctrl N).

7.2 EDITING ENTRIES AND COMMANDS

EDLOG3 accepts the same programming entries and many of the same commands as the datalogger. Entry commands only work when the cursor is in an Instruction field. If the cursor is in the comments area of EDLOG3, the key will be printed as a comment, if printable. Table 7.2-1 summarizes the editing commands.

7.2.1 HELP-- F1

F1 is the Help key. When uncertain what to enter for a specific parameter, press the Help key. Help is only available in the parameter field. If help exists, it will be displayed, otherwise nothing will happen.

On parameters containing the comment word "Loc", the Help key will display the location labels assigned. At Instruction entry fields, the Help key will display the list of Instructions.

7.2.2 FILE COMMANDS-- F2

When the F2 Key is pressed, the file command menu is displayed. Select a command by pressing the first letter of the command or move the cursor to the command with the arrow keys and press return.

TABLE 7.2-1. Editing Command Summary

Help	F1
File Commands	F2
Edit Functions	F3
Go To *Mode Always	F4
Go To * Mode In Data	*
Cancel Command	Esc
Advance In Program Table	A
Advance One Line	Enter
Advance Entry	Ctrl V
Backup In Program Table	B
Change Sign Or Index Location	C Or -
Insert Decimal	D Or .
Insert Dependent Comment Line	Ctrl N (Cursor On Data Line)
Insert Independent Comment Line	Ctrl N (Cursor On Empty Line)
Delete Comment Line	Ctrl Y
Enter Location Label	Ctrl L ("Loc:" In Comment)
Mark Instruction	Ctrl Kb
Delete Instruction	Ctrl Ky
Copy/recall Instruction	Ctrl Kc (Cursor On Program #)
Write To Library File	Ctrl Kw
Read Library File	Ctrl Kr
Edit "User Special" Instruction	Ctrl O
Ram And Disk Status	Ctrl \
------------------ Cursor:------------------	
Left	Ctrl S Or Left Arrow
Right	Ctrl D Or Right Arrow
Up	Ctrl E Or Up Arrow
Down	Ctrl X Or Down Arrow
Scroll Up	Ctrl W
Scroll Down	Ctrl Z
Page Up	Ctrl R Or Pgup
Page Down	Ctrl C Or Pgdn
Word Left	Ctrl A
Word Right	Ctrl F
Go To Parameter Field	Shift Tab
Go To End Of Comment	Tab (Cursor In Instruction Field)
Tab 5 Spaces	Tab (Cursor In Comment Field)
Go To Beginning Of Document	Home
Go To End Of Document	End
Delete At Cursor	Ctrl G Or Del
Delete To Left	Ctrl H Or Backspace
Delete Word Right	Ctrl T

7.2.3 EDIT FUNCTIONS-- F3

When **F3** is pressed, EDLOG3 displays the main editing features for EDLOG3. The feature listing disappears when the cursor is advanced to the next line. Table 7.2-1 gives a summary of editing features and commands.

7.2.4 GO TO-- F4

F4 is used to go directly to some point in the program being edited. When **F4** is pressed, EDLOG3 will display the list of * Modes and wait for the user to select one. The * Modes are *1 Program Table, *A memory allocation, and *C security. The cursor will then move to the first parameter entry location of the selected mode. When entering a Program Table, the user may also specify the entry to go to in that table. This is specified by the table number then the entry number (e.g., 125 moves the cursor to the 25th Instruction in Table 1).

7.2.5 INSERT INSTRUCTION

The user can insert a Program Instruction when the cursor is in the Instruction field of any line starting with "xx: P". For a list of the available Instructions, press the Help key (F1). Type the Instruction number then enter, **A**, or **Tab**. The Instruction and all its parameters are inserted into the file. The cursor advances to the first parameter in the new Instruction unless Tab was pressed, in which case the cursor moves to the Comment field on the first line, allowing comments to be entered.

7.2.6 PARAMETER ENTRY-- 1 TO 9, DECIMAL, CHANGE SIGN

As with the datalogger, the initial state of the parameters shows the number of digits allowed by the number of zeros shown. Decimals are entered with either the . or the **D** key. EDLOG3 will not allow a decimal to be placed in an integer number. The **C** or - key will change the sign of a floating point number or add the indexing sign. Notice that when a parameter field is edited, it is first cleared. Changing sign or indexing cannot be accomplished as the first keystroke. A number must first be entered. The parameter is entered and evaluated when the cursor is moved from the parameter field being edited. This is accomplished with the "A" (Advance) or "B" (Backup) commands, or the cursor movement commands. If a more specific comment exists for the Parameter entered, EDLOG3 places it in the comments area.

If the parameter entered is not within the valid set of values for that parameter, the parameter flashes in reverse video, the proper range is shown and the user is prompted to select a valid entry. If **Enter** is pressed instead of entering a valid number, the flashing will stop, but the parameter will remain highlighted until it is corrected.

Enter values for all parameters in an instruction even if the parameter is zero; this insures that the comments provided by EDLOG3 reflect the entry. The description of a parameter may be updated to a more specific description when the parameter is entered.

7.2.7 ADVANCE IN PROGRAM TABLE-- A

When the cursor is in the parameter area, the **A** key can be used to enter data and advance the cursor to the next parameter. The **A** command in EDLOG3 will pass through Mode boundaries. If the cursor is in a Comment field, the **A** is just the letter.

7.2.8 ADVANCE ONE INSTRUCTION-- CTRL V

Regardless of cursor position in the Program Tables, Ctrl V will advance the cursor to the next Program entry number. Ctrl V will advance the cursor through Table boundaries, but takes no action past the end of Table 3.

7.2.9 BACKUP IN PROGRAM TABLE-- B

When the cursor is in the parameter area, the **B** key can be used to back up the cursor one entry. The **B** command in EDLOG3 will pass through Mode boundaries. If the cursor is in a Comment field, **B** is just the letter.

7.2.10 NEXT LINE-- ENTER

The Enter key will advance the cursor to the next non-empty line. If the next line is a Comment line, the cursor will be positioned to the first character in that line. If the next line is a parameter entry line, the cursor will be positioned in the parameter field of that line. Unlike most editors, Enter will not insert lines.

7.2.11 GO TO END OF COMMENT/TAB-- TAB

Tab allows the user to quickly move to the end of the Comment field from the instruction field to append text. Once in the comment field, Tab will insert five spaces.

7.2.12 GO TO PARAMETER FIELD-- SHIFT TAB

Shift Tab moves the cursor to the first position in the parameter field of that line. If the line is a Comment line, the cursor is moved to the beginning of that line. This command is most useful when moving from the Comment area of a line to the parameter entry field.

7.2.13 DELETE AT CURSOR-- DEL OR CTRL G

The character at the cursor position is deleted by Ctrl G or the Del key.

7.2.14 DELETE LEFT CURSOR-- BACKSPACE OR CTRL H

The character to the left of the cursor is deleted by Ctrl H or the Backspace key.

7.2.15 DELETE WORD RIGHT-- CTRL T

The word to the right of the cursor is deleted by Ctrl T.

7.3 COMMENT AND INSTRUCTION MOVE COMMANDS

7.3.1 ENTER LOCATION LABEL-- CTRL L

Ctrl L is used to enter labels for the first 254 Input Storage locations. Once an Input Location is labeled, EDLOG3 will insert the label into the comment field of every line which refers to that Input Location.

Ctrl L is active only on parameters that assign locations to store measurements or processed data. The comment for those lines will always have the word "Loc:". Comments which have the word "Loc" but no ":" represent locations from which data is extracted for processing or output. Ctrl L will not be active on these lines, but a Label already created elsewhere will be included in the comment once the Location number is entered.

When **Ctrl L** is pressed on a valid line, EDLOG3 will prompt for the Label within a 9 character window. If the Label was already assigned, the current Label assignment will also be displayed. If no Label is entered, Return will leave the Label unchanged.

Enter **#** as the last character in a Label if you wish to create several Labels which differ only by a suffix number. EDLOG3 will ask for the number of reps (repetitions) of the Label, and then create Labels for consecutive Locations, searching the text with each repetition to insert the Label where any reference to that location is made. For instance, entering **Temp#** with 3 reps at Input Storage location 10 will assign the labels Temp#1, Temp#2, and Temp#3 to locations 10, 11, and 12.

Enter + as the last character in a label if you want to enter a different Label for the next input location. The user may continue to edit Labels by entering + as the last character of the Label being edited. The + will not be included as part of the Label unless it is not the last character. Enter only the + to move to the next location without changing a label already assigned.

Entering **erase** for a Label will erase the Label at that location. **Erase#** can be used to erase a number of sequential labels.

7.3.2 INSERT COMMENT LINE-- CTRL N

Ctrl N inserts a comment line below the line which the cursor is on. Comment lines are the same length as the Comment field of parameter entry lines and are left justified.

The Dependent Comment Line

When the cursor is on a non-comment line and **Ctrl N** is pressed, a Dependent Comment line is created. Dependent means the line has the same Entry number, Parameter number, and Program number as the line above it. If the dependent line is part of an Instruction, the line will be moved with the Instruction if the Instruction is moved to another location. If **Ctrl N** is pressed while the cursor is on a Dependent Comment line, another Dependent Comment line will be created.

The Independent Comment Line

When the cursor is on an empty screen line (no characters at all) and **Ctrl N** is pressed, an Independent Comment line is created. This line is not linked to any line near it. It will not be moved with an Instruction that may be next to it. The Independent Comment line is useful for section comments. If **Ctrl N** is pressed while the cursor is on an Independent Comment line, another Independent Comment line will be created.

7.3.3 DELETE COMMENT LINE-- CTRL Y

Ctrl Y will delete any Comment line except the top line of the file. The top line of the file is an Independent Comment line which cannot be deleted although every character in the line can be deleted. Ctrl Y will not delete Parameter entry lines.

7.3.4 MARK INSTRUCTION-- CTRL KB

Ctrl KB will cause all lines of an Instruction to be displayed in low video including any Dependent Comment lines. The instruction is placed in a buffer which allows it to be duplicated at another position in the program. Ctrl KB works on a single instruction. The cursor may be in any position within the Instruction at the time Ctrl KB is pressed.

The Instruction will remain marked as long as no further use is made of the Instruction buffer. Otherwise, the displayed Instruction will be returned to normal video. The Instruction can be unmarked by pressing **Ctrl KB** again.

7.3.5 DELETE INSTRUCTION-- CTRL KY

Ctrl KY will delete an Instruction and place it in the Instruction buffer. The entry numbers of the remaining Instructions are appropriately adjusted. The Instruction may be moved to another position (Ctrl KC). If another Instruction is inserted in the Instruction buffer, the deleted Instruction will be lost. INSTRUCTION SELECTION, MARK INSTRUCTION, DELETE INSTRUCTION, and READ/WRITE LIBRARY FILE, are the functions which place lines in the Instruction buffer. Thus when moving an Instruction, it is good practice to immediately insert the Instruction in its new location. The cursor may be in any position within the Instruction at the time **Ctrl KY** is pressed.

7.3.6 COPY/RECALL INSTRUCTION-- CTRL KC

Ctrl KC will insert the Instruction from the Instruction buffer into the program at the current cursor position. The cursor must be on an Instruction entry position. If the buffer is empty, no change will occur. The operation may be repeated. The inserted Instruction will be displayed in low video. If the Instruction is edited or the Instruction buffer used, the Instruction will return to normal video.

7.3.7 WRITE TO LIBRARY FILE-- CTRL KW

Ctrl KW is used to save a group of Instructions in a Library file (filename.LIB). The Instructions in a library file can be read into the program being edited with Ctrl KR.

The cursor must be on an Instruction number (xx:Pxx) when **Ctrl KW** is pressed. EDLOG3 will ask for the number of Instructions (starting at the cursor position) to send to the library file. Entering a number greater than the number of Instructions between the cursor and the end of the Program Table will save the rest of the table; the write command will not cross Table boundaries.

When EDLOG3 asks for a name for the library file, do not enter an extension. The extension, .LIB, will automatically be added.

7.3.8 READ LIBRARY FILE-- CTRL KR

Ctrl KR will read a library file into the program being edited. The cursor must be on an Instruction number (xx:Pxx) when **Ctrl KR** is pressed. EDLOG3 will ask for the name of the library file. Once the name is entered, the Instructions from the library file will be inserted at the cursor.

7.3.9 RAM AND DISK STATUS-- CTRL \

Ctrl \ will cause the RAM and disk status to be displayed for 10 seconds. The RAM status tells how many lines have been used of the largest possible program which can be created. The disk status tells how many program lines can be stored on the default disk drive. See SAVE (Section 7.4.3).

7.3.10 EDIT SPECIAL INSTRUCTION-- CTRL O

Skip this section unless you have a datalogger with "special" software. Special software means that your datalogger has an Instruction that is not supported by standard EDLOG3; when the Instruction number is entered EDLOG3 will display "user special".

> **NOTE:** DO NOT try this feature without reading this entire section.

The Ctrl O command allows you to create a new instruction in the editor. This allows EDLOG3 to be changed to accommodate special software in a datalogger.

The file PRGINSTR.300 is a special document file which EDLOG3 uses as the source for Instructions and Parameters. The PRGINSTR file must be edited to install the special Instruction, but it is a good idea to practice editing a special Instruction in a test file first.

Enter the special Instruction number (e.g., 99) on an Instruction entry line. EDLOG3 will display the comment "User Special". Move the cursor back to the Instruction line. Press **Ctrl O**. EDLOG3 will prompt:

Options ---------------------
A Add parameter line of type 00
B Add parameter line of type 0000
C Add parameter line of type 0.0000
D Delete parameter line
E Edit protected comment
X Return to main editor
Enter Option:

Press **E** to enter the Instruction name. The parameter lines are added by pressing the appropriate letter. After adding a parameter, press **E** to enter the comment for that parameter.

If the parameter is a source location, end the comment with "Loc". If the parameter is the destination location, end the comment with "Loc :". EDLOG3 will check the comments and allow a Label to be entered for the destination and will show the Label for the source. The location parameters are the only parameters in the special instruction on which EDLOG3 will range check and update comments.

It is not possible to backup and change a comment; you must return to the main editor, position the cursor on the parameter and press **Ctrl O**. Parameter lines are inserted below the displayed line. **D** will delete the line and return to the main editor.

Before editing the PRGINSTR file, make sure that you have a backup disk. Copy the appropriate PRGINSTR.300 file to PRGINSTR.DOC. Edit PRGINSTR.DOC with EDLOG3. Note that the Instruction Location numbers in Table 1 are the same as the Instruction numbers; if they are not, the wrong Instruction will be loaded in the main editor.

Move the cursor to the number of the special Instruction you want to edit. The Instruction will be labeled "Spacer". Press **Ctrl O**, change name from Spacer to the name of the special Instruction and edit the parameters for the Instruction. When you are done, make sure that you have not inserted or deleted any Instructions (i.e., that the Instruction Location numbers and the Instruction numbers match throughout Table 1). Save the .DOC file and quit EDLOG3.

Delete the old PRGINSTR.300 file and rename PRGINSTR.DOC to PRGINSTR.300. Now EDLOG3 will supply the lines and comments for the special instruction.

If a special Instruction is deleted from PRGINSTR.300, the Instruction number and the comment "Spacer" must be inserted in its place.

7.4 FILE COMMANDS

When **F2** is pressed, the file command menu is displayed. Select a command by pressing the first letter of the command or move the cursor to the command with the arrow keys and press **Enter**.

7.4.1 QUIT

If the program has been changed since being loaded or saved, EDLOG3 will ask the user whether it should be saved before exiting EDLOG3. If the program has not been changed, EDLOG3 will exit immediately.

7.4.2 EDIT

EDIT puts EDLOG3 in the Program editing mode.

7.4.3 SAVE

If the .DOC extension of the filename already exists on disk, it will be renamed to a .BAK file-name before the new .DOC file is saved. In addition to saving the file being edited (.DOC), EDLOG3 creates a file with the same filename but with the extension .DLD.

If the default disk does not have enough room to store the files, EDLOG3 will prompt the user to place a floppy disk into drive A. EDLOG3 will recheck the room available on the disk in drive A, and if sufficient, will store the files and prompt the user to replace the disk that previously occupied drive A.

The .DLD, or download, file is used to program the datalogger. The program is loaded with TERM3. The file can also be written to storage module for transfer to the BDR.

Location Labels are stored as comment lines in the .DLD file. If the .DOC file is re-edited, EDLOG3 uses the Labels from the .DLD file. The Location Labels are also used by TERM3 for the monitor option. The Labels are found between the special comment lines designated by ";$". The simple ASCII format of the .DLD file allows datalogger users to make minor modifications to the program and Labels in the .DLD file with common text editors.

7.4.4 PRINT

The file being edited is sent to the default print device or to a Printable DOC File (.PDF). The file being edited is printed first. Next, is a summary of Table and Entry numbers of Instructions which store data in Input Storage Locations. The last page is a list of Label assignments for the first 99 Input Storage Locations. The file being edited does not have to be saved before printing. The .DOC file is not a text file; it cannot be printed or edited with other editors. A .PDF file can be edited with a text editor, however, the .PDF file cannot be substituted for a .DOC file in EDLOG3.

7.4.5 LOAD NEW .DOC FILE

EDLOG3 will ask if the current file should be saved before asking for the next file to be edited. If no filename is entered, EDLOG3 will exit to the operating system when the Enter key is pressed.

7.4.6 DOCUMENT .DLD FILE

Datalogger programs are often entered and edited from the keyboard. EDLOG3 will also allow program input from disk file. The benefit of the file driver facility is to document existing datalogger programs created without EDLOG3. This file may be created by uploading a program from the datalogger to the PC (SAVE option in TERM3). Just as with EDLOG3's downloadable files, these files are given the .DLD extension.

After pressing **D** for file input, EDLOG3 asks for the .DLD filename. The file must have the extension .DLD.

EDLOG3 will take the program from the file and document the program. If Location Labels are included in the file, they will be inserted into the text where referenced. At the end of the file, EDLOG3 will return to the keyboard entry mode.

SECTION 8. BDR INSTRUCTION SET

8.1 ANALOG MEASUREMENT INSTRUCTIONS

TABLE 8.1-1. Input Voltage Ranges and Codes

Range Code	Integration time, ms	Range, mV
00 (auto range)	up to 16.7	-20 to +5000
01*	50	-20 to +80
02*	33.3	-20 to +120
03*	16.7	-20 to +250
04	5.3	-30 to +1000
05	1.08	-50 to +5000

*Ranges 1, 2, and 3 integrate for 3, 2, and 1 periods of a 60 Hz signal to provide rejection of 60 Hz noise. A different PROM is available for countries where 50 Hz AC power is standard (Appendix D).

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 and stored as -6999 in Data Tables.

*** 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.

PAR. NO.	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Range code
03:	2	Channel number for first measurement
04:	4	Input location for first measurement
05:	FP	Multiplier
06:	FP	Offset

Input locations altered: 1 per measurement

*** 2 DIFFERENTIAL VOLTS ***

FUNCTION
This Instruction reads the voltage difference between the high and low inputs of a differential channel. Table 8.1-1 contains all valid voltage ranges and their codes. Both the high and low inputs must be within ±5 volts of Analog Ground (Pin J, Analog Connector) to keep the signal in Common Mode Range. Output is in millivolts.

PAR. NO.	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Range code
03:	2	Channel number for first measurement
04:	4	Input location for first measurement
05:	FP	Multiplier
06:	FP	Offset

Input locations Altered: 1 per measurement

*** 3 PULSE COUNT ***

Instruction 3 counts pulses on pulse count channels 1 or 2. Channel 1 is the slow pulse input, designed for a tipping bucket rain gauge. Channel 2 is for faster pulse inputs, such as a contact closure anemometer. Counts are returned for channel 1, frequency for channel 2. Channel 1 increases the current drain of the BDR in proportion to the frequency of the signal being measured (20ms at 13mA/count).

Pulse input signal types are switch closure and voltage pulse.

Channel 1:

Maximum input frequency is 20 Hz.
Minimum switch closure time is 100 microseconds
Minimum voltage pulse low time is 100 microseconds
Maximum debounce filter time is 3 ms
Maximum voltage magnitude is 5V.
Result in counts

Channel 2:

Maximum input frequency is 150 Hz.
Minimum switch closure time is 200 microseconds
Minimum voltage pulse low time is 200 microseconds
Maximum debounce filter time is 2.5 ms
Maximum voltage magnitude is 5V.
Result in frequency

PAR. NO.	DATA TYPE	DESCRIPTION
01:	2	Channel
02:	4	Input Location
03:	FP	Multiplier
04:	FP	Offset

Input locations altered: 1

Intermediate Storage locations altered: 1

*** 5 HALF BRIDGE ***

FUNCTION
This Instruction is used to make a half bridge measurement using the excitation provided by the datalogger. Output is the ratio of the measured single-ended voltage to excitation voltage, Vs/Vx.

PAR. NO.	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Range code
03:	2	Channel number for first measurement
04:	4	Input location for first measurement
05:	FP	Multiplier
06:	FP	Offset

Input locations altered: 1 per measurement

*** 6 FULL BRIDGE ***

FUNCTION
This Instruction is used to make a full bridge measurement using the excitation provided by the datalogger. Output is the ratio of the measured differential voltage to excitation voltage, Vs/Vx.

PAR. NO.	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Range code
03:	2	Channel number for first measurement
04:	4	Input location for first measurement
05:	FP	Multiplier
06:	FP	Offset

Input locations altered: 1 per measurement

*** 9 6 WIRE FULL BRIDGE ***

FUNCTION
Instruction 9 is used to make a 6 wire full bridge measurement. *All* analog inputs are used. The excitation voltage at the bridge is measured on differential input 1 (CH1/CH2), and the bridge output on differential input 2 (CH3/CH4). With the excitation voltage measured on auto range (0) or the 5 volt range (5), the raw value output is 1000 times the output divided by the excitation voltage (millivolts out per volt of excitation); the typical calibration units for strain gauge pressure transducers. With the excitation measured on any other range, the raw output is output voltage divided by the excitation voltage.

PAR. NO.	DATA TYPE	DESCRIPTION
01:	2	Excitation range code
02:	2	Bridge output range code
03:	4	Location for first measurement
04:	FP	Multiplier
05:	FP	Offset

*** 10 BATTERY VOLTAGE ***

FUNCTION
This Instruction reads the battery voltage and writes it to an input location. The units for

battery voltage are volts. When the batteries are around 8V, false battery readings of 9 to 10V will result, and the quiescent current drain increases to 7mA. At 9.2 to 9.3V, false analog measurements are possible (Example: 2000mV input is measured as 2010 to 2050mV).

PAR. NO.	DATA TYPE	DESCRIPTION
01:	4	Input location

Input locations altered: 1

*** 11 BDR THERMISTOR PROBE ***

FUNCTION
This Instruction makes a half bridge measurement on the 103 or 103B Thermistor Probe and calculates the temperature in °C with a polynomial linearization. Refer to Appendix A for details on the 103 probe.

PAR. NO.	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	2	Input channel number of first measurement
03:	4	Input location for first measurement
04:	FP	Multiplier
05:	FP	Offset

Input locations altered: 1 for each thermistor channel

*** 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 pending International Electrotechnical Commission standard. The range of linearization is -200 °C to 850 °C. The error in the linearization is less than 0.001°C between -100° and +300°, and is less than 0.003°C between -180° and +830°. The error (T calculated T standard) is +0.006° at -200° and -0.006° at +850°. The input must be the ratio R_s/R_0, where R_s is the RTD resistance and R_0 the resistance of the RTD at 0°C.

PAR. NO.	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	4	Input location of Rs/Ro
03:	4	Input location of result
04:	FP	Multiplier
05:	FP	Offset

Input locations altered: 1 for each RTD

*** 17 INTERNAL TEMPERATURE ***

FUNCTION
This Instruction measures the temperature in °C of a thermistor on the datalogger analog board.

PAR. NO.	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 hour, hours into the day, etc. 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 modulo divide will result in the actual time value being stored.

Code	Time values	Maximum
0	Seconds into the minute	60
1	Minutes into the hour	60
2	Hours into the day	24
3	Days into the month	30 (31)
4	Months into the year	12
5	Years into the century	99

PAR. NO.	DATA TYPE	DESCRIPTION
01:	2	Time Code
02:	4	Number to modulo divide by
03:	4	Input location number

Input locations altered: 1

***** 29 SDI READ *****

FUNCTION
This Instruction addresses a SDI sensor and stores the specified value in an input location after applying the multiplier and offset. The SDI address, command and value to store are all specified in parameter 1. At present only command 0 is defined, telling the SDI sensor to commence measuring. The sensor responds with the number of seconds required for measurement and how many values will be sent. Table execution is suspended for the specified time period, after which the value specified is stored in the specified input location. If data is not received, -99999 is loaded into the specified input location(s). If several values from a SDI sensor are required, enter an Instruction 29 for each value. If the instructions are entered next to each other, the sensor will only be addressed once.

PAR. NO.	DATA TYPE	DESCRIPTION
01:	4	Address(0-9), Command (0-9), Value(1-9)
03:	4	Input location
04:	FP	Multiplier
05:	FP	Offset

Input locations altered: SDI sensor dependent

8.2 PROCESSING INSTRUCTIONS

Symbols used to describe the function of the processing instructions are defined as follows:

[Z] = User specified input location number destination
[X] = Input location no. of source X
[Y] = Input location no. of source Y
[F] = Fixed Data (user specified floating point number)

***** 30 Z=F, LOAD FIXED DATA *****

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 x 10^{-19} to 9 x 10^{18}. A value smaller than the minimum is set to 0, while a larger value is set to the maximum.

PAR. NO.	DATA TYPE	DESCRIPTION	
01:	FP	Mantissa	[F]
02:	2	Exponent of 10 (Hit C to change sign)	
03:	4	Input location no. destination	[Z]

Input locations altered: 1

***** 31 Z=X, MOVE INPUT DATA *****

FUNCTION
Move data from one input location to another.

PAR. NO.	DATA TYPE	DESCRIPTION	
01:	4	Input location no. of source	[X]
02:	4	Input location no. destination	[Z]

Input locations altered: 1

***** 32 Z=Z+1, INCREMENT INPUT LOCATION *****

FUNCTION
Add 1 to the value in the specified input location.

PAR. NO.	DATA TYPE	DESCRIPTION	
01:	4	Input location no. destination	[Z]

Input locations altered: 1

***** 33 X + Y *****

FUNCTION
Add X to Y and place the result in a third input location.

PAR. NO.	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.

PAR. NO.	DATA TYPE	DESCRIPTION	
01:	4	Input location of X	[X]
02:	FP	Fixed addend	[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.

PAR. NO.	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

*** 36 X * Y ***

FUNCTION
Multiply X by Y and place the result in an input location.

PAR. NO.	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.

PAR. NO.	DATA TYPE	DESCRIPTION	
01:	4	Input location of X	[X]
02:	FP	Fixed multiplier	[F]
03:	4	Dest. input location for X * F	[Z]

Input locations altered: 1

*** 38 X / Y ***

FUNCTION
Divide X by Y and places the result in an Input location. Division by 0 will cause the result to be set to the maximum datalogger number (+99999).

PAR. NO.	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.

PAR. NO.	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.

PAR. NO.	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.

PAR. NO.	DATA TYPE	DESCRIPTION	
01:	4	Input location of X	[X]
02:	4	Dest. input location 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.

PAR. NO.	DATA TYPE	DESCRIPTION	
01:	4	Input location no. of source 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.

PAR. NO.	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 ***

FUNCTION
Take the fractional (FRAC) value (i.e., the non-integer portion) of X and place the result in an input location.

PAR. NO.	DATA TYPE	DESCRIPTION	
01:	4	Input location of X	[X]
02:	4	Dest. input loc. for FRAC(X)	[Z]

Input locations altered: 1

*** 45 INTEGER VALUE ***

FUNCTION
Take the integer (INT) value of X and place the result in an input location.

PAR. NO.	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.

PAR. NO.	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.

PAR. NO.	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)).

PAR. NO.	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.

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.

PAR. NO.	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.

PAR. NO.	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.

PAR. NO.	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

*** 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.

PAR. NO.	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.

PAR. NO.	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

Intermediate Storage: 0

*** 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 ±0.00001 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 in order to increase the number of significant digits.

PAR. NO.	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* Reps

*** 56 SATURATION VAPOR PRESSURE ***

FUNCTION
Calculate saturation vapor pressure (over water SVPW) in kilopascals from the air temperature (°C) and place it in an input location. The algorithm for obtaining SVPW from air temperature (°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°C to -50°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.

PAR. NO.	DATA TYPE	DESCRIPTION	
01:	4	Input location of air temperature °C	[TEMP]
02:	4	Dest. input location for saturated vapor pressure	[VP or Z]

Input locations altered: 1

*** 57 VAPOR PRESSURE FROM *** WET-/DRY-BULB TEMPERATURES

FUNCTION
This instruction calculates vapor pressure in kilopascals from wet- and dry-bulb temperatures in °C. The algorithm is of the type 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.

PAR. NO.	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	Dest. 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.

$$F(X_i) = W^*X_i + F(X_{i-1}) * (1-W)$$

Where X = input sample
W = user entered weighting function, $0 < W < 1$
If W=O, $F(X_i)$=O; if W=1, $F(X_i)$=X
$F(X_{i-1})$ = 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.

PAR. NO.	DATA TYPE	DESCRIPTION
01:	2	Repetitions [REPS]
02:	4	First input location for input data [X]
03:	4	Dest. input location for filtered data [F(X) or Z]
04:	FP	Weighting function, W [W]

Input locations altered: 1 for each 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 Bridge Measurement Programs (with appropriate multiplier and offset) and R_f represents the MULTIPLIER value. The result of Instruction 59 is stored in the same location that X was.

PAR. NO.	DATA TYPE	DESCRIPTION
01:	2	Repetitions [REPS]
02:	4	Starting input location and destination [X]
03:	FP	Multiplier (R_f) [MULT.]

Input locations altered: 1 for each repetition

*** 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.

PAR. NO.	DATA TYPE	DESCRIPTION
01:	4	Source input location
02:	4	Destination input location

Input locations altered: 1

8.3 OUTPUT PROCESSING INSTRUCTIONS

The "given output interval" described in the following instructions refers to the output interval defined in the first parameter of Instruction 84, Data Table Interval.

*** 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, but the sample is not used in the computation of wind direction or standard deviation. 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, 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].

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.

PAR. NO.	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	4	Samples per sub-interval (number of scans)
03:	2	Sensor/Output 2 digits: A B A Sensor type: 0 = Speed and Direction 1 = East and North B Output option: 0 **S, θ1, σ(θ1)** 1 **S, θ1** 2 **S, U, θu, σ(θu)**
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) for each 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

[1] EPA On-site Meteorological Program Guidance for Regulatory Modeling Applications.

standard deviations. The last sub-interval is weighted if it does not contain the specified number of scans.

There are three Output Options, which specify the values calculated.

Option 0:

Mean horizontal wind speed, **S**.
Unit vector mean wind direction, **θ1**.
Standard deviation of wind direction, **σ(θ1)**.

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, **θ1**.

Option 2:

Mean horizontal wind speed, **S**.
Resultant mean wind speed, **U**.
Resultant mean wind direction, **θu**.
Standard deviation of wind direction, **σ(θu)**.

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, θ1:

$$\theta 1=\text{Arctan}\ (Ux/Uy)$$

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, σ(θ1), 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, σ(θu), 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 and the end of the output interval is reached.

PAR. NO.	DATA TYPE	DESCRIPTION
01:	4	Repetitions
02:	4	Starting input location no.

Outputs generated: 1 for each sample

*** 71 AVERAGE ***

FUNCTION
This instruction stores the average value over the given output interval for each input location value specified.

PAR. NO.	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	4	Starting input location no.

Outputs generated: 1 for each input location

*** 72 TOTALIZE ***

FUNCTION
This instruction stores the totalized value over the given output interval for each input location specified.

PAR. NO.	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	4	Starting input location no.

Outputs generated: 1 for each input location

*** 73 MAXIMIZE ***

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 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 SECONDS
10	Output value with HOUR-MINUTE
11	Output value with HR-MIN,SEC

PAR. NO.	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 or 2 with time of max. option)

*** 74 MINIMIZE ***

FUNCTION
Operating in the same manner as Program 73, this instruction is used for storing the MINIMUM value sensed (for each input location specified) over a given output interval.

PAR. NO.	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 or 2 with time of min. 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, 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	$10^{o}C$
Upper range limit	$30^{o}C$
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).

PAR. NO.	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

*** 79 SAMPLE ON MAXIMUM OR MINIMUM ***

FUNCTION
Used in conjunction with Instructions 73 or 74, this Instruction copies specified input location values into Intermediate Storage whenever a previous Maximize or Minimize Instruction senses a new maximum or minimum value. When the Output Flag is set, the values copied to Intermediate Storage will be transferred to Final Storage.

Instruction 79 looks for a flag that is set by Instruction 73 or 74 when a new maximum or minimum is sampled. This flag is cleared at the start of a Maximize or Minimize Instruction or at the beginning of the program table. If Instruction 73 or 74 has more than 1 repetition, there is no way for Instruction 79 to know which Input location caused the maximum or minimum flag to be set. Thus, Instruction 79 should directly follow the maximum or minimum Instruction to which it refers. If sampling is to occur only when one specific input location shows a new maximum or minimum value, the previous Maximize or Minimize Instruction should only refer to that input location (1 rep).

PAR. NO.	DATA TYPE	DESCRIPTION
01:	2	Repetitions (number of sequential locations to sample)
02:	4	Starting input location no.

Outputs generated: 1 for each repetition

*** 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 N is the number of samples and X_i is an individual measurement (i=1 to N).

PAR. NO.	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	4	Starting input location no.

Outputs generated: 1 for each repetition

8.4 PROGRAM CONTROL INSTRUCTIONS

TABLE 8.4-1. Command Codes

0	-	Go to end of program table
1-9, 79-99	-	Call Subroutine 1-9, 79-99
11-19	-	Set Flag 1-9 high
21-29	-	Set Flag 1-9 low
30	-	Then Do
31	-	Exit loop if true
32	-	Exit loop if false
41	-	Set Port high
51	-	Set Port low

TABLE 8.4-2. Flag Description

Flag 1 to 8	-	User Flags
Flag 9	-	Intermediate Processing Disable Flag

*** 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.

PAR. NO.	DATA TYPE	DESCRIPTION
01:	FP	Fixed value
02:	2	Command (Table 8.4-1)

*** 84 DATA TABLE INTERVAL ***

FUNCTION
Data are stored in Data Tables. The Data Table Interval Instruction defines how often and what data are output. Data output at the specified interval constitutes a record.

Output Processing Instructions (Instructions 70 through 82) following the Data Table Interval Instruction define what data are included in the Table. The first Output Processing Instruction outputs the first data point of the record.

The number of records to store in a Data Table is specified in parameter 2 of Instruction 84. For example, if the interval is 30 minutes, and the time between site visits is 30 days, a minimum of 1440 records must be specified (48 records/day * 30 days = 1440 records). The actual value entered should generally be greater than the calculated minimum to provide a margin of safety. When the Data Table is full, the newest record deletes the oldest record by writing over it. To allocate all remaining memory, enter 99999 for parameter 2.

The number of Data Tables is limited by storage memory. At compile time, an E31 error indicates that too much space is requested, and an E28 error means no Output Processing Instructions follow Instruction 84.

PAR. NO.	DATA TYPE	DESCRIPTION
01:	4	Interval in minutes (XXXX, 1440 max) or seconds (XXXX-) 0 = the program table interval 0-- = output each time Inst. 84 is executed, date and time are stored with data and use 2 data locations. 900X = output when user flag X (1-8) is high
02:	5	Storage space to allocate for this Data Table, in units of records. 99999 for all remaining storage space. If a negative number is entered (-XXX), output is redirected to Input Storage starting an the input location XXX.

*** 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. 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). A subroutine can not call itself or call another subroutine (or series of subroutines) which calls it. This will result in a run time error: When the subroutine is called the second time, error 10 is logged in the Error Log. Execution will not branch to the subroutine; it will continue with the Instruction following that calling the subroutine.

PAR. NO.	DATA TYPE	DESCRIPTION
01:	2	Subroutine number (1-9, 79-99)

*** 86 DO ***

FUNCTION
This Instruction unconditionally executes the specified command.

PAR. NO.	DATA TYPE	DESCRIPTION
01:	2	Command (Table 8.4-1)

*** 87 LOOP ***

FUNCTION
Instructions included between the Loop Instruction (87) 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. Results from Output Processing Instructions contained in a loop are output to Input Storage on the last loop iteration. The results from the Output Processing Instructions are stored in the input Locations that they were acting on.

The first parameter, pass duration, controls how frequently passes through the loop are made. Duration units are in 0.1 seconds. A duration of 0 means there is no delay between passes through the loop. A duration of 10 means passes through the loop are 1 second apart. Pass durations less than a second cause the quiescent current drain to increase to 3 mA while in the loop. All iterations of the loop are completed before continuing with the table. Only those instructions within the loop are executed and other portions of the table are not executed in the interim. (At this time if a value less than 10 is entered, the BDR will wait the specified amount of time after executing the instructions in the loop rather than starting the pass on the interval.)

When a fixed number of iterations are executed, the time spent in the loop is equal to the product of the pass duration and the number of iterations (pass duration ≥ 10). For example, a loop with a delay of 10 and a count of 5 will take 5 seconds, assuming the contents of the loop may be executed within 1 second. 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 defines the end of loop.

If the total duration of a loop with delay exceeds the table execution interval, the table will not be initiated at each execution interval. Once the loop is over the table execution will resume on the next even interval.

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.

Input Locations for Output Processing Instructions may not be 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.

Loops with loop durations less than 1 second (parameter 1 ≤ 9) can be nested. Indexed locations within nested loops are indexed to the inner most loop. The maximum nesting level is 9 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.

PAR. NO.	DATA TYPE	DESCRIPTION
01:	2	Pass duration in tenths of seconds
02:	4	number of passes

*** 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 8.4-3.

PAR. NO.	DATA TYPE	DESCRIPTION
01:	4	Input location for X
02:	2	Comparison code (Table 8.4-3)
03:	4	Input location for Y
04:	2	Command (Table 8.4-1)

TABLE 8.4-3. Comparison Codes

Parameter 1	Function
1	=
2	≠
3	≥
4	<

*** 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 8.4-3.

PAR. NO.	DATA TYPE	DESCRIPTION
01:	4	Input location for X
02:	2	Comparison code (Table 8.4-3)
03:	FP	Fixed value
04:	2	Command

*** 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.

PAR. NO.	DATA TYPE	DESCRIPTION
01:	2	Increment for the loop index counter

*** 91 IF FLAG/PORT ***

FUNCTION
This Instruction checks the status of one of the nine Flags or port 2, 3, or 4 and conditionally performs the specified Command.

The first Parameter specifies the condition to check:

1X Execute command if Flag X is high (X=1-9)
2X Execute command if Flag X is low (X=1-9)
4Y Execute command if Port Y is high (Y=2-4)
5Y Execute command if Port Y is low (Y=2-4)

PAR. NO.	DATA TYPE	DESCRIPTION
01:	2	Flag or Port condition to check
02:	2	Command (Table 8.4-1)

*** 92 IF TIME ***

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 number of minutes 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.

The Command resulting from a true condition is executed only once in any minute (e.g., if the command is to execute a subroutine, and the execution interval of the table is 10 seconds, the subroutine will executed only once in the minute, not 6 times.)

PAR. NO.	DATA TYPE	DESCRIPTION
01:	4	Time into interval (minutes, xxx-- seconds)
02:	4	Time interval (xxx minutes, xxx-- seconds)
03:	2	Command (Table 8.4-1)

*** 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.

PAR. NO.	DATA TYPE	DESCRIPTION
01:	4	Input location for subsequent comparisons

EXAMPLE:

01:P93	Begin CASE
01:0002	Case location
02:P83	IF Case location
01:69.4	< 69.4, THEN
02:3	Execute command 3 (subroutine 3) else
03:P83	IF Case location
01:72	< 72, then
02:10	Set Output Flag else

03:P83	IF Case location
01:77.3	< 77.3, then
02:30	DO
. . . .	
08:P95	END OF THEN-DO
09:P95	END 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. The Else Instruction is optional; when it is omitted, a false comparison will result in execution branching directly to the End Instruction (refer to Section 7.10). 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 If Case sequence (Instruction 93). The End Instruction has no parameters.

*** 100 DO NOTHING ***

PAR. NO.	DATA TYPE	DESCRIPTION
01:	4	Input location to do nothing to

SECTION 9. COMPUTER COMMANDS

It is not necessary for the casual user to understand the computer commands described in this section because TERM3 has options which automatically execute the commands. Some commands such as Data Notes (Section 9.8) and Set Offset (Section 9.10) are not available in the TERM3 menu and must be given after entering TERM3's terminal emulator option.

9.1 COMMAND BASICS

A computer/terminal with its RS232 port connected to the datalogger's RS232 connector is used to retrieve data and program the datalogger. The emphasis of this section is on the commands used when manually (i.e., keyed in by hand) interrogating or programming the datalogger via a computer/terminal. These commands and the responses to them are sent in the American Standard Code for Information Interchange (ASCII). The commands allow the user to perform several operations including:

check BDR status

set real time clock

retrieve and review Data Tables

enter and retrieve data notes

monitor inputs

adjust offsets

send/receive or edit programs

The RTS pin on the RS232 connector must be pulled high to "ring" the datalogger prior to communication. Several carriage returns (CR) must be sent to the datalogger to allow it to set its baud rate to that of the computer (300, 1200, 9600, or 76,800). Once the baud rate is set, the datalogger will send back the prompt

H = Help, errors:nnnn
*

signaling that it is ready to receive a command. "nnnn" is the number of run time errors accumulated since the error log was last erased (Section 9.6).

9.1.1 GENERAL RULES GOVERNING COMMANDS

1. * from datalogger means "ready for command".
2. **CR** (Carriage Return or ENTER) to datalogger means "execute".
3. **CRLF** (Carriage Return, Line Feed) from datalogger means "executing command".
4. Commands are letters, optionally preceded by number parameters, and terminated with **CR**. The response includes a checksum for all ASCII commands.
5. 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 datalogger starts the count over; e.g., if the sum of the ASCII values is 8192, the checksum is 0.
6. An illegal character increments a counter and zeros the command buffer, returning a *.
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.

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

After the datalogger initially responds or completes a command, it waits about 250 seconds for a valid character to arrive. If it does not receive a valid character in this time interval, it "hangs up". The datalogger counts all the invalid characters it receives from the time it answers a ring, and terminates communication after receiving 150 invalid characters.

The datalogger continues to execute its measurement and processing tasks while servicing the commands. If the measurement and processing overhead is large, response to the communication commands is slower.

9.1.2 COMMAND DESCRIPTIONS

In the following descriptions, the command and a brief description are listed in italics at the start of each section. Numeric entries are listed as lower case abbreviations. For example:

mo/day/yr hr:min:secC

means that to set date and time you enter something like

6/23/91 17:53:00C

Remember that the command is given by typing the command (as above) and pressing **ENTER**.

When TERM3 has an option which automatically executes a command, the option is listed in the description.

9.2 HELP COMMAND

H Help menu.

When the H command is issued, the following prompts are displayed on the computer/terminal screen:

A	Status
nU	1U=ErrLog, 1111U erases; 2U=TimSetLog, 2222U erases
D	Send Data: n,b,t,sD (b,t,s optional) n Table Number b records back to start; no entry = since last t Time tag; no entry = w/o, 1 = w/ s records to send; no entry = to most recent
4321S	Enter a Data Note
R	Send Data Notes since last; 1R=all notes
C	Display/Set date,time
nI	Display/Set input location n
n,mI	Measure/Display m locations starting at n
nY	Adjust offset at location n
7W	Prompt programming; 1W=List
7H	Direct programming
nL	Unlock Security with code n
12X	Transparent to SDI-12 sensor
1986T	(Repeat) Power reset

Commands not shown in the HELP list are 2718Q, used to program the BDR320 from a computer file, and 2718,1Q, used to save a BDR320 program to a computer file.

9.3 STATUS

A Status

TERM3 automatically executes this command when the A (stAtus) option is selected. On receiving the A command the datalogger sends the following status information.

Time mm/dd/yy hh:mm:ss	=	Real time
NextExe hh:mm:ss	=	Time of next program or calibration execution
Battery:+xx.xxx	=	Datalogger power supply voltage
Input Locs:xxxx	=	Input Locations allocated
Prgm Bytes:xxxx	=	Program bytes used
Storage:+xxxxx.	=	Storage locations available for Data Tables
Unused:+xxxxx.	=	Unused storage locations
Data Tables:xxxx	=	Number of Data Tables in program
Prgm Sig:+xxxxx.	=	Program signature
EE Sig:+xxxxx.	=	EEPROM (calibration) signature
EE Ver:xx	=	Calibration version
Cal:mm/dd/yy	=	Date of calibration
PROM Sig:+xxxxx.	=	PROM signature
Errors = xxxx	=	Total number of runtime errors since erasing error log
PROM ID:+xxxxx	=	PROM identification number
OBJSrlNo	=	PROM revision
SN:+xxxx.x	=	datalogger serial number
Cxxxx	=	Checksum

9.4 SETTING AND DISPLAYING THE CLOCK - C COMMAND

C	*display time*
hr:min:secC	*set time*
mo/day/yr hr:min:secC	*set date and time*

TERM3 automatically executes this command when the K (clocK view/set) option is selected. The C command is used to display or change time. To display time, send C only.

The clock is always assumed to be correct. Time tags for data records are calculated from the clock time at which the most recent record was recorded. Thus, the next record stored after changing the clock (by more than the recording interval) will change the times calculated for all records in the data table. The Time Set Log (Section 9.7) records the eight most recent times that the clock is set through the C command.

Errors are logged if the time between records is different from the interval set in Instruction 84, and the time has not been changed with the C command.

9.5 DATA RETRIEVAL

n,b,t,sD *Send Data; b, t, and s, are optional*
- *n = Table Number*
- *b = No. of records back to start, no entry = since last*
- *t = Time Tag; no entry = w/o, 1 = w/*
- *s = No. of records to send, no entry = to most recent*

Alternate format:
n,time,sD
time = mo:dd:yy hh:mm:ss

Data are retrieved from the datalogger with the above commands. Options include collecting all records since the last D command, or a number of records from a specified Data Table. Time tags are optional in both cases.

The D command has a number of optional parameters, if the default conditions are desired it is not necessary to enter a parameter (entering 0 is the same as no entry, resulting in the default response). The appropriate number of commas must be entered if one parameter is left default and the next has a specified value. For example:

1D	Send all data collected in Data Table 1 since the last "since last" command to Table 1.
1,25,,5D	Table 1, go back 25 records, do not time tag, send 5 records

Time tags are not stored in the Data Table. When data are retrieved time is generated based on the time of the most recent output and the output table interval.

When storing data, error 12 is logged if the time between records is different from the interval set in Instruction 84, and the time has not been changed with the C command. Resetting the datalogger clock does not cause error 12.

If there is a gap in data where the clock jumps forward (e.g., shutdown due to low batteries), the number of recording intervals skipped is stored in the Data Tables. When the data are retrieved, the time discontinuity is represented by a carriage return for each record skipped.

If for some reason the clock jumps backwards (e.g., it is bombed by a transient from a lightning strike), there is no break in the retrieved data. Error 12 in the error log and the incorrect time provide a record of the incident.

For conditional output (not based on time) time is automatically saved with each record.

The following examples show the response to requesting the last three records of an hourly stage height Data Table, with and without time tags. Notice that the datalogger serial number, table number, time of the first output transmitted, table interval and element labels are header information generated by the datalogger from the program.

Most recent 3 arrays with time:

```
*1,3,1D
SN:+1026.0 Table:0001 Time:08/28/89 11:00:00 Interval:0060 min
MM/DD/YY HH:MM:SS STAGE
                           FT___S
08/28/89 11:00:00 +01.00
08/28/89 12:00:00 +01.00
08/28/89 13:00:00 +01.00
   C2104
```

Request most recent 3 arrays without time:

```
*1,3D
SN:+1026.0 Table:0001 Time:08/28/89 11:00:00 Interval:0060 min
STAGE
FT___S
+01.00
+01.00
+01.00
   C6353
```

9.6 ERROR LOG

1U Send error log
1111U Erase error log

TERM3 collects the error log, time set log, and data notes whenever data are collected (options U or R). A log of how many run time errors have occurred, and the first and most recent times of occurrence, is kept in memory and sent in response to the 1U command. Entering 1111U erases the Error log.. The format of the error log is:

cc nn mo/day/yr hr:min:sec mo/day/yr hr:min:sec

where:
cc = error code
nn = number of occurrences of the error
mo/day/yr hr:min:sec = first and most recent occurrences of error

Compiling and editor errors are displayed on the computer screen, but are not stored in the error log. (Section 5.11)

ERROR LOG CODE DEFINITION

01 EEPROM coefficients not programmed
02 Measurement hardware overranging
03 Unsuccessful datalogger calibration
04 Internal thermistor out of range
05 Start of low power supply condition
06 End of low power supply condition
07 Signature of PROM, program, and EEPROM does not match value stored. (New signature is then stored.)
08 Datalogger reset by watchdog timer
09 Insufficient storage space
10 Outer subroutine called from nested inner subroutine
11 Program table overrun
12 Output did not occur at programmed interval
13 Power-up test failed
14 1) Clock set back.
   2) Watch dog or overrun where time difference is greater than 255 sec.

Once a day at midnight, the signature of the PROM, program, and EEPROM is calculated and compared with the value it has stored. If the calculated and stored value do not match, error 7 is logged and the newly calculated signature is stored.

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 hardware problem or a software bug and should be reported to Campbell Scientific.

When storing data, error 12 is logged if the time between records is different from the interval set in Instruction 84, and the time has not been changed with the C command. Resetting the datalogger clock does not cause error 12.

Error 13 indicates that a fault in RAM, PROM, or CPU memory was detected on power-up. DO NOT CONTINUE TO USE THE DATALOGGER IF THIS ERROR OCCURS. Contact Campbell Scientific to discuss the problem and arrange for the repairs.

An example error log is shown below.

ErrorLog:
E02 99 04/20/89 09:45:00 05/16/89 13:15:00

The example shows that the measurement hardware, error 2, has overranged 99 times or more between April 20 at 9:45 AM and May 16 at 1:15 PM. The maximum number of occurrences that may be recorded is 99.

9.7 TIME SET LOG

2U *Send time set log*
2222U *Erase time set log*

TERM3 collects the error log, time set log, and data notes whenever data are collected (options U or R). Entering **2U** returns a log of the last eight times the datalogger clock was set. **2222U** erases the time set log. The format is:

Previous time	Time set to
mo/day/yr hr:min:sec	mo/day/yr hr:min:sec

9.8 DATA NOTE STORAGE AND RETRIEVAL

R *Send Data Notes since last R command.*
1R *Send all Data Notes, beginning with oldest.*
4321S *Enter a Data Note, ESCape quits.*

TERM3 collects the error log, time set log, and data notes whenever data are collected (options U or R). Up to 1000 characters may be entered as Data Notes using the 4321S command. Editing commands include:

Backspace to backup and delete
Right arrow (^S) for forward
Left arrow (^D) for back

The R command tells the BDR to send only those notes entered since the last R command. The 1R command tells the BDR to send all notes in the memory.

9.9 DISPLAY INPUTS

nI *Display value in Input Location n and permit entry of new value.*
n,mI *Measure/display m Input Locations starting with location n. Sensors are measured if measurement interval > 5 s.*

The I command is used to change the value in an Input storage location or to view one or more locations. If the program execution interval is greater than or equal to five seconds, the n,mI command causes Input/Output Instructions 1, 2, 5, 6, 7, 9, 11, and 29 to be executed. If the table execution interval is less than five seconds, the values displayed are the result of the normal table execution.

9.10 ADJUST OFFSET

nY *Adjust the offset for the measurement that loads input location n.*

The Y command is used to set the offset applied to a sensor reading. For example, to correct a water level reading to match the staff gage. The Y command allows the measurement offset to be edited without entering a programming mode (W or 7H commands). Program changes through the Y command do not cause the datalogger to recompile. Offset adjustments and all other changes made in the 7H mode require the datalogger to recompile, resulting in a loss of data.

The Y command operates on Input Locations assigned in measurement instructions 1, 2, 5, 6, 7, or 29. When the command is sent, the BDR responds with the location number or label, the current offset, and the current reading. Key in the desired reading, press **Return**, and the BDR calculates the necessary offset. The new offset and current reading will then be displayed. Press **ESC**ape to get back to the command state.

Adjusting the offset changes the datalogger program without the need of recompiling. The program signature will change when an offset is changed. The error log will show an error 7 at midnight the night after an offset was changed; this provides a record that something was changed in the program. If the actual value of the offset is to be stored, the program should be saved or an entry made in the data notes.

9.11 PROGRAM DATALOGGER, SAVE, LOAD, AND DELETE PROGRAMS

7W	*Enter Prompt Programming Mode (see Section 5), ANSI standard characters are sent for "up line" and "clear screen".*
1W	*List Program*
7H	*Enter Direct Programming Mode (*1 Mode) to key in, edit, and review program. *A or *C can then be keyed to change to the *A or *C Mode.*
2718Q	*Program datalogger using a program file*
2718,1Q	*Save a program file*
1986T	*Soft reset. Command needs to be repeated. CAUTION: THE PROGRAM AND ALL DATA ARE DELETED BY ISSUING THIS COMMAND*

TERM3 automatically executes the 7W command when the P (Prompt Programming) option is selected. The 7W command prompt programming method is for basic applications requiring measurement, output processing and data storing instructions. Prompt programming also accommodates setting the control port and automated short term measurement averaging.

The 7H command enters the *1 Mode programming table where programs developed using the direct programming method may be edited, or programs developed in either the prompt or direct programming methods may be reviewed. The *1 Mode is rarely used to develop a program. Program development using the direct programming approach is commonly done with EDLOG, a program development software module contained in Campbell's PC208 software package.

Programs generated using the prompt programming method must not be edited in the *1 Mode. Programs developed in the direct programming method (EDLOG3), must not be edited in the prompt programming mode (7W command).

9.12 SECURITY

nL Unlock security with code number n

Three levels of security are possible, as described in Section 7.1.3. Each level is set in the *C Mode using a four character code number, n. Sending nL unlocks that level of security using n as the code number.

9.13 EXIT

E Exit telecommunications.

Sending E and CR terminates communication with the datalogger. TERM3 automatically sends the E when the Q (Quit) option is selected.

9.14 SUMMARY OF COMMANDS

TABLE 9.14-1. Commands

Command	Description
A	Status (8888A Clears Error Log first). Time mm/dd/yy hh:mm:ss = Real time NextExe hh:mm:ss = Time of next program or calibration execution Battery:+xx.xxx = Datalogger power supply voltage Input Locs:xxxx = Input Locations allocated Prgm Bytes:xxxx = Program bytes used Storage:+xxxxx. = Storage locations available for Data Tables Unused:+xxxxx. = Unused storage locations Data Tables:xxxx = Number of Data Tables in program Prgm Sig:+xxxxx. = Program signature EE Sig:+xxxxx. = EEPROM (calibration) signature EE Ver:xx = Calibration version Cal:mm/dd/yy = Date of calibration PROM Sig:+xxxxx. = PROM signature Errors= xxxx = Total number of run time errors since erasing error log PROM ID:+xxxxx = PROM identification number OBJSr1No = PROM version SN:+xxxx.x = datalogger serial number Cxxxx = Checksum
C	Display/set time. C to display time hr:min:secC sets time. mo/day/yr hr:min:secC sets date and time.
E	Exit telecommunications.
H	Help menu.
	Display Inputs/Enter Values
nI	Display value in Input Location n and permit entry of new value.
n,mI	Display m Input Locations starting with location n. Measure sensors if measurement interval is < 5 sec.
nY	Adjust the offset for the measurement that loads input location.

Data Retrieval

n,b,t,sD	Send Data; b, t, and s, are optional n = Table Number b = No. of records back to start, no entry = since last t = Time Tag; no entry = w/o, 1 = w/ s = No. of records to send, no entry = to most recent
Alternate format: n,time,sD	time=mm:dd:yy hh:mm:ss

Program Datalogger

7W	Enter Prompt Programming Mode (see Section 5), ANSI standard characters are sent for "up line" and "clear screen".
1W	List the program generated in the W mode.
7H	Enter Direct Programming Mode (*1 Mode) to key in, edit, and review program. *A or *C can then be keyed to change to the *A or *C Mode.
2718Q	Program datalogger using a program file
2718,1Q	Save a program file
1986T	Soft reset. Command needs to be repeated. CAUTION: THE PROGRAM AND ALL DATA ARE DELETED BY ISSUING THIS COMMAND

Data Notes

R	Get Data Notes since last R command.
1R	Get all Data Notes, beginning with oldest.
4321S	Enter a Data Note, ESCape quits.

Security

nL	Unlock Security (*C Mode) with code word n. The L command temporarily changes the security level. After terminating communications, security is reset.

SDI-12

12X	Communicate with SDI-12 sensor; datalogger is transparent. Datalogger program is interrupted for about 0.25 seconds after sending a command.

Error and Time Set Logs

1U	Send error log
1111U	Erase error log
2U	Send time set log
2222U	Erase time set log

APPENDIX A. DESICCANT

The BDR320 is shipped with two 4 unit packets of DESI PAK desiccant. An active packet should be left in the enclosure at all times. Figure A-1 may be used to determine the maximum time that should be allowed before changing desiccant (one 4 unit bag with the enclosure closed but vented).

The manufacturer recommends that the desiccant be dried by placing it in a oven at 120 °C (250 °F) for 16 hours. Some people have experienced failure of the packets when using this technique. The failures appear to be due to excessive heat from radiative transfer from the heating coils or conduction from the metal shelves. These problems can be reduced by placing the packets in a brown paper bag in the oven. If one of the packet's seams opens during the heating/drying process it will usually reseal if pressed together when removed from the oven.

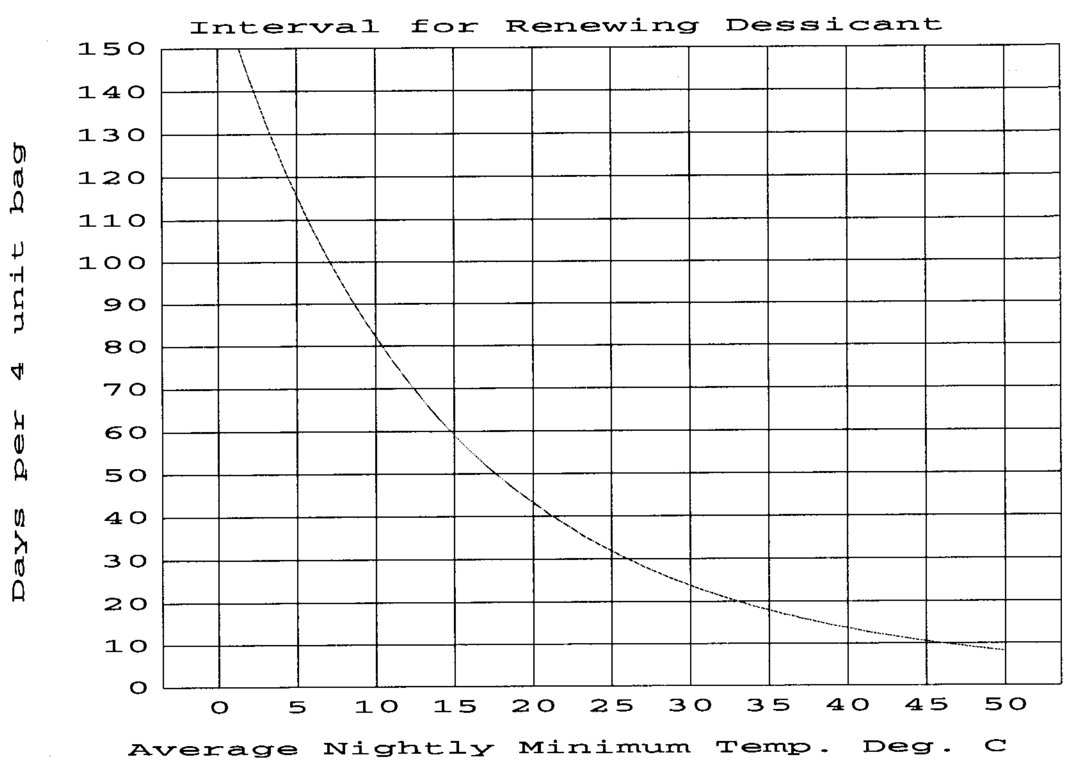


Figure. A-1. Maximum Recommended Interval for Changing Desiccant

APPENDIX B. CAMPBELL SCIENTIFIC POWER SUPPLIES

The PS12 Power Supply is available with either alkaline or lead acid batteries, the PS12 ALK and PS12 LA, respectively. The PS12 ALK has eight D cell alkaline batteries, the PS12 LA has a rechargeable lead acid battery. The alkaline batteries are discarded after use. The lead acid batteries should be float charged with either AC power or a solar panel. The lead acid battery supplies power during a power failure or in times of low charge with a solar panel.

The PS12 Power Supply provides 12 volts, regulates incoming AC or DC power, limits current from the battery, and provides circuitry to connect an external 12 volt battery. The terminals on the PS12 are exposed by unscrewing the two set screws, as shown in Figure B-1.

The two 12 volt and two ground terminals are for supplying power to the datalogger, or other 12 volt devices.

The two terminals, labeled CHG, are for connecting a 20V DC adaptor or solar panel to charge lead acid batteries.

The charge input can be either AC or DC, and it does not matter which terminal is positive or negative. The voltage input must be within 16 to 26V DC, or 16 to 26V AC RMS.

The ON-OFF switch controls power to the 12V ports. Charging of lead acid batteries still occurs when the switch is off. The red charge light is on when the battery is being charged by AC power or a solar panel.

The connectors, labeled INT and EXT, are for connecting the internal (power supply) battery and an external battery, respectively. A six foot cable, with connector, is included with the power supply for connection to an external battery. This is commonly used for supplying power to the datalogger while changing power supply batteries.

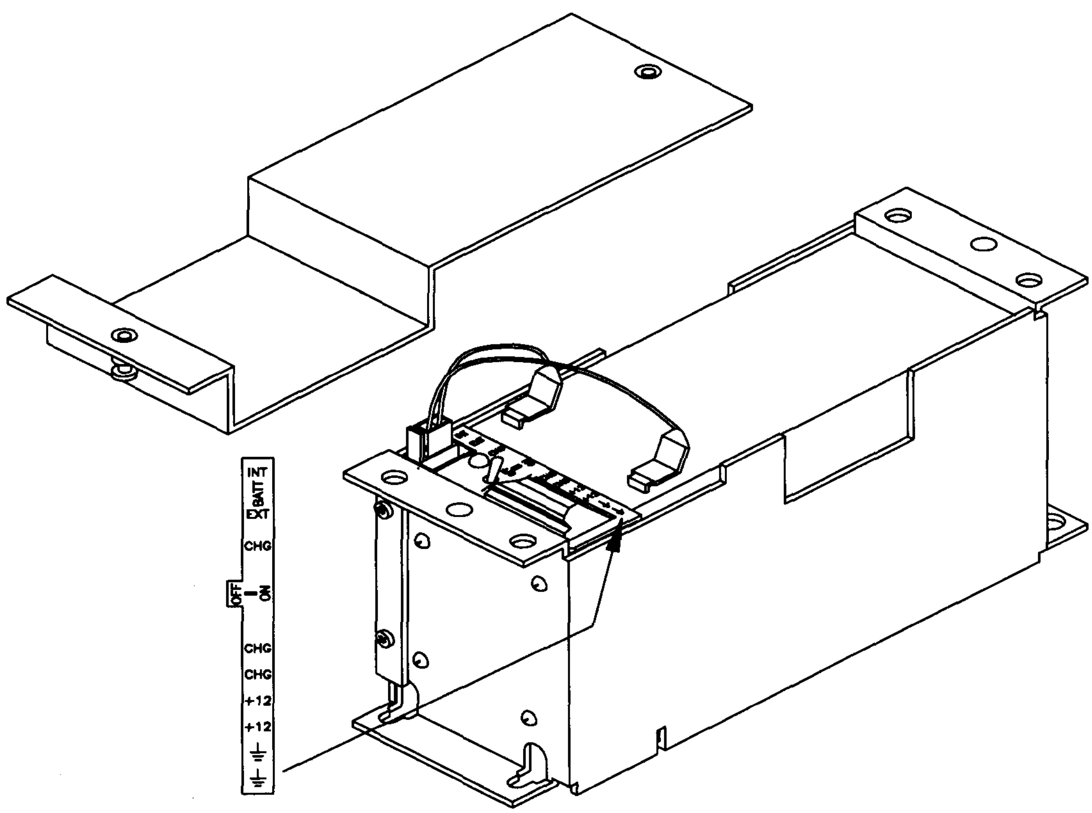


FIGURE B-1. PS12 12 Volt Power Supply and Charging Regulator

A thermal fuse (1.85 amp) in the power circuit limits source current. If excessive current is drawn, the fuse gets hot, increases in resistance, and limits current. When the problem is fixed, the fuse cools and the resistance decreases, eventually allowing current to pass. When excessive current is drawn due to shorting the power leads to the Wiring Panel, allow 10 to 15 seconds for the fuse to cool before connecting power.

B.1 PS12 ALK ALKALINE POWER SUPPLY

The PS12 ALK utilizes eight alkaline D cells mounted in place of the lead acid battery shown in Figure B-1. The PS12 ALK can also be used with a lead acid battery connected to the external battery port; in this case the alkaline batteries act as a backup.

Before installing the alkaline batteries, connect all necessary sensor leads, control lines, and power leads. The datalogger can be turned on and off with the switch on the PS12 ALK.

To replace the batteries without losing the datalogger program and data: 1) do not turn the power switch off, 2) connect an external battery to the port labeled EXT with the supplied six foot cable, 3) remove the old batteries, 4) replace with new alkaline D cell batteries, and 5) remove the external battery.

A fresh set of eight alkaline D cells has 12.4 volts and a nominal rating of 7.5 amp-hours at 20^{o}C. The amp-hour rating decreases with temperature as shown in Table B-1. The "BATT" Prompt Programming Instruction (direct programming Instruction 10) can be used to monitor battery voltage. Replace the alkaline cells before the battery voltage drops below 9.6 V.

TABLE B-1. Typical Alkaline Battery Service and Temperature

Temperature (^{o}C)	**% of 20^{o}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.

The six foot external battery cable can be connected to a large capacity (amp-hour) battery to power the system for an extended period of time. The red and black leads connect to the positive and negative battery posts, respectively. The alkaline cells act as a power backup in this case. A diode in the ALK cable isolates the lead acid supply from the alkaline batteries.

B.2 PS12 LA LEAD ACID POWER SUPPLY

The PS12 LA power supply includes a 12V, 7.0 amp-hour lead acid battery, a AC transformer (20V DC), and a temperature compensated charging circuit with a charge indicating diode. An AC transformer or solar panel should be connected to the PS12 at all times. The charging source powers the datalogger while float charging the lead acid batteries. The internal lead acid battery powers the datalogger if the charging source is interrupted. The PS12 LA specifications are given in Table B-2.

If the battery is fully charged first, it may be stored for up to six months. A battery that has been stored without charging should be given a "top charge" by charging in the PS12 with the AC source for one day prior to being placed in service.

The two leads from the charging source can be inserted into either of the CHG ports, polarity doesn't 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.

Some solar panels are supplied with a connector. This connector must be clipped off so the two wires can be inserted into the two terminal ports. It is recommended that these two leads be stripped and tinned.

The red light (LED) on the PS12 LA is on during charging. The switch turns power on and off from the 12V ports (battery charging still occurs when the switch is off).

> **CAUTION:** Switch the power to "off" before disconnecting or connecting the power leads to the Wiring Panel. The Wiring Panel and PS12 LA are at power ground. If 12V is shorted to either of these, excessive current will be drawn until the thermal fuse opens.

The external port, labeled EXT, is not meant to be used with the PS12 LA. The primary power source is the charging source, and the secondary power source is the internal lead acid battery. Connecting a lead acid battery to the external source is the same as connecting two lead acid batteries in parallel, causing one battery to drop voltage and the other to raise voltage. Alkaline batteries connected to the external port would be charged by the charging source which can cause an explosion.

> **CAUTION:** Do not use the external port, labeled EXT, with the PS12 LA.

Monitor the battery voltage using the Prompt Programming Instruction "BATT" (Direct Instruction 10). Incorporate this instruction into your 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. The datalogger measures the voltage at its 12V port, not the voltage of the lead acid battery. External power sources must be disconnected from the charging circuit in order to measure the actual lead acid battery voltage.

TABLE B-2. PS12 LA Battery and AC Transformer Specifications

Lead Acid Battery	
Battery Type	Yuasa NA 7-12
Float Life @ 25^{o}C	5 years typical
Capacity	7.0 amp-hour
Shelf Life, full charge	Check twice yearly
Charge Time (AC Source)	40 hr full charge, 20 hr 95% charge
AC Transformer	
Input:	120V AC, 60 Hz
Isolated Output:	20V DC @ 350 mA max.

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 CR10 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.

B.3 SOLAR PANELS

Auxiliary photovoltaic power sources, such as Solarex Models MSX5, MSX10, and MSX18 Solar Panels may be used to maintain charge on lead acid batteries.

TABLE B-3. MSX5 and MSX10 Solar Panel Specifications

	MSX5	MSX10	MSX18
Typical Peak Power (Watts)	4.2	8.9	18.6
Current @ Peak (Amps)	.27	.59	1.06
Amp Hrs/week	6.4	14.4	26.4

> **NOTE:** Specifications assume 1 kW/m2 illumination at a panel cell temperature of 25°C. Individual panel performance may vary as much as 10%.

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.

APPENDIX C. USE OF DIGITAL I/O PORTS FOR SWITCHING RELAYS

Each of the four digital I/O ports can be configured as an output port and set low or high (0V low, 5V high) using Port commands 41 - 54 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 (1.5mA at 3.5V).

Campbell Scientific offers the A21REL-12 Four Channel Relay Driver (12V coil) for use with the BDR320.

Figure C-1 shows a typical relay driver circuit such as used in the A21REL-12 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, 12V from the datalogger passes through the relay coil, closing the relay which completes the power circuit to a fan, turning the fan on.

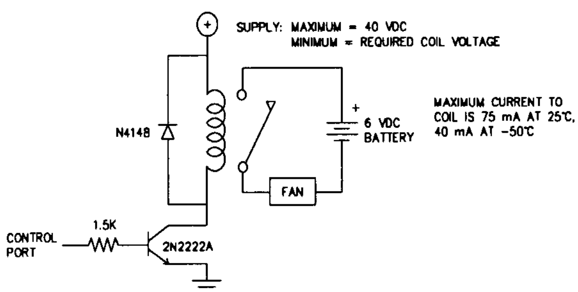


FIGURE C-1. Relay Driver Circuit with Relay

In other applications it may be desirable to simply switch power to a device without going through a relay. Figure C-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 C-1) would be required.

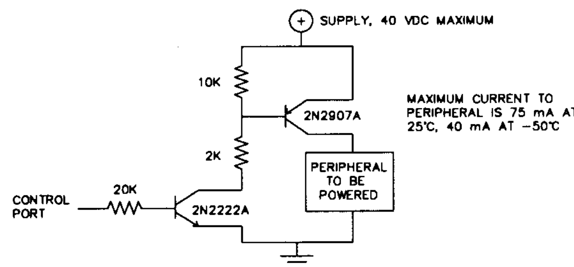


FIGURE C-2. Power Switching without Relay

Other control port activated circuits are possible for applications with greater current/voltage demands than shown in Figures C-1 and C-2. For more information, contact Campbell Scientific's Marketing Department.

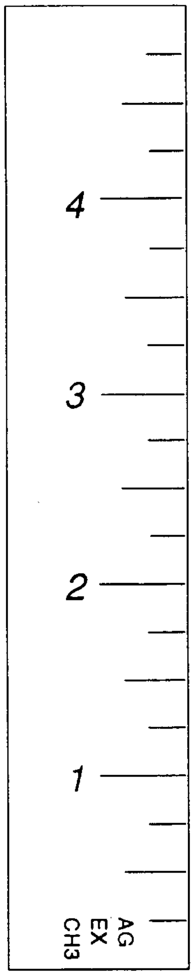
APPENDIX D. PROM WITH 50 HZ NOISE REJECTION

TABLE D-1. Input Voltage Ranges and Codes With 50 Hz PROM

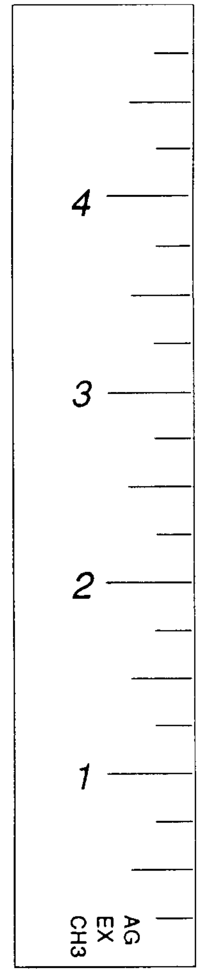
Range Code	Integration time, ms	Range, mV
00 (auto range)	up to 20	-20 to +5000
01*	60	-20 to +60
02*	40	-20 to +100
03*	20	-20 to +230
04	5.3	-30 to +1000
05	1.08	-50 to +5000

PROM 7600 (PROM ID may be seen in response to status command) uses integration times which reject noise from 50 Hz AC sources. Ranges 1, 2, and 3 have integration times of 3, 2, and 1 periods of a 50 Hz signal.

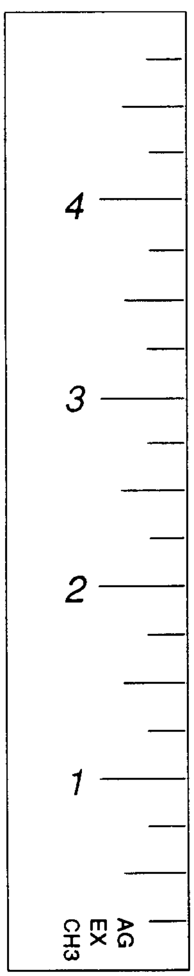
LEVEL SENSOR



LEVEL SENSOR



LEVEL SENSOR



LEVEL SENSOR