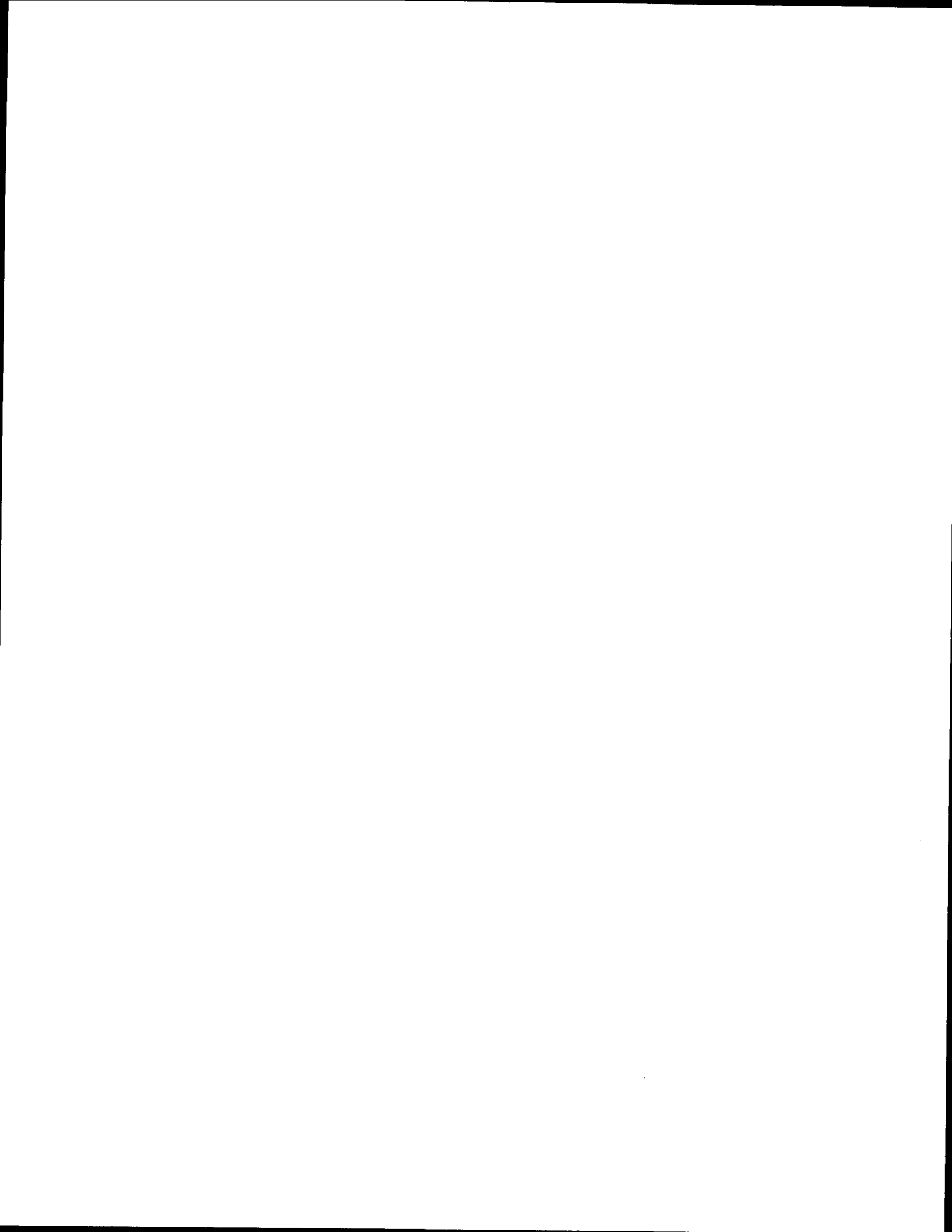


BDR301 BASIC DATA RECORDER  
OPERATOR'S MANUAL

REVISION: 10/94

COPYRIGHT (c) 1989, 1994, CAMPBELL SCIENTIFIC, INC.



## WARRANTY AND ASSISTANCE

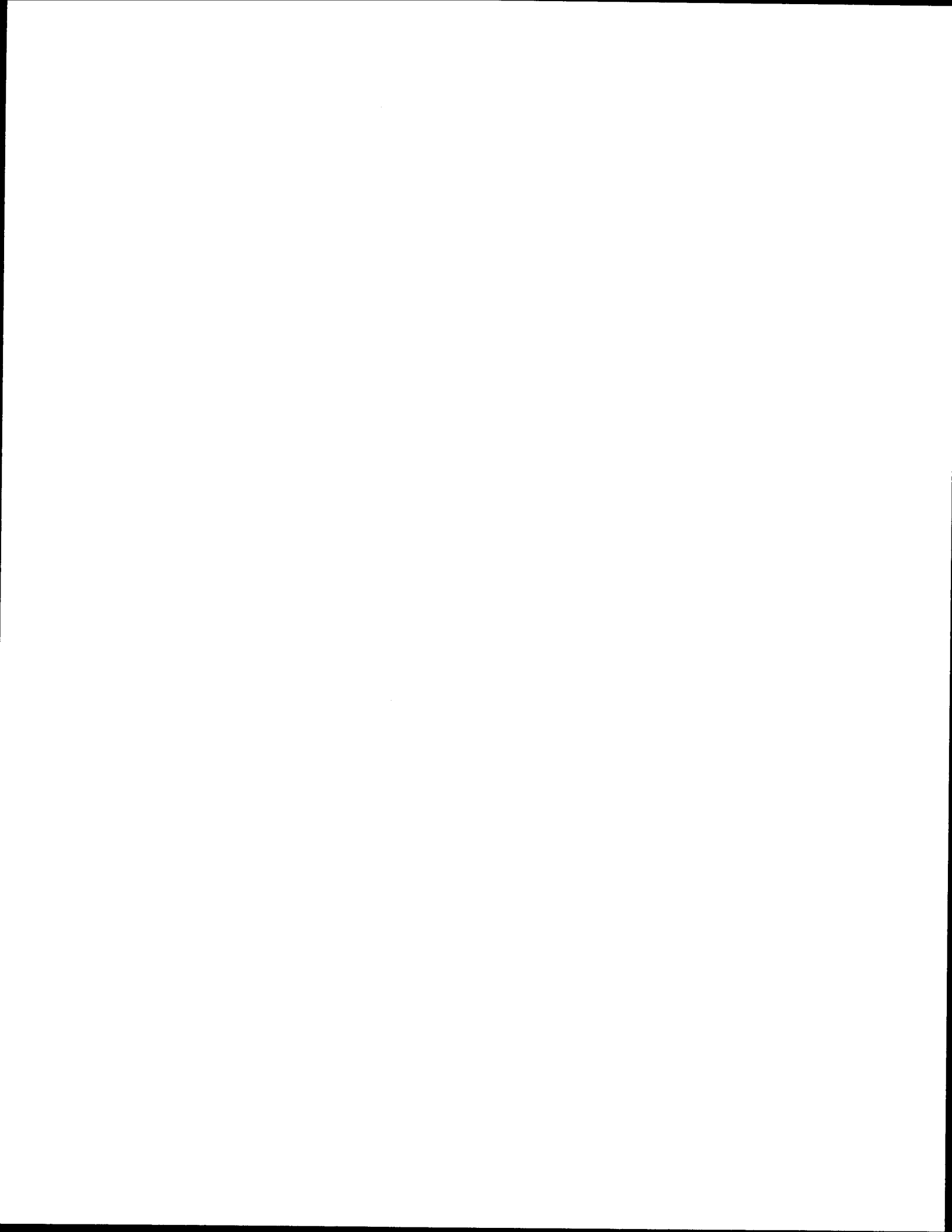
The **BDR301 BASIC DATA RECORDER** is warranted against defects in materials and workmanship. This warranty applies for five years from date of shipment. We will repair or replace products which prove to be defective during the warranty period provided they are returned prepaid to CAMPBELL SCIENTIFIC, INC. CAMPBELL SCIENTIFIC, INC. will return warranted equipment by surface carrier prepaid. No other warranty is expressed or implied. CAMPBELL SCIENTIFIC, INC. is not liable for 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 application 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, UT 84321

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.



# BDR301 BASIC DATA RECORDER

## TABLE OF CONTENTS

	PAGE
WARRANTY . . . . .	i
SELECTED OPERATING DETAILS . . . . .	v
<b>1. OVERVIEW</b>	
1.1 Measurement and Control Capability . . . . .	1-3
1.2 Programming . . . . .	1-3
1.3 Memory . . . . .	1-4
1.4 Computer Operations . . . . .	1-5
1.5 Specifications . . . . .	1-6
<b>2. DESCRIPTION OF CONNECTORS</b>	
2.1 Analog Connector . . . . .	2-1
2.2 RS232 Connector . . . . .	2-2
2.3 SDI-12 Connector . . . . .	2-3
<b>3. DISPLAY BUTTON OPERATIONS</b>	
3.1 Inputs . . . . .	3-1
3.2 Data . . . . .	3-2
3.3 Status . . . . .	3-3
<b>4. COMPUTER COMMANDS</b>	
4.1 Command Basics . . . . .	4-1
4.2 Help Command . . . . .	4-2
4.3 Status . . . . .	4-3
4.4 Setting and Displaying the Clock - C Command . . . . .	4-3
4.5 Data Retrieval . . . . .	4-4
4.6 Error Log . . . . .	4-5
4.7 Time Set Log . . . . .	4-7
4.8 Data Note Storage and Retrieval . . . . .	4-7
4.9 Display Inputs . . . . .	4-7
4.10 Adjust Offset . . . . .	4-8
4.11 Program Datalogger, Save, Load, and Delete Programs . . . . .	4-8
4.12 Security . . . . .	4-9
4.13 Exit . . . . .	4-9
4.14 Summary of Commands . . . . .	4-10
<b>5. PROMPT PROGRAMMING</b>	
5.1 Prompt Programming Overview . . . . .	5-1
5.2 Input Tables . . . . .	5-2
5.3 Data Storage Tables . . . . .	5-10
5.4 Editing . . . . .	5-13

**6. MAINTENANCE AND INSTALLATION**

6.1	Power Supplies . . . . .	6-1
6.2	Protection from the Environment . . . . .	6-3
6.3	Grounding . . . . .	6-3
6.4	Use of the Digital Control Port for Switching Relays . . . . .	6-4
6.5	Maintenance . . . . .	6-5

**7. DIRECT PROGRAMMING**

7.1	*1, *A, and *C Modes . . . . .	7-1
7.2	Program Instruction Types . . . . .	7-4
7.3	Parameter Data Types . . . . .	7-5
7.4	Repetitions . . . . .	7-5
7.5	Entering Negative Numbers . . . . .	7-5
7.6	Indexing Input Locations . . . . .	7-6
7.7	Input Range and Overrange Detection . . . . .	7-6
7.8	Data Storage and Output Processing . . . . .	7-7
7.9	Use of Flags: Program Control . . . . .	7-8
7.10	END, Instruction 95 . . . . .	7-11
7.11	Error Codes . . . . .	7-11
7.12	Program Syntax and Download Rules . . . . .	7-12

**8. BDR INSTRUCTION SET**

8.1	Analog Measurement Instructions . . . . .	8-1
8.2	Processing Instructions . . . . .	8-7
8.3	Output Processing Instructions . . . . .	8-20
8.4	Program Control Instructions . . . . .	8-27

**APPENDIX**

**A. MODEL 103 TEMPERATURE PROBE**

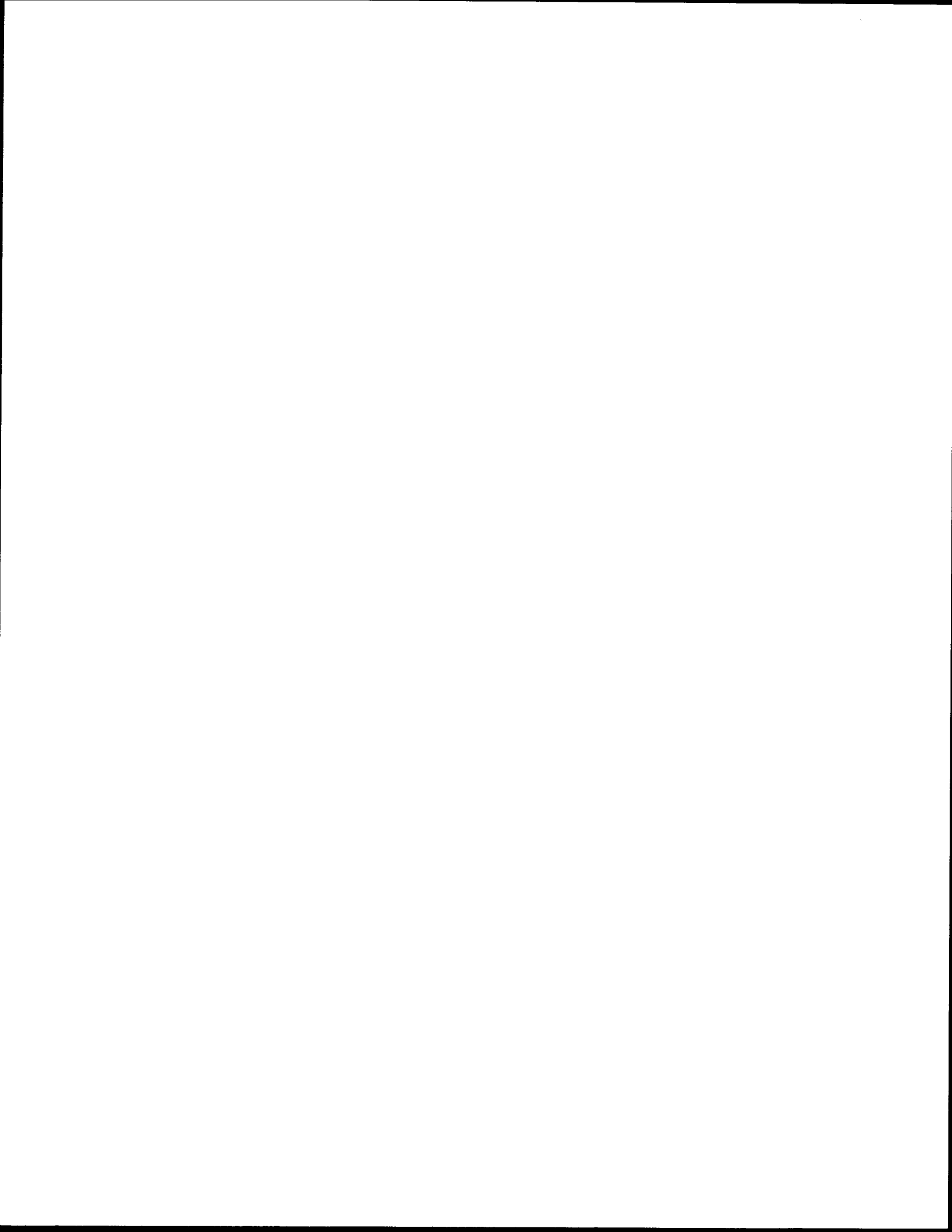
A.1	General . . . . .	A-1
A.2	Accuracy . . . . .	A-1
A.3	Hook-up Notes . . . . .	A-3
A.4	Conversion of 107 Probe to 103 Probe . . . . .	A-3

## TABLES

2.1	Analog Connector Pin Description . . . . .	2-1
2.2	RS232 Connector Pin Description . . . . .	2-3
2.3	SDI-12 Connector Pin Description . . . . .	2-3
3.3	Description of Status Information . . . . .	3-4
4.14	Commands . . . . .	4-10
7.1.2	Description of *A Mode Data . . . . .	7-3
7.1.3	*C Mode Entries . . . . .	7-4
7.7	Input Voltage Ranges and Codes . . . . .	7-6
7.9-1	Flag Description . . . . .	7-8
7.9-2	Command Codes . . . . .	7-8
7.11	Error Codes . . . . .	7-12
7.12	Example Program Listing . . . . .	7-13
8.1-1	Input Voltage Ranges and Codes . . . . .	8-1
8.4-1	Command Codes . . . . .	8-27
8.4-2	Flag Description . . . . .	8-27
8.4-3	Comparison Codes . . . . .	8-31

## FIGURES

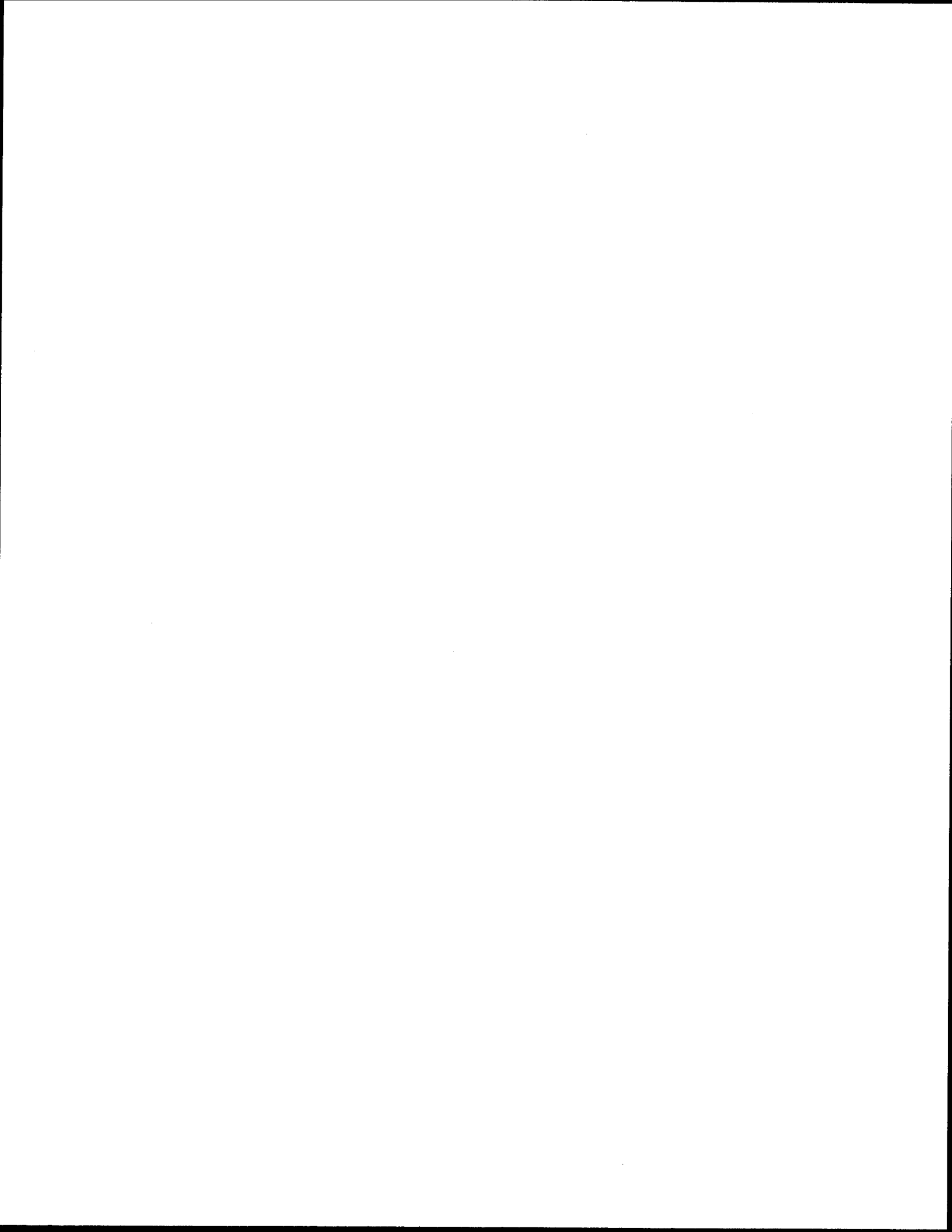
1-1	Perspective View of BDR301 . . . . .	1-1
1-2	System Block Diagram . . . . .	1-2
2.1	Analog Connector Pin Assignments . . . . .	2-1
2.2	RS232 Connector Pin Assignments . . . . .	2-2
2.3	SDI-12 Connector Pin Assignments . . . . .	2-3
6.1	Internal Battery Installation . . . . .	6-2
6.4-1	Relay Driver Circuit with Relay . . . . .	6-5
6.4-2	Power Switching Without Relay . . . . .	6-5
7.9-1	If Then/Else Execution Sequence . . . . .	7-9
7.9-2	Logical AND Construction . . . . .	7-9
7.9-3	Logical OR Construction . . . . .	7-10
A-1	Linearization Error . . . . .	A-2
A-2	103 Probe Schematic and Connections . . . . .	A-3
A-3	Cutting 107 Probe . . . . .	A-3
A-4	Placement of 20K Resistor for 103 Measurement . . . . .	A-4





## SELECTED OPERATING DETAILS

1. Instructions for installing the internal supply are found in Section 6.1.1.
2. Changing an existing datalogger program causes the datalogger to recompile the program. COMPILING ERASES ALL DATA. Always retrieve the existing data prior to making program changes.
3. To aid in data interpretation, retrieve the Error Log and Time Set Log when retrieving data.
4. The minimum and maximum external power supply voltage is 9 and 18 volts, respectively. A supply voltage in excess of 18 volts will damage the datalogger.
5. Damage will occur to the analog input circuitry if voltages in excess of  $\pm 16$  volts are applied for sustained period. The maximum analog input voltage is  $\pm 5$  volts with respect to datalogger ground. Signals in excess of  $\pm 5.5$  volts will cause errors and possible overranging on other analog input channels.
6. Analog measurements are affected if the pulse signal on Pulse Counter 2 is greater than 5.6 volts for longer than 100 milliseconds.
7. External power must be supplied when SDI-12 sensors are used.



## SECTION 1. OVERVIEW

The BDR301 Basic Data Recorder is a fully programmable datalogger/controller in a rugged, sealed enclosure.

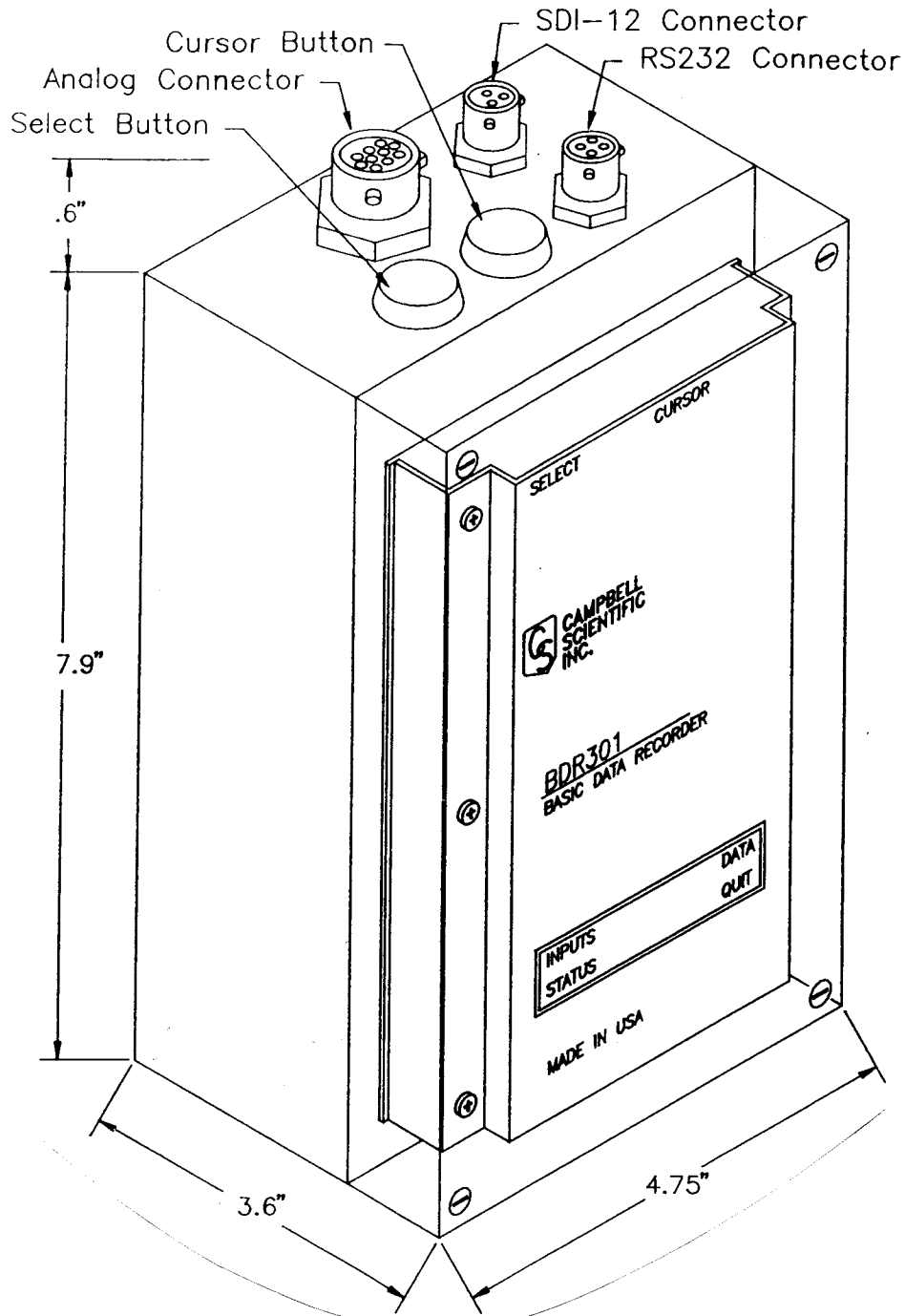


FIGURE 1-1. Perspective View of BDR301

## SECTION 1. OVERVIEW

All connections to the BDR are made through the three environmentally sealed connectors shown in Figure 1-1.

A typical BDR system is shown in Figure 1-2. In most cases, the computer is not connected to the system except during site visits.

The built in buttons and display may be used to verify that the BDR and sensors are operating correctly. The **select** button activates the BDR display and selects the desired category. The **cursor** button moves the cursor to the categories described below:

- INPUTS displays the most recent measurement values
- DATA displays stored data
- STATUS displays information specific to the datalogger and datalogger vital signs
- QUIT deactivates the display

A computer must be connected for any of the following operations:

- programming the BDR
- retrieving data
- storing and retrieving field notes
- setting the BDR clock.

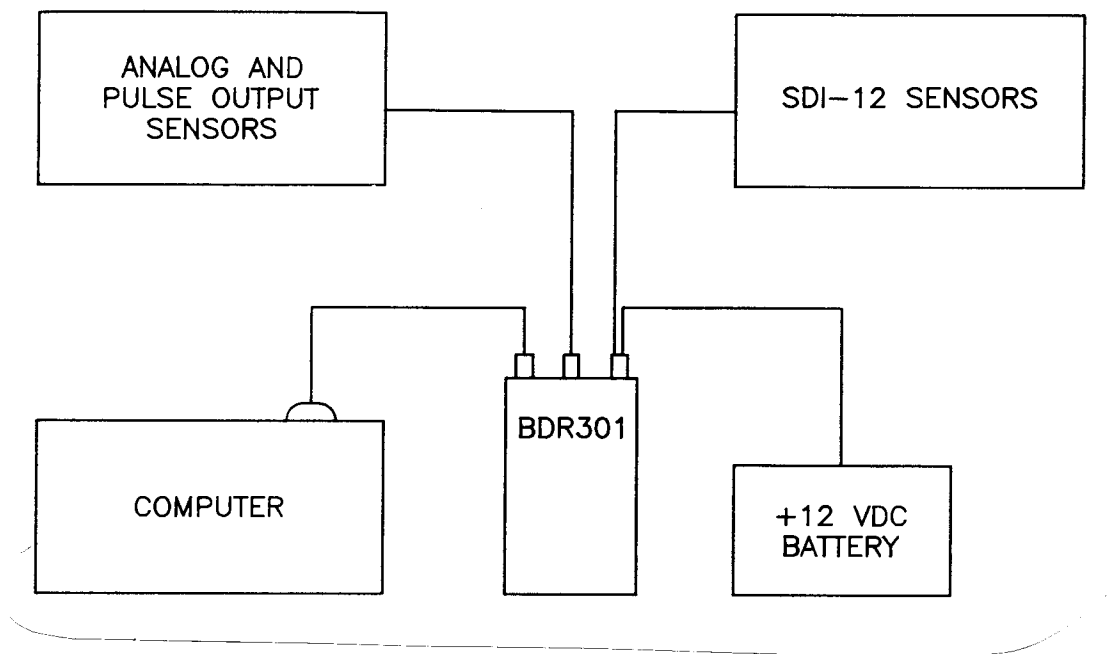


FIGURE 1-2. System Block Diagram

### 1.1 MEASUREMENT AND CONTROL CAPABILITY

The BDR is SDI-12 compatible. SDI sensors are measurement devices that include one or more transducers and a serial digital interface for communicating with a datalogger and transferring results. Up to 10 SDI sensors can be connected to and addressed by the datalogger.

The BDR301 has two differential input channels. Each differential input can be used as 2 single-ended channels. A calibrated excitation, approximately  $4000 \pm 50$  mV, is available for ratiometric bridge measurements (potentiometers, strain gage pressure transducers).

A fast (150 Hz) and slow (20 Hz) pulse count channel are available for measuring switch closure and voltage pulse type sensors. A typical application is a contact closure anemometer (fast) and a tipping bucket rain gage (slow).

Through programming, a control port may be set high (+5 VDC) or low (0 VDC) according to time or a measured event. The control port is typically used to activate an external device through a relay. A relay driver circuit is usually required due to the small current drive of the control port (1.5 mA at 3.5 volts).

### 1.2 PROGRAMMING

In a basic application the BDR measures the sensor(s) and stores the result for subsequent transfer to a computer. 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.

The measuring, processing, and storing tasks performed by the BDR are controlled by a user written program. The program may be developed in the BDR's prompting program generator ("prompt" programming) or in EDLOG ("direct" programming), a datalogger program development module contained in CSI's PC208 software package. EDLOG and direct programming imply the same programming method. Both terms refer to using the BDR instruction set rather than the prompting program generator.

All of the software required for prompt programming is in the datalogger. A terminal emulator program is required in the computer so the user can 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.

## SECTION 1. OVERVIEW

EDLOG requires an IBM-PC or compatible computer. The program is developed and stored for later transfer to the datalogger. This method uses an instruction set similar to other CSI dataloggers (CR10, 21X, CR7). Several instructions in series constitute a program. Instructions are selected and organized in series by the programmer to perform the required tasks.

When a program is entered or edited, the BDR301 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.**

### 1.3 MEMORY

For most applications, the BDR301 can store in excess of 30,000 datapoints, depending on the size of the program. A fixed amount of memory is distributed between the following four storage areas.

- **Final storage** is where final processed data are stored for transfer to the computer. This area is also the pool from which the remaining three areas take memory. The amount of memory in Final storage depends on the amount of memory used by Input storage, Intermediate storage and Program memory.
- **Input storage** is where measurement (input) results and the results of arithmetic and transcendental operations are stored. For example, the datalogger is programmed to measure a stage height and store the result in Input storage location 1. The stage value is then applied to an equation which calculates and stores discharge in location 2. 28 Input storage locations are available on power-up. Input storage locations can be expanded using the \*A Mode (refer to Section 7).
- **Intermediate storage** is an internal scratch pad used to calculate averages, maximums, minimums, totals, and standard deviations through time. Calculations performed and the amount of memory used in Intermediate memory is defined by the program. The user does not have access to Intermediate memory.
- **Program memory** is available for user programs entered via the "prompt" or direct programming methods.

## SECTION 1. OVERVIEW

On power-up, approximately 61700 bytes of memory are available. One data point consumes two bytes and a moderately large program is 1600 bytes, including Input and Intermediate storage. With a 1600 byte program, more than 30,000 data points are available for Final storage.

### 1.4 COMPUTER OPERATIONS

A computer is required for programming the BDR, retrieving data from Final storage, storing and retrieving field notes, and setting the BDR clock. The computer can also be used instead of the display buttons to view current measurements and display stored data and datalogger status. Computer operations are enabled by issuing commands from the computer. The command set is described in Section 4.

## 1.5 BDR301 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:

60 Hz Rejection PROM:

Full Scale Range mV	Integration ms	Resolution $\mu$ V
Auto range		
-20 to +5000	$\leq 16.7$	3.1 to 49.5
Fixed range		
-20 to +80	50.0	1.0
-20 to +120	33.3	1.5
-20 to +250	16.7	3.1
-30 to +1000	5.3	10.2
-50 to +5000	1.08	49.5

50 Hz Rejection PROM:

Full Scale Range mV	Integration ms	Resolution $\mu$ V
Auto range		
-20 to +5000	$\leq 20.0$	2.6 to 49.5
Fixed range		
-20 to +60	60.0	.9
-20 to +100	40.0	1.3
-20 to +230	20.0	2.6
-30 to +1000	5.3	10.2
-50 to +5000	1.08	49.5

INPUT NOISE VOLTAGE (on -20 to +80 mV range): 50 Hz: 1.5  $\mu$ V RMS,  
60 Hz: 1.76  $\mu$ V RMS

COMMON MODE RANGE:  $\pm 5$  volts

INPUT CURRENT: 10 nanoamperes

INPUT RESISTANCE: 6 gigohms

**ANALOG OUTPUTS**

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

*NOTE: The precise value of the excitation voltage is measured during the BDR 301 calibration. Bridge measurements are ratiometric with an accuracy of 0.02% of full scale.*

**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 ( $\mu$ s)	100	200.0
Min voltage pulse low time ( $\mu$ s)	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

**DIGITAL CONTROL OUTPUTS**

A single digital control output; enabled according to programmed time or event.

OUTPUT VOLTAGE (no load):  
High, 5 V  $\pm 0.1$  V  
Low,  $<0.1$  V

OUTPUT RESISTANCE: 1000 ohms

**RS-232 PORT**

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

TYPE: RS-232C

BAUD: 300, 1200, 9600

MODE: Full duplex, asynchronous

**DISPLAY**

The LCD display has a useful operating temperature range of 0 to 50°C. "SELECT" and "CURSOR" control buttons enable viewing of:  
-Instantaneous measurements  
-Stored records  
-Datalogger status

**TRANSIENT PROTECTION**

All input and output connections to the BDR301 are protected using RC filters and transzorbors or spark gaps.

**POWER**

POWER SUPPLY: 9 VDC minimum; 18 VDC maximum

CURRENT DRAIN: 200 microamps quiescent; 27 milliamps active.

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**

SIZE: 8.5 x 3.6 x 4.75 inches without mating connectors

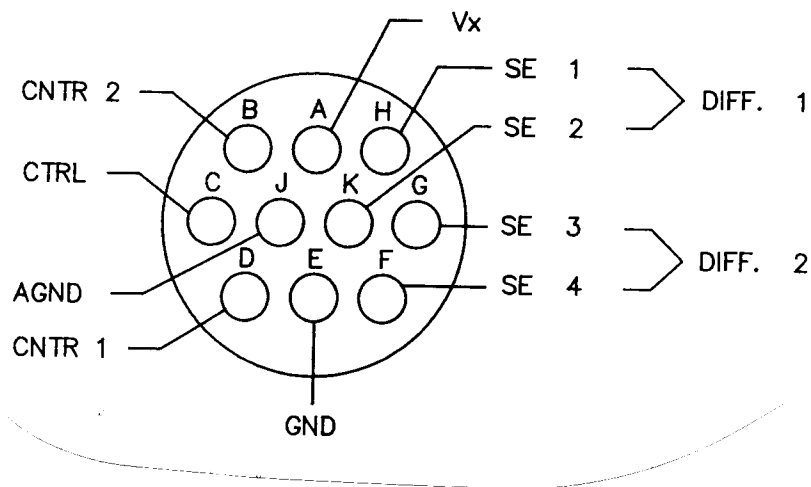
WEIGHT: 3.7 lbs. including batteries



## SECTION 2. DESCRIPTION OF CONNECTORS

All connections to the BDR are made through the three environmentally sealed connectors shown in Figure 1-1. The mating connectors come with the BDR; the user must build the required cables. This section provides the pin description of these connectors.

### 2.1 ANALOG CONNECTOR



**FIGURE 2.1. Analog Connector Pin Assignments**  
(looking into connector sockets)

Mating connector used on the cable is Mil. spec. MS 3116 F12-10P.

**TABLE 2.1. Analog Connector Pin Description**

<u>Pin</u>	<u>Name</u>	<u>Description</u>
A	Vx	4V ( $\pm 50$ mV) excitation switched on/off at beginning/end of Program Table execution. For use in resistive bridge measurements.
B	CNTR 2	pulse counter number 2, 150 hz maximum; input.
C	CTRL	Control port, 5V active, 0V inactive.
D	CNTR 1	pulse counter number 1, 20 Hz maximum; input.
E	GND	Ground for sensor shields and pulse signals.

## SECTION 2. DESCRIPTION OF CONNECTORS

TABLE 2.1 Cont.

F	SE 4	Single-ended analog channel 4, input.
G	SE 3	Single-ended analog channel 3, input.
H	SE 1	Single-ended analog channel 1, input.
J	AGND	Analog ground; reference for single-ended measurements and Vx return.
K	SE 2	Single-ended analog channel 2, input.

**NOTE:** SE 1 and 2 are the high and low sides of differential channel 1 (DIFF 1), respectively. SE 3 and 4 are the high and low sides of differential channel 2 (DIFF 2), respectively.

### 2.2 RS232 CONNECTOR

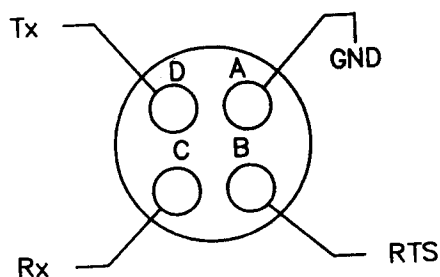


FIGURE 2.2. RS232 Connector Pin Assignments  
(looking into connector sockets)

Mating connector used on the cable is Mil. spec. MS 3116 F8-4P.

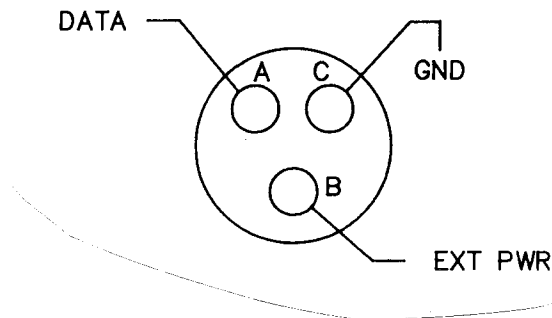
## SECTION 2. DESCRIPTION OF CONNECTORS

**TABLE 2.2. RS232 Connector Pin Description**

<u>Pin</u>	<u>Name</u>	<u>Description</u>
A	GND	Ground; tied to power ground (SDI-12 connector, pin C).
B	RTS	Request to send; must initially be brought high for the datalogger to respond to the computer.
C	Rx	Data receive; input.
D	Tx	Data transmit; output.

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

### 2.3 SDI-12 CONNECTOR

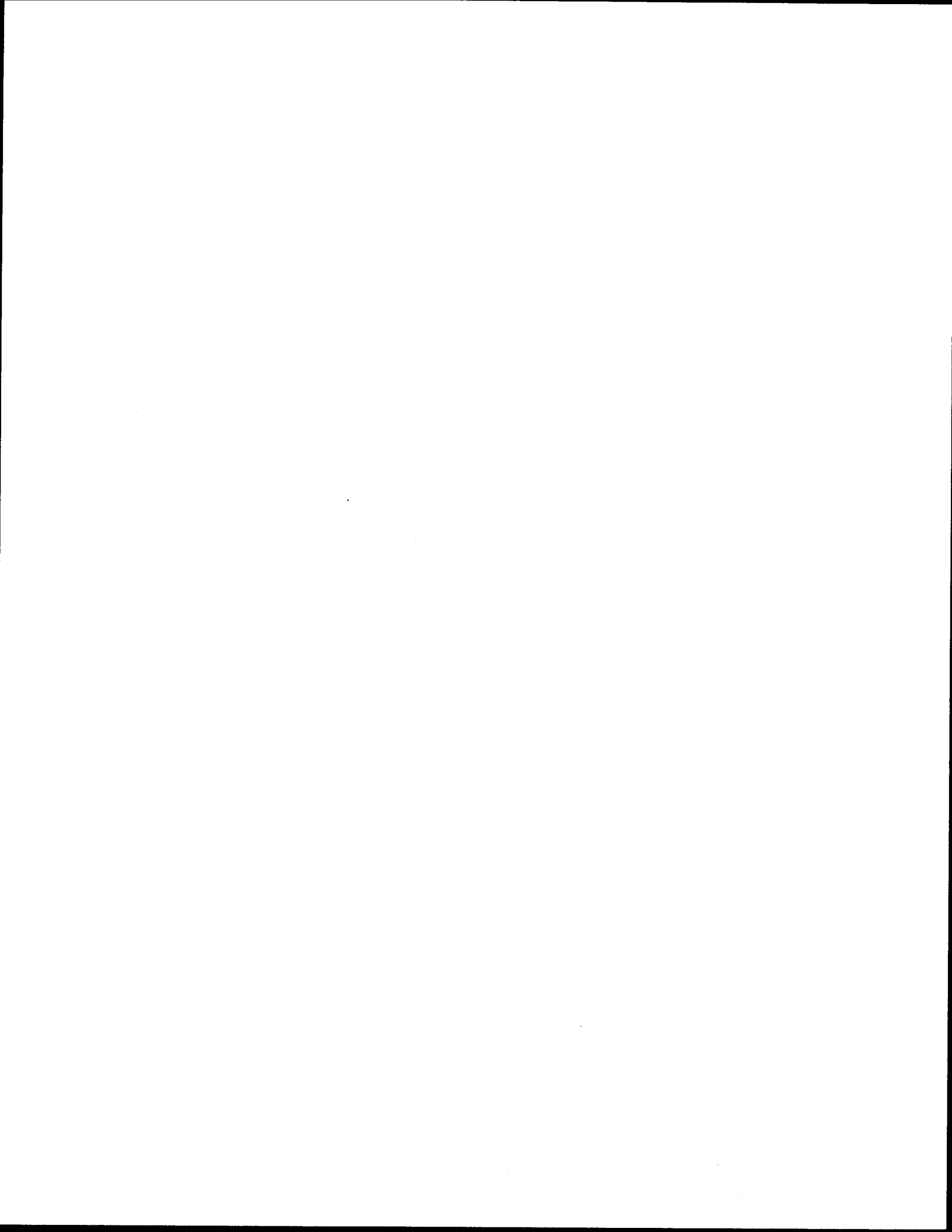


**FIGURE 2.3. SDI-12 Connector Pin Assignments**  
(looking into connector sockets)

Mating connector used on the cable is Mil. spec. MS 3116 F8-3P.

**TABLE 2.3. SDI-12 Connector Pin Description**

<u>Pin</u>	<u>Name</u>	<u>Description</u>
A	DATA	SDI-12 data line; input and output.
B	EXT PWR	External power; 9 to 18 VDC; primary power input. An external power supply must be present when SDI-12 sensors are used.
C	GND	Power ground.



### SECTION 3. DISPLAY BUTTON OPERATIONS

The select and cursor buttons (refer to Figure 1-1) are used to access and display input measurements, stored data, and datalogger status. If the display is off, press the Select button to activate the main menu shown below.

```
-----  
Inputs   Data  
Status   Quit  
-----
```

The cursor is initially located to the left of Inputs. Keying the cursor button advances the cursor to Data, Status, Quit and then back to Inputs. Press the select button when the cursor is next to the desired option. Selecting Quit turns the display off or after 250 seconds of no activity, the display turns off automatically.

#### 3.1 INPUTS

When "Inputs" is selected, programmed measurement instructions are executed at least every five seconds to update input locations. The initial display is the value in Input Location 1. The display format is as follows:

```
-----  
Loc:nnnn      A  
InpLbl+00.000Unt  
-----
```

Loc:nnnn = nnnn is the Input Location number being viewed

InpLbl = First six characters of the input label assigned to this location. If no input label is assigned in the program, this portion of the display is blank.

+00.000 = value stored in the Input Location.

Unt = 7th, 8th and 9th characters of the input label assigned to this location, usually units. If no units are assigned in the program, this portion of the display is blank.

A = Advance  
B = Backup  
Q = Quit

### SECTION 3. DISPLAY BUTTON OPERATIONS

Keying the cursor button changes A to B and Q. When the A is displayed and the select button is pressed, the display advances to the next Input Location. With the B displayed, pressing the select button backs up the display by one location. If Q is selected, the display returns to the main menu with the cursor next to Quit.

Hold the select button down to increase the speed with which the display advances or backs through Input Locations.

#### 3.2 DATA

Select "Data" to view final processed data in Final storage. The initial display is

```
-----  
Table: nnnn  
Int: nnnn min or s  
-----
```

showing header information for Data Table 1. Final storage consists of one or more Data Tables. If more than one recording interval is desired, more than one Data Table is created. For example, if hourly and daily recording intervals are desired, two Data Tables are created in the program.

"nnnn" is the Data Table number. The order in which the tables are created in the program is the order in which they are numbered and displayed. "Int" is the time interval between Data Table records. Press the cursor button to display header information of other Data Tables. If no Data Tables exist, Quit is displayed to the right of "Table:". Select Quit to return to the main menu.

Press the select button to view time tagged data stored in the selected table. The initial display shows the first data point of the most recent record. The format is shown below.

```
-----  
mmdd HH:MM:SS A  
InpLbl+00.00UntX  
-----
```

mmdd HH:MM:SS = month, day, hour-minute and second that the record is stored in the Table.

InpLbl = Input location label, same as for "Inputs".

+00.00 = data value stored.

Unt = Input location label, same as for "Inputs".

### SECTION 3. DISPLAY BUTTON OPERATIONS

X = Single letter code for the Output Processing Instruction applied to the data.  
S = sample                      A = average  
H = maximum (high)            T = total  
L = minimum (low)             D = standard deviation  
R = real time                  O = sample on max/min

A = Advance to next data point in record.  
B = Backup to previous data point in record.  
N = Advance to same data point location in next record.  
L = Backup to same data point location in previous (last) record.  
Q = Quit

Keying the cursor button changes A to B, N, L, and Q. When the A is displayed and the select button is pressed, the display advances to the next data point in the same record. With the B displayed, pressing the select button backs up the display by one data point. A and B do not cross record boundaries. If N is selected, the display advances to the same data point location in the next record. L backs up the display to the same data point location in the previous (last) record. Hold the select button down to rapidly advance or backup through Data Table locations.

If Q is selected, the display returns to the Data Table Menu and shows the following.

```
-----  
Table: Quit  
Int:  
-----
```

At this point, pressing the Cursor button brings up the next Data Table header. Pressing the Select button returns the display to the Main Menu.

#### 3.3 STATUS

Select Status to display information specific to the datalogger. The initial display is

```
-----  
mmdd HH:MM:SS A  
Year:nn  
-----
```

mmdd HH:MM:SS = Current date and time (month, day, hour-minute and second).

nn = two digit representation of the current year.

### SECTION 3. DISPLAY BUTTON OPERATIONS

A = Advance  
B = Backup  
Q = Quit

Keying the cursor button changes A to B and Q and back to A. When the A is displayed and the select button is pressed, the display advances to the next Status line. With the B displayed, pressing the select button backs up the display to the previous Status line. If Q is selected, the display returns to the Main Menu.

Hold the select button down to rapidly advance or backup through Status lines.

Current day and time are always shown on the first line of the display while in Status. The second display line shows Status information one line at a time (refer to Table 3.3).

**TABLE 3.3. Description of 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
SN:+xxxx.x	= datalogger serial number
Cxxxx	= Checksum



## SECTION 4. COMPUTER COMMANDS

### 4.1 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 B on the RS232 connector, Figure 2.2) 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 4.6).

#### GENERAL RULES GOVERNING COMMANDS

1. \* from datalogger means "ready for command".
2. CR to datalogger means "execute".
3. CRLF 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

## SECTION 4. COMPUTER COMMANDS

- 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.

### 4.2 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
7H     Direct programming
nL     Unlock Security with code n
12X    Transparent to SDI-12 sensor
1986T  (Repeat) Power reset
```

## SECTION 4. COMPUTER COMMANDS

Commands not shown in the HELP list are 2718Q, used to program the BDR301 from a computer file, and 2718,1Q, used to save a BDR301 program to a computer file.

### 4.3 STATUS

A                    *Status*

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
SN:+xxxx.x	= datalogger serial number
Cxxxx	= Checksum

### 4.4 SETTING AND DISPLAYING THE CLOCK - C COMMAND

	C	<i>display time</i>
	hr:min:secC	<i>set time</i>
mo/day/yr	hr:min:secC	<i>set date and time</i>

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 4.7) 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.

## 4.5 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

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.

## SECTION 4. COMPUTER COMMANDS

The following examples show the response to requesting the last 3 records of an hourly stage height Data Table, with and without time tags. Notice 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
```

### 4.6 ERROR LOG

```
1U           Send error log
1111U       Erase error log
```

A log of how many run time errors have occurred and the first and most recent times of occurrences 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 7.11)

## 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

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.

## 4.7 TIME SET LOG

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

Entering 2U returns a log of the last 8 times the datalogger clock was set (Section 4.4, C command). 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

## 4.8 DATA NOTE STORAGE AND RETRIEVAL

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

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

All Data Notes may be retrieved using the 1R command or, with the R command only those Notes entered since the last R command may be retrieved.

## 4.9 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 5 seconds, the n,mI command causes Input/Output Instructions 1, 2, 5, 6, 7, 11, and 29 to be executed. If the table execution interval is less than 5 seconds the values displayed are the result of the normal table execution.

## 4.10 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 gauge. The Y command allows the measurement offset to be edited without entering a programming mode (7W 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 escape 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.

## 4.11 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* will send Radio Shack "up line" and "cls".

*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**



## SECTION 4. COMPUTER COMMANDS

The 7W command prompt programming method is for basic applications requiring measurement, output processing and data storing instructions (refer to Section 5). 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 (EDLOG), must not be edited in the prompt programming mode (7W command).

### 4.12 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 maximum four character code number, *n*. Sending *nL* unlocks that level of security using *n* as the code number.

### 4.13 EXIT

*E*                    *Exit telecommunications.*

Sending **E** and **CR** terminates communication with the datalogger.

## 4.14 SUMMARY OF COMMANDS

TABLE 4.14. Commands

<u>Command</u>	<u>Description</u>
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 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.
<b><u>Display Inputs/Enter Values</u></b>	
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 n.

## SECTION 4. COMPUTER COMMANDS

### 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

### Program Datalogger

7W Enter Prompt Programming Mode (see Section 5), ANSI standard characters are sent for "up line" and "clear screen". 9W will send Radio Shack "up line" and "cls". 1W lists the program generated in the 7W 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, Ctrl C quits.

### Security

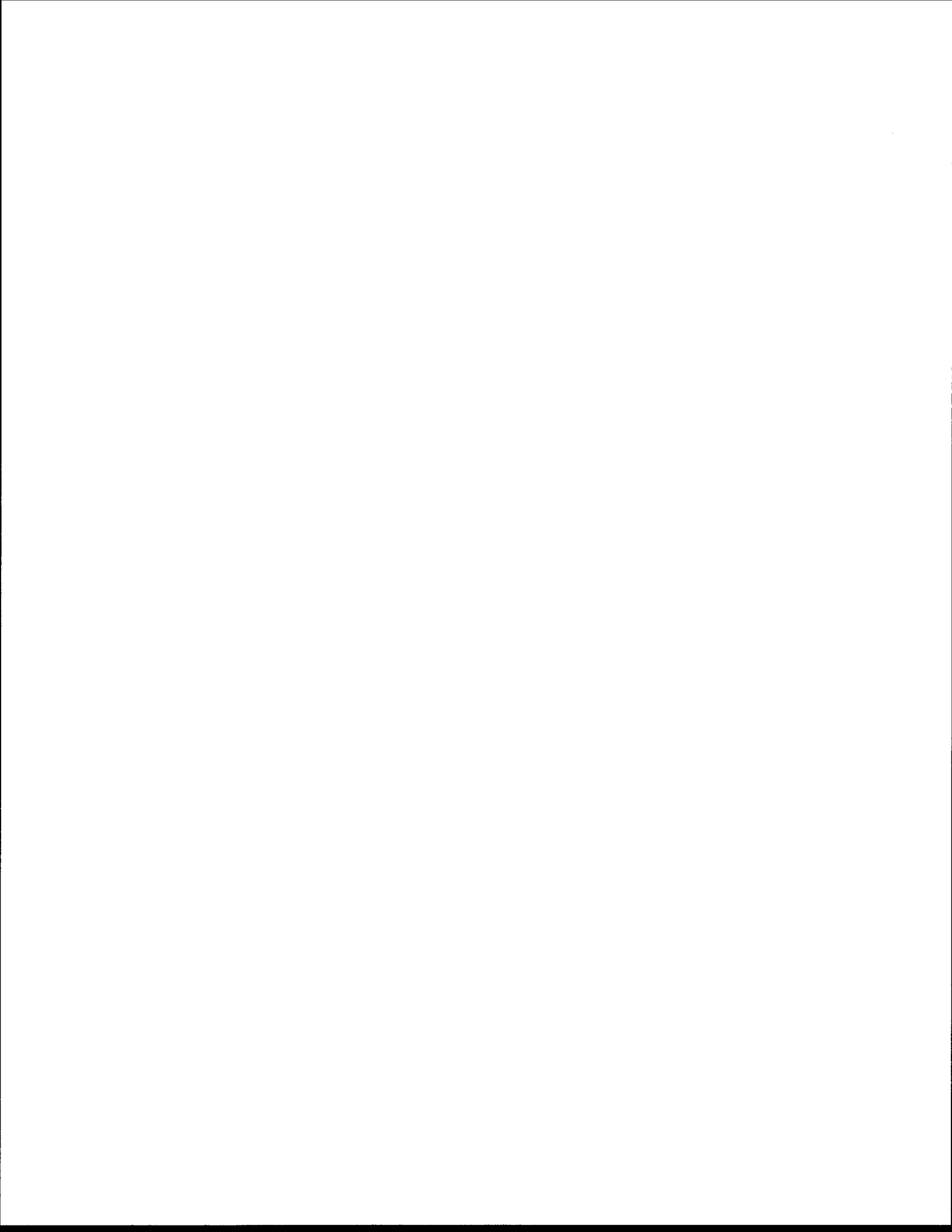
nL Unlock Security (\*C Mode) with code word n. The L command temporarily changes the security level. After terminating communications, security is reset.

### 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



## SECTION 5. PROMPT PROGRAMMING

Prompt programming simplifies programming requirements for routine BDR301 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 7.

**CAUTION:** The two methods of programming, i.e., prompt programming and direct programming (Sections 7 and 8) can not be mixed. A program generated in the direct method can not be edited in the prompting version. The reverse is also true.

### 5.1 PROMPT PROGRAMMING OVERVIEW

The prompt programming main menu, shown below, is displayed by issuing the 7W command (see note on 7W command entry below).

```
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 what 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.

## SECTION 5. PROMPT PROGRAMMING

**ESC** exits the 7W mode, and returns operation to the terminal emulator command state. If a program was entered or edited, the datalogger will compile the program and **erase all stored data**.

### **NOTE: 7W COMMAND ENTRY**

The sequence of characters to enable "clear screen" and "cursor up" on the computer is not standard in all computers. When the 7W Mode is entered using:

- the "7W" command, the sequence of characters for clear screen and cursor up follow standard ANSI convention.
- the "9W" command, the BDR returns clear screen and cursor up sequences compatible with Radio Shack and Tandy computers.
- "1W" lists the program generated in prompt programming.

## **5.2 INPUT TABLES**

All inputs to the datalogger are programmed through the Input Tables option. Datalogger instructions are entered defining what measurements are made on what channels, 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 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 a 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.

### **5.2.1 TABLE NUMBER AND MEASUREMENT INTERVAL - HEADER**

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 change tables, enter the other table number.

**Measurement Interval**

The measurement interval is the time between measurements programmed in the Input Table. The default interval is 5 minutes. 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 1 second.

**5.2.2 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 the control port 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
```

**Ctrl Port prior turn on (s)**

This parameter specifies the number of seconds or minutes before the start of the measurement interval to set the control port high. For example, the dissolved oxygen sensor in the USGS Mini-Monitor must be activated 1 minute before measurements to allow it to equilibrate. The interval must be less than the control interval. Note that units for the "Ctrl Port prior turn on" will change to remain the same as the "Measurement Interval" units. The control port is automatically set low after program execution. Refer to Section 6.4 for relay driver information.

## SECTION 5. PROMPT PROGRAMMING

### 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 on a surging river.

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" can not 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. 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 1 second, and the actual duration is 11 seconds.

**Example #2** - Assume the duration is 2 seconds and the number of samples is 7. Duration/samples is 0.28 seconds. The actual interval is 0.2 seconds and the actual duration is 1.4 seconds.

### 5.2.3 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. Section 3.2 describes how labels are used to identify data on the datalogger's display.

**units**

Up to three characters may be used to describe measurement units. For example, "FT" may be the unit of the stage measurement. Section 3.2 describes how labels are used to identify data on the datalogger's display.

**Type**

"Type" refers to the type of sensor to be measured or type of measurement to be made. Strike the space bar to display 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. Contact Campbell Scientific's Water Resources Group for assistance in selecting an SDI-12 sensor.

Some SDI sensors output more than one data point per output. For example, the Hydrolab SCOUT may be programmed to output 9 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 type must be entered four times, once for each parameter. Refer to the SDI channel ("Chn") description below for further information.

**NOTE:** An external power supply must be used to power SDI-12 sensors (refer to Figure 2.3).

## SECTION 5. PROMPT PROGRAMMING

- POT**            *Potentiometer*  
                  *Input Range = -90 to 5000 mV*  
                  *Units of measurement = Ratiometric,  $V_s/V_x$*   
                                   *$V_s$  = signal voltage;  $V_x$  = excitation voltage*  
The most likely sensor to be used with this instruction is the 10 turn potentiometer commonly used by the USGS. POT is a half bridge measurement which is also used to measure linear thermistors or wind vanes. Single ended analog channels 1 through 4 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 =  $^{\circ}C$*   
This instruction measures the datalogger temperature in degrees Centigrade. None of the analog channels are used in this measurement.
- THERM**            *103, 103B Thermistor*  
                  *Units of measurement =  $^{\circ}C$*   
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. Refer to Appendix A for probe details.
- 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 1 or 2. 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 anemometer or a tipping bucket rain gage. Channel 2 requires less power to operate than channel 1.

## SECTION 5. PROMPT PROGRAMMING

**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 and 2. 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 and 2. A strain gage pressure transducer is the most likely sensor to be used with this instruction.

### **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, SE, and HALF. Differential channels 1 and 2 are accessed by measurement types DIFF and FULL, and the pulse count channels by 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 whole number between 0 and 9.
- 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 a 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

## SECTION 5. PROMPT PROGRAMMING

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 (V) to store 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

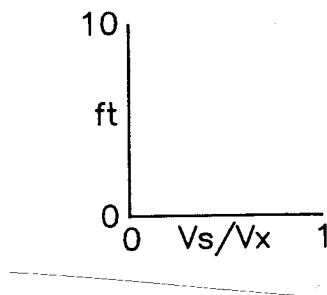
**NOTE:** The above example illustrates the use of "V", the value parameter. The order of output from the Scout is temperature, conductivity, dissolved oxygen, and pH but the order of storage in the datalogger is temperature, dissolved oxygen, pH, and conductivity.

### 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 1 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  $V_s/V_x$ .



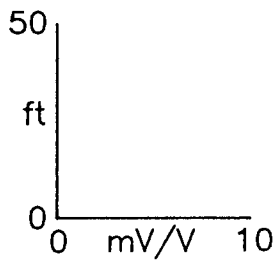
The multiplier,  $M$ , is  $(10-0)/(1-0)$ , or 10. The program for this example is shown below.

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

## SECTION 5. PROMPT PROGRAMMING

### Multiplier Example #2:

The sensor is 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 ( $1000 \cdot V_s / V_x$ ).  $V_s$  and  $V_x$  are in millivolts, making  $1000 \cdot V_s / V_x$  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 (refer to Section 4.10). 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 7W 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,I" command. Calculate the offset as follows:

$$\text{Offset} = \text{Reference} - \text{Measurement}$$

Re-enter the 7W mode, advance to the desired program line in the Input Table and enter the offset.

### 5.3 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.

#### 5.3.1 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. Available memory is the only constraint to the number of Data Tables which 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 1 second.

#### 5.3.2 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

## SECTION 5. PROMPT PROGRAMMING

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 POT          SMPL
```

Note that the offset is not entered in the Input Table. After exiting the 7W 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 POT          SMPL
01 01 POT          OPT?
```

## SECTION 5. PROMPT PROGRAMMING

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 POT	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.



### 5.3.3 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 7W Mode. For example, given Example Program #2, after exiting the 7W Mode the display reads

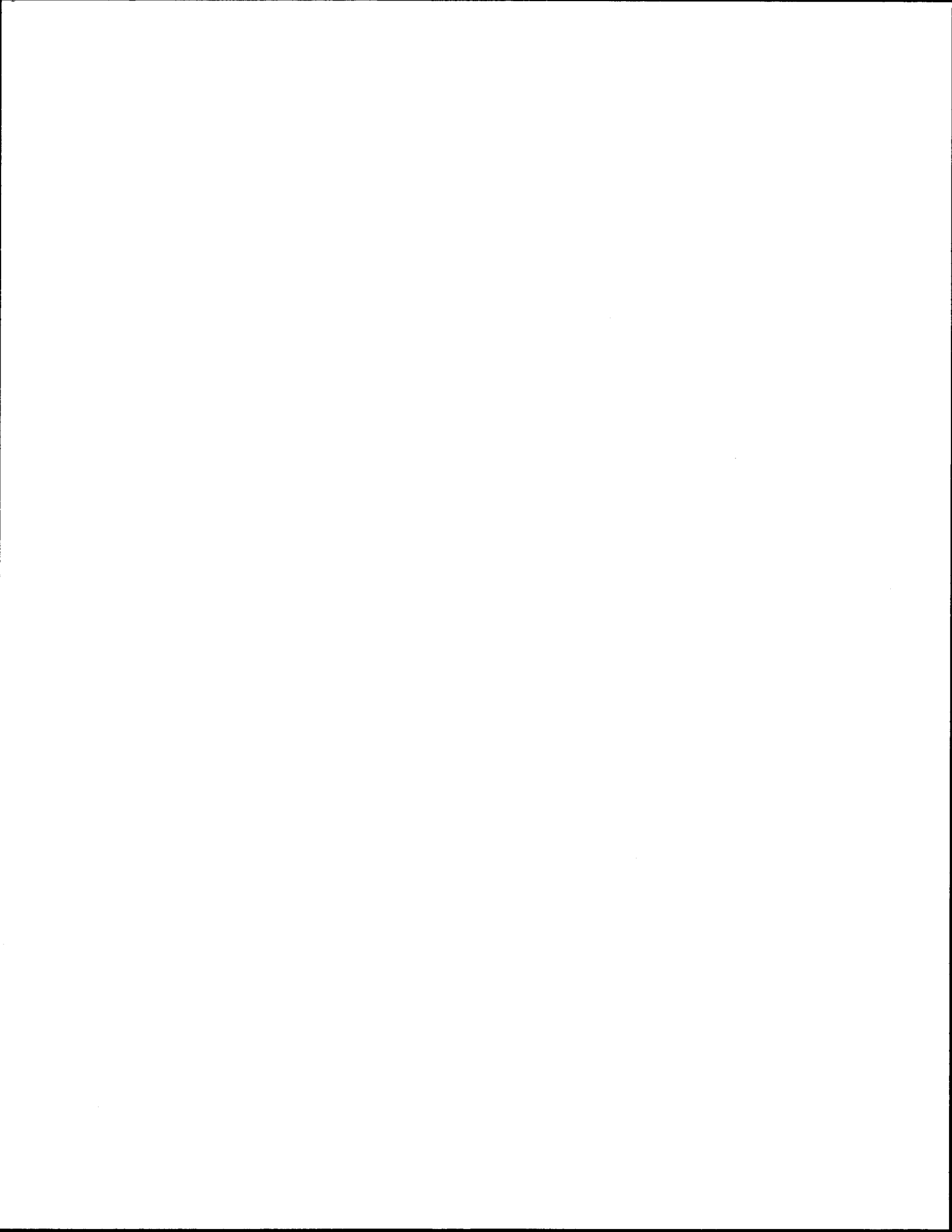
```
Storage filled in +304.80 days
Exiting programming mode
H = Help; errors:0000
```

### 5.4 EDITING

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

When using the ESC key to exit the 7W 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 7W Mode main menu
- select the correct option to go and fix the error **or** strike ESC to enter the terminal emulator command mode



## SECTION 6. MAINTENANCE AND INSTALLATION

### 6.1 POWER SUPPLIES

The BDR301 is equipped with internal batteries and a connector to go to a user supplied external battery (SDI-12 connector). Diodes in both power supply circuits provide reverse polarity protection and protection against battery to battery charging. The maximum power supply voltage is 18 volts. **Power supply voltages in excess of 18 VDC will damage the datalogger.**

The internal batteries are a backup to the external supply. The sealed enclosure contains desiccant to keep the electrical components dry, and should rarely be opened. For routine operation, use an external supply to delay the need of replacing the internal batteries.

The BDR301 typically draws 80 microamps in the quiescent state, 22 milliamps during an analog measurement, and 13 milliamps when communicating with a computer, an SDI-12 sensor, or when the display is active.

#### 6.1.1 INTERNAL POWER SUPPLY

The internal power supply is eight alkaline "C" cells. A fresh supply provides approximately 12 volts and 5 amp hours. The minimum internal supply voltage is 9.8 VDC, which appears as 8.6 when measured by the datalogger due to two diodes in the measurement circuit.

### INSTALLATION

Batteries are shipped with the datalogger but not installed. Battery installation should be done in a clean dry environment. When replacing batteries, retrieve all stored data before removing the old batteries. **All power, data, and programs are lost when changing batteries.**

To install the batteries, first unscrew the four corner screws and remove the enclosure lid. Disconnect the datalogger from the panel board as shown in Figure 6.1. Install the eight batteries in the battery holders located in the enclosure base. Note that the correct battery alignment is imprinted on the holders.

Reconnect the datalogger to the panel board in one smooth motion. If the motion is not smooth, the connection might make and break causing a power-up malfunction. If the connection is successful, "POWER UP TEST" is displayed while the datalogger checks its memory, PROM and EEPROM chips. **If "POWER UP TEST" is**

## SECTION 6. MAINTENANCE AND INSTALLATION

not displayed within 10 seconds, or if POWER UP message remains on longer than 30 seconds, disconnect the datalogger, wait 30 seconds and try again. Replace the enclosure lid.

### 6.1.2 EXTERNAL POWER SUPPLY

Connections to the user supplied external battery are made through the SDI-12 connector. If SDI-12 sensors are used an external power supply must be used (refer to Figure 2.3). The minimum and maximum external supply voltages are 9 and 18 VDC, respectively. Connections to the battery must be protected from the environment to prevent rusting, shorting, breaking, etc.

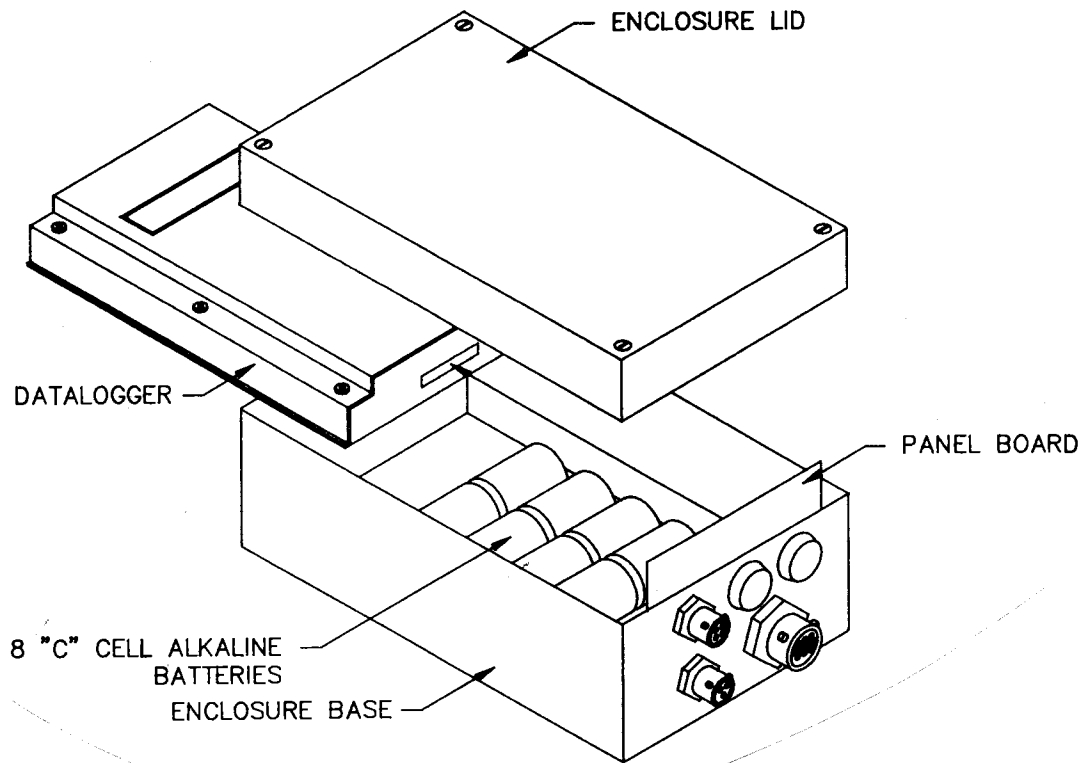


FIGURE 6.1. Internal Battery Installation

A rechargeable external battery may be float charged using a PS12 12V Charger if AC power is available. Assuming there is adequate exposure to the sun, the MSX10R Solar Panel may be used for recharging when AC power is not available.

## SECTION 6. MAINTENANCE AND INSTALLATION

### 6.1.3 LOW POWER SUPPLY

If the external supply drops below 9.0 VDC and the internal batteries drop below 9.8 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 5 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 the select button is pressed, 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.

### 6.2 PROTECTION FROM THE ENVIRONMENT

The normal environmental variables of concern are temperature and moisture. The standard BDR is production tested to operate from -35 to +55°C. For outdoor use the datalogger should be kept in the shade to minimize the temperature inside the enclosure.

Two half unit packets of DESI PAK desiccant are located by the internal batteries. This desiccant will keep the electronic components and junctions dry between battery changes. Desiccant packets can be dried out by placing them in an oven at 120 degrees C for 16 hours.

### 6.3 GROUNDING

The purpose of an earth ground is to minimize damage to the system by providing a low resistance path around the system to a point of low potential. CAMPBELL SCIENTIFIC RECOMMENDS THAT ALL DATALOGGERS DEPLOYED IN THE FIELD BE EARTH GROUNDED. It is the users responsibility to provide this earth ground. An earth ground is not necessarily required if the datalogger is powered by the internal batteries, and only analog or pulse output sensors with less than 50 feet of lead are connected.

All components of a system (datalogger, sensors, external power supplies, mounts, housings, etc.) should be referenced to one common earth ground. The easiest way to earth ground the BDR301 system is to connect the negative post of the external battery to earth ground. If no external battery exists, connect ground (pin C) of the SDI-12 connector to earth ground.

## SECTION 6. MAINTENANCE AND INSTALLATION

### EFFECT OF GROUNDING ON MEASUREMENTS: COMMON MODE RANGE

The common mode range is the voltage range, relative to the BDR ground, within which both inputs of a differential measurement must lie in order for the differential measurement to be made. Common mode range for the BDR is  $\pm 5V$ . 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. A similar differential 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.

### 6.4 USE OF THE DIGITAL CONTROL PORT FOR SWITCHING RELAYS

The digital control port can be set low or high (0V low, 5V high) using the port commands 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. Figure 6.4-1 shows a typical relay driver circuit in conjunction with a coil driven relay which may be used to switch external power to some device. In this example, when the control port is set high, 12V from the datalogger passes through the relay coil, closing the relay which completes the power circuit to a fan, turning the fan on. Campbell Scientific offers the A21REL-12 Four Channel Relay Driver for use with the BDR.

In other applications it may be desirable to switch power to a device without going through a relay. Figure 6.4-2 illustrates a circuit for switching external power to a device without going through a relay.

Other control port activated circuits are possible for applications with greater current/voltage demands than shown in Figures 6.4-1 and 2. For more information contact Campbell Scientific's Marketing Department.

## SECTION 6. MAINTENANCE AND INSTALLATION

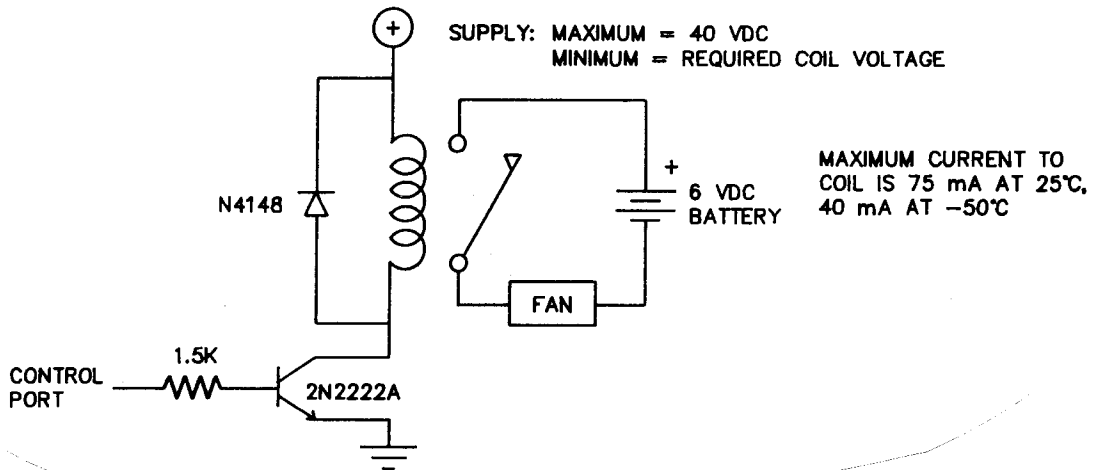


FIGURE 6.4-1. Relay Driver Circuit with Relay

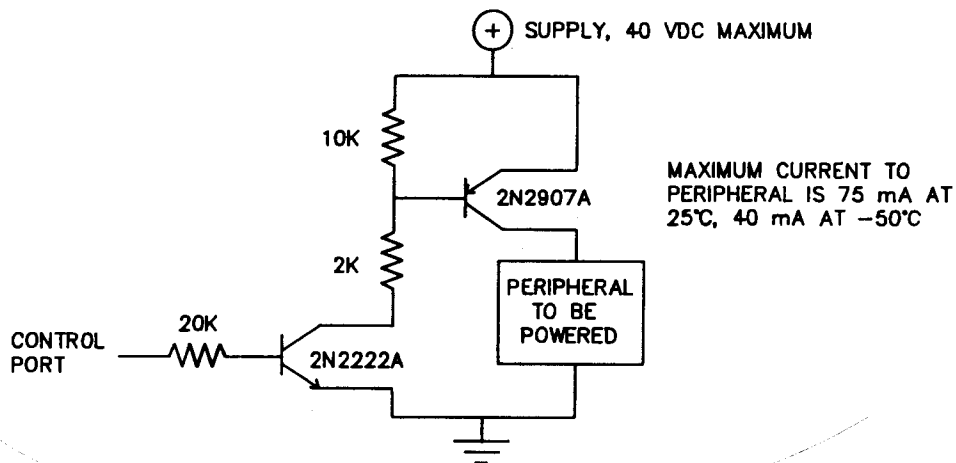
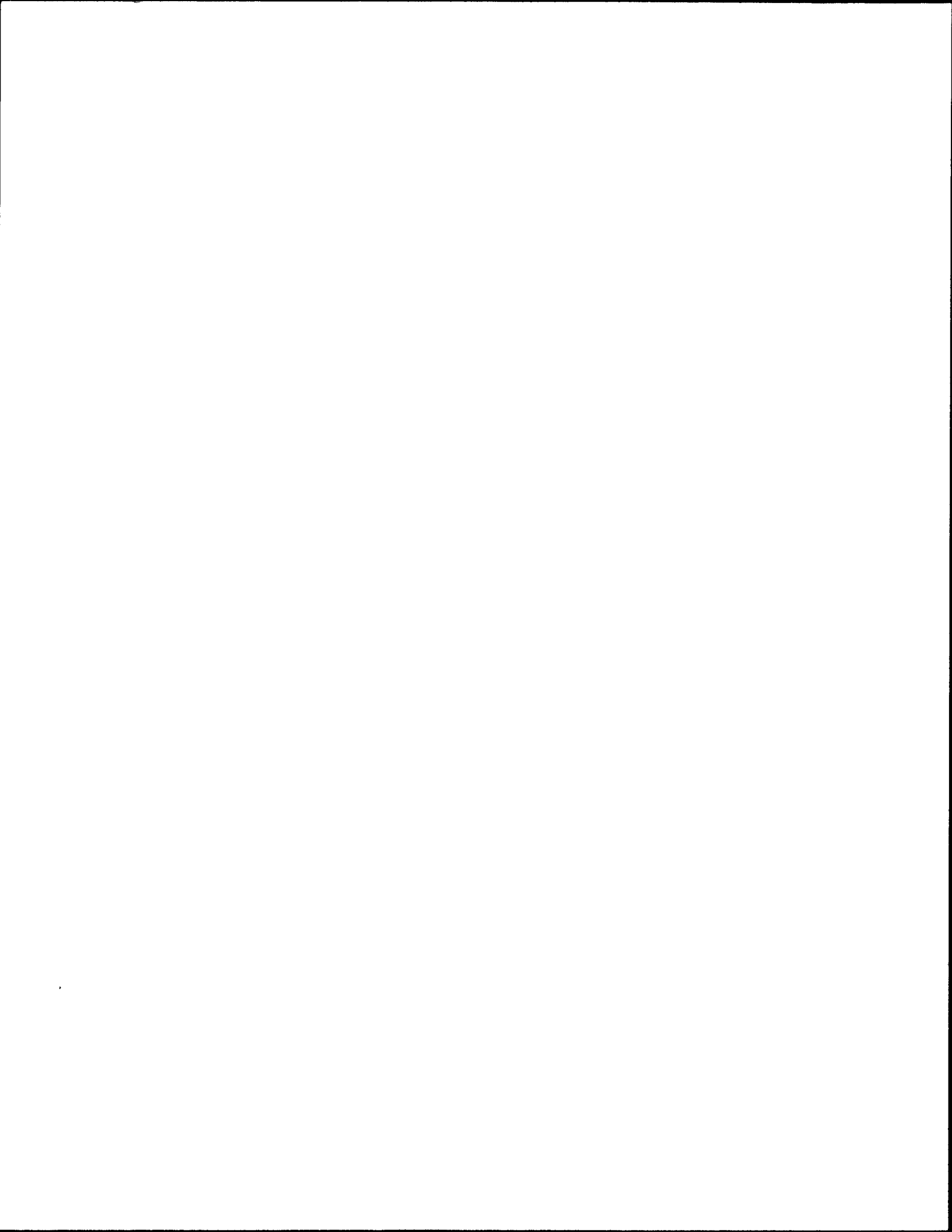


FIGURE 6.4-2. Power Switching Without Relay

### 6.5 MAINTENANCE

The BDR and power supplies require a minimum of routine maintenance. The internal batteries need to be replaced before they go below 9.8. The external battery should be replaced at 10.8 volts to avoid discharging the internal batteries. When not in use, remove the internal batteries to eliminate possible corrosion and store in a cool dry place.

Protection is required for the cables connected to the datalogger. Terminal junctions should be placed in a desiccated environment, and the desiccant should be replaced routinely.





## SECTION 7. 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, or in EDLOG, a software module for datalogger program development contained in Campbell's PC208 Datalogger Support Software. Campbell's CR10, 21X and CR7 dataloggers use direct programming.

### 7.1 \*1, \*A, and \*C MODES

The BDR has 3 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 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.**

#### 7.1.1 \*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).

## SECTION 7. DIRECT PROGRAMMING

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 below).

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).

If 0-- is entered for the Table Interval, table execution is defined by the time programmed in one or more Instruction 92's. Multiple table executions are effected with multiple Instruction 92's that have different time entries.

### - 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 ESCape.

## SECTION 7. DIRECT PROGRAMMING

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.**

### 7.1.2 \*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, 2) repartition this memory.

After repartitioning memory, the **BDR COMPILES, ERASING ALL STORED DATA.**

**TABLE 7.1.2. Description of \*A Mode Data**

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

### 7.1.3 \*C MODE -- SECURITY

The \*C Mode is accessed with the 7H computer command, followed by \*C. This mode is used to block access to the user's program information and certain BDR functions. There are 3 levels of security, each with its own 4 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.

## SECTION 7. DIRECT PROGRAMMING

**TABLE 7.1.3. \*C Mode Entries**

<b>SECURITY DISABLED</b>		
<u>Keyboard Entry</u>	<u>Display ID: Data</u>	<u>Description</u>
*C	01:XXXX	Level 1: Non-zero password blocks 7H and 7W commands, and Levels 2 and 3.
A	02:XXXX	Level 2: Non-zero password blocks C, I, 1W commands, and Level 3.
A	03:XXXX	Level 3: Non-zero password blocks all commands except A, L, N, H, U, and E.
<b>SECURITY ENABLED</b>		
<u>Keyboard Entry</u>	<u>Display ID: Data</u>	<u>Description</u>
*C	12:0000	Enter password. If correct, security is temporarily unlocked through that level.
A	01:XX	Level to which security has been disabled. 0 -- Password 1 entered (everything unlocked) 1 -- Password 2 entered 2 -- Password 3 entered

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

### 7.2 PROGRAM INSTRUCTION TYPES

The instructions used to program the BDR are divided into 4 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.

## SECTION 7. DIRECT PROGRAMMING

- Output Processing Instructions (numbers 70-82) 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.

### 7.3 PARAMETER DATA TYPES

There are 3 different data types used for Instruction parameters: Floating Point (FP), 4 digit integers (4), and 2 digit integers (2). 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  $\pm .00001$  to  $\pm 99999$ ), the internal format has a much greater range ( $1 \times 10^{-19}$  to  $9 \times 10^{18}$ ). Instruction 30 can be used to enter a number in scientific notation to be loaded into an input location.

### 7.4 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 2 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 2 times, enter it once with 2 repetitions. The instruction will make 2 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 2 measurements can be calculated by entering the Average Instruction with 2 repetitions.

### 7.5 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.

## SECTION 7. DIRECT PROGRAMMING

"--" is also used to the right of the Program Table Interval to indicate seconds and indexed Input Locations in a loop (Section 7.6). A Table Interval of 0-- means execute the table according to the time entered in one or more Instruction 92's.

### 7.6 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 1 after each pass through the loop. Instruction 90, Step Loop Index allows the increment step to be changed. See Instructions 87 and 90, Section 8, for more details.

### 7.7 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 7.7.).

**TABLE 7.7. Input Voltage Ranges and Codes**

<u>Range Code</u>	<u>Integration time, ms</u>	<u>Range, mV</u>
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

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.

## SECTION 7. DIRECT PROGRAMMING

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.

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  $\pm 5$  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.

### 7.8 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.

The number and size of Data Tables are limited only by memory.

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

## SECTION 7. DIRECT PROGRAMMING

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.

### 7.9 USE OF FLAGS: PROGRAM CONTROL

There are 9 flags which may be used in datalogger programs (Table 7.9-1). 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 7.9-1. Flag Description**

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

Flags are set with Program Control Instructions.

#### 7.9.1 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 7.9-2 lists these codes.

**TABLE 7.9-2. 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



## SECTION 7. DIRECT PROGRAMMING

```
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 7.9-3. 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).

### 7.9.3 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 1 second in order to nest loops. If the pass duration is greater than or equal to 1 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 7.11). Any loops or IF/THEN DO sequences started within a subroutine must end before the subroutine.

**7.9.4 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.

**7.10 END, INSTRUCTION 95**

The END instruction is required to mark the end of:

- A Subroutine (starts with Instruction 87)
- A Loop (starts with Instruction 85)
- 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 7.11).

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 7.9-2). The lines must not cross. To debug logic or find a missing or extra END error, list the program and draw the lines.

**7.11 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 12).

The codes and descriptions of compile and editor errors is given in Table 7.11. These errors appear on the computer screen at the time of illegal operations.

TABLE 7.11. Error Codes

<u>Code</u>	<u>Type</u>	<u>Description</u>
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

## 7.12 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 EDLOG, a program development software module contained in Campbell's PC208 Datalogger Support Software.

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 7.12 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.

SECTION 7. DIRECT PROGRAMMING

TABLE 7.12. 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
```

TABLE 7.12 Cont.

```

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 5 groups of 9 characters each, separated by colons (:). The first 6 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 other than a carriage return, line feed, characters in a comment line which start with a semicolon, or "}" (7D Hex). The order that the Modes are sent does not matter (i.e., the information for Mode 12 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 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 8. BDR INSTRUCTION SET

### 8.1 ANALOG MEASUREMENT INSTRUCTIONS

TABLE 8.1-1. Input Voltage Ranges and Codes

<u>Range Code</u>	<u>Integration time, ms</u>	<u>Range, mV</u>
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

**NOTE:** When a voltage input exceeds the range programmed, the value which is stored is set to the maximum negative number and displayed as -9999 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.

<u>PARAMETER NUMBER</u>	<u>DATA TYPE</u>	<u>DESCRIPTION</u>
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  $\pm 5$  volts of Analog Ground (Pin J, Analog Connector) to keep the signal in Common Mode Range. Output is in millivolts.

## SECTION 8.1 ANALOG MEASUREMENT INSTRUCTIONS

PARAMETER NUMBER	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 gage (pin D, analog connector). Channel 2 is for faster pulse inputs, such as a contact closure anemometer (pin B, analog connector). 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

## SECTION 8.1 ANALOG MEASUREMENT INSTRUCTIONS

PARAMETER NUMBER	DATA TYPE	DESCRIPTION
01:	2	Channel 01 = Pin D, Analog Connector 02 = Pin B, Analog Connector
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,  $V_s/V_x$ .

PARAMETER NUMBER	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,  $V_s/V_x$ .



## SECTION 8.1 ANALOG MEASUREMENT INSTRUCTIONS

PARAMETER NUMBER	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

### \*\*\*\* 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).

PARAMETER NUMBER	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.

PARAMETER NUMBER	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

## SECTION 8.1 ANALOG MEASUREMENT INSTRUCTIONS

### \*\*\*\* 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  $-200^{\circ}\text{C}$  to  $850^{\circ}\text{C}$ . The error in the linearization is less than  $0.001^{\circ}\text{C}$  between  $-100^{\circ}$  and  $+300^{\circ}$ , and is less than  $0.003^{\circ}\text{C}$  between  $-180^{\circ}$  and  $+830^{\circ}$ . The error (T calculated - T standard) is  $+0.006^{\circ}$  at  $-200^{\circ}$  and  $-0.006^{\circ}$  at  $+850^{\circ}$ . 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  $0^{\circ}\text{C}$ .

PARAMETER NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	4	Input location of $R_s/R_0$
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 degrees C of a thermistor on the datalogger analog board.

PARAMETER NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location number for temperature

INPUT LOCATIONS ALTERED            1

### \*\*\*\* 18 MOVE TIME TO INPUT LOCATION \*\*\*\*

#### FUNCTION

This Instruction takes the current time in seconds into the minute, minutes into the 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.

## SECTION 8.1 ANALOG MEASUREMENT INSTRUCTIONS

<u>Code</u>	<u>Time values</u>	<u>Maximum</u>
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

<u>PARAMETER NUMBER</u>	<u>DATA TYPE</u>	<u>DESCRIPTION</u>
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.

<u>PARAMETER NUMBER</u>	<u>DATA TYPE</u>	<u>DESCRIPTION</u>
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

## SECTION 8.2 PROCESSING INSTRUCTIONS

### 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  $1 \times 10^{-19}$  to  $9 \times 10^{18}$ . A value smaller than the minimum is set to 0, while a larger value is set to the maximum.

PARAMETER NUMBER	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.

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

INPUT LOCATIONS ALTERED 1

## SECTION 8.2 PROCESSING INSTRUCTIONS

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

FUNCTION

Add 1 to the value in the specified input location.

PARAMETER NUMBER	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.

PARAMETER NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Input location no. of source X	[X]
02:	4	Input location no. of source Y	[Y]
03:	4	Input location no. destination 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.

PARAMETER NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Input location no. of source X	[X]
02:	FP	Fixed addend	[F]
03:	4	Input location no. destination of X + F	[Z]
INPUT LOCATIONS ALTERED		1	

## SECTION 8.2 PROCESSING INSTRUCTIONS

\*\*\*\* 35 X - Y \*\*\*\*

FUNCTION

Subtract Y from X and place the result in an Input location.

PARAMETER NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Input location no. of source X	[X]
02:	4	Input location no. of source Y	[Y]
03:	4	Input location no. destination of X - Y	[Z]

INPUT LOCATIONS ALTERED            1

\*\*\*\* 36 X \* Y \*\*\*\*

FUNCTION

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

PARAMETER NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Input location no. of source X	[X]
02:	4	Input location no. of source Y	[Y]
03:	4	Input location no. destination of 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.

PARAMETER NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Input location no. of source X	[X]
02:	FP	Fixed multiplier	[F]
03:	4	Input location no. destination of X * F	[Z]

INPUT LOCATIONS ALTERED            1

## SECTION 8.2 PROCESSING INSTRUCTIONS

\*\*\*\* 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 ( $\pm 99999$ ).

PARAMETER NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Input location no. of source X	[X]
02:	4	Input location no. of source Y	[Y]
03:	4	Input location no. destination of 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.

PARAMETER NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Input location no. of source X	[X]
02:	4	Input location no. destination of square root of X	[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.

PARAMETER NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Input location no. of source X	[X]
02:	4	Input location no. destination of LN(X)	[Z]

INPUT LOCATIONS ALTERED            1

## SECTION 8.2 PROCESSING INSTRUCTIONS

\*\*\*\* 41 EXP(X) \*\*\*\*

FUNCTION

Raise the exponential (EXP) base e to the X power and place it in an input location.

PARAMETER NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Input location no. of source X	[X]
02:	4	Input location no. destination of 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.

PARAMETER NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Input location no. of source X	[X]
02:	4	Input location no. destination of 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.

PARAMETER NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Input location no. of source X	[X]
02:	4	Input location no. destination of ABS(X)	[Z]

INPUT LOCATIONS ALTERED            1



## SECTION 8.2 PROCESSING INSTRUCTIONS

### \*\*\*\* 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.

PARAMETER NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Input location no. of source X	[X]
02:	4	Input location no. destination of 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.

PARAMETER NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Input location no. of source X	[X]
02:	4	Input location no. destination of 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.

PARAMETER NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Input location no. of source X	[X]
02:	FP	Fixed divisor	[F]
03:	4	Input location no. destination of X MOD F	[Z]

INPUT LOCATIONS ALTERED            1

## SECTION 8.2 PROCESSING INSTRUCTIONS

\*\*\*\* 47 X<sup>Y</sup> \*\*\*\*

FUNCTION

Raise X to the Y power and place the result in an input location.

PARAMETER NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Input location no. of source X	[X]
02:	4	Input location no. of source Y	[Y]
03:	4	Input location no. destination of X <sup>Y</sup>	[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)).

PARAMETER NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Input location no. of source X	[X]
02:	4	Input location no. destination of 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.

## SECTION 8.2 PROCESSING INSTRUCTIONS

PARAMETER NUMBER	DATA TYPE	DESCRIPTION
01:	4	Swath [SWATH]
02:	4	Starting input location no. [1ST LOC]
03:	4	Input location no. destination of 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.

PARAMETER NUMBER	DATA TYPE	DESCRIPTION
01:	4	Swath [SWATH]
02:	4	Starting input location no. [1ST LOC]
03:	4	Input location no. destination of 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.

PARAMETER NUMBER	DATA TYPE	DESCRIPTION
01:	4	Swath [SWATH]
02:	4	Starting input location no. [1ST LOC]
03:	4	Input location no. destination of average [AVG or Z]
INPUT LOCATIONS ALTERED		1

## SECTION 8.2 PROCESSING INSTRUCTIONS

### \*\*\*\* 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.

PARAMETER NUMBER	DATA TYPE	DESCRIPTION	
01:	4	Starting input location no.	[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.

PARAMETER NUMBER	DATA TYPE	DESCRIPTION
01:	4	Number of values to move
02:	4	1st source location
03:	2	Step of source
04:	4	1st destination location
05:	2	Step of destination

INTERMEDIATE STORAGE            0

## SECTION 8.2 PROCESSING INSTRUCTIONS

### \*\*\*\* 55 5TH ORDER POLYNOMIAL \*\*\*\*

**FUNCTION**

Evaluate a 5th order polynomial of the form.

$$F(X) = C_0 + C_1X + C_2X^2 + C_3X^3 + C_4X^4 + C_5X^5$$

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 - 99999 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.

PARAMETER NUMBER	DATA TYPE	DESCRIPTION	
01:	2	Repetitions	[REPS]
02:	4	Starting input location no. for X	[X]
03:	4	Starting input location no. destination 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

$$SVPI = -.00486 + .85471 X + .2441 X^2$$

## SECTION 8.2 PROCESSING INSTRUCTIONS

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.

PARAMETER NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location no. for air temperature °C [TEMP.]
02:	4	Input location no. destination 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:

$$VP = VPW - A(1 + B \cdot TW)(TA - TW) P$$

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.

PARAMETER NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location no. of atmospheric pressure in kilopascals [PRESSURE]
02:	4	Input location no. of dry-bulb temp. [DB TEMP.]
03:	4	Input location no. of wet-bulb temp. [WB TEMP.]
04:	4	Input location no. destination of vapor pressure [VP or Z]

INPUT LOCATIONS ALTERED 1

## SECTION 8.2 PROCESSING INSTRUCTIONS

### \*\*\*\* 58 LOW PASS FILTER \*\*\*\*

**FUNCTION**

Apply a numerical approximation to an analog resistor capacitor (RC) low pass (LP) filter using the following algorithm.  $F(X_i) = W \cdot X_i + F(X_{i-1}) \cdot (1-W)$

Where  $X$  = input sample  
 $W$  = user entered weighting function,  $0 < W < 1$   
 If  $W=0$ ,  $F(X_i)=0$ ; 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  $W/(2 T)$ . For larger values of  $W$ , this "analog" estimate of the cutoff frequency becomes less representative.

PARAMETER NUMBER	DATA TYPE	DESCRIPTION	
01:	2	Repetitions	[REPS]
02:	4	Starting input location no. for input data	[X]
03:	4	Starting input location no. destination 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.

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

INPUT LOCATIONS ALTERED 1 FOR EACH REPETITION

## SECTION 8.2 PROCESSING INSTRUCTIONS

\*\*\*\* 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.

PARAMETER NUMBER	DATA TYPE	DESCRIPTION
01:	4	Source Input location
02:	4	Destination Input location

INPUT LOCATIONS ALTERED            1



## SECTION 8.3 OUTPUT PROCESSING INSTRUCTIONS

### 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.

\*\*\*\* 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.

PARAMETER NUMBER	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.

PARAMETER NUMBER	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.

PARAMETER NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	4	Starting input location no.

OUTPUTS GENERATED 1 FOR EACH INPUT LOCATION



## SECTION 8.3 OUTPUT PROCESSING INSTRUCTIONS

\*\*\*\* 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).

### SECTION 8.3 OUTPUT PROCESSING INSTRUCTIONS

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 deg. C	
Upper range limit	30 deg. 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).

PARAMETER NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	4	Number of bins
03:	2	Form code (0=open form, 1=closed form)
04:	4	Bin select value input location no.
05:	4	Weighted value input location no. (0 = frequency distribution option)
06:	FP	Lower limit of range
07:	FP	Upper limit of range

OUTPUTS GENERATED                      NUMBER OF BINS \* REPETITIONS

\*\*\*\* 76 WIND VECTOR \*\*\*\*

#### FUNCTION

Output one of several combinations of the following:

1. Mean wind speed
2. Mean wind vector magnitude
3. Mean wind vector direction
4. Standard deviation of direction

This Instruction will work with either polar (wind speed and direction) or orthogonal (fixed East and North props) sensors and accommodates multiple sets of sensors through the "repetitions" parameter. When used with polar sensors, the wind direction in degrees may be 0 to 360, 0 to 540, less than zero, or greater

## SECTION 8.3 OUTPUT PROCESSING INSTRUCTIONS

than 540. This instruction does a modulo divide by 360 which enables it to handle all ranges. 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 a -20 to 340 degree output.

Parameter 2 is used to select the type of sensors used and which values are output.

PARAMETER NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions of wind speed and direction
02:	2	Sensor/Output 2 digits: AB B Sensor type: 0 speed and direction 1 East and North A Output option: 0 All 4 outputs 1 Mean wind speed Mean wind vector direction Standard deviation of direction 2 Mean wind speed Mean wind vector direction 3 Mean wind vector direction Standard deviation of direction
03:	4	First wind speed input location no. (East wind speed)
04:	4	First wind direction input location no. (North wind speed)
OUTPUTS GENERATED		2-4 (depending on output option) FOR EACH REPETITION

\*\*\*\* 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

### SECTION 8.3 OUTPUT PROCESSING INSTRUCTIONS

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).

PARAMETER NUMBER	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:

$$S = (( X_i^2 - ( X_i )^2/N )/N)^{1/2}$$

where  $X_i$  is the  $i$ th measurement and  $N$  is the number of samples.

PARAMETER NUMBER	DATA TYPE	DESCRIPTION
01:	2	Repetitions
02:	4	Starting input location no.
OUTPUTS GENERATED		1 FOR EACH REPETITION

SECTION 8.4 PROGRAM CONTROL INSTRUCTIONS

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.

PARAMETER NUMBER	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

## SECTION 8.4 PROGRAM CONTROL INSTRUCTIONS

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.

PARAMETER NUMBER	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 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.



## SECTION 8.4 PROGRAM CONTROL INSTRUCTIONS

PARAMETER NUMBER	DATA TYPE	DESCRIPTION
01:	2	Subroutine number (1-9, 79-99)

\*\*\*\* 86 DO \*\*\*\*

### FUNCTION

This Instruction unconditionally executes the specified command.

PARAMETER NUMBER	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  $\geq$  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

## SECTION 8.4 PROGRAM CONTROL INSTRUCTIONS

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  $\leq$  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.

PARAMETER NUMBER	DATA TYPE	DESCRIPTION
01:	2	Pass duration in tenths of seconds
02:	4	number of passes

## SECTION 8.4 PROGRAM CONTROL INSTRUCTIONS

### \*\*\*\* 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.

PARAMETER NUMBER	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**

<u>Parameter 1</u>	<u>Function</u>
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.

PARAMETER NUMBER	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.

## SECTION 8.4 PROGRAM CONTROL INSTRUCTIONS

PARAMETER NUMBER	DATA TYPE	DESCRIPTION
01:	2	Increment for the loop index counter

\*\*\*\* 91 IF FLAG \*\*\*\*

### FUNCTION

This Instruction checks the status of one of the nine Flags and conditionally performs the specified Command.

The first Parameter specifies the condition to check:

- 1X Execute command if Flag X is high
- 2X Execute command if Flag X is low

PARAMETER NUMBER	DATA TYPE	DESCRIPTION
01:	2	Flag 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 be executed only once in the minute, not 6 times.)

## SECTION 8.4 PROGRAM CONTROL INSTRUCTIONS

PARAMETER NUMBER	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.

PARAMETER NUMBER	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.

## SECTION 8.4 PROGRAM CONTROL INSTRUCTIONS

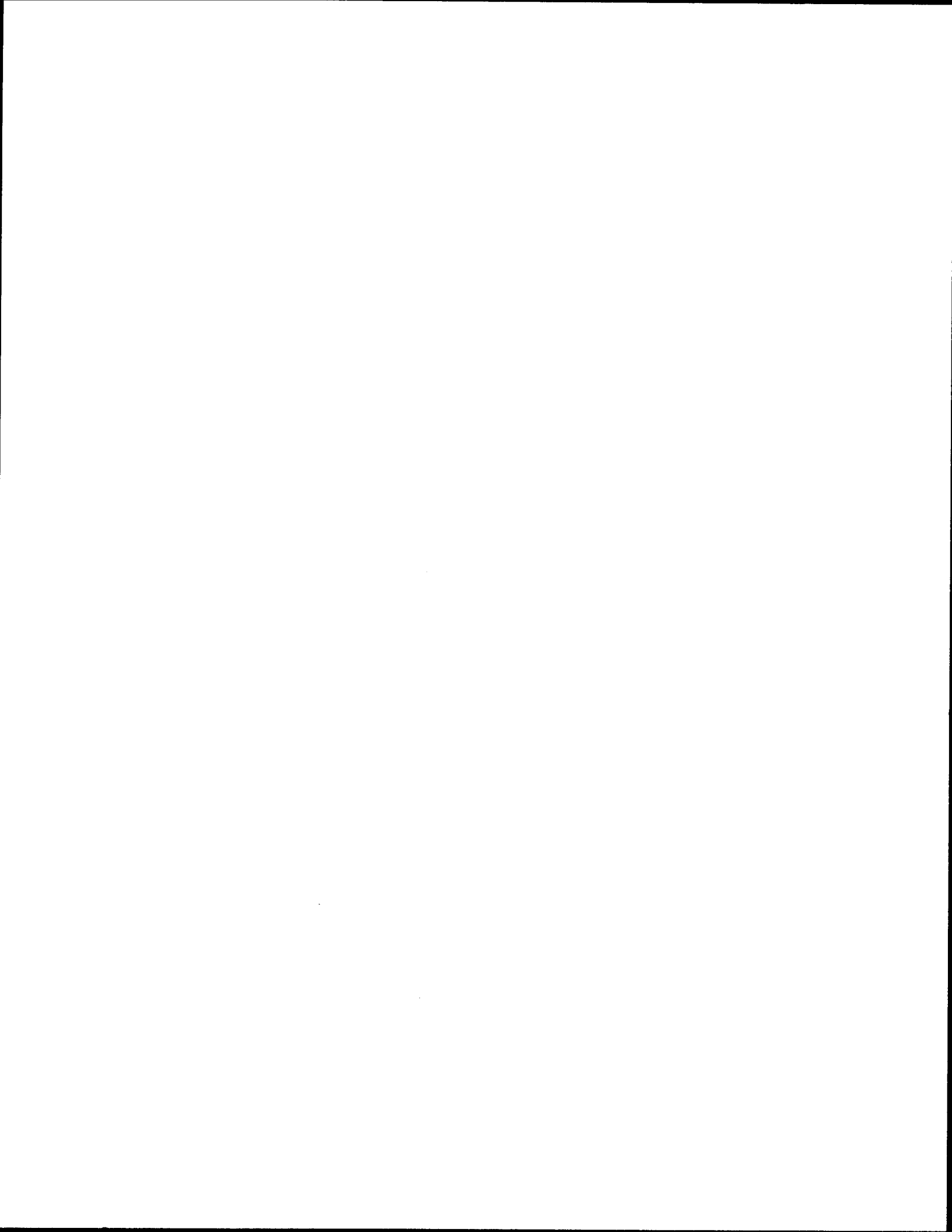
\*\*\*\* 95 END \*\*\*\*

### 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 \*\*\*\*

PARAMETER NUMBER	DATA TYPE	DESCRIPTION
01:	4	Input location to do nothing to



## APPENDIX A. MODEL 103 TEMPERATURE PROBE

### A.1 GENERAL

The 103 and 103B Thermistor Probes are for sensing air and soil temperatures, respectively, and are compatible with the BDR301 Basic Data Recorder. The probes incorporate the Fenwal Electronics UUT51J1 Thermistor and are identical except that the 103B is designed for burial. The standard lead length is 10 feet; other lead lengths are available on request.

To prevent direct solar radiation from heating the probe, causing high air temperature readings, the 103 probe should be mounted in the model 41301-5 6 Plate Radiation Shield or equivalent.

The 103 is read by Instruction 11, "THERM" in the Prompt Programming mode, which makes a ratiometric measurement and converts the result to °C through a seventh order polynomial.

### A.2 ACCURACY

The overall probe accuracy is a combination of Fenwal's interchangeability specification, the precision of the bridge resistors and the linearization error. In a "worst case" example all of the errors add in one direction to yield a  $\pm 0.4^{\circ}\text{C}$  accuracy over the range of  $-33^{\circ}\text{C}$  to  $+48^{\circ}\text{C}$ . The major error component is the  $\pm 0.2^{\circ}\text{C}$  (0 to  $60^{\circ}\text{C}$ ,  $\pm 0.5^{\circ}\text{C}$  at  $-40^{\circ}\text{C}$ ) interchangeability specification of the thermistor. The error is typically less than the specification and if desired can be reduced with a single point calibration. The interchangeability error is predominantly offset and can be determined with a single point calibration. The error can then be compensated for with the offset entered in the measurement instruction.

The bridge resistors are 0.1% tolerance with a 10ppm temperature coefficient.

The error in the linearization used to calculate temperature from the measurement error is shown in Figure A-1.



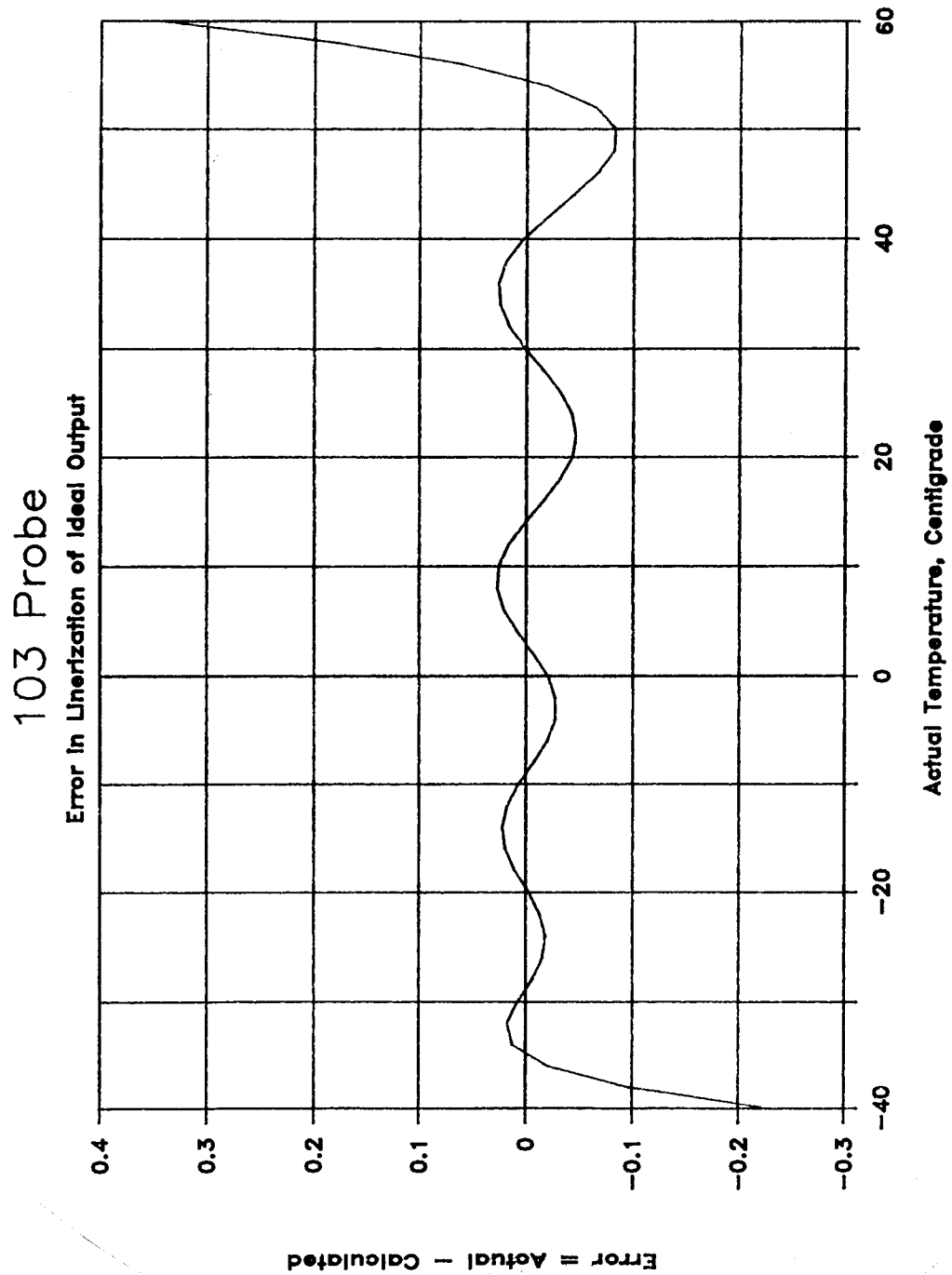


FIGURE A-1. Linearization Error

## APPENDIX A. MODEL 103 TEMPERATURE PROBE

### A.3 HOOK-UP NOTES

The 103 probe is connected to excitation, analog ground, ground, and one of the 4 single-ended input channels as shown in Figure A-2.

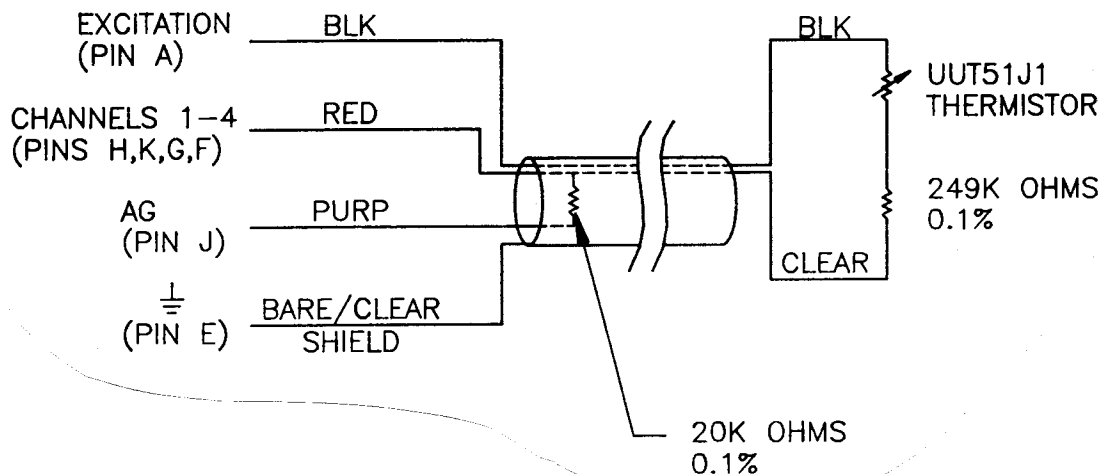


FIGURE A-2. 103 Probe Schematic and Connections

### A.4 CONVERSION OF 107 PROBE TO 103 PROBE

The Campbell Scientific 107 and 107B temperature probes used with the CR10, 21X, and CR7 dataloggers use the same thermistor as the 103 probe with a different "pick off" resistor. If there is not time to obtain a 103 probe and a 107 probe is on hand, it is possible to modify the 107 probe. The end of the probe with the wire connections must be cut off as shown in Figure A-3, and a 20 Kohm 0.1% resistor installed as close to the BDR301 connector as possible (Figure A-4).

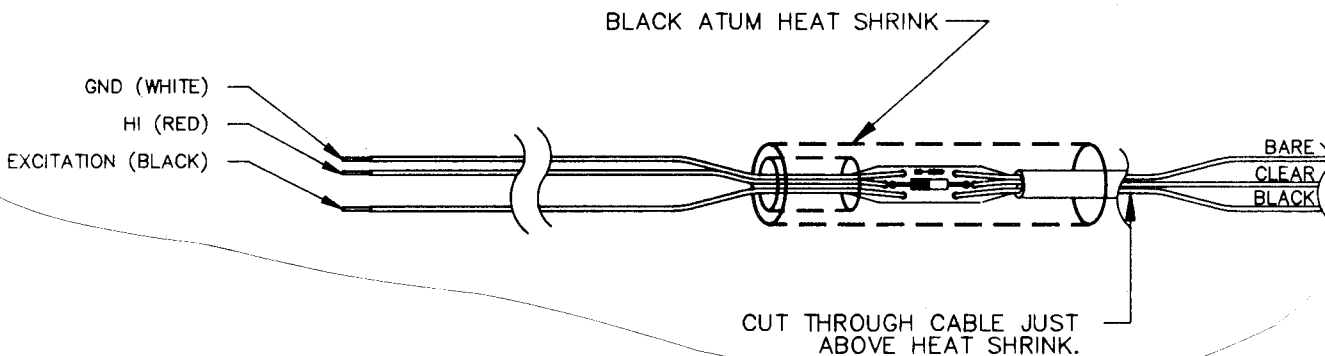


FIGURE A-3. Cutting 107 Probe

APPENDIX A. MODEL 103 TEMPERATURE PROBE

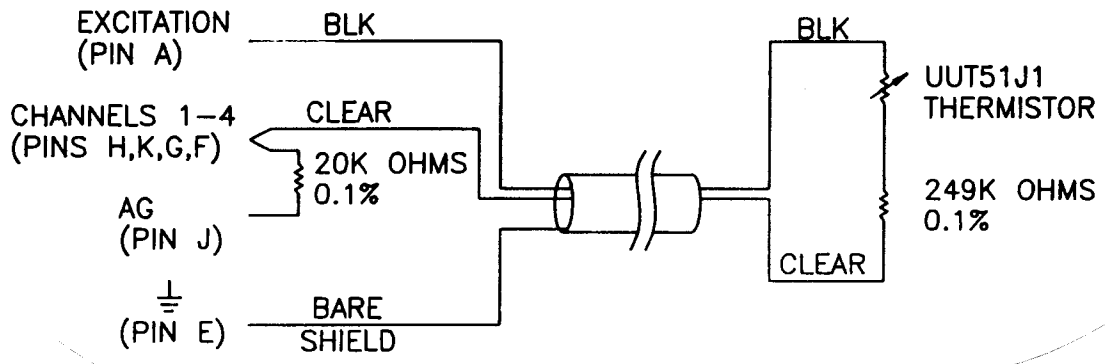


Figure A-4. Placement of 20K Resistor for 103 Measurement.