**≡Texas Transportation Institute**

---

# CR7 Measurement and Control System Instruction Manual

## Product 5-4518-01-P3

**TEXAS TRANSPORTATION INSTITUTE**
**THE TEXAS A&M UNIVERSITY SYSTEM**
**COLLEGE STATION, TEXAS**

**TEXAS DEPARTMENT OF TRANSPORTATION**

# CR7 MEASUREMENT AND CONTROL SYSTEM
## INSTRUCTION MANUAL

**REVISION: 7/97**

# WARRANTY AND ASSISTANCE

The **CR7 MEASUREMENT AND CONTROL SYSTEM** is warranted by CAMPBELL SCIENTIFIC, INC. to be free from defects in materials and workmanship under normal use and service for thirty-six (36) months from date of shipment unless specified otherwise. Batteries have no warranty. CAMPBELL SCIENTIFIC, INC.'s obligation under this warranty is limited to repairing or replacing (at CAMPBELL SCIENTIFIC, INC.'s option) defective products. The customer shall assume all costs of removing, reinstalling, and shipping defective products to CAMPBELL SCIENTIFIC, INC. CAMPBELL SCIENTIFIC, INC. will return such products by surface carrier prepaid. This warranty shall not apply to any CAMPBELL SCIENTIFIC, INC. products which have been subjected to modification, misuse, neglect, accidents of nature, or shipping damage. This warranty is in lieu of all other warranties, expressed or implied, including warranties of merchantability or fitness for a particular purpose. CAMPBELL SCIENTIFIC, INC. is not liable for special, indirect, incidental, or consequential damages.

Products may not be returned without prior authorization. To obtain a Returned Materials Authorization (RMA), contact CAMPBELL SCIENTIFIC, INC., phone (435) 753-2342. After an applications engineer determines the nature of the problem, an RMA number will be issued. Please write this number clearly on the outside of the shipping container. CAMPBELL SCIENTIFIC's shipping address is:

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

CAMPBELL SCIENTIFIC, INC. does not accept collect calls.

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

# CAMPBELL SCIENTIFIC, INC.

| | | |
|---|---|---|
| 815 W. 1800 N. | Campbell Scientific Canada Corp. | Campbell Scientific Ltd. |
| Logan, UT 84321-1784 | 11564 -149th Street | Campbell Park |
| USA | Edmonton, Alberta T5M 1W7 | 80 Hathern Road |
| Phone (435) 753-2342 | CANADA | Shepshed, Loughborough |
| FAX (435) 750-9540 | Phone (780) 454-2505 | LE12 9GX, U.K. |
| www.campbellsci.com | FAX (780) 454-2655 | Phone +44 (0) 1509 601141 |
| | | FAX +44 (0) 1509 601091 |

# CR7 OPERATOR'S MANUAL
# TABLE OF CONTENTS

## OVERVIEW

### OV1. PHYSICAL DESCRIPTION

### OV2. MEMORY AND PROGRAMMING CONCEPTS

### OV3. PROGRAMMING THE CR7

### OV4. PROGRAMMING EXAMPLE

### OV5. DATA RETRIEVAL OPTIONS

### OV6. SPECIFICATIONS

# PROGRAMMING

## 1. FUNCTIONAL MODES

## 2. INTERNAL DATA STORAGE

## 3. INSTRUCTION SET BASICS

# DATA RETRIEVAL/COMMUNICATION

## 4. EXTERNAL STORAGE PERIPHERALS

## 5. TELECOMMUNICATIONS

## 6. 9 PIN SERIAL INPUT/OUTPUT

# PROGRAMMING EXAMPLES

## 7. MEASUREMENT PROGRAMMING EXAMPLES

## 8. PROCESSING AND PROGRAM CONTROL EXAMPLES

# INSTRUCTIONS

# MEASUREMENTS

## 13. CR7 MEASUREMENTS

# INSTALLATION

## 14. INSTALLATION

## 15. I/O CARD ADDRESSING AND MULTIPLE I/0 MODULES

# APPENDICES

## A. GLOSSARY

## B. CR7 PROM SIGNATURES FOR SYSTEMS EQUIPPED WITH STANDARD SOFTWARE

## C. BINARY TELECOMMUNICATIONS

## D. CALIBRATION PROCEDURES

# SELECTED OPERATING DETAILS

The channel numbering on the Analog Input Card refers to differential measurements. Single ended measurements assume the HI and LO side of each differential channel are two independent single ended channels, e.g., the HI and LO side of differential channel 2 are single ended channels 3 and 4 respectively.

When multiple measurements are specified in one measurement instruction (through use of the "Repetitions Parameter") the CR7 I/O Module is capable of sequencing through 500 fast, single-ended measurements per second. This specification is the MEASUREMENT SPEED and should not be confused with throughput which is the rate at which measurements are made, converted to engineering units and stored in Final memory. With the 700X Control Module (6303 CPU board), the maximum throughput rate for fast, single-ended measurements is approximately 310 measurements per second (1 second execution: Instruction 1 entered 4 times, 3 times with 99 repetitions, once with 11 repetitions).

Data is stored in Final Memory only by Output Processing Instructions and only when the Output Flag is set.

The default case for data stored in Final Memory is low resolution (4 characters). High resolution values (5 characters) must be specified through use of Instruction 78. All data contained in Input Memory is displayed (*6) as HIGH RESOLUTION (5 characters) but the default case for all data stored in Final Memory is LOW RESOLUTION unless high resolution is specified through use of Instruction 78.

Floating Point Format - The computations performed in the CR7 use floating point arithmetic. CSI's 4 byte floating point numbers contain a 23 bit binary mantissa and a 6 bit binary exponent. The largest and smallest numbers that can be stored and processed are $9 \times 10^{18}$ and $1 \times 10^{-19}$, respectively.

The computations performed in the CR7 are done in floating point arithmetic. Internally, the number is stored and processed as a binary number with a 23 bit binary mantissa and a 6 bit binary exponent. The largest and smallest numbers that can be stored and processed are $9 \times 10^{18}$ and $1 \times 10^{-19}$ respectively. The size of the mantissa limits the resolution of the arithmetic to 1 part in $2^{23}$ binary ($1.3 \times 10^{9}$ decimal).

Time is stored with data in Final Memory only if specifically requested through use of the Real Time Instruction 77.

Data in Final Storage can be erased without altering the program by using the *A Mode to repartition memory. The simplest method is to re-enter the current allocation for Input Storage (32 locations is the default allocation). All memory can be erased and the CR7 completely reset by entering 1744 for the number of bytes left in Program Memory.

On-line (as opposed to a manually initiated dump) data transfer to peripherals (printer, storage module, etc.) occurs only if enabled through use of the *4 Mode or Instruction 96.

Data transfer to cassette tape is no longer supported.

# CAUTIONARY NOTES

The typical current drain for the CR7 is approximately 100 mA while executing and 8-10 mA quiescent. Do not allow the lead-acid batteries (2.5 Ahr) to drop below 11.76 V as irreversible battery damage may result.

An external battery connected to the I/O Module +12V and ground terminals continues to power the CR7 system even though the CR7 power switch is off. Reverse polarity protection is NOT provided on this connection so exercise extreme care if connecting external power supplies.

Damage will occur to the analog input channel circuitry if voltages in excess of +16V are applied for a sustained period.

A POTENTIALLY DANGEROUS situation can result due to hydrogen gas build up if the CR7 is housed in a gas tight enclosure and the internal lead acid batteries are shorted or overcharged. Hydrogen concentration levels may occur which are capable of causing injury or equipment damage if ignited.

# CR7 MEASUREMENT AND CONTROL SYSTEM OVERVIEW

*The CR7 Measurement and Control System combines precision measurement with processing and control capability in a battery operated system.*

*Campbell Scientific, Inc. provides three documents to aid in understanding and operating the CR7:*

*1.  This Overview*
*2.  The CR7 Operator's Manual*
*3.  The CR7 Prompt Sheet*

*This Overview introduces the concepts required to take advantage of the CR7's capabilities.  Hands-on programming examples start in Section OV4.  Working with a CR7 will help the learning process, so don't just read the examples, turn on the CR7 and do them.  If you want to start this minute, go ahead and try the examples, then come back and read the rest of the Overview.*

*The sections of the Operator's Manual which should be read to complete a basic understanding of the CR7 operation are the Programming Sections 1-3, the portions of the data retrieval Sections 4 and 5 appropriate to the method(s) you are using (see OV5), and Section 14 which covers installation and maintenance.*

*Section 6 covers the details of serial communications.  Sections 7 and 8 contain programming examples. Sections 9-12 have detailed descriptions of the programming instructions, and Section 13 goes into detail on the CR7 measurement procedures.*

*The Prompt Sheet is an abbreviated description of the programming instructions.  Once familiar with the CR7, it is possible to program it using only the Prompt Sheet as a reference, consulting the manual if further detail is needed.*

*Read the Selected Operating Details and Cautionary Notes at the front of the Manual before using the CR7.*

## OV1.  PHYSICAL DESCRIPTION

The CR7 features a modular, multiple processor design that provides precision measurement and control capability in a rugged, battery operated system.  Control Module functions include real-time task initiation, measurement processing, data storage, telecommunications, and keyboard/display interaction.  The I/O Module performs all analog and pulse signal measurement functions as well as the analog and digital control output functions.  The I/O Module contains its own processor card, a precision analog interface card, and seven card slots which can accommodate any combination of I/O Cards. Sensor leads are connected to the I/O cards via screw terminals.

A maximum of four I/O modules, separated by up to 1,000 feet, may be connected to a single Control Module in applications that require distributed measurement capability.

### OV1.1 700X  CONTROL MODULE

Contains the CPU card, with 24K of system PROM and 40K of RAM; the serial interface card for peripheral communication and connection of up to four I/O Modules; and the keyboard display card.  Two slots are present for optional RAM expansion.  The system's 2.5 Ahr lead-acid batteries and AC charging circuitry are also contained in this module.

The CS I/O 9-pin port provides connection to data storage peripherals, such as the SM192/716 Storage Module, and provides serial communication to computer or modem devices for data transfer or remote programming (Section 6).  *This 9 pin port does NOT have the same pin configuration as the*

*RS232 9 pin serial ports used on many computers.*

The SDM terminals adjacent to the serial port allow connection to Synchronous Device for Measurement (SDM) peripherals. These peripherals include the SDM-INT8 Interval Timer, the SDM-SW8A Switch Closure Module, the SDM-CD16AC AC/DC Controller, and the SDM-OBDII Engine Controller Interface.

**709    512K MEMORY CARD:** This card provides RAM storage for an additional 262,126 Final Data values. Only one 709 card may be installed.

## OV1.2  720 I/O MODULE

The processor card provides regulated power for analog and digital functions from the unregulated 12 volt supply. The analog interface card contains a 16-bit A/D-D/A converter, and a precision voltage reference. The standard I/O Module contains slots for 7 I/O Cards; the expanded Model 720XL contains 14 slots. All input and output connections to the I/O module are transient protected with spark gaps.

The +12 volt and ground terminals provide a direct connection to the CR7 power supply.

**723    ANALOG INPUT CARD:** Contains 14 differential or 28 single ended inputs. Input ground terminals connect to a heavy copper bar, which reduces single ended measurement offsets to less than 5µV.

**723-T  ANALOG INPUT CARD WITH RTD:** Identical to the 723 Card except that a platinum resistance thermometer is mounted in the center of the terminal strip. The PRT provides a reference junction temperature for thermocouple measurement. The PRT measurement is accurate to ±0.1°C over a range of -40°C to +60°C.

The numbering on the terminals refers to the differential channels; i.e., the voltage on the HI input is measured with respect to the voltage on the Low input. When making single-ended measurements either the HI or the Low channel may be used independently to measure the voltage with respect to the CR7 ground. Single-ended channels are numbered sequentially, e.g., the HI and LOW sides of differential channels 2 are single-ended channels 3 and 4, respectively (Section 13.2).

**724    PULSE COUNTER CARD:** Provides 4 pulse counting channels for switch closures, low level AC cycles, or high frequency pulse signals.

**725    EXCITATION CARD:** There are 8 switched analog excitation channels. These supply programmable excitation voltages for resistive bridge measurements. The excitation channels are only switched on during the measurement. Only one is on at a time.

The two Continuous Analog Output (CAO) channels supply continuous output voltages, under program control, for use with strip charts, X-Y plotters, or proportional controllers.

The 8 Digital Control Ports (0 or 5 volt states) allow on-off control of external devices. These control ports have a very limited current output (5mA) and are used to switch solid state devices which in turn provide power to relay coils (Section 14.4).

**726    50 VOLT ANALOG INPUT CARD:** Provides 8 differential or 16 single ended inputs for full scale DC ranges of ±50 V and ±15V. Resolution is 1.66 millivolts on the ±50 V and 0.5 millivolts on the ±15 V range. The common mode range is ±50 volts.

## OV1.3 ENCLOSURES AND CONNECTOR OPTIONS

**ENC-7L    ALUMINUM FRAME FOR LABORATORY ENVIRONMENTS:** 17" x 12" x 6"; provides a housing for benchtop use or a frame for attachment to a wall or a NEMA type enclosure.

**ENC-7F    ENVIRONMENTALLY SEALED FIBERGLASS ENCLOSURE:** 20" x 13" x 10"; housing for harsh environments. Sensor leads enter via two ports fitted with 0.75" conduit bushings, and plugged with removable stoppers. The 1.040" hole size accommodates #14 shell size circular connectors.
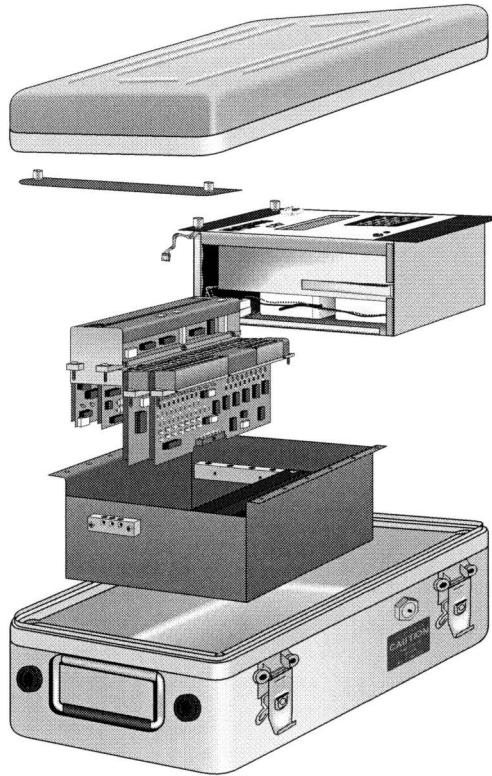
**FIGURE OV1-1.  CR7 Measurement and Control System**

## OV2.  MEMORY AND PROGRAMMING CONCEPTS

The CR7 must be programmed before it will make any measurements.  A program consists of a group of instructions entered into a program table.  The program table is given an execution interval which determines how frequently that table is executed.  When the table is executed, the instructions are executed in sequence from beginning to end.  After executing the table, the CR7 waits the remainder of the execution interval and then executes the table again starting at the beginning.

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

Figure OV2-1 represents the measurement, processing, and data storage sequence in the CR7 and shows the types of instructions used to accomplish these tasks.

### OV2.1  INTERNAL MEMORY

The CR7 has 40,960 bytes of Random Access Memory (RAM), divided into five areas.   The five areas of RAM are:

1.  **Input Storage** - Input Storage holds the results of measurements or calculations.  The *6 Mode is used to view Input Storage locations to check current sensor readings or calculated values.  Input Storage defaults to 28 locations.  Additional locations can be assigned using the *A Mode.

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

# CR7 MEASUREMENT AND CONTROL SYSTEM OVERVIEW

**ANALOG IPUTS**

Input/Output Instructions
1. Volt (SE)
2. Volt (DIFF)
4. Ex-Del-Se
5. AC Half Br
6. Full Br
7. 3W Half Br
9. Full Br-Mex
11. Temp (107)
12. RH-(07)
13. Temp-TC SE
14. Temp-TC DIFF
17. Temp-Panel

**SDM PORTS**

101 SDM-INT8
102 SDM-SW8
103 SDM-AO4
104 SDM-CD16
113 SDM-SIO4
115 Set SDM Clock
118 SDM-OBDII

**CS I/O PORT**

Telecommunications
Program Control Instructions
96 (Storage Module, Printer)
97 Initiate Telecommunications
98 Print Character

**PULSE INPUTS**

Input/Output Instructions
3. Pulse

**EXCITATION OUTPUTS**

Input/Output Instructions
4. Ex-Del-Se
5. AC Half Br
6. Full Br
7. 3W Half Br
9. Full Br-Mex
11. Temp (107)
12. RH (207)
22. Excit-Del

**CAO**

21 ANALOG OUT

**CONTROL PORTS**

Input/Output Instructions
20 Set Port
Program Control Instructions
83 If Case < F
86 Do
88 If x < = > y
89 If x < = > f
91 If flag, port
92 If Time
Command Codes:
4x Set port x high
5x Set port x low
6x Toggle port x
7x Pulse port x

**FIGURE OV1-2. CR7 Wiring Panel and Associated Programming Instructions**

**INPUT/OUTPUT INSTRUCTIONS**

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

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

**INPUT STORAGE**

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

**PROCESSING INSTRUCTIONS**

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

**OUTPUT PROCESSING INSTRUCTIONS**

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

**INTERMEDIATE STORAGE**

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

*Output Flag set high*

**FINAL STORAGE**

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

**FIGURE OV2-1. Instruction Types and Storage Areas**

3. **Final Storage** - Final, processed values are stored here for transfer to printer, solid state Storage Module or for retrieval via telecommunication links. Values are stored in Final Storage only by the Output Processing Instructions and only when the Output Flag is set in the users program. The 18,336 locations allocated to Final Storage at power up is reduced if Input or Intermediate Storage is increased.

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

5. **Program Memory** - available for user programs entered in Program Tables 1 and 2, and Subroutine Table 3. (Sections OV3, 1.1)

The use of the Input, Intermediate, and Final Storage in the measurement and data processing sequence is shown in Figure OV2-1. While the total size of these three areas remains constant, memory may be reallocated between the areas to accommodate different measurement and processing needs (*A Mode, Section 1.5). The size of system and program memory are fixed.

## OV2.2  CR7 INSTRUCTION TYPES

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

1. **INPUT/OUTPUT INSTRUCTIONS** (1-26,101-104, Section 9) control the terminal strip inputs and outputs (the sensor is the source, Figure OV1-2), storing the results in Input Storage (destination). Multiplier and offset parameters allow conversion of linear signals into engineering units. The Control Ports and Continuous Analog Outputs are also addressed with I/O Instructions.

2. **PROCESSING INSTRUCTIONS** (30-66, Section 10) perform numerical operations on values located in Input Storage (source) and store the results back in Input Storage (destination). These instructions can be used to develop high level algorithms to process measurements prior to Output Processing (Section 10).

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

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

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

4. **PROGRAM CONTROL INSTRUCTIONS** (85-98, Section 12) are used for logic decisions and conditional statements. They can set flags, compare values or times, execute loops, call subroutines, conditionally execute portions of the program, etc.

## OV2.3  PROGRAM TABLES AND THE EXECUTION AND OUTPUT INTERVALS

Programs are entered in Tables 1 and 2. Subroutines, called from Tables 1 and 2, are entered in Subroutine Table 3. The size of each table is flexible, limited only by the total amount of program memory. If Table 1 is the only table programmed, the entire program memory is available for Table 1.

Table 1 and Table 2 have independent execution intervals, entered in units of seconds with an allowable range of 0.0125 to 6553 seconds. Intervals shorter than 0.1 seconds are allowed only in Table 1. Subroutine Table 3 has no execution interval; subroutines are only executed when called from Table 1 or 2.

| Table 1.<br>Execute every x sec.<br>$0.0125 \leq x \leq 6553$ | Table 2.<br>Execute every y sec.<br>$0.1 \leq y \leq 6553$ | Table 3.<br>Subroutines |
|---|---|---|
| *Instructions are executed sequentially in the order they are entered in the table. One complete pass through the table is made each execution interval unless program control instructions are used to loop or branch execution.* | *Table 2 is used if there is a need to measure and process data on a separate interval from that in Table 1.* | *A subroutine is executed only when called from Table 1 or 2.* |
| Normal Order:<br>MEASURE<br>PROCESS<br>CHECK OUTPUT COND.<br>OUTPUT PROCESSING | | Subroutine Label<br>Instructions<br>End |
| | | Subroutine Label<br>Instructions<br>End |
| | | Subroutine Label<br>Instructions<br>End |

**FIGURE OV2-2. Program and Subroutine Tables**

## OV2.3.1 THE EXECUTION INTERVAL

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

Each instruction in the table requires a finite time to execute. If the execution interval is less than the time required to process the table, the CR7 overruns the execution interval, finishes processing the table and waits for the next execution interval before initiating the table. When an overrun occurs, decimal points are shown on either side of the G on the display in the LOG mode (*0). Overruns and table priority are discussed in Section 1.1.

## OV2.3.2 THE OUTPUT INTERVAL

The interval at which output occurs is independent from the execution interval, other than the fact that it must occur when the table is executed (i.e., a table cannot have a 10 minute execution interval and output every 15 minutes).

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

## OV3. PROGRAMMING THE CR7

A program is created by keying it directly into the datalogger or on a PC using the PC208 or PC208W Datalogger Support Software program EDLOG. This manual describes direct interaction with the CR7. Work through the direct programming examples in this overview before using EDLOG and you will have the basics of CR7 operation as well as an appreciation for the help provided by the software. Section OV3.5 describes options for loading the program into the CR7.

## OV3.1 FUNCTIONAL MODES

User interaction with the CR7 is broken into different functional MODES, (e.g., programming the measurements and output, setting time, manually initiating a block data transfer to Storage Module, etc.). The modes are referred to as Star (*) Modes since they are accessed by first keying *, then the mode number or letter. Table OV3.1 lists the CR7 Modes.

### TABLE OV3-1. * Mode Summary

| Key | Mode |
| --- | --- |
| *0 | LOG data and indicate active Tables |
| *1 | Program Table 1 |
| *2 | Program Table 2 |
| *3 | Program Table 3, subroutines only |
| *4 | Enable/disable printer output |
| *5 | Display/set real time clock |
| *6 | Display/alter Input Storage data, toggle flags |
| *7 | Display Final Storage data |
| *8 | Final Storage data transfer to cassette tape |
| *9 | Final Storage data transfer to printer |
| *A | Memory allocation/reset |
| *B | Signature test/PROM version |
| *C | Security |
| *D | Save/load Program |

## OV3.2 KEY DEFINITION

Keys and key sequences have specific functions when using the CR7 keyboard or a terminal/computer in the remote keyboard state (Section 5). Table OV3.2 lists these functions. In some cases, the exact action of a key depends on the mode the CR7 is in and is described with the mode in the manual.

### TABLE OV3-2. Key Description/Editing Functions

| Key | Action |
| --- | --- |
| 0-9 | Key numeric entries into display |
| * | Enter Mode (followed by Mode Number) |
| A | Enter/Advance |
| B | Back up |
| C | Change the sign of a number or index an input location to loop counter |
| D | Enter the decimal point |
| # | Clear the rightmost digit keyed into the display |
| #A | Advance to next instruction in program table (*1, *2, *3) or to next output array in Final Storage (*7) |
| #B | Back up to previous instruction in program table or to previous output array in Final Storage |
| #D | Delete entire instruction |

## OV3.3 PROGRAMMING SEQUENCE

In routine applications, sensor signals are measured, processed over some time interval, and the results are stored in Final Storage. A generalized programming sequence is:

1. Enter the execution interval, determined by the desired sensor scan rate.

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

3. Enter any Processing Instructions required to get the data ready for Output Processing.

4. Enter a Program Control Instruction to test the output condition and Set the Output Flag when the condition is met. For example, use Instruction 92 to output based on time, 86 to output each time the table is executed, and 88 or 89 to compare input values. This instruction must precede the Output Processing Instructions.

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

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

## OV3.4  INSTRUCTION FORMAT

Instructions are identified by an instruction number. Each instruction has a number of parameters that give the CR7 the information it needs to execute the instruction.

The CR7 Prompt Sheet has the instruction numbers in red, with the parameters briefly listed in columns following the description. Some parameters are footnoted with further description under the "Instruction Option Codes" heading.

For example, Instruction 73 stores the maximum value that occurred in an Input Storage Location over the output interval. The instruction has three parameters (1) REPetitionS, the number of sequential Input Storage locations on which to find maxima, (2) TIME, an option of storing the time of occurrence with the maximum value, and (3) LOC the first Input Storage Location operated on by the Maximum Instruction. The codes for the TIME parameter are listed in the "Instruction Option Codes".

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

Detailed descriptions of the instructions are given in Sections 9-12.

## OV3.5  ENTERING A PROGRAM

Programs are entered into the CR7 in one of four ways:

1.  Keyed in using the CR7 keyboard.

2.  Loaded from a pre-recorded listing using the *D Mode. There are two types of storage/input:

    a.  Stored on disk/sent from computer (PC208 software).

    b.  Stored/loaded from SM192/716 Storage Module

3.  Loaded from Storage Module or internal PROM (special software) upon power-up.

A program is created by keying it directly into the datalogger as described in the following Section, or on a PC using the PC208 Datalogger Support Software.

PC208 Software programs are used to develop and send programs to the CR7. Program files developed can be downloaded directly to the CR7 via direct wire, telephone, or Radio Frequency (RF).

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

If the SM192/716 Storage Module is connected when the CR7 is powered-up, the CR7 will automatically load program number 8, provided that a program 8 is loaded in the Storage Module (Section 1.8).

It is also possible (with special software) to create a PROM (Programmable Read Only Memory) that contains a datalogger program. With this PROM installed in the datalogger, the program will automatically be loaded and run when the datalogger is powered-up, requiring only that the clock be set.

# OV4.  PROGRAMMING EXAMPLE

The best way to become acquainted with the CR7 is to program it and make some measurements. If your CR7 contains either a 723 or 723-T Analog Input card, a short copper-constantan thermocouple (TC) should be connected to channel 5. In this example, you will program the CR7 to sample the thermocouple temperature. If you have not purchased the 723-T with a Resistive Temperature Device (RTD) to measure the TC reference junction temperature, a "dummy" reference temperature will be used.

Tables OV3-1 and OV3-2 summarize the Keyboard Commands and Control Modes used to program the CR7, monitor Input and Final Storage and control data output to peripherals. The instructions, and their associated parameters, are the CR7's programming steps and are used to build the CR7's program. It is not necessary to understand all the commands to proceed with this programming exercise. It is helpful to find the example's instructions on the CR7 Prompt Sheet provided with this manual. As you become familiar with programming the CR7, you will find that the Prompt Sheet or the PC208 program EDLOG has all the information you need to write your program. By following along on the Prompt Sheet as you proceed with this exercise, you will learn how to use it to write your own programs.

## OV4.1 MEASUREMENT

To make a thermocouple temperature measurement, the CR7 must know the temperature of the reference junction. The CR7 takes the reference temperature, converts it to the equivalent TC voltage, adds the measured TC voltage and converts the sum to temperature through a polynomial fit to the TC output curve. In this example, the reference junction is at the Analog Input Card. Its temperature is measured with Instruction 17, Panel Temperature. If you have an Analog Input Card with RTD, check to see which number is assigned to it. A tag labeled RTD is on the left hand side and the card number is on the right hand side of the Analog Input Card. If the RTD card is not card 1, you must enter the correct card number as Parameter 1 of Instruction 17. If you do not have an Analog Input Card with RTD, you will omit Instruction 17 from the Program and enter a "dummy" reference temperature after the Program is compiled.

The thermocouple temperature measurement is made using Instruction 14 (differential voltage measurement of TC) on differential channel 5. When using a copper-constantan thermocouple, the copper lead is connected to the high input of a differential channel and the constantan lead is connected to the low side. The channel numbering printed on the Analog Input Cards refers only to differential channels.

Either the high or low side of a differential channel may be used for single ended measurements. (Each side is counted when assigning single ended channel numbers; e.g., the high side of differential channel 8 is single ended channel 15 and the low side is single ended channel 16).

The first parameter in Instruction 14 is the number of times to repeat the measurement: 1 is entered because only one thermocouple is measured. If more thermocouple measurements were desired, the copper leads would be connected to the high sides of consecutive differential channels, the constantan leads to the low sides and the number of repetitions entered in Parameter 1 would equal the number of thermocouples.

Parameter 2 is the voltage range to use when making the measurement. The output of a copper-constantan thermocouple is approximately 40 microvolts per $^{\circ}$C difference in temperature between the two junctions. The +5000 uV scale will provide a range of +5000/40 = +125 $^{\circ}$C (i.e., this scale will not overrange as long as the measuring junction is within 125 $^{\circ}$C of the panel temperature). The resolution of the +5000 uV range is 166 nV or 0.004 $^{\circ}$C.

Parameter 3 is the Input Card number and Parameter 4 is the channel on which to make the first measurement. If more than one thermocouple is measured, the CR7 will automatically advance through the channels and on to the next card if necessary. Similarly, Parameter 7 is the Input Storage Location in which to store the first measurement; e.g., if there are five repetitions and the first measurement is stored in location 3, the final measurement will be stored in location 7.

Parameter 6 is the Input Storage location in which the reference temperature is stored, and Parameters 8 and 9 are the multiplier and offset to apply to the temperature value. A multiplier of 1 and an offset of 0 give the result in $^{\circ}$C, a multiplier of 1.8 and an offset of 32 give the result in $^{\circ}$F.

Now that you have some idea of what you are telling the CR7 by entering the parameters, we will proceed with programming the CR7.

**TABLE OV4-1. Thermocouple Measurement Programming Example**

TURN ON THE POWER SWITCH AND PROCEED AS FOLLOWS:

| Display ID:Data | Key | Display ID:Data | Key | Description |
|---|---|---|---|---|
| HELLO 01 | | :0064 | * | The number after "HELLO" will count up as memory is checked. If you have a 512K Memory Card, this can take a long time; key # to abort the test. The result of the CPU board memory check is then displayed (Sect. 1.5) |
| 00:00 | 1 | 01:00 | A | Enter Program Table 1, advance to Execution Interval |
| 01:0.0000 | 2 | 01:2 | A | Enter 2 second Execution Interval advance to first instruction |

-------Users without RTD omit next Instruction------

| | | | | |
|---|---|---|---|---|
| 01:P00 | 17 | 01:P17 | A | Measure Panel Temp., advance to first Parameter |
| 01:00 | 1 | 01:1 | A | RTD in input card #1, if RTD card other than #1, enter correct card # |
| 02:0000 | 1 | 02:1 | A | Store temp in location 1 |

-------Users without RTD continue here-------
Instruction Location Number will be 1 less (i.e., 01:P00)

| | | | | |
|---|---|---|---|---|
| 02:P00 | 14 | 02:P14 | A | TC temp., differential meas. |
| 01:00 | 1 | 01:1 | A | 1 repetition |
| 02:00 | 2 | 02:2 | A | Range code (5000uV, slow) |
| 03:00 | 1 | 03:1 | A | Input card #1 |
| 04:00 | 5 | 04:5 | A | Input channel of 1st TC |
| 05:00 | 1 | 05:1 | A | TC type (copper-constantan) |
| 06:0000 | 1 | 06:1 | A | Reference temp. is in location 1 |
| 07:0000 | 2 | 07:2 | A | Store TC temp. in location 2 |
| 08: 0.0000 | 1 | 08:1 | A | Multiplier of 1 |
| 09: 0.0000 | | 09:0.0000 | A | No offset entered (offset=0), advance to next instruction |
| 03:P00 | | | * | Exit Table 1 |
| 00:00 | 0 | :LOG 1 | | Enter *0 Mode, compile table |

*The CR7 is now programmed to measure the thermocouple temperature and to store the result in Input Storage Location 2. The colon between the ID and Data fields blinks each time Table 1 is executed, every 2 seconds in this example. If you do not have an RTD, the "reference temperature" is 0.0 and the value stored in Location 2 is the difference in temperature between the panel and the thermocouple. The *6 Mode can be used to monitor the values in the Input Storage and to change the value of the dummy reference temperature.*

**TABLE OV4-2. Using *6 Mode to Observe Example TC Measurements**
**(User with Model 723-T RTD Card)**

| Display ID:Data | Key | Display ID:Data | Key | Description |
|---|---|---|---|---|
| :LOG 1 | *6 | 06:0000 | A | Enter *6 Mode, advance to first location |
| | | 01:21.234 | A | Panel temp is 21.234 $^\circ$C, advance to location 2 |
| | | 02:22.433 | B | TC temp is 22.433 $^\circ$C, backup to location 1 |
| | | 01:21.199 | * | Panel temp is now 21.199 $^\circ$C |
| 00:00 | 0 | :LOG 1 | | Return to *0 Mode |

**TABLE OV4-3. Using *6 Mode to Observe Example TC Measurements**
**(User with Model 723 Card, No RTD)**

| Display ID:Data | Key | Display ID:Data | Key | Description |
|---|---|---|---|---|
| :LOG 1 | *6 | 06:0000 | A | Enter *6 Mode, advance to first location |
| | | 01:0.0000 | A | Reference temp is 0.0$^\circ$C, advance to location 2 |
| | | 02:2.9533 | B | TC "temp" is 2.9533 C, backup to location 1 |
| | | 01:0.0000 | C | Setup to change stored value |
| :0.0000 | 20 | :20 | A | Store 20 in location 1 |
| | | 01:20.000 | A | Advance to location 2 |
| | | 02:22.866 | * | The TC temp in location 2 using a reference temperature of 20$^\circ$ |
| 00:00 | 0 | :LOG 1 | | Return to *0 Mode |

You can advance through Input Storage by keying in the advance command, A, or backup by keying in the backup command, B. The Input Location you are observing is shown on the left in the display ID field. The temperature data stored in the Input locations are updated every 2 seconds, each time Table 1 is executed. Verify this by changing the temperature of the thermocouple (hold it in your fingers) while monitoring the proper Input Location.

It is possible to go directly to a specific Input Storage location by entering the *6 Mode and keying in the desired location before keying A. A similar utility is available in other Modes.

## OV4.2 OUTPUT

In the following example instructions are appended to Table 1 to output the time and the average temperatures to Final Storage every 5 minutes.

**TABLE OV4-4. Example Programming to Obtain Five Minute Averages**

| Display ID:Data | Key | Display ID:Data | Key | Description |
|---|---|---|---|---|
| | | : LOG 1 | * | |
| 00:00 | 1 | 01:00 | | Program Table 1 |
| 01:00 | 3 | 01:3 | A | Advance to 3rd Instruction location (Key in 2 if Instruction 17 was not entered, Instruction Location Number will be 1 less than shown in table) |
| 03:P00 | 92 | 03:P92 | A | Enter If Time Instruction |
| 01:0000 | 0 | 01:0 | A | Enter 0 minutes into interval |
| 02:0000 | 5 | 02:5 | A | Enter 5 minute time interval |
| 03:00 | 10 | 03:10 | A | Set output Flag 0 |
| 04:P00 | 77 | 04:P77 | A | Enter Output Time Instruction |
| 01:0000 | 10 | :10 | A | Code for HR:MIN |
| 05:P00 | 71 | 05:P71 | A | Enter Average Instruction |
| 01:00 | 2 | 01:2 | A | 2 repetitions |
| 02:0000 | 1 | 02:1 | A | Location of 1st input data to be averaged |
| 06:P00 | | | * | Exit Table 1 |
| 00:00 | 5 | :00:21:32 | A | Enter *5 Mode to set clock (the clock will be running) |
| 05:00 | 85 | 05:85 | A | Enter Year |
| 05:0000 | 11 | 05:11 | A | Enter Julian day (January 11 assumed in this example) |
| 05:00:21 | 1324 | 05:13:24 | A | Enter Hours:Minutes (24 hour time, 1:24 PM assumed in this example) |
| 13:24:01 | | | *0 | Exit *5 Mode, compile Table 1, commence logging data |
| : LOG 1 | | | | |

*The CR7 is now programmed to sample the panel and thermocouple temperatures every 2 seconds and to output the time and the average temperatures to Final Storage every 5 minutes. Each Output Array sent to Final Storage will consist of 4 data values. The first value will be an output identifier which gives the number of the Table which caused the output, and the instruction location number of the instruction which set the output flag. The second value will be the time, and the third and fourth values will be the average temperatures of the I/O Module and the thermocouple. Values stored in Final Storage can be viewed using the *7 Mode. Table 1.2-5 shows an example of the use of the *7 Mode, it is assumed that the CR7 has been logging data for 8 minutes since the time was set in the previous example.*

**TABLE OV4-5. Using *7 Mode to View Values in Final Storage**

| Display ID:Data | Key | Display ID:Data | Key | Description |
|---|---|---|---|---|
| :LOG 1 | | | * | |
| 00:00 | 7 | 07:9.0000 | A | Enter *7 Mode. The DSP is at Final Storage location 9, advance to first data value |
| 01:0103. | | | A | Output identifier: users who did not enter Instruction 17 will see 01: 0102 because the output flag is set by the second instruction in Table 1 |
| 02:1325. | | | A | Time |
| 03:22.57 | | | A | Average panel temp for readings between 1:24 and 1:25 P.M. |
| 04:23.43 | | | A | Average thermocouple temp. |
| 01:0103. | | | A | Output identifier |
| 02:1330. | | | A | Time |
| 03:22.61 | | | * | Average panel temp for readings between 1:25 and 1:30 P.M. |
| 00:00 | 0 | :LOG 1 | | Enter *0 Mode |

## OV4.3 EDITING AN EXISTING PROGRAM

When editing an existing program in the CR7, entering a new instruction inserts the instruction; entering a new value for an instruction parameter replaces the previous value.

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

An instruction is deleted by advancing to the instruction number (P in display) and keying #D (Table OV3-2).

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

## OV4.4 EDLOG PROGRAM LISTING

The examples in the rest of this manual use program listings generated by EDLOG, the datalogger Program Editor for the PC (PC208(W) Software). The EDLOG listing does not show the CR7 display or the "A" keystrokes used to enter data. The EDLOG listing for the previous example is given in Table OV4-6.

**TABLE OV4-6. EDLOG Listing of Example Program**

| * | 1 | Table 1 Programs |
|---|---|---|
| 01: | 2 | Sec. Execution Interval |
| | | |
| 01: | P17 | Panel Temperature |
| 01: | 1 | IN Card |
| 02: | 1 | Loc : |
| | | |
| 02: | P14 | Thermocouple Temp (DIFF) |
| 01: | 1 | Rep |
| 02: | 2 | 5000 uV slow Range |
| 03: | 1 | IN Card |
| 04: | 5 | IN Chan |
| 05: | 1 | Type T (Copper-Constantan) |
| 06: | 1 | Ref Temp Loc |
| 07: | 2 | Loc [:TC Temp ] |
| 08: | 1 | Mult |
| 09: | 0 | Offset |
| | | |
| 03: | P92 | If time is |
| 01: | 0 | minutes into a |
| 02: | 5 | minute interval |
| 03: | 10 | Set high Flag 0 (output) |
| | | |
| 04: | P77 | Real Time |
| 01: | 10 | Hour-Minute |
| | | |
| 05: | P71 | Average |
| 01: | 2 | Reps |
| 02: | 1 | Loc |

## OV5. DATA RETRIEVAL OPTIONS

There are several options for data storage and retrieval. These options are covered in detail in Sections 2, 4, and 5. Figure OV5-1 summarizes the various possible methods.

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

1. On-line output of Final Storage data to a peripheral storage device. On a regular schedule, that storage device is brought back to the office/lab where the data is transferred to the computer. Another storage device is usually taken into the field and exchanged for the one which is retrieved so that data collection can continue uninterrupted.

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

3. Retrieve the data over some form of telecommunications link, that is, Radio Frequency (RF), telephone, short haul modem, multi-drop interface, or satellite. The PC208 software automates this process.

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

- they are written over by new data (Section 2.1)

- memory is reallocated (Section 1.5)

- the power to the datalogger is turned off.

Table OV5-1 lists the instructions used with the various methods of data retrieval.

**TABLE OV5-1. Data Retrieval Methods and Related Instructions**

| Storage Module | Printer, other Serial Device | Telecommunications (RF, Phone, Short Haul, SC32A) |
|---|---|---|
| Inst. 96, | Inst. 96, 98 | Inst. 97 |
| *4 | *4 | |
| *9 | *9 | (Telecommunications Commands) |
| *D | *D | |

**TABLE OV5-2. Data Retrieval Sections in Manual**

| Topic | Section in Manual |
|---|---|
| Instr. 96 | 4.1, 12 |
| Instr. 97 | 12 |
| *4 | 4.1 |
| *8 | 4.2 |
| *9 | 4.2 |
| *D | 1.8 |
| Storage Module | 4.3 |
| Telecommunications | 5 |

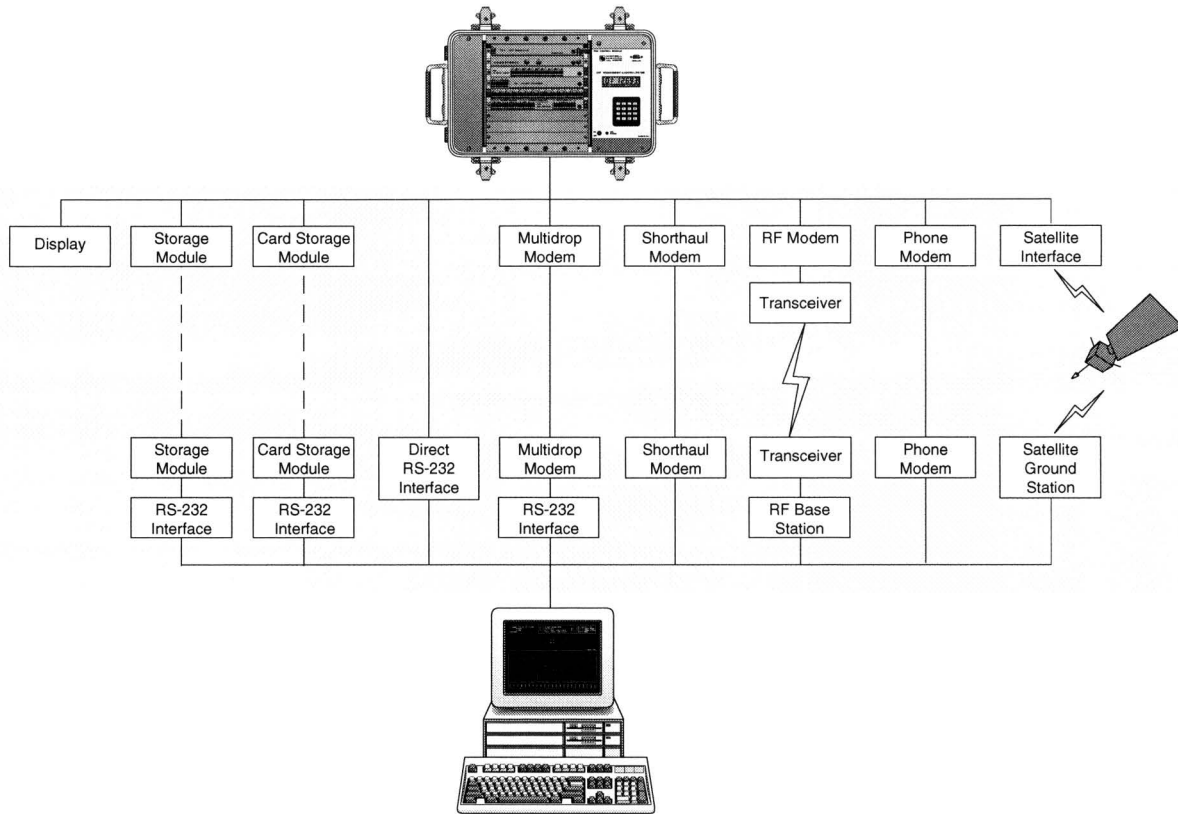# CR7 MEASUREMENT AND CONTROL SYSTEM OVERVIEW



**FIGURE OV5-1. Data Retrieval Hardware Options**

# OV6. SPECIFICATIONS

*Electrical specifications are valid for over a -25° to +50°C range unless otherwise specified.*

## Analog Inputs

(723T or 723 Card specifications below; 726 ±50 V Card specifications discussed in *System Description*)

**Voltage Measurement Types:** Single-ended or differential.

**Range and Resolution:** Ranges are software selectable on any input channel.

| Full Scale Input Range (mV) | Resolution Differential | | Single-ended | |
|---|---|---|---|---|
| ±5000 | 166 | µV | 333 | µV |
| ±1500 | 50 | µV | 100 | µV |
| ±500 | 16.6 | µV | 33.3 | µV |
| ±150 | 5 | µV | 10 | µV |
| ±50 | 1.66 | µV | 3.33 | µV |
| ±15 | 500 | nV | 1000 | nV |
| ±5 | 166 | nV | 333 | nV |
| ±1.5 | 50 | nV | 100 | nV |

**Accuracy of Voltage Measurements:**
Differential: ±0.02% FSR (±0.01%, 0-40°C)
(e.g. ±0.02% FSR = ±2.0 mV for ±5 V range)
Positive single-ended: ±0.02% FSR (±0.01%, 0-40°C) ±5 µV
Negative single-ended: ±0.03% FSR (±0.015%, 0-40°C) ±5 µV

**Input Sample Rates:** Fast A/D conversions are integrated over 250 µs. Slow A/D conversions are integrated over 16.67 ms for 60 Hz AC rejection or optionally, 20.0 ms for 50 Hz AC rejection. Differential measurements include two conversions, one with reversed input polarity, to reduce thermal offset and common mode errors. The following intervals do not include the self-calibration measurement which occurs once per instruction.

| | Input sample rates ms/channel | Typical input noise nV/RMS |
|---|---|---|
| Fast Single-ended | 2.9 | 350 |
| Fast Differential | 4.7 | 250 |
| Slow Single-ended | 22.0 | 43 |
| Slow Differential | 43.0 | 30 |
| Fast Differential (TC) | 7.9 | 250 |

**Common Mode Range:** ±5 V

**Common Mode Rejection:** > 140 dB (DC to 100 Hz)

**Normal Mode Rejection:** 70 dB (60 Hz with slow differential measurement)

**Input Current:** 100 pA max

**Input Current Noise:** 9 pA RMS (slow differential)

**Input Resistance:** 2.5 GΩ typical

**Sustained Input Voltage without Damage:** ≤ ±16 VDC

## Pulse Counters

(724 Card)

**Pulse Counters per Card:** 4

**Maximum Counts per Interval:** 32,767 (with overrange detection)

**Modes:** Programmable modes are switch closure, high frequency pulse, and low level AC.

**Switch Closure Mode**
Minimum Switch Closed Time: 1 ms
Minimum Switch Open Time: 4 ms
Maximum Bounce Time: 1.4 ms open without being counted.

**High Frequency Pulse Mode**
Minimum Pulse Width: 2 µs

Maximum Input Frequency: 250 kHz

Voltage Thresholds: The count is incremented when the input voltage changes from below 1.5 V to above 3.5 V.

Maximum Input Voltage: ±20 V

**Low Level AC Mode**
This mode is used for counting the frequency of low voltage, sine wave signals.

Input Hysteresis: 11 mV

Maximum AC Input Voltage (RMS): 20 V

Frequency Range:

| Minimum AC Input Voltage (mV RMS) | Range (Hz) |
|---|---|
| 15 | 1 to 100 |
| 25 | 1 to 1,000 |
| 50 | 1 to 3,000 |
| 160 | 1 to 10,000 |

## Digital Control Outputs

(725 Card)

Each card includes 8 digital control outputs.

**Output Voltages (no load):**
High: 5.0 V ±0.1 V
Low: < 0.1 V

**Output Resistance:** 400 Ω

## Analog Outputs

(725 Card)

Each card contains 8 switched and 2 continuous analog outputs.

**Switched:** Provides a precision voltage for resistance measurement, then switches off (high impedance). Only one switched output can be active at a time.

**Continuous:** A preset voltage is held until updated. Voltage degrades 0.17 mV every 7 seconds. All continuous analog outputs (and digital control ports) can be active simultaneously.

**Range:** ±5 V

**Resolution:** 166 µV

**Accuracy:** Same as voltage measurements.

**Output Current:** 25 mA at ±5 V, 50 mA at ±2 V

## Resistance and Conductivity Measurements

(Combination of 723 and 725 Cards)

**Accuracy:** ±0.01% of full scale bridge output provided the matching bridge resistors are not the limiting factor.

**Measurement Types:** 6-wire and 4-wire full bridge, 4-wire, 3-wire, and 2-wire half bridges. High accuracy, low impedance bridge measurements are made ratiometrically with dual polarity measurements of excitation and output to eliminate thermal emfs. AC resistance and conductivity measurements use a 750 µs excitation pulse with the signal integration occurring over the last 250 µs. An equal duration pulse of opposite polarity is applied for ionic depolarization.

## Transient Protection

All input and output connections to the I/O Module are protected using spark gaps that are rated to 10,000 A. The spark gaps are connected directly to a heavy copper bar on each input card with no more than 2 inches of 20 AWG copper wire.

## Control Module

**Processor:** Hitachi 6303

**Memory:** 24K ROM; 40K RAM, 709 Card provides an additional 512K RAM.

**Data Storage:** 18.8K values, standard; 280K values, expanded.

**Display:** 8 digit LCD (0.5" digits).

**Peripheral Interface:** 9-pin, D-type connector on the Control Module panel for connection to storage module, card storage module, multidrop interface, modem, printer, or RS-232 adapter. Baud rates selectable at 300, 1200, 9600, and 76,800.

**I/O Module Interface:** Optically isolated current loops allow connection of up to 4 I/O Modules. I/O Modules can be separated from the Control Module by up to 1,000 feet.

**Clock Accuracy:** ±1 minute per month.

**Maximum Program Execution Rate:** System tasks can be initiated in sync with real-time up to 80 Hz.

## System Power Requirements

**Voltage:** 9.6 to 15 VDC

**Typical Current Drain:** 3.5 - 6 mA (minimum system) quiescent, 16 mA during processing, 100 mA during analog measurement.

**Internal Batteries:** Sealed rechargeable with 2.5 Ahr capacity per charge.

**Charging Circuit:** Requires DC or rectified AC voltage from 15 to 25 V. Thermal compensation is included to optimize charging voltage according to ambient temperature.

**External Batteries:** Any 12 V external battery can be a primary power source; internal batteries provide a backup while the external batteries are changed.

**Operation from AC Sources:** An AC operated battery charger is included with the enclosure to maintain full charge on the batteries where AC power is available. In the event of power failure, the internal batteries will keep the system operational for up to 5 days in most applications.

## Physical Specifications

**Size:**
| | | |
|---|---|---|
| | ENC 7L | 17" x 12" x 6" |
| | ENC 7F | 20" x 13" x 10" |
| | ENC 7XL | 19" x 19" x 10" |

**Weight:** ~40 lbs (ENC 7F with 700X, 720, & seven I/O cards).

## Warranty

Three years against defects in materials and workmanship.

# SECTION 1. FUNCTIONAL MODES

## 1.1 PROGRAM TABLES - *1, *2, AND *3 MODES

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

When a program table is first entered, the display shows the table number in the ID Field and 00 in the Data Field. Press **A** and the CR7 will advance to the execution interval. If there is an existing program in the table, enter an instruction location number prior to **A** and the CR7 will advance directly to the instruction (e.g., 5 will advance to the fifth instruction in the table).

### 1.1.1 EXECUTION INTERVAL

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

0.0125 .... 0.1 seconds, in multiples of 0.0125

0.1 .....6553 seconds, in multiples of 0.1 second

Intervals less than 0.1 second are allowed in Table 1 only. Execution of the table is repeated at the rate determined by this entry. The table will not be executed if 0 is entered. Values less than 0.1 are rounded to the nearest even multiple of 0.0125. If the Interval is 0.1 or greater, the CR7 will not allow entry of digits beyond 0.1.

The rate at which the CR7 can execute a given table must not be confused with the sample rates for the measurements contained within the table. When a table is executed and a measurement is made, the Control Module instructs the I/O Module which measurement to make and how many times to repeat it on successive channels. The I/O module then repeats the measurement as fast as possible and stores the data until the Control Module is ready for it. The Control Module takes the raw data and scales it as required by the instruction initiating the measurement. The next instruction in the table is not executed until the scaling is completed. The maximum sample rate for a measurement is the rate at which the I/O Module can make a number of measurements specified by a single input instruction. Because the sample rate does not include the processing time required to scale the measurements into engineer units, the execution time of an instruction will be greater than the sample rate for the measurement specified by the instruction. The execution times for the instructions are given in Section 3.9.

The throughput rate is the rate at which a measurement can be made and the resulting value stored in Final Storage. The **maximum** throughput rate for fast single ended measurements is approximately 310 measurements per second.

If the specified execution interval for a table is less than the time required to process that table, the CR7 overruns the execution interval, finishes processing the table and waits for the next occurrence of the execution interval before again initiating the table (i.e., when the execution interval is up and the table is still executing, that execution is skipped). Since no advantage is gained in the rate of execution with this situation, it should be avoided by specifying an execution interval adequate for the table processing time.

> **NOTE:** Whenever an overrun occurs, decimal points are displayed on both sides of the sixth digit of the CR7 display (e.g., L O.G. in the *0 Mode).

When the Output Flag is set high, extra time is consumed by final output processing. It may be acceptable if the execution interval is exceeded at this time. For example, suppose it is desired to measure every 0.1 seconds and output processed data every ten minutes. The table requires less than 0.1 seconds to process except when output occurs (every 10 minutes). With final output processing the time required is one second. With the execution interval set at 0.1 seconds, and a one second lag between samples once every 10 minutes, 10 measurements out of 6000 (.17%) are missed: an acceptable statistical error for most populations.

### 1.1.2 SUBROUTINES

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

### 1.1.3 TABLE PRIORITY/INTERRUPTS

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

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

### 1.1.4 COMPILING A PROGRAM

When a program is entered, or any changes are made in the *1, *2, *3, *4, *A, or *C Modes, the program must be compiled before it starts running. The compile function checks for programming errors and optimizes program information for execution. If errors are detected, the appropriate error codes are indicated on the Display (Section 3.10). Compiling occurs when the *0 , *6, or *B Modes are entered and prior to saving a program listing in the *D Mode. Compiling only occurs after a program change has been made; subsequent use of any of these Modes does not cause compiling.

---

Compiling with the *0, *B, or *D Mode sets all output ports and flags low and resets the timer (Instruction 26) and all data in Input and Intermediate Storage to ZERO.

When the *6 Mode is used to compile data in Input Storage, the state of flags, control ports, and the timer are UNALTERED. Compiling always zeros Intermediate Storage.

---

## 1.2 SETTING AND DISPLAYING THE CLOCK - *5 MODE

The *5 Mode is used to display time or change the year, day of year, or time. When *5 is pressed, the current time is displayed. The time parameters displayed in the *5 Mode are given in Table 1.2-1.

The CR7 powers-up with hours and minutes set to 0 and the day and year set for the date that the PROMs were first released by Campbell Scientific. To set the year, day, or time, enter the *5 Mode and advance to display the appropriate value. Key in the desired number and enter the value by pressing A. When a new value for hours and minutes is entered, the seconds are set to zero and current time is again displayed. To exit the *5 Mode, press *. When the time is changed, a partial recompile is done automatically to resynchronize program execution with real time. The resynchronization process can change the interval of a pulse rate measurements for one execution interval as explained in the PULSE COUNT Instruction 3 in Section 9.

---

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

| Key | Display ID:DATA | Description |
|-----|-----------------|-------------|
| *5 | :HH:MM:SS | Display current time |
| A | 05:XX | Display/enter year |
| A | 05:XXXX | Display/enter day of year |
| A | 05:HH:MM: | Display/enter hours:minutes |

---

## 1.3 DISPLAYING AND ALTERING INPUT MEMORY OR FLAGS - *6 MODE

The *6 Mode is used to display or change Input Storage values and to toggle and display user flags. If the *6 Mode is entered immediately following any changes in program tables or the *4 Mode, the programs will be compiled and execution will begin.

When the *6 Mode is used to compile data values contained in Input Storage, the state of flags, control ports, and the timer are UNALTERED. Compiling always zeros Intermediate Storage.

## TABLE 1.3-1. *6 Mode Commands

| Key | Action |
|-----|--------|
| A | Advance to next location or enter new value |
| B | Back-up to previous location |
| C | Change value in displayed location(Key **C**, then value, then **A**) |
| D | Display/alter user flags |
| # | Display current location and allow a location no. to be keyed in, followed by A to jump to that location |
| * | Exit *6 Mode |

### 1.3.1 DISPLAYING AND ALTERING INPUT STORAGE

When *6 is keyed, the display will read "06:0000". One can advance to view the value stored in Input Storage location 1 by pressing **A**. To go directly to a specific location, key in the location number before keying **A**. For example, to view the value contained in Input Storage location 20, key in **\*6 20 A**. The ID portion of the display shows the last two digits of the location number. If the value stored in the location being monitored is the result of a program instruction, the value will be updated each time the instruction is executed.

Values may be entered into input locations using the change command, **C**. While viewing the contents of the input location in which the value is to be entered, key **C**; the location number in the ID field will disappear. Key in the desired value and then enter it by pressing **A**.

If an algorithm requires parameters to be manually modified during execution of the program WITHOUT INTERRUPTION of the Table execution process, the parameters can be loaded in Input Storage locations and the *6 Mode can be used to change the values. If values must be in place before program execution commences, use Instruction 91 at the beginning of the program table to prevent execution until a flag is set high (see next section). The initial values can be entered into input locations using the *6 Mode after compiling the table. The flag can then be set high to enable the table(s).

If any program tables *1, *2, *3, or *4 output options are altered and complied in the *0 Mode, values in Input Storage will be set to 0. To preserve values entered in Input Storage, compile with *6.

### 1.3.2 DISPLAYING AND TOGGLING USER FLAGS

If D is keyed while the CR7 is displaying a location value, the current status of the user flags will be displayed in the following format: "00:01:00:10". The characters represent the flags, the left-most digit represents Flag 1 and right most Flag 8. A "0" indicates the flag is low and a "1" indicates the flag is high. In the above example, Flags 4 and 7 are set high. To toggle a flag, simply key the corresponding number. To return to displaying the input location, press **A**.

Entering appropriate flag tests into the program allows manual control of program execution. For example: It is desired to be able to manually start the execution of Table 2. Instruction 91 is the first instruction entered in Table 2:

| 01: | P91 | If Flag |
|-----|-----|---------|
| 01: | 25 | 5 is set low |
| 02: | 0 | Go to end of program table |

If Flag 5 is low, all subsequent instructions in Table 2 will be skipped. Flag 5 can be toggled from the *6 Mode, effectively starting and stopping the execution of Table 2.

### 1.3.3 DISPLAYING AND TOGGLING PORTS

The current status of the Digital Control ports on the active 725 excitation card can be displayed by hitting "0" while looking at an input location (e.g., *6A0). Ports are displayed left to right as C8, C7, ..., C1 (exactly opposite to the flags). A port can be toggled by pressing its number on the keypad while in the port display mode.

The active excitation card defaults to address 1. The active card may only be changed with Instruction 20 in the CR7 Program (Section 9).

## 1.4 COMPILING AND LOGGING DATA - *0 MODE

When the *0 Mode is entered after programming the CR7, the program is compiled (Section 1.1.4) and the display shows "LOG" and the numbers of the program tables that were enabled at compilation. The display is not updated after entering *0.

# SECTION 1. FUNCTIONAL MODES

When the *0, *B, or *D Mode is used to compile, all output ports and flags are set low, the timer (Instruction 26) is reset, and data in Input and Intermediate Storage are RESET TO ZERO.

The CR7 should normally be left in the *0 Mode when logging data. This Mode requires slightly less power than Modes which frequently update the display.

## 1.5 MEMORY ALLOCATION - *A

### 1.5.1 INTERNAL MEMORY

There are eight sockets on the CR7 CPU board which are used for Read Only Memory (ROM) or Random Access Memory (RAM). The basic CR7 is provided with 64K of memory: three 8K Programmable Read Only Memory (PROM) chips for a total of 24K ROM and five 8K RAM chips. Appendix E describes how to change RAM and ROM chips.

When powered up, the CR7 displays HELLO while performing a memory check. As the check is performed, a number on the right of the display is incremented as each 8 K block of memory is checked. With standard memory the count will stop at 8. If additional memory card(s) are present, the count will proceed accordingly. The Power-up memory check is quite extensive and can take considerable time if the 709 512K Memory Card is installed. To abort the extensive test (a shorter version is still performed), press the # key. When the memory test is completed, the number of K bytes of RAM plus ROM is displayed.

The size of RAM, including any additional memory cards which may be present, can be determined with the *A Mode (Section 2.4.2)

There are 1744 bytes allotted to program memory. This memory may be used for one program table or shared among all program tables. Tables 3.9-1 to 3.9-4 list the amount of memory used by each program instruction.

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

Intermediate Storage is used by Output Processing Instructions to store the results of intermediate calculations necessary for averages, standard deviations, histograms, etc.

Final Storage holds output data, the results of Output Processing Instructions which are stored when the Output Flag is set high (Section 3.7). The data in Final Storage can be displayed using the *7 Mode (Section 2.3).

Figure OV2-1 illustrates the use of Input, Intermediate, and Final Storage.

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

Table 1.5-1 lists the basic memory areas and the amount of memory allotted to them in the standard CR7.

### TABLE 1.5-1. Memory Allocation in Standard CR7

| PROM | Memory | Program Memory | System Storage* | Input Storage* | Intermediate Storage* | Final Storage* Standard | Final Storage* w/709 Memory Card |
|---|---|---|---|---|---|---|---|
| Avail. bytes | 24K | 1744 | 2160 | 128 | 256 | 36,672 | 590,960 |
| Avail. Loc. | - | - | - | 32 | 64 | 18,336 | 280,480 |

*Default allocation on power-up

**TABLE 1.5-2. Description of *A Mode Data**

| Key Entry | Display ID: Data | Description of Data |
|---|---|---|
| *A | 01: XXXX | The number of memory locations currently allocated to Input Storage. This value can be changed by keying in the desired number (minimum of 32, maximum limited by available memory). |
| A | 02: XXXX | The number of memory locations currently allocated to Intermediate Storage. This value can be changed by keying in the desired number (limited by available memory). |
| A | 03: XXXXX | The number of memory locations currently allocated to Final Storage. This number is automatically altered when the number of memory locations in Input and/or Intermediate Storage is changed. A minimum of 768 locations are always retained in Final Storage. |
| A | 04:XXXX | The number of bytes remaining in Program memory (1744 bytes total). Entering 1744 will ERASE ALL MEMORY and put the CR7 through the initial power-up routine. |

### 1.5.2 *A MODE

The *A Mode is used to 1) determine the number of locations allocated to Input, Intermediate, and Final Storage; 2) repartition this memory; 3) check the number of bytes remaining in program memory; 4) erase Final Storage; and 5) to completely reset the datalogger. When *A is keyed, the first value displayed is the number of memory locations allocated to Input Storage. Press **A** to advance through the memory values. Table 1.5-2 describes what the values seen in the *A Mode represent.

The numbers of memory locations allocated to Input, Intermediate and Final Storage default at power-up to the values in Table 1.5-1.

The sizes of Input and Intermediate Storage may be altered by keying in the desired value and entering it by keying **A**. The size of Final Storage will be adjusted automatically.

One input or Intermediate Storage location can be exchanged for two Final Storage locations and vice-versa. Input and Intermediate Storage must reside in the CPU board RAM. If additional memory boards are present, it is possible to use all of the CPU board RAM for Input and Intermediate Storage. A minimum 32 Input and 768 Final Storage locations will ALWAYS be retained. If no Intermediate Storage is required, its size may be reduced to 0.

All data in Intermediate and Final Storage are erased when memory is repartitioned. This feature may be used to clear memory without altering programming. The number of locations does not actually need to be changed; the same value can be keyed in and entered. After repartitioning memory, the Tables must be recompiled. Recompiling with *0 erases Input Storage; recompiling with *6 leaves Input Storage unaltered.

If Intermediate Storage size is too small to accommodate the programs or instructions entered, the program will not compile and the "E:04" ERROR CODE will be displayed; the size of Intermediate Storage must be increased before the program will compile. Final Storage size can be maximized by limiting Intermediate Storage size to the minimum number of memory locations necessary to accommodate the programs entered. The number of Final Storage locations and the rate at which data are stored determines how long it will take for Final Storage to fill, at which point new data will write over old.

## 1.6 MEMORY TESTING AND SYSTEM STATUS - *B

The *B Mode is used to 1) read the signature of the program memory and the software PROMs, 2) display the power-up memory status, 3) display the number of E08 occurrences (Section 3.10), 4) display the number of overrun occurrences (Section 1.1.1), and 5) display PROM version and revision number. Table 1.6-

## SECTION 1. FUNCTIONAL MODES

1 describes what the values seen in the *B Mode represent. The correct signatures of the CR7 PROMs are listed in Appendix B.

A signature is a number which is a function of the data and the sequence of data in memory. It is derived using an algorithm which assures a 99.998% probability that if either the data or its sequence changes, the signature changes. The signature of the program memory is used to determine if the program tables have been altered. During the self check on power-up, the signature computed for a PROM is compared with a signature stored in the PROM to determine if a failure has occurred. The algorithm used to calculate the signature is described in Appendix C.

The contents of windows 8 and 9, PROM version and version revision, are helpful in determining what PROM is in the datalogger. Over the years, several different PROM versions have been released, each with operational differences. When calling Campbell Scientific for datalogger assistance, please have these two numbers available.

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

| Key Entry | Display ID: Data | Description of Data |
|---|---|---|
| *B | 01: XXXXX | Program memory Signature. The value is dependent upon the programming entered and memory allotment. If the Tables have not been previously compiled, they will be compiled and run. |
| A | 02: XXXXX | First PROM Signature |
| | 03: XXXXX | Second PROM Signature |
| | 04: XXXXX | Third PROM Signature |
| A | 05: XXXXX | Memory status, No. K RAM and ROM |
| A | 06: XXXXX | No. of E08 occurrences (Key in 88 to reset) |
| A | 07: XXXXX | No. of overrun occurrences (Key in 88 to reset) |
| A | 08: X.XXXX | PROM version number |
| A | 09: XXXX. | Version revision number |
| A | 01:00 | Enter I/O Module No. to test (usually 1) |
| 1A | 01:XXXXX | I/O Module 1 RAM Signature |
| | 01:XXXXX | I/O Module 1 PROM Signature |

## 1.7 *C MODE -- SECURITY

The *C Mode is used to secure the user's program information. If security is activated, then the CR7 will block keyboard access to the *1, *2, *3, *4, and *A Modes. Activated security will also block Telecommunications access to the *1, *2, *3, *4, *5, and *D Modes and the Telecommunications C command. A four digit password allows entry to the *C Mode and becomes part of the program memory, affecting the program signature. If security is enabled when *C is keyed, the password must be keyed in before one can advance to window 1. If security is disabled, keying *C brings up window 1 immediately. In window 1 a command can be entered to either enter a new password (1), or temporarily disable security (00) in order to check or alter the programming. The password on power-up is 0000 (unless *D was used to create a custom PROM with the password built in), which disables security. When security is temporarily disabled, it is possible to enter all modes and to alter programming. Keying *0 or *6 will automatically re-enable security, unless the password is 0000.

Entering the four digit password as an indexed value (i.e. xxxx--, entered by keying C after entering the four digits) blocks access to the *1, *2, *3, *4, and *A Modes, but allows the user to view and change the password.

## TABLE 1.7-1. *C Mode Entries and Codes

| Key Entry | Display ID: Data | Description |
|---|---|---|
| *C | 12:0000 | Enter current password. If correct, then advance, else exit *C Mode. 12:00 indicates *C Mode is not in PROMs. If security is disabled, *C advances directly to window 1. |
| A | 01:00 | Window 1, enter command:<br>00 = disable security and advance to window 2; subsequent *0 or *6 enables security.<br>01 = security remains enabled, but it advances to window 2 and allows entry of a new password. |
| A | 02:XXXX | Set new password (XXXX is current password). |
| A | | Returns to window 1. Entering 0000 disables security (window 1 must be set to 0). |

## 1.8 *D MODE -- SAVE OR LOAD PROGRAM

The *D Mode allows the user's program information in the *1, *2, *3, *4, *A, *C (if OSX-0), and *B Modes to be output to or loaded from printer/computer (ASCII) or SM192/716 Storage Module. Table execution and on-line printer outputs are suspended while in the *D Mode. When *D is keyed, the CR7 will display "13:00".

## TABLE 1.8-1. *D Mode Commands

| Command | Description |
|---|---|
| 1 | Save ASCII Program |
| 2 | Load ASCII Program |
| 71 | Save/Load/Clear Program from Storage Module |

A command is entered by keying the command number and **A**. When Command 1, 2, or 71 is entered, the command number is displayed in the ID field. The user must then key in a baud rate code for command 1 or 2 or the command code for the Storage Module (Table 1.8-2). After the code is keyed in, key **A** to execute the command. After a command is executed, "13:0000" is displayed; *D must be entered again before another command can be given.

If the CR7 program has not been compiled when a command to save the program is entered, it will be compiled before the command is executed.

## TABLE 1.8-2. *D Mode Baud Rate and Storage Module Codes

| BAUD RATE CODES | STORAGE MODULE COMMAND CODES |
|---|---|
| 0 - 300 baud<br>1 - 1200<br>2 - 9600<br>3 - 76,800 | 1X Save Program X to Storage Module (X=1-8)<br>2X Load Program X from Storage Module (X=1-8)<br>3X Erase Program X from Storage Module (X=1-8) |

# SECTION 1. FUNCTIONAL MODES

All data in Input, Intermediate and Final Storage are erased when a command to load a program is executed or when a program is written to tape.

If nothing is received within 30-40 seconds after giving the command to load a program, the command will be aborted and an error code displayed (E99 for Storage Module or ASCII). Commands 1, 2, and 71 are the only commands that can be executed via telecommunications (Section 5). For commands 1 and 2, the CR7 will use the baud rate already established in telecommunications and will be ready to receive or send the file as soon as the command is received.

---

### TABLE 1.8-3. Program Load Error Codes

| | |
|---|---|
| E 98 | Uncorrectable errors detected |
| E 99 | Wrong type of file or no data received |

## 1.8.1 TRANSFER TO COMPUTER/PRINTER

This section describes commands 1 and 2 (Table 1.8-1). The PC208 Software automatically uses these commands for uploading and downloading programs.

SENDING ASCII PROGRAM INFORMATION

Command 1 is to send the program listing in ASCII. At the end of the listing, the CR7 sends control E (5 hex or decimal) twice. Except when in telecommunications, the baud rate code must be entered after command 1.

Table 1.8-4 is an example of the program listing sent in response to command 1 (the actual listing is in one column but is printed in two columns to save space). Note that the listing uses numbers for each mode: The numbers for *A, *B, and *C modes are 10, 11, and 12, respectively.

---

### TABLE 1.8-4. Example Program Listing From *D Command 1

```
MODE 1
SCAN RATE 2
1:P17
1:1
1:P0

2:P14
1:1
2:1
3:5
4:1
5:1
6:2
7:1
8:0

3:P92
1:0
2:5
3:10

MODE 1
SCAN RATE 2

4:P71
1:2
2:1

5:P0

MODE 2
SCAN RATE 0

MODE 3

MODE 4
1:0
2:0

MODE 10
1:28
2:64
3:19328
4:934

MODE 12
1:0
2:0

MODE 2
SCAN RATE 0
```

## LOAD PROGRAM FROM ASCII FILE

Command 2 sets up the CR7 to load a serial ASCII program. The format is the same as sent in response to command 1 (Table 1.8.4). Except when in telecommunications, the baud rate code must be entered after command 2.

A download file need not follow exactly the same format that is used when listing a program (i.e., some of the characters sent in the listing are not really used when a program is loaded). Some rules which must be followed are:

1. "M" must be the first character other than a carriage return (CR) or line feed (LF). The "M" serves the same function as "*" does from the keyboard. The order that the Modes are sent in does not matter (i.e., the information for Mode 4 could be sent before that for Mode 1).

2. "S" is necessary prior to the execution interval (Scan rate).

3. The colons (:) are used to mark the start of actual data.

4. A semicolon (;) tells the CR7 to ignore the rest of the line and can be used after an entry so that a comment can be added.

There are 4 two-character control codes which may be used to verify that the CR7 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) | Exit and compile program |

As a download file is received, the CR7 buffers the data in memory; the data is not loaded into the editor or compiled until the CR7 receives a command to do so. The minimum file that could be sent is the program listing, then **^E^E**. **^C^C** tells the CR7 to send the signature (Section C.3) for the current buffer of data. If this signature does not match that calculated by the sending device, **^B^B** can be sent to discard the current buffer and reset the signature. If the signature is correct, **^D^D** can be sent to tell the CR7 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.

## 1.8.2 PROGRAM TRANSFER WITH STORAGE MODULE

The SM192/716 Storage Module must be connected to the CR7. Key **\*D**, then enter command 71. The command to save, load, or clear a program and the program number (Table 1.8-2) is entered. After the operation is finished, "13:0000" is displayed.

The datalogger can be programmed on power-up using a Storage Module. Storage Modules can store up to eight separate programs. If a program is stored as program number 8, and if the Storage Module is connected to the datalogger serial port at power-up, program number 8 is downloaded to the datalogger and compiled.

# SECTION 1. FUNCTIONAL MODES

# SECTION 2. INTERNAL DATA STORAGE

## 2.1 FINAL STORAGE AREAS, OUTPUT ARRAYS, AND MEMORY POINTERS

Final Storage is that portion of memory where final, processed data are stored. Data must be sent to Final Storage before they can be transferred to a computer or external storage peripheral.

The size of Final Storage is expressed in terms of memory locations or bytes. A low resolution data point (4 decimal characters) occupies one memory location (2 bytes), whereas a high resolution data point (5 decimal characters) requires two memory locations (4 bytes). Table 1.5-1 shows the default allocation of memory locations to Input, Intermediate, and Final Storage. The *A Mode is used to reallocate memory or erase Final Storage (Section 1.5). A minimum of 768 memory locations will ALWAYS be retained in Final Storage.

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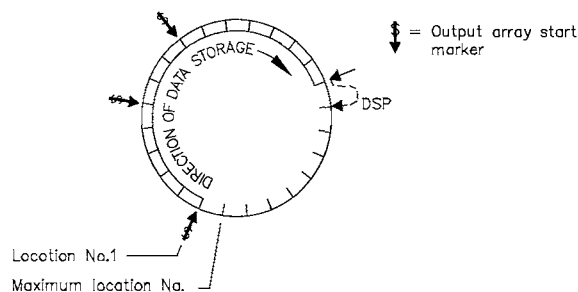


### FIGURE 2.1-1. Ring Memory Representation of Final Data Storage

Output Processing Instructions store data into Final Storage only when the Output Flag is set high. The string of data stored each time the Output Flag is set high is called an **output array**. The first data point in the output array is a 4 digit **Output Array ID**. This ID number is set in one of two ways:

1) In the default condition, the ID consists of the program table number and the Instruction Location Number of the instruction which set the Output Flag for

that output array. For example, the ID of 118 in Figure 2.1-2 indicates that the 18th instruction in Table 1 set the Output Flag high.
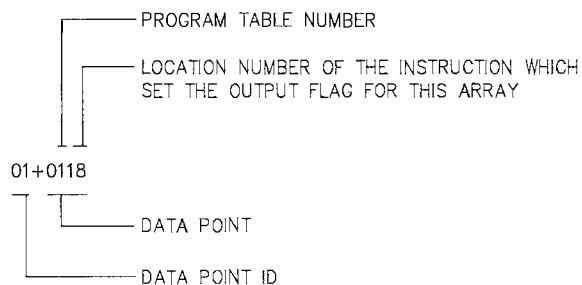


### FIGURE 2.1-2. Output Array ID

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

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

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

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

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

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

The **PPTR** is used to control data transmission to a printer, Storage Module, or other serial device. Whenever on-line printer transfer is activated (*4 Mode or Instruction 96), data between the PPTR and DSP are transmitted.

When on-line transfer to a SM192/716 Storage Module is activated by Instruction 96 with output code 30, data is transmitted each time an output array is stored in Final Storage IF THE STORAGE MODULE IS CONNECTED TO THE CR7. If the Storage Module is not connected, the CR7 does not transmit the data nor does it advance the PPTR to the new DSP location. It saves the data until the Storage Module is connected. Then, during the next execution of Instruction 96, the CR7 outputs all of the data between the PPTR and the DSP and updates the PPTR to the DSP location (Section 4.1)

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

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

## 2.2 DATA OUTPUT FORMAT AND RANGE LIMITS

Data are stored internally in Campbell Scientific's Final Storage Format (Appendix C.2). Data may be sent to Final Storage in either LOW RESOLUTION or HIGH RESOLUTION format. Low resolution is the default. To change the resolution, Instruction 78 (Section 11) must precede the Output Instructions in the program table.

### 2.2.1 RESOLUTION AND RANGE LIMITS

Low resolution data is a 2 byte format with 3 or 4 significant digits and a maximum magnitude of ±6999. High resolution is a 4 byte format with 5 significant digits and a maximum possible output value of ±99999 (see Table 2.2-1 below).

**TABLE 2.2-1. Resolution Range Limits of CR7 Data**

| Resolution | Zero | Minimum Magnitude | Maximum Magnitude |
|---|---|---|---|
| Low | 0.000 | ±0.001 | ±6999. |
| High | 0.0000 | ±.00001 | ±99999. |

The resolution of the low resolution format is reduced to 3 significant digits when the first (left most) digit is 7 or greater (Table 2.2-2). Thus, it may be necessary to use high resolution output or an offset to maintain the desired resolution of a measurement. For example, if water level is to be measured and output to the nearest 0.01 foot, the level must be less than 70 feet for low resolution output to display the 0.01 foot increment. If the water level is expected to range from 50 to 80 feet the data could either be output in high resolution or could be offset by 20 feet (transforming the range to 30 to 60 feet).

#### TABLE 2.2-2. Decimal Location in Low Resolution Format

| Absolute Value | Decimal Location |
|---|---|
| 0 - 6.999 | X.XXX |
| 7 - 69.99 | XX.XX |
| 70 - 699.9 | XXX.X |
| 700 - 6999. | XXXX. |

While output data have the limits described above, the computations performed in the CR7 are done in floating point arithmetic. Values are rounded when converting to Final Storage Format.

### 2.2.2 INPUT AND INTERMEDIATE STORAGE DATA FORMAT

In Input and Intermediate Storage, numbers are stored and processed in a binary format with a 23 bit binary mantissa and a 6 bit binary exponent. The largest and smallest numbers that can be stored and processed are $9 \times 10^{18}$ and $1 \times 10^{-19}$, respectively. The size of the number determines the resolution of the arithmetic. A rough approximation of the resolution is that it is better than 1 in the seventh digit. For example, the resolution of 97,386,9$\underline{2}$4 is better than 10. The resolution of 0.0086731$\underline{92}$4 is better than 0.000000001.

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

## 2.3 DISPLAYING STORED DATA ON KEYBOARD/DISPLAY - *7 MODE

The *7 Mode is used to display Final Storage data. Enter the Mode by keying *7. The display will show "07:XXXXX", where XXXXX is the Final Storage location (DSP) where the next data will be stored. Two options are available:

1. Press A to advance and display the output array ID of the oldest array in Final Storage.

2. Enter a Final Storage location number. When A is pressed, the DPTR will jump to the location entered and, if it is not at the start of an array, advance to the first start of array. The display will show the Array ID.

Repeated use of the A key advances through the output array, while use of the B key backs the DPTR through memory.

The Final Storage location of the data point being viewed may be displayed by keying #. At this point, another location may be entered followed by A to jump to the start of the output array equal to or just ahead of the location entered. Whenever a location number is displayed by keying #, the corresponding data point can be displayed by keying C. To advance to the start of the next output array, key #A. To back up one output array, key #B.

#### TABLE 2.3-1. *7 Mode Command Summary

| Key | Action |
|---|---|
| A | Advance to next data point |
| B | Back-up to previous data point |
| # | Display location number of currently displayed data point value |
| C | Display value of current location |
| #A | Advance to start of next output array |
| #B | Back-up to previous output array |

# SECTION 2. INTERNAL DATA STORAGE

# SECTION 3. INSTRUCTION SET BASICS

*The instructions used to program the CR7 are divided into four types: Input/Output (I/O), Processing, Output Processing, and Program Control. I/O Instructions are used to make measurements and store the readings in input locations or to initiate analog or digital port output. Processing Instructions perform numerical operations using data from Input Storage locations and place the results back into specified Input Storage locations. Output Processing Instructions provide a method for generating time or event dependent data summaries from processed sensor readings residing in specified Input Storage locations. Program Control Instructions are used to direct program execution based on time and/or conditional tests on input data and to direct output to external devices.*

*Instructions are identified by a number. Each instruction has a number of parameters which give the CR7 the information it needs to execute the instruction.*

*The set of instructions available in the CR7 is determined by the Programmable Read Only Memory chips (PROMS) that are installed. Appendix B lists the software options available.*

## 3.1 PARAMETER DATA TYPES

There are three 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 (Sections 9-12), the parameter data type is identified by its abbreviation. Different data types are used to allow the CR7 to make the most efficient use of its memory.

Floating Point parameters are used to enter numeric constants for calibrations or arithmetic operations. While it is only possible to enter five digits (magnitude $\pm.00001$ to $\pm99999.$), the internal format has a much greater range ($1\times10^{-19}$ to $9\times10^{18}$, Section 2.2.1).

## 3.2 REPETITIONS/CARD NUMBER

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. Separate parameters are used to specify the card and input channel on which to make the first measurement. For example, if you have four 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 4 times, enter it once with four repetitions. The instruction will make four measurements starting on the specified channel number and continuing through the three succeeding differential channels, with the results being stored in the specified input location and the three succeeding input

locations. Averages for all four measurements can be calculated by entering the Average Instruction with four repetitions.

The CR7 will automatically continue repetitions from the last channel of one card to the first channel of the next sequentially numbered 723(-T) Analog Input Card or 725 Pulse Counter Card. Measurements on the 726 50 volt Analog Input Card will not advance correctly from one card to the next; enter separate measurement instructions for each card.

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

## 3.3 ENTERING NEGATIVE NUMBERS

After keying in a number, press **C** or "-" to change the number's sign. On floating point numbers, a minus sign (-) will appear to the left of the number.

Excitation voltages in millivolts for I/O Instructions are 4 digit integers; when **C** is pressed, minus signs (-) will appear to the right of the number indicating a negative excitation.

Even though this display is the same as that indicating an indexed input location, (Section 3.4) there is no indexing effect on excitation voltage.

## 3.4 INDEXING INPUT LOCATIONS

When used within a Loop, the parameters for input locations can be Indexed to the loop counter. The loop counter is added to the indexed value to determine the actual input location the instruction acts on. Normally, the loop counter is incremented by one after each pass through the loop. Instruction 90, Step Loop Index, allows the increment step to be changed. See Instructions 87 and 90, Section 12, for more details.

To index an input location (4 digit integer), key C after keying the value but before entering the parameter. Two minus signs (-) will be displayed to the right of the parameter.

## 3.5 VOLTAGE RANGE AND OVERRANGE DETECTION

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

Select the smallest full scale range that is greater than or equal to the full scale output of the sensor being measured. Using the smallest possible range will result in the best resolution for the measurement.

Two different integration sequences are possible. The slow integration, 16.67 milliseconds, is one 60 Hz cycle and rejects noise from 60 Hz AC line power as well as having better rejection of random noise than the fast integration. A PROM with 50Hz rejection is available for countries whose electric utilities operate at 50 Hz (Appendix B).

When a voltage input exceeds the range programmed, the value stored is the maximum negative number, displayed in the *6 Mode as -99999. In output data from Final Storage, this becomes -6999 in low resolution or -99999. in high resolution.

An input voltage greater than +8 volts on one of the analog inputs will result in errors and possible overranging on the other analog inputs. Voltages greater than 16 volts may permanently damage the CR7.

### TABLE 3.5-1. Input Voltage Ranges and Codes

| Range Code | | Full Scale Range | | Resolution* | |
|---|---|---|---|---|---|
| Slow 16.67ms Integ. | Fast 250µs Integ. | | | | |
| 1 | 11 | ±1500 | microvolts | 50 | nanovolts |
| 2 | 12 | ±5000 | microvolts | 166 | nanovolts |
| 3 | 13 | ±15 | millivolts | 500 | nanovolts |
| 4 | 14 | ±50 | millivolts | 1.66 | microvolts |
| 5 | 15 | ±150 | millivolts | 5 | microvolts |
| 6 | 16 | ±500 | millivolts | 16.6 | microvolts |
| 7 | 17 | ±1500 | millivolts | 50 | microvolts |
| 8 | 18 | ±5000 | millivolts | 166 | microvolts |

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

## 3.6 OUTPUT PROCESSING

Most Output Processing Instructions require both an intermediate processing operation and a final processing operation. For example, when the Average Instruction, 71, is executed, the intermediate processing operation increments a sample count and adds each new Input Storage value to a cumulative total residing in Intermediate Storage. When the Output Flag is set, the final processing operation divides the total by the number of

sample counts, stores the resulting average in Final Storage and zeros the value in Intermediate Storage so that the process starts over with the next execution.

Final Storage is the default destination of data output by Output Processing Instructions (Sections OV2, 1.5, 2.1). Instruction 80 may be used to direct output to Input Storage or to Final Storage.

Output Processing Instructions requiring intermediate processing sample the specified input location(s) each time the Output Instruction is executed, NOT necessarily each time the location value is updated by an I/O Instruction. For example: Suppose a temperature measurement is initiated by Table 1 which has an execution interval of one second. The instructions to output the average temperature every 10 minutes are in Table 2 which has an execution interval of 10 seconds. The temperature will be measured 600 times in the 10 minute period, but the average will be the result of only 60 of those measurements because the instruction to average is executed only one tenth as often as the instruction to make the measurement.

Final processing occurs only when the Output Flag is set (Section 3.7.1). The Output Flag, Flag 0, is set at desired intervals or in response to specified conditions by using an appropriate Program Control Instruction (Section 11).

## 3.7 USE OF FLAGS: OUTPUT AND PROGRAM CONTROL

There are 10 flags which may be used in CR7 programs. Two of the flags have functions with Output Processing Instructions: Flag 0 controls final processing and data storage, and Flag 9 can disable intermediate processing. Flags 1-8 may be used as desired in programming the CR7. Flags 0 and 9 are automatically set low at the beginning of the program table. Flags 1-8 remain unchanged until acted on by a Program Control Instruction or until manually toggled from the *6 Mode.

---

**TABLE 3.7-1. Flag Description**

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

---

Flags are set with Program Control Instructions. The Output Flag, Flag 0, and the intermediate processing disable flag, Flag 9, will always be set low if the set high condition is not met. The status of flags 1-8 are not changed if a conditional test is false.

### 3.7.1 THE OUTPUT FLAG

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

Each group of Output Processing Instructions creating an output array must be preceded by a Program Control Instruction that sets the Output Flag.

Output is most often desired at fixed intervals; this is accomplished with Instruction 92, If Time, which checks the clock to see if it is X minutes into a Y minute interval. If the time condition is met, a command is executed. Output at the beginning of the interval by making Parameter 1, time into the interval, 0. Parameter 2, the time interval in minutes, is how often output will occur; i.e., the Output Interval. Enter 10 for parameter 3, the command code, to set Flag 0 high. Instruction 92 is followed in the program table by the Output Instructions which define the output array desired.

The time interval is synchronized to 24 hour time; output will occur on each integer multiple of the Output Interval starting from midnight (0 minutes). If the Output Interval is not an even divisor of 1440 minutes (24 hours), the last output interval of the day will be less than the specified time interval. Output will occur at midnight and will resume synchronized to the new day.

> **NOTE:** If the Output Flag is already set high and the test condition of a subsequent Program Control Instruction acting on the flag fails, the flag is set low. This feature eliminates having to enter another instruction to specifically reset the Output Flag at the end of an output array before proceeding to another group of Output Instructions with a different output interval (see example in OV4.3).

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

## 3.7.2 THE INTERMEDIATE PROCESSING DISABLE FLAG

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

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

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

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

## 3.7.3 USER FLAGS

Flags 1-8 are not dedicated to a specific purpose and are available to the user for general programming needs. The user flags can be manually toggled from the keyboard in the *6 Mode (Section 1.3) or from a computer using TERM's monitor feature. By inserting flag tests (Instruction 91) at appropriate points in the program, the user can manually set flags to direct program execution.

## 3.8 PROGRAM CONTROL LOGICAL CONSTRUCTIONS

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

### TABLE 3.8-1. Command Codes

| | | |
|---|---|---|
| 0 | - | Go to end of program table |
| 1-9, 79-99 | - | Call Subroutine 1-9, 79-99 |
| 10-19 | - | Set Flag 0-9 high |
| 20-29 | - | Set Flag 0-9 low |
| 30 | - | Then Do |
| 31 | - | Exit loop if true |
| 32 | - | Exit loop if false |
| 41-48 | - | Set port 1 - 8 high* |
| 51-58 | - | Set port 1 - 8 low* |
| 61-68 | - | Toggle port 1 - 8* |
| 71-78 | - | Pulse port 1 - 8* 100 ms |

\* Port commands default to Excitation Card 1; Instruction 20 is used to change to another card.

### 3.8.1 IF THEN/ELSE COMPARISONS

When Command 30, THEN DO, is used with one of the IF Instructions, 88-92, the instruction is followed immediately by instructions to

execute if the comparison is true. The Else Instruction, 94, is optional and is followed by the instructions to execute if the comparison is false. The End Instruction, 95, marks the end of the branching started by the IF Instruction. Subsequent instructions are executed regardless of the outcome of the comparison (Figure 3.8-1).
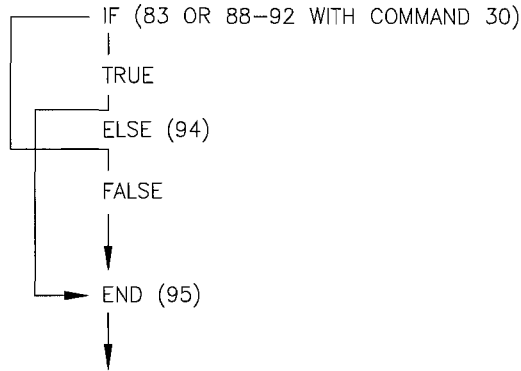
```
     ┌─────── IF  (83 OR 88-92 WITH COMMAND 30)
     │        │
     │        TRUE
     │     ┌──┘
     │  ┌──┤  ELSE (94)
     └──┤  ┌─┘
        │  │  FALSE
        │  │  │
        │  │  ▼
        └──┼─► END (95)
           │
           ▼
```

**FIGURE 3.8-1. If Then/Else Execution Sequence**

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

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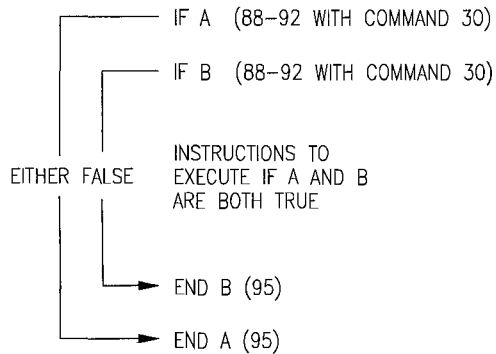
**FIGURE 3.8-2. Logical AND Construction**

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

---

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

---

Call subroutine X (86, command=X)
END B (95)
END A (95)

---
**FIGURE 3.8-3. Logical OR Construction**

---

A logical OR can also be constructed by setting a flag if a comparison is true. (The flag is 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; when the next Instruction 83 is encountered, execution branches to the END Instruction 95 which closes the case test (see Instruction 93).

## 3.8.2 END, INSTRUCTION 95

END, Instruction 95, is required to mark the end of:

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

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, the CR7 will display error 22 when compiling the program. Error 21 is displayed if END is used without being preceded by one of these instructions (Section 3.10).

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 3.8-2). The lines must not cross. To debug logic or find a missing or extra END error, list the program and draw the lines.

# SECTION 3. INSTRUCTION SET BASICS

Subroutines can be called from other subroutines; they cannot be embedded within other subroutines. A subroutine must end before another subroutine begins (Error 20). Any loops or IF...THEN DO sequences started within a subroutine must end before the subroutine.

## 3.8.3 NESTING

A branching or loop instruction which occurs before a previous branch or loop has been closed with the END instruction is nested. The maximum nesting level is 9 deep. Error 30 is displayed when attempting to compile a program which is nested too deep.

The Loop Instruction, 87, counts as 1 level. Instructions 86, 88, 89, 91, and 92 each count as one level when used with the THEN DO command 30. 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. Branching and loop nesting

starts at zero within each subroutine and then returns to the previous level after returning from the subroutine.

Subroutine calls do not count as nesting with the above instructions. They have a separate nesting limit of seven (Instruction 85, Section 12).

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

## 3.9 INSTRUCTION MEMORY AND EXECUTION TIME

The standard CR7 has 1744 bytes of program memory available for the programs entered in the *1, *2, and *3 program tables. Each instruction also makes use of varying numbers of Input, Intermediate, and Final Storage locations. The following tables list the memory used by each instruction and the approximate time required to execute the instruction.

---

**TABLE 3.9-1. Input/Output Instruction Memory**

R = No. of Reps.
D = Delay

| INSTRUCTION | | MEMORY | | EXECUTION TIME (ms) | | |
| --- | --- | --- | --- | --- | --- | --- |
| | | INPUT LOC. | PROG. BYTES | Slow or No Integration | | Fast Integration |
| 1 | VOLT (SE) | R | 15 | 57.4 + 22R | | 16 + 2.9R |
| 2 | VOLT (DIFF) | R | 15 | 54 + 43.4R | | 19 + 4.7R |
| 3 | PULSE | R | 15 | 4 + 2R | | |
| 4 | EX-DEL-SE | R | 20 | 56.8 + (22.6 + D)R | | 23.4 + (3.3 + D)R |
| 5 | AC HALF BR | R | 18 | 57.7 + 44R | | 21.1 + 5.5R |
| 6 | FULL BR | R | 18 | 58 + 87.3R | | 24.2 + 9.6R |
| 7 | 3W HALF BR | R | 18 | 58.8 + 88.7R | | 24.3 + 11.7R |
| 9 | FULL BR-MEX | R | 19 | 104 + 175R | | 31.5 + 20.4R |
| 10 | BATT. VOLT | 1 | 4 | 22.6 | | |
| 11 | TEMP (107) | R | 15 | 23 + 5.4R | | |
| 12 | RH (207) | R | 17 | 23.3 + 5.4R | | |
| 13 | TEMP-TC SE | R | 18 | 59.8 + 21.9R | | 25.2 + 6.1R |
| 14 | TEMP-TC DIF | R | 18 | 61 + 43.2R | | 21.5 + 7.85R |
| 16 | TEMP-RTD | R | 15 | 0.4 + 2.7R | | |
| 17 | TEMP-INTERNL | 1 | 4 | 116.2 | | |
| 18 | TIME | 1 | 7 | 1.4 | | |
| 19 | SIGNATURE | 1 | 4 | 607.2 | | |
| 20 | PORT SET | 1 | 4 | 2.9 | | |
| 21 | ANALOG OUT | 1 | 5 | 3.6 | | |
| 22 | EXCIT-DEL | 1 | 11 | 10.8 + D | | |
| 23 | SELECT I/O MODULE | | | 0.4 | | |
| 26 | TIMER | 1 or 0 | 4 | 0.54 to reset, 0.25 to load into location | | |

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

| INSTRUCTION | | INPUT LOC. | MEMORY INTER. LOC. | PROG. BYTES | EXECUTION TIME (ms) |
|---|---|---|---|---|---|
| 30 | Z=F | 1 | 0 | 8 | 0.3 |
| 31 | Z=X | 1 | 0 | 6 | 0.5 |
| 32 | Z=Z+1 | 1 | 0 | 4 | 0.6 |
| 33 | Z=X+Y | 1 | 0 | 8 | 1.1 |
| 34 | Z=X+F | 1 | 0 | 10 | 0.9 |
| 35 | Z=X-Y | 1 | 0 | 8 | 1.1 |
| 36 | Z=X*Y | 1 | 0 | 8 | 1.2 |
| 37 | Z=X*F | 1 | 0 | 10 | 0.9 |
| 38 | Z=X/Y | 1 | 0 | 8 | 2.7 |
| 39 | Z=SQRT(X) | 1 | 0 | 6 | 12.0 |
| 40 | Z=LN(X) | 1 | 0 | 6 | 7.4 |
| 41 | Z=EXP(X) | 1 | 0 | 6 | 5.9 |
| 42 | Z=1/X | 1 | 0 | 6 | 2.6 |
| 43 | Z=ABS(X) | 1 | 0 | 6 | 0.7 |
| 44 | Z=FRAC(X) | 1 | 0 | 6 | 0.3 |
| 45 | Z=INT(X) | 1 | 0 | 6 | 1.0 |
| 46 | Z=X MOD F | 1 | 0 | 10 | 3.2 |
| 47 | Z=XY | 1 | 0 | 8 | 13.3 |
| 48 | Z=SIN(X) | 1 | 0 | 6 | 6.5 |
| 49 | SPA MAX | 1 or 2 | 0 | 7 | 1.5 + 0.9 (swath-1) |
| 50 | SPA MIN | 1 or 2 | 0 | 7 | 1.7 + 0.9 (swath-1) |
| 51 | SPA AVG | 1 | 0 | 7 | 3.3 + 0.6 (swath-1) |
| 53 | A*X+B | 4 | 0 | 36 | 2.5 + 0.4 scaling pair |
| 54 | BLOCK MOVE | R | 0 | 10 | 0.18 + 0.17R |
| 55 | POLYNOMIAL | R | 0 | 31 | 1.2 + R(2.0 + 0.4 * order) |
| 56 | SAT VP | 1 | 0 | 6 | 4.2 |
| 57 | WDT-VP | 1 | 0 | 10 | 8.1 |
| 58 | LP FILTER | R | R + 1 | 13 | 0.5 + 2.2R |
| 59 | X/(1-X) | 1 | 0 | 9 | 0.4 + 3.0R |
| 61 | INDIR MOVE | 1 | 0 | 6 | 0.35 neither indexed |
| | | | | | 0.54 one location indexed |
| | | | | | 0.73 both locations indexed |
| 66 | ARC TAN | 1 | 0 | 8 | 6.7 |

**TABLE 3.9-3. Output Instruction Memory and Execution Times**
**R = No. of Reps.**

| INSTRUCTION | MEMORY | | | EXECUTION TIME (ms) | |
| | INTER. LOC. | FINAL VALUES[1] | PROG. BYTES | FLAG 0 LOW | FLAG 0 HIGH |
| --- | --- | --- | --- | --- | --- |
| 69 WIND VECTOR | 2+9R | (2, 3, or 4)R | 12 | | |
| | | Options 00, 01, 02 | | 3.5 + 17.5R | 3.5 + 75R |
| | | Options 10, 11, 12 | | 3.5 + 16R | 3.5 + 30R |
| 70 SAMPLE | 0 | R | 5 | 0.1 | 0.4+ 0.6R |
| 71 AVERAGE | 1+R | R | 7 | 0.9+ 0.5R | 2.1+ 3.0R |
| 72 TOTALIZE | R | R | 7 | 0.6+ 0.5R | 1.1+ 1.0R |
| 73 MAXIMIZE | (1 or 2)R | (1,2,or3)R | 8 | 0.9+ 1.7R | 1.3+ 2.8R |
| 74 MINIMIZE | (1 or 2)R | (1,2,or3)R | 8 | 0.9+ 1.7R | 1.3+ 2.8R |
| 75 HISTOGRAM | 1+bins*R | bins*R | 24 | 0.4+ 3.1R | 0.9+ R(3.3+2.8*bins) |
| 77 REAL TIME | 0 | 1 to 4 | 4 | 0.1 | 1.0 |
| 78 RESOLUTION | 0 | 0 | 3 | 0.4 | 0.4 |
| 79 SMPL ON MM | R | R | 7 | 0.3 | 1.1 |
| 80 STORE AREA | 0 | 0 | 5 | | |
| 82 STD. DEV. | 1+3R | R | 7 | | |

[1]Output values may be sent to either Final Storage or Input Storage with Instruction 80.

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

| INSTRUCTION | MEMORY | | EXECUTION TIME (ms) |
| | INTER. LOC. | PROG. BYTES | |
| --- | --- | --- | --- |
| 83 IF CASE <F | 0 | 9 | 0.5 |
| 85 LABEL SUBR | 0 | 3 | 0.0 |
| 86 DO | 0 | 5 | 0.1 |
| 87 LOOP | 1 | 9 | 0.2 |
| 88 IF X<=>Y | 0 | 10 | 0.6 |
| 89 IF X<=>F | 0 | 12 | 0.4 |
| 90 LOOP INDEX | 0 | 3 | 0.5 |
| 91 IF FLAG | 0 | 6 | 0.2 |
| 92 IF TIME | 1 | 11 | 0.3 |
| 93 BEGIN CASE | 1 | 8 | 0.2 |
| 94 ELSE | 0 | 4 | 0.2 |
| 95 END | 0 | 4 | 0.2 |
| 96 SERIAL OUT | 0 | 3 | |
| 98 SEND CHAR. | 0 | 4 | |

## 3.10 ERROR CODES

There are four types of errors flagged by the CR7: Compile, Run Time, Editor, and *D Mode. When an error is detected, an E is displayed followed by the 2 digit error code.

**Compile errors** are errors in programming which are detected once the program is keyed in and compiled for the first time (*0, *6, or *B Mode entered).

**Run Time errors** are detected while the program is running. Error 31 is the result of a programming error. 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.

Error 8 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. The CR7 keeps track of the number of times (up to 99) that E08 has occurred. The number can be displayed and reset with the Telecommunications A command (Section 5.1).

**Editor errors** are detected as soon as an incorrect value is entered and are displayed immediately.

**\*D Mode errors** indicate problems with saving or loading a program. Only the error code is displayed.

**TABLE 3.10-1. Error Codes**

| Code | Type | Description |
|------|------|-------------|
| 01 | Run Time | I/O Module does not respond |
| 03 | Editor | Program table full |
| 04 | Compile | Intermediate Storage full |
| 08 | Run Time | CR7 reset by watchdog timer |
| 09 | Run Time | Data sent to unallocated Input Storage |
| 11 | Editor | Attempt to allocate more Input or Intermediate Storage than is available |
| 20 | Compile | SUBROUTINE encountered before END of previous subroutine |
| 21 | Compile | END without IF, LOOP or SUBROUTINE |
| 22 | Compile | Missing END, nonexistent SUBROUTINE |
| 24 | Compile | ELSE in SUBROUTINE without IF |
| 25 | Compile | ELSE without IF |
| 26 | Compile | EXIT LOOP without LOOP |
| 30 | Compile | IF and/or LOOP nested too deep |
| 31 | Run Time | SUBROUTINES nested too deep |
| 40 | Compile | Table 2 Execution interval too short |
| 40 | Editor | Instruction not in PROM |
| 60 | Compile | Inadequate Input Storage for FFT |
| 61 | Compile | Burst Mode Scan Rate too short |
| 97 | *D MODE | Tape data not received within 30 seconds |
| 98 | *D MODE | Uncorrectable errors detected |
| 99 | *D MODE | Wrong file type, editor error or program not received |

# SECTION 4. EXTERNAL STORAGE PERIPHERALS

*External data storage devices are used to provide a data transfer medium that the user can carry from the test site to the lab and to supplement the internal storage capacity of the CR7, allowing longer periods between visits to the site. The standard data storage peripherals for the CR7 are the Storage Modules (Section 4.4). Output to a printer or related device is also possible (Section 4.5). These peripherals are connected to the CR7 through the 9 pin serial connector.*

*Data output to a peripheral device can take place ON-LINE (automatically, as part of the CR7's routine operation) or it can be MANUALLY INITIATED. On-line data transfer is accomplished with Instruction 96 or with the *4 Mode (Section 4.1). Manual initiation is done in the *8 or *9 Modes (Section 4.2). Regardless of the method, the source of any data transferred is Final Storage.*

*A modem is another type of peripheral that can be connected to the CR7. Communication via a modem (Telecommunications) is discussed in Section 5.*

*The CR7 can output data to multiple peripherals (i.e., a modem and Storage Module could be connected at the same time). However, only one modem may be connected to the CR7 at any one time. It is possible to connect two Storage Modules, although it is seldom necessary.*

*The CR7 can tell whether or not a SM192 or SM716 Storage Module is present. When Instruction 96 or *9 is used to send data to one of these Storage Modules, the CR7 will not send data if the Storage Module is not connected (Section 4.4.2).*

## 4.1 ON-LINE DATA TRANSFER - INSTRUCTION 96, *4 MODE

On-line data transfer is accomplished with Instruction 96 entered in the datalogger program. The *4 Mode is retained from earlier software to maintain compatibility with existing programs. Use only one method to enable output. If using Instruction 96, do not enable output in the *4 Mode.

### 4.1.1 INSTRUCTION 96

Instruction 96 enables output to external storage peripherals under program control. This instruction must be included in the datalogger program for on-line data transfer to take place. Instruction 96 needs to be included only once in the program tables and should follow the Output Processing Instructions. The suggested programming sequence is:

1. Set the Output Flag.
2. If you wish to set the output array ID, enter Instruction 80 (Section 11).
3. Enter the appropriate Output Processing Instructions.

4. Enter Instruction 96 to enable the on-line transfer of Final Storage data to the specified device. If outputting to both tape and a Storage Module or printer option, Instruction 96 must be entered twice.

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

---

### TABLE 4.1-1. Output Device Codes for Instruction 96

**CODE   DEVICE**

| CODE | DEVICE |
|------|--------|
| 1x | Printer, Printable ASCII |
| 2x | Printer, Binary |
| 30 | SM192/716 Storage Module |
| 31 | Send filemark to SM192/716 |

x = BAUD RATE CODE

| | |
|---|---|
| 0 | 300 |
| 1 | 1200 |
| 2 | 9600 |
| 3 | 76,800 |

---

Only one of the options 1x, 2x, or 30 may be used in a program. If using a SM64 Storage Module, output code 21 should be used. Use of the SM192/716 is discussed further in Section 4.4, print output formats are discussed in Section 4.5.

### 4.1.2 *4 MODE

The *4 Mode may be used in place of Instruction 96 to enable or disable printer output and to set the printer baud rate. The first parameter is a two digit number determining the printer status. The second is the baud rate code. To enter a different status, key in the appropriate code from Table 4.1-2, followed by "A". **Printer data is sent in the printable ASCII format only (Section 4.5).** If printer status is changed during execution of the program tables, execution stops until the programs are recompiled. Instruction 96 should be used to send data to the SM192/716 Storage Modules. Do not use *4 if Instruction 96 is used in the program.

### TABLE 4.1-2. *4 Mode Parameters and Codes

| Keyboard Entry | Display ID: Data | Description of Data |
|---|---|---|
| *4 | 04:00 | |
| A | 01:XX | Output Enable Code |
| A | 02:XX | Baud Rate Code |

**Output Enable Codes**

| Code | Description |
|---|---|
| 00 | printer disabled |
| 01 | printer enabled, ASCII |

**Baud Rate Codes**

| Code | Baud Rate |
|---|---|
| 00 | 300 |
| 01 | 1200 |
| 02 | 9600 |
| 03 | 76,800 |

## 4.2 MANUALLY INITIATED DATA OUTPUT - *9 MODE

Data may be transferred to tape using the *8 Mode and to printer or Storage Module using the *9 Mode. These Modes allow the user to retrieve a specific block of data, on demand, regardless of whether or not the CR7 is programmed for on-line data output.

If external storage peripherals are not left on-line, the maximum allowable time between visiting the site to retrieve data must be calculated to insure that data placed in Final Storage are not written over before they are collected. In order to make this calculation, users must determine: (1) the size of Final Storage, (2) the number of output arrays being generated, (3) the number of low and/or high resolution data points included in each output array, and (4) the rate at which output arrays are stored in Final Storage. When calculating the number of data points per output array, remember to add 1 overhead data point (2 bytes) per array for the output array ID.

For example, assume that 19,296 locations are assigned to Final Storage (*A Mode), and that 1 output array, containing the Array ID (1 memory location), 9 low resolution data points (9 memory locations) and 5 high resolution data points (10 memory locations) is stored each hour. In addition, an output array with the Array ID and 5 high resolution data points (11 memory locations) is stored daily. This is a total of 491 memory locations per day ((20 x 24) + 11). 19,296 divided by 491 = 39.3 days. Therefore, the CR7 would have to be visited every 39 days to retrieve data, because write-over would begin in the 40th day.

### 4.2.1 MANUAL STORAGE MODULE OR PRINTER DUMP - *9 MODE

Using the *9 Mode, data in Final Storage can be transmitted as ASCII or binary data out the serial port by manually initiating a dump. If on-line printing is enabled with Instruction 96 or the *4 Mode, entering *9 will stop it. On-line printing will be re-enabled if no keyboard entries are made for 3.4 minutes. Return to the *0 Mode when the dump is completed.

When on-line Storage Module or printer transfer is not enabled and the *9 Mode is used to dump new data, the start of dump pointer (PPTR) will remain where it was when the dump was completed or

aborted until the next time the *9 Mode is entered. If the End of Dump location (window 2) is changed while in the *9 Mode, the TPTR will be set to its previous value when the *9 Mode is exited. Changing the program and compiling moves the PPTR to the current DSP location.

> **NOTE:** A printer dump is aborted by keying #.

### TABLE 4.2-2. *9 Mode Entries

| Key | Display<br>ID:DATA | Description |
|---|---|---|
| *9 | 09:00 | Output Code<br>1X Printable ASCII<br>2X Final Storage Format<br>30 SM192/716 Storage Module<br>31 Send File Mark to SM192/716 than send data |
| | | x = Baud Rate Code<br>0    300<br>1   1200<br>2   9600<br>3  76800 |
| A | 01:XXXXX | Start of Dump location, initially the PPTR location, a different location may be keyed in if desired. To dump all data in Final Storage, enter into window #1 a number 1 greater than the End of Dump location. |
| A | 02:XXXXX | End of Dump location, initially the DSP location, a different location may be keyed in if desired. |
| A | 03:00 | Ready to Dump, to initiate dump, key any number then A. While dumping, "09:" will be displayed in the ID field and the location number in the Data field. The location number will stop incrementing when the dump is complete. |

## 4.3 STORAGE MODULE

The Storage Module stores data in battery backed RAM. Backup is provided by an internal lithium battery. The RAM is internal on the SM192/716 and on a PCMCIA card on the CSM1. Operating power is supplied by the CR7 over pin 1 of the CS I/O connector. When power is applied to the Storage Module, a File Mark is placed in the data (if a File Mark is not the last data point already in storage).

The File Mark separates data. For example, if you retrieve data from one CR7, disconnect the Storage Module and connect it to a second CR7; a File Mark is placed in the data. This mark follows the data from the first CR7, but precedes the data from the second.

The SM192 has 192K bytes of RAM storage; the SM716 has 716K bytes. Both can be configured as either ring or fill and stop memory. The size of memory in the CSM1 depends on the PC card used. The CSM1 is always fill and stop.

### 4.3.1 USE OF TWO STORAGE MODULES

It is possible to connect two Storage Modules to the CR7 for on-line storage. One module must be configured as fill and stop and the other as ring memory (see Storage Module operator's manual for configuring information). Data is written to both modules simultaneously. The module configured as fill and stop quits accepting data once it is full while that with the ring memory continues to store new data over old. The Storage Modules must be retrieved before the module configured as ring memory wraps around memory a second time.

### 4.3.2 STORAGE MODULE USE WITH INSTRUCTION 96

When output to the Storage Module is enabled with Instruction 96, the Storage Module(s) may be either left with the CR7 for on-line data transfer and periodically exchanged, or brought to the site for data transfer.

USE OF STORAGE MODULE TO PICK UP DATA

The CR7 can tell when the Storage Module is connected. Each time Instruction 96 is executed and there is data to output, the CR7 checks for the presence of the Storage Module.

If a Storage Module is not connected no data are sent and the Printer Pointer (PPTR, Section 2.1) is not advanced.

When a Storage Module is connected, two things happen:

1. Immediately upon connection, a File Mark is placed in the Storage Module Memory following the last data stored.

2. During the next execution of Instruction 96, the CR7 detects the Storage Module and outputs all data between the PPTR and the DSP location.

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

If the SM is just brought to the site to pick-up data, the SC90 Serial Line Monitor can be used to visually confirm that data were transferred. The SC90 contains an LED which lights during data transmission. When the light goes OFF, data transfer is complete and the SM can be disconnected from the CR7.

### 4.3.3 *9 DUMP TO STORAGE MODULE

In addition to the on-line data output procedures described above, output from CR7 Final Storage to the SM192 and SM716 can be manually initiated in the *9 Mode. The procedure for setting up and transferring data is as follows:

1. Connect the Storage Module to the CR7 using the SC12 cable.

2. Enter the appropriate commands as listed in Table 4.2-2.

## 4.4 PRINTER OUTPUT FORMATS

Printer output can be sent in the binary Final Storage Format (Appendix C.2) or Printable ASCII. If using the *4 Mode to enable on-line output, Printable ASCII is the only format available.

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

Figure 4.5-1 shows both high and low resolution data points in a 12 data point output array. The example data contains Day, Hour-Minute, and Seconds in the 2nd - 4th data points. The output array ID and time values (year, day, hour-minute, and seconds) are always four character numbers, even when high resolution output is specified.

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

DATA POINT ID
OUTPUT ARRAY ID (PRG. TBL. + LOC. NO. OF OUTPUT FLAG ENABLE INST.)
2 SPACES
DAY OF YEAR
TIME
SECONDS
LO RES DATA
HI RES DATA
1 SPACE
LOW RES ZERO
HI RES ZERO
CARRIAGE RETURN, LINE FEED
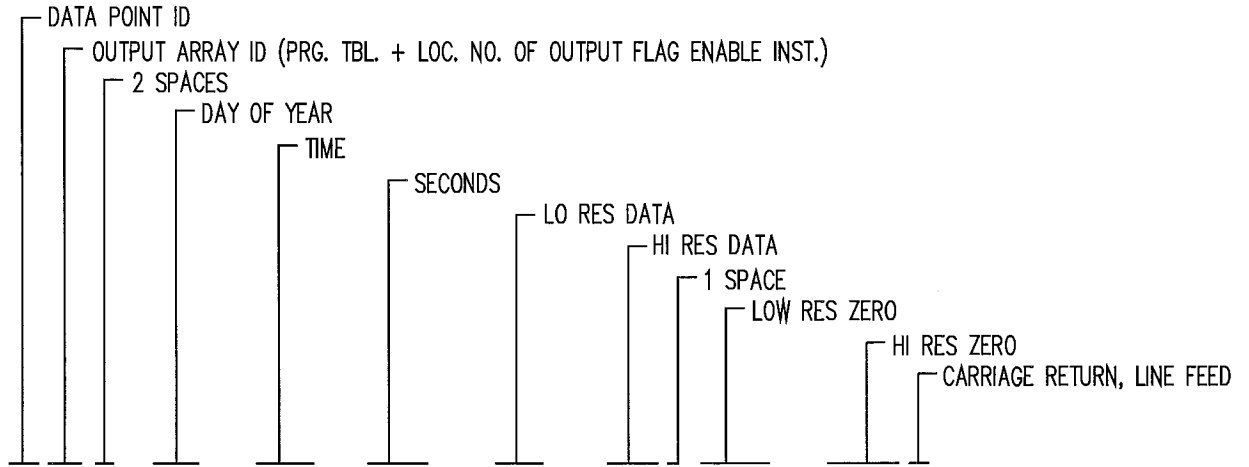
01+0101.  02+0011.  03+2203.  04+0.000  05-0.100  06+2.0410 07+0.000  08+0.0000
09-0.001  10+.00001 11+6999.  12+99999.

CARRIAGE RETURN, LINE FEED
MAX HI RES DATA
MAX LO RES DATA
MIN HI RES DATA
MIN LO RES DATA
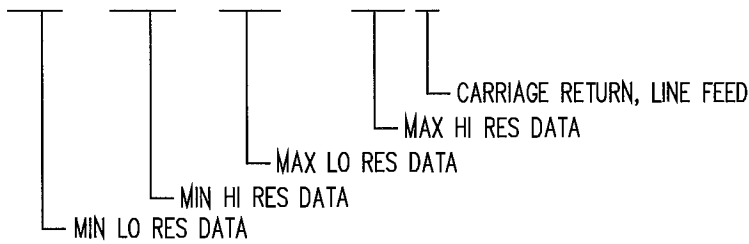
**FIGURE 4.4-1. Example of CR7 Printable ASCII Output Format**

# SECTION 5. TELECOMMUNICATIONS

*Telecommunications allows a computer to retrieve data directly from Final Storage and may be used to program the CR7 and monitor sensor readings in real time. Any user communication with the CR7 that makes use of a computer or terminal instead of the CR7 keyboard is through Telecommunications.*

*Telecommunications can take place over a variety of links including:*

- *telephone*
- *radio frequency*
- *short haul modem*
- *SC32A and ribbon cable*
- *multi-drop interface and coax cable*

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

*Data retrieval can take place in either ASCII or BINARY. The BINARY format is five times more compact than ASCII. The shorter transmission times for binary result in lower long distance telephone charges and more reliable data transfer. On "noisy" links shorter blocks of data are more likely to get through without interruption.*

*In addition to more efficient data transfer, binary data retrieval makes use of a signature for error detection. The signature algorithm assures a 99.998% probability that if either the data or its sequence changes, the signature changes.*

*The PC208 Datalogger Support Software for PCs and compatibles contains the programs which automate data retrieval, program transfer, and real time monitoring. The PC208 package has been designed to meet the most common needs in datalogger support and telecommunications. This section in not intended to furnish sufficient detail to write Telecommunications software. Appendix C contains some details of binary data transfer and Campbell Scientific's binary data format.*

*This section emphasizes the commands that a person would use when manually (i.e., entered by hand) interrogating or programming the CR7 via a computer/terminal. These commands and the responses to them are sent in the American Standard Code for Information Interchange (ASCII). The Remote Keyboard State (Section 5.2) allows the user with a computer/terminal to use the same commands as the CR7 keyboard.*

## 5.1 TELECOMMUNICATIONS COMMANDS

When the CR7 is rung by a modem, it answers (enables the modem) almost immediately. Several carriage returns (CR) must be sent from the computer to allow the CR7 to set its baud rate to that of the modem/terminal (300, 1200, 9600, or 76,800). Once the baud rate is set, the CR7 sends the prompt, *, signaling that it is ready to receive a command.

GENERAL RULES governing the telecommunications commands are:

1. * from datalogger means "ready for command".
2. All commands are of the form: [no.]letter, where the number may or may not be optional.
3. Valid characters are the numbers 0-9, the capital letters A-L, the colon (:), and the carriage return (CR).
4. An illegal character increments a counter and zeros the command buffer, returning *.
5. CR to datalogger means "execute".

6. **CRLF** from datalogger means "executing command".
7. ANY character besides a **CR** sent to the datalogger with a legal command in its buffer causes the datalogger to abort the command sequence with **CRLF\*** and to zero the command buffer.
8. All commands return a response code, usually at least a checksum.
9. The checksum includes all characters sent by the datalogger since the last \*, including the echoed command sequence, excluding only the checksum itself. The checksum is formed by summing the ASCII values, without parity, of the transmitted characters. The largest possible checksum value is 8191. Each time 8191 is exceeded, the CR7 starts the count over; e.g., if the sum of the ASCII values is 8192, the checksum is 0.
10. Commands that return Campbell Scientific binary format data (F and K commands) return a signature (Appendix C).

The CR7 sends ASCII data with eight data bits, no parity, plus one start bit and one stop bit.

After answering a ring, or completing a command, the CR7 waits about 40 seconds (147 seconds in the Remote Keyboard State) for a valid character to arrive. If a valid character is not received, the CR7 "hangs up". Some modems are quite noisy when not on line; it is possible for valid characters to appear in the noise pattern. To insure that this situation does not keep the CR7 in telecommunications, the CR7 counts all the invalid characters it receives from the time it answers a ring, and terminates communication after receiving 150 invalid characters.

The CR7 continues to execute its measurement and processing tasks while servicing the telecommunication requests. If the processing overhead is large (short execution interval), the processing tasks will slow the telecommunication functions. In a worst case situation, the CR7 interrupts the processing tasks to transmit a data point every 0.1 second.

The best way to become familiar with the Telecommunication Commands is to try them from a terminal connected to the CR7 via the SC32A or other modem interface (Section 6.5). Telecommunications Commands are described in the following Table. The Data Storage Pointer (DSP) and Telecommunications Modem Pointer (MPTR) referred to in the table are described in Section 2.1.

---

**TABLE 5.1-1. Telecommunications Commands**

| Command | Description |
|---------|-------------|
| **A** | STATUS - Datalogger returns Reference, the DSP location; the number of filled Final Storage locations; Version of datalogger; Errors #1 and #2 where #1 is the number of E08 and #2 is the number of overrun that have occurred (cleared by entering 8888A); Memory status, the decimal number (in ASCII characters) that is the equivalent of the 8 bit binary number shown as the result of the memory check on power-up; Location of MPTR; and Checksum. All in the following format: |
| | R+xxxxx F+xxxxx Vx Exx xx Mxxxx L+xxxxx Cxxxx |
| | If data are stored while in telecommunications, the A command must be issued to update the Reference to the new DSP. |
| [no. of arrays]**B** | BACK-UP - MPTR is backed-up the specified number of output arrays (no number defaults to 1) and advanced to the nearest start of array. CR7 sends the MPTR Location and Checksum: L+xxxxx Cxxxx |

| | |
|---|---|
| [YR:DAY:HR:MM:SS]**C** | RESET/SEND TIME - If time is entered the time is reset. If only 2 colons are in the time string, HR:MM:SS is assumed; 3 colons means DAY:HR:MM:SS. If only the C is entered, time is unaltered. CR7 returns year, Julian day, hr:min:sec, and Checksum: Y:xx Dxxxx Txx:xx:xx Cxxxx |
| [no. of arrays]**D** | ASCII DUMP - If necessary, the MPTR is advanced to the next start of array. CR7 sends the number of arrays specified (no number defaults to 1) or the number of arrays between MPTR and Reference, whichever is smaller, CRLF, Location, Checksum. |
| **E** | End call. Datalogger sends CRLF only. |
| [no. of loc.]**F** | BINARY DUMP - Used in TELCOM (PC208). See Appendix C. |
| [F.S. loc. no.]**G** | MOVE MPTR - MPTR is moved to specified Final Storage location. The location number must be entered. CR7 sends Location and Checksum: L+xxxxx Cxxxx |
| **2718H** | REMOTE KEYBOARD - CR7 sends the prompt ">" and is ready to execute standard keyboard commands (Section 5.2). |
| [loc. no.]**I** | Display/change value at Input Storage location. CR7 sends the value stored at the location. A new value and CR may then be sent. CR7 sends checksum. If no new value is sent (CR only) the location value will remain the same. |
| **3142J** | TOGGLE FLAGS AND SET UP FOR K COMMAND - Used in the Monitor Mode and with the Heads Up Display. See Appendix C for details. |
| **K** | CURRENT INFORMATION - In response to the K command, the CR7 sends datalogger time, user flag status, the data at the input locations requested in the J command, and Final Storage Data if requested by the J command. Used in the Monitor Mode and with Heads Up Display. See Appendix C. |
| [Password]**L** | Unlocks security (if enabled) to the level determined by the password entered (See *C Mode, Section 1.7). CR7 sends security level (0-3) and checksum: Sxx Cxxxx |

## 5.2 REMOTE PROGRAMMING OF THE CR7

The CR7 can be programmed via telecommunications using the PC208 software or manually through the Remote Keyboard State.

The PC208 Datalogger Support Software was developed for use with IBM or compatible PCs.

The CR7 is placed in the Remote Keyboard State by sending "2718H" and a carriage return (CR). The CR7 responds by sending a CR, line feed (LF), and the prompt ">". The CR7 is then ready to receive the standard keyboard commands (Section OV3); it recognizes all the standard CR7 keyboard characters plus the decimal point. While in the Remote Keyboard State, the CR7 sends the ASCII character control Q (17 decimal) after each user entry. Entering *0 returns the CR7 to the telecommunications command state.

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

## SECTION 5. TELECOMMUNICATIONS

Telecommunications Command State and the Remote Keyboard State.

Keying *0 will compile and run the CR7 program if program changes have been made. To compile and run the program without leaving the Remote Keyboard State, use *6 (Section 1.1.4). The CR7 display will show "LOG" when *0 is executed via telecommunications. It will not indicate active tables (enter *0 via the keyboard and the display will show the tables).

2718H

<div align="center">

Telecommunications       *0       Remote
Command                  Keyboard
State                   State

$\longrightarrow$

$\longleftarrow$

</div>
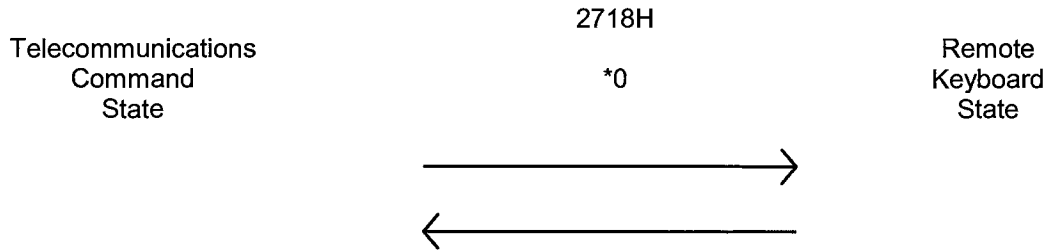
# SECTION 6. CS I/O 9 PIN SERIAL INPUT/OUTPUT

## 6.1 PIN DESCRIPTION

All external communication peripherals connect
to the CR7 through the 9-pin CS I/O connector
(Figure 6.1-1). Table 6.1-1 gives a brief
description of each pin's function.
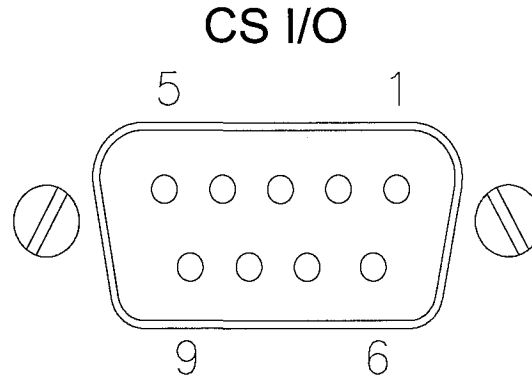
## CS I/O



**FIGURE 6.1-1.  CS I/O 9 Pin Connection**

### TABLE 6.1-1.  Pin Description

ABR  =  Abbreviation for the function name.
PIN  =  Pin number.
O    =  Signal Out of the CR7 to a peripheral.
I    =  Signal Into the CR7 from a peripheral.

| PIN | ABR | I/O | Description |
|-----|-----|-----|-------------|
| 1 | 5V | O | 5V:  Sources 5V DC, used to power some peripherals. |
| 2 | G | | Ground:  Provides a power return for pin 1 (5V), and is used as a reference for voltage levels. |
| 3 | RING | I | Ring:  When raised by a peripheral the CR7 enters telecommunications. |
| 4 | RXD | I | Receive Data:  Serial data transmitted by a peripheral are received on pin 4. |
| 5 | ME | O | Modem Enable:  Raised by the CR7 after the ring line has been raised. |

| PIN | ABR | I/O | Description |
|-----|-----|-----|-------------|
| 6 | PE | O | Printer Enable:  Raised to enable Storage Module or other print device. |
| 7 | G | I/O | Ground, common with pin 2. |
| 8 | 12 V | O | 12 volt power for peripherals. |
| 9 | TXD | O | Transmit Data:  Serial data are transmitted from the CR7 to peripherals on pin 9; logic low marking (0V) logic high spacing (5V) standard asynchronous ASCII, 8 data bits, no parity, 1 start bit, 1 stop bit, 300, 1200, 9600, 76,800 baud (user selectable). |

## 6.2 ENABLING PERIPHERALS

Several peripherals may be connected in parallel to the CS I/O 9-pin port. The CR7 directs data to a particular peripheral by raising the voltage on a specific pin dedicated to the peripheral; the peripheral is enabled when the pin goes high. Two pins are dedicated to specific devices Modem Enable pin 5 and Print Enable pin 6.

Modem Enable (**ME**), pin 5, is raised to enable a modem that has raised the ring line. *Only one modem/terminal may be connected to the CR7.*

Print Enable (**PE**), pin 6, is raised to enable a Storage Module or other print peripheral. Print peripherals are defined as peripherals which have an asynchronous serial communications port used to RECEIVE data transferred by the CR7. In most cases the peripheral is a printer, but could also be an on-line computer or other device. It is possible to have more than one print peripheral connected to the CR7 at one time, as long as they don't load down the TXD line (e.g., two Storage Modules, Section 4.4.1); all connected receive the same data.

## 6.3 INTERRUPTING DATA TRANSFER TO STORAGE PERIPHERALS

Instruction 96 is used for on-line data transfer to peripherals (Section 4.1). Data transfer is aborted when a modem raises the Ring line and the CR7 then enters Telecommunications (Section 5, 6.4). After the CR7 exits Telecommunications, data transfer to the peripheral is resumed the next time Instruction 96 is executed, or, if activated by the *4 Mode, at the completion of the next active table.

The *8 and *9 Modes are used to position the Memory Pointers, and to manually initiate data transfer from Final Storage to a peripheral. If the # key is pressed during data transfer, the transfer is stopped and the display shows the Final Storage location where the pointer stopped.

Data transfer can be stopped as follows:

1. Printable ASCII - after every output array.

2. Binary - after every Final Storage location.

## 6.4 TELECOMMUNICATIONS - MODEM PERIPHERALS

Any serial communication device which raises the Ring line and holds it high until the ME line is raised is a modem. The CSI field modem (DC112, COM200, COM100, or DC1765), RF95 RF modem, MD9 Multi-Drop Interface, and the SC32A RS232 interface used with computers or terminals are modems.

When a modem raises the Ring line, the CR7 responds by raising the ME line. The CR7 must be sent carriage returns until it sets the baud rate. When the baud rate is set, the CR7 sends a carriage return, line feed, *.

The ME line is held high until the CR7 receives an E to exit telecommunications or until a time limit expires without receiving a character. The colon in CR7 display is not shown while the CR7 is in telecommunications.

Some modems are quite noisy when not on line; it is possible for valid characters to appear in the noise pattern. For this reason, the CR7 counts all the invalid characters it receives from the time it answers a ring and terminates communication (lowers the ME line and returns to the *0 Mode) after receiving 150 invalid characters.

## 6.5 INTERFACING WITH COMPUTERS, TERMINALS, AND PRINTERS

This section deals with some of the basics of serial communication between the CR7 and common computer equipment. If you have an IBM compatible PC, the PC208(W) Datalogger Support Software takes care of the software protocol required in communicating with the CR7. This section does not discuss modem interfaces other than the SC32A. Please refer to the PC208 software and modem operator's manuals for interfacing details on other modems.

## 6.5.1 SC32A INTERFACE

Most computers, terminals, and printers require the SC32A Optically Isolated RS232 Interface for a "direct" connection to the CR7. The SC32A raises the CR7's ring line when it receives characters from the computer or terminal, and converts the CR7's logic levels (0V logic low, 5V logic high) to RS232 logic levels.

The SC32A 25 pin port is configured as Data Communications Equipment (DCE) which allows direct connection to Data Terminal Equipment (DTE), which includes most PCs and printers. For connection to DCE devices such as modems and some computers, use SC932 interface in place of SC32A.

When the SC32A receives a character from the computer or terminal (pin 2), 5V is applied to the datalogger Ring line (pin 3) for one second or until the Modem Enable line (ME) goes high. The CR7 waits approximately 40 seconds to receive carriage returns, which it uses to establish baud rate. After the baud rate is set the CR7 transmits a carriage return, line feed, *, and enters the Telecommunications Command State (Section 5). If the carriage returns are not received within the 40 seconds, the CR7 "hangs up".

---

**NOTE:** The SC32A has a jumper. With the jumper in place, the SC32A blocks printer data and passes data only when the CR7 is in Telecommunications.

---

## 6.5.2 COMPUTER/TERMINAL REQUIREMENTS

Computers, terminals and printers are usually configured as Data Terminal Equipment (DTE). Pins 4 and 20 are used as handshake lines, which are set high when the serial port is enabled. Power for the SC32A is taken from these pins. For equipment configured as DTE, a direct ribbon cable connects the modem/terminal to the SC32A. Clear to Send (CTS) pin 5, Data Set Ready (DSR) pin 6, and Received Line Signal Detect (RLSD) pin 8 are held high by the SC32A (when the RS232 section is powered) which should satisfy hardware handshake requirements of the modem/terminal.

Table 6.5-1 lists the most common RS232 configuration for Data Terminal Equipment.

### TABLE 6.5-1. DTE Pin Configuration

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

| PIN | ABR | I/O | FUNCTION |
|-----|-----|-----|----------|
| 2 | TD | O | Transmitted Data: Data is transmitted from the terminal on this line. |
| 3 | RD | I | Received Data: Data is received by the terminal on this line. |
| 4 | RTS | O | Request to Send: The terminal raises this line to ask a receiving device if the terminal can transmit data. |
| 5 | CTS | I | Clear to Send: The receiving device raises this line to let the terminal know that the receiving device is ready to accept data. |
| 20 | DTR | O | Data Terminal Ready: The terminal raises this line to tell the modem to connect itself to the telephone line. |
| 6 | DSR | I | Data Set Ready: The modem raises this line to tell the terminal that the modem is connected to the phone line. |
| 8 | DCD | I | Data Carrier Detect: The modem raises this line to tell the terminal that the modem is receiving a valid carrier signal from the phone line. |
| 22 | RI | I | Ring Indicator: The modem raises this line to tell the terminal that the phone is ringing. |
| 7 | SG | | Signal Ground: Voltages are measured relative to this point. |
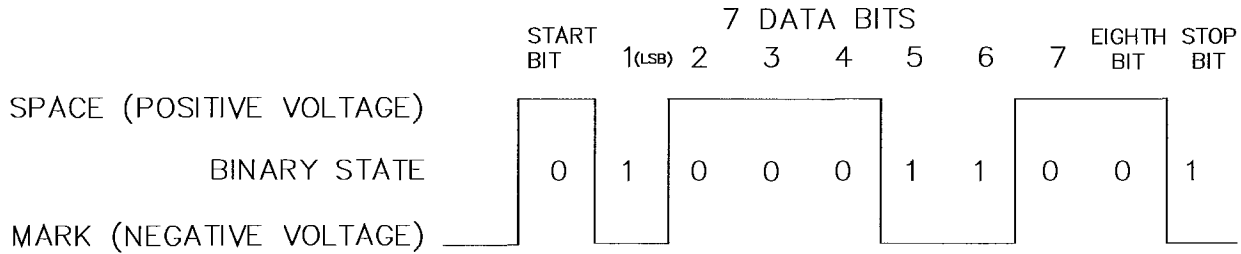
**FIGURE 6.5-1. Transmitting the ASCII Character 1**

### 6.5.3 COMMUNICATION PROTOCOL/TROUBLE SHOOTING

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

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

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

Most computers use 8-bits (1 byte) for data communications. The eighth bit is sometimes used for a type of error checking called parity checking. Even parity binary numbers have an even number of 1's, odd-parity characters have an odd number of 1's. When parity checking is used, the eighth bit is set to either a 1 or a 0 to make the parity of the character correct. The CR7 ignores the eighth bit of a character that is receives, and transmits the eighth bit as a binary 0. This method is generally described as "no parity".

To separate ASCII characters, a Start bit is sent before the first data bit, and a Stop bit is sent after the eighth data bit. The start bit is always a space, and the stop bit is always a mark. Between characters, the signal is in the marking condition.

Figure 6.5-1 shows how the ASCII character "1" is transmitted. The SC32A interface transmits spacing and marking voltages which are positive and negative, as shown. Signal voltages at the CR7 I/O port are 5 volts in the spacing condition, and 0 volts in the marking condition.

BAUD RATE

BAUD RATE is the number of bits transmitted per second. The CR7 can communicate at 300, 1200, 9600, and 76,800 baud. In the Telecommunications State, the CR7 will set its baud rate to match the baud rate of the modem.

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

DUPLEX

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

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

terminal is set to half duplex rather than the correct setting of full duplex.

IF NOTHING HAPPENS

If the CR7 is connected via the SC32A interface to a terminal or computer and * is not received after sending carriage returns:

1.  Verify that the CR7 has power and that the cables connecting the devices are securely connected.

2.  Verify that the port of the computer or terminal is an asynchronous serial communications port configured as DTE (see Table 6.5-1). The most common problems occur when the user tries to use a parallel port, or doesn't know the port address (i.e. COM1 or COM2). IBM, and most compatibles come with a Diagnostic disk which can be used to identify ports, and their addresses. If the serial port is standard equipment, then the operators manual should give you this information. Some serial ports such as the Super Serial Card for Apple computers, can be configured as DTE or DCE with a jumper block. Pin functions must match with Table 6.5-1.

If you are using a computer without the PC208 software, then a program or communication software must be used to enable the serial port and to make the computer function as a terminal. The port should be enabled for 300, 1200, or 9600 baud, 8 data bits, 1 stop bit, and no parity.

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

IF GARBAGE APPEARS

If garbage characters appear on the modem/terminal, check that the modem/terminal's baud rate is supported by the CR7. If the baud rate is correct, verify that the modem/terminal is set for 8 Data bits, and no Parity. Garbage will appear if 7 Data bits and no Parity are used. If the modem/terminal is set to 8 Data bits and even or odd Parity, communication cannot be established.

# SECTION 7. MEASUREMENT PROGRAMMING EXAMPLES

*This section gives some examples of Input Programming for common sensors used with the CR7. These examples detail only the connections, Input, Program Control and Processing Instructions necessary to perform measurements and store the data in engineering units in Input Storage. Output Processing Instructions are omitted, it is left for the user to program the necessary instructions to obtain the final data in the form desired. NO OUTPUT TO FINAL STORAGE WILL TAKE PLACE WITHOUT ADDITIONAL PROGRAMMING.*

*The examples given in this section would likely be only fragments of larger program tables. In general, the examples are written with the measurements made by the first channels on the first cards in the I/O Module, the instructions at the beginning of a program table, and low number Input Storage locations used to Store the data. Because it is unlikely that an application and CR7 configuration exactly duplicates that assumed in an example, THESE EXAMPLES ARE NOT MEANT TO BE USED VERBATIM; CARDS AND CHANNELS REFERENCED, SENSOR CALIBRATION AND INPUT LOCATIONS SELECTED MUST BE ADJUSTED FOR THE ACTUAL CIRCUMSTANCES. UNLESS OTHERWISE NOTED, ALL EXCITATION CHANNELS ARE SWITCHED ANALOG OUTPUT.*

## 7.1 SINGLE ENDED VOLTAGE - LI200S SILICON PYRANOMETER

The silicon pyranometer puts out a current which is dependent upon the solar radiation incident upon the sensor. The current is measured as the voltage drop across a fixed resistor. The Campbell Scientific LI200S uses a 100 ohm resistor. The calibration supplied by LI-COR, the manufacturers of the pyranometer, is given in uA/kW/m$^2$. The calibration in terms of volts is determined by multiplying the $\mu$A calibration by the resistance of the fixed resistor.

The calibration of the pyranometer used in this example is assumed to be 76.9 $\mu$A/kW/m$^2$, which when multiplied by 100 ohms equals 7.69 mV/kW/m2. The multiplier used to convert the voltage reading to kW/m$^2$ is 1 / 7.69 mV/kW/m$^2$ = 0.13004.

Most LI-COR calibrations run between 60 and 90 $\mu$A/kW/m$^2$, which correspond to calibrations of 6.0 to 9.0 mV/kW/m$^2$. The flux density through a surface normal to the solar beam above the earth's atmosphere is 1.36 kW/m$^2$; radiation on earth will be less than this. Thus, the 15 mV scale provides an adequate range (9.0 mV/kW/m$^2$ x 1.36 kW/m$^2$ < 15 mV).

CONNECTIONS

The pyranometer output is measured with a single ended voltage measurement on channel 5. There are twice as many single ended

channels as differential channels and they are numbered accordingly: single ended channel 5 is the high side of differential channel 3, and the low side is single ended channel 6.
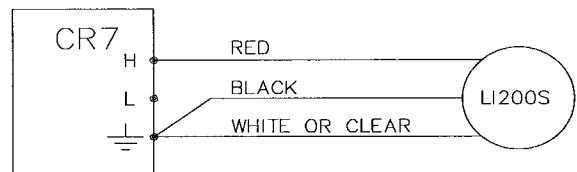


**FIGURE 7.1-1. Wiring Diagram for LI200S**

### PROGRAM

| 01: | P1 |  | Volt (SE) |
|-----|-----|--------|-----------|
| 01: | 1 |  | Rep |
| 02: | 3 |  | 15 mV slow Range |
| 03: | 1 |  | IN Card |
| 04: | 5 |  | IN Chan |
| 05: | 1 |  | Loc [:R kW/m^2 ] |
| 06: |  | .13004 | Mult |
| 07: | 0 |  | Offset |

## 7.2 DIFFERENTIAL VOLTAGE MEASUREMENT

Some sensors either contain or require active signal conditioning circuitry to provide an easily measured analog voltage output. Generally, the output is referenced to the sensor ground. The associated current drain usually requires a power source external to the CR7. A typical connection scheme where AC power is not available and both the CR7 and sensor are powered by an external battery is shown in

Figure 7.2-1.  Since a single ended measurement is referenced to the CR7 ground, any voltage difference between the sensor ground and CR7 ground becomes a measurement error.  A differential measurement avoids this error by measuring the signal between the 2 leads without reference to ground.  This example analyzes the potential error on a water pH measurement using a Martek Mark V water quality analyzer.



**FIGURE 7.2-1.  Typical Connection for Active Sensor with External Battery**

The wire used to supply power from the external battery is 18 AWG with an average resistance of 6.5 ohms/1000 ft.  The power runs to the CR7 and pH meter are 2 ft. and 10 ft., respectively. Typical current drain for the pH meter is 300 mA.  When making measurements, the CR7 draws about 100 mA.  Since voltage is equal to current times resistance (V=IR), ground voltages at the pH meter and the CR7 relative to battery ground are:

pH meter ground =
0.3A x 10/1000 x 6.5 Ohms = +0.0195V

CR7 ground =
0.1A x 2/1000 x 6.5ohms = +0.0013V

Ground at the pH meter is 0.0182V higher than ground at the CR7.  The meter output is 0-1 volt referenced to meter ground, for the full range of 14 pH units, or 0.0714V/pH.  Thus, if the output is measured with a single ended voltage measurement, it is 0.0182V or 0.25 pH units too high.  If this offset remained constant, it could be corrected in programming the CR7. However, it is better to use a differential voltage measurement which does not rely on the current drain remaining constant.  The Program that follows illustrates the use of Instruction #2 to make the measurement.  A multiplier of 0.014 is used to convert the millivolt output into pH units.

**PROGRAM**

| 01: | P2 | Volt (DIFF) |
|---|---|---|
| 01: | 1 | Rep |
| 02: | 7 | 1500 mV slow Range |
| 03: | 1 | IN Card |
| 04: | 1 | IN Chan |
| 05: | 1 | Loc [:pH     ] |
| 06: | 0.014 | Mult |
| 07: | 0 | Offset |

## 7.3  THERMOCOUPLE TEMPERATURES USING 723-T REFERENCE

The use of the 723-T Analog Input Card RTD to measure the reference temperature is described in the introductory programming example (Section OV4).

## 7.4  THERMOCOUPLE TEMPERATURES USING AN EXTERNAL REFERENCE JUNCTION

When a number of thermocouple measurements are made at some distance from the CR7, it is often better to use a reference junction box located at the site rather than using the panel temperature of the CR7.  This reduces the required length of expensive thermocouple wire as regular copper wire can be used between the junction box (J-box) and CR7.  In addition, if the temperature gradient between the J-box and the thermocouple measurement junction is smaller than the gradient between the CR7 and the measurement junction, thermocouple inaccuracy is reduced.  In the following example, an external reference junction is used on ten thermocouple measurements.  A Campbell Scientific 107 Temperature Probe is used to measure the reference temperature.  The connection scheme is shown in Figure 7.4-1.



**FIGURE 7.4-1.  Thermocouples with External Reference Junction**

The temperature of the 107 Probe is stored in Input Location 1 and the thermocouple temperatures in Locations 2-11.

**PROGRAM**

| 01: | P11 | Temp 107 Probe |
|---|---|---|
| 01: | 1 | Rep |
| 02: | 1 | IN Card |
| 03: | 21 | IN Chan |
| 04: | 1 | EX Card |
| 05: | 1 | EX Chan |
| 06: | 1 | Loc [:Ref. Temp] |
| 07: | 1 | Mult |
| 08: | 0 | Offset |
| 02: | P14 | Thermocouple Temp (DIFF) |
| 01: | 10 | Reps |
| 02: | 3 | 15 mV slow Range |
| 03: | 1 | IN Card |
| 04: | 1 | IN Chan |
| 05: | 1 | Type T (Copper-Constantan) |
| 06: | 1 | Ref Temp Loc Ref. Temp |
| 07: | 2 | Loc [:TC temp#1] |
| 08: | 1 | Mult |
| 09: | 0 | Offset |

## 7.5 THERMOCOUPLES FOR DIFFERENTIAL TEMPERATURE MEASUREMENT

When configured correctly, thermocouples are capable of measuring small temperature gradients very accurately (Section 13.4). In this example, the CR7 is used to make five differential temperature measurements with chromel-constantan thermocouples. The connections are shown in Figure 7.5-1 where the voltage measured between the chromel leads is proportional to the temperature difference between junctions R and D.



CR — CHROMEL
CO — CONSTANTAN
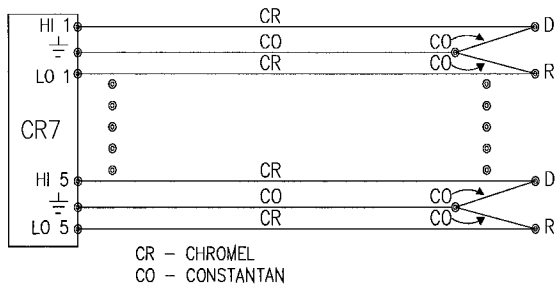
**FIGURE 7.5-1. Connection for Thermocouple Differential Temperature Measurement**

When the temperatures are within the reference junction compensation range (Table 13.4-3), three instructions are required in the measurement sequence:

1. A CR7 Panel Temperature Measurement (#17) used as a reference temperature for the measurement at R.

2. A single ended TC measurement (#13) of R temperatures to be used as reference temperatures for the measurement D.

3. A differential TC measurement of D temperatures where the reference temperature at R are subtracted from the results as specified in Parameter 5.

The connection shown in Figure 7.4-1 yields the conventional polarity (sign) for the temperature difference, i.e., D>R=+T, D<R=-T. Using R as the reference temperature maintains this convention whereas using D reverses the sign of the output.

Prefixing a 2 onto the TC type in Parameter 5 of Instruction 13 causes the CR7 to skip every other single ended channel. Keying C before entering Parameter 6 in Instruction 14 causes the reference temperature location to be incremented each rep.

The ±5mV range used in Instruction 13 allows measurement of temperatures at R within a range of approximately ±80 °C of the I/O Module temperature. The ±1.5mV range used in Instruction 14 allows the temperature difference between D and R to approach a range of ±24 °C, for temperatures around 25 °C (output = 61 uV/°C). The resolution of the differential temperature measurement is approximately 0.0008 °C (50 nV/61µV/°C).

The panel temperature is stored in Input Location 1, the temperatures of the R junctions in locations 2-6 and the temperature differences (D-R) in locations 7-11. If it is not necessary to retain the temperatures of the R junctions, the temperature differences could be stored in locations 2-6 by changing Parameter #7 in Instruction 14 to 2.

**PROGRAM**

| 01: | P17 | Panel Temperature |
|---|---|---|
| 01: | 1 | IN Card |
| 02: | 1 | Loc [:PANL TEMP] |

| 02: | P13 | Thermocouple Temp (SE) |
|---|---|---|
| 01: | 5 | Reps |
| 02: | 2 | 5000 uV slow Range |
| 03: | 1 | IN Card |
| 04: | 2 | IN Chan |
| 05: | 22 | Type E  (Skip every other chan) |
| 06: | 1 | Ref Temp Loc PANL TEMP |
| 07: | 2 | Loc [:S.E. T#1 ] |
| 08: | 1 | Mult |
| 09: | 0 | Offset |

| 03: | P14 | Thermocouple Temp (DIFF) |
|---|---|---|
| 01: | 5 | Reps |
| 02: | 1 | 1500 uV slow Range |
| 03: | 1 | IN Card |
| 04: | 1 | IN Chan |
| 05: | 12 | Type E  (Temp difference) |
| 06: | 2-- | Ref Temp Loc S.E. T#1 |
| 07: | 7 | Loc [:DIFF T #1] |
| 08: | 1 | Mult |
| 09: | 0 | Offset |

When the temperature of the R junction is outside of the CR7 reference junction compensation range (Table 13.4-3), the TCs must be connected in the normal fashion, one TC per input channel; both temperatures measured and one subtracted from the other to find the difference. This must be done because any error in the reference junction compensation becomes an error in the temperature difference.

## 7.6 TEMPERATURE WITH CALIBRATED THERMOCOUPLES

Thermocouple calibration (Section 13.4) results in a slope correction. The correction must be applied only to the thermocouple output. When Instructions 13 and 14 are used to measure temperature, the temperature is the sum of the reference temperature and the temperature difference calculated from the thermocouple output. The correction must be applied to the temperature difference before the reference temperature is added.

Example A demonstrates the use of a scaling array (Instruction 53) to correct the calibration of four individually calibrated thermocouples.

Another means of applying a correction factor to a number of thermocouples is to group together those with a similar correction factor. In example B, the slope correction factor for a group of 5 thermocouples is entered as the multiplier (Parameter 8) in the instruction to read those thermocouples. The example only shows one group of thermocouples. If there were several groups with similar correction factors, Instruction 14 would be used to read and correct each group.

After the slope correction is made, a loop is used to add the reference temperature to the corrected temperature differences.

CONNECTIONS

The thermocouples are connected in the normal manner: chromel to Hi and constantan to Low. In both examples, the first thermocouple is connected to Channel 1. Care must be taken that the correction factors called for in the programming match the channels that the calibrated thermocouples are connected to.

**PROGRAM A**

| 01: | P17 | Panel Temperature |
|---|---|---|
| 01: | 1 | IN Card |
| 02: | 1 | Loc [:REF TEMP ] |

| 02: | P14 | Thermocouple Temp (DIFF) |
|---|---|---|
| 01: | 4 | Reps |
| 02: | 3 | 15 mV slow Range |
| 03: | 1 | IN Card |
| 04: | 1 | IN Chan |
| 05: | .12 | Type E  (Temp difference) |
| 06: | 1 | Ref Temp Loc REF TEMP |
| 07: | 2 | Loc [:TC temp#1] |
| 08: | 1 | Mult |
| 09: | 0 | Offset |

| 03: | P53 | Scaling Array (A*loc +B) |
|---|---|---|
| 01: | 2 | Start Loc [:TC temp#1] |
| 02: | .99255 | A1 |
| 03: | 0 | B1 |
| 04: | .99703 | A2 |
| 05: | 0 | B2 |
| 06: | 1.0045 | A3 |
| 07: | 0 | B3 |
| 08: | 1.0075 | A4 |
| 09: | 0 | B4 |

| 04: | P87 | Beginning of Loop |
|---|---|---|
| 01: | 0 | Delay |
| 02: | 4 | Loop Count |
| | | |
| 05: | P33 | Z=X+Y |
| 01: | 1 | X Loc REF TEMP |
| 02: | 2-- | Y Loc TC temp#1 |
| 03: | 2-- | Z Loc [:TC temp#1] |
| | | |
| 06: | P95 | End |

### PROGRAM B

| 01: | P17 | Panel Temperature |
|---|---|---|
| 01: | 1 | IN Card |
| 02: | 1 | Loc [:REF TEMP ] |
| | | |
| 02: | P14 | Thermocouple Temp (DIFF) |
| 01: | 5 | Reps |
| 02: | 3 | 15 mV slow Range |
| 03: | 1 | IN Card |
| 04: | 1 | IN Chan |
| 05: | 12 | Type E  (Temp difference) |
| 06: | 1 | Ref Temp Loc REF TEMP |
| 07: | 2 | Loc [:TC temp#1] |
| 08: | .99253 | Mult |
| 09: | 0 | Offset |

*If there were additional groups of thermocouples, the Instructions to measure them would be inserted here and Parameter 2 in Instruction 87 adjusted accordingly.*

| 03: | P87 | Beginning of Loop |
|---|---|---|
| 01: | 0 | Delay |
| 02: | 5 | Loop Count |
| | | |
| 04: | P33 | Z=X+Y |
| 01: | 1 | X Loc REF TEMP |
| 02: | 2-- | Y Loc TC temp#1 |
| 03: | 2-- | Z Loc [:TC temp#1] |
| | | |
| 05: | P95 | End |

## 7.7  107 TEMPERATURE PROBE

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

CONNECTIONS

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

### PROGRAM

| 01: | P11 | Temp 107 Probe |
|---|---|---|
| 01: | 3 | Reps |
| 02: | 1 | IN Card |
| 03: | 1 | IN Chan |
| 04: | 1 | EX Card |
| 05: | 1 | EX Chan |
| 06: | 1 | Loc [:107 T #1 ] |
| 07: | 1 | Mult |
| 08: | 0 | Offset |

## 7.8  207 TEMPERATURE AND RH PROBE

Instruction 12 excites and measures the RH portion of the Campbell Scientific 207 temperature and relative humidity probe. This instruction relies on a previously measured temperature to compute the RH from the probe resistance. Instruction 12 has the option of using a single temperature to provide the compensation reference for several RH probes. In this example, three probes will be measured; the temperature of each probe will be measured and used to provide temperature compensation for that probe. Instruction 11 is used to obtain the temperatures of the three probes which are stored in Input locations 1-3, the RH values are stored in Input locations 4-6. The temperature measurements are made on single ended input channels 1-3, just as in example 7.7. The program listed below is a continuation of the program given in example 7.7.

CONNECTIONS

The black leads from the probes are connected to excitation channel 1, the clear leads are connected to ground. The red leads are from the thermistor circuit and are connected to single ended channels 1-3. The white leads are from the RH circuit and are connected to single ended channels 4-6. The correct order must be maintained when connecting the red and white leads, i.e., the red lead from the first probe is connected to single ended channel 1 and the white lead from that probe is connected to single ended channel 4, etc.

**PROGRAM**

| 02: | P12 | RH 207 Probe |
|---|---|---|
| 01: | 3 | Reps |
| 02: | 1 | IN Card |
| 03: | 4 | IN Chan |
| 04: | 1 | EX Card |
| 05: | 1 | EX Chan |
| 06: | 1 | Meas/Temp |
| 07: | 1 | Temperature Loc 207 T#1 |
| 08: | 4 | Loc [:RH #1   ] |
| 09: | 1 | Mult |
| 10: | 0 | Offset |

## 7.9 ANEMOMETER WITH PHOTOCHOPPER OUTPUT

An anemometer with a photochopper transducer produces a pulsed output which is monitored with the Pulse Count Instruction, configured for High Frequency Pulses. The Pulse Count Instruction counts the number of pulses occurring in each execution interval. An option in the instruction allows this to be converted to frequency in Hertz (i.e., Pulses/Second). The anemometer used in this example is the R. M. Young Model No. 12102D Cup Anemometer, with a 10 window chopper wheel. The photochopper circuitry is powered from the CR7 12V supply; AC power or backup batteries should be used to compensate for the increased current drain.

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

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

The multiplier and offset to convert pulses per second to meters per second are:

$$m/s = 0.01632 \text{ m/s/rpm} \times 6 \text{ rpm/(pulse/s)} + 0.2 \text{ m/s} = 0.0979 \text{ m/s/pulse} \times \text{pulses} + 0.2 \text{ m/s}$$

There are occasionally times when the CR7's CPU is occupied and does not reset the pulse counters at the exact time interval programmed. If the artificially large wind speed that results from a long interval is used, it causes a false average or maximum value. To avoid this, the CR7 is instructed to discard values resulting from long intervals, and use the previous value instead.
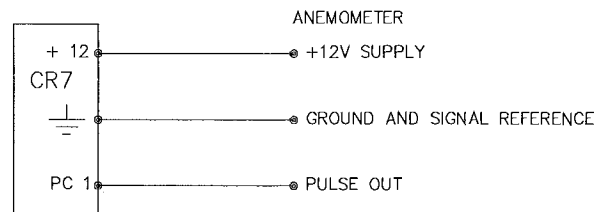


**FIGURE 7.9-1. Wiring Diagram for Anemometer**

**PROGRAM**
(Execution interval 10 seconds)

| 01: | P3 | Pulse |
|---|---|---|
| 01: | 2 | Reps |
| 02: | 2 | IN Card |
| 03: | 2 | Pulse Input Chan |
| 04: | 20 | High frequency; Output Hz. |
| 05: | 10 | Loc [:WS m/s  ] |
| 06: | .0979 | Mult |
| 07: | .2 | Offset |

## 7.10 TIPPING BUCKET RAINGAGE WITH LONG LEADS

A tipping bucket raingage is measured with the Pulse Count Instruction configured for Switch Closure. Counts from long intervals will be used, as the final output desired is total rainfall (obtained with Instruction 72, Totalize). If counts from long intervals were discarded, less rainfall would be recorded than was actually measured by the gage (assuming there were counts in the long intervals). Output is desired in millimeters of precipitation; the gage is calibrated for a 0.01 inch tip so a multiplier of 0.254 is used.
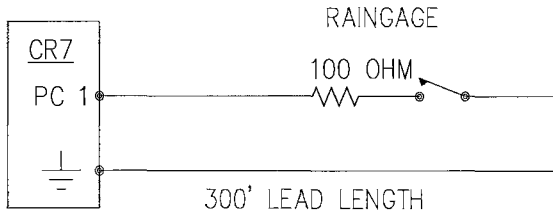
**FIGURE 7.10-1. Wiring Diagram for Raingage with Long Leads**

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

**PROGRAM**

| 01: | P3 | Pulse |
|-----|------|-------|
| 01: | 1 | Rep |
| 02: | 2 | IN Card |
| 03: | 1 | Pulse Input Chan |
| 04: | 2 | Switch closure |
| 05: | 11 | Loc [:RAIN mm  ] |
| 06: | 0.254 | Mult |
| 07: | 0 | Offset |

## 7.11  100 OHM PRT IN 4 WIRE HALF BRIDGE

Instruction 9 is the best choice for accuracy where the Platinum Resistance Thermometer (PRT) is separated from other bridge completion resistors by a lead length having more than a few thousandths of an ohm resistance. In this example, it is desired to measure a temperature in the range of -10 to 40 $^\circ$C. The length of the cable from the CR7 to the PRT is 500 feet.
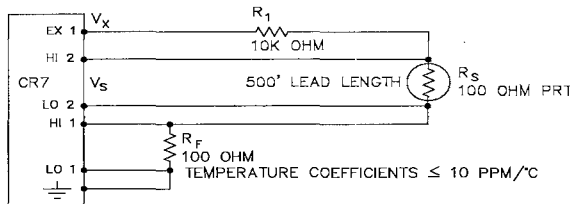


**FIGURE 7.11-1. Wiring Diagram for PRT in 4 Wire Half-Bridge**

Figure 7.11-1 diagrams the circuit used to measure the PRT. The 10 kohm resistor allows the use of a high excitation voltage and low voltage ranges on the measurements. This insures that noise in the excitation does not have an effect on signal noise. Because the fixed resistor ($R_f$) and the PRT ($R_s$) have approximately the same resistance, the differential measurement of the voltage drop across the PRT can be made on the same range as the differential measurement of the voltage drop across $R_f$. The use of the same range eliminates any range translation error that might arise from the 0.01% tolerance of the range translation resistors in the CR7.

If the voltage drop across the PRT ($V_2$) is kept on the 50 mV range, self heating of the PRT should be less than 0.001 $^\circ$C in still air. The resolution of the measurement is increased as the excitation voltage ($V_x$) is increased. The voltage drop across the PRT is equal to $V_x$ multiplied by the ratio of $R_s$ to the total resistance, and is greatest when $R_s$ is greatest ($R_s$=115.54 ohms at 40 $^\circ$C). To find the maximum excitation voltage that can be used, we assume $V_2$ equal to 50 mV and use Ohm's Law to solve for the resulting current, I.

$$I = 50mV/R_s = 50mV/115. 54\ Ohms = 0.433mA$$

Next solve for $V_x$:

$$V_x = I(R_1+R_s+R_f) = 4.42V$$

If the actual resistances were the nominal values, the CR7 would not overrange with $V_x$ = 4.4 V. To allow for the tolerances in the actual resistances it is decided to set $V_x$ equal to 4.2 volts (e.g., if the 10 kohms resistor is 5% low, $R_s/(R_1+R_s+R_f)$=115.54/9715.54 and $V_x$ must be 4.204V to keep $V_s$ less than 50 mV).

The result of Instruction 9 when the first differential measurement ($V_1$) is not made on the 5V range is equivalent to $R_s/R_f$. Instruction 16 computes the temperature ($^\circ$C) for a DIN 43760 standard PRT from the ratio of the PRT resistance to its resistance at 0 $^\circ$C ($R_s/R_0$). Thus, a multiplier of $R_f/R_0$ is used in Instruction 9 to obtain the desired intermediate, $R_s/R_0$ (= $R_s/R_f$ x $R_f/R_0$). If $R_s$ and $R_0$ were each exactly 100 ohms the multiplier would be 1. However, neither resistance is likely to be exact. The correct multiplier is found by connecting the PRT to the CR7 and entering Instruction 9 with

a multiplier of 1. The PRT is then placed in an ice bath (0 °C; $R_s=R_0$), and the result of the bridge measurement is read using the *6 Mode. The reading is $R_s/R_f$, which is equal to $R_0/R_f$ since $R_s = R_0$, the correct value of the multiplier, $R_f/R_0$, is the reciprocal of this reading. The initial reading assumed for this example was 0.9890, the correct multiplier is: $R_f/R_0 = 1/0.9890 = 1.0111$.

The fixed 100 ohm resistor must be thermally stable. Its precision is not important because the exact resistance is incorporated, along with that of the PRT, into the calibrated multiplier. The 10 ppm/°C temperature coefficient of the fixed resistor will limit the error due to its change in resistance with temperature to less than 0.15 °C over the specified temperature range. Because the measurement is ratiometric ($R_s/R_f$), the properties of the 10 kohm resistor do not affect the result.

## PROGRAM

| 01: | P9 | Full BR w/Compensation |
|---|---|---|
| 01: | 1 | Rep |
| 02: | 4 | 50 mV slow EX Range |
| 03: | 4 | 50 mV slow BR Range |
| 04: | 1 | IN Card |
| 05: | 1 | IN Chan |
| 06: | 1 | EX Card |
| 07: | 1 | EX Chan |
| 08: | 1 | Meas/EX |
| 09: | 4200 | mV Excitation |
| 10: | 1 | Loc [:Rs/Ro   ] |
| 11: | 1.0111 | Mult |
| 12: | 0 | Offset |
| | | |
| 02: | P16 | Temperature RTD |
| 01: | 1 | Rep |
| 02: | 1 | R/Ro Loc Rs/Ro |
| 03: | 2 | Loc [:TEMP degC] |
| 04: | 1 | Mult |
| 05: | 0 | Offset |

## 7.12 100 OHM PRT IN 3 WIRE HALF BRIDGE

The temperature measurement requirements in this example are the same as in section 7.11. In this case a three wire half bridge, Instruction 7, is used to measure the resistance of the PRT. The diagram of the PRT circuit is shown in Figure 7.12-1.
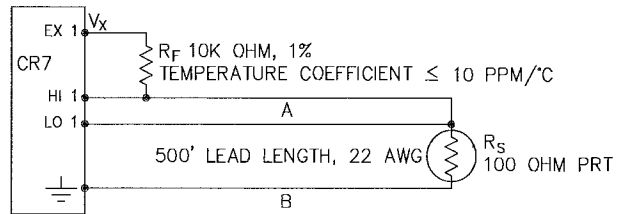


**Figure 7.12-1. 3 Wire Half-Bridge Used to Measure 100 ohm PRT**

As in the example in section 7.11, the excitation voltage is calculated to be the maximum possible yet allow the ±50 mV measurement range. The 10 kohm resistor has a tolerance of ±1%, thus, the lowest resistance to expect from it is 9.9 kohms. We calculate the maximum excitation voltage ($V_x$) to keep the voltage drop across the PRT less than 50 mV:

$0.050V > V_x \ 115.54/(9900+115.54); \ V_x < 4.33V$

The excitation voltage used is 4.3V.

The multiplier used in Instruction 7 is determined in the same manner as in section 7.11. In this example the multiplier ($R_f/R_0$) is assumed to be 100.93.

The 3 wire half bridge compensates for lead wire resistance by assuming that the resistance of wire A is the same as the resistance of wire B. The maximum difference expected in wire resistance is 2%, but is more likely to be on the order of 1%. The resistance of $R_s$ calculated with Instruction 7, is actually $R_s$ plus the difference in resistance of wires A and B. The average resistance of 22 AWG wire is 16.5 ohms per 1000 feet, which would give each 500 foot lead wire a nominal resistance of 8.3 ohms. Two percent of 8.3 ohms is 0.17 ohms. Assuming that the greater resistance is in wire B, the resistance measured for the PRT ($R_0 = 100$ ohms) in the ice bath would be 100.17 ohms, and the resistance at 40 °C would be 115.71. The measured ratio $R_s/R_0$ is 1.1551, the actual ratio is 115.54/100 = 1.1554. The temperature computed by Instruction 17 from the measured ratio would be about 0.1 °C lower than the actual temperature of the PRT. This source of error does not exist in the example in section 7.11, where the 4 wire half bridge is used to measure PRT resistance.

The advantages of the 3 wire half bridge are that it only requires 3 lead wires going to the sensor, and takes 2 single ended input channels whereas the 4 wire half bridge requires 2 differential input channels.

## PROGRAM

| 01: | P | 7 3 Wire Half Bridge |
|---|---|---|
| 01: | 1 | Rep |
| 02: | 4 | 50 mV slow Range |
| 03: | 1 | IN Card |
| 04: | 1 | IN Chan |
| 05: | 1 | EX Card |
| 06: | 1 | EX Chan |
| 07: | 1 | Meas/EX |
| 08: | 4300 | mV Excitation |
| 09: | 1 | Loc [:Rs/Ro   ] |
| 10: | 100.93 | Mult |
| 11: | 0 | Offset |

| 02: | P | 16 Temperature RTD |
|---|---|---|
| 01: | 1 | Rep |
| 02: | 1 | R/Ro Loc Rs/Ro |
| 03: | 2 | Loc [:TEMP degC] |
| 04: | 1 | Mult |
| 05: | 0 | Offset |

## 7.13 100 OHM PRT IN 4 WIRE FULL BRIDGE

This example describes obtaining the temperature from a 100 ohm PRT in a 4 wire full bridge (Instruction 6). The temperature being measured is in a constant temperature bath and is to be used as the input for a control algorithm. The PRT in this case does not adhere to the DIN standard (alpha = 0.00385) used in the temperature calculating Instruction 16. Alpha is defined as $(R_{100}/R_0-1)/100$ where $R_{100}$ and $R_0$ are the resistances of the PRT at 100 °C and 0 °C, respectively. In this PRT alpha is equal to 0.00392.



TEMPERATURE COEFFICIENT
OF ALL FIXED RESISTORS ≤ 5 PPM/°C
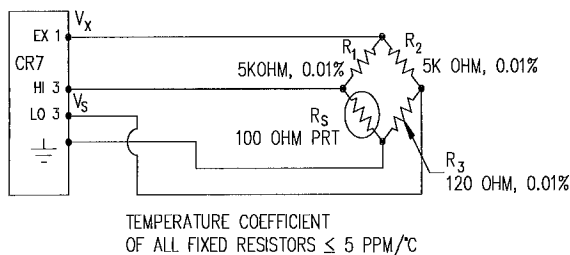
**FIGURE 7.13-1. Full Bridge Schematic For 100 Ohm PRT**

The result given by Instruction 6 (X) is 1000 $V_s/V_x$ (where $V_s$ is the measured bridge output voltage and $V_x$ is the excitation voltage) which is:

$$X = 1000 (R_s/(R_s+R_1)-R_3/(R_2+R_3))$$

The resistance of the PRT ($R_s$) is calculated with the Bridge Transform Instruction 59:

$$R_s = R_1 X'/(1-X')$$

Where

$$X' = X/1000 + R_3/(R_2+R_3)$$

Thus, to obtain the value $R_s/R_0$, ($R_0 = R_s$ @ 0°C) for the temperature calculating Instruction 16, the multiplier and offset used in Instruction 6 are 0.001 and $R_3/(R_2+R_3)$, respectively. The multiplier used in Instruction 59 to obtain $R_s/R_0$ is $R_1/R_0$ (5000/100 = 50).

It is desired to control the temperature bath at 50°C with as little variation as possible. High resolution is desired so the control algorithm will be able to respond to minute changes in temperature. The highest resolution is obtained when the temperature range results in an output voltage ($V_s$) range which fills the measurement range selected in Instruction 6. The full bridge configuration allows the bridge to be balanced ($V_s = 0V$) at or near the control temperature. Thus, the output voltage can go both positive and negative as the bath temperature changes, allowing the full use of the measurement range.

The resistance of the PRT is approximately 119.6 ohms at 50 °C. The 120 ohm fixed resistor balances the bridge at approximately 51 °C. The output voltage is:

$$V_s = V_x [R_s/(R_s+R_1) - R_3/(R_2+R_3)]$$

$$= V_x [R_s/(R_s+5000) - 0.023438]$$

The temperature range to be covered is 50 +5°C. At 45 °C $R_s$ is approximately 117.6 ohms, or:

$$V_s = -458.448 \times 10^{-6} V_x$$

$V_s$ can be measured on the ±1500 μV scale. Setting $V_s$ equal to -1500 μV and solving for $V_x$ results in $V_x = 3.272$ V. $V_x$ is entered as 3270 mV in Parameter 8 of Instruction 6.

The 5 ppm/$^{\circ}$C temperature coefficient of the fixed resistors was chosen so that their 0.01% accuracy tolerance would hold over the desired temperature range.

There is a change of approximately 1500 µV from the output at 45 $^{\circ}$C to the output at 51 $^{\circ}$C, or 250 µV/$^{\circ}$C. With a resolution of 50 nV on the 1500 µV range, this means that the temperature resolution is 0.0002 $^{\circ}$C.

The relationship between temperature and PRT resistance is a slightly nonlinear one. Instruction 16 computes this relationship for a DIN standard PRT where the nominal temperature coefficient is 0.00385/$^{\circ}$C. The change in nonlinearity of a PRT with the temperature coefficient of 0.00392/$^{\circ}$C is minute compared with the slope change. Entering a slope correction factor of 0.00385/0.00392 = 0.98214 as the multiplier in Instruction 16 results in a calculated temperature which is well within the accuracy specifications of the PRT.

**PROGRAM**

```
01:    P6        Full Bridge
  01:    1        Rep
  02:    1        1500 uV slow Range
  03:    1        IN Card
  04:    3        IN Chan
  05:    1        EX Card
  06:    1        EX Chan
  07:    1        Meas/EX
  08: 3270        mV Excitation
  09:   11        Loc [:Rs/Ro   ]
  10:      .001   Mult
  11:    .02344   Offset

02:    P59       BR Transform Rf[X/(1-X)]
  01:    1        Rep
  02:   11        Loc [:Rs/Ro   ]
  03:   50        Multiplier (Rf)

03:    P16       Temperature RTD
  01:    1        Rep
  02:   11        R/Ro Loc Rs/Ro
  03:   12        Loc :
  04:    .98214   Mult
  05:    0        Offset
```

## 7.14  PRESSURE TRANSDUCER - 4 WIRE FULL BRIDGE

This example describes a measurement made with a Druck PDCR 10/D depth measurement pressure transducer. The pressure transducer was ordered for use with 5 volt positive or negative excitation (passive temperature compensation) and has a range of 5 psi or about 3.5 meters of water. The transducer is used to measure the depth of water in a stilling well.

Instruction 6, 4 wire full bridge, is used to measure the pressure transducer. The high output of the semiconductor strain gage necessitates the use of the 50mV input range. The sensor is calibrated by connecting it to the CR7 and using Instruction 6 with a multiplier of 1 and an offset of 0, noting the readings (*6 Mode) with 10 cm of water above the sensor and with 334.6 cm of water above the sensor. The output of Instruction 6 is 1000 $V_s/V_x$ or millivolts per volt excitation. At 10 cm the reading is 0.19963 mV/V and at 334.6 cm the reading is 6.6485 mV/V. The multiplier to yield output in cm is:

$$(334.6 - 10)/(6.6485-.19963) = 50.334 \text{ cm/mV/V}$$

The offset is determined after the pressure transducer is installed in the stilling well. The sensor is installed 65 cm below the water level at the time of installation. The depth of water at this time is determined to be 72.6 cm relative to the desired reference. When programmed with the multiplier determined above and an offset of 0, a reading of 65.12 is obtained. The offset for the actual measurements is thus determined to be 72.6 - 65.12 = 7.48 cm.

The lead length is approximately 10 feet, so there is no appreciable error due to lead wire resistance.
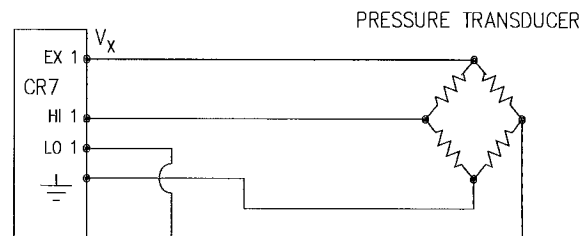


**FIGURE 7.14-1.  Wiring Diagram for Full Bridge Pressure Transducer**

**PROGRAM**

| 01: | P6 | Full Bridge |
|---|---|---|
| 01: | 1 | Rep |
| 02: | 4 | 50 mV slow Range |
| 03: | 1 | IN Card |
| 04: | 1 | IN Chan |
| 05: | 1 | EX Card |
| 06: | 1 | EX Chan |
| 07: | 1 | Meas/EX |
| 08: | 5000 | mV Excitation |
| 09: | 13 | Loc [:HEIGHT cm] |
| 10: | 50.334 | Mult |
| 11: | 7.48 | Offset |

## 7.15 LYSIMETER - 6 WIRE LOAD CELL

When a long cable is required between a load cell and the CR7, the resistance of the wire can create a substantial error in the measurement if the 4 wire full bridge (Instruction 6) is used to excite and measure the load cell. This error arises because the excitation voltage is lower at the load cell than at the CR7 due to voltage drop in the cable. The 6 wire full bridge (Instruction 9) avoids this problem by measuring the excitation voltage at the load cell. This example shows the errors one would encounter if the actual excitation voltage was not measured and shows the use of a 6 wire full bridge to measure a load cell on a weighing lysimeter (a container buried in the ground, filled with plants and soil, used for measuring evapotranspiration).

The lysimeter is 2 meters in diameter and 1.5 meters deep. The total weight of the lysimeter with its container is approximately 8000 kg. The lysimeter has a mechanically adjustable counterbalance, and changes in weight are measured with a 250 pound (113.6 kg) capacity Sensotec Model 41 tension/compression load cell. The load cell has a 4:1 mechanical advantage on the lysimeter (ie., a change of 4 kg in the mass of the lysimeter will change the force on the load cell by 1 kg-force or 980 N).

The surface area of the lysimeter is 3.1416 $m^2$ or 31,416 $cm^2$, so 1 cm of rainfall or evaporation results in a 31.416 kg change in mass. The load cell can measure ±113.6 kg, a 227 kg range. This represents a maximum change of 909 kg, or 28 cm of water in the lysimeter before the counterbalance would have to be readjusted.



**FIGURE 7.15-1. Diagrammatic Representation of Lysimeter Weighing Mechanism**

There is 1000 feet of 22 AWG cable between the CR7 and the load cell. The output of the load cell is directly proportional to the excitation voltage. When Instruction 6 (4 wire 1/2 bridge) is used, the assumption is that the voltage drop in the connecting cable is negligible. The average resistance of 22 AWG wire is 16.5 ohms per 1000 feet. Thus, the resistance in the excitation lead going out to the load cell added to that in the lead coming back to ground is 33 ohms. The resistance of the bridge in the load cell is 350 ohms. The voltage drop across the load cell is equal to the voltage at the CR7 multiplied by the ratio of the load cell resistance $R_s$, to the total resistance, $R_T$, of the circuit. If Instruction 6 were used to measure the load cell, the excitation voltage actually applied to the load cell, $V_1$ would be:

$$V_1 = V_x R_s/R_T = V_x 350/(350+33) = 0.91 V_x$$

Where $V_x$ is the voltage applied at the excitation card. This means that the voltage output by the load cell would only be 91% of that expected. If recording of the lysimeter data was initiated with the load cell output at 0 volts, and 100 mm of evapotranspiration had occurred, calculation of the change with Instruction 6 would indicate that only 91 mm of water had been lost. Because the error is a fixed percentage of the output, the actual magnitude of the error increases with the force applied to the load cell. If the resistance of the wire was constant, one could correct for the voltage drop with a fixed multiplier. However, the resistance of copper changes 0.4% per $^\circ$C change in temperature. Assume that the cable between the load cell and the CR7 lays on the soil surface and undergoes a 25 $^\circ$C diurnal temperature fluctuation. If the resistance is 33

ohms at the maximum temperature, then, at the minimum temperature, the resistance is:

$$(1-25 \times 0.004)33 \text{ ohms} = 29.7 \text{ ohms}$$

The actual excitation voltage at the load cell is:

$$V_1 = 350/(350+29.7) V_x = .92 V_x$$

The excitation voltage has increased by 1%, relative to the voltage applied at the CR7. In the case where we were recording a 91 mm change in water content, there would be a 1 mm diurnal change in the recorded water content that would actually be due to the change in temperature. Instruction 9 solves this problem by actually measuring the voltage drop across the load cell bridge. The drawbacks to using Instruction 9 are that it requires an extra differential channel and the added expense of a 6 wire cable. In this case the benefits are worth the expense.

The load cell has a nominal full scale output of 3 millivolts per volt excitation. If the excitation is 5 volts, the full scale output is 15 millivolts; thus the ±15 millivolt range is selected. The calibrated output of the load cell is 3.106 mV/$V_1$ at a load of 250 pounds. Output is desired in millimeters of water, with respect to a fixed point. The calibration in mV/$V_1$/mm is:

$$3.106 \text{mV/} V_1/250 \text{lb} \times 2.2 \text{lb/kg} \times$$
$$3.1416 \text{kg/mm}/4 =$$

$$0.02147 \text{mV/} V_1/\text{mm}$$

The reciprocal of this gives the multiplier to convert mV/$V_1$ into millimeters (the result of Instruction 9 is the ratio of the output voltage to the actual excitation voltage multiplied by 1000, which is mV/$V_1$):

$$1/0.02147 \text{mV/} V_1/\text{mm} = 46.583 \text{ mm/mV/} V_1$$

The output from the load cell is connected so that the voltage increases as the mass of the lysimeter increases (if the actual mechanical linkage was as diagrammed in Figure 7.15-1, the output voltage would be positive when the load cell was under tension).

When the experiment is started, the water content of the soil in the lysimeter is approximately 25% on a volume basis. It is decided to use this as the reference, (i.e., 0.25 x 1500mm = 375 mm). The experiment is

started at the beginning of what is expected to be a period during which evapotranspiration exceeds precipitation. Instruction 9 is programmed with the correct multiplier and no offset. After hooking everything up, the counterbalance is adjusted so that the load cell is near the top of its range, this will allow a longer period before readjustment is necessary. The result of Instruction 9 (monitored with the *6 Mode) is 109. The offset needed to give the desired initial value of 375mm is 266. However, it is decided to add this offset in a separate program so that the result of Instruction 9 can be used as a ready reminder of the strain on the load cell (range = ±140mm). When the strain on the load cell nears its rated limits, the counterbalance is readjusted and the offset recalculated to provide a continuous record of the water budget.

The program table has an execution interval of 10 seconds. The average value in millimeters is output to Final Storage (not shown in Table) every hour. The average is used, instead of a sample, in order to cancel out the effects of wind loading on the lysimeter.
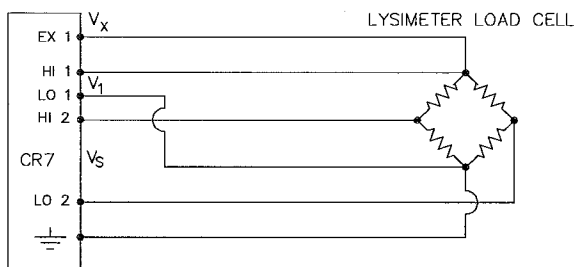


FIGURE 7.15-2.  6 Wire Full Bridge
Connection for Load Cell

**PROGRAM**

| 01: | P9 | Full BR w/Compensation |
|---|---|---|
| 01: | 1 | Rep |
| 02: | 8 | 5000 mV slow EX Range |
| 03: | 3 | 15 mV slow BR Range |
| 04: | 1 | IN Card |
| 05: | 1 | IN Chan |
| 06: | 1 | EX Card |
| 07: | 1 | EX Chan |
| 08: | 1 | Meas/EX |
| 09: | 5000 | mV Excitation |
| 10: | 1 | Loc [:mm RAW   ] |
| 11: | 46.583 | Mult |
| 12: | 0 | Offset |

```
02:     P34        Z=X+F
   01:     1        X Loc mm RAW
   02:   266        F
   03:     2        Z Loc [:mm CORECT]
```

## 7.16  227 GYPSUM SOIL MOISTURE BLOCK

Soil moisture is measured with a gypsum block by relating the change in moisture to the change in resistance of the block.  An AC Half Bridge (Instruction 5) is used to determine the resistance of the gypsum block.  Rapid reversal of the excitation voltage inhibits polarization of the sensor.  Polarization creates an error in the output so the fast integration time is used.  The output of Instruction 5 is the ratio of the mid-bridge voltage to the excitation voltage, this output is converted to gypsum block resistance with Instruction 59, Bridge Transform.

The Campbell Scientific 227 Soil Moisture Block uses a Delmhorst gypsum block with a 1 kohm bridge completion resistor.  Using data supplied by Delmhorst, Campbell Scientific has computed coefficients for a 5th order polynomial to convert block resistance to water potential in bars.  There are two polynomials, one to optimize the range from -0.1 to -2 bars and one to cover the range from -0.1 to -15 bars (the minus sign is omitted in the output).  The -0.1 to -2 bar polynomial requires a multiplier of 1 in the Bridge Transform Instruction (result in Kohms) and the -0.1 to -15 bar polynomial requires a multiplier of 0.1 (result in 10,000s of ohms).  The multiplier is a scaling factor to maintain the maximum number of significant digits in the coefficients of the polynomial.

In this example, we wish to make measurements on 12 gypsum blocks and output the final data in bars.  The soil where the moisture measurements are to be made is quite wet at the time the data logging is initiated, but is expected to dry beyond the -2 bar limit of the wet range polynomial.  The dry range polynomial is used, so a multiplier of 0.1 is entered in the bridge transform instruction.

When the water potential is computed it is written over the resistance value.  The potentials are stored in Input locations 1-12 where they may be accessed for output to Final Storage.  If it was desired to retain the resistance values the potential measurements could be stored in locations 13-24 by changing parameter 3 in Instruction 55 to 13.
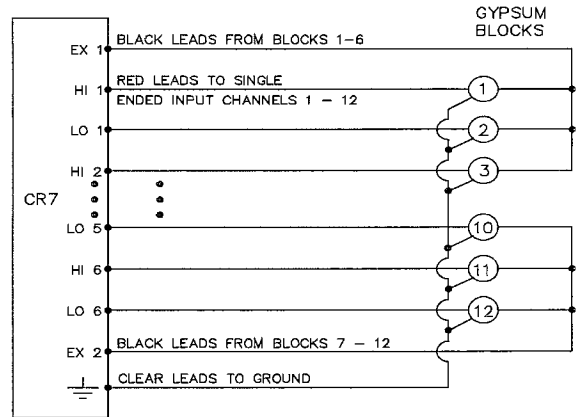


**FIGURE 7.16-1.  12 Gypsum Blocks Connected to the CR7**

The first 6 blocks are excited by excitation channel 1 and the last 6 by channel 2.  Thus, 6 is entered for the number of measurements per excitation channel in Parameter 7 of Instruction 5.

### PROGRAM

```
01:     P5         AC Half Bridge
   01:    12        Reps
   02:    16        500 mV fast Range
   03:     1        IN Card
   04:     1        IN Chan
   05:     1        EX Card
   06:     6        EX Chan
   07:     6        Meas/EX
   08:   500        mV Excitation
   09:     1        Loc [:POTEN #1 ]
   10:     1        Mult
   11:     0        Offset

02:     P59        BR Transform Rf[X/(1-X)]
   01:    12        Reps
   02:     1        Loc [:POTEN #1 ]
   03:     .1        Multiplier (Rf)

03:     P55        Polynomial
   01:    12        Reps
   02:     1        X Loc POTEN #1
   03:     1        F(X) Loc [:POTEN #1 ]
   04:    .15836    C0
   05:   6.1445     C1
   06:  -8.4139     C2
   07:   9.2493     C3
   08:  -3.1685     C4
   09:    .33392    C5
```

## 7.17 NONLINEAR THERMISTOR IN HALF BRIDGE (CAMPBELL SCIENTIFIC MODEL 101)

Instruction 11, 107 Thermistor Probe, automatically linearizes the output of the nonlinear thermistor in the 107 Probe by transforming the millivolt reading with a 5th order polynomial.  Instruction 55, Polynomial, can be used to linearize the output of any nonlinear thermistor, provided the correlation between temperature and probe output is known, and an appropriate polynomial fit has been determined.  In this example, the CR7 is used to measure the temperature of 5 Campbell Scientific 101 Probes (used with the CR21). Instruction 4, Excite, Delay and Measure, is used because the high source resistance of the probe requires a long input settling time (See Section 13.3.1).  The excitation voltage is 2000 mV, the same as used in the CR21. The signal voltage is then transformed to temperature using the Polynomial Instruction.

The manual for the 101 Probe gives the coefficients of the 5th order polynomial used to convert the output in millivolts to temperature (E denotes the power of 10 by which the mantissa is multiplied):

| | |
|---|---|
| C0 | -53.7842 |
| C1 | 0.147974 |
| C2 | -2.18755E-4 |
| C3 | 2.19046E-7 |
| C4 | -1.11341E-10 |
| C5 | 2.33651E-14 |

The CR7 will only allow 5 significant digits to the right or left of the decimal point to be entered from the key board.  The polynomial can not be applied exactly as given in the 101 manual. The initial millivolt reading must be scaled if the coefficients of the higher order terms are to be entered with the maximum number of significant digits.  If 0.001 is used as a multiplier on the millivolt output, the coefficients are divided by 0.001 raised to the appropriate power, (i.e., C0=C0, C1=C1/0.001, C2=C2/.000001 etc.). With this adjustment, the coefficients entered in Parameters 4-9 of Instruction 55 become:

| | |
|---|---|
| C0 | -53.784 |
| C1 | 147.97 |
| C2 | -218.76 |
| C3 | 219.05 |
| C4 | -111.34 |
| C5 | 23.365 |
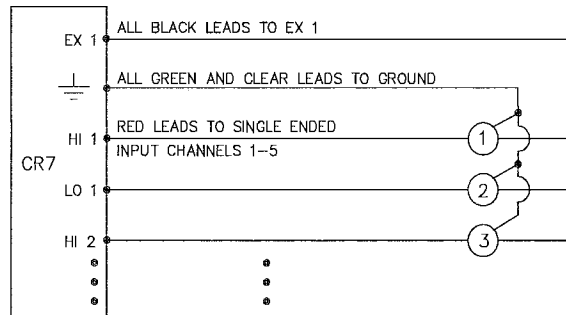


**FIGURE 7.17-1.  101 Thermistor Probes Connected to CR7**

### PROGRAM

| 01: | P4 | Excite,Delay,Volt(SE) |
|---|---|---|
| 01: | 5 | Reps |
| 02: | 8 | 5000 mV slow Range |
| 03: | 1 | IN Card |
| 04: | 1 | IN Chan |
| 05: | 1 | EX Card |
| 06: | 1 | EX Chan |
| 07: | 5 | Meas/EX |
| 08: | 10 | Delay (units .01sec) |
| 09: | 2000 | mV Excitation |
| 10: | 1 | Loc [:101 T #1 ] |
| 11: | 0.001 | Mult |
| 12: | 0 | Offset |

| 02: | P55 | Polynomial |
|---|---|---|
| 01: | 5 | Reps |
| 02: | 1 | X Loc 101 T #1 |
| 03: | 1 | F(X) Loc [:101 T #1 ] |
| 04: | -53.784 | C0 |
| 05: | 147.97 | C1 |
| 06: | -218.76 | C2 |
| 07: | 219.05 | C3 |
| 08: | -111.34 | C4 |
| 09: | 23.365 | C5 |

# SECTION 8. PROCESSING AND PROGRAM CONTROL EXAMPLES

*The following examples are intended to illustrate the use of Processing and Program Control Instructions, flags, and the capability to direct the results of Output Processing Instructions to Input Storage.*

*The specific examples may not be as important as some of the techniques employed, for example:*

*Directing Output Processing to Input Storage is used in the Running Average and Rainfall Intensity examples (8.1 and 8.2).*

*Flags tests are used in the Running Average, Interrupt Subroutine, and Converting Wind Direction examples (8.1, 8.4, and 8.6)*

*These examples are not complete programs to be taken verbatim. They need to be altered to fit specific needs.*

## 8.1 COMPUTATION OF RUNNING AVERAGE

It is sometimes necessary to compute a running average (i.e., the average includes a fixed number of samples and is continuously updated as new samples are taken). Because the output interval is shorter than the averaging period, Instruction 71 cannot be used; the algorithm for computing this average must be programmed by the user. The following example demonstrates a program for computing a running average.

In this example, each time a new measurement is made (in this case a thermocouple temperature) an average is computed for the 10 most recent samples. This is done by saving all 10 temperatures in contiguous input locations and using the Spatial Average Instruction (51) to compute the average. The temperatures are stored in locations 11 through 20. Each time the table is executed, the new measurement is stored in location 20 and the average is stored in location 2. The Block Move Instruction (54) is then used to move the temperatures from locations 12 through 20 down by one location; the oldest measurement (in location 11) is lost when the temperature from location 12 is written over it.

Input Location Labels:

| | |
|---|---|
| 1:Panl Temp | 15:Temp i-5 |
| 2:10smpl av | 16:Temp i-4 |
| 11:Temp i-9 | 17:Temp i-3 |
| 12:Temp i-8 | 18:Temp i-2 |
| 13:Temp i-7 | 19:Temp i-1 |
| 14:Temp i-6 | 20:Temp i |

*Where i is current reading,*
*i-1 is previous reading, etc.*

| | | |
|---|---|---|
| * | 1 | Table 1 Programs |
| 01: | 1 | Sec. Execution Interval |
| | | |
| 01: | P17 | Panel Temperature |
| 01: | 1 | IN Card |
| 02: | 1 | Loc [:Panl Temp] |
| | | |
| 02: | P14 | Thermocouple Temp (DIFF) |
| 01: | 1 | Rep |
| 02: | 1 | 1500 uV slow Range |
| 03: | 1 | IN Card |
| 04: | 1 | IN Chan |
| 05: | 1 | Type T (Copper-Constantan) |
| 06: | 1 | Ref Temp Loc Panl Temp |
| 07: | 20 | Loc [:Temp i   ] |
| 08: | 1 | Mult |
| 09: | 0.0000 | Offset |
| | | |
| 03: | P51 | Spatial Average |
| 01: | 10 | Swath |
| 02: | 11 | First Loc Temp i-9 |
| 03: | 2 | Avg Loc [:10smpl av] |

8-1

| | | |
|---|---|---|
| 04: | P54 | Block Move |
| 01: | 9 | No. of Values |
| 02: | 12 | First Source Loc Temp i-8 |
| 03: | 1 | Source Step |
| 04: | 11 | First Destin. Loc [:Temp i-9 ] |
| 05: | 1 | Destination Step |
| | | |
| 05: | P86 | Do |
| 01: | 10 | Set high Flag 0 (output) |
| | | |
| 06: | P70 | Sample |
| 01: | 1 | Rep |
| 02: | 2 | Loc 10smpl av |
| | | |
| 07: | P | End Table 1 |

In the above example, all samples for the average are stored in input locations. This is necessary when an average must be output with each new sample. In most cases, averages are desired less frequently than sampling. For example, it may be necessary to sample some parameter every five seconds and output every hour an average of the previous three hours' readings. If all samples were saved, this would require 2160 input locations. The same value can be obtained by computing an hourly average and averaging the hourly averages for the past three hours. To do this requires that hourly averages be stored in input locations.

Instruction 80 is used to send the one hour average to Input Storage and again to send the three hour average to Final Storage.

Input Location Labels:

1:AVG i-2
2:AVG i-1
3:AVG i
4:3 HR AVG
5:XX mg/M3

| | | |
|---|---|---|
| * | 1 | Table 1 Programs |
| 01: | 5 | Sec. Execution Interval |
| | | |
| 01: | P2 | Volt (DIFF) |
| 01: | 1 | Rep |
| 02: | 8 | 5000 mV slow Range |
| 03: | 1 | IN Card |
| 04: | 3 | IN Chan |
| 05: | 3 | Loc [:XX mg/m3 ] |
| 06: | 10 | Mult |
| 07: | 0 | Offset |

| | | |
|---|---|---|
| 02: | P92 | If time is |
| 01: | 0 | minutes into a |
| 02: | 60 | minute interval |
| 03: | 10 | Set high Flag 0 (output) |
| | | |
| 03: | P80 | Set Active Storage Area |
| 01: | 3 | Input Storage Area |
| 02: | 3 | Array ID or location |
| | | |
| 04: | P71 | Average |
| 01: | 1 | Rep |
| 02: | 5 | Loc |
| | | |
| 05: | P51 | Spatial Average |
| 01: | 3 | Swath |
| 02: | 1 | First Loc AVG i-2 |
| 03: | 4 | Avg Loc [:3 HR AVG ] |
| | | |
| 06: | P80 | Set Active Storage Area |
| 01: | 1 | Final Storage Area |
| 02: | 25 | Array ID or location |
| | | |
| 07: | P77 | Real Time |
| 01: | 220 | Day,Hour-Minute |
| | | |
| 08: | P70 | Sample |
| 01: | 1 | Rep |
| 02: | 4 | Loc 3 HR AVG |
| | | |
| 09: | P91 | If Flag |
| 01: | 10 | 0 (output) is set |
| 02: | 30 | Then Do |
| | | |
| 10: | P54 | Block Move |
| 01: | 2 | No. of Values |
| 02: | 2 | First Source Loc |
| 03: | 1 | Source Step |
| 04: | 1 | First Destination Loc [:AVG i-2 ] |
| 05: | 1 | Destination Step |
| | | |
| 11: | P95 | End |
| | | |
| 12: | P | End Table 1 |

## 8.2 RAINFALL INTENSITY

In this example, the total rain for the last 15 minutes is output only if any rain has occurred. The program makes use of the capability to direct the output of Output Processing Instructions to Input Storage.

Every 15 minutes, the total rain is sent to Input Storage. If the total is greater than 0, output is redirected to Final Storage, the time is output, and the total is sampled.

Input Location Labels:

1:Rain (mm)
2:15min tot

| * | 1 | Table 1 Programs |
|---|---|---|
| 01: | 60 | Sec. Execution Interval |
| | | |
| 01: | P3 | Pulse |
| 01: | 1 | Rep |
| 02: | 3 | IN Card |
| 03: | 1 | Pulse Input Chan |
| 04: | 2 | Switch closure |
| 05: | 1 | Loc [:Rain (mm)] |
| 06: | .254 | Mult |
| 07: | 0 | Offset |
| | | |
| 02: | P92 | If time is |
| 01: | 0 | minutes into a |
| 02: | 15 | minute interval |
| 03: | 10 | Set high Flag 0 (output) |
| | | |
| 03: | P80 | Set Active Storage Area |
| 01: | 3 | Input Storage Area |
| 02: | 2 | Array ID or location |
| | | |
| 04: | P72 | Totalize |
| 01: | 1 | Rep |
| 02: | 1 | Loc Rain (mm) |
| | | |
| 05: | P89 | If X<=>F |
| 01: | 2 | X Loc |
| 02: | 3 | >= |
| 03: | 0 | F |
| 04: | 30 | Then Do |
| | | |
| 06: | P80 | Set Active Storage Area |
| 01: | 1 | Final Storage Area |
| 02: | 25 | Array ID or location |
| | | |
| 07: | P77 | Real Time |
| 01: | 220 | Day,Hour-Minute |
| | | |
| 08: | P70 | Sample |
| 01: | 1 | Rep |
| 02: | 2 | Loc |
| | | |
| 09: | P95 | End |

## 8.3 SUB 1 MINUTE OUTPUT INTERVAL SYNCHED TO REAL TIME

Instruction 92 has one minute resolution. If processed output is required on an interval less than one minute, Instructions 18 and 89 can be used to set the Output Flag on a shorter interval.

Instruction 18 takes time (tenths of seconds into minute, minutes into day, or hours into year), performs a modulo divide by a user specified value and loads it into an input location.

When the modulo divisor divides evenly into the interval, one gets a counter in an input location that goes to 0 on a periodic interval. In this example, tenths of seconds into the minute is modulo divided by 300. The counter counts up to 295 then goes to 0 (i.e., every 30 seconds; tenths of seconds into minute has a resolution of 0.1 seconds).

Instruction 89 is used to set the Output Flag when the tenths of seconds counter is less than 5 (the execution interval, 0.5 seconds). With this short program, the Output Flag could be set when the seconds counter equaled 0. However, if Instruction 18 followed a series of instructions that took longer than 0.1 seconds to execute or was in Table 2, executed at the same interval as an extensive Table 1, the time at which Instruction 18 was executed might be 0.1 seconds or more beyond the modulo divisor. The value output would not equal 0. Setting the Output Flag when the seconds counter is less than the execution interval avoids this problem.

Using Instruction 18 keeps the output interval synchronized with real time. If a counter incremented within the program was used to determine when to set the Output Flag, output would depend on the number of times the table was executed. The actual time of output would depend on when the program was actually compiled and started running. If the table overran its execution interval (Section 1.1.1), the output interval would not be the count multiplied by the execution interval, but some longer interval.

In this example a temperature (type E thermocouple) is measured every 0.5 seconds and the average output every 30 seconds.

Input Location Assignments:

1:TEMP DEG C
10:30 SEC 0

| * | 1 | Table 1 Programs |
|---|---|---|
| 01: | .5 | Sec. Execution Interval |
| | | |
| 01: | P18 | Time |
| 01: | 0 | Tenths of seconds into minute (maximum 600) |
| 02: | 300 | Mod/by |
| 03: | 10 | Loc [:30 SEC 0 ] |
| | | |
| 02: | P17 | Panel Temperature |
| 01: | 1 | IN Card |
| 02: | 1 | Loc [:REF TEMP ] |
| | | |
| 03: | P14 | Thermocouple Temp (DIFF) |
| 01: | 1 | Rep |
| 02: | 13 | 15 mV fast Range |
| 03: | 1 | IN Card |
| 04: | 2 | IN Chan |
| 05: | 2 | Type E (Chromel-Constantan) |
| 06: | 1 | Ref Temp Loc REF TEMP |
| 07: | 2 | Loc [:TC TEMP ] |
| 08: | 1 | Mult |
| 09: | 0 | Offset |
| | | |
| 04: | P | 89  If X<=>F |
| 01: | 10 | X Loc 30 SEC 0 |
| 02: | 4 | < |
| 03: | .5 | F |
| 04: | 10 | Set high Flag 0 (output) |
| | | |
| 05: | P | 71  Average |
| 01: | 1 | Rep |
| 02: | 2 | Loc TC TEMP |
| | | |
| 06: | P | End Table 1 |

## 8.4 ANALOG OUTPUT TO STRIP CHART

This example illustrates the use of the Analog Output Instruction 21 to output 2 analog voltages to strip chart.

While of questionable value because of power requirements and strip chart reliability, some archaic regulations require strip chart backup on weather data.   Instruction 21 may be used with the CR7 to provide two continuous analog

outputs for strip charts.  The output values in this example are wind speed and wind direction.

The following program measures the sensors every five seconds.  The readings are moved to another two locations and scaled to a 0 to 1000 millivolt output for the strip chart.  Wind direction is changed from a 0-360 degree input to output representing 0 to 540 degrees.  This conversion is done in a subroutine which is described in the next example.

The example also includes instructions to output wind vector every hour.

Input Location Assignments:

01:WS
02:0-360 WD
03:0-540 WD
04:WS output
05:WD output

| * | 1 | | Table 1 Programs |
|---|---|---|---|
| 01: | 3 | | Sec. Execution Interval |
| | | | |
| 01: | P3 | | Pulse |
| 01: | 1 | | Rep |
| 02: | 5 | | IN Card |
| 03: | 1 | | Pulse Input Chan |
| 04: | 22 | | Switch closure; Output Hz. |
| 05: | 1 | | Loc [:WS     ] |
| 06: | 1.789 | | Mult |
| 07: | 1 | | Offset |
| | | | |
| 02: | P4 | | Excite,Delay,Volt(SE) |
| 01: | 1 | | Rep |
| 02: | 16 | | 500 mV fast Range |
| 03: | 2 | | IN Card |
| 04: | 1 | | IN Chan |
| 05: | 1 | | EX Card |
| 06: | 1 | | EX Chan |
| 07: | 1 | | Meas/EX |
| 08: | 2 | | Delay (units .01sec) |
| 09: | 1000 | | mV Excitation |
| 10: | 2 | | Loc [:0-360 WD ] |
| 11: | .72 | | Mult |
| 12: | 0 | | Offset |
| | | | |
| 03: | P86 | | Do |
| 01: | 1 | | Call Subroutine 1 |

| | | |
|---|---|---|
| 04: | P37 | Z=X*F |
| 01: | 1 | X Loc WS |
| 02: | 10 | F |
| 03: | 4 | Z Loc [:WS output] |
| | | |
| 05: | P37 | Z=X*F |
| 01: | 3 | X Loc 0-540 WD |
| 02: | 1.8519 | F |
| 03: | 5 | Z Loc [:WD output] |
| | | |
| 06: | P21 | Analog Out |
| 01: | 1 | EX Card |
| 02: | 1 | CAO Chan |
| 03: | 4 | mv Loc WS output |
| | | |
| 07: | P21 | Analog Out |
| 01: | 1 | EX Card |
| 02: | 2 | CAO Chan |
| 03: | 5 | mv Loc WD output |
| | | |
| 08: | P92 | If time is |
| 01: | 0 | minutes into a |
| 02: | 60 | minute interval |
| 03: | 10 | Set high Flag 0 (output) |
| | | |
| 09: | P69 | Wind Vector |
| 01: | 1 | Rep |
| 02: | 180 | Samples per sub-interval |
| 03: | 00 | Polar Sensor/(S, D1, SD1) |
| 04: | 1 | Wind Speed/East Loc WS |
| 05: | 2 | Wind Dir./North Loc 0-360 WD |
| | | |
| 10: | P | End Table 1 |

## 8.5 CONVERTING 0-360 WIND DIRECTION OUTPUT TO 0-540 FOR STRIP CHART

If 0-360 degree wind direction is output to a strip chart, the discontinuity at 0/360 will cause the pen to jump back and forth full scale when the winds are varying from the north. In the days of strip charts this was solved with a 0-540 degree pot on the wind vane (direction changes from 540 to 180 and from 0 to 360 so the pen only jumps once when the wind is out of the north or south).

When faced with the necessity of strip chart output (see previous example), the following algorithm can be used to change a 0-360 degree input to 0-540. (If you have a 0-540 pot, it can be used with the 21X since the Wind Vector Instruction, 69, will work with this output.)

To change 0-360 degrees to the 0-540 degrees, 360 degrees must sometimes be added to the reading when it is in the range of 0 to 180. The following algorithm does this by assuming that if the previous reading was less than 270, the vane has shifted through 180 degrees and does not need to be altered. If the previous 0-540 reading was greater than 270, 360 degrees is added.

This example is written as a subroutine which is used by the previous example to output an analog voltage to a strip chart.

Input Location Labels:

1:WS
2:0-360 WD
3:0-540 WD
4:WS output
5:WD output

| | | |
|---|---|---|
| * | 3 | Table 3 Subroutines |
| | | |
| 01: | P85 | Beginning of Subroutine |
| 01: | 1 | Subroutine Number |
| | | |
| 02: | P89 | If X<=>F |
| 01: | 3 | X Loc 0-540 WD |
| 02: | 3 | >= |
| 03: | 270 | F |
| 04: | 30 | Then Do |
| | | |
| 03: | P86 | Do |
| 01: | 11 | Set high Flag 1 |
| | | |
| 04: | P94 | Else |
| | | |
| 05: | P86 | Do |
| 01: | 21 | Set low Flag 1 |
| | | |
| 06: | P95 | End |
| | | |
| 07: | P31 | Z=X |
| 01: | 2 | X Loc 0-360 WD |
| 02: | 3 | Z Loc [:0-540 WD ] |
| | | |
| 08: | P89 | If X<=>F |
| 01: | 3 | X Loc 0-540 WD |
| 02: | 4 | < |
| 03: | 180 | F |
| 04: | 30 | Then Do |

```
09:     P91     If Flag
  01:     11      1 is set
  02:     30      Then Do

10:     P34     Z=X+F
  01:      3      X Loc 0-540 WD
  02:    360      F
  03:      3      Z Loc [:0-540 WD ]

11:     P95     End

12:     P95     End

13:     P95     End

14:     P       End Table 3
```

## 8.6  COVARIANCE CORRELATION PROGRAMMING EXAMPLE

The example is a two level meteorological tower with five sensors at each level.  The three components of the wind are measured using prop anemometers.  Two thermocouples (TC) are used to measure ambient and wet-bulb temperatures and calculate water vapor pressure on-line.  All sensors are scanned once per second (1 Hz) and a five minute averaging period with a 30 minute Output Interval is specified.  The example optimizes the input measurement sequence for speed and shows the instructions necessary to provide calibrated inputs, properly ordered to produce the desired outputs from the Covariance Correlation (CV/CR) Instruction.  Table 8.7-1 groups the sensors according to measurement type and gives the CR7 multiplier and offset.

The props can all be measured as single-ended voltages, but the vertical wind prop calibration differs from the U and V prop calibration.  The fastest input sequence is to measure both levels (6 props) with a single instruction using the U and V calibration and correct the W measurements with the Fixed Multiply, Instruction 37.

The type E thermocouples are measured on the most sensitive input range, 5mV, accommodating a $\pm80\,^{\circ}$C range between the measurement and CR7 reference junction.  The resolution is $(.33\mu V/(60\mu V/^{\circ}C)$ or about 0.006 $^{\circ}$C.  Measuring absolute temperature with TCs requires a reference junction temperature measurement.  This is measured with Instruction 17.

The specified outputs determine the input order required by the CV/CR Instruction.  Table 8.6-2 lists the desired outputs from the two levels along with the Input Storage locations for the processed results.

### TABLE 8.6-1. Example Sensor Description and CR7 Multiplier and Offset

| DESCRIPTION | SYMBOL | SENSOR | CALIB | MEAS TYPE | MULT | OFFSET |
|---|---|---|---|---|---|---|
| Horiz. Wind | U | prop | 18m/s/V | S.E.V. | .018m/s/mV | 0.0 |
| Horiz. Wind | V | prop | 18m/s/V | S.E.V. | .018 | 0.0 |
| Vert. Wind | W | prop | 22m/s/V | S.E.V. | .022 | 0.0 |
| Air Temp. | Ta | TC | - | TC DIFF. | 1.0$^{\circ}$C | 0.0 |
| Wet-bulb Temp. | Tw | TC | - | TC DIFF. | 1.0$^{\circ}$C | 0.0 |
| Vap. Pressure | e | derived | - | - | - | - |

**TABLE 8.6-2.  Example Outputs and Input Storage Locations**

## LEVEL 1 OUTPUTS

| MEANS LOC | | VARIANCES LOC | | COVARIANCES LOC | | CORRELATIONS LOC | |
|---|---|---|---|---|---|---|---|
| M(W1) | 20 | V(W1) | 25 | CV(W1,U1) | 30 | CR(W1,U1) | 34 |
| M(U1) | 21 | V(U1) | 26 | CV(W1,V1) | 31 | CR(W1,V1) | 35 |
| M(V1) | 22 | V(V1) | 27 | CV(W1,Tal) | 32 | | |
| M(Tal) | 23 | V(Tal) | 28 | CV(W1,e1) | 33 | | |
| M(e1) | 24 | V(e1) | 29 | | | | |

## LEVEL 2 OUTPUTS

| MEANS LOC | | VARIANCES LOC | | COVARIANCES LOC | |
|---|---|---|---|---|---|
| M(W2) | 36 | V(W2) | 41 | CV(W2,U2) | 46 |
| M(U2) | 37 | V(U2) | 42 | CV(W2,V2) | 47 |
| M(V2) | 38 | V(V2) | 43 | CV(W2,Ta2) | 48 |
| M(Ta2) | 39 | V(Ta2) | 44 | CV(W2,e2) | 49 |
| M(e2) | 40 | V(e2) | 45 | CV(U2,V2) | 50 |
| | | | | CV(U2,Ta2) | 51 |
| | | | | CV(U2,e2) | 52 |
| | | | | CV(V2,Ta2) | 53 |
| | | | | CV(V2,e2) | 54 |

Table 8.6-3 lists the input channel configuration and Input Storage allocation for the measured values.  After reading the new input samples, the Level 2 measurements are relocated using the Block Move Instruction 54, then Ta1 is relocated through a separate move and e1 is positioned by specifying the destination location in the Wet/Dry-Bulb Instruction.  The CV/CR Instruction must be entered twice, once for each level.

In addition to ordering Level 1 and Level 2 in locations 1-5 and 11-15 respectively, two more locations are required.  Converting the wet-/dry-bulb measurements to vapor pressure using Instruction 57 requires atmospheric pressure.  We'll use the standard atmosphere for the site elevation and key the value into Location 17 using the C command in the *6 Mode.  The reference junction temperature obtained by Instruction 17 is stored in Location 16.

This example requires that 54 locations be allotted to Input Storage and 79 to Intermediate Storage (35 for the 1st CV/CR Instruction, 43 for the second, and 1 for Instruction 92).

**TABLE 8.6-3.  Example Input Channel and Location Assignments**

| PARAM | INPUT CHAN | INPUT LOC | | PARAM | INPUT LOC | | PARA | INPUT LOC |
|---|---|---|---|---|---|---|---|---|
| W1 | 1 | 1 | | W1 | 1 | | W1 | 1 |
| U1 | 2 | 2 | | U1 | 2 | | U1 | 2 |
| V1 | 3 | 3 | | V1 | 3 | | V1 | 3 |
| W2 | 4 | 4 | | Ta1 | 9 ---------------------- | | Ta1 | 4 |
| U2 | 5 | 5 | | Tw1 | 10 Separate moves | | e1 | 5 |
| V2 | 6 | 6 | | W2 | 11 | | W2 | 11 |
| Ta2 | 7 | 7 | ->Block-> | U2 | 12 | | U2 | 12 |
| Tw2 | 8 | 8 | move | V2 | 13 | | V2 | 13 |
| Ta1 | 9 | 9 | | Ta2 | 14 | | Ta2 | 14 |
| Tw1 | 10 | 10 | | Tw2 | 15 | | e2 | 15 |

| | | |
|---|---|---|
| * | 1 | Table 1 Programs |
| 01: | 1 | Sec. Execution Interval |
| | | |
| 01: | P17 | Panel Temperature |
| 01: | 1 | IN Card |
| 02: | 16 | Loc [:PANL TEMP] |
| | | |
| 02: | P1 | Volt (SE) |
| 01: | 6 | Reps |
| 02: | 8 | 5000 mV slow Range |
| 03: | 1 | IN Card |
| 04: | 1 | IN Chan |
| 05: | 1 | Loc [:W1     ] |
| 06: | .018 | Mult |
| 07: | 0 | Offset |
| | | |
| 03: | P14 | Thermocouple Temp (DIFF) |
| 01: | 4 | Reps |
| 02: | 15 | 150 mV fast Range |
| 03: | 1 | IN Card |
| 04: | 7 | IN Chan |
| 05: | 2 | Type E (Chromel-Constantan) |
| 06: | 16 | Ref Temp Loc PANL TEMP |
| 07: | 7 | Loc [:Ta2     ] |
| 08: | 1 | Mult |
| 09: | 0 | Offset |
| | | |
| 04: | P37 | Z=X*F |
| 01: | 1 | X Loc W1 |
| 02: | 1.22 | F |
| 03: | 1 | Z Loc [:W1     ] |
| | | |
| 05: | P37 | Z=X*F |
| 01: | 4 | X Loc W2 |
| 02: | 1.22 | F |
| 03: | 4 | Z Loc [:W2     ] |
| | | |
| 06: | P54 | Block Move |
| 01: | 5 | No. of Values |
| 02: | 4 | First Source Loc W2 |
| 03: | 1 | Source Step |
| 04: | 11 | First Destination Loc [:W2     ] |
| 05: | 1 | Destination Step |
| | | |
| 07: | P31 | Z=X |
| 01: | 9 | X Loc |
| 02: | 4 | Z Loc [:W2     ] |
| | | |
| 08: | P57 | Wet/Dry Bulb Temp to VP |
| 01: | 17 | Pressure Loc |
| 02: | 9 | Dry Bulb Temp Loc |
| 03: | 10 | Wet Bulb Temp Loc |
| 04: | 5 | Loc [:U2     ] |

| | | |
|---|---|---|
| 09: | P57 | Wet/Dry Bulb Temp to VP |
| 01: | 17 | Pressure Loc |
| 02: | 14 | Dry Bulb Temp Loc |
| 03: | 15 | Wet Bulb Temp Loc Tw2 |
| 04: | 15 | Loc [:Tw2     ] |
| | | |
| 10: | P92 | If time is |
| 01: | 0 | minutes into a |
| 02: | 30 | minute interval |
| 03: | 10 | Set high Flag 0 (output) |
| | | |
| 11: | P62 | CV/CR |
| 01: | 5 | No. of Input Values |
| 02: | 5 | No. of Means |
| 03: | 5 | No. of Variances |
| 04: | 0 | No. of Std. Dev. |
| 05: | 4 | No. of Covariances |
| 06: | 2 | No. of Correlations |
| 07: | 300 | Samples per Average |
| 08: | 1 | First Sample Loc W1 |
| 09: | 20 | Loc [:MEAN (W1)] |
| | | |
| 12: | P62 | CV/CR |
| 01: | 5 | No. of Input Values |
| 02: | 5 | No. of Means |
| 03: | 5 | No. of Variances |
| 04: | 0 | No. of Std. Dev. |
| 05: | 4 | No. of Covariances |
| 06: | 2 | No. of Correlations |
| 07: | 300 | Samples per Average |
| 08: | 11 | First Sample Loc W2 |
| 09: | 36 | Loc [:MEAN (W2)] |
| | | |
| 13: | P77 | Real Time |
| 01: | 110 | Day,Hour-Minute |
| | | |
| 14: | P70 | Sample |
| 01: | 35 | Reps |
| 02: | 20 | Loc MEAN (W1) |
| | | |
| 15: | P | End Table 1 |
| | | |
| * | A | Mode 10 Memory Allocation |
| 01: | 54 | Input Locations |
| 02: | 79 | Intermediate Locations |

# SECTION 9. INPUT/OUTPUT INSTRUCTIONS

## TABLE 9-1. Input Voltage Ranges and Codes

| Range Code | | Full Scale Range | | Resolution* | |
|---|---|---|---|---|---|
| Slow 16.67ms Integ. | Fast 250µs Integ. | | | | |
| 1 | 11 | ±1500 | microvolts | 50 | nanovolts |
| 2 | 12 | ±5000 | microvolts | 166 | nanovolts |
| 3 | 13 | ±15 | millivolts | 500 | nanovolts |
| 4 | 14 | ±50 | millivolts | 1.66 | microvolts |
| 5 | 15 | ±150 | millivolts | 5 | microvolts |
| 6 | 16 | ±500 | millivolts | 16.6 | microvolts |
| 7 | 17 | ±1500 | millivolts | 50 | microvolts |
| 8 | 18 | ±5000 | millivolts | 166 | microvolts |

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

When measuring voltages with the 723 Analog Input Card, the ±1500 µV and ±5000 µV ranges read out in microvolts, the rest of the ranges have the decimal point placed to display millivolts. The 726 50V Analog Input Card divides the input voltages by 10 before making the measurements, thus, to shift the decimal point so as to display millivolts a factor of ten must be used in the multiplier. Repetitions cannot be used to advance from one 726 card to the next.

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

### *** 1 SINGLE ENDED VOLTS ***

FUNCTION
This instruction is used to measure voltage at a single ended input with respect to ground.

| PAR. NO. | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 2 | Repetitions |
| 02: | 2 | Range code (Table 9-1) |
| 03: | 2 | Card number for first measurement |
| 04: | 2 | Single-ended channel number for first measurement |
| 05: | 4 | Input location number for first measurement |
| 06: | FP | Multiplier |
| 07: | FP | Offset |

Input locations altered: 1 per repetition

### *** 2 DIFFERENTIAL VOLTS ***

FUNCTION
This Instruction reads the voltage difference between the HI and LO inputs of a differential channel in the selected range from the selected card and channel(s) and places it in an input location(s). Table 9-1 lists the range codes. Both the high and low inputs must be within ±5V of the datalogger's ground (Common Mode Range Section 13.2). Pyranometers and thermopile sensors require a jumper between LO and Ground to keep them in Common Mode Range.

| PAR. NO. | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 2 | Repetitions |
| 02: | 2 | Range code (Table 9-1) |
| 03: | 2 | Card number for first measurement |
| 04: | 2 | Differential channel number for first measurement |
| 05: | 4 | Input location number for first measurement |
| 06: | FP | Multiplier |
| 07: | FP | Offset |

Input locations altered: 1 per repetition

### *** 3 PULSE COUNT ***

INPUT RANGE - - - 32767 Counts per input interval

There are three input configurations which may be measured with the Pulse Count Instruction.

HIGH FREQUENCY PULSE
In this configuration the minimum pulse width is 2 microseconds. The maximum input frequency is 250 kilohertz. The count is incremented when the input voltage changes from below 1.5 volts to above 3.5 volts. The maximum input voltage is ±20 volts.

LOW LEVEL AC
This configuration is used for counting the frequency of AC signals from magnetic pulse flow transducers or other low voltage, sine wave inputs. The minimum input voltage is 6 mV RMS. Input hysteresis is 11 mV. The maximum AC input voltage is 20 volts RMS. The maximum input frequency ranges from 100Hz at 6mV RMS to 10,000Hz at 50mV or greater. Consult the factory if higher frequencies are desired.

SWITCH CLOSURE
In this configuration the minimum switch closed time is 1 millisecond. The minimum switch open time is 4 milliseconds. The maximum bounce time is 1.4 milliseconds open without being counted.

All pulse counters in one I/O Module are reset at the same time. The reset time interval is equal to the execution interval of the program table in which the Pulse Count Instruction(s) occur. When programs are compiled, the CR7 will set the reset time interval to the execution interval of the first program table in which a Pulse Count Instruction occurs. The execution intervals of subsequent program tables containing Pulse Count Instructions will have no effect on the reset time interval. (The maximum input frequency is 250KHz. The maximum number that can be stored in an accumulation register is 65,535.)

When datalogger time is changed, whether through the keyboard or with a telecommunications program, a partial recompile is automatically done to resynchronize program execution with real time. The resynchronization process resets the pulse accumulation interval resulting in an interval whose length can be anywhere between one second too short to almost twice as long. Pulses are not lost during resynchronization so totalized values are correct but pulse rate information such as wind speed can be up to almost twice the correct value.

The options of discarding counts from long intervals and pulse input type are selected by the code entered for the 4th parameter (Table 9-2).

All Pulse count instructions for the same I/O module and output instructions for the pulse count data should be kept in the same program table, preferably Table 1. If the pulse count instruction is contained in a subroutine, that subroutine must be called from Table 2.

When the system is interrupted for a task of sufficient priority to allow the pulse counters to exceed the programmed reset time interval, the resulting larger count can either be discarded leaving the value in the input location unchanged from the previous value or it can be used. If pulse counts are being totalized, a missing count could be significant; the value from the erroneously long interval should be used. If the pulse count is being processed a way in which the resultant value is dependent upon the sampling interval, such as sample, average, maximum, or minimum, it should be discarded. The option of discarding counts from long intervals and the input configuration are determined by the 4th parameter according to the following table.

| | | |
|---|---|---|
| 09: | FP | Excitation voltage (millivolts) |
| 10: | 4 | Input location number for first measurement |
| 11: | FP | Multiplier |
| 12: | FP | Offset |

Input locations altered: 1 per repetition

### *** 5 AC HALF BRIDGE ***

FUNCTION
This Instruction is used to apply an excitation voltage to a half bridge (Figure 13.5-1), make a single ended voltage measurement of the bridge output, reverse the excitation voltage, then repeat the measurement. The difference between the two measurements is used to calculate the resulting value which is the ratio of the measurement to the excitation voltage. When the 1500 or 5000 µV input range is selected, the value is returned as 1000 times the ratio. For all other input ranges the value is just the ratio.

The excitation "on time" for each polarity is exactly the same to ensure that ionic sensors do not polarize with repetitive measurements. The range should be selected to be a fast measurement (range 11-18) limiting the excitation on time to 800 microseconds at each polarity. A slow integration time should not be used with ionic sensors because of polarization error.

| PAR. NO. | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 2 | Repetitions |
| 02: | 2 | Range code (Table 9-1) |
| 03: | 2 | Analog card number for first measurement |
| 04: | 2 | Single-ended channel number for first measurement |
| 05: | 2 | Excitation card for first measurement |
| 06: | 2 | Excitation channel number for first measurement |
| 07: | 2 | Number of measurements per excitation channel |
| 08: | 4 | Excitation voltage (millivolts) |
| 09: | 4 | Input location number for first measurement |
| 10: | FP | Multiplier |
| 11: | FP | Offset |

Input locations altered: 1 per repetition

---

### TABLE 9-2. Pulse Count Configuration Codes

| Code | Configuration |
|---|---|
| 00 | High frequency pulse, all pulses counted |
| 01 | Low level AC, all pulses counted |
| 02 | Switch closure, all pulses counted |
| 1X | Long interval data discarded, where X is configuration code |
| 2X | Long interval data discarded, frequency (Hz) output |

| PAR. NO. | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 2 | Repetitions |
| 02: | 2 | Card number for first measurement |
| 03: | 2 | Pulse channel number for first measurement |
| 04: | 2 | Configuration code (Table 9-2) |
| 05: | 4 | Input location number for first measurement |
| 06: | FP | Multiplier |
| 07: | FP | Offset |

Input locations altered: 1 per repetition

### *** 4 EXCITE, DELAY AND MEASURE **

FUNCTION
This Instruction is used to apply an excitation voltage, delay a specified time and then make a single ended voltage measurement.

| PAR. NO. | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 2 | Repetitions |
| 02: | 2 | Range code (Table 9-1) |
| 03: | 2 | Analog card number for first measurement |
| 04: | 2 | Single-ended channel number for first measurement |
| 05: | 2 | Excitation card for first measurement |
| 06: | 2 | Excitation channel number for first measurement |
| 07: | 2 | Number of measurements per excitation channel |
| 08: | 4 | Delay (0.01 sec) |

## *** 6 FULL BRIDGE WITH SINGLE *** DIFFERENTIAL MEASUREMENT

**FUNCTION**
This Instruction is used to apply an excitation voltage to a full bridge (Figure 13.5-1), make a differential voltage measurement of the bridge output, reverse the excitation voltage, then repeat the measurement. The resulting value is 1000 times the ratio of the measurement to the excitation voltage.

| PAR. NO. | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 2 | Repetitions (95 max) |
| 02: | 2 | Range code (Table 9-1) |
| 03: | 2 | Analog card number for first measurement |
| 04: | 2 | Differential channel number for first measurement |
| 05: | 2 | Excitation card for first measurement |
| 06: | 2 | Excitation channel number for first measurement |
| 07: | 2 | Number of measurements per excitation channel |
| 08: | 4 | Excitation voltage (millivolts) |
| 09: | 4 | Input location number for first measurement |
| 10: | FP | Multiplier |
| 11: | FP | Offset |

Input locations altered: 1 per repetition

## *** 7 THREE WIRE HALF BRIDGE ***

**FUNCTION**
This Instruction is used to determine the ratio of the sensor resistance to a known resistance using a separate voltage sensing wire from the sensor to compensate for lead wire resistance.

The measurement sequence is to apply an excitation voltage, make two voltage measurements on two adjacent single ended channels, the first on the reference resistor and the second on the voltage sensing wire from the sensor (Figure 13.5-1), then reverse the excitation voltage and repeat the measurements. The two measurements are used to calculate the resulting value which is the ratio of the voltage across the sensor to the voltage across the reference resistor.

| PAR. NO. | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 2 | Repetitions (95 max) |
| 02: | 2 | Range code for both measurements (Table 9-1) |
| 03: | 2 | Analog card number for first measurement |
| 04: | 2 | Single-ended channel number for first measurement |
| 05: | 2 | Excitation card for first measurement |
| 06: | 2 | Excitation channel number for first measurement |
| 07: | 2 | Number of measurements per excitation channel |
| 08: | 4 | Excitation voltage (millivolts) |
| 09: | 4 | Input location number for first measurement |
| 10: | FP | Multiplier |
| 11: | FP | Offset |

Input locations altered: 1 per repetition

## *** 9 FULL BRIDGE WITH EXCITATION *** COMPENSATION

**FUNCTION**
This Instruction is used to apply an excitation voltage and make two differential voltage measurements, then reverse the polarity of the excitation and repeat the measurements. The measurements are made on sequential channels. The result is the voltage measured on the second channel ($V_2$) divided by the voltage measured on the first ($V_1$). If $V_1$ is measured on the 5V range (code 8 or 18 in Parameter 2), then the result is 1000 times $V_2/V_1$. A 1 before the excitation channel number (1X) causes the excitation channel to be incremented with each repetition.

When used as a 6 wire full bridge (Figure 13.5-1), the connections are made so that $V_1$ is the measurement of the voltage drop across the full bridge, and $V_2$ is the measurement of the bridge output. Because the excitation voltage for a full bridge measurement is usually in the 5V range, the output is usually 1000 $V_2/V_1$ or millivolts output per volt excitation. When used to measure a 4 wire half bridge, the connections are made so that $V_1$ is the voltage drop across the fixed resistor ($R_f$), and $V_2$ is the drop across the sensor ($R_s$). As long as $V_1$ is not measured on the 5V range, the result is $V_2/V_1$ which equals $R_s/R_f$.

| PAR. NO. | DATA TYPE | DESCRIPTION |
| --- | --- | --- |
| 01: | 2 | Repetitions (47 max) |
| 02: | 2 | Excitation range code (Table 9-1) |
| 03: | 2 | Bridge range code for (Table 9-1) |
| 04: | 2 | Analog card number for first measurement |
| 05: | 2 | Differential channel number for first measurement |
| 06: | 2 | Excitation card for first measurement |
| 07: | 2 | Excitation channel number for first measurement |
| 08: | 2 | Number of measurements per excitation channel |
| 09: | 4 | Excitation voltage (millivolts) |
| 10: | 4 | Input location number for first measurement |
| 11: | FP | Multiplier |
| 12: | FP | Offset |

Input locations altered: 1 per repetition

### *** 10 BATTERY VOLTAGE ***

FUNCTION
This instruction reads the battery voltage from the currently active I/O module and writes it to an input location. The units for battery voltage are volts.

| PAR. NO. | DATA TYPE | DESCRIPTION |
| --- | --- | --- |
| 01: | 4 | Input location number |

Input locations altered: 1

### *** 11 107 THERMISTOR PROBE ***

FUNCTION
This instruction applies a 4 VAC excitation voltage to Campbell Scientific's Model 107 Thermistor Probe, makes a fast, single ended voltage measurement on the 15 mV range across a resistor in series with the thermistor and calculates the temperature in $^{\circ}$C with a polynomial. The maximum polynomial error from -40 $^{\circ}$C to +55 $^{\circ}$C is given below:

Curve Fit Error --

| Range ($^{\circ}$C) | Error ($^{\circ}$C) |
| --- | --- |
| -40 to +55 | ±1.0 |
| -35 to +48 | ±0.1 |

This instruction uses a single excitation channel since several hundred probes can be driven by a single excitation output. For this reason, Instruction 11 does not require a "measurement/excitation" parameter.

A multiplier of 1 and an offset of 0 yields temperature in degrees C.

| PAR. NO. | DATA TYPE | DESCRIPTION |
| --- | --- | --- |
| 01: | 2 | Repetitions |
| 02: | 2 | Analog card number for first measurement |
| 03: | 2 | Single-ended channel number for first measurement |
| 04: | 2 | Excitation card number for first measurement |
| 05: | 2 | Excitation channel number for first measurement |
| 06: | 4 | Input location of first measurement |
| 07: | FP | Multiplier |
| 08: | FP | Offset |

Input locations altered: 1 per repetition

### *** 12 207 RELATIVE HUMIDITY PROBE ***

FUNCTION
This instruction applies a 3 VAC excitation across Campbell Scientific's Model 207 Temperature and RH Probe, makes a fast single ended measurement across a series resistor on the 150 mV range, linearizes the result with a 5th order polynomial and performs the required temperature compensation before outputting the result in % RH.

When measuring several probes, all the RH elements should be connected sequentially. Any temperature elements used for compensating the respective RH value should also be sequentially connected to make use of the REP feature in Instruction 11.

> The temperature value used in compensating the RH value (Parameter 7) must be obtained (see Instruction 11) prior to executing Instruction 12.

The RH results are placed sequentially into the input locations beginning with the first RH value. In some situations the RH sensors might be deployed such that only small temperature variations exist within a given set of probes. In these cases a single temperature may be used to compensate the subset of RH measurements instead of making a temperature measurement for each RH probe. If the complete set of temperature values are not needed, this approach uses less input channels. Parameter 6 is used to specify how many consecutive RH values get compensated per temperature measurement.

In the 207 probe, the RH and temperature elements use a common excitation line. Since a single excitation channel can drive several hundred probes, there is no "measurements/excitation" parameter in Instruction 12. NEVER EXCITE THE 207 PROBE WITH DC EXCITATION as the RH chip will be damaged.

The maximum RH polynomial error is given below:

Curve Fit Error --

| Range (%RH) | Error (%RH) |
| --- | --- |
| 10 - 100 | ±4 |
| 15 - 94 | ±1 |

| PAR. NO. | DATA TYPE | DESCRIPTION |
| --- | --- | --- |
| 01: | 2 | Repetitions |
| 02: | 2 | Analog card number for first measurement |
| 03: | 2 | Single-ended channel for first measurement |
| 04: | 2 | Excitation card number for first measurement |
| 05: | 2 | Excitation channel number for first measurement |
| 06: | 2 | Number of R.H. measurements per compensating temperature measurement |
| 07: | 4 | Input location for first compensating temperature measurement |
| 08: | 4 | Input location for first measurement |
| 09: | FP | Multiplier |
| 10: | FP | Offset |

Input locations altered:  1 per repetition

## *** 13 THERMOCOUPLE ***
## TEMPERATURE, SINGLE ENDED

FUNCTION
This Instruction uses the selected thermocouple calibration to calculate the thermocouple output voltage at the reference temperature, then it makes a SINGLE ENDED VOLTAGE MEASUREMENT (Section 13.2) on the thermocouple and adds the measured voltage to the calculated reference voltage, then converts the voltage to temperature in $^{\circ}$C (Section 13.4).

Table 9-3 gives the thermocouple type codes. The reference temperature location will be incremented by one each repetition if C is keyed before entering Parameter 6. When this option is exercised, two minus signs (--) will appear as the right most characters of the display.

A multiplier of 1 and an offset of 0 yields temperature in degrees C.

### TABLE 9-3.  Thermocouple Type Codes

| Code | Thermocouple Type |
| --- | --- |
| 1 | T (copper - constantan) |
| 2 | E (chromel - constantan) |
| 3 | K (chromel - alumel) |
| 4 | J (iron - constantan) |
| 5 | B (platinum - rhodium) |
| 6 | R (platinum - rhodium) |
| 7 | S (platinum - rhodium) |
| 1X | Output temperature difference between Reference and Thermocouple |
| 2X | Skip every other single ended channel |
| 8X | TC input from A5B40 Isolation Amplifier (use 5 V range) |

| PAR. NO. | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 2 | Repetitions |
| 02: | 2 | Range code (Table 9-1) |
| 03: | 2 | Analog card number |
| 04: | 2 | Single-ended channel number for first measurement |
| 05: | 2 | TC type code (Table 9-3) |
| 06: | 4 | Reference temperature location. (When indexed (--) this is incremented with each rep.) |
| 07: | 4 | Input location number |
| 08: | FP | Multiplier |
| 09: | FP | Offset |

Input locations altered:   1 per repetition

### *** 14 THERMOCOUPLE ***
### TEMPERATURE, DIFFERENTIAL MEASUREMENT

FUNCTION
This instruction calculates the thermocouple temperature for the thermocouple type selected. The instruction specifies a DIFFERENTIAL VOLTAGE MEASUREMENT (Section 13.2) on the thermocouple, adds the measured voltage to the voltage calculated for the reference temperature relative to 0 $^{\circ}$C, and converts the combined voltage to temperature in $^{\circ}$C. The differential inputs are briefly shorted to ground prior to making the voltage measurement to insure that they are within the common mode range. (Section 13.4)

Table 9-3 gives the thermocouple type codes for Parameter 5, the option of skipping every other channel applies only to Instruction 13. The reference temperature location will be incremented by one each repetition if **C** is keyed before entering Parameter 6. When this option is exercised, two minus signs (--) will appear as the right most characters of the display.

A multiplier of 1 and an offset of 0 yields temperature in degrees C.

| PAR. NO. | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 2 | Repetitions |
| 02: | 2 | Range code (Table 9-1) |
| 03: | 2 | Card number |
| 04: | 2 | Beginning channel |
| 05: | 2 | TC type code (Table 9-3) |

| PAR. NO. | DATA TYPE | DESCRIPTION |
|---|---|---|
| 06: | 4 | Reference temperature location. (When indexed (--) this is incremented with each rep.) |
| 07: | 4 | Input location number |
| 08: | FP | Multiplier |
| 09: | FP | Offset |

Input locations altered:   1 per repetition

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

A multiplier of 1 and an offset of 0 yields temperature in degrees C.

| PAR. NO. | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 2 | Repetitions |
| 02: | 4 | Input location number of (R/Ro) |
| 03: | 4 | Input location number of result |
| 04: | FP | Multiplier |
| 05: | FP | Offset |

Input locations altered:   1 per repetition

### *** 17 TEMPERATURE OF INPUT PANEL ***

FUNCTION
This instruction measures the temperature in $^{\circ}$C of the specified analog input card with RTD (Model 723-T).

| PAR. NO. | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 2 | Analog card number |
| 02: | 4 | Input location number |

Input locations altered:  1

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

FUNCTION
This instruction takes the current time in tenths of seconds into the minute, minutes into the day, or hours into the year and does a modulo divide (see Instruction 46) on the time value with the number specified in the second parameter. The result is stored in the specified input location. Entering 0 or a number which is greater than the maximum value of the time for the modulo divide will result in the actual time value being stored.

PARAMETER 1 OPTION CODES

CODE   TIME UNITS

0      Tenths of seconds into minute
       (maximum 600)
1      Minutes into current day (maximum
       1440)
2      Hours into current year (maximum
       8784)

| PAR. NO. | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 2 | Option Code (see above) |
| 02: | 4 | Number to modulo divide by |
| 03: | 4 | Input location number |

Input locations altered: 1

## *** 19 MOVE SIGNATURE INTO *** INPUT LOCATION

FUNCTION
This instruction stores the signature of the Read Only Memory (ROM) and user program memory (RAM) into an input location. This signature is not the same as the signatures given in the *B mode. Recording the signature allows detection of any program change or ROM failure.

| PAR. NO. | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 4 | Input location number |

Input locations altered: 1

## *** 20 PORT SET ***

FUNCTION
This instruction sets a specified Digital Control output port or is used to set the active excitation card for port commands with Program Control Instructions or manual toggling (Section 1.3.3). Ports may be set as specified by a flag or unconditionally.

PARAMETER 1 OPTION CODES

Code   Function

00     Set port low
01     Set port high
1X     Set port according to flag X
2X     Set opposite to flag X
30     Set active port card

| PAR. NO. | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 2 | Option code (see above) |
| 02: | 2 | Excitation card number |
| 03: | 2 | Port number (1-8) |

Input locations altered: 0

## *** 21 ANALOG OUTPUT ***

FUNCTION
This instruction sets the continuous Analog Output (CAO) to a voltage level specified in an input location. The analog output degrades approximately 0.17mV every seven seconds requiring the instruction to be periodically repeated to maintain a given output accuracy.

| PAR. NO. | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 2 | Excitation card number |
| 02: | 2 | CAO channel number |
| 03: | 4 | Input location number containing analog output magnitude in millivolts |

Input locations altered: 0
Input locations read: 1

## *** 22 EXCITATION WITH DELAY ***

FUNCTION
This instruction is used in conjunction with others for measuring a response to a timed excitation using the switched analog outputs. It sets the selected excitation output to a specific value, waits for a specified time, turns off the excitation and waits an additional specified time before continuing execution of the following instruction. The excitation on time can be set to zero and the off time delay can be used if the only requirement is the delay of Program execution.

> This instruction cannot be interrupted by Program Table 1 in order to make a measurement. This means that if it resides in Table 2 or Table 3 then Table 1 may be delayed.

| PAR. NO. | DATA TYPE | DESCRIPTION |
| --- | --- | --- |
| 01: | 2 | Excitation card number |
| 02: | 2 | Excitation channel number |
| 03: | 4 | Delay that excitation is on (0.01 sec) |
| 04: | 4 | Delay time after excitation is turned off (0.01 sec) |
| 05: | FP | Excitation voltage (millivolts) |

Input locations altered: 0

## *** 23 SELECT I/O MODULE ***

FUNCTION
This instruction is used when more than one I/O Module is under control and is used to specify which I/O Module subsequent instructions refer to. The I/O Module to which Instructions are addressed defaults to #1 at the start of each program table.

| PAR. NO. | DATA TYPE | DESCRIPTION |
| --- | --- | --- |
| 01: | 2 | Module number (1,2,3 or 4) |

Input locations altered: 0

## *** 26 TIMER ***

FUNCTION
This instruction will reset a timer or store the elapsed time registered by the timer in an Input Storage location. Instruction 26 can be used with Program Control Instructions to measure the elapsed time between specific input conditions. There is only one timer and it is common to all tables (e.g., if the timer is reset in Table 1 and later in Table 2, a subsequent instruction in Table 1 to read the timer will store the elapsed time since the timer was reset in Table 2).

Elapsed time is tracked in 0.1 second increments but displayed as an integer. For example, a 20 second elapsed time is displayed as "200".

The timer is also reset in response to certain keyboard entries:

1. When tables are changed and compiled with the *0 Mode, the timer is reset automatically.

2. When tables are changed and then compiled in the *B Mode, the timer is automatically reset and Tables 1 and 2 are disabled. Entering "*0" at this point enables both tables and resets the timer.

3. Entering "*6" after changing the tables compiles the programs, but does NOT reset the timer.

| PAR. NO. | DATA TYPE | DESCRIPTION |
| --- | --- | --- |
| 01: | 4 | Input location number (enter 0 to reset) |

Input locations altered: 1 (0 if timer is being reset)

## *** 101 SDM-INT8 ***

The 8 channel Interval Timer (INT8) is a measurement module which provides processed timing information to the datalogger. Each of the 8 input channels may be independently configured to detect either rising or falling edges of either high level or a low level signal. Each channel may be independently programmed. See the SDM-INT8 manual for detailed

instructions and examples. This instruction is not in all PROM options.

| PARAM. NUMBER | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 2 | SDM address (base 4:00..33) |
| 02: | 4 | *Input configuration; channels 8,7,6,5 |
| 03: | 4 | *Input configuration; channels 4,3,2,1 |
| 04: | 4 | **Function; channels 8,7,6,5 |
| 05: | 4 | **Function; channels 4,3,2,1 |
| 06: | 4 | ***Output option |
| 07: | 4 | Starting input location number |
| 08: | FP | Mult |
| 09: | FP | Offset |

  *   Input configurations:
      0 = high level, rising edge
      1 = high level, falling edge
      2 = low level, rising edge
      3 = low level, falling edge

 **   Functions:
      0 = no value returned
      1 = period in ms
      2 = frequency in kHz
      3 = time since previous channel's edge in ms
      4 = time since channel 1 in ms
      5 = counts on channel 2 since channel 1, linear interpolation
      6 = frequency in kHz (low resolution)
      7 = counts
      8 = counts on Channel 2 since Channel 1, no interpolation

*** Output option:

| 0 | Average over execution interval |
|---|---|
| 0-- | Continuous averaging |
| XXXX | Averaging interval in msec, XXXX>0 |
| XXXX-- | Capture all events until XXXX edges of channel 1 (0<XXXX,9999) |
| 9999-- | Test memory |

Input locations altered: 1 per function

*** 102 SDM-SW8A ***

The 8 channel SDM-SW8A Switch Closure Input Module is a peripheral for measuring up to 8 channels of switch closure or voltage pulse inputs. Each channel may be configured to read single-pole double-throw (SPDT) switch closure, or single-pole single-throw (SPST) switch closure, or voltage pulse. Output options include counts, duty cycle, and state. This instruction is not in all PROM options.

The SW8A is addressed by the datalogger, allowing multiple SW8A's to be connected to one datalogger. 16 addresses are available.

If more channels are requested than exist in one module, the datalogger automatically increments the address and continues to the next SW8A. The address settings for multiple SW8A's must sequentially increase. For example, assume 2 SW8A's addressed as 22 and 23 are connected, and 12 Reps are requested. 8 channels from the first SW8A and the first 4 channels from the next will be read.

Only one Function Option (**Parameter 3**) may be specified per Instruction 102. If all 4 functions are desired, the instruction must be entered 4 times.

Function Option 0 provides the state of the signal at the time P102 is executed. A 1 or 0 corresponds to high or low states, respectively.

Function Option 1 provides signal duty cycle. The result is the percentage of time the signal is high during the sample interval.

Function Option 2 provides a count of the number of positive transitions of the signal.

Function Option 3 provides the signature of the SW8A PROM. A positive number (signature) indicates the PROM and RAM are good, a zero (0) indicates bad PROM, and a negative number indicates bad RAM. Function Option 3 is not used routinely, but is helpful in "debugging". Only one Rep is required for Option 3.

**Parameter 4** specifies the first SW8A channel to be read (1..8). One or more sequential channels are read depending on the Reps. To optimize program efficiency, the sensors should be wired sequentially.

Data are stored in sequential datalogger input locations, starting at the location specified in **Parameter 5**. The number of input locations consumed is equal to the number of Reps.

The scaling multiplier and offset (**Parameters 6 and 7**) are applied to all readings. If a multiplier is not entered, all readings are set to 0.

If the SW8A does not respond, -99999 will be loaded into input locations. Modules which do not respond when addressed by the datalogger may be wired or addressed incorrectly. Verify that the address specified in Parameter 2 corresponds to the jumper setting and that all connections are correct and secure. See the SDM-SW8A Manual for examples.

| PARAM. NUMBER | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 2 | Repetitions (number of channels) |
| 02: | 2 | SDM Address (base 4:00..33) |
| 03: | 2 | Function Option (0=State, 1=Duty 2=Counts, 3=Signature) |
| 04: | 2 | SDM-SW8A Starting Channel (1..8) |
| 05: | 4 | Starting input location number |
| 06: | FP | Mult |
| 07: | FP | Offset |

Input locations altered: 1 per repetition

### *** 103 SDM-AO4 ***

Instruction 103 is used to activate a SDM-AO4 4 Channel Continuous Analog Output Module connected to ports C1, C2, and C3. This instruction is not in all PROM options.

There are 4 analog voltage outputs per SDM-AO4. The output voltages in millivolts must be stored in 4 adjacent input locations starting with the location entered in parameter 4. Four reps are required for each SDM-A04. Every 4 reps another device at the next higher address is selected.

| PARAM. NUMBER | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01 | 2 | Repetitions (number of outputs) |
| 02 | 2 | SDM address (base 4:00..33) |
| 03 | 4 | Starting input location number |

C1 is Data line
C2 is Clk/Hand Shake line
C3 is SDE (Enable) line

Input locations read: 1 per repetition

### *** 104 SDM-CD16AC ***

The SDM-CD16AC Control Port Expansion Module has 16 digital control ports with drivers. Each port can be controlled by a datalogger or controlled manually with an override toggle switch. Each port can be thought of as a switch to ground; closed when active, open when inactive. The primary function is to activate DC powered external relays, solenoids, or resistive loads under datalogger control. This instruction is not in all PROM options.

The SDM-CD16AC is a synchronously addressed datalogger peripheral. Datalogger control ports 1, 2 and 3 are used to address the SDM-CD16AC then clock out the desired state of each of the 16 control ports. Up to 16 SDM-CD16AC 's may be addressed, making it possible to control a maximum of 256 ports from the first three datalogger control ports.

For each Rep, the 16 ports of the addressed SDM-CD16AC are set according to 16 sequential input locations starting at the input location specified in parameter 3. Any non-zero value stored in an input location activates (connects to ground) the associated SDM-CD16AC port. A value of zero (0) deactivates the port (open circuit). For example, assuming 2 Reps and a starting input location of 33, OUTPUT 1 through 16 of the first SDM-CD16AC are set according to Input Locations 33 through 48, and OUTPUT 1 through 16 of the second SDM-CD16AC are set according to Input Location 49 through 64. See the SDM-CD16AC manual for detailed instructions and examples.

# SECTION 9. INPUT/OUTPUT INSTRUCTIONS

| PARAM. NUMBER | DATA TYPE | DESCRIPTION |
|---|---|---|
| 1 | 2 | Reps (# of CD16AC modules sequentially addressed) |
| 2 | 2 | Starting SDM address (base 4: 00..33) |
| 3 | 4 | Starting input location number |

Input locations read: 16 per repetition

## *** 113 SDM-SI04 ***

FUNCTION
Instruction 113 communicates with the SDM-SI04 serial input multiplexer. See the SDM-SI04 manual for directions.

## *** 114 SET TIME ***

FUNCTION
Instruction 114 can be used to set the CR7 clock from values in input locations.

| PARAM. NUMBER | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 2 | Option code:<br>0   set time with hr,min,sec with values in 3 input locations.<br>1   set time with day,hr,min,sec using 4 input locations.<br>2   set time with yr,day,hr,min,sec using 5 input locations. |
| 02: | 4 | Input location number |

Input locations read: 3-5 depending on option

## *** 115 SET SDM BAUD ***

FUNCTION
Instruction 115 may be used to set the SDM communication rate. This may be necessary when communicating over longer cable lengths. The default bit period is 10 microseconds (entering either 0 or 1 will result in this period).

| PARAM. NUMBER | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 4 | Bit period, 10$\mu$s units |

Normally this parameter represents the bit period. If the parameter is indexed (--), the value entered is an Input Location that contains the bit period to use.

> NOTE: The SDM-SI04 Instruction 113 automatically adjusts the SDM communication rate to the fastest that will work.

## *** 118 SDM-OBD2 ***

FUNCTION
Instruction 118 is used to read sensor values in from an On-Board Data Acquisition II device. The On-Board Data Acquisition II device is used to measure different functions of a vehicle's engine.

| PARAM NUMBER | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 2 | SDM Address (base 4:00..33) |
| 02: | 2 | # of PIDs (Number of values to store) |
| 03: | 4 | Starting Input Location |

Instruction 118 must be followed by Instruction(s) 63 or 68. An E68 error will be displayed when Instruction 118 is not followed enough Instruction 63 or 68. If an "E68 104" appears, this means that the Instruction 118 is the 3rd instruction in Table 1 and that not enough Instruction 63/68 are following it. The instructions are used to list the sensor numbers in the order one wants to store the information. For example, to store the first 5 readings and the 12th reading in a specific order (4th, 3rd, 1st, 5th, 12th, and 2nd), use the below Instruction 63:

```
P63
  1:      4
  2:      3
  3:      1
  4:      5
  5:      12
  6:      2
  7:      0
  8:      0
```

Input Locations Altered: 1 per # of PIDs

# SECTION 10. PROCESSING INSTRUCTIONS

To facilitate cross referencing, parameter descriptions are keyed [] to the values given on the PROMPT SHEET. These values are defined as follows:

[Z] = 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, entered via the keyboard)

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

FUNCTION
Store a fixed value into input location Z.

PAR. DATA
NO. TYPE  DESCRIPTION

| 01: | FP | Fixed data | [F] |
| 02: | 4 | Destination for input location | [Z] |

Input locations altered: 1

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

FUNCTION
Move data from one input location to another.

PAR. DATA
NO. TYPE  DESCRIPTION

| 01: | 4 | Input location of X | [X] |
| 02: | 4 | Destination for input location | [Z] |

Input locations altered: 1

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

FUNCTION
Add 1 to the current value in input location Z.

PAR. DATA
NO. TYPE  DESCRIPTION

| 01: | 4 | Destination for input location | [Z] |

Input locations altered: 1

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

FUNCTION
Add the value in Input location X to the value in location Y and place the result in location Z.

PAR. DATA
NO. TYPE  DESCRIPTION

| 01: | 4 | Input location of X | [X] |
| 02: | 4 | Input location of Y | [Y] |
| 03: | 4 | Dest. input location of X + Y | [Z] |

Input locations altered: 1

### *** 34 X + F ***

FUNCTION
Add the fixed number F to the value in location X and place the result in location Z.

PAR. DATA
NO. TYPE  DESCRIPTION

| 01: | 4 | Input location of X | [X] |
| 02: | FP | Value to add | [F] |
| 03: | 4 | Dest. input location of X + F | [Z] |

Input locations altered: 1

### *** 35 X - Y ***

FUNCTION
Subtract the value in location Y from the value in location X and place the result in location Z.

PAR. DATA
NO. TYPE  DESCRIPTION

| 01: | 4 | Input location of X | [X] |
| 02: | 4 | Input location of Y | [Y] |
| 03: | 4 | Dest. input location for X - Y | [Z] |

Input locations altered: 1

# SECTION 10. PROCESSING INSTRUCTIONS

### *** 36  X * Y ***

FUNCTION
Multiply the value in location X by the value in
location Y and place the result in location Z.

PAR. DATA
NO.  TYPE  DESCRIPTION

01:   4     Input location of X              [X]
02:   4     Input location of Y              [Y]
03:   4     Dest. input location for X * Y   [Z]

Input locations altered:  1

### *** 37  X * F ***

FUNCTION
Multiply the value in location X by the fixed
number F and place the result in location Z.

PAR. DATA
NO.  TYPE  DESCRIPTION

01:   4     Input location of X              [X]
02:   FP    Fixed multiplier                 [F]
03:   4     Dest. input location for X * F   [Z]

Input locations altered:  1

### *** 38  X / Y ***

FUNCTION
Divide the value in location X by the value in
location Y and place the result in location Z.
Division by 0 will cause the result to be set to
the maximum CR7 number (+99999 if positive, -
99999 negative).

PAR. DATA
NO.  TYPE  DESCRIPTION

01:   4     Input location of X              [X]
02:   4     Input location of Y              [Y]
03:   4     Dest. input location for X / Y   [Z]

Input locations altered:  1

### *** 39 SQUARE ROOT ***

FUNCTION
Take the square root of the value in location X
and place the result in location Z.  If the value in
X is negative, 0 will be stored as the result.

PAR. DATA
NO.  TYPE  DESCRIPTION

01:   4     Input location of X              [X]
02:   4     Dest. input location for $X^{1/2}$   [Z]

Input locations altered:  1

### *** 40  LN(X) ***

FUNCTION
Take the natural logarithm of the value in
location X and place the result in location Z.  If
the value in X is 0 or negative, -99999 will be
stored as the result.

PAR. DATA
NO.  TYPE  DESCRIPTION

01:   4     Input location of X              [X]
02:   4     Dest. input location for LN(X)   [Z]

Input locations altered:  1

### *** 41  EXP(X) ***

FUNCTION
Raise the exponential base e to the value in
location X power and place the result in location
Z.

PAR. DATA
NO.  TYPE  DESCRIPTION

01:   4     Input location of X              [X]
02:   4     Dest. input location for EXP(X)  [Z]

Input locations altered:  1

### *** 42  1/X ***

FUNCTION
Take the inverse of the value in location X and
place the result in location Z.  If X=0, 99999 will
be stored as the result.

PAR. DATA
NO.  TYPE  DESCRIPTION

01:   4     Input location of X              [X]
02:   4     Dest. input location for 1/X     [Z]

Input locations altered:  1

### *** 43 ABS(X) ***

FUNCTION
Take the absolute value of the value in location X and place the result in location Z.

PAR. DATA
NO.  TYPE  DESCRIPTION

| 01: | 4 | Input location of X | [X] |
| 02: | 4 | Dest. input location for ABS(X) | [Z] |

Input locations altered: 1

### *** 44 FRACTIONAL VALUE ***

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

PAR. DATA
NO.  TYPE  DESCRIPTION

| 01: | 4 | Input location of X | [X] |
| 02: | 4 | Dest. input location for FRAC(X) | [Z] |

Input locations altered: 1

### *** 45 INTEGER VALUE ***

FUNCTION
Take the integer portion of the value in location X and place the result in location Z.

PAR. DATA
NO.  TYPE  DESCRIPTION

| 01: | 4 | Input location of X | [X] |
| 02: | 4 | Dest. input location for INT(X) | [Z] |

Input locations altered: 1

### *** 46  X MOD F ***

FUNCTION
Do a modulo divide of the value in location X by the fixed value F and place the result in location Z.  X MOD F is defined as the REMAINDER obtained when X is divided by F (e.g., 3 MOD 2 = 1).  X MOD 0 returns X.

PAR. DATA
NO.  TYPE  DESCRIPTION

| 01: | 4 | Input location of X | [X] |
| 02: | FP | Fixed divisor | [F] |
| 03: | 4 | Dest. input location for X MOD F | [Z] |

Input locations altered: 1

### *** 47  $X^Y$ ***

FUNCTION
Raise the value in location X to the value in location Y power and place the result in location Z.

PAR. DATA
NO.  TYPE  DESCRIPTION

| 01: | 4 | Input location of X | [X] |
| 02: | 4 | Input location of Y | [Y] |
| 03: | 4 | Dest. input location for $X^Y$ | [Z] |

Input locations altered: 1

### *** 48 SIN(X) ****

FUNCTION
Calculate the sine of the value in location X (assumed to be in degrees) and place the result in location Z.  The cosine of a number can be obtained by adding 90 to the number and taking the sine (COSX = SIN (X + 90)).

PAR. DATA
NO.  TYPE  DESCRIPTION

| 01: | 4 | Input location of X | [X] |
| 02: | 4 | Dest. input location for SIN(X) | [Z] |

Input locations altered: 1

### *** 49 SPATIAL MAXIMUM ***

FUNCTION
Find the maximum value in the given set or SWATH of contiguous input locations and place the result in location Z.  To find the input location where the maximum value occurs, enter 1000 + the input location number (1000 + Z) as Parameter 03.  The input location 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 Instruction 31 (Move Data). In Instruction 31, enter the location in which 49 stores its result as the source (fixed) and enter the destination as an indexed location.

PAR. DATA
NO.  TYPE  DESCRIPTION

| 01: | 2 | Swath | [SWATH] |
| 02: | 4 | Starting input location | [1ST LOC] |
| 03: | 4 | Dest. input location for maximum | [MAX or Z] |

Input locations altered:  1 or 2

### *** 50  SPATIAL MINIMUM ***

FUNCTION
Find the minimum value in the given set or SWATH of contiguous input locations and place the result in location Z.  To find the input location where the minimum value occurs, follow the instructions given above for SPATIAL MAXIMUM.

Parameter 3 cannot be entered as an indexed location in a loop.  Within a loop, Instruction 50 must be used in conjunction with Instruction 31 as described for Instruction 49.

PAR. DATA
NO.  TYPE  DESCRIPTION

| 01: | 2 | Swath | [SWATH] |
| 02: | 4 | Starting input location | [1ST LOC] |
| 03: | 4 | Dest. input location for minimum | [MIN or Z] |

Input locations altered:  1 or 2

### *** 51  SPATIAL AVERAGE ***

FUNCTION
Calculate the average of the values in the given set or SWATH of contiguous input locations and place the result in location Z.

PAR. DATA
NO.  TYPE  DESCRIPTION

| 01: | 2 | Swath | [SWATH] |
| 02: | 4 | Starting input location | [1ST LOC] |
| 03: | 4 | Dest. input location for average | [AVG or Z] |

Input locations altered:  1

### *** 53  SCALING ARRAY WITH *** MULTIPLIER AND OFFSET

FUNCTION
Take 4 input location values, multiply each by a floating point constant, then add another floating point constant to the resulting products and place the results back into each of the original 4 input locations.

PAR. DATA
NO.  TYPE  DESCRIPTION

| 01: | 4 | First input location | [STRT LOC] |
| 02: | FP | Multiplier 1 | [A1] |
| 03: | FP | Offset 1 | [B1] |
| 04: | FP | Multiplier 2 | [A2] |
| 05: | FP | Offset 2 | [B2] |
| 06: | FP | Multiplier 3 | [A3] |
| 07: | FP | Offset 3 | [B3] |
| 08: | FP | Multiplier 4 | [A4] |
| 09: | FP | Offset 4 | [B4] |

Input locations altered:  4

### *** 54  BLOCK MOVE ***

FUNCTION
Moves a block of data from one set of input locations to another.  Parameters specify the number of values to move, the source, source step, destination, and destination step.  The "step" parameters designate the increment of the source and destination Input locations for each value that is moved.  For example, a "source step" of 2 and a "destination step" of 1 will move data from every other Input location to a contiguous block of Input locations.

PAR. DATA
NO. TYPE DESCRIPTION

01: 4 Number of values to move
02: 4 1st source location
03: 2 Step of source
04: 4 1st destination location
05: 2 Step of destination

Input locations altered: number of values to
move

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

FUNCTION
Evaluate a 5th order polynomial of the form.

$$F(X)=C0+C1X+C2X^2+C3X^3+C4X^4+C5X^5$$

where C0 through C5 are the coefficients for the
argument X raised to the zero through fifth
power, respectively. The magnitude of the user
entered coefficient is limited to a range of
+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.

PAR. DATA
NO. TYPE DESCRIPTION

01: 2 Repetitions [REPS]
02: 4 Starting input location for X [X]
03: 4 Dest. input location for F(X)
[F(X) or Z]
04: FP C0 coefficient [C0]
05: FP C1 coefficient [C1]
06: FP C2 coefficient [C2]
07: FP C3 coefficient [C3]
08: FP C4 coefficient [C4]
09: FP C5 coefficient [C5]

Input locations altered: 1 per repetition

### *** 56 SATURATION VAPOR PRESSURE ***

FUNCTION
Calculate saturation vapor pressure over water
(SVPW) in kilopascals from the air temperature
($^o$C) and place it in an input location. The
algorithm for obtaining SVPW from air
temperature ($^o$C) is taken from: Lowe, Paul R.,

1976: 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 $^o$C to -50 $^o$C range can be
obtained using Instruction 55 and the
relationship

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

where X is the SVPW derived by Instruction 56.
This relationship was derived by Campbell
Scientific from the equations for the SVPW and
the SVPI given in Lowe's paper.

PAR. DATA
NO. TYPE DESCRIPTION

01: 4 Input location of air temperature $^o$C
[TEMP.]
02: 4 Dest. input location for saturated
vapor pressure [VP or Z]

Input locations altered: 1

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

FUNCTION
This instruction calculates vapor pressure in
kilopascals from wet- and dry-bulb
temperatures in $^o$C. The algorithm is of the
type used by the National Weather Service:

VP = VPW - A(1 + B*TW)(TA - TW) P
VP = ambient vapor pressure in
kilopascals
VPW = saturation vapor pressure at the wet-
bulb temperature in kilopascals
TW = wet-bulb temperature, deg. C
TA = ambient air temperature, deg. 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.

## SECTION 10. PROCESSING INSTRUCTIONS

PAR. DATA
NO. 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     Dest. input location for vapor
           pressure              [VP or Z]

Input locations altered: 1

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

FUNCTION
Apply a numerical approximation to an analog
resistor capacitor (RC) low pass (LP) filter using
the following algorithm:

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

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

On the first execution after compiling, F(x) is set
equal to X.

PAR. DATA
NO. TYPE  DESCRIPTION

01:  2     Repetitions              [REPS]
02:  4     First input location for input data
                                       [X]
03:  4     Dest. input location for filtered
           data                 [F(X) or Z]
04:  FP    Weighting function, W        [W]

Input locations altered: 1 per repetition

### *** 59 BRIDGE TRANSFORM ***

FUNCTION
This instruction is used to aid in the conversion
of a ratiometric Bridge measurement by
obtaining the value for $R_s$ which is equivalent to
$R_f[X/(1-X)]$, where X is the value derived by the
standard CR7 Bridge Measurement Programs
(with appropriate multiplier and offset, Section
13.5) and $R_f$ represents the MULTIPLIER value.
The result of Instruction 59 is stored in the
same location that X was.

PAR. DATA
NO. TYPE  DESCRIPTION

01:  2     Repetitions              [REPS]
02:  4     Starting input location and
           destination                 [X]
03:  FP    Multiplier ($R_f$)       [MULT.]

Input locations altered: 1 per repetition

### *** 61 INDIRECT INDEXED MOVE ***

FUNCTION
Moves input data from location X to location Y,
where X and Y are indirectly addressed. The
values of the location numbers X and Y are
stored in the locations specified by Parameters
1 and 2. The CR7 looks in the locations
specified in the parameters to find the locations
to use as the source and destination of the data.
When used within a LOOP, a location
parameter can be 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.

PAR. DATA
NO. TYPE  DESCRIPTION

01:  4     Input location containing source
           location X
02:  4     Input location containing destination
           location Y

### *** 62 COVARIANCE/CORRELATION ***

FUNCTION
The special Covariance/Correlation Instruction
(CV/CR) for the CR7 calculates: 1) means (M),
2) variances (V), 3) standard deviations (SD), 4)
covariances (CV), and 5) correlations (CR) for a
set of input values and stores the results in

Input Storage. The instruction requires the set of input values to be located contiguously in Input Storage. The user specifies the location of the first value and how many total values exist. The number of input values processed by each type of calculation (means, variances, etc.) is independently specified for each type. The order of the input values determines which inputs are processed for each type of calculation.

The instruction does not conform to the CR7's four instruction types. Data located in Input Storage is processed, and the results returned to Input Storage whenever an averaging period is completed (Parameter 7) or the Output Flag is set. The instruction controlling the Output Flag must precede the CV/CR Instruction. The reason the calculated results are returned to Input Storage is to allow the user access for additional processing before storing the values in Final Storage. Sample Instruction 70 must be used to transfer final results from Input to Final Storage.

To accommodate cases where it is desirable to calculate the statistical quantities over time periods shorter than the Output Interval, an averaging period shorter than the Output Interval may be specified. The final values obtained at the Output Interval are the properly weighted average of the values calculated at the subinterval averaging periods. This feature allows the recording of statistical data over longer time periods by removing the effect of longer period frequencies in the input signals; i.e., it provides a high pass filter. For example, assume the variance of an input is desired. It is determined that the averaging period should not exceed five minutes due to variation in the mean over longer time intervals. One approach is to calculate and record the variance every five minutes. By specifying the subinterval averaging period as five minutes and the Output Interval as one hour, however, the average of the five minute variances are recorded every hour. The averaging period is entered as the number of input samples in Parameter 7 of the CV/CR Instruction. The number of samples for a given period is given by:

$$\text{Number of Samples} = \frac{\text{Averaging period in seconds}}{\text{Table execution interval in seconds}}$$

| PAR. NO. | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 2 | Number of input values located sequentially in input memory |
| 02: | 2 | Number of means desired |
| 03: | 2 | Number of variances desired |
| 04: | 2 | Number of standard deviations desired |
| 05: | 2 | Number of covariances desired |
| 06: | 2 | Number of correlations desired |
| 07: | FP | Number of input samples in averaging period |
| 08: | 4 | Input storage location of first value in sequential input string |
| 09: | 4 | First Input Storage location to store string of final results |

If the specified number of samples in the averaging period (Parameter 7) exceeds the actual number of samples occurring in the Output Interval, the Output Interval becomes the averaging period.

**TABLE 10-1.  Maximum Number of Outputs and Output Order for K Input Values.**
**(The output order flows from left to right and from top to bottom)**

| TYPE | INPUTS: MAX NO. OUTPUTS | X1 (1st) | X2 (2nd) | X3 (3rd) | X4 OUTPUTS (4th) | ..... | XK (Kth) |
|------|------------------------|----------|----------|----------|-------------------|-------|----------|
| Means | K | M(X1) | M(X2) | M(X3) | M(X4) | ..... | M(XK) |
| Variances | K | V(X1) | V(X2) | V(X3) | V(X4) | ..... | V(XK) |
| Std. Deviation | K | SD(X1) | SD(X2) | SD(X3) | SD(X4) | ..... | SD(XK) |
| Covariance | K/2(K-1) | | CV(S1,X2) | CV(X1,X3) CV(X2,X3) | CV(X1,X4) CV(X2,X4) CV(X3,X4) ... .. . | ..... ..... ..... | CV(X1,XK) CV(X2,XK) CV(X3,XK) . . . CV(XK-1,XK) |
| Correlation | K/2(K-1) | | CR(X1,X2) | CR(X1,X3) CR(X2,X3) | CR(X1,X4) CR(X2,X4) CR(X3,X4) ... .. . | ..... ..... ..... | CR(X1,XK) CR(X2,XK) CR(X3,XK) . . . CR(XK-1,XK) |

| SYMBOL | DEFINITION |
|--------|-----------|
| M(XK) | Mean of Kth value |
| V(XK) | Variance of Kth value |
| SD(XK) | Standard deviation of Kth value |
| CV(XK,X1) | Covariance of Kth and Ith value |
| CR(XK,X1) | Correlation of Kth and Ith value |

## MAXIMUM NUMBER OF POSSIBLE OUTPUTS

No limitation exists on the number of inputs that can be processed by the CV/CR Instruction, but the processing time and Intermediate Storage requirements increase rapidly.  The instruction requires that the input values reside sequentially in Input Storage.  Since the number of outputs is specified for each type of statistical calculation, the instruction starts with the first value, working sequentially through the input values.  For this reason, the order of the input values determines which values are processed.

Table 10-1 shows the maximum number of outputs which can be generated and the output order for K input values located sequentially in Input Storage.  The output order shown in Table 10-1 flows from left to right and top to bottom.

INSTRUCTION PROCESSING
The CV/CR Instruction contains three separate processing phases:

1.  Input Processing
2.  Averaging Period Processing
3.  Output Processing

The Input Processing phase is where new input values are received, the necessary squares or cross products formed, and the appropriate summations calculated as required by the desired final output. The rate at which the measurements can be made, the input values ordered, and the input processing phase completed without interruption determines the maximum rate of execution (see Execution Time).

The Averaging Period Processing occurs whenever the number of input samples entered in Parameter 7 is satisfied or whenever an Output Interval occurs (i.e., whenever the Output Flag is set). Results from these calculations are stored sequentially in Input Storage locations starting with the location specified in Parameter 9. The calculations performed are shown below, where N is the number of input samples in the averaging period:

1. Means:
$$M(X) = \Sigma X / N$$
2. Variances:
$$V(X) = \Sigma X^2 / N - (\Sigma X / N)^2$$
3. Standard Deviations:
$$SD(X) = V(X)^{1/2}$$
4. Covariances:
$$CV(X,Y) = \Sigma XY / N - \Sigma X \ \Sigma Y / N^2$$
5. Correlations:
$$CR(X) = CV(X,Y)/(SD(X)SD(Y))$$

---

**NOTES:** 1. The square root algorithm in the CR7 returns a result of 0 for negative arguments.

2. The divide algorithm returns the largest floating point number possible ($\pm 10^{18}$ displayed as $\pm 99999$) for a divide by 0.

3. When computing the variance of a constant signal, round off error produces a small negative result. The CR7 returns a 0 for the square root of a negative number; therefore, the standard deviation is set to 0. If the signal is also used in a correlation calculation, division by 0 returns an overrange value for the correlation result.

---

If a fast execution interval is specified, it is possible that the combined execution times of the input and averaging period processing may exceed the program table execution interval. The occurrence of an execution interval Overrun (see Section 2.1 of CR7 Operator's Manual) is noted by decimal points on either side of the G in LOG (*0 MODE). This results in the omission of one input sample. The calculations are not affected, however, since the number of input samples is incremented only when valid input processing occurs. Averaging period processing occurs only when the number of input samples specified in Parameter 7 is accumulated.

Regardless of whether all of the input samples for the averaging period (specified in Parameter 7) have occurred or not, averaging period processing occurs whenever the Output Flag is set. This accommodates situations where the Output Interval may not be an integer multiple of the averaging period. If for example a 30 minute Output Interval is set by Instruction 92 and an 8 minute averaging period is specified by Parameter 7, then three 8 minute and one 6 minute calculations will occur. The proper weighting of these values in producing the final output is described below.

The Output Processing occurs only at the Output Interval and involves averaging the statistical results obtained at the averaging periods. These final results are then stored sequentially in Input Storage beginning with the location specified in Parameter 9. The Sample Instruction 70 must be used to transfer the data to Final Storage. All but the last averaging period in the Output Interval will contain the same number of input scans as specified by Parameter 7. To insure that results from the averaging periods contribute to the final result proportional to their averaging periods, the output processing uses the following equation:

$$RF = (NR1 + NR2 + .... + N'RL)/NT$$

RF is the final result
R1 R2, etc., are the results from the averaging period processing
RL is the result from the last averaging period in the Output Interval
N is the number of input samples in the specified averaging period (Parameter 7)
N' is the number of input scans in the last averaging period

## SECTION 10. PROCESSING INSTRUCTIONS

NT is the total number of input samples
processed in the Output Interval

### INTERMEDIATE STORAGE REQUIREMENTS

The number of Intermediate locations will
depend upon the number of input values and
outputs desired:

1. Define K as the number of input values.
2. Define S as the maximum of either the
   variances, standard deviations, or C, where
   C = K if K < the number of correlations
   requested, or
   C = number of correlations + 1 if K > the
   number of correlations requested.
3. Define Q as the maximum of either the
   covariances or correlations desired.
4. Define P as the total number of outputs
   desired.

The amount of intermediate memory locations
(IML) required, is then given by:

$$IML = K + S + Q + P + 2$$

EXECUTION TIME
If K, S, and Q are defined as in the previous
section, the execution time of the CV/CR
Instruction in milliseconds can be approximated
by:

$$T(ms) = 1.1K + 0.5S + 0.9Q + 1.8$$

When evaluating how frequently input samples
can be processed by the CV/CR Instruction
(i.e., determining the minimum program table
execution interval), the time required to make
the measurements and order the input values
must be added to the CV/CR execution time.
Two alternatives exist for the measurement
portion of the programming. The fastest
method is to group as many sensors as
possible into the fewest measurement
instructions, ignoring the Input location order
required by the CV/CR Instruction. After the
measurements are made, use "move"
instructions (i.e., 31 and 54) to obtain the proper
input order. The slower alternative is to order
separate measurement instructions directly as
required by the CV/CR Instruction. While
avoiding "move" instructions, this approach
uses more measurement instructions. The
reason the first method is in general faster is

that less overhead time is required in going
from one measurement to another within a
single instruction (using the "repetitions"
feature) than in going from one measurement
instruction to another.

In many situations, the CR7 must perform
measurement and processing tasks in addition
to those associated with the CV/CR Instruction.
Uninterrupted operation of the CV/CR
Instruction is assured by entering it in Program
Table 1 (highest priority) and placing the
additional tasks in Program Table 2.

A covariance correlation example is given in
Section 8.

### *** 66 ARCTAN ***

FUNCTION
Calculate the angle in degrees whose tangent is
X/Y. The polarity of X and Y must be known to
determine the quadrant of the angle, as shown
here. If 0 is entered for Parameter 2, the
Arctangent of X is the result (limits of the
function are $-90° < ARCTAN < 90°$).

| Quadrant | Sign of X | Sign of Y |
|----------|-----------|-----------|
| I        | +         | +         |
| II       | +         | -         |
| III      |           | -         |
| IV       |           | +         |

| PAR. NO. | DATA TYPE | DESCRIPTION | |
|----------|-----------|-------------|---|
| 01: | 4 | Input location of X | [X] |
| 02: | 4 | Input location of Y | [Y] |
| 03: | 4 | Destination input location for ARCTAN(X/Y) | |

Input locations altered: 1

### *** 68 EXTENDED PARAMETERS 4 DIGIT ***

FUNCTION
This instruction is used to give other instructions
additional parameters. Each of the eight
parameters in Instruction 68 is defined by the
instruction it follows. Refer to the specific
instruction that uses extended parameters.

Input location altered: 0

# SECTION 11. OUTPUT PROCESSING INSTRUCTIONS

## *** 69 WIND VECTOR ***

### FUNCTION

Instruction 69 processes the primary variables of wind speed and direction from either polar (wind speed and direction) or orthogonal (fixed East and North propellers) sensors. It uses the raw data to generate the mean wind speed, the mean wind vector magnitude, and the mean wind vector direction over an output interval. Two different calculations of wind vector direction (and standard deviation of wind vector direction) are available, one of which is weighted for wind speed.

When used with polar sensors, the instruction does a modulo divide by 360 on wind direction, which allows the wind direction (in degrees) to be 0 to 360, 0 to 540, less than 0, or greater than 540. The ability to handle a negative reading is useful in an example where a difficult to reach wind vane is improperly oriented and outputs 0 degrees at a true reading of 340 degrees. The simplest solution is to enter an offset of -20 in the instruction measuring the wind vane, which results in the correct output following processing.

When a wind speed sample is 0, the instruction uses 0 to process scalar or resultant vector wind speed and standard deviation, but the sample is not used in the computation of wind direction. The user may not want a sample less than the sensor threshold used in the standard deviation. If this is the case instruction 89 can be used to check wind speed, and if less than the threshold, Instruction 30 can set the input location equal to 0.

Standard deviation can be processed one of two ways: 1) using every sample taken during the output period (enter 0 for parameter 2), or 2) by averaging standard deviations processed from shorter sub-intervals of the output period (enter the number of scans in the sub-interval for parameter 2). Averaging sub-interval standard deviations minimizes the effects of meander under light wind conditions, and it provides more complete information for periods of transition[1].

Standard deviation of horizontal wind fluctuations from sub-intervals is calculated as follows:

$$\sigma(\theta)=[((\sigma\theta_1)^2+(\sigma\theta_2)^2 ...+(\sigma\theta_M)^2)/M]^{1/2}$$

where $\sigma(\theta)$ is the standard deviation over the output interval, and $\sigma\theta_1$ ... $\sigma\theta_M$ are sub-interval standard deviations.

| PARAM. NUMBER | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 2 | Repetitions |
| 02: | 4 | Samples per sub-interval (number of scans) |
| 03: | 2 | Sensor/Output 2 digits: AB |
| | | A  Sensor type: |
| | | 0 = Speed and Direction |
| | | 1 = East and North |
| | | B  Output option: |
| | | 0  **S, θ1, σ(θ1)** |
| | | 1  **S, θ1** |
| | | 2  **S, U, θu, σ(θu)** |
| 04: | 4 | First wind speed input location number (East wind speed) |
| 05: | 4 | First wind direction input location number (North wind speed) |

Outputs Generated:  2-4 (depending on output option) per repetition

A sub-interval is specified as a number of scans. The number of scans for a sub-interval is given by:

Desired sub-interval (secs) / scan rate (secs)

In an example where the scan rate is one second and the Output Flag is set every 60

---

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

minutes, the standard deviation is calculated from all 3600 scans when the sub-interval is 0. With a sub-interval of 900 scans (15 minutes), the standard deviation is the average of the four sub-interval standard deviations. The last sub-interval is weighted if it does not contain the specified number of scans.

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

Option 0:

Mean horizontal wind speed, **S**.
Unit vector mean wind direction, $\theta 1$.
Standard deviation of wind direction, $\sigma(\theta 1)$.

Standard deviation is calculated using the Yamartino algorithm. This option complies with EPA guidelines for use with straight-line Gaussian dispersion models to model plume transport.

Option 1:

Mean horizontal wind speed, **S**.
Unit vector mean wind direction, $\theta 1$.

Option 2:

Mean horizontal wind speed, **S**.
Resultant mean wind speed, **U**.
Resultant mean wind direction, $\theta u$.
Standard deviation of wind direction, $\sigma(\theta u)$.

This standard deviation is calculated using Campbell Scientific's wind speed weighted algorithm.

Use of the resultant mean horizontal wind direction is not recommended for straight-line Gaussian dispersion models, but may be used to model transport direction in a variable-trajectory model.

Measured raw data:

$S_i$ = horizontal wind speed
$\theta_i$ = horizontal wind direction
$Ue_i$ = east-west component of wind
$Un_i$ = north-south component of wind
$N$ = number of samples

Calculations:

**Scalar mean horizontal wind speed, S:**

$$S=(\Sigma S_i)/N$$

where in the case of orthogonal sensors:

$$S_i=(Ue_i{}^2+Un_i{}^2)^{1/2}$$

**Unit vector mean wind direction, $\theta 1$:**

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

where

$$Ux=(\Sigma\sin\theta_i)/N$$

$$Uy=(\Sigma\cos\theta_i)/N$$

or, in the case of orthogonal sensors

$$Ux=(\Sigma(Ue_i/U_i))/N$$

$$Uy=(\Sigma(Un_i/U_i))/N$$

$$\text{where } U_i=(Ue_i{}^2+Un_i{}^2)^{1/2}$$

**Standard deviation of wind direction, $\sigma(\theta 1)$,** using Yamartino algorithm:

$$\sigma(\theta 1)=\text{arc }\sin(\varepsilon)[1+0.1547\ \varepsilon^3]$$

where,

$$\varepsilon=[1-((Ux)^2+(Uy)^2)]^{1/2}$$

and Ux and Uy are as defined above.

**Resultant mean horizontal wind speed, U:**

$$U=(Ue^2+Un^2)^{1/2}$$

where for polar sensors:

$$Ue=(\Sigma S_i\ \text{Sin }\theta_i)/N$$

$$Un=(\Sigma S_i\ \text{Cos }\theta_i)/N$$

or, in the case of orthogonal sensors:

$$Ue=(\Sigma Ue_i)/N$$

$$Un=(\Sigma Un_i)/N$$

**Resultant mean wind direction, $\theta u$:**

$$\theta u=\text{Arctan }(Ue/Un)$$

**Standard deviation of wind direction, $\sigma(\theta u)$,** using Campbell Scientific algorithm:

$$\sigma(\theta u)=81(1-U/S)^{1/2}$$

### *** 70 SAMPLE ***

FUNCTION
This instruction stores the value from each specified input location.

| PAR. NO. | DATA TYPE | DESCRIPTION |
|----------|-----------|-------------|
| 01: | 2 | Repetitions |
| 02: | 4 | Starting input location number |

Outputs generated: 1 per repetition

### *** 71 AVERAGE ***

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

| PAR. NO. | DATA TYPE | DESCRIPTION |
|----------|-----------|-------------|
| 01: | 2 | Repetitions |
| 02: | 4 | Starting input location number |

Outputs generated: 1 per repetition

### *** 72 TOTALIZE ***

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

| PAR. NO. | DATA TYPE | DESCRIPTION |
|----------|-----------|-------------|
| 01: | 2 | Repetitions |
| 02: | 4 | Starting input location number |

Outputs generated: 1 per repetition

### *** 73 MAXIMUM ***

FUNCTION
This instruction stores the MAXIMUM value taken (for each input location specified) over a given output interval. An internal FLAG is set whenever a new maximum value is seen. This FLAG may be tested by Instruction 79. Time of maximum value(s) is OPTIONAL output information, which is selected by entering the appropriate code for Parameter 2.

| PAR. NO. | DATA TYPE | DESCRIPTION |
|----------|-----------|-------------|
| 01: | 2 | Repetitions |
| 02: | 2 | Time of maximum (optional) |
| 03: | 4 | Starting input location number |

Outputs generated: 1 per repetition (plus 1 or 2 with time of max. option)

**CODE OPTIONS**

| | |
|---|---|
| 00 | Output the maximum (or minimum) value ONLY |
| 01 | Output the max. (or min.) value with SECONDS information |
| 10 | Output the max. (or min.) value with HOUR-MINUTE information |
| 11 | Output the max. (or min.) value with HR-MIN,SEC information |

### *** 74 MINIMUM ***

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

| PAR. NO. | DATA TYPE | DESCRIPTION |
|----------|-----------|-------------|
| 01: | 2 | Repetitions |
| 02: | 2 | Time of minimum (optional) |
| 03: | 4 | Starting input location number |

Outputs generated: 1 per repetition (plus 1 or 2 with time of min. option)

### *** 75 STANDARD AND WEIGHTED *** VALUE HISTOGRAM

FUNCTION
Processes input data as either a standard histogram (frequency distribution) or a weighted value histogram.

The standard histogram outputs the fraction of time that the value in a specified input location (defined as the bin select value) is within a particular subrange of the total specified range. The count in the bin associated with each subrange is incremented whenever the value falls within that subrange. 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 two input locations. One location contains the bin select value; the other contains the weighted value. Each time the instruction is executed, the weighted value is added to a bin. The sub-range that the bin select value is in determines the bin to which the weighted value is added. When the Output Flag is set, the value accumulated in each bin is divided by the TOTAL number of input scans to obtain the values that are output to Final Storage. These values are the contributions of the sub-ranges to the overall weighted value. To obtain the average of the weighted values that occurred while the bin select value was within a particular sub-range, the value output to Final Storage must be divided by the fraction of time that the bin select value was within that particular sub-range (i.e., a standard histogram of the bin select value must also be output).

For either histogram, the user must specify: 1) the number of repetitions, 2) the number of bins, 3) a form code specifying whether a closed or open form histogram is desired (see below), 4) the bin select value input location, 5) the weighted value input location (see below), 6) the lower range limit, and 7) the upper range limit.

The standard histogram (frequency distribution) is specified by entering "0" in the weighted value input location parameter. Otherwise, this parameter specifies the input location of the weighted value. With more than one repetition the bin select value location will be incremented each repetition and the weighted value location will remain the same (same weighted value sorted on the basis of different bin select values). The weighted value location will be incremented if it is entered as an indexed location (key "C" at some point while keying in Parameter 5; two dashes, --, will appear on the right of the display).

At the user's option, the histogram may be either closed or open. The open form includes all values below the lower range limit in the first bin and all values above the upper range limit in the last bin. The closed form excludes any values falling outside of the histogram range.

The difference between the closed and open form is shown in the following example for temperature values:

| | | |
|---|---|---|
| Lower range limit | 10 °C | |
| Upper range limit | 30 °C | |
| Number of bins | 10 | |
| | | |
| | Closed Form | Open Form |
| | | |
| Range of first bin | 10 to 11.99° | <12° |
| Range of last bin | 28 to 29.99° | >28° |

A common use of a closed form weighted value histogram is the wind speed rose. Wind speed values (the weighted value input) are accumulated into corresponding direction sectors (bin select input).

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

Outputs generated: number of bins * repetitions

### *** 77 RECORD REAL TIME ***

FUNCTION
This instruction stores the current time in Final Storage. At midnight the clock rolls over from 23:59 to 00:00. The day also changes.

If hourly or daily summary data is output, it may be desirable to have the previous day specified with the output, since that is when the measurements were made. Entering a 2 for the day code causes the previous day to be output if it is the first minute of the day. Similarly, entering 2 for the hour-minute code causes 2400 instead of 0000 to be output (the next minute is still 0001). When day and hour-minute are both output, a 2 for either code results in the previous day at 2400.

The year is output as 19xx if xx is greater than 85, otherwise it will be output as 20xx. The CR7 will require a PROM update in the year 2085. If year is output along with a 2 option in day or hour-minute, the previous year will be output during the first minute of the new year.

## CODE RESULTS

xxx1   SECONDS (with a resolution of 0.1 sec.)
xx1x   HOUR-MINUTE
xx2x   HOUR-MINUTE, 2400 instead of 0000
x1xx   DAY OF YEAR
x2xx   DAY OF YEAR, previous day during first minute of new day
1xxx   YEAR

Any combination of Year, Day, HR-MIN, and seconds is possible (e.g., 1011: YEAR, HR-MIN, SEC).

| PAR.<br>NO. | DATA<br>TYPE | DESCRIPTION |
|---|---|---|
| 01: | 4 | Enter appropriate TIME option code |

Outputs generated: 1 for each time parameter selected

## *** 78 SET HIGH OR LOW RESOLUTION ***
### FINAL STORAGE FORMAT

FUNCTION
This instruction sets the Final Storage Format to high resolution (5 character) or low resolution (4 character, Section 2.2). Instruction 78 should be entered ahead of the output instructions for which the specified resolution is desired. The default format is low resolution. At the beginning of each program table execution, the low resolution format is automatically enabled.

| PAR.<br>NO. | DATA<br>TYPE | DESCRIPTION |
|---|---|---|
| 01: | 2 | 0 = low resolution;<br>1 = high resolution |

Outputs generated: 0

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

FUNCTION
Instruction 79 samples specified input location values at the time a new maximum or minimum value is detected by a previous Maximize (73) or Minimize (74) Instruction. When the Output Flag is set, the values are transferred to Final Storage.

Instruction 79 must directly follow the maximum or minimum Instruction to which it refers. If the previous Instruction 73 or 74 has more than 1 repetition, Instruction 79 will sample whenever a new maximum or minimum is detected in any of the locations. If sampling is to occur only when a specific input location shows a new maximum or minimum, the previous Maximize or Minimize Instruction should have 1 repetition referring to that input location.

| PAR.<br>NO. | DATA<br>TYPE | DESCRIPTION |
|---|---|---|
| 01: | 2 | Repetitions (number of sequential locations to sample) |
| 02: | 4 | Starting input location number |

Outputs generated: 1 per repetition

## *** 80 SET ACTIVE OUTPUT AREA ***

Instruction 80 is used to direct Output data to Final Storage or to Input Storage. At the beginning of each table the Active Output area is set to Final Storage. When directed to Final Storage, the second parameter can be used to set the output array ID. If 80 is used to direct output to Final Storage, and Parameter 2 is 0, the array ID is determined by the instruction location number of Instruction 80 or by the Instruction that set the Output Flag, whichever came last. When data is sent to Input Storage, no array ID is sent.

## SECTION 11. OUTPUT PROCESSING INSTRUCTIONS

| PAR. NO. | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 2 | Storage area option<br>01 = Final Storage (00 and<br>02 also default to<br>Final Storage)<br>03 = Input Storage Area |
| 02: | 4 | Starting input location<br>destination if option 03<br>Output Array ID if options 0-2<br>(1-511 are valid IDs) |

### *** 82  STANDARD DEVIATION IN TIME ***

FUNCTION
Calculate the standard deviation (STD DEV) of
a given input location.  The standard deviation
is calculated using the formula:

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

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

| PAR. NO. | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 2 | Repetitions |
| 02: | 4 | Starting input location number |

Outputs generated:  1 per repetition

# SECTION 12. PROGRAM CONTROL INSTRUCTIONS

## TABLE 12-1. Flag Description

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

## TABLE 12-2. Port Command Codes

| | | |
|---|---|---|
| 0 | - | Go to end of program table |
| 1-9, 79-99 | - | Call Subroutine 1-9, 79-99 |
| 10-19 | - | Set Flag 0-9 high |
| 20-29 | - | Set Flag 0-9 low |
| 30 | - | Then Do |
| 31 | - | Exit loop if true |
| 32 | - | Exit loop if false |
| 41-48* | - | Set port 1 - 8 high |
| 51-58* | - | Set port 1 - 8 low |
| 61-68* | - | Toggle port 1 - 8 |
| 71-78* | - | Pulse port 1 - 8 100 ms |

*The port commands operate on excitation card 1 or on the excitation card set active by Instruction 20.*

## *** 83  IF CASE  X < F ***

### FUNCTION
Instruction 83 tests the value in an input location specified in the Begin Case Instruction 93. A series of Instructions 83 are used to compare the value in the input location to ever increasing fixed values. If the value in the location is less than the fixed value entered as Parameter 1, then the command in Parameter 2 is executed and when the next Instruction 83 is encountered execution branches to the end of the case statement. If the fixed value is less, the next Instruction 83 is executed. See Instruction 93 for an example.

| PAR. NO. | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | FP | Fixed value |
| 02: | 2 | Command (Table 12-2) |

## *** 85 LABEL SUBROUTINE ***

### FUNCTION
This instruction marks the start of a subroutine. A subroutine is a series of instructions beginning with Instruction 85 and terminated with Instruction 95, END. All subroutines must be placed in Table 3 (Subroutine Table). When a subroutine is called by a command in a Program Control Instruction, the subroutine is executed, then program flow continues with the instruction following that which called the subroutine.

Subroutines cannot be embedded within other subroutines; a subroutine must end before the next one begins. Subroutines may be called from within other subroutines (nested). The maximum nesting level for subroutines is 7 deep. Attempts to nest more than 7 deep will not be detected at compilation, but will result in a run time error. When the seventh subroutine attempts to call the eighth, error 31 will be displayed. Execution will not branch to the eighth subroutine; it will continue with the Instruction following that calling the subroutine.

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

## *** 86 DO ***

### FUNCTION
This instruction unconditionally executes the specified command.

| PAR. NO. | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 2 | Command (Table 12-2) |

## *** 87 LOOP ***

### FUNCTION
Instructions included between the Loop Instruction and the End Instruction (95) are repeated the number of times specified by the iteration count (Parameter 2), or until an Exit Loop command (31, 32) is executed by a Program Control Instruction within the Loop. If

0 is entered for the count, the loop is repeated until an Exit Loop command is executed.

The first parameter, delay, controls how frequently passes through the loop are made. The delay unit is the table execution interval:

> A delay of 0 means that there is no delay between passes through the loop. Each time the table is executed all iterations of the loop will be completed and execution will pass on to the following instructions.

> If the delay is 5, every fifth time that the execution interval comes up, one pass through the loop is made; only those instructions in the loop will be executed and other portions of the table are not executed in the interim. When the loop is executed, execution starts at the loop, skipping over any previous instructions in the table.

When a fixed number of iterations are executed, the time spent in the loop is equal to the product of the execution interval, delay, and the number of iterations. For example, a loop with a delay of 1 and a count of 5 will take 5 seconds if the execution interval is 1 second. When the loop is first entered, one pass through the loop is made, then the CR7 delays until the next execution interval and makes the second pass through the loop. After making the fifth pass through the loop, there is the fifth delay, after which execution passes to the instruction following the END instruction which goes with the loop.

While in a loop with delay, the table will not be initiated at each execution interval. (However, the overrun decimals will not be displayed.) Some consequences of this are: The Output Flag will not be automatically cleared between passes through the loop. Because Table 2 cannot interrupt Table 1, Table 2 will not be executed while Table 1 is in a loop with delay. Table 1 will not interrupt Table 2 in the middle of an output array. Thus, if the Output Flag is set in Table 2 prior to entering the loop or within the loop, the flag must be specifically cleared before the end of the pass or Table 1 will not be able to interrupt.

Input locations for Processing Instructions within a loop can be entered as Indexed locations. An Indexed location causes the input location to be incremented by 1 with each pass through the

loop. (The Index counter is added to the location number in the program table.) Input locations which are not indexed will remain constant.

To specify an Indexed location, depress the C key at some point while keying in the digits for the input location and before entering the location with the A key. Two dashes, --, appear in the two right most characters of the display, indicating the entry is Indexed.

When the same output processing is required on values in sequential input locations, it must be accomplished by using the repetitions parameter of the Output Instruction, not by indexing the input location within a loop.

An Output Instruction within a loop is allotted the same number of Intermediate Storage locations as it would receive if it were not in the loop. For example, the Average instruction with a single repetition is allotted only two Intermediate locations: one for the number of samples and one for the running total. Each time through the loop the sample counter is incremented and the value in the referenced input location is added to the total. If the input location is indexed, the values from all input locations are added to the same total. If the Average instruction with 1 repetition and location 1 indexed is placed within a loop of 10 and the Output Flag set high prior to entering the loop, 10 values will be output. These will not be the averages for locations 1-10. The first will be the average of all the readings in locations 1-10 since the previous output. Because the Intermediate locations are zeroed each time an output occurs, the next nine values will be the current values (samples at the time of output) of Locations 2-10.

Loops can be nested. Indexed locations within nested loops are indexed to the inner most loop that they are within. The maximum nesting level in the CR7 is 9 deep. This applies to If Then/Else comparisons and Loops or any combination thereof. An If Then/Else comparison which uses the Else Instruction, 94 counts as being nested 2 deep.

| PAR. NO. | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 2 | Delay |
| 02: | 4 | Iteration count |

The following example involves the use of the Loop Instruction, without a delay, to perform a block data transformation.

The user wants one hour averages of the vapor pressure calculated from the wet- and dry-bulb temperatures of five psychrometers. One pressure transducer measurement is also available for use in the vapor pressure calculation.

1. The input locations are assigned as follows:
   a) pressure               -location 10
   b) dry-bulb temperatures    -location 11-15
   c) wet-bulb temperatures    -location 16-20
   d) calculated vapor pressure  -location 16-20
   (vapor pressure is written over the wet-bulb temperatures.)

2. The program flow is as follows:
   a) Enter the Loop Instruction 87 with delay=0 and iteration count=5.
   b) Calculate the vapor pressure with Instruction 57 using a normal location entry of 10 for atmospheric pressure and Indexed locations of 11, 16, and 16 for the dry-bulb, wet-bulb and calculated vapor pressure, respectively.
   c) End loop with Instruction 95.
   d) Use the If Time Instruction 92 to set the Output Flag every hour.
   e) Use the Average Instruction 71 with 5 repetitions starting at Input Location 16 to average the vapor pressure over the hour.

The actual keyboard entries for the examples are shown below with the first example Instruction location equal to 10. The Input Instructions to make the pressure and temperature measurements are assumed.

---

**TABLE 12-3. Loop Example: Block Data Transform**

| 10: | P87 | Beginning of Loop |
|---|---|---|
| 01: | 0 | Delay |
| 02: | 5 | Loop Count |
| | | |
| 11: | P57 | Wet/Dry Bulb Temp to VP |
| 01: | 10 | Pressure Loc |
| 02: | 11-- | Dry Bulb Temp Loc DRY BLB#1 |
| 03: | 16-- | Wet Bulb Temp Loc VP #1 |
| 04: | 16-- | Loc [:VP #1   ] |
| | | |
| 12: | P95 | End |
| | | |
| 13: | P92 | If time is |
| 01: | 0 | minutes into a |
| 02: | 60 | minute interval |
| 03: | 10 | Set high Flag 0 (output) |
| | | |
| 14: | P71 | Average |
| 01: | 5 | Reps |
| 02: | 16 | Loc VP #1 |

The Loop with a delay may be used so that only those instructions within the Loop are executed while certain conditions are met. As a simple example, suppose it is desired to execute one set of instructions from midnight until 6 AM, another set between 6 AM and 4 PM, and a third set between 4 PM and midnight. Between 6 AM and 4 PM, samples are desired every ten seconds; the rest of the time one minute between samples is sufficient. The execution interval is set to ten seconds; when a one minute sample rate is desired, a delay of 6 (6 x 10s = 60s) is used in the loop.

---

**TABLE 12-4. Example: Loop with Delay Execution Interval = 10 seconds**

| * | 1 | Table 1 Programs |
|---|---|---|
| 01: | 10 | Sec. Execution Interval |
| | | |
| 01: | P87 | Beginning of Loop |
| 01: | 6 | Delay |
| 02: | 0 | Loop Count |
| | | |
| 11: | P86 | Do |
| 01: | 1 | Call Subroutine 1 |

| 12: | P89 | If X<=>F |
|---|---|---|
| 01: | 25 | X Loc DAY |
| 02: | 3 | >= |
| 03: | 6 | F |
| 04: | 31 | Exit Loop if true |

| 13: | P95 | End |
|---|---|---|

| 14: | P87 | Beginning of Loop |
|---|---|---|
| 01: | 1 | Delay |
| 02: | 0 | Loop Count |

| 27: | P86 | Do |
|---|---|---|
| 01: | 1 | Call Subroutine 1 |

| 28: | P89 | If X<=>F |
|---|---|---|
| 01: | 25 | X Loc DAY |
| 02: | 3 | >= |
| 03: | 16 | F |
| 04: | 31 | Exit Loop if true |

| 29: | P95 | End |
|---|---|---|

| 30: | P87 | Beginning of Loop |
|---|---|---|
| 01: | 6 | Delay |
| 02: | 0 | Loop Count |

| 36: | P86 | Do |
|---|---|---|
| 01: | 1 | Call Subroutine 1 |

| 37: | P89 | If X<=>F |
|---|---|---|
| 01: | 25 | X Loc DAY |
| 02: | 3 | >= |
| 03: | 5 | F |
| 04: | 32 | Exit Loop if false |

| 38: | P95 | End |
|---|---|---|

| 39: | P | End Table 1 |
|---|---|---|

| * | 3 | Table 3 Subroutines |
|---|---|---|

| 01: | P85 | Beginning of Subroutine |
|---|---|---|
| 01: | 1 | Subroutine Number |

| 02: | P18 | Time |
|---|---|---|
| 01: | 2 | Hours into current year (maximum 8784) |
| 02: | 24 | Mod/by |
| 03: | 25 | Loc [:DAY    ] |

| 03: | P95 | End |
|---|---|---|

*** **88  IF X COMPARED TO Y** ***

FUNCTION
This instruction compares two input locations and, if the result is true, executes the specified Command. The comparison codes are given in Table 12-5.

| PAR. NO. | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 4 | Input location for X |
| 02: | 2 | Comparison code (Table 12-5) |
| 03: | 4 | Input location for Y |
| 04: | 2 | Command (Table 12-2) |

Input locations altered:  0
Execution time:            0.6ms

---

**TABLE 12-5. Comparison Codes**

| Parameter 1 | Function |
|---|---|
| 1 | IF X = Y |
| 2 | IF X ≠ Y |
| 3 | IF X ≥ Y |
| 4 | IF X < Y |

*** **89  IF X COMPARED TO F** ***

FUNCTION
This instruction compares an input location to a fixed value and, if the result is true, performs the specified Command. The comparison codes are given in Table 12-5.

| PAR. NO. | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 4 | Input location for X |
| 02: | 2 | Comparison code (Table 12-5) |
| 03: | FP | Fixed value |
| 04: | 2 | Command (Table 12-2) |

*** **90  STEP LOOP INDEX** ***

FUNCTION
When used within a Loop (Instruction 87), Instruction 90 will increment the index counter by a specified amount after the first time through the loop, thus affecting all indexed input location parameters in subsequent instructions within the loop. For example, if 4 is specified, the index counter will count up by 4 (0, 4, 8, 12,...) inside the loop. Instruction 90 does not affect the loop counter which still counts by 1.

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

### *** 91 IF FLAG ***

FUNCTION
This instruction checks one of the ten flags and conditionally performs the specified command.

The first parameter specifies the flag to check and the flag status (high or low) on which to execute the command.

    1X = execute Command if Flag X is high
    2X = execute if Flag X is low

| PAR. NO. | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 2 | Flag/status |
| 02: | 2 | Command (Table 12-2) |

### *** 92 IF TIME ***

FUNCTION
The user specifies the number of minutes 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 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 automatically synchronized by making a modulo divide of the number of minutes since midnight by the specified real time interval. If the result is 0, the interval is up. Thus, the first interval of the day always starts at midnight (0 minutes). Only one execution is allowed in any minute (e.g., if the command is to set the Output Flag, and the execution interval of the table is ten seconds, there will only be one output generated by this instruction, not six.)

The Output Flag (Flag 0) is a special case in that it will automatically be set low if it is not time to set it high.

| PAR. NO. | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 4 | Time into interval (minutes) |
| 02: | 4 | Time interval (minutes) |
| 03: | 2 | Command (Table 12-2) |

### *** 93 BEGIN CASE STATEMENT ***

Instruction 93 specifies an input location for comparison with fixed values in subsequent If Case instructions (83). When a comparison is true, the command in the If Case instruction is executed and at the next Instruction 83 execution jumps to the End Instruction 95 associated with the Begin Case Instruction.

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

    EXAMPLE:

| 01: | P93 | Case |
|---|---|---|
| 01: | 2 | Case Loc |
| | | |
| 02: | P83 | If Case Location < F |
| 01: | 69.4 | F |
| 02: | 3 | Call Subroutine 3 |
| | *else* | |
| 03: | P83 | If Case Location < F |
| 01: | 72 | F |
| 02: | 10 | Set high Flag 0 (output) |
| | *else* | |
| 04: | P83 | If Case Location < F |
| 01: | 77.3 | F |
| 02: | 30 | Then Do |
| | | |
| 05: | P30 | Z=F |
| 01: | 0 | F |
| 02: | 0 | Exponent of 10 |
| 03: | 25 | Z Loc : |
| | | |
| 06: | P95 | End  *Then Do* |
| | | |
| 07: | 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 (Figure 3.8-1). The

Else Instruction is optional; when it is omitted, a false comparison will result in execution branching directly to the End Instruction. Instruction 94 has no parameters.

### *** 95 END ***

FUNCTION
Instruction 95 is used to indicate the end/return of a subroutine (Instruction 85), the end of a loop (Instruction 87), the end of an If Then/Else sequence, or the end of the Case statement (Instructions 88-93 when used with command 30). The End Instruction has no parameters.

### *** 96 ACTIVATE SERIAL DATA OUTPUT ***

FUNCTION
Instruction 96 is used instead to activate the Storage Module or serial port output. By using Program Control Instructions to allow execution of Instruction 96 only at certain times, the user can control when the output is active.

When used to send data to the SM192 or SM716 Storage Module, the CR7 can determine whether or not the Storage Module is connected. If the Storage Module is not connected, the data will not be sent until it is connected. Instruction 96 also allows for faster data output via the serial port with the Binary Option, which outputs FINAL STORAGE FORMAT (2 bytes per low resolution data point) instead of ASCII (10 bytes per data point, Section 4.5). Appendix C describes FINAL STORAGE FORMAT.

A single parameter is used to select whether the instruction is to control the tape, Storage Module, or the printer output, and if the printer is selected, the format and baud rate. Instruction 96 must be entered separately for each output device used.

Instruction 96 uses the same Printer pointer as the *9 mode which can be used to get a residual or partial dump of Final Storage (Section 4).

```
Do not use *4 to activate a device that is
activated by Instruction 96.
```

| PAR. NO. | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 2 | The left digit specifies the Option and the right specifies the baud rate for the printer. The code for Storage Module is 30. |

| Option | X = Baud Rate |
|---|---|
| | 0 --    300 |
| 1X -- PRINTER, ASCII | 1 --   1200 |
| 2X -- PRINTER, Binary | 2 --   9600 |
| 30 -- SM192 or SM716 | 3 -- 76,800 |
| 31 -- Send filemark to SM192/716 | |

### *** 98 SEND CHARACTER ***

Instruction 98 is used to send a character to the printer. The single parameter sets the baud rate and gives the decimal equivalent of the 7 bit character (sent as 8 bits, no parity). For example, to send the ASCII character control R at 9600 baud, 2018 would be entered for Parameter 1. This instruction can be used to send a control character to activate some device. The specified character is sent at the time Instruction 98 is executed; this will cause it to precede any output arrays generated in the same table, since the output data is sent to the printer at the completion of the table.

| PAR. NO. | DATA TYPE | DESCRIPTION |
|---|---|---|
| 01: | 4 | Decimal Value of ASCII character: yxxx  xxx=ASCII value (1-127) y=Baud rate code |

| | | |
|---|---|---|
| 0 | 300 | baud |
| 1 | 1200 | baud |
| 2 | 9600 | baud |
| 3 | 76,800 | baud |

# SECTION 13.  CR7 MEASUREMENTS

## 13.1 FAST AND SLOW MEASUREMENT SEQUENCE

The CR7 makes voltage measurements by integrating the input signal for a fixed time and then holding the integrated value for the analog to digital (A/D) conversion.  The A/D conversion is made with a 16 bit successive approximation technique which resolves the signal voltage to approximately one part in 30,000 of either the + or - side of the full scale range (e.g., 1/30,000 x 5V = 166μV).

Integrating the signal removes noise that could create an error if the signal were instantaneously sampled and held for the A/D conversion.  The slow integration time provides a more noise-free reading than the fast integration time.  One of the most common sources of noise is 60 Hz from AC power lines. The slow integration time of 16.67 milliseconds is equal to one 60 Hz cycle so that during the integration time the AC noise would integrate to 0.

There are several situations where the fast integration time of 250 microseconds is preferred. The fast integration time minimizes time skew between measurements and increases the throughput rate. The current drain on the CR7 batteries is lower when fast integration time is used because the I/O CPU is switched on for shorter time periods.  The fast integration time should ALWAYS be used with the AC half bridge (Instruction 5) when measuring AC resistance or the output of an LVDT.  An AC resistive sensor will polarize if a DC voltage is applied, causing erroneous readings and sensor decay.  The induced voltage in an LVDT decays with time as current in the primary coil shifts from the inductor to the series resistance, a long integration time would result in most of the integration taking place after the signal had disappeared.
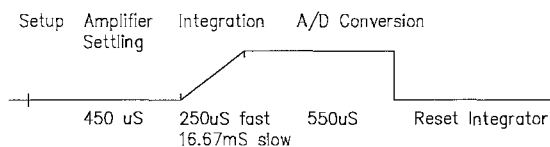


**FIGURE 13.1-1.  Timing of Single Ended Measurement**

Before making a series of measurements prescribed by an Input Instruction, the CR7 makes a calibration measurement.  The calibration is accomplished by measuring two known voltages which are sent through the same amplifier circuit that will be used for the measurements.  The calibration for a single ended measurement consists of measuring a voltage which is 4/5ths of full scale and then making a measurement with the input grounded.  A differential measurement is made once with the inputs as connected and a second time with the inputs reversed (Section 13.2): calibration for differential measurements uses voltages at ±4/5ths of full scale.

An offset error of up to 1 least significant bit can occur in a slow, single ended measurement as a result of dielectric absorption in the integrating capacitor.  This error is a function of the previous measurement.  If the CR7 is programmed to make a single ended measurement on the 5 volt range with the inputs shorted, an error of -166 μV can be observed.

## 13.2 SINGLE ENDED AND DIFFERENTIAL VOLTAGE MEASUREMENTS

> **NOTE:** The channel numbering on the Analog Input cards refers to differential channels. Either the high or low side of a differential channel can be used for single ended measurements so each side must be counted when numbering single ended channels, e.g., the high and low sides of differential channel 14 are single ended channels 27 and 28, respectively.

The timing and sequence of a single ended measurement is shown in Figure 13.1-1.  A single ended measurement is made on a single input which is referenced to ground.  A single integration is performed for each measurement. A differential measurement measures the difference in voltage between two inputs.  The measurement sequence on a differential measurement involves two integrations:  first with the high input referenced to the low, then with the inputs reversed.  (Figure 13.2-1).  The CR7 computes the differential voltage by averaging the magnitude of the results from the two integrations and using the polarity from the first.

Setup & Integ.    A/D Conv. Switch Inpt.    Integ.    A/D Conv.
Settle                       & Settle

250us fast    550us    600us    250us fast    550us
16.67ms slow                    16.67ms slow
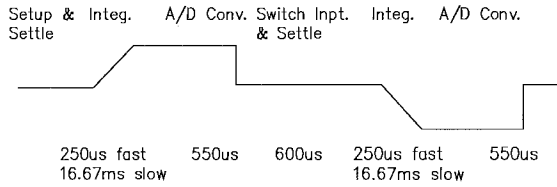
**FIGURE 13.2-1.  Differential Voltage
Measurement Sequence**

Because a single ended measurement is referenced to CR7 ground, any difference in ground potential between the sensor and the CR7 will result in an error in the measurement. For example, if the measuring junction of a copper-constantan thermocouple, being used to measure soil temperature, is not insulated and the potential of earth ground is 1 mV greater at the sensor than at the point where the CR7 is grounded, the measured voltage would be 1 mV greater than the thermocouple output, or approximately 25 °C high.  Another instance where a ground potential difference creates a problem is in a case such as described in Section 7.2, where external signal conditioning circuitry is powered from the same source as the CR7.  Despite being tied to the same ground, differences in current drain and lead resistance result in different ground potential at the two instruments.  For this reason, a differential measurement should be made on an analog output from the external signal conditioner.  Differential measurements MUST be the choice where the inputs are known to be different from ground, such as the output from a full bridge.

In order to make a differential measurement, however, the inputs must be within the CR7 common mode range of ±5V.  The common mode range is the voltage range, relative to CR7 ground, within which both inputs of a differential measurement must lie, in order for the differential measurement to be made.  For example, if the high side of a differential input is at 4V and the low side is at 3V relative to CR7 ground, there is no problem, a measurement made on the ±1.5V range would indicate a signal of 1V.  However, if the high input is at 5.8V and the low input is at 4.8V, the measurement can not be made because the high input is outside of the CR7 common mode range (the CR7 will indicate the overrange with the maximum negative number, Section 2.2).

Problems with exceeding common mode range may be encountered when the CR7 is used to read the output of external signal conditioning circuitry if a good ground connection does not exist between the external circuitry and the CR7.  When operating where AC power is available, it is not always safe to assume that a good ground connection exists through the AC wiring.  If a CR7 is used to measure the output from a laboratory instrument (both plugged into AC power and referencing ground to outlet ground), it is best to run a ground wire between the CR7 and the external circuitry.  Even with this ground connection, the ground potential of the two instruments may not be at exactly the same level, which is why a differential measurement is desired.

A differential measurement has better noise rejection than a single ended measurement. Integrating the signal in both directions also reduces input offset voltage due to thermal effects in the amplifier section of the CR7.  Input offset voltage on a differential measurement is on the order of 0.1 microvolts, the input offset voltage on a single ended measurement may be as high as 1 to 2 microvolts.

A single ended measurement is quite satisfactory in cases where noise is not a problem and care is taken to avoid ground potential problems.  Twice as many single ended measurements can be made per Analog Input Card.  A single ended measurement takes about half the time of a differential measurement which is valuable in cases where rapid sampling is a requirement.

**NOTE:** Sustained voltages in excess of ±16 VDC applied to the analog inputs will damage the CR7 input circuitry.

## 13.3 THE EFFECT OF SENSOR LEAD LENGTH ON THE SIGNAL SETTLING TIME

Whenever an analog input is switched into the CR7 measurement circuitry prior to making a measurement, a finite amount of time is required for the signal to stabilize at it's correct value. The rate at which the signal settles is determined by the input settling time constant which is a function of both the source resistance and input capacitance (explained below). The CR7 allows a 0.5ms settling time before initiating the measurement. In most applications, this settling time is adequate but the additional wire capacitance associated with long sensor leads can increase the settling time constant to the point that measurement errors may occur. There are three potential sources of error which must settle before the measurement is made:

1. The signal must rise to its correct value.

2. A small transient (5mV) caused by switching the analog input into the measurement circuitry must settle.

3. A larger transient, usually about 40 mV/V, caused by the switched, precision excitation voltage used in resistive bridge measurements must settle.

The purpose of this section is to bring attention to potential measurement errors caused when the input settling time constant gets too large and discuss procedures whereby the effects of lead length on the measurement can be estimated. In addition, physical values are given for three types of wire used in Campbell Scientific sensors and error estimates for given lead lengths are provided. Finally, techniques are discussed for minimizing input settling error when long leads are mandatory.

### 13.3.1 THE INPUT SETTLING TIME CONSTANT

The rate at which an input voltage rises to its full value or that a transient decays to the correct input level are both determined by the input settling time constant. In both cases the waveform is an exponential. Figure 13.3-1 shows both a rising and decaying waveform settling to the signal level, $V_{so}$. The rising input voltage is described by Equation 13.3-1 and the decaying input voltage by Equation 13.3-2,
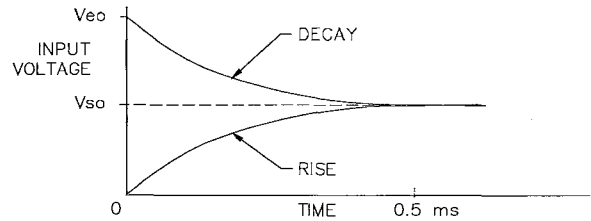


**FIGURE 13.3-1. Input Voltage Rise and Transient Decay**

$$V_s = V_{so}(1 - e^{-t/R_o C_T}), \text{ rise} \qquad [13.3-1]$$

$$V_s = V_{so} + (V_{eo} - V_{so})e^{-t/R_o C_T}, \text{ decay} \quad [13.3-2]$$

where $V_s$ is the input voltage, $V_{so}$ the true signal voltage, $V_{eo}$ the peak transient voltage, t is time in seconds, $R_o$ the source resistance in ohms and $C_T$ is the total capacitance between the signal lead and ground (or some other fixed reference value) in farads.

The settling time constant, $\tau$ in seconds, and the capacitance relationships are given in Equations 13.3-3 through 13.3-5,

$$\tau = R_o C_T \qquad [13.3-3]$$

$$C_T = C_f + C_w L \qquad [13.3-4]$$

$$C_f = 0.01 \text{ nfd} \qquad [13.3-5]$$

where $C_f$ is the fixed CR7 input capacitance in farads, $C_w$ is the wire capacitance in farads/foot and L is the wire length in feet.

Equations 13.3-1 and 13.3-2 can be used to estimate the input settling error, $V_e$, directly. For the rising case, $V_s = V_{so} - V_e$ whereas for the decaying transient $V_s = V_{so} + V_e$. Substituting these relationships for $V_s$ in Equations 13.3-1 and 13.3-2, respectively, yields expressions in $V_e$, the input settling error:

$$V_e = V_{so} e^{-t/R_o C_T}, \text{ rise} \qquad [13.3-6]$$

$$V_e = V'_{eo} e^{-t/R_o C_T}, \text{ decay} \qquad [13.3-7]$$

Where $V'_{eo} = V_{eo} - V_{so}$, the difference between the peak transient voltage and the true signal voltage.

## SECTION 13. CR7 MEASUREMENTS

Since the peak transient, $V_{eo}$, causes significant error only if it is several times larger than the signal, $V_{so}$, error calculations made in this section approximate $V_{e'o}$ by $V_{eo}$, i.e., $V_{eo} \approx V_{eo}-V_{so}$.

If the input settling time constant, $\tau$, is known, a quick estimation of the settling error as a percentage of the maximum error ($V_{so}$ for rising, $V_{eo}$ for decaying) is obtained by knowing how many time constants ($t/\tau$) are contained in the 0.5 ms CR7 input settling interval (t). The familiar exponential decay relationship is given in Table 13.3-1 for reference.

### TABLE 13.3-1. Exponential Decay, Percent of Maximum Error vs. Time in Units of $\tau$

| Time Constants | % Max. Error | Time Constants | % Max. Error |
|---|---|---|---|
| 0 | 100.0 | 5 | 0.7 |
| 1 | 36.8 | 7 | 0.1 |
| 3 | 5.0 | 10 | 0.004 |

Before proceeding with examples of the effect of long lead lengths on the measurement, a discussion on obtaining the source resistance, $R_o$, and lead capacitance, $C_wL$, is necessary.

## DETERMINING SOURCE RESISTANCE

The source resistance used to estimate the settling time constant is the resistance the CR7 input "sees" looking out at the sensor. For our purposes the source resistance can be defined as the resistance from the CR7 input through all external paths back to the CR7. Figure 13.3-2 shows a typical resistive sensor, (e.g., a thermistor) configured as a half-bridge. Figure 13.3-3 shows Figure 13.3-2 redrawn in terms of the resistive paths determining the source resistance $R_o$, is given by the parallel resistance of $R_s$ and $R_f$, as shown in Equation 13.3-8.
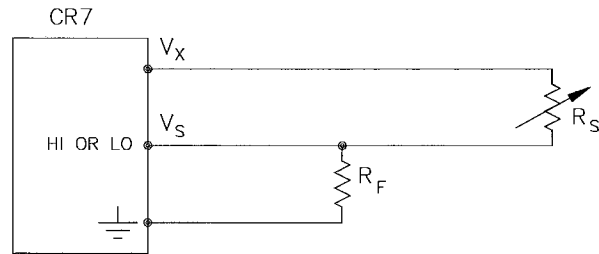


### FIGURE 13.3-2. Typical Resistive Half-Bridge
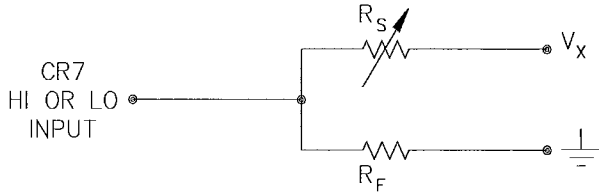


### FIGURE 13.3-3. Source Resistance Model for Half-Bridge Connected to the CR7

$$R_o = R_sR_f/(R_s+R_f) \qquad [13.3-8]$$

If $R_f$ is much smaller, equal to or much greater than $R_s$, the source resistance can be approximated by Equations 13.3-9 through 13.3-11, respectively.

$$R_o \approx R_f, \quad R_f \ll R_s \qquad [13.3-9]$$

$$R_o = R_f/2, \quad R_f = R_s \qquad [13.3-10]$$

$$R_o \approx R_s, \quad R_f \gg R_s \qquad [13.3-11]$$

The source resistance for several Campbell Scientific sensors are given in column 3 of Table 13.3-5.

## DETERMINING LEAD CAPACITANCE

Wire manufacturers typically provide two capacitance specifications 1) the capacitance between the two leads with the shield floating and 2) the capacitance between the two leads with the shield tied to one lead. Since the input lead and the shield are tied to ground (often through a bridge resistor, $R_f$) in single ended measurements such as Figure 13.3-2, the second specification is used in determining lead capacitance. Figure 13.3-4 is a representation of this capacitance, $C_w$, usually specified as pfd/ft. $C_w$ is actually the sum of capacitance between the two conductors and the capacitance between the top conductor and the shield. Capacitance for 3 Belden leadwires used in Campbell Scientific sensors is shown in column 6 of Table 13.3-2.

**TABLE 13.3-2. Properties of Three Belden Lead Wires Used by Campbell Scientific**

| Belden Wire # | Conductors | Insulation | AWG | $R_l$ (ohms/1000ft.) | $C_w$ (pfd/ft.) |
|---|---|---|---|---|---|
| 8641 | 1 shld. pair | polyethylene | 24 | 23 | 42 |
| 8771 | 1 shld. 3 cond. | polyethylene | 22 | 15 | 41 |
| 8723 | 2 shld. pair | polypropylene | 22 | 15 | 62 |
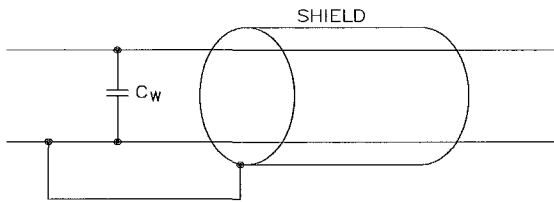


**FIGURE 13.3-4. Wire Manufacturers Capacitance Specifications, $C_w$**
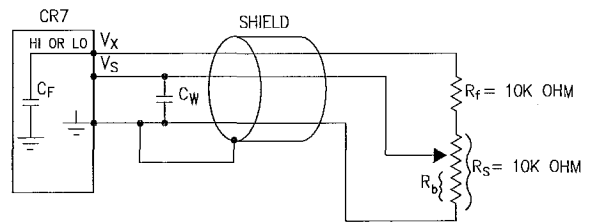


**FIGURE 13.3-5. Model 024A Wind Direction Sensor**

## DIELECTRIC ABSORPTION

The dielectric absorption of insulation surrounding individual conductors can seriously effect the settling waveform by increasing the time required to settle as compared to a simple exponential. Dielectric absorption is difficult to quantify but it can have a serious effect on low level measurements, for example 50mV or less. The primary rule to follow in minimizing dielectric absorption is: AVOID PVC INSULATION around conductors. PVC cable jackets are permissible since the jackets don't contribute to the lead capacitance because the jacket is outside the shield. Campbell Scientific uses only polyethylene and polypropylene insulated conductors in CR7 sensors (see Table 13.3-2) since these materials have negligible dielectric absorption. Teflon insulation is also very good but quite expensive.

## 13.3.2 EFFECT OF LEAD LENGTH ON SIGNAL RISE TIME

In the 024A Windvane, a potentiometer sensor, the peak transient voltage is much less than the true signal voltage (Table 13.3-5). This means the signal rise time is the major source of error and the time constant is the same as if $C_w$ were between the signal lead and ground as represented below.

$R_o$, the source resistance, is not constant because $R_b$ varies from 0 to 10 kohms over the 0 to 360 degree wind direction range. The source resistance is given by:

$$R_o = R_b(R_s-R_b+R_f)/(R_s+R_f) = R_b(20k-R_b)/20k$$
[13.3-12]

Note that at $360°$, $R_o$ is at a maximum of 5k ($R_b$=10k) and at $0°$, $R_o$ is 0 ($R_b$=0). It follows that settling errors are less at lower direction values.

The value of $R_b$ for any direction D (degrees) is given by:

$$R_b(kohms) = (10k)(D)/360$$
[13.3-13]

Equation 13.3-6 can be rewritten to yield the settling error of a rising signal directly in units of degrees.

$$\text{Error (degrees)} = De^{-t/(R_o(C_f+C_wL))}$$
[13.3-14]

Equation 13.3-12, -13 and -14 can be combined to estimate the error directly in degrees at various directions and lead lengths (Table 13.3-3). Constants used in the calculations are given below:

$$C_f = 0.01 \text{ ufd}$$

$$C_w = 41 \text{ pfd/ft, Belden \#8771 wire}$$

$$t = 0.5 \text{ ms}$$

## TABLE 13.3-3. Settling Error (Degrees) for 024A Wind Direction Sensor vs. Lead Length

| Wind Direction | - - - - - Error - - - - - | |
| | L=1000 ft. | L=500 ft. |
|---|---|---|
| $360^\circ$ | $47^\circ$ | $8^\circ$ |
| $270^\circ$ | $31^\circ$ | $5^\circ$ |
| $180^\circ$ | $12^\circ$ | $1^\circ$ |
| $90^\circ$ | $1^\circ$ | $0^\circ$ |

The values in Table 13.3-3 show that significant error occurs at large direction values for leads in excess of 250 feet. Instruction 4, Excite, Delay and Measure should be used to eliminate errors in these types of situations. Using a 10ms delay, settling errors are eliminated up to lengths that exceed the drive capability of the excitation channel ($\approx$2000 ft.).

## 13.3.3 TRANSIENTS INDUCED BY SWITCHED EXCITATION

Figure 13.3-6 shows a typical half-bridge, resistive sensor such as Campbell Scientific's Model 107 Temperature probe, connected to the CR7. The leadwire is a single shielded pair, used for conducting the excitation, $V_x$ and signal, $V_s$ voltages. When $V_x$ is switched on, a transient is capacitively induced in $V_s$, the signal voltage. If the peak transient level, $V_{eo}$, is less than the true signal, $V_{so}$, the transient has no effect on the measurement but if $V_{eo}$ is greater than $V_{so}$ it must settle to the correct signal voltage to avoid errors.

NOTE: Excitation transients are eliminated if an option exists to contain excitation leads in a shield independent from the signal leads.
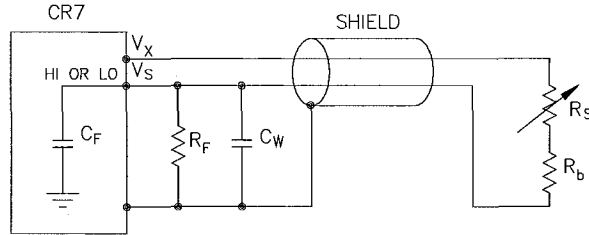


**FIGURE 13.3-6. Resistive Half-Bridge Connected to Single-Ended CR7 Input**

The size of the peak transient is linearly related to the excitation voltage and increases as the bridge resistor, $R_f$, increases. Table 13.3-4 shows measured levels of $V_{eo}$ for 1000 foot lengths of three Belden wires used in Campbell Scientific sensors. Values are given for $R_f$ equal 1 kohm and 10 kohm. Table 13.3-4 is meant only to provide estimates of the size of excitation transients encountered since the exact level will depend upon the specific sensor configuration.

Equation 13.3-7 can be solved for the maximum lead length, L, permitted to maintain a specified error limit. Combining Equations 7 and 4 and solving for L gives:

$$L = -(R_o C_f + (t/\ln(V_e/V_{eo})))/R_o C_w \qquad [13.3\text{-}15]$$

where $V_e$ is the measurement error limit.

## TABLE 13.3-4. Measured Peak Excitation Transients for 1000 Foot Lengths of Three Belden Lead Wires Used by Campbell Scientific

| $V_x$(mV) | - - - - - - - - - - $V_{eo}$(mV) - - - - - - - - - | | | | | |
| | $R_f$=1 kohm | | | $R_f$=10 kohm | | |
| | # 8641 | # 8771 | # 8723 | # 8641 | # 8771 | # 8723 |
|---|---|---|---|---|---|---|
| 5000 | 125 | 200 | 130 | 215 | 320 | 180 |
| 4000 | 100 | 165 | 110 | 180 | 260 | 150 |
| 3000 | 75 | 130 | 90 | 140 | 200 | 110 |
| 2000 | 50 | 100 | 60 | 100 | 140 | 80 |
| 1000 | 25 | 65 | 40 | 60 | 90 | 40 |

## EXAMPLE LEAD LENGTH CALCULATION FOR CAMPBELL SCIENTIFIC 107 TEMPERATURE SENSOR

Assume a limit of 0.05°C over a 0°C to +40°C range is established for the transient settling error. This limit is a reasonable choice since it approximates the linearization error over that range. The output signal from the thermistor bridge varies non-linearly with temperature (refer to 107 Operator's Manual), ranging from about 200 $\mu V$/°C at 0°C to 100 $\mu V$/°C at 40°C. Taking the most conservative figure yields an error limit of $V_e = 5\ \mu V$. The other values needed to calculate the maximum lead length are summarized in Table 13.3-5 and listed below:

1) $V_{eo} \approx 100mV$, peak transient at 4V excitation

2) $V_e \approx 5\mu V$, allowable measurement error

3) $t = 500\mu s$, CR7 input settling time

4) $R_O = 1kohm$, 107 probe source resistance

5) $C_f = 0.01nfd$, CR7 input capacitance

6) $C_W = 42pfd/ft.$, lead wire capacitance

Solving Equation 13.3-15 gives a maximum lead length of:

$$L \approx 965\ ft., error \approx 0.05°C$$

Setting the allowable error at 0.1°C or approximately 10$\mu V$, the maximum lead length increases to:

$$L \approx 1050\ ft., error \approx 0.1°C$$

## 13.3.4 SUMMARY OF SETTLING ERRORS FOR CAMPBELL SCIENTIFIC RESISTIVE SENSORS.

Table 13.3-5 summarizes the data required to estimate the effect of lead length on settling errors for Campbell Scientific's resistive sensors. Comparing the transient level, $V_{eo}$, to the input range, one suspects that transient errors are the most likely limitation for the 107 sensor. The sensors in the WVU-7 are the same as in the Model 107 (the lead wire is different) but the signal leads for the WVU-7 wet and dry bulbs are not subject to excitation transients because they are shielded independently from the excitation.

The comparatively small transient yet large source resistance of the 024A sensor indicates that signal rise time may be the most important limitation. The analysis in Section 13.3.2 confirms this.

The Model 227 Soil Moisture Block has a relatively short time constant and essentially no transient. Lead lengths in excess of 2000 feet produce less than a 0.1 bar (0-10 bar range) input settling error. With this sensor, the drive capability of the excitation channel limits the lead length. If the capacitive load exceeds 0.1 ufd and the resistive load is negligible, $V_x$ will oscillate about it's control point. If the capacitive load is 0.1 ufd or less, $V_x$ will settle to within 0.1% of its correct value in 150$\mu s$. A lead length of 2000 feet is permitted for the Model 227 before approaching the drive limitation.

**TABLE 13.3-5. Summary of Input Settling Data for Campbell Scientific Resistive Sensors**

| Sensor Model # | Belden Wire # | $R_O$ (kohms) | $C_W$ (pfd/ft.) | $\tau^*$ (us) | Input Range(mV) | $V_x$(mV) | $V_{eo}$(mV)** |
|---|---|---|---|---|---|---|---|
| 107 | 8641 | 1 | 42 | 52 | 15 | 4000 | 100 |
| 207(RH) | 8771 | 1 | 41 | 51 | 150 | 3000 | 130 |
| WVU-7 | 8723 | 1 | 62 | 72 | 15 | 4000 | 0 |
| 227 | 8641 | 0.1-1 | 42 | 5-52 | 500 | 500 | 0 |
| 237 | 8641 | 1 | 42 | 52 | 50 | 5000 | 125 |
| 024A | 8771 | 0-5 | 41 | 1-255 | 500 | 1000 | 0-90 |

* Estimated time constants are for 1000 foot lead lengths and include 0.01nfd CR7 input capacitance.
** Measured peak transients for 1000 foot lead lengths at corresponding excitation, $V_x$.

Table 13.3-6 summarizes maximum lead lengths for corresponding error limits in six Campbell Scientific sensors. Since the first three sensors are non-linear, the voltage error, $V_e$, is the most conservative value corresponding to the error over the range shown.

## MINIMIZING SETTLING ERRORS IN NON-CAMPBELL SCIENTIFIC SENSORS

When long lead lengths are mandatory in sensors configured by the user, the following general practices can be used to minimize or measure settling errors:

1.  When measurement speed is not a prime consideration, Instruction 4 (Excite, Delay and Measure) can be used to ensure ample settling time for half-bridge, single-ended sensors.

2.  An additional low value bridge resistor can be added to decrease the source resistance, $R_o$. For example, assume a YSI non-linear thermistor such as the model 44032 is used with a 30 kohm bridge resistor, $R'_f$. A typical configuration is shown in Figure 13.3-7A. The disadvantage with this configuration is the

high source resistance shown in column 3 of Table 13.3-7. Adding another 1K resistor, $R_f$, as shown in Figure 13.3-7B lowers the source resistance of the CR7 input but offers no improvement over configuration A because $R'_f$ still combines with the lead capacitance to slow the signal response at point P. The source resistance at point P (column 5) is essentially the same as the input source resistance of configuration A. Moving $R_f'$ out to the thermistor as shown in Figure 13.3-7C optimizes the signal settling time because it becomes a function of $R_f$ and $C_w$ only.

Columns 4 and 7 list the signal voltages as a function of temperature with a 5V excitation for configurations A and C, respectively. Although configuration A has a higher output signal (5V input range), it does not yield any higher resolution than configuration C which uses the ±150 mV input range.

> **NOTE:** Since $R_f'$ attenuates the signal in configurations B and C, one might consider eliminating it altogether. However, its inclusion "flattens" the non-linearity of the thermistor, allowing more accurate curve fitting over a broader temperature range.

**TABLE 13.3-6. Maximum Lead Length vs. Error for Campbell Scientific Resistive Sensors**

| Sensor Model # | Error | Range | $V_e(\mu V)$ | Maximum Length(ft.) |
|---|---|---|---|---|
| 107 | 0.05°C | 0°C to 40°C | 5 | 965[1] |
| 207(RH) | 1%RH | 20% to 90% | 500 | 1950[3] |
| WVU-7 | 0.05°C | 0°C to 40°C | 5 | 850[2] |
| 024A | 3° | @ 360° | 1390 | 250[2] |
| 227 | - | - | - | 2000[3] |
| 237 | 10 kohm | 20k to 300k | 500 | 1900[3] |

[1] based on transient settling
[2] based on signal rise time
[3] limit of excitation drive

TABLE 13.3-7.  Source Resistances and Signal Levels for YSI #44032 Thermistor Configurations
Shown in Figure 13.3-7 (2V Excitation)

| T | $R_S$ (kohms) | --- A --- | | - B - | --- C --- | |
|---|---|---|---|---|---|---|
| | | $R_O$ (kohms) | $V_S$(mV) | $R_O$@P (kohms) | $R_O$ (kohms) | $V_S$(mV) |
| -40 | 884.6 | 29.0 | 164 | 30.0 | 1 | 5.5 |
| -20 | 271.2 | 27 | 498 | 27.8 | 1 | 16.5 |
| 0 | 94.98 | 22.8 | 1200 | 23.4 | 1 | 39.5 |
| +25 | 30.00 | 15.0 | 2500 | 15.2 | 1 | 82.0 |
| +40 | 16.15 | 10.5 | 3250 | 10.6 | 1 | 106.0 |
| +60 | 7.60 | 6.1 | 3989 | 6.1 | 1 | 129.5 |

3.  Where possible run excitation leads and signal leads in separate shields to minimize transients.

4.  AVOID PVC INSULATED CONDUCTORS to minimize the effect of dielectric absorption on input settling time.

5.  Use the CR7 to measure the input settling error associated with a given configuration. For example  assume long leads are required but the lead capacitance, $C_W$, is unknown.  Configure $R_f$ on a length of cable similar to the measurement.  Leave the sensor end open as shown in Figure 13.3-8 and measure the result using the same instruction parameters to be used with the sensor.  The measured deviation from 0V is the input settling error.

A) $Ro = RsRf/(Rs+Rf)$, $Vs = VxRf/(Rs+Rf)$



B) $Ro@P = Rs(Rf+Rf)/(Rs+Rf+Rf)$



C) $Ro = Rf$, $Vs = VxRf/(Rs+Rf+Rf)$
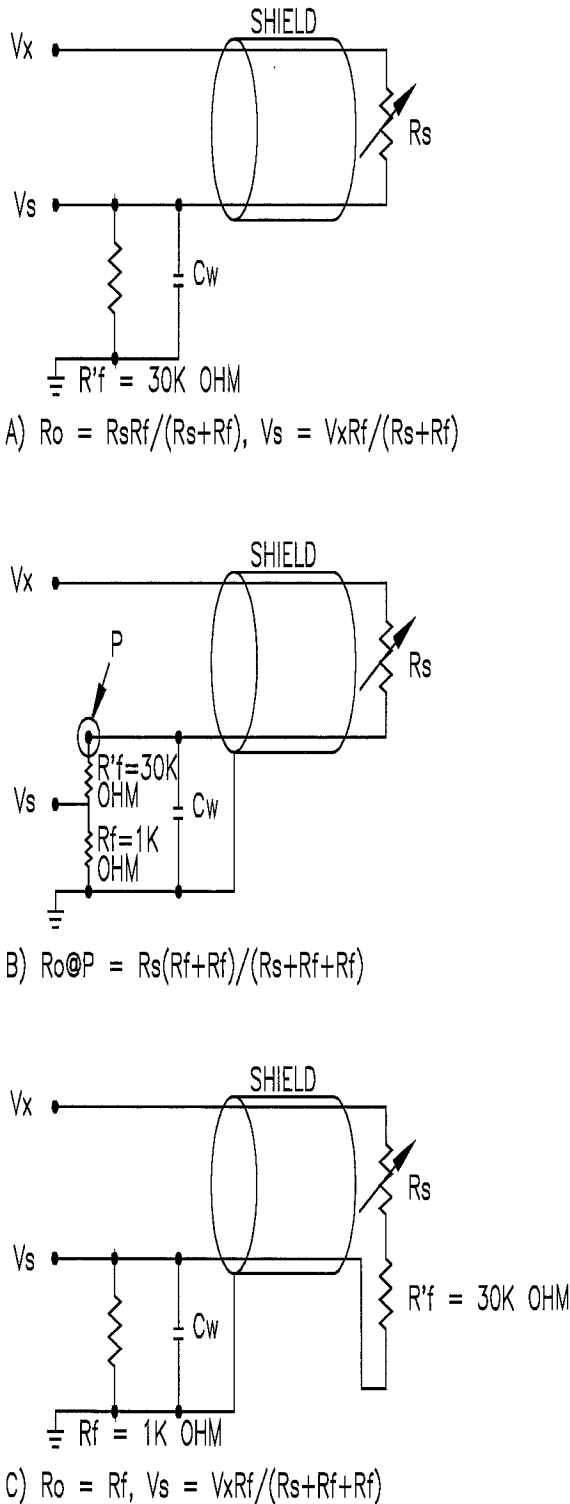
**FIGURE 13.3-7. Half-Bridge Configuration for YSI #44032 Thermistor Connected to CR7 Showing: A) Large source resistance, B) Large source resistance at point P, and C) Configuration optimized for input settling.**
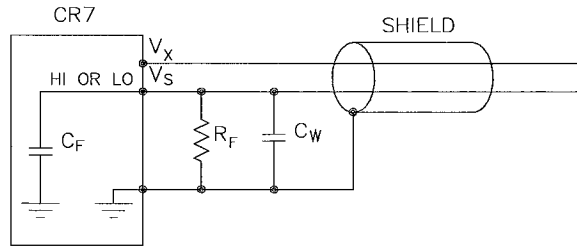


**FIGURE 13.3-8. Measuring Input Settling Error with the CR7**

6. Most Campbell Scientific sensors are configured with a small bridge resistor, $R_f$, (typically 1 kohm) to minimize the source resistance. If the lead length of a Campbell Scientific sensor is extended by connecting to the pigtails directly, the effect of the lead resistance, $R_l$, on the signal must be considered. Figure 13.3-9 shows a Campbell Scientific Model 107 sensor with 500 feet of extension lead connected directly to the pigtails. Normally the signal voltage is proportional to $R_f/(R_s+R_b+R_f)$ but when the pigtails are extended the signal is proportional to $(R_f+R_l)/(R_s+R_b+R_f+R_l)$. $R_l$ is much smaller than the other terms in the denominator and can be discarded. The effect on the signal can be analyzed by taking the ratio of the signal with extended leads, $V_{sl}$ to the normal signal, $V_s$:

$$V_{sl}/V_s = (R_f+R_l)/R_f$$

Plugging in values of $R_f$=1k and $R_l$=.012k (500Ω at 23ohms/1000Ω, Table 13.3-2) gives an approximate 1% error in the signal with extended leads. Converting the error to °C gives approximately a 0.3°C error at 0°C, 0.6°C error at 20°C and a 1.5°C error at 40°C. The error can be avoided by maintaining the pigtails on the CR7 end of the extended leads because $R_l$ does not add to the bridge completion resistor, $R_f$, and its influence on the thermistor resistance is negligible.
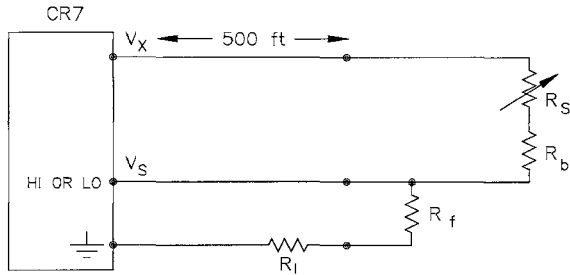
**FIGURE 13.3-9. Incorrect Leadwire Extension on Model 107 Temperature Sensor**

## 13.4 THERMOCOUPLE MEASUREMENTS

A thermocouple consists of two wires, each of a different metal or alloy, which are joined together at each end. If the two junctions are at different temperatures, a voltage proportional to the difference in temperatures is induced in the wires. When a thermocouple is used for temperature measurement, the wires are soldered or welded together at the measuring junction. The second junction, which becomes the reference junction, is formed where the other ends of the wires are connected to the measuring device. (With the connectors at the same temperature, the chemical dissimilarity between the thermocouple wire and the connector does not induce any voltage.) When the temperature of the reference junction is known, the temperature of the measuring junction can be determined by measuring the thermocouple voltage and adding the corresponding temperature difference to the reference temperature.

The CR7 determines thermocouple temperatures using the following sequence. First the temperature of the reference junction is measured. If the reference junction is the CR7 I/O Module, the temperature is measured with the PRT in the 723-T Analog Input Card (Instruction 17). The reference junction temperature in $^{\circ}$C is stored in an input location which is accessed by the thermocouple measurement instruction (Instruction 13 or 14). The CR7 calculates the voltage that a thermocouple of the type specified would output at the reference junction temperature if its reference junction were at $0^{\circ}$C, and adds this voltage to the measured thermocouple voltage. The temperature of the measuring junction is then calculated from a polynomial approximation of the NBS TC calibrations. If

the CR7 has been instructed to calculate the temperature difference between the reference and measuring junctions it will subtract the reference temperature before storing the temperature value.

### 13.4.1 ERROR ANALYSIS

The error in the measurement of a thermocouple temperature is the sum of the errors in the reference junction temperature, the thermocouple output (deviation from standards published in NBS Monograph 125), the thermocouple voltage measurement, and the linearization error (difference between NBS standard and CR7 polynomial approximations). The discussion of errors which follows is limited to these errors in calibration and measurement and does not include errors in installation or matching the sensor to the environment being measured.

**REFERENCE JUNCTION TEMPERATURE WITH 723-T**

The PRT in the CR7 is mounted in the center of the 723-T terminal strip. This resistance temperature device (RTD) is accurate to $\pm0.1^{\circ}$C over the CR7 operating range. The I/O Module was designed to minimize thermal gradients. It is encased in an aluminum box which is thermally isolated from the CR7 enclosure. Heavy copper grounding bars underlying the terminal strips on the I/O cards and large brass bars running the length of the I/O Module provide thermal conduction for rapid equilibration of thermal gradients. Sources of heat within the CR7 enclosure exist due to power dissipation by the electronic components or charging batteries. In a situation where the CR7 is at an ambient temperature of approximately $20^{\circ}$C and no external temperature gradients exist, the temperature gradient between one end of an Analog Input card to the other is likely to be $0.05^{\circ}$C, and the gradient between the cards, from one end of the I/O Module to the other, is likely to be 0.1 to $0.2^{\circ}$C. The end of the module with the CPU card will be warmer due to heat dissipated by the processor.

Given the above conditions, if it is desired to make a series of thermocouple measurements with the reference junctions within $0.05^{\circ}$C of the RTD temperature, the temperature obtained from the 723-T card can be used for

thermocouples attached to it and to one 723 Analog Input card to either side of it (i.e. Analog Input cards 1,2, and 3, where card 2 contains RTD). If more than these three cards are used, it is necessary to measure a new reference temperature to stay within the desired 0.05°C limit. This can be done by using one of the thermocouples from the first set of measurements to measure the reference temperature for the next set. The second reference temperature could provide the reference for another bank of three cards. The measuring junction for this reference should be clamped (along with the lead from the thermocouple being measured) into one of the inputs in the center of the second card in this bank. If more severe temperature gradients

within the I/O Module are anticipated or suspected this technique can be used to quantify these gradients and supply additional reference temperatures if necessary.

## THERMOCOUPLE LIMITS OF ERROR

The standard reference which lists thermocouple output voltage as a function of temperature (reference junction at 0°C) is the National Bureau of Standards Monograph 125 (1974). The American National Standards Institute has established limits of error on thermocouple wire which is accepted as an industry standard (ANSI MC 96.1, 1975). Table 13.4-1 gives the ANSI limits of error for standard and special grade thermocouple wire of the types accommodated by the CR7.

**TABLE 13.4-1. Limits of Error for Thermocouple Wire (Reference Junction at 0°C)**

| Thermocouple Type | Temperature Range °C | Limits of Error (Whichever is greater) | |
| --- | --- | --- | --- |
| | | Standard | Special |
| T | -200 to 0 | ± 1.0°C or 1.5% | |
| | 0 to 350 | ± 1.0°C or 0.75% | ± 0.5°C or 0.4% |
| J | 0 to 750 | ± 2.2°C or 0.75% | ± 1.1°C or 0.4% |
| E | -200 to 0 | ± 1.7°C or 1.0% | |
| | 0 to 900 | ± 1.7°C or 0.5% | ± 1.0°C or 0.4% |
| K | -200 to 0 | ± 2.2°C or 2.0% | |
| | 0 to 1250 | ± 2.2°C or 0.75% | ± 1.1°C or 0.4% |
| R or S | 0 to 1450 | ± 1.5°C or 0.25% | ± 0.6°C or 0.1% |
| B | 800 to 1700 | ± 0.5% | Not Estab. |

When both junctions of a thermocouple are at the same temperature there is no voltage produced (law of intermediate metals). A consequence of this is that a thermocouple can not have an offset error; any deviation from a standard (assuming the wires are each homogeneous and no secondary junctions exist) is due to a deviation in slope. In light of this, the fixed temperature limits of error (e.g., ±1.0°C for type T as opposed to the slope error of 0.75% of the temperature) in the table above are probably greater than one would experience when considering temperatures in the environmental range (i.e., the reference junction, at 0°C, is relatively close to the

temperature being measured, so the absolute error - the product of the temperature difference and the slope error - should be closer to the percentage error than the fixed error). Likewise, because thermocouple calibration error is a slope error, accuracy can be increased when the reference junction temperature is close to the measurement temperature. For the same reason differential temperature measurements, over a small temperature gradient, can be extremely accurate.

In order to quantitatively evaluate thermocouple error when the reference junction is not fixed at 0 °C, one needs limits of error for the Seebeck coefficient (slope of thermocouple voltage vs. temperature curve) for the various thermocouples. Lacking this information, a reasonable approach is to apply the percentage errors, with perhaps 0.25% added on, to the difference in temperature being measured by the thermocouple.

## ACCURACY OF THE THERMOCOUPLE VOLTAGE MEASUREMENT

The accuracy of a CR7 voltage measurement is specified as 0.02% (0.01% 0 to 40 °C) of the full scale range being used to make the measurement. The actual accuracy may be better than this as it involves a slope error (the error is proportional to the measurement being made, though limited by the resolution). The error in the temperature due to inaccuracy in the measurement of the thermocouple voltage is worst at temperature extremes, where a relatively large scale is necessary to read the thermocouple output. For example, assume type K (chromel-alumel) thermocouples are used to measure temperatures at 1000 °C. The TC output is on the order of 40mV, requiring the ±50mV input range. The accuracy specification of 0.01% FSR is 10μV which is a measurement error of about 0.2 °C. In the environmental temperature range with voltage measured on an appropriate scale, error in temperature due to the voltage measurements is a few hundredths of a degree.

## THERMOCOUPLE POLYNOMIAL: Voltage to Temperature

NBS Monograph 125 gives high order polynomials for computing the output voltage of a given thermocouple type over a broad range of temperatures. In order to speed processing and accommodate the CR7's math and storage capabilities, 4 separate 6th order polynomials are used to convert from volts to temperature over the range covered by each thermocouple type. Table 13.4-2 gives error limits for the thermocouple polynomials.

### TABLE 13.4-2. Limits of Error on CR7 Thermocouple Polynomials (Relative to NBS Standards)

| TC Type | Range °C | | | Limits of Error °C |
|---------|------|----|------|---------|
| T | -270 | to | 400 | |
| | -270 | to | -200 | +18 @ -270 |
| | -200 | to | -100 | ±0.08 |
| | -100 | to | 100 | ±0.001 |
| | 100 | to | 400 | ±0.015 |
| J | -150 | to | 760 | ±0.008 |
| | -100 | to | 300 | ±0.002 |
| E | -240 | to | 1000 | |
| | -240 | to | -130 | ±0.4 |
| | -130 | to | 200 | ±0.005 |
| | 200 | to | 1000 | ±0.02 |
| K | -50 | to | 1372 | |
| | -50 | to | 950 | ±0.01 |
| | 950 | to | 1372 | ±0.04 |

## REFERENCE JUNCTION COMPENSATION: Temperature to Voltage

The polynomials used for reference junction compensation (converting reference temperature to equivalent TC output voltage) do not cover the entire thermocouple range. Substantial errors will result if the reference junction temperature is outside of the linearization range. The ranges covered by these linearizations include the CR7 environmental operating range, so there is no problem when the CR7 is used as the reference junction. External reference junction boxes however, must also be within these temperature ranges. Temperature difference measurements made outside of the reference temperature range should be made by obtaining the actual temperatures referenced to a junction within the reference temperature range and subtracting as described in Section 7.5. Table 13.4-3 gives the reference temperature ranges covered and the limits of error in the linearizations within these ranges.

Two sources of error arise when the reference temperature is out of range. The most significant error is in the calculated compensation voltage, however error is also created in the temperature difference calculated

from the thermocouple output. For example, suppose the reference temperature for a measurement on a type T thermocouple is 300 °C. The compensation voltage calculated by the CR7 corresponds to a temperature of 272.6 °C, a -27.4 °C error. The type T thermocouple with the measuring junction at 290 °C and reference at 300 °C would output -578.7 μV; using the reference temperature of 272.6 °C, the CR7 calculates a temperature difference of -10.2 °C, a -0.2 °C error. The temperature calculated by the CR7 would be 262.4 °C, 27.6 °C low.

#### TABLE 13.4-3. Reference Temperature Compensation Range and Polynomial Error Relative to NBS Standards

| TC Type | Range °C | Limits of Error °C |
|---------|----------|--------------------|
| T | -100 to 100 | ± 0.001 |
| J | -150 to 296 | ± 0.005 |
| E | -150 to 206 | ± 0.005 |
| K | -50 to 100 | ± 0.01 |

## ERROR SUMMARY

The magnitude of the errors described in the previous sections illustrate that the greatest sources of error in a thermocouple temperature measurement with the CR7 are likely to be due to the limits of error on the thermocouple wire and in the reference temperature determined with the 723-T RTD. Errors in the thermocouple and reference temperature linearizations are extremely small, and error in the voltage measurement is negligible.

To illustrate the relative magnitude of these errors in the environmental range, we will take a worst case situation where all errors are maximum and additive. A temperature of 45 °C is measured with a type T (copper-constantan) thermocouple, using the ±5 mV range. The nominal accuracy on this range is 1μV(0.01% of 10mV) which at 45 °C changes the temperature by 0.012 °C. The RTD is 25 °C but is indicating 25.1 °C, and the terminal that the thermocouple is connected to is 0.05 °C cooler than the RTD.

#### TABLE 13.4-4. Example of Errors in Thermocouple Temperature

| Source | Error °C | % of Total Error 1°C Error | 1% Slope Error |
|--------|----------|----------------------------|----------------|
| Reference Temp. | 0.15 | 12.8 | 39.9 |
| TC Output ANSI | 1.0 | 85.0 | |
| 0.01 x 20°C | 0.2 | | 53.2 |
| Voltage Measurement | 0.024 | 2.0 | 6.3 |
| Reference Linearization | 0.001 | 0.1 | 0.3 |
| Output Linearization | 0.001 | 0.1 | 0.3 |
| Total Error With ANSI error | 1.176 | 100 | |
| Assuming 1% slope error | 0.376 | | 100 |

### 13.4.2 USE OF EXTERNAL REFERENCE JUNCTION OR JUNCTION BOX

An external junction box is often used to facilitate connections and to reduce the expense of thermocouple wire when the temperature measurements are to be made at a distance from the CR7. In most situations it is preferable to make the box the reference junction in which case its temperature is measured and used as the reference for the thermocouples and copper wires are run from the box to the CR7 (Section 7.4). Alternatively, the junction box can be used to couple extension grade thermocouple wire to the thermocouples being used for measurement, and the CR7 I/O Module used as the reference junction. Extension grade thermocouple wire has a smaller temperature range than standard thermocouple wire, but meets the same limits of error within that range. The only situation where it would be necessary to use extension grade wire instead of a external measuring junction is where the junction box temperature is outside the range of reference junction compensation provided by the CR7. This is only a factor when using type K thermocouples, where the upper limit of the reference compensation linearization

is 100 °C and the upper limit of the extension grade wire is 200 °C. With the other types of thermocouples the reference compensation range equals or is greater than the extension wire range. In any case, errors can arise if temperature gradients exist within the junction box.

Figure 13.4-1 illustrates a typical junction box. Terminal strips will be a different metal than the thermocouple wire. Thus, if a temperature gradient exists between A and A' or B and B', the junction box will act as another thermocouple in series, creating an error in the voltage measured by the CR7. This thermoelectric offset voltage is a factor whether or not the junction box is used for the reference. it can be minimized by making the thermal conduction between the two points large and the distance small. The best solution in the case where extension grade wire is being connected to thermocouple wire would be to use connectors which clamped the two wires in contact with each other.
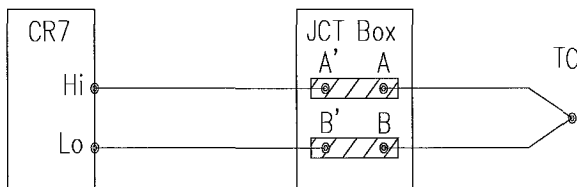


**Figure 13.4-1. Diagram of Junction Box**

An external reference junction box must be constructed so that the entire terminal area is very close to the same temperature. This is necessary so that a valid reference temperature can be measured and to avoid a thermoelectric offset voltage which will be induced if the terminals at which the thermocouple leads are connected (points A and B in Figure 13.4-2) are at different temperatures. The box should contain elements of high thermal conductivity, which will act to rapidly equilibrate any thermal gradients to which the box is subjected. It is not necessary to design a constant temperature box, it is desirable that the box respond slowly to external temperature fluctuations.

Radiation shielding must be provided when a junction box is installed in the field. Care must also be taken that a thermal gradient is not induced by conduction through the incoming wires. The CR7 can be used to measure the temperature gradients within the junction box.

## 13.5 BRIDGE RESISTANCE MEASUREMENTS

There are five bridge measurement instructions included in the standard CR7 software. Figure 13.5-1 shows the circuits that would typically be measured with these instructions. In the diagrams, the resistors labeled $R_s$ would normally be the sensors and those labeled $R_f$ would normally be fixed resistors. Circuits other than those diagrammed could be measured, provided the excitation and type of measurements were appropriate.

With the exception of Instruction 4, which applies an excitation voltage then waits a specified time before making a single ended measurement, all of the bridge measurements make one set of measurements with the excitation as programmed and another set of measurements with the excitation polarity reversed. The error in the two measurements due to thermal emfs can then be accounted for in the processing of the measurement instruction. In Instructions 6-9 the excitation channel maintains the excitation voltage until after the analog to digital conversion is completed. In Instruction 5, the AC half bridge grounds the excitation channel as soon as the integration portion of the measurement is completed. Figure 13.5-2 shows the excitation and measurement sequence for Instruction 6, a 4 wire full bridge. When more than one measurement per sensor is necessary (Instructions 7 and 9), excitation is applied separately for each measurement (e.g., in Instruction 9 used for a 4 wire half bridge, the differential measurement of the voltage drop across the sensor is made with the excitation at both polarities and then excitation is again applied and reversed for the single ended measurement of the voltage drop across the fixed resistor.

| INSTR. # | DIAGRAM | DESCRIPTION | RESULT = X |
|---|---|---|---|
| 4 | | DC HALF BRIDGE WITH USER ENTERED SETTLING TIME | $X = V_1 = V_x \dfrac{R_s}{R_s + R_f}$ |
| 5 | | AC HALF BRIDGE EXCITATION ALTERNATES POLARITY FOR ION DEPOLARIZATION | $X = \dfrac{V_1}{V_x} = \dfrac{R_s}{R_s + R_f}$ |
| 6 | | 4 WIRE FULL BRIDGE | $X = 1000 \dfrac{V_1}{V_x} = 1000 \left( \dfrac{R_3}{R_3 + R_4} - \dfrac{R_2}{R_1 + R_2} \right)$ |
| 7 | | 3 WIRE HALF BRIDGE | $X = \dfrac{2V_2 - V_1}{V_x - V_1} = \dfrac{R_s}{R_f}$ |
| 9 | | 6 WIRE FULL BRIDGE WITH EXCITATION LEAD COMPENSATION | $X = 1000 \dfrac{V_2}{V_1} = 1000 \left( \dfrac{R_3}{R_3 + R_4} - \dfrac{R_2}{R_1 + R_2} \right)$  ($V_1$ ON 2.5 V RANGE) |
| 9 | | 4 WIRE HALF BRIDGE | $X = \dfrac{V_2}{V_1} = \dfrac{R_s}{R_f}$  ($V_1$ NOT ON 2.5 V RANGE) |



**FIGURE 13.5-1.  Circuits Used with Instructions 4-9**
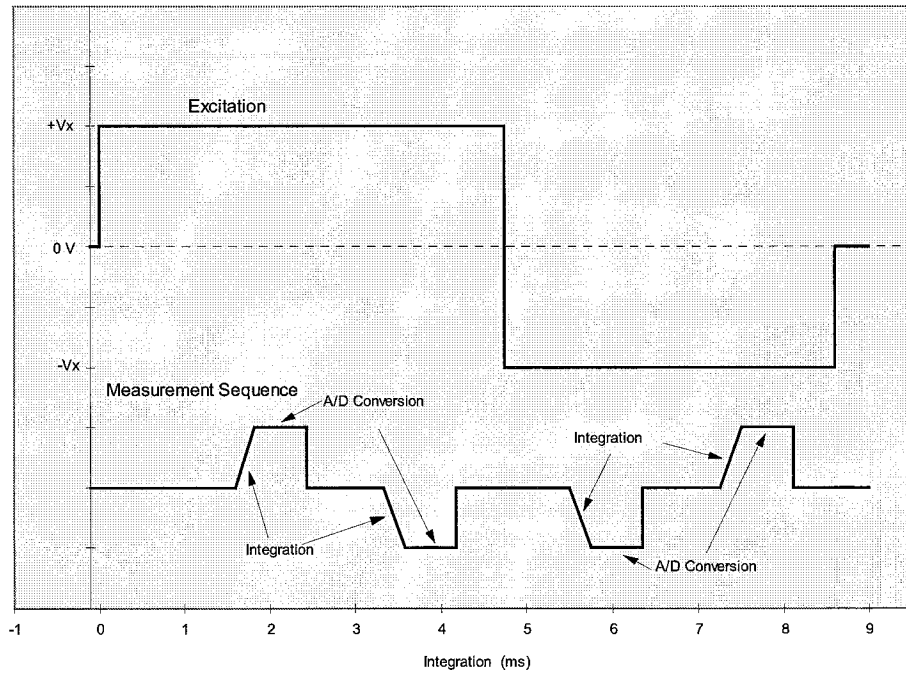
FIGURE 13.5-2. Excitation and Measurement Sequence for 4 Wire Full Bridge

**TABLE 13.5-1. Comparison of Bridge Measurement Instructions**

| Instr. | Circuit | Description |
|---|---|---|
| 4 | DC Half Bridge | User entered settling time allows compensation for capacitance in long lead lengths. No polarity reversal. One single-ended measurement. Measured voltage output. |
| 5 | AC Half Bridge | Rapid reversal of excitation polarity for ion depolarization. One single-ended measurement at each excitation polarity. Ratiometric output. |
| 6 | 4 Wire Full Bridge | Slightly lower noise than 9. One differential measurement at each excitation polarity. Ratiometric output. |
| 7 | 3 Wire Half Bridge | Compensates for lead wire resistance, assuming resistance is same in both wires. Two single-ended measurements at each excitation polarity. Ratiometric output. |
| 9 | 6 Wire Full Bridge or 4 Wire Half Bridge | Compensates for lead wire resistance. Two differential measurements at each excitation polarity. Ratiometric output. |

Calculating the actual resistance of a sensor which is one of the legs of a resistive bridge usually requires the use of one or two Processing Instructions in addition to the bridge measurement instruction. Instruction 59 takes a value, X, in a specified input location and computes the value $MX/(1-X)$, where M is the multiplier and stores the result in the original location. Instruction 42 computes the reciprocal of a value in an input location. Table 13.5-2 lists the instructions used to compute the resistance of any single resistor shown in the diagrams in Figure 13.5-1, provided the values of the other resistors in the bridge circuit are known.

**TABLE 13.5-2.  Calculating Resistance Values from Bridge Measurement**

| Instr. | Result | Instr. | Multiplier and Offset |
|---|---|---|---|
| 4 | $X = V_x(R_s / (R_s + R_f))$ | | |
| | $R_s = R_f \dfrac{X / V_x}{1 - X / V_x}$ | 4.<br>59. | Mult. = $1/V_x$; ofs. = 0<br>Mult. = $R_f$ |
| | $R_f = \dfrac{1}{((X / V_x) / (1 - X / V_x)) / R_s}$ | 4.<br>59.<br>42. | Mult. = $1/V_x$; ofs. = 0<br>Mult. = $1/R_s$ |
| 5 | $X = R_s / (R_s + R_f)$ | | |
| | $R_s = R_f \dfrac{X}{1 - X}$ | 5.<br>59. | Mult. = 1; ofs. = 0<br>Mult. = $R_f$ |
| | $R_f = \dfrac{1}{(X / (1 - X)) / R_s}$ | 5.<br>59.<br>42. | Mult. = 1; ofs. = 0<br>Mult. = $1/R_s$; ofs. = 0 |
| 6 or 9* | $X = 1000[R_3 / (R_3 + R_4) - R_2 / (R_1 + R_2)]$ | | *used for full bridge* |
| | $R_1 = \dfrac{1}{(X_1 / (1 - X_1)) / R_2}$ | 6 or 9.<br>59.<br>42. | Mult. = -0.001; ofs. = $R_3/(R_3+R_4)$<br>Mult. = $1/R_2$ |
| | where $X_1 = -X / 1000 + R_3 / (R_3 + R_4)$ | | |
| | $R_2 = R_1(X_2 / (1 - X_2))$ | 6 or 9.<br>59. | Mult. = -0.001; ofs. = $R_3/(R_3+R_4)$<br>Mult. = $R_1$ |
| | where $X_2 = X_1$ | | |
| | $R_3 = R_4(X_3 / (1 - X_3))$ | 6 or 9.<br>59. | Mult. = 0.001; ofs. = $R_2/(R_1+R_2)$<br>Mult. = $R_4$ |
| | where $X_3 = X / 1000 + R_2 / (R_1 + R_2)$ | | |
| | $R_4 = \dfrac{1}{(X_4 / (1 - X_4)) / R_3}$ | 6 or 9.<br>59.<br>42. | Mult. = 0.001; ofs. = $R_2/(R_1+R_2)$<br>Mult. = $1/R_3$ |
| | where $X_4 = X_3$ | | |
| 7&9* | $X = R_s / R_f$ | | *used as half bridge* |
| | $R_s = R_f X$ | 7 or 9. | Mult. = Rf; ofs. = 0 |
| | $R_f = R_s / X$ | 7 or 9<br>42. | Mult. = $1/R_s$; ofs. = 0 |

## 13.6 RESISTANCE MEASUREMENTS REQUIRING AC EXCITATION

Some resistive sensors require AC excitation. These include the 207 relative humidity probe, soil moisture blocks, water conductivity sensors and wetness sensing grids. The use of DC excitation with these sensors can result in polarization, which will cause an erroneous measurement, and may shift the calibration of the sensor and/or lead to its rapid decay.

The AC half bridge Instruction 5 (incorporated into the 207 relative humidity measurement Instruction 12) reverses excitation polarity to provide ion depolarization and, in order to minimize the time excitation is on, grounds the excitation as soon as the signal is integrated (Figure 13.6-1). The slow integration time should never be used with a sensor requiring AC excitation because it results in the excitation lasting about 20 times as long, allowing polarization to affect the measurement.
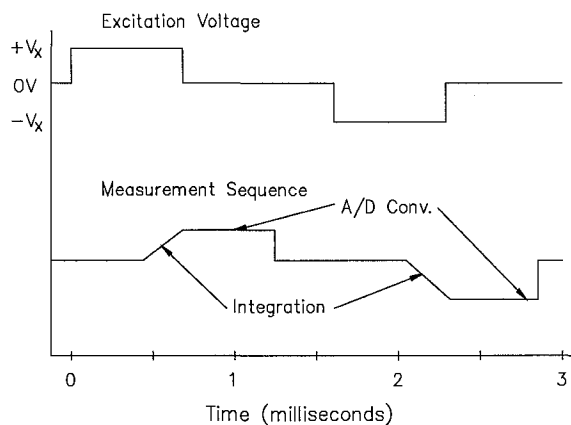


**FIGURE 13.6-1. AC Excitation and Measurement Sequence for AC Half-Bridge**

### INFLUENCE OF GROUND LOOP ON MEASUREMENTS

When measuring soil moisture blocks or water conductivity the potential exists for a ground loop which can adversely affect the measurement. This ground loop arises because the soil and water provide an alternate path for the excitation to return to CR7 ground, and can be represented by the model diagrammed in Figure 13.6-2.
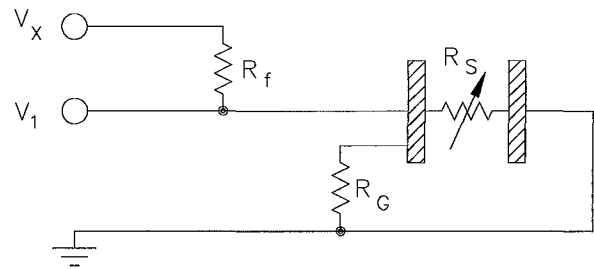


**FIGURE 13.6-2. Model of Resistive Sensor with Ground Loop**

In Figure 13.6-2, $V_x$ is the excitation voltage, $R_f$ is a fixed resistor, $R_s$ is the sensor resistance, and $R_G$ is the resistance between the excited electrode and CR7 earth ground. With $R_G$ in the network, the measured signal is:

$$V_1 = V_x \frac{R_s}{(R_s + R_f) + R_s R_f / R_G} \qquad [13.6\text{-}1]$$

$R_s R_f / R_G$ is the source of error due to the ground loop. When $R_G$ is large the equation reduces to the ideal. The geometry of the electrodes has a great effect on the magnitude of this error. The Delmhorst gypsum block used in the 227 probe has two concentric cylindrical electrodes. The center electrode is used for excitation; because it is encircled by the ground electrode, the path for a ground loop through the soil is greatly reduced. Moisture blocks which consist of two parallel plate electrodes are particularly susceptible to ground loop problems. Similar considerations apply to the geometry of the electrodes in water conductivity sensors.

The ground electrode of the conductivity or soil moisture probe and the CR7 earth ground form a galvanic cell, with the water/soil solution acting as the electrolyte. If current was allowed to flow, the resulting oxidation or reduction would soon damage the electrode, just as if DC excitation was used to make the measurement. Campbell Scientific probes are built with series capacitors in the leads to block this DC current. In addition to preventing sensor deterioration, the capacitors block any DC component from affecting the measurement.

## 13.7 PULSE COUNT MEASUREMENTS

Many pulse output type sensors (e.g., anemometers and flow-meters) are calibrated in terms of frequency (counts/second). For these measurements the accuracy is related directly to the accuracy of the time interval over which the pulses are accumulated. Variation in the pulse sampling interval DOES NOT effect those cases where the pulse measurement is independent of time, i.e., where the total pulse count is of interest instead of frequency.

The Pulse Count Instruction (#3) causes the pulse channel counters to be read and reset to zero every time the program table is executed. The CR7 operating system checks the program tables every 0.1 seconds to determine whether or not a table should be executed. If any of the 3 tables requires execution and contains a Pulse Count Instruction, ALL active pulse channels in the specified I/O Module are read and reset to zero before the table is executed. Reading the pulse channels immediately instead of when the Pulse Count Instruction is encountered in the program sequence eliminates variation in the pulse sampling interval caused by variable program execution times.

If the table execution is delayed, for example by lengthy output processing, the pulse channel counters are not read until the next execution interval occurs. Whenever a table execution overrun occurs, a decimal point appears below the colon that separates the ID and Data field of the display. Frequency data taken during an overrun is invalid because the pulse sampling interval is extended. The configuration code (Parameter 4) entered in the Pulse Counting Instruction allows measurements taken during overruns to be discarded and replaced by the count obtained during the previous, correct sampling interval.

## PULSE COUNT MEASUREMENTS USING MULTIPLE I/O MODULES

Pulse channels contained in multiple I/O Modules are read and reset based on the program table priority. Within a specific program table, the channels are reset according to the priority of the I/O Module, i.e., Module 1 first and Module 4 last. For example, if Table 1 and 2 have simultaneous execution intervals, all the active pulse channels in the I/O Modules referenced by Pulse Count Instructions in Table 1 are read and reset first, followed by the I/O Modules referenced in Table 2. Within Table 1 however, pulse channels contained in I/O Module 1 are read first, followed by I/O Module 2, etc. All Pulse Counting Instructions referencing the same I/O Module should be contained in the same program table.

Approximately 0.6ms is required to read and reset the pulse channels contained in one I/O Module. Thus different I/O Modules having the same pulse sample intervals are offset in real time by 0.6ms per I/O Module. The pulse sample interval associated with an I/O Module is constant however, as long as the programming rules listed below are observed.

1. If possible place all Pulse Count Instructions in the same program table.

2. If more than one program table contains Pulse Count Instructions, give Table 1 the fastest execution interval and make the other execution intervals even multiples of Table 1's interval.

If these rules are violated a 0.6ms variation could exist in an I/O module's sample interval. For example, suppose Table 1 resets the pulse counters in I/O Module 1 every 3 seconds and Table 2 resets the pulse counters I/O Module 2 every second. When Table 1 is executed the pulse counters in I/O Module 2 will be reset 0.6ms later than when only Table 2 is executed. The frequencies must exceed 1.67 kHz before the measurement is affected by a 0.6ms variation in the pulse sampling interval.

# SECTION 14. INSTALLATION

## 14.1 ENVIRONMENTAL ENCLOSURE, CONNECTORS AND JUNCTION BOXES

The standard CR7 is equipped with the Model ENC-7F Fiberglass Case. During the manufacturing of the case, the base and lid are formed together to insure a perfectly matched fit. A six digit serial number is stamped into the extruded aluminum rims on both the base and lid. When more than one CR7 is owned, care should be taken to avoid a mismatch which could prevent a gas-tight seal.

In addition to the AC power input connector and pressure relief valve, the STANDARD ENC-7F enclosure has two 1.040" dia. access ports, located to the left side of the I/O Module (Figure OV.1-1). These access ports are provided to allow the entry of sensor leads, power cables, etc. Unless otherwise specified at time of ordering, these two access ports are fitted with 0.75" dia. conduit bushings, 0-Ring seals and removable neoprene plugs. The neoprene plugs can be used to form a reasonable seal by drilling holes in them for accommodating the passage of sensor leads to the I/O Module.

> **NOTE:** Users wishing to punch additional access ports should be aware that the 1.040" dia. punch is a special size. The greenlee punch normally used to make holes for 0.75" conduit bushings is 1.07" dia. This diameter is too large for mounting the 19 pin connector option.

Access ports fitted with drilled neoprene plugs may not be sufficient for certain data acquisition situations. Alternatively, the access ports can have connectors installed.

Factors which may influence selection of the appropriate connector type are (1) the seal required, (2) the permanent or temporary nature of the CR7 installation, and (3) cost considerations.

## 14.1.1 ACCESS PORTS FITTED WITH ELBOWS

Standard 0.75" dia. male conduit elbows may be screwed directly into the conduit bushings provided with the standard environmental enclosure. Elbows allow entry of individual sensor leads and power cables while preventing precipitation from entering the enclosure directly. Silicon sealer can be used to seal off the space between the elbow wall and leads. Conduit elbows are inexpensive and well suited for field applications where sensor arrays are frequently changed and a gas-tight seal is NOT required.

> **NOTE:** Larger conduit elbows (1.5" dia.) are required for allowing a 9 pin D-type connector with ribbon cable attached to pass through and access port. In order to accommodate the larger elbows, access ports must be enlarged and fitted with the appropriate conduit bushings and O-Ring seals.

## 14.1.2 SOCKET CONNECTORS

Access ports may be reconfigured as connector ports by replacing the conduit bushings with sealed (shell size 14) 19-pin circular socket connectors. Individual leads are soldered to the backside of the mounted connector and routed to appropriate I/O terminals. Socket connectors are recommended for applications where a gas-tight seal IS required.

The connector utilizes a compression screw-actuated sealing gland. This connector is recommended when multi-connector sensor cable is used between the CR7 and a junction box.

## 14.1.3 JUNCTION BOXES

Individual sensor leads (and multiconductor cables) may be routed directly from the sensor locations to the CR7 or routed to a junction box and then to the CR7. Advantages of using a junction box are two-fold: it provides a convenient method for changing sensors/sensor leads quickly and can provide additional protection against instrumentation damage as a result of lightning induced high voltages. Junction boxes generally do not require a gas-

tight seal but do require protection from thermal gradients when used for thermocouple lead wires (Section 13.4).

## 14.2 SYSTEM POWER REQUIREMENTS AND OPTIONS

The standard CR7 is equipped with sealed lead acid battery packs and charging circuitry for accommodating (1) 120/240 VAC line power, (2) solar panels, (3) vehicular 12V power sources, and (4) external 12V batteries. When fully charged, the internal batteries of the CR7 are capable of providing 2.5 Amp-Hours of service or about 5 days of operation in a typical application where the CR7 is active 10% of the time.

### 14.2.1 POWER SUPPLY AND AC CHARGER

Power for charging the internal batteries from an external AC source is provided to the CR7's 12V charging regulator circuit board through the power input connector located on the outside of the case to the right of the Control Module.

The LABORATORY ENCLOSURE has a 4-position voltage select (100, 120, 220 or 240) insert chip and 0.1 amp slow blow fuse located next to the power input connector (on the outside of the enclosure). CR7s leave the Factory with the voltage insert chip set to 120 VAC. If the intended line voltage is not 120V, reposition the chip to correspond as closely as possible with the AC line voltage available (the number showing when the chip is inserted specifies the voltage level).

ENVIRONMENTAL ENCLOSURES are equipped with a weather-tight bayonet mount 3-pin circular power connector. Access to the 2-position (115 and 230) voltage select switch and 0.1 amp slow blow fuse is by lifting up the Control Module Panel. The switch is set to 115 VAC at the factory.

A temperature compensated 12V CHARGING REGULATOR CIRCUIT BOARD beneath the Control Module regulates the charging voltage supplied to the lead acid batteries and the voltage to the CR7 operating system. DC Power sources are connected to the terminal block on the charging regulator board. The terminals labeled "EXT BATT" are for connecting a 12V power source. The terminals labeled "SOLAR" are for connecting a solar panel or DC source with sufficient voltage (15-25 VDC) to charge the internal lead acid batteries.

The LED auxiliary power light located on the face of the Control Module, is activated by the charging regulator when solar or AC charging circuitry are connected to the CR7. The LED operates with the ON/OFF switch in either position but is NOT designed to operate when an external 12V battery only is connected to the CR7.

Power to the CR7 operating system is controlled by the position of the ON/OFF toggle switch located on the face of the Control Module.

The sealed lead acid battery packs, located below the Control Module, are rated at 6V each and are connected in series to provide 12 VDC.

### TABLE 11.2-1. CR7 Battery and Charging Circuitry Specifications

| | |
|---|---|
| **BATTERY** Type | Gates #810-0011X |
| Float Life @ 25 °C | 8 yrs minimum |
| Amp Hour Rating | 2.5 amp-hour |
| Open Circuit Voltage @ Full Charge | 12.95 typ. with charging circuitry deactivated, 14.1 @ 25 °C when activated |
| Open Circuit Voltage @ Full (SAFE) Discharge | 11.76 VDC minimum |
| Shelf Life, Full Charge | Check twice yearly |
| Charging Time from Full Discharge (AC Source) | 40 hrs for full charge, 20 hrs for 95% full charge |
| **Charging Circuit** | Float charge with temperature compensated voltage regulation |
| AC Line Filter (LAB.) | 6 amp max, 48-440 Hz, 250 VAC max. |
| AC Line Filter (ENVIR.) | 1 amp max, 50-60 Hz, 250 VAC max. |
| Power Supply Transformer | 6 watt output |

Battery voltage should NOT be allowed to drop below 11.76V before recharging; otherwise, permanent damage to the lead acid cells may occur. CSIs warranty does NOT cover battery or cell damage resulting from deep discharge.

Avoid deep discharge states by periodically monitoring voltage level of the CR7s internal batteries, using Input/Output Instruction 10. Incorporate the battery voltage measurement to the data acquisition program to avoid deep discharge of the CR7 internal batteries.

All external charging devices must be disconnected from the CR7 in order to measure the true voltage level of the internal batteries.

The internal lead acid batteries of the CR7 will continue to discharge with the CR7 turned on but not scanning or processing data. This quiescent current drain will vary depending on the number of I/O Modules, Excitation and Pulse Counter cards contained in the CR7, and the number and type of external devices powered by the CR7's Power Supply. Users can approximate the quiescent current drain of their specific CR7 System from the information provided in Table 14.2-2.

### TABLE 14.2-2. Calculating Quiescent Current Drain

| Module/Card | Current Drain |
|---|---|
| Control Module | 0.4 mA |
| I/O Module | 2.5 mA |
| Excitation card | 2.0 mA |
| Pulse Counter card | 0.8 mA |
| Analog Input card | 0.7 mA |

As an example, the quiescent current drain of a CR7 System containing a Control Module, an I/O Module, 1 Excitation card, 2 Pulse Counter cards and 4 Analog Input cards is about 9.3mA. At this rate of quiescent current drain, fully charged internal batteries (2.5 AH) are depleted to a full SAFE discharge level (11.76V) after 268 hours (about 11 days). When the CR7 is active, it draws approximately 100mA so the actual current drain is a function of the program being executed.

### 14.2.2 SOLAR PANELS

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

### TABLE 14.2-1. Solar Panel Specifications

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

NOTE: Specifications assume 1 kW/m$^2$ illumination at a panel cell temperature of 25°C. Individual panel performance may vary as much as 10%.

When selecting a solar panel, a rule-of-thumb is that on a stormy overcast day the panel should provide enough charge to meet the system current drain (assume 10% of average annual global radiation, kW/m$^2$). Specific site information, if available, could strongly influence the solar panel selection. For example, local effects such as mountain shadows, fog from valley inversion, snow, ice, leaves, birds, etc. shading the panel should be considered.

Guidelines are available from the Solarex Corporation for solar panel selection called "DESIGN AIDS FOR SMALL PV POWER SYSTEMS". It provides a method for calculating solar panel size based on general site location and system power requirements. If you need help in determining your system power requirements, contact Campbell Scientific's Marketing Department.

The solar panel is connected to the CR7 by attaching the 2 lead wires of the power cable to the terminal block located on the charging regulator circuit board (Figure 14.2-1). The free end of the solar panel power cable is equipped with a 12V power plug for use with the 21XL Micrologger. Cut this plug off with side cutters and remove about 1.5" of the cable's outer insulation. Remove about 0.5" of insulation from the exposed black and clear leads. The BLACK lead is GROUND and the CLEAR lead is positive (+).

> Regulated solar panels (e.g., MSX18R) limit voltage to approximately 14V. The CR7 Solar Panel input requires 15-25 VDC to charge.

## 14.2.3 EXTERNAL BATTERY CONNECTION

An external battery may be used to supplement the internal lead acid batteries of the CR7. The ground and +12 leads are connected to the appropriate "EXT BATT" terminals.

The recommended procedure for connecting the CR7 to an external battery is to make all required ground lead connections before connecting the battery. Accidental shorting can be prevented by insulating one of the power leads until cable routing is completed. When disconnecting a battery, remove the positive lead before disconnecting the ground lead.

Power for operating the CR7 may also be provided by connecting the power leads from an external 12V battery to the 2 terminals located on the left side of the I/O Module CPU card. This is a quick and convenient method of connecting an alternate power supply to the CR7 if it is necessary to disconnect or replace the CR7s internal lead acid batteries. However, the primary purpose of the terminals on the I/O Module is to provide access to 12V for powering external devices such as sensors.

> Reverse polarity protection is NOT provided on these terminals and CR7 damage will occur if external power is connected with reverse polarity.

If an external power supply is connected to the I/O Module terminals the CR7 remains powered-up even when the power switch is off. The external supply must be removed to power the CR7 down.

CSI recommends using 22 AWG lead wires or larger when connecting an external battery to the CR7.

## 14.2.4 CONNECTING TO VEHICLE POWER SUPPLY

When the starting motor of a motor vehicle with a 12 Volt electrical system is engaged, the voltage drops considerably below the nominal 12 volts. If the CR7 were connected directly to the vehicle power supply, the CR7 batteries would be pulled down as well, causing the CR7 to "bomb" any time the vehicle was started. To avoid this problem, a diode (i.e., 1N4001) and a 3 ohm 5 watt resistor must be placed in series in the positive lead going to the EXTERNAL BATTERY terminal. The diode allows the vehicle to power the CR7 without the CR7 attempting to power the vehicle (diode installed with bar end toward datalogger). To reduce the potential for ground reference errors in measurements, the ground lead should be 16 AWG or larger.
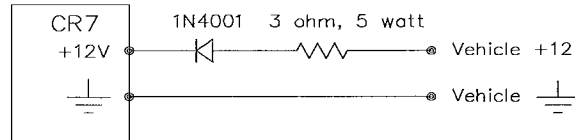


**FIGURE 14.2-1. Connecting Vehicle Power Supply to CR7**

## 14.2.5 SAFETY PRECAUTIONS

There are inherent hazards associated with the use of sealed lead acid batteries. Under normal operation, lead acid batteries generate a small amount of hydrogen gas. This gaseous by-product is generally insignificant because the hydrogen dissipates naturally before build up to an explosive level (4%) occurs. However, if the batteries are shorted or overcharging takes place, hydrogen gas may be generated at a rate sufficient to create a hazard. Because the potential for excessive hydrogen build up does exist, CSI makes the following recommendations:

1.  A CR7 equipped with standard lead acid batteries should NEVER be used in environments requiring INTRINSICALLY SAFE EQUIPMENT.

2.  When attaching an external battery to the CR7, insulate the bare lead ends to protect against accidental shorting while routing the power leads.

3. When the CR7 is to be located in a gas-tight enclosure or used in a gas-tight mode with the standard ENVIRONMENTALLY SEALED FIBERGLASS CASE, the internal lead acid batteries SHOULD BE REMOVED and an external battery substituted.

## 14.3 HUMIDITY EFFECTS AND CONTROL

The CR7 system is designed to operate reliably under environmental conditions where the relative humidity inside its enclosure does not exceed 90% (noncondensing). Situations where the humidity tolerances are exceeded may result in damage to IC chips, microprocessor failure and/or measurement inaccuracies due to condensation on the various PC board runners. Effective humidity control is the responsibility of the user and is particularly important in operational environments where the CR7 is exposed to salty air.

Several precautionary methods are available for protecting the CR7 against excessive humidity and subsequent component damage. Selection of the appropriate method or combination of methods will depend on the environmental characteristics prevailing at a specific data acquisition site. Humidity control methods include:

1. the use of desiccant

2. sealing the CR7 enclosure gas-tight

3. the nitrogen purging technique

### 14.3.1 DESICCANT

As a minimal precaution, the packets of HUMI-SORB desiccant shipped with the CR7 should be placed inside the Control Module. These packets should be routinely removed from the CR7 and reactivated by warming them in an oven for about 16 hours at a temperature of 120 °C (250 °F). The recommended time between reactivations varies from one location to another. Obviously, the desiccant requires more frequent attention in environments where the relative humidity is high.

### 14.3.2 GAS-TIGHT ENVIRONMENT

Another method for controlling humidity involves the proper selection of connectors for use with the sealed fiberglass enclosure such that the CR7 operates in a gas-tight environment. Various connector options and associated details are described in Section 12. Desiccant is still required with this method.

### 14.3.3 NITROGEN PURGING

Several CSI customers have had success in preventing humidity-related equipment malfunctions in harsh environments by allowing nitrogen gas to slowly bleed into the datalogger enclosure. The sensor leads, power cables, etc., are routed to the terminal blocks of the datalogger through simple, inexpensive conduit elbows which are left unplugged. A nitrogen bottle is then left at the field site with its regulator valve slightly open so that nitrogen is allowed to escape slowly through a rubber tube which is routed along with the sensor leads through the conduit elbows into the CR7 enclosure. The tube vent should be positioned underneath the Control Module.

Equipment required for this method of humidity control generally can be obtained from any local welding supply shop and includes a nitrogen bottle, regulator with tube adapter (content gauge, optional), hose clamp and a suitable length of small diameter rubber tubing. Nitrogen bottles are available in various sizes and capacities. The size of the nitrogen bottle used depends on the transport facilities available to and from the field site and on the time interval between site visitations. Where practical, larger nitrogen bottles should be used to reduce cost and refilling frequency.

## 14.4 RECOMMENDED GROUNDING PRACTICES

### 14.4.1 PROTECTION FROM LIGHTNING

Primary lightning strikes are those where the lightning hits the datalogger or sensors. Secondary strikes occur when the lightning strikes somewhere near the lead in wires and induces a voltage in the wires. All input and output connections in the I/O Module are protected using spark gaps that are rated to 10,000 amps. The spark gaps are connected directly to the heavy copper grounding bar on each input card with no more than 2 inches of

20 AWG wire. This transient protection is useless if there is not a good connection between the CR7 and earth ground.

All dataloggers in use in the field should be grounded. A 12 AWG or larger wire should be run from the grounding terminal on the left side of the I/O Module (Figure OV1-1) to a grounding rod driven far enough into the soil to provide a good earth ground.

A modem/phone line connection to the CR7 provides another pathway for transients to enter and damage the datalogger. The DC112 Modem has spark gaps on the phone lines. A ground wire should be run between the ground terminal on the modem and earth ground.

## 14.4.2 EFFECT ON MEASUREMENTS: COMMON MODE RANGE

The effects that a difference in ground potential between a sensor or signal conditioner and the CR7 can have on a measurement were discussed in Sections 7.2 and 13.2. These sections stress that differential voltage measurement gets rid of offset caused by a difference in ground potential. However, in order to make a differential measurement, the inputs must be within the CR7 common mode range of ±5V.

The common mode range is the voltage range, relative to CR7 ground, within which both inputs of a differential measurement must lie, in order for the differential measurement to be made. For example, if the high side of a differential input is at 4V and the low side is at 3V relative to CR7 ground, there is no problem, a measurement made on the ±1.5V range would indicate a signal of 1V. However, if the high input is at 5.8V and the low input is at 4.8V, the measurement cannot be made because the high input is outside of the CR7 common mode range.

Problems with exceeding common mode range may be encountered when the CR7 is used to read the output of external signal conditioning circuitry if a good ground connection does not exist between the external circuitry and the CR7. When operating where AC power is available, it is not always safe to assume that a good ground connection exists through the AC wiring. If a CR7 is used to measure the output from a laboratory instrument (both plugged into

AC power and referencing ground to outlet ground), it is best to run a ground wire between the CR7 and the external circuitry. Even with this ground connection, the ground potential of the two instruments may not be at exactly the same level, which is why a differential measurement is desired.

## 14.5 USE OF DIGITAL CONTROL PORTS FOR SWITCHING RELAYS

Each of the eight digital control output ports can be set high or low by the PORT SET command (Instruction 20). Because of current supply limitations, a digital control output port normally is used to operate an external relay driver. These relays may be used for activating an external power source to run a fan motor or for altering an external circuit as a means of multiplexing signal lines, etc.

Figure 14.5-1 is a schematic representation of a typical external coil driven relay configuration which may be used in conjunction with one of the CR7s digital control output ports. The example shows a DC fan motor (typical of a ventilated psychrometer) and 6V battery in the circuit, but the configuration may be used for other purposes. This particular configuration has a coil current limitation of 75mA because of the NPN Medium Power Transistors used (Part No. 2N2222). CSI's Model A21REL-12 and A6 REL12 are 12 VDC Relay Controllers available for use with the CR7 system.
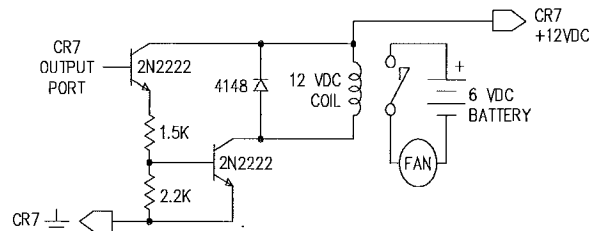


**FIGURE 14.5-1. Typical Connection for Activating/Powering External Devices, Using a Digital Control Output Port and Relay Driver.**

# 15. I/O CARD ADDRESSING AND MULTIPLE I/O MODULES

## 15.1 I/O CARD IDENTIFICATION NUMBER DECODING

Each I/O card must be assigned a unique card identification number and have jumpers set for that number. The numbers allow the cards to decode signals addressed to them by the I/O Module. CR7s leave the factory with card numbers assigned. These numbers may be reassigned by the user when a CR7 needs to be expanded with additional cards or reconfigured for a particular application.

### 15.1.1 ASSIGNING CARD IDENTIFICATION NUMBERS

Cards fall into one of two Categories. Category 1 includes all Excitation and Pulse Counter cards; Category 2 includes Analog Input cards only. NUMBERS MUST NOT BE DUPLICATED WITHIN A CATEGORY, but a number assigned in one category may be duplicated in the other category. While the identification numbers do not need to follow physical order of the cards, it simplifies wiring sensors if the card ID numbers match the physical order.

When assigning card numbers the user can avoid confusion by following 4 steps:

1. Categorize the cards.

2. Assign a valid number to each card.

3. Set the jumpers on the cards according to the numbers assigned.

4. Insert the cards into the Control Module, making certain that the correct number tag is secured with each card (Figure 10.1-1).

Tables 15.1-2 and 15.1-3 list card numbers and the corresponding jumper placement. The bus structure allows one CR7 Control Module to address up to 16 Excitation cards, 16 Pulse Counter cards and 32 Analog Input cards. One Control Module can control up to 4 I/O Modules.

### 15.1.2 PLACING JUMPERS IN THEIR PROPER LOCATIONS

Jumpers are blue rectangular plastic and metal sleeves, approximately 0.2" in length, with two holes which are slipped over pins on the circuit board.

Category 1 cards have 2 jumpers and Category 2 cards have 3 jumpers. Jumper positions are labeled on the card. Figure 15.1-1 shows the location of the jumpers on the Excitation, Pulse Counter and Analog Input cards. Jumper settings are listed in Tables 15.1-2 and 15.1-3.

**TABLE 15.1-1. Jumper Locations and Labels**

| Card Type | Cat. | Jmp/ Card | Card Loc. | No. Pins |
|-----------|------|-----------|-----------|----------|
| Excitation | 1 | 2 | R2 | 6 |
|  |  |  | R6 | 6 |
| Pulse Count | 1 | 2 | R36 | 6 |
|  |  |  | T32 | 6 |
| Analog Input | 2 | 3 | R3 | 3 |
|  |  |  | R4 | 6 |
|  |  |  | R8 | 6 |
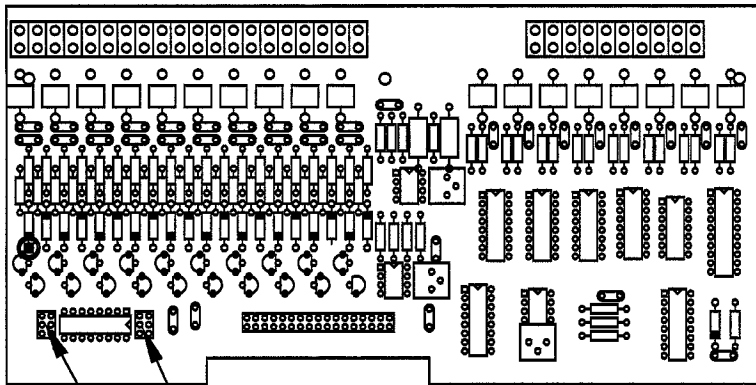
MODEL 725
EXCITATION CARD
(CATEGORY 1)

— JUMPER POSITION R6
— JUMPER POSITION R2
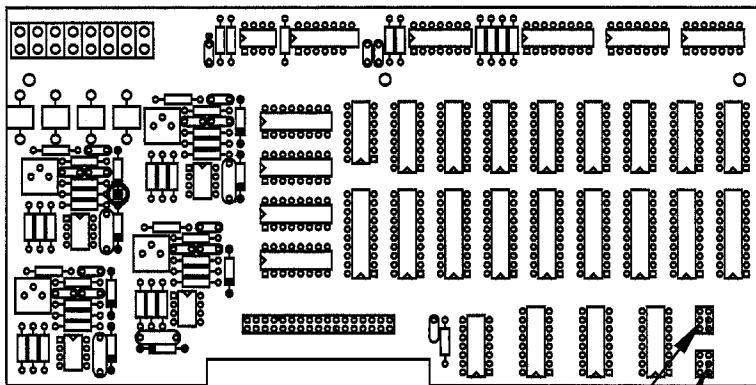
MODEL 724
PULSE COUNTER CARD
(CATEGORY 1)

JUMPER POSITION R32 —
JUMPER POSITION T32 —
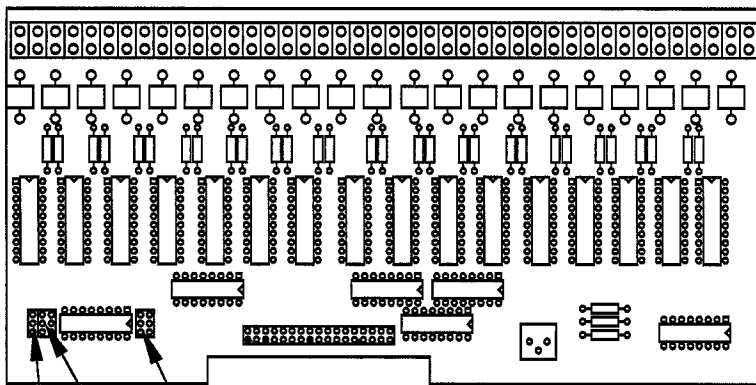
MODEL 723, 723−T OR 726
ANALOG INPUT CARD
(CATEGORY 2)

— JUMPER POSITION R8
— JUMPER POSITION R4
— JUMPER POSITION R3

FIGURE 15.1-1.  Position of Decoding Jumpers on Excitation, Pulse Counter and Analog Input Cards.

| VALID EXCIT. CARD NO. | JUMPER R2 | POSITION/PLACEMENT R6 | |
|---|---|---|---|
| 1 | A | A | |
| 2 | A | B | |
| 3 | A | C | |
| 4 | A | D | |
| 5 | B | A | |
| 6 | B | B | |
| 7 | B | C | |
| 8 | B | D | |
| 9 | C | A | |
| 10 | C | B | |
| 11 | C | C | |
| 12 | C | D | |
| 13 | D | A | |
| 14 | D | B | |
| 15 | D | C | |
| 16 | D | D | |

| VALID PULSE CARD NO. | JUMPER R32 | POSITION/PLACEMENT T32 | |
|---|---|---|---|
| 1 | D | D | |
| 2 | D | C | |
| 3 | D | B | |
| 4 | D | A | |
| 5 | C | D | |
| 6 | C | C | |
| 7 | C | B | |
| 8 | C | A | |
| 9 | B | D | |
| 10 | B | C | |
| 11 | B | B | |
| 12 | B | A | |
| 13 | A | D | |
| 14 | A | C | |
| 15 | A | B | |
| 16 | A | A | |

**TABLE 15.1-2.  Jumper Settings for Excitation and Pulse Counter Cards**

| VALID INPUT | ANALOG CARD NO. | JUMPER R3 | POSITION/PLACEMENT R4 | R8 |
|---|---|---|---|---|
| 1 | | I | E | A |
| 2 | | I | E | B |
| 3 | | I | E | C |
| 4 | | I | E | D |
| 5 | | I | F | A |
| 6 | | I | F | B |
| 7 | | I | F | C |
| 8 | | I | F | D |
| 9 | | I | G | A |
| 10 | | I | G | B |
| 11 | | I | G | C |
| 12 | | I | G | D |
| 13 | | I | H | A |
| 14 | | I | H | B |
| 15 | | I | H | C |
| 16 | | I | H | D |
| 17 | | J | E | A |
| 18 | | J | E | B |
| 19 | | J | E | C |
| 20 | | J | E | D |
| 21 | | J | F | A |
| 22 | | J | F | B |
| 23 | | J | F | C |
| 24 | | J | F | D |
| 25 | | J | G | A |
| 26 | | J | G | B |
| 27 | | J | G | C |
| 28 | | J | G | D |
| 29 | | J | H | A |
| 30 | | J | H | B |
| 31 | | J | H | C |
| 32 | | J | H | D |

EXAMPLE

EXAMPLE

**TABLE 15.1-3. Jumper Settings for Analog Input Cards.**

## 15.2 USE OF MULTIPLE I/O MODULES

Up to four I/O Modules can be connected to one control module. Additional I/O Modules may be remotely located from the Control Module.

Two enclosures are available for the 720 I/O Module; the standard ENC-7F Fiberglass Environmental Enclosure or the ENC-7L Aluminum Laboratory Enclosure. Options also are available for 19" rack mounting (M197) or mounting in a NEMA type enclosure (uses M720 Back Mount Brackets).

The 720XL I/O Module mounts directly in a 19" rack or may be mounted in a NEMA type enclosure with the Model M720 Back Mount Brackets.

**TABLE 15.2-1. SC94 Pin Description.**

| Pin ID | Control Module | I/O Module |
|---|---|---|
| A | Transmit data | Receive data |
| B | Transmit data return | Receive data return |
| C | Receive data return | Transmit data return |
| D | Receive data | Transmit data |

> **NOTE:** Interconnect cable lengths in excess of a 1000 ft. limit the maximum baud rate at which data may be transmitted between the Control Module and the Remote I/O Module. Cable length and recommended baud rates are listed in Table 15.2-3.

## TABLE 15.2-2. Hardware Components in SC94

| Component | Description |
|---|---|
| Interconnect Cable (1 ea.) | Length of the 4-wire cable is made to order, circular connectors attached at both ends. |
| Mating Circular Connector (2 ea.) | One connector affixed to Control Module enclosure; other connector affixed to remote I/O Module enclosure; each joined to a SC94 circuit card. |
| SC94 Circuit Card (2 ea.) | One card fastens to the remote I/O Module enclosure, the other to the Control Module enclosure; both have circular socket connectors. |
| 10-Conductor Ribbon Cable | Ribbon cable connects SC94 circuit card to Control Module's Serial Interface cord or to remote I/O Module's Controller card. |
| Grounding Wire (2 ea.) | 12 AWG; wires are routed between terminals located on SC94 cards and 1 of 3 available terminals on the grounding bar on left side of I/O Module framework; provides transient protection at both ends of Interconnect cable. |

Mating connectors and the associated circuitry are installed at the factory when the CR7 System and multiple I/O Modules are purchased together. An SC94 Four Wire Current Loop Interconnect assembly is required for each remote I/O Module operating in a CR7 System. Table 15.2-2 describes the SC94 hardware.

Remote I/O Modules require their own power supply. When remote I/O Modules are ordered with either the Model ENC-7F or the Model ENC-7L enclosure option, the space normally taken up by the Control Module can be used for housing the PS12-LA Power Supply. The PS12-LA 12V charging regulator and battery may also be mounted with a remote I/O Module inside a NEMA type enclosure. Multiple I/O Modules can be powered by a single Power Supply if the distance is not too great.

Once all grounding leads are in place the power supply leads are usually connected to the terminal block located on the top left of the I/O Module's Controller card. Alternatively, the two power leads can be routed to the terminals, marked +12V and, ground positioned on the SC94 card.

**NOTE:** Whenever power is connected to the I/O Module, it draws current even when the power switch on the Control Module is in the OFF position.

### 15.2.1 PROGRAMMING CONSIDERATIONS

The mating connector(s) affixed to CR7 Systems configured to operate with multiple I/O Modules are labeled with a unique I/O Module identification number (e.g., 1, 2, 3, 4) before they leave the Factory. Instruction 23 is used to address subsequent programming instructions to a particular I/O Module.

At the start of each Program Table, the Control Module assumes all instructions are meant for I/O Module #1, the I/O Module housed with the Control Module. Instruction 23 must be used to address instructions to I/O Modules other than #1. Once Instruction 23 is executed, subsequent instructions are addressed to the specified I/O Module until Instruction 23 is executed again or the I/O Module again defaults to #1 at the start of a Program Table.

A typical programming example for a CR7 System containing two I/O Modules is given in the following Program Table. A separate Power Supply powers the remote I/O Module. The objective of the programming example is to conduct a preliminary system check-out by measuring the battery voltage of the remote Power Supply and of the power supply powering the Control Module and I/O Module #1.

**PROGRAM**
**Execution Interval 1 Second**

| Inst. Loc. | Param. No. | Entry | Description |
|---|---|---|---|
| 1 | P | 10 | Measure Control Module voltage |
|  | 1 | 1 | Store result in Input Loc. 1 |
| 2 | P | 23 | Select I/O Module |
|  | 1 | 2 | I/O Module 2 |
| 3 | P | 10 | Measure remote battery voltage |
|  | 1 | 2 | Store result in Input Loc. 2 |

Any subsequent instructions in the example would address I/O Module #2 unless Instruction 23 were executed again specifying I/O Module #1.

## 15.2.2 SETTING BAUD RATE BETWEEN I/O AND CONTROL MODULES

The baud rate that data are transmitted between the Control and I/O Modules is pre-set to 38.4 k baud at the Factory. This baud rate can be used for most datalogging applications provided the distance separating the Control Module and the I/O Module(s) does not exceed 1,000 ft. Distances in excess of 1,000 ft.

require a slower baud rate setting to compensate for connector capacitance. Slower baud rates may limit table Execution Intervals or throughput rate. Table 15.2-3 lists recommended baud rates for communicating with I/O Modules at varying distance from the Control Module.

**TABLE 15.2-3. Recommended Baud Rates for Remote I/O Modules**

| Distance in Feet | Maximum Recommended Baud Rate |
|---|---|
| 0 to 1000 | 38.4 k baud |
| 1000 to 2000 | 19.2 k baud |

The baud rate is determined by jumper placement at location S5 of the Control Module's Serial Interface card (Figure 15.2-1) and position G7 of the Model 720 I/O Module Controller card (Figure 15.2-2).

Position S5 has eight pairs of pins. Starting with the pair nearest to the adjacent board edge and moving left, baud rates of 1200, 2400, 4800, 9600, 19200, 38400, 76300 or 153600 baud can be selected. The 76.3 and 153.6 k baud settings are not recommended.

Figure 15.2-2 shows the I/O Module Controller card. Position G7 has 14 vertically aligned pin pairs with the 8 uppermost pairs determining baud rate. A baud rate of 153.6 k baud is obtained by jumpering the uppermost pair. The baud rates from the top pair working down are: 153600, 76300, 38400, 19200, 9600, 4800, 2400, and 1200 baud.

Baud rate settings for the Control Module and the I/O Module(s) must be the same.

POSITION S5
(I/O MODULES)
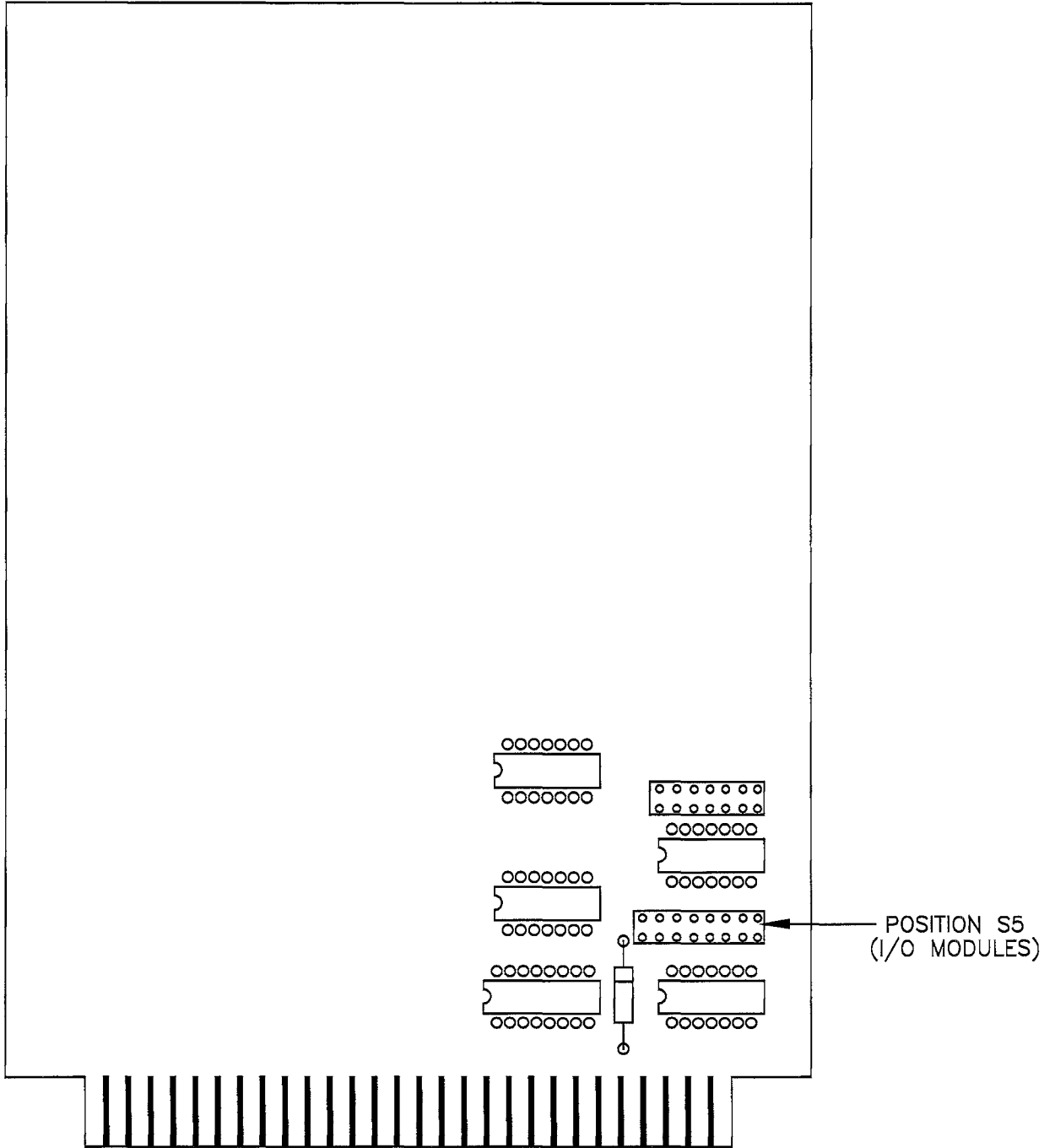
Figure 15.2-1. Location of Jumper Controlling Baud Rate to I/O Modules
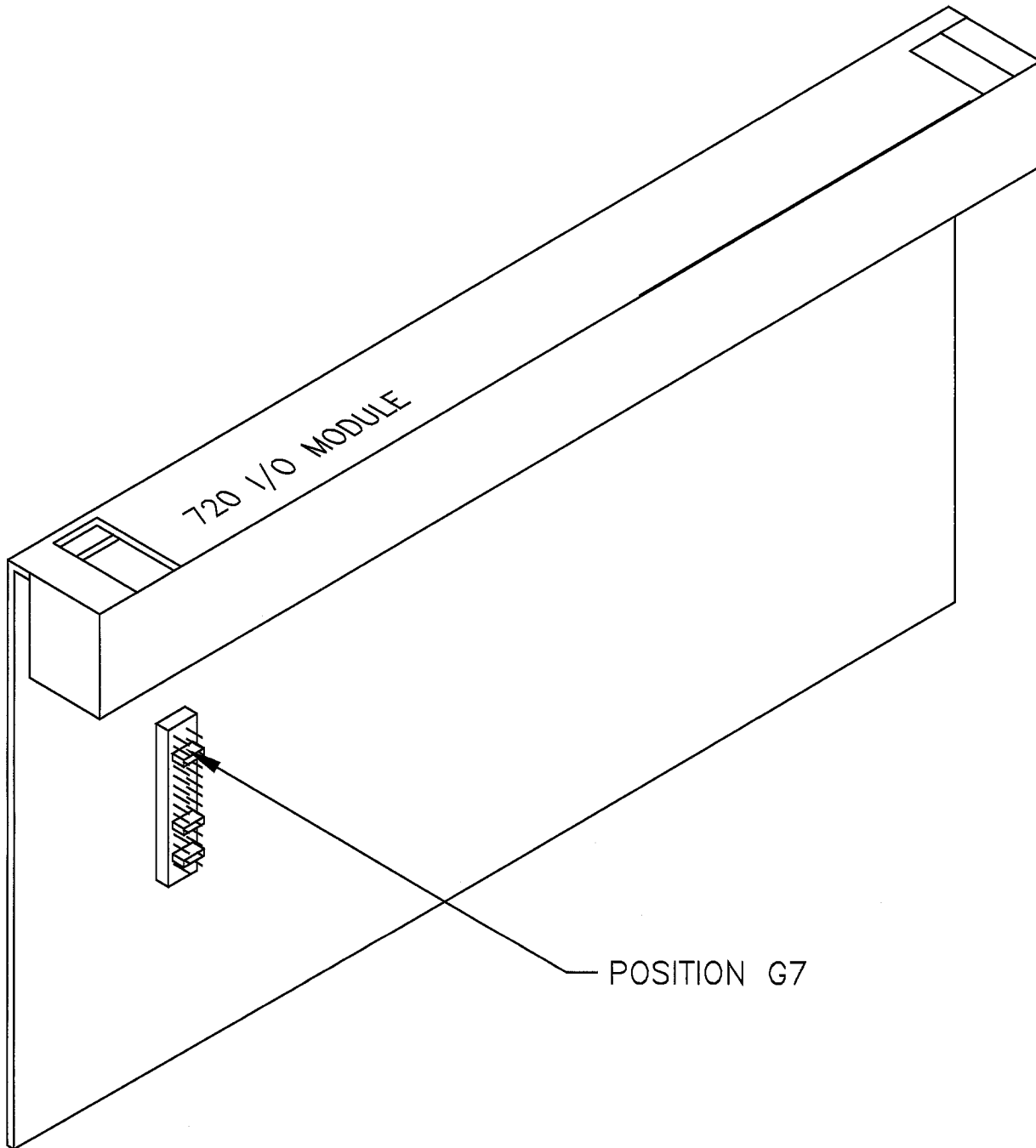
Figure 15.2-2. Location of I/O Module Jumper Controlling Baud Rate between the I/O Module and the Control Module

# APPENDIX A. GLOSSARY

**ASCII:** Abbreviation for American Standard Code for Information Interchange (pronounced "askee"). A specific binary code of 128 characters represented by 7 bit binary numbers.

**BAUD RATE:** The speed of transmission of information across a serial interface, expressed in units of bits per second. For example, 9600 baud refers to bits being transmitted (or received) from one piece of equipment to another at a rate of 9600 bits per second. Thus, a 7 bit ASCII character plus parity bit plus 1 stop bit (total 9 bits) would be transmitted in 9/9600 sec = .94 ms or about 1000 characters/sec. When communicating via a serial interface, the baud rate settings of two pieces of equipment must match each other.

**DATA POINT:** A data value which is sent to Final Storage as the result of an Output Instruction. Strings of data points output at the same time make up Output Arrays.

**EXECUTION INTERVAL:** The time interval between initiating each execution of a given Program Table. If the Execution Interval is evenly divisible into 24 hours (86,400 seconds), the Execution Interval will be synchronized with 24-hour time so that the table is executed at midnight and every execution interval thereafter. The table will be executed for the first time at the first occurrence of the Execution Interval after compilation. If the Execution Interval does not divide evenly into 24 hours, execution will start on the first even second after compilation.

**EXECUTION TIME:** The time that it actually takes the CR7 to execute an instruction or group of instructions. If the execution time of a Program Table exceeds the table's Execution Interval, the Program Table will be executed less frequently than programmed.

**FINAL STORAGE:** That portion of memory allocated for storing Output Arrays. Final Storage may be viewed as a ring memory, with the newest data being written over the oldest. Data in Final Storage may be displayed using the *7 Mode or sent to various peripherals.

**HIGH RESOLUTION:** A high resolution data value has 5 significant digits and may range in magnitude from ±.00001 to ±99999. A high resolution data value requires 2 Final Storage locations (4 bytes). All Input and Intermediate Storage locations are high resolution. Output to Final Storage defaults to low resolution: high resolution output must be specified by Instruction 78.

**INDEXED INPUT LOCATION:** An Input location entered as an instruction parameter may be indexed by keying "C" before entering by keying "A", two dashes (--) will appear at the right of the display. Within a loop (Instruction 87), this will cause the location to be incremented each pass through the loop. Indexing is also used with Instructions 13, 14 and 75 to cause an Input location, which normally remains constant, to be incremented with each repetition.

**INPUT/OUTPUT INSTRUCTIONS:** These Instructions tell the I/O Module what to do. Input/Output Instructions are used to initiate measurements and store the results in Input Storage or to set Digital Control Ports or Continuous Analog Output channels.

**INPUT STORAGE:** That portion of memory allocated for storing the results of Input and Processing Instructions. The values in Input Storage can be displayed and altered from the *6 Mode.

**INSTRUCTION LOCATION NUMBER:** As instructions are entered in a Program Table they are numbered sequentially. The instruction location number is the number giving an instruction's order in the Program Table. When programming a table, the instruction location number and a P (e.g., 04: P00) prompts the user when it is time to enter an instruction.

**INTERMEDIATE STORAGE:** That portion of memory allocated for storing the results of intermediate calculations necessary for operations such as averages or standard deviations. Intermediate storage is not accessible to the user.

**LOW RESOLUTION:** This is the default output resolution. A low resolution data value has 4 significant decimal digits and may range in magnitude from ±0.001 to ±6999. A low resolution data value requires 1 Final Storage location (2 bytes).

**OUTPUT ARRAY:** A string of data points output to Final Storage. Output occurs only when the output flag is set. The first point of an Output Array is the Output Array ID, which gives the table and the Instruction Location Number of the Instruction which set the Output Flag. The data points which complete the Array are the result of the Output Processing Instructions which are executed while the Output Flag is set. The Array ends when the Output Flag is reset at the end of the table or when another Instruction acts upon the Output Flag. Output occurs only when the output flag is set.

**OUTPUT INTERVAL:** The time interval between initiation of a particular Output Array. Output occurs only when the output flag is set. The flag may be set at fixed intervals or in response to certain conditions.

**OUTPUT PROCESSING INSTRUCTIONS:** These Instructions process data values and generate Output Arrays. Examples of Output Processing Instructions include Totalize, Maximize, Minimize, Average, etc. The data sources for these Instructions are values in Input Storage. The results of intermediate calculations are stored in Intermediate Storage. The ultimate destination of data generated by Output Processing Instructions is Final Storage. The transfer of processed summaries to Final Storage takes place when the Output Flag is set by a Program Control Instruction.

**PARAMETER:** When used in conjunction with CR7 instructions, parameters are numbers or codes which are entered when programming the CR7 to specify exactly what the instruction is to do. Once the instruction number has been entered in a Program Table, the CR7 will prompt for the parameters by displaying the parameter number in the ID Field of the display.

**PROCESSING INSTRUCTIONS:** These Instructions allow the user to further process input data values and return the result to Input Storage where it can be accessed for output processing. Arithmetic and transcendental functions are included in these Instructions.

**PROGRAM CONTROL INSTRUCTIONS:** These Instructions are used to modify the sequence of execution of Instructions contained in Program Tables, and to set or clear flags.

**PROGRAM TABLE:** That portion of memory allocated for storing programs consisting of a sequence of user instructions which control data acquisition and processing. Programming can be separated into 2 tables, each having its own user entered Execution Interval. A third table is available for programming subroutines which may be called by instructions in Tables 1 or 2. The *1 and *2 Modes are used to access Tables 1 and 2. The *3 Mode is used to access Subroutine Table 3. The length of the tables are constrained only by the total memory available for programming.

**SAMPLE RATE:** The rate at which measurements are made by the I/O Module. The measurement sample rate is primarily of interest when considering the effect of time skew (i.e., how close in time are a series of measurements). The maximum sample rates are the rates at which the measurements are made when initiated by a single instruction with multiple repetitions. When the I/O Module is told to make several repetitions of a measurement it will make those measurements as fast as possible and buffer the data for use by the Control Module. In normal operation the Control Module will complete all processing called for by the Input Instruction before instructing the I/O Module to make a

measurement specified by a subsequent instruction. The time involved in processing the measurement data to obtain the values stored in Input, Intermediate, and Final Storage makes the throughput rate slower than the measurement sample rate.

**SIGNATURE:** A number which is a function of the data and the sequence of data in memory. It is derived using an algorithm which assures a 99.998% probability that if either the data or its sequence changes, the signature changes.

**THROUGHPUT:** CR7 throughput rate is the rate at which a measurement can be made, scaled to engineering units and the reading stored in Final Storage. The CR7 I/O Module has the ability to scan sensors at a rate exceeding the throughput rate (see SAMPLE RATE). The primary factor affecting throughput rate is the amount of processing specified by the user. In normal operation, all processing called for by an instruction must be completed before moving on to the next instruction. With the 700X Control Module (6303 CPU board), the maximum throughput rate for fast, single-ended measurements is approximately 310 measurements per second (1 second execution: Instruction 1 entered 4 times, 3 times with 99 repetitions, once with 11 repetitions).

**APPENDIX A.  GLOSSARY**

# APPENDIX B. CR7 PROM SIGNATURES FOR SYSTEMS EQUIPPED WITH STANDARD SOFTWARE

| KEY ENTRY | DISPLAY ID FIELD | DATA FIELD | PROM NO. | REMARKS |
|---|---|---|---|---|
| *B | 01: | XXXX | | Program Memory Sig. |
| A | 02: | 22764 | 10437-A | Control Mod. PROM #8 |
| A | 03: | 50101 | 10437-B | Control Mod. PROM #7 |
| A | 04: | 15398 | 10437-C | Control Mod. PROM #6 |
| A | 05: | XXXXX | | Number of K RAM + PROM |
| A | 06: | XX | | Number of E08s |
| A | 07: | XX | | Number of overruns |
| A | 08: | .10000 | | PROM Version 0.1 |
| A | 09: | 0004 | | PROM Revision 4 |
| A | 11:00 | | | |
| 1A | 01: | 21444 | | RAM Sig. of I/O Mod. #1 |
| | | 12196 | 357 | EPROM Sig. of I/O Mod. #1 |
| | OR 11:00 | | 38407 | 357A*1 |

*PROM 357A and a hardware modification make the slow integration time 20 ms (one 50 Hz cycle). This option is available for countries where 50 Hz Ac power is used.*

**- - - - - CR7 SYSTEMS WITH 2 OR MORE I/O MODULES - - - - -**

| | | | | |
|---|---|---|---|---|
| "2A" | 02: | 21444 | | RAM Sig. of I/O Mod. #2 |
| | | 12196 | 357 | EPROM Sig. of I/O Mod. #2 |
| | 11:00 | | | |
| "3A" | 03: | 21444 | - | RAM Sig. of I/O Mod. #3 |
| | | 12196 | 357 | EPROM Sig. of I/O Mod. #3 |
| | 11:00 | | | |
| "4A" | 04: | 21444 | - | RAM Sig. of I/O Mod. #4 |
| | | 12196 | 357 | EPROM Sig. of I/O Mod. #4 |
| | 11:00 | | | |

# APPENDIX C.  BINARY TELECOMMUNICATIONS

*The response time and size of the input buffer of the datalogger must be accounted for when attempting to write a program to make use of the binary commands. The datalogger may delay up to 100 ms before responding to a command or between bytes in a response. The input buffer in the CR10, 21X, and CR7 will now hold 64 bytes of commands; earlier versions of the 21X and CR7 software would only buffer 7 bytes.*

## C.1  TELECOMMUNICATIONS COMMAND WITH BINARY RESPONSES

| Command | Description |
|---------|-------------|
| [no. of loc.]F | BINARY DUMP - CR7 sends, in Final Storage Format (binary, the number of Final Storage locations specified (from current MPTR locations), then Signature (no prompt). |

### DATALOGGER J AND K COMMANDS

**3142J** The 3142J command is used to toggle datalogger user flags, request Final Storage data, and to establish the input locations returned by the K command. The format of the command is as follows:

3142J<CR>abcd...nNULL

where

1)  "3142J<CR>" is the command.

2)  "a" is a 1 byte value representing the user flags to be toggled. The most significant bit (MSB), if set, will toggle datalogger user flag 8. Likewise, the 2nd most significant bit, if set, will toggle user flag 7, and so on to the least significant bit which, if set, toggles user flag 1. Toggle means that if a flag is set, it will be then reset, or if it is reset, it will be set.

3)  "b" is a 1 byte value whose MSB will determine whether Final Storage Data is returned after the K command. If the bit is set, Final Storage Data, if any, will be returned after the next K command. The datalogger initially has this bit reset upon entering telecommunications, but once set by a J command, it will remain set until reset by another J command or telecommunications is terminated.

The 2nd MSB set means a port toggle byte will follow and port status is to be returned with the K command. Like the MSB, this bit is reset upon entering telecommunications, but remains set once set until reset by another J command or telecommunications is terminated.

The remaining bits are reserved.

4)  If the 2nd MSB in "b" was set then "c" is a port toggle byte, otherwise "c,d,...,n" are each 1 byte binary values each representing a datalogger Input Storage location. The data at those locations will be returned after the next K command. ASCII code 1 (0000001 binary) represents input location 1. ASCII codes 2 (00000010 binary) represents input location 2, and so on. The order of the location requests is not important. The list is limited, however, to 62 total location requests.

5)  "Null" or ASCII code 0 (00000000 binary ) terminates the J command. Alternately, 11111111 binary aborts the J command. If aborted, flags will not be toggled and location requests will not be saved.

| User Enters | Datalogger Echo |
|-------------|-----------------|
| 3 | 3 |
| 1 | 1 |
| 4 | 4 |
| 2 | 2 |
| J | J |
| CR | CR |
|  | LF |
|  | < |
| a | a |
| b | b |
| c | c |
| d | d |
| n | n |
| Null | Null |

**K**  The K command returns datalogger time, user flag status, port status if requested, the data at the input locations requested in the J command, and Final Storage Data if requested by the J command.  The format of the command is K<CR> (K Return).  The datalogger will echo the K and Return and send a Line Feed.  The amount of data that follows depends on the J command previously executed;  four time bytes, a user flags byte, four bytes for each input location requested in the J command, Final Storage data in Campbell Scientific's binary format if requested by the J command, and terminating in 7F 00 HEX and two signature bytes.

| User Enters | Datalogger Echo |
| --- | --- |
| K | K |
| CR | CR |
|  | LF |
|  | Time Minutes byte 1 |
|  | Time Minutes byte 2 |
|  | Time Tenths byte 1 |
|  | Time Tenths byte 2 |
|  | Flags byte |
|  | Ports byte (if requested) |
|  | Data1 byte 1 |
|  | Data1 byte 2 |
|  | Data1 byte 3 |
|  | Data1 byte 4 |
|  | Data2 byte 1 |
|  | Data2 byte 2 |
|  | Data2 byte 3 |
|  | Data2 byte 4 |
|  | DataN byte 1 |
|  | DataN byte 2 |
|  | DataN byte 3 |
|  | DataN byte 4 |
|  | Final Storage Data bytes |
|  | 01111111 binary byte |
|  | 00000000 binary byte |
|  | Signature byte 1 |
|  | Signature byte 2 |

Time Minutes byte 1 is most significant.  Convert from binary to decimal.  Divide by 60 to get hours, the remainder is minutes.  For example, 00000001 01011001 (01 59 HEX) is 345 decimal minutes or 5:45.

Time Tenths byte 1 is most significant.  Convert from binary to decimal.  Divide by 10 to get seconds and tenths of seconds.  For example, 00000001 11000110 (01 C6 HEX) is 454 decimal or 45.4 seconds.  Thus the datalogger time for 01 59 01 C6 HEX is 5:45:45.4.

The Flags byte expresses datalogger user flag status.  The most significant bit represents Flag 8, and so on to the least significant bit which represents Flag 1.  If a bit is set, the user flag is set in the datalogger.

The optional ports byte expresses the datalogger port status.  The most significant bit represents Port 8, and so on to the least significant bit which represents Port1.

For each input location requested by the J command, four bytes of data are returned.  The bytes are coded in Campbell Scientific, Inc. Floating Point Format.  The format is decoded to the following:

$$Sign(Mantissa*2^{(Exponent)})$$

Data byte 1 contains the Sign and the Exponent.  The most significant bit represents the Sign; if zero, the Sign is positive, if one, the Sign is negative.  The signed exponent is obtained by subtracting 40 HEX (or 64 decimal) from the 7 remaining least significant bits.

Data bytes 2 to 4 are a binary representation of the mantissa with byte 2 the most significant and 4 the least.  The mantissa ranges in value from 80 00 00 hex (0.5 decimal) to FF FF FF HEX ($1-2^{-24}$ decimal).  The one exception is for zero which is 00 00 00 00 HEX.

The Mantissa is calculated by converting Data bytes 2 to 4 into binary.  Each bit represents some fractional value which is summed for all 24 bits.  The bits are arranged from MSB to LSB with the most significant as $bit_{23}$ and least significant as $bit_0$.  The value that each bit represents = $2^{n-24}$; where n=bit location.  For example, if there was a 1 at $bit_{20}$, it's value would be $2^{(20-24)}$ or $2^{-4}$.

A simple method for programming this is as follows:

Set Mantissa = 0.

Set Bit Value = 0.5.

For loop count = 1 to 24 do the following:

> If the MSB is one, then add Bit Value to the Mantissa.
>
> Shift the 24 bit binary value obtained from Data bytes 2 to 4 one bit to the left.
>
> Multiply Bit Value by 0.5.

End of loop.

Another method that can be used as an estimate is to convert Data bytes 2 to 4 from a long integer to floating point and dividing this value by 16777216.

**As an example of a negative value, the datalogger returns BF 82 0C 49 HEX.**

Data byte 1 = BF HEX.

Data byte 2 to 4 = 82 0C 49 HEX (or 8522825 decimal).

Data byte 1 is converted to binary to find the Sign.  BF HEX = 10111111 BINARY.

The most significant bit is 1 so the Sign is NEGATIVE.

The exponent is found by subtracting 40 HEX from the remaining least significant bits. Converting the binary to hexadecimal, 111111 BINARY = 3F HEX (or 63 decimal).

3F - 40 HEX = FF FF FF FF FF HEX.  Or in decimal:  63 - 64 = -1.

Exponent is -1 decimal.

The binary representation of Data bytes 2 to 4 is:  10000010 00001100 01001001.

Summing all the fractional values:  $2^{-1} + 2^{-7} + 2^{-13} + 2^{-14} + 2^{-18} + 2^{-21} + 2^{-24} = 0.50800$.

Using the estimate method to find the Mantissa = 82 0C 49 HEX / 1 00 00 00 HEX (or 8522825 / 16777216) which is 0.50800 decimal.

The value is then $(-)0.508*2^{-1}$ which equals -0.254.

**As an example of a positive value, the datalogger returns 44 D9 99 9A HEX.**

Data byte 1 = 44 HEX.

Data byte 2 to 4 = D9 99 9A HEX (or 891290 decimal).

Data byte 1 is converted to binary to find the Sign.  44 HEX = 01000100 BINARY.

| NOTE:  Don't lose the leading zero! |
| --- |

The most significant bit is 0 so the Sign is POSITIVE.

The exponent is found by subtracting 40 HEX from the remaining least significant bits. Converting the binary to hexadecimal, 1000100 BINARY = 44 HEX (or 68 decimal).

44 - 40 HEX = 4 HEX.  Or in decimal: 68 - 64 = 4.

Exponent is 4 decimal.

The binary equivalent of Data bytes 2 to 4 is: 11011001 10011001 10011010.

Summing all the fractional values:

$2^{-1} + 2^{-2} + 2^{-4} + 2^{-5} + 2^{-8} + 2^{-9} + 2^{-12} + 2^{-13} + 2^{-16} + 2^{-17} + 2^{-20} + 2^{-21} + 2^{-23} = 0.85000$.

Using the estimate method to find the Mantissa = D9 99 9A HEX / 1 00 00 00 HEX (or 14260634 / 16777216) which is 0.85000 decimal.

The value is then $(+)0.85*2^4$ which equals 13.60.

If appropriately requested by a J command, Final Storage data, if any, will immediately follow the input location data.  Refer to the datalogger manual for a description of how to decode Final Storage data in Campbell Scientific's binary data format.  Final Storage data will be limited to not more than 1024 bytes per K command.

The K command data is terminated with 7F 00 HEX (a unique binary format code) followed by two signature bytes.  Refer to the datalogger manual for the meaning and calculation of the signature bytes.  The signature in this case is a function of the first time byte through the 7F 00 HEX bytes.  Calculate the signature of the bytes received and compare with the signature received to determine the validity of the transmission.

## C.2  FINAL STORAGE FORMAT

CR7 data is formatted as either 2 byte LO Resolution or 4 byte HI Resolution values.  The first two bytes of an output array contain a code noting the start of the output array and the output array ID, followed by the 2 or 4 byte data values.  At the end of the data sent in response

# APPENDIX C. BINARY TELECOMMUNICATIONS

to the telecommunications F command a 2 byte signature is sent (see below).

Representing the bits in the first byte of each two byte pair as ABCD EFGH (A is the most significant bit, MSB), the byte pairs are described below.

## LO RESOLUTION FORMAT - D,E,F, NOT ALL ONES

| BITS | DESCRIPTION |
|---|---|
| A | Polarity, 0 = +, 1 = -. |
| B, C | Decimal locators as defined below. |
| D-H plus second byte | 13 bit binary value (D=MSB). Largest possible number without D, E, and F all 1 is 7167, but CAMPBELL SCIENTIFIC defines the largest allowable range as 6999. |

The decimal locators can be viewed as a negative base 10 exponent with decimal locations as follows:

| B | C | Decimal Location |
|---|---|---|
| 0 | 0 | XXXX. |
| 0 | 1 | XXX.X |
| 1 | 0 | XX.XX |
| 1 | 1 | X.XXX |

## DATA TYPE WHEN D,E,F, ALL EQUAL ONE

If D, E, and F are all ones, the data type is determined by the other bits as shown below. X implies a "don't care" condition; i.e., the bit can be either 1 or 0 and is not used in the decode decision.

| A | B | C | D | E | F | G | H | DATA TYPE AND SECOND BYTE FORMAT |
|---|---|---|---|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 | 1 | 0 | X | A,B,C, = 1 - Start of output array, G=0. H is the most significant bit of the output array ID. All 8 bits of the 2nd byte are also included in the ID. |
| X | X | 0 | 1 | 1 | 1 | X | X | C = 0 - First byte of a 4 byte value. |
| 0 | 0 | 1 | 1 | 1 | 1 | X | X | A,B = 0; C = 1 - Third byte of a 4 byte value. |
| 0 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | A = 0; remaining bits = 1 - First byte of a 2 byte "dummy" word. The CR10 always transmits a 0 for the 2nd byte, but the word can be decoded on the basis of the 1st byte only. |

## HI RESOLUTION FORMAT

Continuing to use the A-H bit representation, the four byte number is shown below as two two byte pairs.

| AB0111GH | XXXXXXXX | 001111GH | XXXXXXXX |
|---|---|---|---|

| BITS, 1ST BYTE, 1ST PAIR | DESCRIPTION |
|---|---|
| CDEF = 0111 | Code designating 1st byte pair of four byte number. |
| B | Polarity , 0 = +, 1 = -. |
| G,H,A, | Decimal locator as defined below. |
| 2nd byte | 16th - 9th bit (left to right) of 17 bit binary value. |
| ABCDEF = 001111 | Code designating 2nd byte pair of four byte number. |
| G | Unused bit. |
| H | 17th and MSB of 17 bit binary value. |
| 2nd byte | 8th - 1st bit (left to right) of 17 bit binary value. |

CAMPBELL SCIENTIFIC defines the largest allowable range of a high resolution number to be 99999.

Interpretation of the decimal locator for a 4 byte data value is given below. The decimal equivalent of bits GH is the negative exponent to the base 10.

| BITS G H A | DECIMAL FORMAT 5 digits |
|---|---|
| 0 0 0 | XXXXX. |
| 0 0 1 | XXXX.X |
| 0 1 0 | XXX.XX |
| 0 1 1 | XX.XXX |
| 1 0 0 | X.XXXX |
| 1 0 1 | .XXXXX |

## C.3 GENERATION OF SIGNATURE

At the end of a binary transmission, a signature is sent. The signature is a 2 byte integer value which is a function of the data and the sequence of data in the output array. It is derived with an algorithm that assures a 99.998% probability of detecting a change in the data or its sequence. The CR7 calculates the signature using each transmitted byte except the 2 byte signature itself. By calculating the signature of the received data and comparing it to the transmitted signature, it can be determined whether the data was received correctly.

### SIGNATURE ALGORITHM

- S1,S0    represent the high and low bytes of the signature, respectively
- M    represents a transmitted data byte

- n    represents the existing byte
- n+1    represents the new byte
- T    represents a temporary location
- C    represents the carry bit from a shift operation

1. The signature is initialized with both bytes set to hexadecimal AA.

$$S_1(n) = S_0(n) = AA$$

2. When a transmitted byte, M(n+1), is received, form a new high signature byte by setting it equal to the existing low byte. Save the old high byte for later use.

$$T_1 = S_1(n)$$
$$S_1(n+1) = S_0(n)$$

3. Form a temporary byte by shifting the old low signature byte one bit to the left and adding any carry bit which results from the shift operation. A "shift left" is identical to a multiply by 2. Ignore any carry bit resulting from the add.

$$T_2 = \text{shift left } (S_0(n)) + \text{carry}$$

4. Form the new low signature byte by adding the results of operation 3 to the old high signature byte and the transmitted byte. Ignore any carry bits resulting from these add operations.

$$S_0(n+1) = T_2 + S_1(n) + M(n+1)$$

As each new transmitted byte is received, the procedure is repeated.

# APPENDIX D. CALIBRATION PROCEDURES

*The CR7 requires very little maintenance or calibration. Measurements are made in such a way that small errors in the calibration are automatically removed. Over time, shifts in the calibration are possible, however. Measurements can be made to determine whether the accuracy of the CR7 is within the specifications given in Section I.3. If needed, the calibration procedures described in this section can be performed by an experienced technician having the suggested equipment.*

> *NOTE: The precision of the CR7 exceeds that of most standard electronic equipment. These procedures require that the test equipment have a precision equal to or better than the CR7.*

*The following procedures are for calibrating the voltage reference and the clock. Other factors such as range ratios, DAC non-linearity, and offset in either the switched excitation or the CAO voltage require that the CR7 be returned to the factory for repair. Please call the factory to obtain authorization before sending in the unit.*

## D.1 VOLTAGE REFERENCE CALIBRATION PROCEDURE

The following procedure assumes that the CR7 being calibrated has an analog input card configured as analog input card #1 and an excitation card configured as excitation card #1. Adjustments would need to be made in the programming example if the cards are numbered differently.

**SUGGESTED INSTRUMENTS:**

Five and one half digit digital volt meter (DVM) with 10 microvolt resolution. The accuracy of the DVM needs to be equal to that of the CR7 which is ±1 mV at 5 VDC.

**PROCEDURE:**

1. With a small, flat screw driver, pry off the three silver caps on the top of the Analog Interface Card in the I/O module (see Figure D.2-1).

2. Monitor the Digital to Analog Converter (DAC) output by connecting the positive lead of the DVM to the DAC OUTPUT TEST JUMPER. Connect the negative lead of the DVM to a ground on the analog input card. Set the DVM to read on its most sensitive DC Volt scale.

3. Set the DAC to output 0.000 volts DC, by programming the CR7 as follows:

| | | |
|---|---|---|
| * | 1 | Table 1 Programs |
| 01: | 1 | Sec. Execution Interval |
| | | |
| 01: | P22 | Excitation with Delay |
| 01: | 1 | EX Card |
| 02: | 1 | EX Chan |
| 03: | 900 | Delay w/EX (units=.01sec) |
| 04: | 0000 | Delay after EX (units=.01sec) |
| 05: | 0.0000 | mV Excitation  CHANGE AS INSTRUCTED |
| | | |
| 02: | P | End Table 1 |
| * | 0 | Compiles the instructions |

4. Adjust the DAC OFFSET POTENTIOMETER for a DVM reading of 0.0000 V (±0.0001 V).

5. Change the CR7's program given in step three so that parameter 5 reads 4000 to change the DAC output to 4 V. Set up the DVM to read 4 VDC.

6. Adjust the VOLTAGE REFERENCE GAIN POTENTIOMETER until a DVM reading of 4.0000 V (±0.001 V) is obtained.

7. Change the CR7s program given in step 3 so that parameter 5 reads -4000 to change the DAC output to -4V.

8. A DVM reading of -4.000 V (±0.001 V) verifies the DAC linearity.

9. Lock both potentiometers into position with a dab of finger nail polish.

## D.2 CLOCK CALIBRATION PROCEDURE

The 700X control module contains 3, 4, or 5 cards. The CPU card has one blue connector with a ribbon cable connecting it to the 9 pin SERIAL I/O port on the front of the CR7. The clock circuitry resides on this card.

The frequency of the crystal exhibits a parabolic response to temperature. The frequency maximum occurs at room temperature and drops off slowly at hotter or colder temperatures. When the CR7 leaves the factory it is calibrated to be 20 ppm fast. If the CR7 is placed in a controlled environment with the temperature close to room temperature, it is better to set the crystal frequency exactly on.

**SUGGESTED INSTRUMENTS:**

Digital frequency counter
STD Bus extender card

**PROCEDURE:**

1. Remove the CPU card from the control module. Insert an STD Bus extender card into the emptied slot. Plug the CPU card into the extender card. This is the best way to gain access to the adjustment points. However, if an "extender card" cannot be obtained, remove all the other cards and move the CPU card into the back most slot for better, though still difficult, accessibility.

2. Set up the frequency counter to measure period and connect to pin 3 of the Integrated Circuit shown at Location B16 on Figure D.2-2. Connect the ground lead to the negative (flat) side of the 10 µF capacitor at location F21. Adjust the variable capacitor at location E21 on the same figure for a period of 49,999.00 microseconds (20 ppm fast) or 50,000.00 microseconds (exactly on).

3. Lock the variable capacitor into position with a dab of finger nail polish.

DAC OFFSET
POTENTIOMETER

VOLTAGE REFERENCE
GAIN POTENTIOMETER

DAC OUTPUT
TEST JUMPER

I/O PROCESSOR CARD
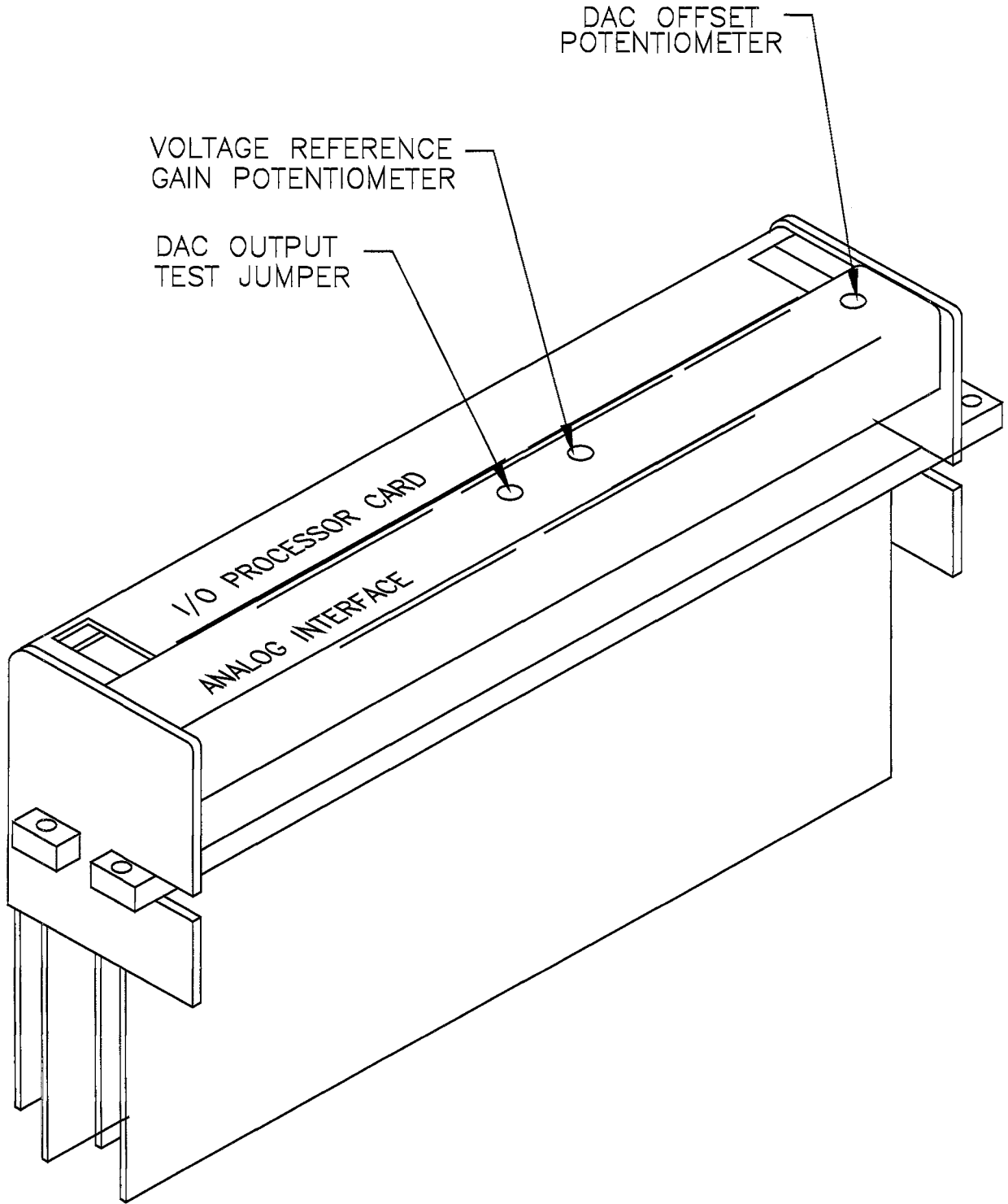
ANALOG INTERFACE

FIGURE D.2-1.  Calibration Points for the Analog Interface Card

PIN 3 ————————→

PROM 3

PROM 2

PROM 1

FIGURE D.2-2.  CR7X CPU Card

# LIST OF TABLES

# LIST OF FIGURES

## 13. CR7 MEASUREMENTS

## 14. INSTALLATION

## 15. I/O CARD ADDRESSING AND MULTIPLE I/O MODULES

## APPENDIX D.   CALIBRATION PROCEDURES

# CR7 INDEX

# X

X * F  [Instruction 37]  **10-2**
X * Y  [Instruction 36]  **10-2**
X + F  [Instruction 34]  **10-1**
X + Y  [Instruction 33]  **10-1**
X - Y  [Instruction 35]  **10-1**
X / (1-X)  [Instruction 59]  **10-6**
X / Y  [Instruction 38]  **10-2**
X Mod F  [Instruction 46]  **10-3**
$X^Y$  [Instruction 47]  **10-3**

# Y

Year, Day or time (*5 Mode),
    Setting/displaying  **1-2**

# Z

Z = 1 / X  [Instruction 42]  **10-2**
Z = ABS(X)  [Instruction 43]  **10-3**
Z = ARCTAN (X/Y) [Instruction 66]  **10-10**
Z = EXP(X)  [Instruction 41]  **10-2**
Z = F  [Instruction 30]  **10-1**
Z = FRAC(X)  [Instruction 44]  **10-3**
Z = INT(X)  [Instruction 45]  **10-3**
Z = LN(X)  [Instruction 40]  **10-2**
Z = SIN(X)  [Instruction 48]  **10-3**
Z = SQRT(X)  [Instruction 39]  **10-2**
Z = X  [Instruction 31]  **10-1**
Z = X * F  [Instruction 37]  **10-2**
Z = X + F  [Instruction 34]  **10-1**
Z = X * Y  [Instruction 36]  **10-2**
Z = X + Y  [Instruction 33]  **10-1**
Z = X - Y  [Instruction 35]  **10-1**
Z = X / Y  [Instruction 38]  **10-2**
Z = X MOD F  [Instruction 46]  **10-3**
Z = $X^Y$  [Instruction 47]  **10-3**
Z = Z + 1  [Instruction 32]  **10-1**

# CR7 PROMPT SHEET

## (OS7-0.2 PROMs)

This prompt sheet briefly details the CR7's instruction set and command structure. It is intended for field use or as a reference by those familiar with CR7 programming; more detailed information and examples are available in the CR7 Operator's Manual. Computer-assisted programming and communications are supported by PC208 and PC208W Software.

## *Modes

To enter each Mode, key in a * ("star"), followed by the desired mode number.

### KEY DEFINITION SUMMARY

Common keystrokes used to interrogate and program the CR7 via the keyboard/display or a remote terminal.

| | |
|---|---|
| `0`-`9` Key in numeric data, instruction number, or parameter | `C` Change the sign of a floating point number, or index an input location |
| `A` Enter the displayed number, or advance through a program table or data storage | `D` Enter a decimal point |
| `B` Back up through a program table or data storage | `#` Clear digit just keyed; display storage location number |

---

`*` `0` **Compile program, begin LOGging data, indicate Active Table. Resets Input Storage, Flags, and Ports**

`*` `1` **Display/Enter Program Instructions**

01:xx      Advance to a given Instruction location

01:x.xxxx    Execution Interval:

Valid entries
| are multiples of | for Range of |
|---|---|
| 0.0125 s .... | 0.0125 to 0.1 s (Table 1 only) |
| 0.1 s .... | 0.1 to 6553 s |

01:Pxx     Program Instruction (see following pages)

`*` `2` **Display/Enter Program Instructions** Same structure as *1. Allows use of a different Execution Interval

`*` `3` **Display/Change Subroutine Program Table** Programming similar to *1 and *2; no Execution Interval

#### *1, *2 and *3 Commands

| | | |
|---|---|---|
| `#` | `A` | Advance to next instruction |
| `#` | `B` | Back up to previous instruction |
| `#` | `D` | Delete entire instruction |

`*` `4` **Enable Final Storage Output to Peripheral Device (do not use if Instruction 96 is in program)**

01: A B Output Enable Code
  A   Tape
      0 = disabled; 1 = enabled
  B   Printer
      0 = disabled; 1 = enabled

02:0y Baud Rate Code (printer)
| 0 | 300 baud |
| 1 | 1200 baud |
| 2 | 9600 baud |
| 3 | 76800 baud |

`*` `5` **Display/Set Datalogger Time**
:HH:MM:SS (displays current datalogger time)
05:xx      Year
05:xxxx    Day of Year (calendar on back)
05:HHMM   Hours Minutes

---

`*` `6` **Display/Change: Input Storage Data Values/Flags, Ports. Compile Program without resetting Input Storage, Flags, or Ports.**

06:xxxx      Advance to a given Input Storage Location

#### *6 Commands

| | |
|---|---|
| `#` | Display Input Location Number or enter location to jump to |
| `C` | Enter or Change value in Input Location |
| `D` | Display flags 1-8, toggle flag with keys 1-8 |
| `0` | Display ports 8-1, toggle with keys 1-8. Active port card is set with Instruction 20 |

---

`*` `7` **Display Final Storage Data**
07:xxxx      DSP location or enter location to advance to

#### *7 Commands

| | |
|---|---|
| `#` | Display Final Storage location number; enter location to jump to, or C to display data |
| `#` `A` | Advance to start next array |
| `#` `B` | Back up to start of array |

---

`*` `8` **Manual Data Dump to Tape**
01:xxxxx    TPTR location/start of dumping
02:xxxxx    DSP location/end of dump
03:xx        Enter any number to start dump
            (# Aborts dump)

`*` `9` **Manual Dump to Printer/Storage Module**
09:xx        Enter Output Code
 1y         Printable ASCII       y=Baud Rate
 2y         Final Storage Format   (see *4)
 30        SM192/716 Storage Module
 31        Filemark to SM192/716
01:xxxxx    PPTR location/start of dump
02:xxxxx    DSP location/end of dump
03:xx        Enter any number to start dump

---

`*` `A` **Display/Change Memory Allocation**
01:xxxx    Input Storage locations
02:xxxx    Intermediate Storage location
03:xxxx    Final Storage location
04:xxxx    Remaining program memory (bytes)

`*` `B` **Display Signatures**
01:xxxxx   Program signature
02:xxxxx   First PROM signature
03:xxxxx   Second PROM signature
04:xxxxx   Third PROM signature
05:xxx     Memory Test
06:xx      No. of E08 Errors
07:xx      No. of Overrun Errors
08:xxxx.x   Version number
09:xxxx    Revision number
11:00      Enter I/O Module No.
01:xxxxx   I/O RAM Signature
01:xxxxx   I/O PROM Signature

`*` `C` **Display/Change Security**
12:0000    (if enabled) Enter password
01:xx
   00       Temporarily disable security
   01       Advance to window 2 to set new password
02:xxxx    Set password, 0000 disables security if window 1 is set to 0

`*` `D` **Store/Load Program**
13:xx      Enter command (Commands 1 and 2 require baud rate code. See *4 mode)
 1         - Print program (ASCII)
 2         - Load program (ASCII)
 71        - Store/Load/Clear program from Storage Module

Storage Module Command Codes
   1x   Store program x in Storage Module
   2x   Load program x from Storage Module
   3x   Clear program x from Storage Module
      x =1-8

---

*NOTE: x represents a digit from 0 to 9 unless otherwise defined*

**CAMPBELL SCIENTIFIC, INC.**

# CR7 INSTRUCTION AND PARAMETER SUMMARY

## INPUT/OUTPUT INSTRUCTIONS

| INST. | | 01: | 02: | 03: | 04: | 05: | 06: | 07: | 08: | 09: | 10: | 11: | 12: |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | VOLT (SE) | REPS | RANGE† | IN CARD | SE CHAN | LOC | MULT | OFFSET | | | | | |
| 2 | VOLT (DIFF) | REPS | RANGE† | IN CARD | DIFF CHAN | LOC | MULT | OFFSET | | | | | |
| 3 | PULSE | REPS | IN CARD | PULSE CHAN | CONFIG† | LOC | MULT | OFFSET | | | | | |
| 4 | EX DEL (SE) | REPS | RANGE† | IN CARD | SE CHAN | EX CARD | EX CHAN | MEAS/EX | DEL 0.01 s | EXCIT mV | LOC | MULT | OFFSET |
| 5 | AC HALF BR | REPS | RANGE† | IN CARD | SE CHAN | EX CARD | EX CHAN | MEAS/EX | EXCIT mV | LOC | MULT | OFFSET | |
| 6 | FULL BR | REPS | RANGE† | IN CARD | DIFF CHAN | EX CARD | EX CHAN | MEAS/EX | EXCIT mV | LOC | MULT | OFFSET | |
| 7 | 3W HALF BR | REPS | RANGE† | IN CARD | SE CHAN | EX CARD | EX CHAN | MEAS/EX | EXCIT mV | LOC | MULT | OFFSET | |
| 9 | FULL BR W/COMP | REPS | EX RANGE† | BR RANGE† | IN CARD | 1ST DIFF CHAN | EX CARD | EX CHAN | MEAS/EX | EXCIT MV | LOC | MULT | OFFSET |
| 10 | BATT VOLT | LOC | | | | | | | | | | | |
| 11 | TEMP (107) | REPS | IN CARD | SE CHAN | EX CARD | EX CHAN | LOC | MULT | OFFSET | | | | |
| 12 | RH (207) | REPS | IN CARD | SE CHAN | EX CARD | EX CHAN | MEAS/TEMP | TEMP LOC | RH LOC | MULT | OFFSET | | |
| 13 | TEMP TC (SE) | REPS | RANGE† | IN CARD | SE CHAN | TC TYPE† | REF LOC | LOC | MULT | OFFSET | | | |
| 14 | TEMP TC (DIFF) | REPS | RANGE† | IN CARD | DIFF CHAN | TC TYPE† | REF LOC | LOC | MULT | OFFSET | | | |
| 16 | TEMP RTD | REPS | R/Ro LOC | LOC | MULT | OFFSET | | | | | | | |
| 17 | TEMP PANEL | IN CARD | LOC | | | | | | | | | | |
| 18 | TIME | OPTION† | MOD/BY | LOC | | | | | | | | | |
| 19 | SIGNATURE | LOC | | | | | | | | | | | |
| 20 | PORT SET | OPTION† | EX CARD | PORT NO | | | | | | | | | |
| 21 | ANALOG OUT | EX CARD | CAO CHAN | mV LOC | | | | | | | | | |
| 22 | EXCIT W/DEL | EX CARD | EX CHAN | DEL W/EX | DEL AFTER | EXCIT mV (del. units- 0.01 s) | | | | | | | |
| 23 | I/O MODULE | MODULE NO (1,2,3,4) | | | | | | | | | | | |
| 26 | TIMER | LOC (0 reset timer) | | | | | | | | | | | |
| 101 | SDM-INT8 | ADDR | C:8765† | C:4321† | F:8765† | F:4321† | AVG OPT† | LOC | MULT | OFFSET | | | |
| 102 | SDM-SW8A | REPS | ADDR | FUNCTION† | CHAN | LOC | MULT | OFFSET | | | | | |
| 103 | SDM-AO4 | REPS | ADDR | LOC | | | | | | | | | |
| 104 | SDM-CD16AC | REPS | ADDR | LOC | | | | | | | | | |
| 113 | SDM-SIO4 | REPS | ADDR | MODE | CMD | 1STPAR | 2NDPAR | VALREP | LOC | MULT | OFFSET | | |
| 114 | SET TIME | OPTION† | LOC | | | | | | | | | | |
| 115 | SDM BAUD | BIT PERIOD | 10 µs | | | | | | | | | | |

## †Option Codes

### 1-14 RANGE codes:
Slow (16.66 ms integ. time)

Fast (250 µs integ. time

| | | Full scale range | | |
|---|---|---|---|---|
| 1 | 11 | ± | 1500 | µV |
| 2 | 12 | ± | 5000 | µV |
| 3 | 13 | ± | 15 | mV |
| 4 | 14 | ± | 50 | mV |
| 5 | 15 | ± | 150 | mV |
| 6 | 16 | ± | 500 | mV |
| 7 | 17 | ± | 1500 | mV |
| 8 | 18 | ± | 5000 | mV |

### 3 CONFIGuration codes:
| x | Record all counts |
|---|---|
| 1x | Discard counts beyond execution interval |
| 2x | Discard counts, output frequency (Hz) |

Where x is:
| 0 | High frequency |
|---|---|
| 1 | Low frequency |
| 2 | Switch Closure |

### 13,14 ThermoCouple TYPE codes:
| 1 | T (copper-constantan) |
|---|---|
| 2 | E (chromel-constantan) |
| 3 | K (chromel-alumel) |
| 4 | J (iron-constantan) |
| 5 | B (platinum-rhodium) |
| 6 | R (platinum-rhodium) |
| 7 | S (platinum-rhodium) |
| 1x | Output temp difference |
| 2x | Skip every other channel (Inst. 13 only) |
| 8x | Use with A5B40 |
| x | =TC Type 1-7 |

To increment REF LOC with each rep, enter the REF LOC as xxxx--

### 18 OPTION codes:
| 0 | 0.1 seconds into minute (max 600) |
|---|---|
| 1 | minutes into day (max 1440) |
| 2 | hours into year (max 8784) |

### 20 OPTION codes:
| 0 | Set low |
|---|---|
| 1 | Set high |
| 1x | Set according to flag x |
| 2x | Set opposite of flag x |
| 30 | Set active card |

### 101 C:8765, C:4321 Each digit Configures respective channel
| 0 | High level, rising edge |
|---|---|
| 1 | High level, falling edge |
| 2 | Low level ac, rising edge |
| 3 | Low level ac, falling edge |

### 101 F:8765, F:4321 Each digit sets Function for respective channel
| 0 | No value returned |
|---|---|
| 1 | Period, ms |
| 2 | Frequency, kHz |
| 3 | Time since previous channel, ms |
| 4 | Time since channel 1, ms |
| 5 | Counts on 2 since 1 interpolated |
| 6 | Low resolution frequency, kHz |
| 7 | Counts |
| 8 | Integral counts on 2 since 1 |

### 101 AVeraGing OPTion
| 0 | Execution interval averaging |
|---|---|
| 0-- | Continuous averaging |
| xxxx | Specify average interval in ms |
| xxxx-- | Capture all events until xxxx edges of channel 1 |
| 9999-- | Test Memory |

### 102 FUNCTion
| 0 | Channel state |
|---|---|
| 1 | Duty cycle |
| 2 | Counts |
| 3 | Memory test |

### 114 OPTION codes:
| 0 - | Set time with hr, min, sec values from 3 input locations |
|---|---|
| 1 - | Set time with day, hr, min, sec values from 4 input locations |
| 2 - | Set time with yr, day, hr, min sec from 5 input locations |

## PROCESSING INSTRUCTIONS

| INST. | | 01: | 02: | 03: |
|---|---|---|---|---|
| 30 | Z=F | F | Z | |
| 31 | Z=X | X | Z | |
| 32 | Z=Z+1 | Z | | |
| 33 | Z=X+Y | X | Y | Z |
| 34 | Z=X+F | X | F | Z |
| 35 | Z=X-Y | X | Y | Z |
| 36 | Z=X*Y | X | Y | Z |
| 37 | Z=X*F | X | F | Z |

| INST. | | 01: | 02: | 03: |
|---|---|---|---|---|
| 38 | Z=X/Y | X | Y | Z |
| 39 | Z=SQRT(X) | X | Z | |
| 40 | Z=LN(X) | X | Z | |
| 41 | Z=EXP(X) | X | Z | |
| 42 | Z=1/X | X | Z | |
| 43 | Z=ABS(X) | X | Z | |
| 44 | Z=FRAC(X) | X | Z | |

| INST | | 01: | 02: | 03: |
|---|---|---|---|---|
| 45 | Z=INT(X) | X | Z | |
| 46 | Z=X MOD F | X | F | Z |
| 47 | Z=X^y | X | Y | Z |
| 48 | Z=SIN(X) | X | Z | |
| 49 | SPA. MAX | SWATH | 1ST LOC | MAX† |
| 50 | SPA. MIN | SWATH | 1ST LOC | MIN† |
| 51 | SPA. AVG | SWATH | 1ST LOC | AVG |

| INST. | | 01: | 02: | 03: | 04: | 05: | 06: | 07: | 08: | 09: |
|---|---|---|---|---|---|---|---|---|---|---|
| 53 | A*X+B | START LOC | A1 | B1 | A2 | B2 | A3 | B3 | A4 | B4 |
| 54 | BLOCK MOVE | NO VALS | S LOC | STEP | D LOC | STEP | | | | |
| 55 | POLYNOMIAL | REPS | X | F(X) LOC | C0 | C1 | C2 | C3 | C4 | C5 |
| 56 | SAT VP | TEMP LOC | VP | | | | | | | |
| 57 | W/D TEMP VP | PRES LOC | DB TEMP | WB TEMP | VP LOC | | | | | |
| 58 | LP FILTER | REPS | SMPL LOC | LOC | W | | | | | |
| 59 | X/(1-X) | REPS | X | MULT | | | | | | |
| 61 | INDIR MOVE | SOURCE | DEST | | | | | | | |
| 66 | Z=ARCTAN (X/Y) | X | Y | Z | | | | | | |
| 68 | PARA EXTN | SENSOR NO | SENSOR NO | SENSOR NO | SENSOR NO | SENSOR NO | SENSOR NO | SENSOR NO | SENSOR NO | |

## †Option Codes

### 49, 50 MAX/MIN codes:
| 0xxx | Store spatial max or min at location xxx |
|---|---|
| 1xxx | Max or min at location xxx and location of max or min at xxx + 1 |

# OUTPUT PROCESSING INSTRUCTIONS

| INST. | | 01: | 02: | 03: | 04: | 05: | 06: | 07: |
|---|---|---|---|---|---|---|---|---|
| 69 | WIND VECTOR | REPS | SMPL/SUB INT† | SEN/OUT† | WS/E | WD/N | | |
| 70 | SAMPLE | REPS | LOC | | | | | |
| 71 | AVERAGE | REPS | LOC | | | | | |
| 72 | TOTALIZE | REPS | LOC | | | | | |
| 73 | MAXIMIZE | REPS | TIME† | LOC | | | | |
| 74 | MINIMIZE | REPS | TIME† | LOC | | | | |
| 75 | HISTOGRAM | REPS | BINS | FORM† | B SEL LOC | WV LOC† | LO LIM | HI LIM |
| 77 | REAL TIME | OPTION† | | | | | | |
| 78 | RESOLUTION | OPTION† | | | | | | |
| 79 | SMPL ON MAX/MIN | REPS | LOC | | | | | |
| 80 | STORE AREA | AREA† | ID/LOC | | | | | |
| 82 | STD DEV | REPS | LOC | | | | | |

## †Option Codes

**69 SaMPLes/SUBINTerval (std dev):**

| | |
|---|---|
| 0 | No subinterval |
| xxxx | Number of scans per subinterval |

**SENsor type/OUTput codes:**

| | |
|---|---|
| x0 | Avg WS, θ1; σ(θ1) |
| x1 | Avg WS, θ1 |
| x2 | Avg WS, resultant U,θu: σ(θu) |
| | x=0 (anemometers & vanes) |
| | x=1 (north- & east-facing propellers) |
| Where: | θ1 = Avg unit vector dir |
| | σ(θ1) = Std dev dir (Yamartino) |
| | θu = Avg resultant vector dir |
| | σ(θu) = Std dev dir (CSI) |

**73, 74 TIME of max or min options:**

| | |
|---|---|
| 00 | Max/min value only |
| 01 | With Seconds |
| 10 | With Hour-Minute |
| 11 | With Hour-Minute, Seconds |

**75 FORM codes:**

| | |
|---|---|
| 0 | Open form (data beyond limits is included) |
| 1 | Closed form (data beyond limits is excluded) |

**WV LOCation:**

| | |
|---|---|
| 0 | Frequency Distribution |
| xxxx | Weighted Value Loc |

**77 OPTION codes:**

| | |
|---|---|
| xxx1 | Seconds |
| xx1x | Hour-Minute |
| xx2x | Hour-Minute, 2400 at midnight |
| x1xx | Day |
| x2xx | Day, Previous day at midnight |
| 1xxx | Year |
| (0 = no output, e.g., 110 = Day, Hour-Minute) | |

**78 OPTION codes:**

| | |
|---|---|
| 0 | Low resolution |
| 1 | High resolution |

**80 AREA codes:**

| | |
|---|---|
| 1 | Final Storage |
| 3 | Input Storage |

# PROGRAM CONTROL INSTRUCTIONS

(F is fixed data (constant); X, Y, and Z are Input Locations)

| INST | | 01: | 02: | 03: | 04: |
|---|---|---|---|---|---|
| 83 | IF CASE <F | F | CMD† | | |
| 85 | BEG SUBR | SUBR NO (1-9, 79-99) | | | |
| 86 | DO | CMD† | | | |
| 87 | LOOP | DELAY | LOOP CNT | | |
| 88 | IF X <=> Y | X | COMP† | Y | CMD† |
| 89 | IF X <=> F | X | COMP† | F | CMD† |
| 90 | LOOP INDEX | STEP | | | |

| INST | | 01: | 02: | 03: |
|---|---|---|---|---|
| 91 | IF FLAG | FLAG† | CMD† | |
| 92 | IF TIME | T into INT | INT (min) | CMD† |
| 93 | BEGIN CASE | CASE LOC | | |
| 94 | ELSE | | | |
| 95 | END | | | |
| 96 | SERIAL OUT | DEST† | | |
| 98 | SEND CHAR | BAUD/CHAR | | |

## †Option Codes

**83-92 CoMmanD codes:**

| | |
|---|---|
| 0 | Go to end of Pgm Table |
| 1-9, 79-99 | Call Subroutines 1-9, 79-99 |
| 10-19 | Set flag 0-9 high |
| 20-29 | Reset flag 0-9 low |
| 30 | Then Do |
| 31 | Exit Loop if true |
| 32 | Exit Loop if false |
| 41-48 | Set Port high |
| 51-58 | Set Port low |
| 61-68 | Toggle Port } Assumes Card #1 |
| 71-78 | Pulse Port |

**88, 89 COMParison codes:**

| | |
|---|---|
| 1 | = |
| 2 | ≠ |
| 3 | ≥ |
| 4 | < |

**91 FLAG codes:**

| | |
|---|---|
| 1x | Do if flag x is set high |
| 2x | Do if flag x is reset low |

**96 DESTination codes, 2 digits:**

| | |
|---|---|
| 00 | Tape |
| 1x | Printer, ASCII |
| 2x | Printer, binary |
| x | Baud rate digit |
| 30 | SM192/716/CSM1 |
| 31 | Filemark to SM192/716/CSM1 |

**FLAG DESCRIPTIONS:**

| | |
|---|---|
| 0 | Output flag |
| 1-8 | User flags |
| 9 | Intermediate processing flag |

**BAUD RATE DIGITS:**

| | |
|---|---|
| x = 0 | 300 |
| 1 | 1200 |
| 2 | 9600 |
| 3 | 76,800 |

# ERROR CODES

<table>
<tr><td>

1 - I/O Module does not respond
3 - Program Table full
4 - Intermediate Storage full
6 - I/O Module Address not between 1 and 4
8 - CR7 was reset by watch dog timer
9 - Input Storage not allocated
11 - Attempt to allocate unavailable storage
20 - Subroutine encountered before necessary END
21 - END without IF, LOOP, or SUBROUTINE
22 - Missing END, Nonexistent SUBROUTINE
24 - ELSE in SUBROUTINE without IF

</td><td>

25 - ELSE without IF
26 - EXIT LOOP without LOOP
27 - IF CASE without BEGIN CASE
30 - IFs and/or LOOPs nested too deep
31 - SUBROUTINEs nested too deep
40 - Table 2 Execution Interval too short or
      instruction does not exist
68 - Missing or Insufficient P68 after P118
97 - Time out in *D Mode
98 - Uncorrectable tape read errors
99 - Wrong file type

</td></tr>
</table>

## DAY OF YEAR CALENDAR

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| JAN | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 | 22 | 23 | 24 | 25 | 26 | 27 | 28 | 29 | 30 | 31 |
| FEB | 32 | 33 | 34 | 35 | 36 | 37 | 38 | 39 | 40 | 41 | 42 | 43 | 44 | 45 | 46 | 47 | 48 | 49 | 50 | 51 | 52 | 53 | 54 | 55 | 56 | 57 | 58 | 59 | 60 | | |
| MAR | 60 | 61 | 62 | 63 | 64 | 65 | 66 | 67 | 68 | 69 | 70 | 71 | 72 | 73 | 74 | 75 | 76 | 77 | 78 | 79 | 80 | 81 | 82 | 83 | 84 | 85 | 86 | 87 | 88 | 89 | 90 |
| APR | 91 | 92 | 93 | 94 | 95 | 96 | 97 | 98 | 99 | 100 | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 | 110 | 111 | 112 | 113 | 114 | 115 | 116 | 117 | 118 | 119 | 120 | |
| MAY | 121 | 122 | 123 | 124 | 125 | 126 | 127 | 128 | 129 | 130 | 131 | 132 | 133 | 134 | 135 | 136 | 137 | 138 | 139 | 140 | 141 | 142 | 143 | 144 | 145 | 146 | 147 | 148 | 149 | 150 | 151 |
| JUN | 152 | 153 | 154 | 155 | 156 | 157 | 158 | 159 | 160 | 161 | 162 | 163 | 164 | 165 | 166 | 167 | 168 | 169 | 170 | 171 | 172 | 173 | 174 | 175 | 176 | 177 | 178 | 179 | 180 | 181 | |
| JUL | 182 | 183 | 184 | 185 | 186 | 187 | 188 | 189 | 190 | 191 | 192 | 193 | 194 | 195 | 196 | 197 | 198 | 199 | 200 | 201 | 202 | 203 | 204 | 205 | 206 | 207 | 208 | 209 | 210 | 211 | 212 |
| AUG | 213 | 214 | 215 | 216 | 217 | 218 | 219 | 220 | 221 | 222 | 223 | 224 | 225 | 226 | 227 | 228 | 229 | 230 | 231 | 232 | 233 | 234 | 235 | 236 | 237 | 238 | 239 | 240 | 241 | 242 | 243 |
| SEP | 244 | 245 | 246 | 247 | 248 | 249 | 250 | 251 | 252 | 253 | 254 | 255 | 256 | 257 | 258 | 259 | 260 | 261 | 262 | 263 | 264 | 265 | 266 | 267 | 268 | 269 | 270 | 271 | 272 | 273 | |
| OCT | 274 | 275 | 276 | 277 | 278 | 279 | 280 | 281 | 282 | 283 | 284 | 285 | 286 | 287 | 288 | 289 | 290 | 291 | 292 | 293 | 294 | 295 | 296 | 297 | 298 | 299 | 300 | 301 | 302 | 303 | 304 |
| NOV | 305 | 306 | 307 | 308 | 309 | 310 | 311 | 312 | 313 | 314 | 315 | 316 | 317 | 318 | 319 | 320 | 321 | 322 | 323 | 324 | 325 | 326 | 327 | 328 | 329 | 330 | 331 | 332 | 333 | 334 | |
| DEC | 335 | 336 | 337 | 338 | 339 | 340 | 341 | 342 | 343 | 344 | 345 | 346 | 347 | 348 | 349 | 350 | 351 | 352 | 353 | 354 | 355 | 356 | 357 | 358 | 359 | 360 | 361 | 362 | 363 | 364 | 365 |

Add 1 to unshaded values during leap years.