HELIOS PLUS

2287A DATA ACQUISITION FRONT END

System Manual

Volume 2 (Sections 5-9)

P/N 865295 (2 Volume Set)
February 1990

FLUKE ®

TABLE OF CONTENTS
Volume 2

NOTE

See Volume 1 for these sections:

Section 1. Getting Started
Section 2. Specifications
Section 2A Performance Testing
Section 3. Installation and Setup
Section 4. Using the Commands

Section 5. Command Reference

Contents, Volume 2

ALPHABETICAL REFERENCE TO THE COMMANDS

Contents, Volume 2

Section 8. Error Messages

Section 9. Appendices

Section 5
Command Reference

CONTENTS

ALPHABETICAL REFERENCE TO THE COMMANDS

Command	Parameter	Related Keyword(s)	Page
CAL		ON OFF	5-23
CHAN			5-25
COUNT		ON OFF	5-29
DATE$			5-31
DEF ABUF			5-33
DEF BSCAN	FAD		5-35
	XTRIGTYPE	LO HI LOHI HILO	
	FILTERCNT		
	TRIGPOS		
	SCANINTERVAL		
	CAL	ON OFF	
DEF CHAN Alarm Limits			5-45
	LOLO		
	HI		
	LO		
	HYST		
	ALARM		
DEF CHAN Analog Output			5-51
	BIPOLV		
	PVOUT		
	DCOUT		
	UNIPOLV		
DEF CHAN (Channel Function)			5-53
	CHFN	TABLE POLY SQR	
DEF CHAN Counter/Totalizer			5-61
	FREQ		
	TOTAL		

		Related	
Command	Parameter	Keyword(s)	Page
EOL			5-103
FORMAT		BINARY	5-105
		DECIMAL	
		HEX	
		XASCII	
		XBINARY	
		XDECIMAL	
		XHEX	
HOSTTO			5-111
INTERRUPT		ON OFF	5-113
LABEL CHAN	UNITS		5-115
	FORMAT		
	CTRL		
	NUMBER		
LINEFR			5-121
LIST ABUF			5-123
LIST BSCAN			5-125
LIST CHAN			5-129
LIST ERROR			5-149
LIST SBUF			5-153
LIST SCAN			5-155
LIST TABLE			5-157
MODE		COMP TERM	5-161
RESET			5-163
RESET ALL			5-163
RESET CHAN			5-163
RESET ABUF			5-165
RESET SBUF			5-167
SEND ABUF			5-169
SEND BSCAN			5-176
SEND BSCANIPS			5-179
SEND CHAN			5-181
SEND SBUF			5-187
SEND SCANSIZ			5-191

Command	Parameter	Related Keyword(s)	Page
SEND SCANUM			5-193
SEND SINTERVALOVR			5-201
SEND <System Variable>			5-203
	CAL		
	COUNT		
	DATE$		
	EOL		
	FORMAT		
	HOSTTO		
	INTERRUPT		
	LINEFR		
	MEMSIZ		
	MODE		
	STATUS		
	TERM		
	TIME$		
	TIME		
	TUNIT		
	VERSION$		
SEND TRIGCHAN			5-211
SEND TRIGFAD			5-213
SHOW ABUF			5-215
SHOW FIRST ABUF			5-215
SHOW LAST ABUF			5-215
SHOW AGAIN ABUF			5-215
SHOW SBUF			5-219
SHOW FIRST SBUF			5-219
SHOW LAST SBUF			5-219
SHOW AGAIN SBUF			5-219
START BSCAN			5-223
START SCAN	OUTPUT	HOST PRINTER SBUF	5-225
	INTERVAL		
	DELAY		
	PERIOD		
	HEADER		
	FOOTER		

Command	Parameter	Related Keyword(s)	Page
STOP BSCAN			5-231
STOP SCAN			5-233
TERM		ON OFF	5-235
TEST			5-237
TEST CHAN			5-237
TIME			5-239
TIME$			5-241
TUNIT		CELSIUS FAHRENHEIT KELVIN RANKINE	5-243

Section 5
Command Reference

INTRODUCTION

This section provides complete, alphabetically arranged reference information to all available commands for the Front End. The section is divided into three main parts:

- The first several pages describe how to read and use the reference information. This first part discusses notation conventions and how to read the syntax diagrams.

- The second main topic is a discussion of the channel function, channel numbers, and numerical representation. These topics are presented separately because they are global in nature; that is, they apply to virtually all the available commands and are needed as background information.

- The largest part of this section is the reference information. Each command is discussed separately. The commands are arranged alphabetically so needed information can be located quickly.

USING THE COMMAND REFERENCE INFORMATION

The description of a command is often keyed to a specific function. For example, the DEF command is expanded to separate reference pages for each type of measurement or output that can be defined. Relevant uses of DEF can therefore be accessed directly.

Reference information is presented in complete examples. Each item appears in the context of a command line. For ease of expression, command lines are represented in three ways. First, the **Format** of a command gives a general description of the construction of a valid command line. The second type of expression is graphical, showing each type of command line as a syntax diagram. Finally, a textual description uses notation conventions to express the relationship of the various elements.

Notation Conventions

Several conventions are used to distinguish command line keystrokes from the surrounding text. In the following examples, brackets ([ ]) and angle brackets (< >) are used to separate parts of the command line, but are not actual parts of the command line. Do not type these symbols.

<XXX> Angle brackets enclosing all upper-case letters mean "press the XXX key".

Example: <CR>

means to press the CARRIAGE RETURN or ENTER key.

<xxx> Angle brackets enclosing all lower-case letters indicate required information to be entered by the user.

Example:

DEF TABLE(<table number>) = <table definition>

means supply a "table definition" for TABLE(table number).

[xxx] Indicates an optional input statement.

Example: **[,MAX = <volts>]**

means that specifying a maximum expected voltage is optional.

XXX Front End keywords are represented as all upper-case characters to distinguish them from surrounding text. However, keywords can be entered in any combination of upper and lower case characters.

Example: **MODE**

means to type the command "MODE" or "mode" or "Mode".

| Indicates that the user can choose among the options listed.

Example: **MODE = COMP | TERM**

means that the user can enter either MODE = COMP or MODE = TERM.

.. "Double Periods" define an inclusive range.

Example: **CHAN(0..99) = 1**

assigns logic '1' to channels zero through ninety-nine.

Syntax Diagrams

The structure and syntax of the commands are graphically represented in the form of syntax diagrams. These visual portrayals provide an excellent way to learn and reference the various command usages.

Syntax diagrams use a standard symbol for each type of activity. Following is an overview of these symbols:

(WORD) A Word inside an oval is a keyword. In the syntax diagram, keywords will be in upper case. However, they can be entered in any combination of upper and lower case characters.

(( ) This indicates a left parenthesis.

( )) This indicates a right parenthesis.

[channel (s)] A box with lower-case words means that you supply some information. In this case, you would enter channel number(s).

(explanation) Words in parentheses are explanations of some kind. They give information about the nearest block or path.

Syntax Diagram Examples

A syntax diagram represents a critical path for entering a command line. The diagram shows the most direct route, along with any optional paths. To read a syntax diagram, start at the upper left and follow the arrows. Any branches in the diagram define an optional part. Either continue along the critical path, or enter the optional command line parts before returning to the critical path. Figures 5-1 and 5-2 present syntax diagrams for the Front End.

EXAMPLE 1

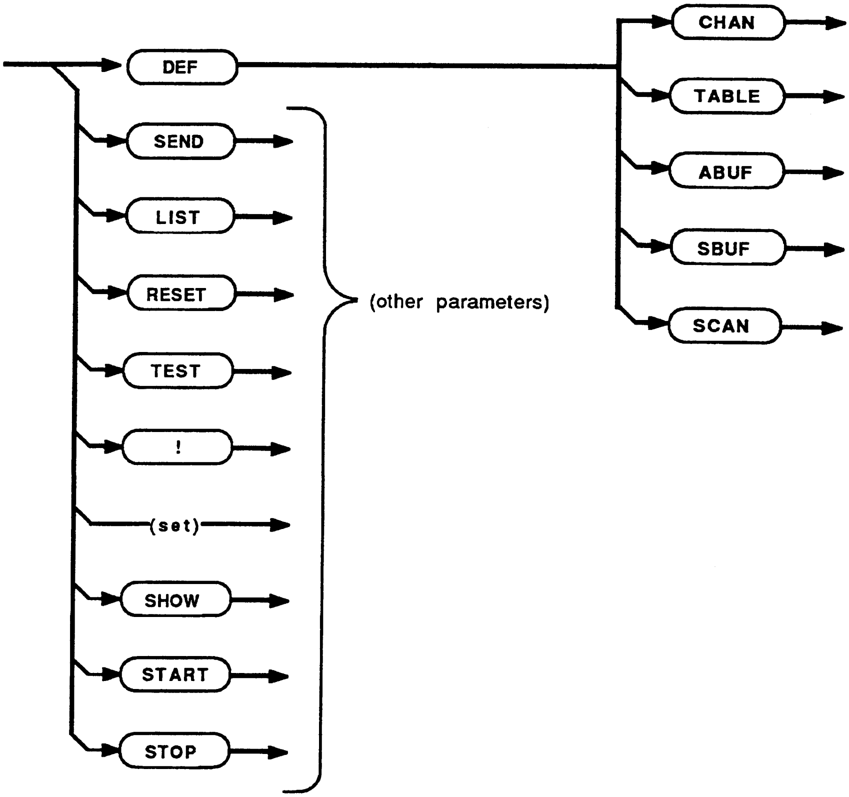


Figure 5-1. Syntax Diagram - Commands

This example shows the highest level of syntax. All command lines start with one of these keywords: DEF, SHOW, START, STOP, SEND, LIST, RESET, TEST, !. "Set" requires no initial command keyword. Some syntax rules are as follows:

- o Keywords are case insensitive, although they are rendered in upper-case characters for emphasis.
- o Spaces may not be inserted between the letters of a keyword, but a space character is required between the keywords of a command.
- o To enhance readability, spaces may be inserted between elements of a command line.

EXAMPLE 2

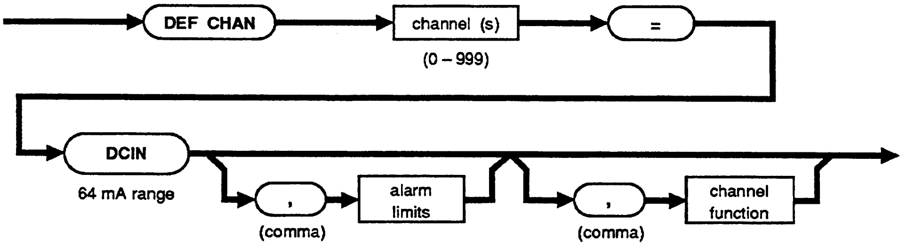


Figure 5-2. Syntax Diagram - DEF

This example expands on syntax for the DEF command. DEF can be used to define a channel, an interpolation table, a scan, an alarm buffer, or a scan group. The example demonstrates some additional rules about syntax diagrams:

- Where indicated, commas must be used as separators within the command string.

- Explanatory information, appearing in parentheses, further defines the boxes or paths. Parentheses can also indicate the range of a parameter. An instance here shows that the 64-mA range is the only range available.

- Optional syntax is shown as a branch from the main (shortest) path. Here, a channel function table can be called.

Based on this syntax, a valid command line would be:

```
DEF CHAN(00..09)=DCIN
```

COMMON SYNTACTIC ELEMENTS

This second part of the command reference section gives needed background information on three topics:

- The Channel Function. Most Front End operations take place via measurement channels. Therefore, the channel function is used with nearly all of the commands. Rather than repeat the necessary information many times, this function is discussed here as a separate topic.

- Channel Numbers. As a part of all channel commands, Channel Numbers are repetitive. A clear understanding of how to represent Channel Numbers is an aid to using the Front End command set efficiently.

- Numeric Representation. This discussion describes the valid ways of expressing numbers when sending commands to the Front End.

The Channel Function

Although optional, the channel function can be appended to any channel definition (DEF CHAN) command. It applies one or more of the following values to specified channels:

- An interpolation table, which is defined in the TABLE statement appended to the DEF CHAN command but separately set up with the table definition command DEF TABLE.

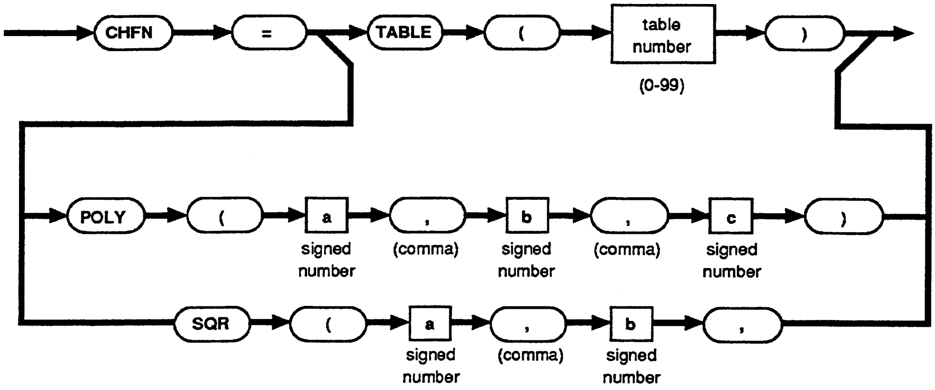
- A polynomial function, which is defined in the POLY(a,b,c) statement appended to the DEF CHAN command.

- A square root function, which is defined in the SQR(a,b) statement appended to the DEF CHAN command.

SYNTAX DIAGRAM

A syntax diagram with this block: channel function

expands to the following complete syntax:



INTERPOLATION TABLE (TABLE)

An interpolation table requires two types of commands:

```
DEF TABLE(1) = 0.004,0/0.020,100/...
DEF CHAN(33) = DCIN, CHFN =TABLE(1)
```

In this example, the first line defines points on an interpolation table (Table 1). The second line applies these points to the defined channel. Channel 33 is defined for direct current measurements. Measurements are filtered through the table. This command makes possible very flexible linearization and result interpolation, so be sure to refer to the DEF TABLE command reference page for details on how to construct useful tables.

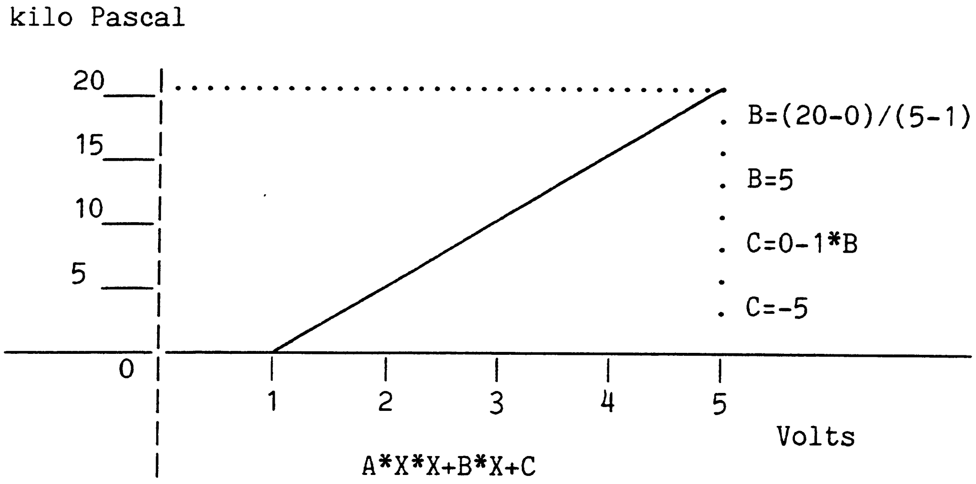
POLYNOMIAL FUNCTION (POLY)

The polynomial function is organized as:

$$\text{Result} = aX^2 + bX + c$$

The coefficients (numbers) a, b and c have to be given by the user; X is the unscaled (measurement) value. The syntax "POLY" (no number between the brackets) indicates that the polynomial function is a local one. This means that the polynomial parameters can be accessed only by the associated channel; this is in contrast with Table definitions where the same Table can be shared by all system channels.

Suppose you want to measure pressure using a transducer whose output is between 1 and 5 volts, corresponding to 0 to 20 kiloPascals. The relationship between pressure and output voltage is linear but the measurement contains an offset. Refer to the figure below for a representation of the problem.



The channel can be defined as follows:

```
DEF CHAN(0) = DVIN, CHFN = POLY(0,5,-5) <CR>
```

SQUARE ROOT FUNCTION (SQR)

The square root function is structured as:

```
Result = a*SQR(x + b)
```

The number coefficients "a" (multiplier) and "b" (offset) can be specified by the user. With no specification, "a" defaults to 1, and "b" defaults to 0. The channel input value is represented by "x".

For example, the channel definition

```
DEF CHAN(19)=DVIN, MAX=4, CHFN=SQR(20.0,0)
```

could be used for a channel connected to a flow transducer that outputs a signal proportional to the difference in pressure across an orifice plate. The rate of flow is proportional to the square root of the differential pressure. The maximum anticipated signal output of 4V corresponds to 40 gallons/minute.

Channel Numbers

Channel Numbers, which can be specified as part of several commands (DEF CHAN, LIST CHAN, SEND CHAN, (set) CHAN, RESET CHAN, DEF SCAN, etc), identify the channels being defined, listed, and so on.

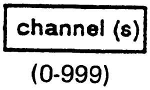
A valid channel identifier can consist of one or more channel numbers separated by commas (,), a range of channels, or multiple ranges separated by commas.

Channel numbers must be represented as positive numbers in the range 0 through 999. Negative channel numbers or numbers outside the allowable range produce an error.

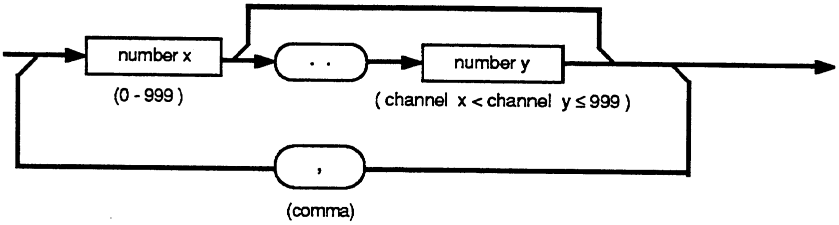
A valid range of channel identifiers is indicated by two numbers separated by double-periods (..). In specifying a channel range, the first channel number must always be less than the second or an error results.

SYNTAX DIAGRAM

This block in a syntax diagram:



expands to:



Examples of valid channel identifiers:

```
CHAN(3)         CHAN(6..12)
CHAN(30,50)     CHAN(9,13..31,33,70..77)
CHAN(77,32,55)  CHAN(1,3..4)
```

Examples of invalid channel identifiers:

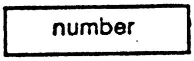
CHAN(55..33)	First channel larger than second.
CHAN(-3)	Negative channel number.
CHAN(1..1000)	Second channel outside valid range.

Numeric Representation

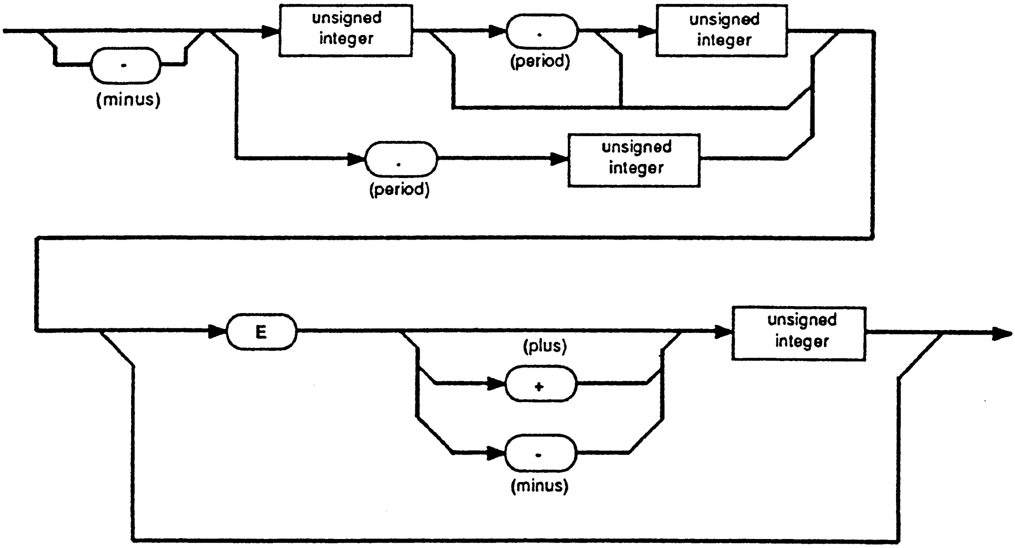
Numbers may be expressed in fixed decimal-point format or in scientific notation.

The legal range is any number between 0.29E-38 and 0.17E+39. Numbers are stored in the memory of the Front End in a 32-bit floating-point format with 8 bits of exponent, 24 bits of mantissa (most significant bit under exponent) and 1 sign bit. No spaces, carriage returns, or line feeds can be embedded in a number.

SYNTAX DIAGRAM

This block in a syntax diagram: 

expands to:



EXAMPLES

Examples of valid numbers are:

```
10
1.
12.E3
.75
-3.0E-01
```

The number

```
1.2E 4
```

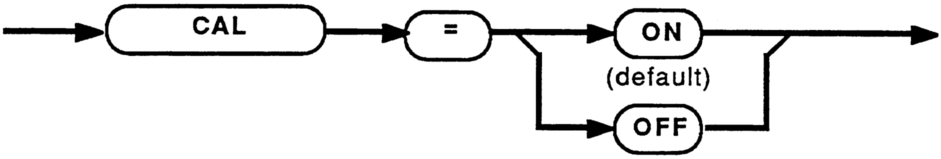
is invalid because it contains a space.

CAL
Set Auto-Calibration System Variable

Format

```
CAL = ON | OFF
```

Syntax Diagram



Description

This variable sets auto-calibration of a/d converters "ON" or "OFF". The default is "ON".

HIGH PERFORMANCE A/D CONVERTER (-161)

With the -161 High Performance A/D Converter, CAL=ON specifies automatic calibration operation at 10-minute intervals. When CAL=OFF, auto-calibration is disabled, allowing for uninterrupted measurement cycles.

The calibration operation runs for about 2 seconds, during which the Front End ignores all other activities. If such timing interruptions interfere with a particular application, set CAL=OFF. Then, using intervals in the application program that better allow for interruption, send CAL=ON, immediately followed by CAL=OFF. This sequence ensures a complete auto-calibration cycle.

If ambient operating temperature is fairly stable (varies less than 2°C/2 hours), you can send CAL=ON CAL=OFF as infrequently as every 2 hours with little loss of accuracy.

FAST A/D CONVERTER (-165)

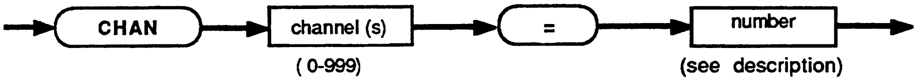
This command does not affect the -165 Fast A/D Converter, which randomly interleaves measurement readings with auto-calibration readings, eliminating any discrete auto-calibration delay. However, for burst scan operations, disabling auto-calibration altogether (CAL=OFF) does yield shorter burst scan intervals.

CHAN
Set Analog Output Channel

Format

CHAN (<channels>) = <analog output value>

Syntax Diagram



Description

This commands assigns an analog output value to the designated channel(s).

The way the assigned value is interpreted depends on the type of analog output (BIPOLV, UNIPOLV, DCOUT, PVOUT). Undefined channels default to unipolar (UNIPOLV) output.

CHAN
Set Analog Output Channel

Examples

To apply a current of 15 mA at the current output of channel 22, enter

```
DEF CHAN(22) = DCOUT
CHAN(22) = 0.015
```

To apply a voltage of -2.9V on the bipolar output of channel 101, enter

```
DEF CHAN(101) = BIPOLV
CHAN(101) = -2.9
```

See Also

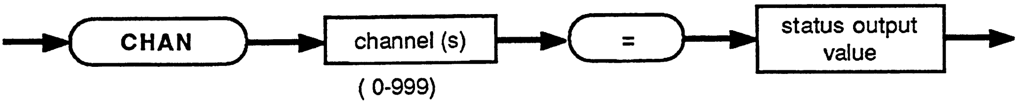
Section 2, Analog Output

CHAN
Set Status Output Channel

Format

```
CHAN(<channels>) = <status output value>
```

Syntax Diagram



Description

This command sets a status output channel either on (logic "1") or off (logic "0"). Any non-zero value is interpreted as a logic "1". The actual output may differ if a channel function is used.

Example

To set 25 bits to logic "1", enter:

```
CHAN(200..224) = 1
```

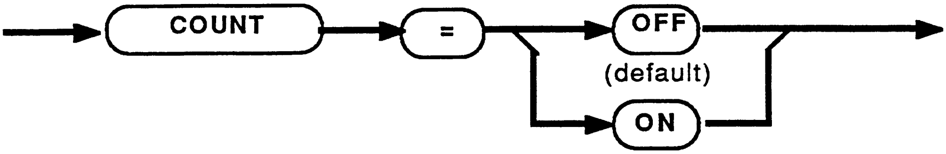
See Also

Section 6f, Status Output

Format

```
COUNT = ON | OFF
```

Syntax Diagram



Description

This variable can be set to "ON" or "OFF". The default is "OFF".

When "ON", the first line returned in response to a SEND CHAN command indicates the number of measurement channels being returned. When COUNT is "OFF" the first line of a returned SEND CHAN command measurement is an actual measurement. COUNT also affects Scan Record output in a similar way. Count is sent only when in the computer mode.

Example

After receiving the commands:

```
MODE = COMP
COUNT = ON
SEND CHAN(0..2)
```

COUNT
Set Count System Variable

Helios Plus could yield a response similar to:

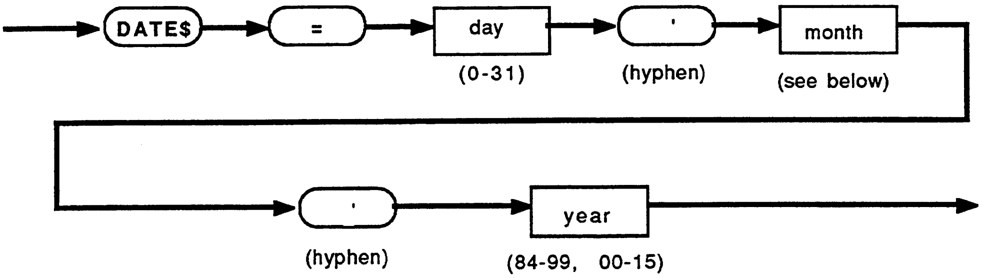
```
3.00000E+00
1.12345E+00
0.00000E+00
-3.87654E+02
```

The first returned value, "3.00000E+00", indicates the number of channel measurements being returned by this SEND command. Being the first number returned distinguishes this value from an actual measurement. If COUNT is set to "OFF", this number of channels is not included in the response.

Format

```
DATE$ = <day>-<month>-<year>
```

Syntax Diagram



Description

This variable represents the system date. Once set, the internal clock keeps track of the date. Leap years are handled automatically.

DATE$
Set Date System Variable

To set the date, enter

```
DATE$ = <day>-<Month>-<year>
```

where <day> is 0-31, <Month> is a month keyword, and <year> is number from "84" to "99" or between "00" and "15". The keyword for each month is:

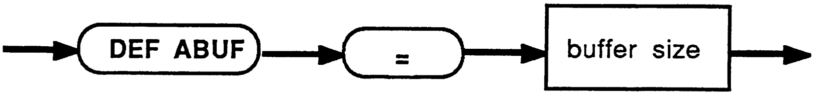
JAN = January	JUL = July
FEB = February	AUG = August
MAR = March	SEP = September
APR = April	OCT = October
MAY = May	NOV = November
JUN = June	DEC = December

DEF ABUF
Define Alarm Buffer

Format

```
DEF ABUF=<buffer size>
```

Syntax Diagram



(0-32767)

Description

This command defines a memory buffer for temporary storage of alarm data. The buffer size indicates the number of alarm records that can be stored. This number is limited by the size of available memory.

The different alarm types stored in the alarm buffer are:

1. System and channel status transitions
2. Scan buffer overruns
3. System hardware errors

Each alarm record stored in the buffer contains the following items:

1. Alarm type
2. Date and time
3. Channel or scan task number (if applicable)
4. Status word (if applicable)
5. A measurement value (if applicable)

DEF ABUF
Define Alarm Buffer

An alarm buffer is a circular 'first in/first out' buffer of a defined size (i.e. a number of records), in which the alarm records are stored as soon as they are available and from which they can be retrieved whenever necessary.

Alarm information sent to the printer port is appended with any UNITS and label information specified for that channel. Refer to LABEL CHAN later in this section for additional information.

Example

An alarm buffer that is capable of holding the last 25 alarm messages should be defined as follows:

```
DEF ABUF=25 <CR>
```

See Also

SEND ABUF
SHOW ABUF
SHOW FIRST|LAST|AGAIN ABUF
RESET ABUF

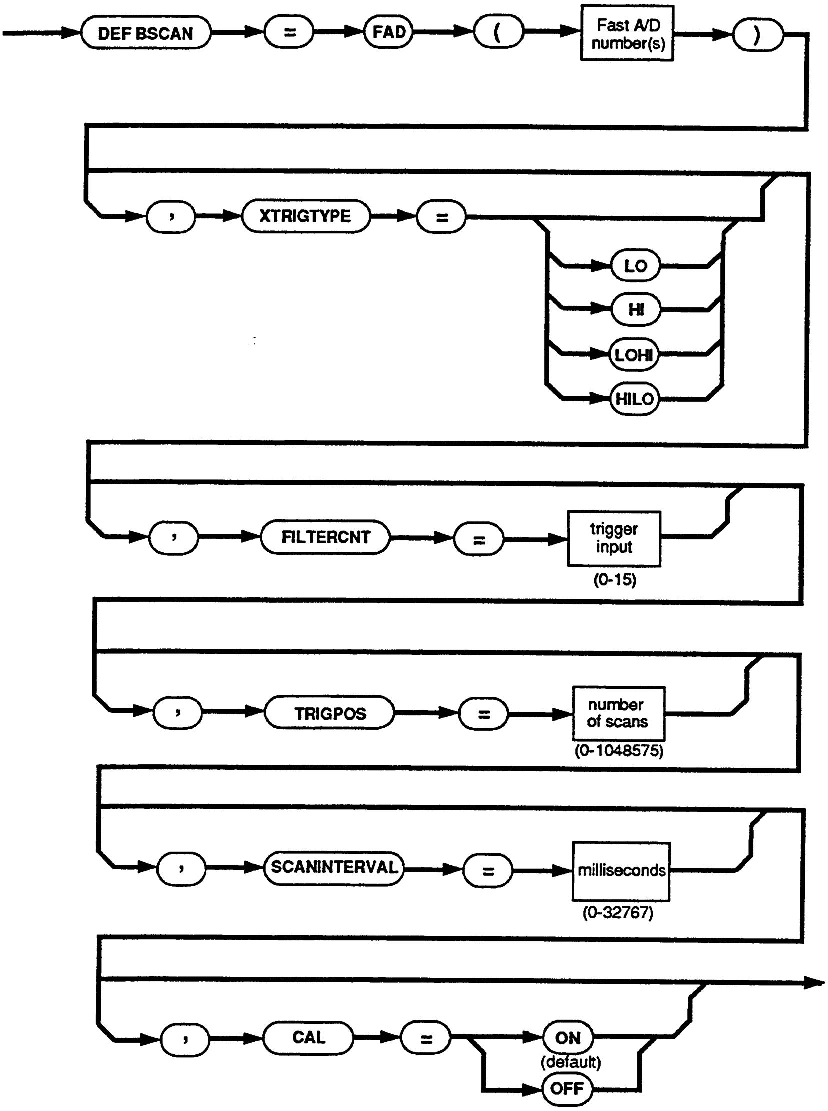
DEF BSCAN
XTRIGTYPE FILTERCNT TRIGPOS SCANINTERVAL CAL
Define Burst Scan Characteristics

Format

```
DEF BSCAN = FAD(<Fast A/D Converter numbers>)
     [,XTRIGTYPE = LO]
     [,XTRIGTYPE = HI]
     [,XTRIGTYPE = LOHI]
     [,XTRIGTYPE = HILO]
     [,FILTERCNT = <trigger inputs>]
     [,TRIGPOS = <number of scans>]
     [,SCANINTERVAL = <milliseconds>]
     [,CAL = ON | OFF]
```

DEF BSCAN
XTRIGTYPE FILTERCNT TRIGPOS SCANINTERVAL CA
Define Burst Scan Characteristics

Syntax Diagram



DEF BSCAN
XTRIGTYPE FILTERCNT TRIGPOS SCANINTERVAL CAL
Define Burst Scan Characteristics

Description

DEF BSCAN defines characteristics of the Burst Scan Mode for a Fast A/D Converter. Multiple Fast A/D Converters can also be specified.

Note that the channels to be burst scanned are defined separately with the DEF CHAN command. When setting up burst scanning, always specify the channels (DEF CHAN) before defining the characteristics (DEF BSCAN).

Burst scanning is started with the START BSCAN command.

Burst scanning is stopped when:

- The STOP BSCAN command is sent, or

- A specified number of scans beyond the trigger occur. This number is set with the TRIGPOS parameter.

Burst scanning operation resembles that of a logic analyzer. Sampling is started with the START BSCAN command, and data is captured until sampling (scanning) is stopped. With triggering, data can be captured in the burst scan buffer before, during, and after an event of interest. The trigger can be provided in two ways:

- External triggering as set with the XTRIGTYPE parameter of the DEF BSCAN command.

- Channel value triggering as set with the HITRIGGER and LOTRIGGER parameters of the DEF CHAN Trigger Value(s) command.

FILTERCNT is used to filter both types of trigger.

DEF BSCAN
XTRIGTYPE FILTERCNT TRIGPOS SCANINTERVAL CA
Define Burst Scan Characteristics

Burst Scan Mode characteristics that can be specified with the DEF BSCAN command are summarized below.

NOTE

External triggering is enabled by positioning a hardware jumper on the Fast A/D Converter PCA. When this pca is shipped from the factory, the jumper is set for normal measurements on the related 1st and 21st channel (0 and 20, 50 and 70, etc.) When the jumper position is changed, these two channels are redefined as external trigger input/output. Refer to the -165 option information in Section 3B of this manual for more information about this jumper. And remember, setting an XTRIGTYPE only specifies the conditions under which the external input generates a trigger input; the jumper must also be correctly positioned for the specified XTRIGTYPE(s) to influence Fast A/D Converter operation.

- XTRIGTYPE

  XTRIGTYPE specifies the condition under which an external hardware input generates a trigger input. The following specifications are possible:

  LO generates a trigger input when the external trigger logic state both goes low during burst scanning and stays low for the number of scans specified by FILTERCNT. If the input stays low for fewer scans than specified by FILTERCNT, no trigger event occurs and the Fast A/D Converter continues burst scanning.

  HI generates a trigger input when the external trigger logic state goes high during burst scanning and stays high for the number of scans specified by FILTERCNT. If the input stays high for fewer scans than specified by FILTERCNT, no trigger event occurs and the Fast A/D Converter continues burst scanning.

  LOHI generates a trigger input when the external trigger logic state goes high during burst scanning and stays high for the number of scans specified by FILTERCNT. If the input stays high for fewer scans than specified by FILTERCNT, no trigger event occurs and the Fast A/D Converter continues burst scanning. If a FILTERCNT of zero is used, LOHI acts as an edge trigger. However, in order to be detected, the high state must be present for the duration of one scan.

HILO generates a trigger input when the external trigger logic state goes low during burst scanning and stays low for the number of scans specified by FILTERCNT. If the input stays low for fewer scans than specified by FILTERCNT, no trigger event occurs and the Fast A/D Converter continues burst scanning. If a FILTERCNT of zero is used, HILO acts as an edge trigger. However, in order to be detected, the low state must be present for the duration of one scan.

Multiple trigger types can be specified. If XTRIGTYPE is not specified, external trigger inputs are not used.

o FILTERCNT

FILTERCNT specifies the number of scans during which the trigger state must be true in order for a trigger event to occur. If FILTERCNT is not specified or is set to 0, a single trigger input causes a trigger event. FILTERCNT also filters Channel Value Triggering (via HITRIGGER and LOTRIGGER in the DEF CHAN Trigger Value(s) command). The maximum FILTERCNT allowed is 15.

Trigger inputs occur when a trigger value is exceeded or an external trigger input is detected. When consecutive scans produce the number of trigger inputs specified by FILTERCNT, a trigger event occurs. Burst scanning then stops after the number of additional scans specified by TRIGPOS. Note that this process is aborted if a trigger input is not found on any of the required consecutive scans; the trigger count is reset whenever a scan does not find a trigger input. Each trigger input is independently filtered.

- TRIGPOS

  TRIGPOS specifies the number of scans to execute after the recognition of a trigger event. A trigger event signifies that at least one trigger input has satisfied its enabling and filtering criteria. If TRIGPOS is not specified or is set to 0, scans are halted after a trigger event is recognized. Maximum allowable TRIGPOS is 1,048,575. With such large TRIGPOS values permitted, the burst scan buffer can be overwritten several times after a trigger event.

- SCANINTERVAL

  The SCANINTERVAL value specifies the time in milliseconds between the start of each scan in Burst Scan Mode. If the Fast A/D Converter requires more time between scan starts than allowed by SCANINTERVAL, an error is reported. If SCANINTERVAL is 0 (or not specified), the minimum required scan time is used. A minimum of 1 ms per channel reading in the scan plus 1 ms between scans is required.

  The maximum SCANINTERVAL allowed is 32,767 (milliseconds).

NOTE

Use the DEF CHAN command to define channels to be burst scanned before using the DEF BSCAN command. If DEF CHAN is sent after the DEF BSCAN command, the Fast A/D Converter may not be able to do scans at the rate specified by SCANINTERVAL.

If SCANINTERVAL is specified while the Fast A/D Converter is operating in Burst Scan Mode, an error is reported (in response to DEF BSCAN) and no change to SCANINTERVAL is made.

o CAL

This system variable controls the frequency with which certain Fast A/D Converter operations are performed. These operations include self-calibration and open thermocouple checking. When CAL is ON (the default), these tasks are performed on a regular basis.

Further, when a HITRIGGER or LOTRIGGER value is assigned to a thermocouple channel, the trigger value must change as the reference junction temperature on the Fast A/D Converter drifts. When CAL is ON, this check is made once each minute for each a/d converter. Trigger values are corrected only when the reference junction has drifted by more than 0.2 degrees Celsius since the last correction.

When CAL is OFF, these tasks are performed only once (when Burst Scan Mode is activated.) Trigger values are only specified when the channel is initially defined and are not corrected to account for subsequent temperature changes. The OFF state thereby allows for shorter scan intervals.

Channel readings made by a Fast A/D Converter (in response to a DEF BSCAN, START BSCAN command sequence) are not automatically checked against limits, and related alarms are not automatically activated. Refer to DEF CHAN Alarm Limits for more information.

DEF BSCAN
XTRIGTYPE FILTERCNT TRIGPOS SCANINTERVAL CAL
Define Burst Scan Characteristics

Example

The following example sets the first channel on Fast A/D Converter 15 as an external trigger input, generates a trigger input at a low logic state, filters for 8 additional scans, then stops scanning 6 scans after the trigger event, and allows for 30 milliseconds between scan starts (with calibration on):

```
DEF BSCAN = FAD(15), XTRIGTYPE = LO, FILTERCNT = 8,&
TRIGPOS = 6, SCANINTERVAL = 30, CAL=ON.
```

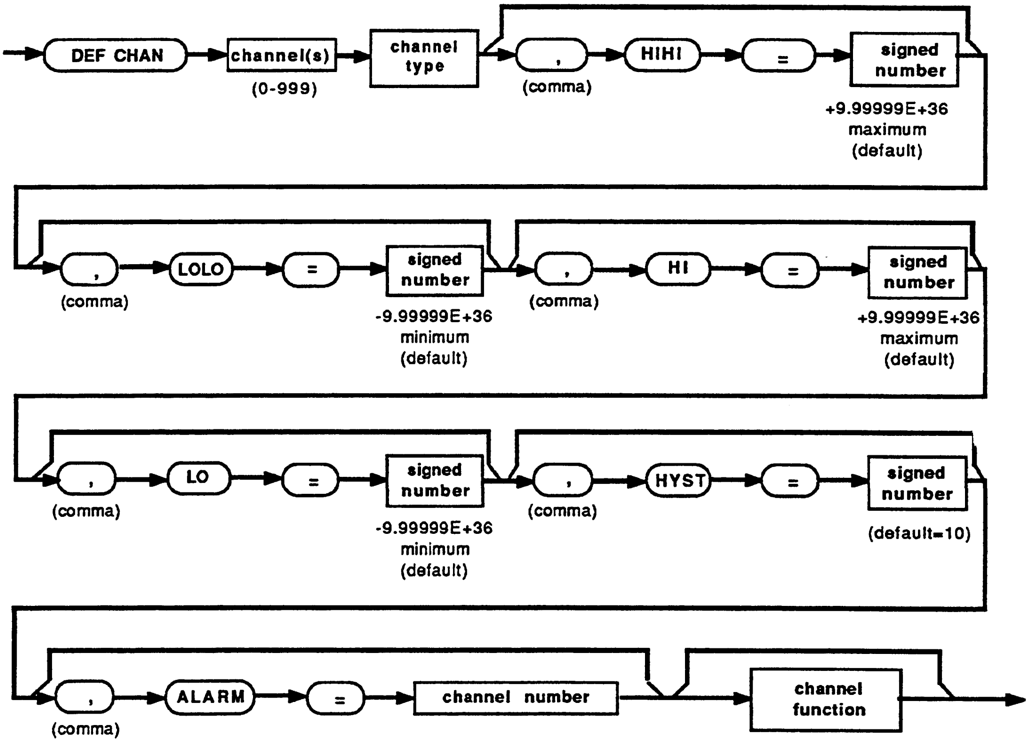
DEF CHAN Alarm Limits
HIHI LOLO HI LO HYST ALARM
Define Channel Limits

Format

```
DEF CHAN(channels) = <channel type>
     [,HIHI = <high-high limit>]
     [,LOLO = <low-low limit>]
     [,HI = <high limit>]
     [,LO = <low limit>]
     [,HYST = <hysteresis>]
     [,ALARM = <alarm channel number>]
```

DEF CHAN Alarm Limits
HIHI LOLO HI LO HYST ALARM
Define Channel Limits

Syntax Diagram



Description

Each channel definition, no matter which type has been specified, can be extended with the definition of a limit set. This limit set consists of up to four limits, a hysteresis value, and an alarm output channel. The alarm output channel, which changes value when any of the four alarms occurs, can be assigned to any one of 40 status output channels.

o HI, LO, HIHI, and LOLO limits

These values are applied to the scaled value of the measurement. They can be used with both analog inputs and analog outputs. Operation with the High Performance A/D and Fast A/D Converters is identical.

Default values are:

HI	+9.99999E+36
LO	-9.99999E+36
HIHI	+9.99999E+36
LOLO	-9.99999E+36

o HYST

The hysteresis value is expressed as a percentage of the difference between the most positive (higher of HIHI or HI) and the most negative (lower of LOLO or LO) limits. This difference may be quite large if any of the four limit values is not specified; default values are used for any limit (HIHI, HI, LOLO, and LO) not otherwise specified.

The hysteresis value should not exceed 100 (100%), with values below 10 being normal. If no hysteresis value is specified, a default of 10% is used.

NOTE

To ensure a predictable hysteresis amount, be sure to specify all four alarm limits.

- ALARM

  The HIHI, LOLO, HI, and LO limit conditions are always checked directly by the Mainframe in response to a SEND CHAN command or when a scan task is getting data from the High Performance A/D Converter or the Fast A/D Converter. These Mainframe limit checks provide the only means to set and report alarms based on limits.

Examples

To measure temperature using a K-type thermocouple, with a temperature range between 100 and 120 degrees, the channel definition could be:

```
DEF CHAN(0) = TC, TYPE = KNBS, HIHI = 120, HI = 115&
LOLO = 100, LO = 105, HYST = 10, ALARM = 100
```

This definition causes status output channel 100 to equal 1 when an alarm limit (HIHI, HI, LOLO, or LO) is tripped. Notice that the hysteresis is 2, defined by 10% of the 20-degree span from LOLO to HIHI. Hysteresis of 2, as applied to each of the four limits, yields:

LIMIT	LIMIT	LIMIT VALUE
TYPE	VALUE	±HYSTERESIS
HIHI	120	118
HI	115	113
LOLO	100	102
LO	105	107

For example, a reading of 121 exceeds both the HIHI limit of 120 and the HI limit of 115; two alarms are set. One alarm is withdrawn when the reading falls below 118 and the second alarm is removed when the reading falls below 113. Similarly, a reading of 99 sets the LO alarm of 105 and the LOLO alarm of 100, with alarms being withdrawn above 102 and 107, respectively. In all cases, status output channel 100 (alarm output) remains set until all alarms are withdrawn.

Alarm Processing During Burst Scanning

For either the -161 High Performance A/D Converter or the -165 Fast A/D Converter (operating in continuous scan mode), alarm processing occurs when either the SEND CHAN command or the DEF SCAN START SCAN command sequence is sent. For the channels being measured by either a/d converter, readings are then checked against limits and alarms can be activated.

DEF CHAN Alarm Limits
HIHI LOLO HI LO HYST ALARM
Define Channel Limits

For the -165 Fast A/D Converter operating in burst scan mode, alarm processing cannot occur in the manner described above. Readings are not automatically checked against limits, and alarms are not automatically activated. However, the following two alternative alarm processing methods are available:

- Alternative A: Using a Scan Task (preferred):

  1. Using the DEF CHAN (Alarm Limits) command, define the desired Fast A/D Converter channel(s) for limit values and alarm output channels.

  2. Ensure that there are no HITRIGGER and LOTRIGGER definitions for these channels. With no definitions, burst scanning continues indefinitely or until STOP BSCAN is sent.

  3. Initiate burst scanning for these channels with the DEF BSCAN START BSCAN command sequence.

  4. Now initiate a scan task by sending the DEF SCAN START SCAN command sequence. Alarm processing will now occur without intervention by the host computer.

- Alternative B: Using External Trigger Outputs

  Use the external trigger output as an indication that the a/d converter has crossed its trigger threshold. This output then goes low within one scan of any channel crossing the HITRIGGER or LOTRIGGER parameter.

See Also

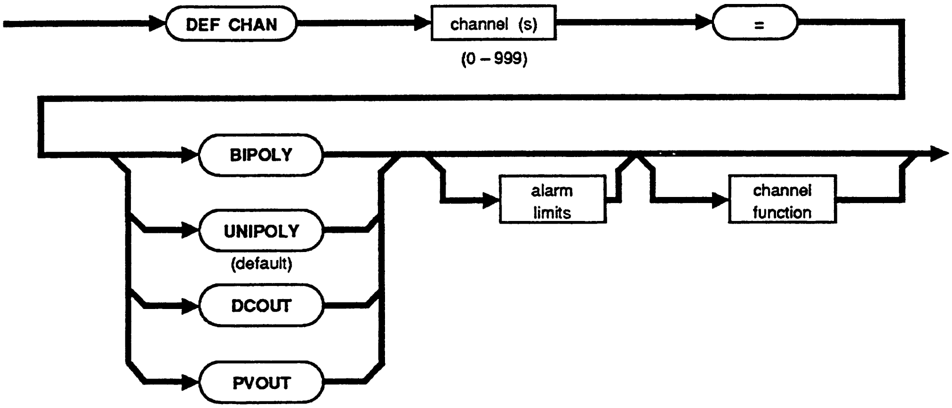
SEND ABUF

DEF CHAN Analog Output
BIPOLV UNIPOLV DCOUT PVOUT
Define Analog Output Channel

Format

```
DEF CHAN(<channels>) = <analog output> [<alarm limits>]
                       [<channel function>]
```

Syntax Diagram



Description

This command defines a type of analog output for designated channel(s). The analog output type is defined in order to allow a value assigned to a channel to be translated properly into an electrical signal. The definition of an analog output channel must correspond to hardware connections or the output will be incorrect.

DEF CHAN Analog Output
BIPOLV UNIPOLV DCOUT PVOUT
Define Analog Output Channel

The analog output card (option -170) provides three sets of output terminals for each of four channels. The output terminals can be configured to any of the following four types of <analog output>:

```
BIPOLV - A bipolar voltage source from -5 to +5V
UNIPOLV - A unipolar voltage source from 0 to +10V
DCOUT - A current source 4 to 20 mA
PVOUT - A current source as Process Variable
        0 to 100%
```

Default is unipolar (UNIPOLV).

For BIPOLV and UNIPOLV, the assigned value corresponds directly with the output voltage. For PVOUT, an assigned value of 0% corresponds with 4 mA and 100% yields 20 mA. When DCOUT is used, current can be specified directly.

Examples

To define channels 100 through 103 to be output channels of the process variable type, enter:

```
DEF CHAN(100..103) = PVOUT<CR>
```

To apply a current of 19.5 mA (0.0195A) on the current output terminals of channel 100, enter

```
DEF CHAN(100) =DCOUT<CR>
CHAN(100) = 0.0195<CR>
```

See Also

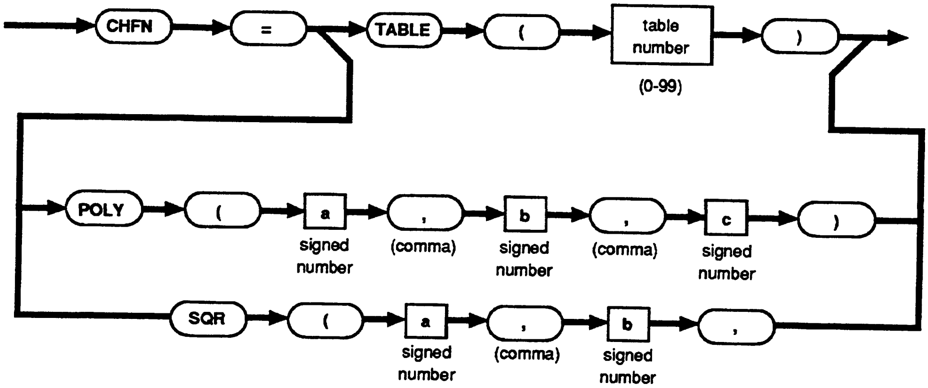
Introduction to this section
Section 6, Measurement Reference: Analog Output

DEF CHAN Channel Function
CHFN
Define Channel Function

Format

```
DEF CHAN(<channels>) = <channel type>
     <alarm limits> [, CHFN = TABLE (<table number>)
                     | CHFN = POLY (<a>,<b>,<c>)
                     | CHFN = SQR (<a>,<b>)]
```

Syntax Diagram



DEF CHAN Channel Function
CHFN
Define Channel Function

Description

The channel function (CHFN) for a channel (or group of channels) can be defined for an interpolation table (TABLE), a polynomial equation (POLY), or a square root function (SQR).

INTERPOLATION TABLE (TABLE)

An interpolation table requires two types of commands:

- o DEF TABLE(<table number>) = <x,y values>

  This command specifies both the table number and a series of input/output value pairs.

- o DEF CHAN(channels) = <channel type>,
  CHFN =TABLE(<table number>)

  This command applies the designated interpolation table to a channel or set of channels.

POLYNOMIAL FUNCTION (POLY)

The polynomial function is organized as:

$$\text{Result} = aX^2 + bX + c$$

The coefficients (numbers) a, b and c have to be given by the user; X is the unscaled (measurement) value. The syntax "POLY" (no number between the brackets) indicates that the polynomial function is a local one. This means that the polynomial parameters can be accessed only by the associated channel; this is in contrast with Table definitions where the same Table can be shared by all system channels.

SQUARE ROOT FUNCTION (SQR)

The square root function is structured as:

```
Result = a*SQR(x + b)
```

The number coefficients "a" (multiplier) and "b" (offset) can be specified by the user. If no such specification is made, "a" defaults to 1 and "b" defaults to 0. The channel input value is represented by "x".

Channel Functions and Trigger Values

Triggering in Burst Scan Mode (Fast A/D Converter) requires some special considerations if a channel function is specified for the same channel(s).

A channel function converts a sensor output into a mathematical form more suitable to the host. If a channel function and HITRIGGER and/or LOTRIGGER values are defined for the same channel(s), trigger values specified are assumed to be in the units seen by the host in response to a SEND CHAN. At this point, the original measurement value has passed through sensor linearizations and the channel function. However, Fast A/D Converter triggering relies on the original measurement value without such conversions. Therefore, the controller must perform the reverse mathematics equivalent to the channel function (and any sensor linearization) to provide the a/d converter with an equivalent raw direct voltage trigger value.

DEF CHAN Channel Function
CHFN
Define Channel Function

In most cases, determination of this reverse channel function is straightforward. Observe the following considerations:

- CHFN = TABLE (0-99)

  When using an interpolation table channel function, table data must be "reversible" in a one-to-one manner. For each possible table output value (including interpolation between data points), there should be only one equivalent sensor input value. In cases where there is more than one possible input value for a given output value, the trigger is set at a value equivalent to the sensor input value appearing first (left-most) in the table definition.

- CHFN = SQR (a, b)

  For the square root channel function, the reverse mathematics required to convert the trigger value into a sensor value cause multiplier "a" to become a divisor. If the multiplier (divisor) is zero, the corresponding implied trigger value is infinite, and error 70 ("Trigger is outside of measurement range") is returned.

  Although the solution of a general square root equation can yield a positive value and an equivalent negative value, the Front End's SQR channel function always calculates the positive answer. Therefore, only positive trigger values may be specified when using the SQR channel function. If a negative trigger value is attempted, error 70 ("Trigger is outside of measurement range") results.

o CHFN = POLY (a, b, c, [,x])

Be aware of the following considerations for quadratic, linear, or constant polynomial channel functions:

Quadratic

When the polynomial is quadratic ("a" is non-zero) there are two possible output values for each possible "x" input (sensor) value. To reverse such a function, the Front End needs to know which of the two possible sensor values corresponds to the region of the polynomial normally used by the host for converting sensor values. Without this knowledge, the Front End could convert the specified trigger value from the host into the wrong, but mathematically valid, trigger voltage.

To avoid this problem, an optional "x" parameter may be supplied when the polynomial is defined. This value should be a typical sensor value near the middle of the expected range of "x" values that the polynomial is designed to convert. Optional parameter "x" is used as a starting (seed) value for an iterative reverse solution of the polynomial.

The "x" parameter is required whenever a quadratic polynomial is used on a channel programmed with a trigger. It is required only if both triggering is desired and the polynomial has a non-zero "a" coefficient.

If "x" is required but not specified, error 69 ("?Can't trigger using POLY() unless a typical X value has been specified") results. In rare cases, the iterative reverse solution to the polynomial may not converge, with error 18 ("?No convergence") resulting.

Linear

When the polynomial is linear ("a" is zero, but "b" is not), the reverse solution to determine the trigger value is straightforward and does not involve iteration. Consequently, no typical "x" value is required.

Constant

When the polynomial is a constant ("a" and "b" are both zero), it is not possible to reverse the function since any input value yields only one output value. This is not at all useful (in either direction) and is unlikely to occur. However, if the host makes a mistake when programming and inadvertently ends up with such a polynomial, the Front End responds with error 18 ("?No convergence") when triggering is attempted.

Refer to the DEF CHAN Trigger Value part of this section for more information.

Examples

INTERPOLATION TABLE

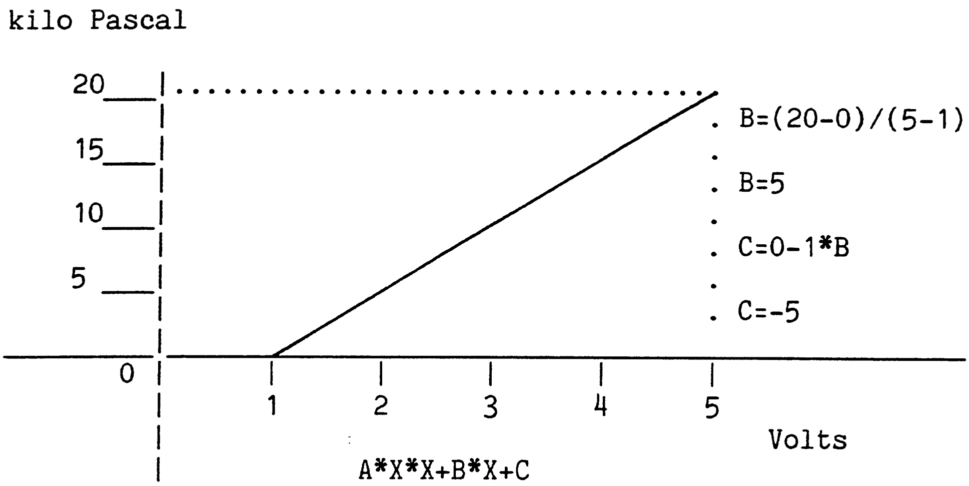
```
DEF TABLE(1) = 0.004,0/0.020,100/...
DEF CHAN(33) = DCIN, CHFN =TABLE(1)
```

In this example, the first line defines points on an interpolation table (Table 1). The second line applies these points to the defined channel. Channel 33 is defined for direct current measurements. Measurements are filtered through the table. This command makes possible very flexible linearization and result interpolation, so be sure to refer to the DEF TABLE command reference page for details on how to construct useful tables.

POLYNOMIAL FUNCTION (POLY)

Suppose you want to measure pressure using a transducer whose output is between 1 and 5 volts, corresponding to 0 to 20 kiloPascals. The relationship between pressure and output voltage is linear but the measurement contains an offset. Refer to the following figure for a representation of the problem. The channel can be defined as follows:

```
DEF CHAN(0) = DVIN, CHFN = POLY(0,5,-5) <CR>
```



SQUARE ROOT FUNCTION (SQR)

The channel definition

```
DEF CHAN(19)=DVIN, MAX=4, CHFN=SQR(20.0,0)
```

could be used for a channel connected to a flow transducer that outputs a signal proportional to the difference in pressure across an orifice plate. The rate of flow is proportional to the square root of the differential pressure. The maximum signal output anticipated is 4V (corresponding to 40 gallons/minute), which would be evaluated as follows:

Reading = 20.0 Square(4.0 - 0.0) = 40

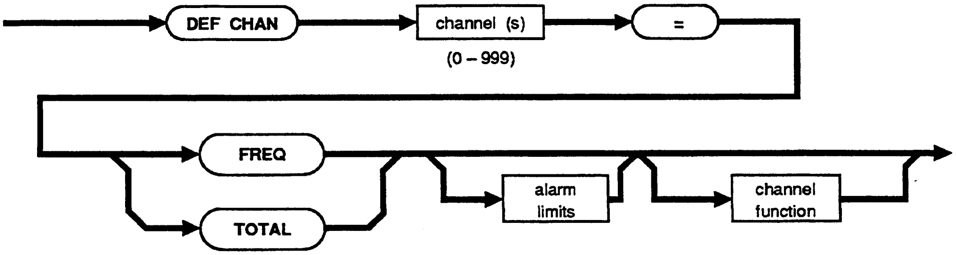
If the orifice plate exhibits an initial zero-flow output, a non-zero correcting value could be added to this definition. For example, SQR(20.0,-0.5) would correct for a 0.5V zero-flow output.

DEF CHAN Counter/Totalizer
FREQ TOTAL
Define Counter/Totalizer Input Channel

Format

```
DEF CHAN(<channels>) = FREQ | TOTAL [<alarm limits>]
                       [<channel function>]
```

Syntax Diagram



Description

This command defines designated channel(s) as Counter (FREQ) or Totalizer (TOTAL) input channel(s).

Switches on the Counter/Totalizer card (option -167), set the board to frequency or totalizing operation. It is not necessary to use this explicit DEF CHAN command. However, to use alarm limits or a channel function, a counter/totalizer input must be defined explicitly. The definition must, of course, correspond to the hardware setup.

DEF CHAN Counter/Totalizer
FREQ TOTAL
Define Counter/Totalizer Input Channel

NOTE

> There are six counter/totalizer channels per option assembly. The switch settings occur in adjacent pairs, which must be defined as the same mode.

Example

To set up the Counter/Totalizer as two frequency channels and four totalizers, enter

```
DEF CHAN(900..901) = FREQ
DEF CHAN(902..905) = TOTAL
```

To utilize a channel function, enter

```
DEF CHAN(90, 91) = FREQ, CHFN = TABLE(4)
```

See Also

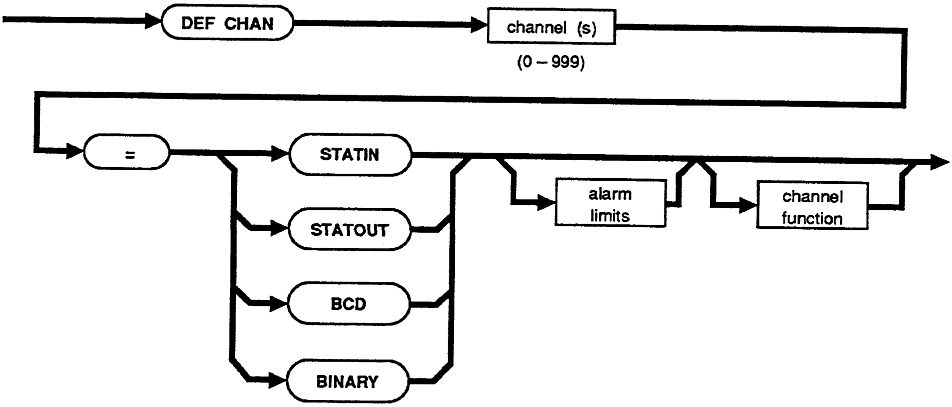
Section 6, Measurement Reference: Frequency Measurement and Totalizing Measurement

DEF CHAN Digital I/O
STATIN STATOUT BCD BINARY
Define Digital I/O Channel

Format

```
DEF CHAN(<channels>) = <digital I/O type>
[<alarm limits>] [<channel function>]
```

Syntax Diagram



Description

This command defines designated channel(s) as a type of digital I/O. The Digital I/O card (option -168) can be used as either an input or output card. This is determined by the type of connector that is used with it.

DEF CHAN Digital I/O
STATIN STATOUT BCD BINARY
Define Digital I/O Channel

Channel numbers and types are determined by connectors and jumpers on the assembly, so there is usually no need for explicit channel definition statements. However, to use a channel function or alarm limits on a digital I/O channel, it must be defined explicitly. The definition must, of course, correspond to the hardware setup.

The following I/O types are allowed:

STATIN	One-bit status input
STATOUT	One-bit status output
BCD	Five-digit binary-coded decimal input
BINARY	Seventeen-bit binary input

Examples

To define channel 120 as BINARY input, enter

```
DEF CHAN(120) = BINARY
```

To apply a table function, enter

```
DEF CHAN(120) = BINARY, CHFN = TABLE(11)
```

See Also

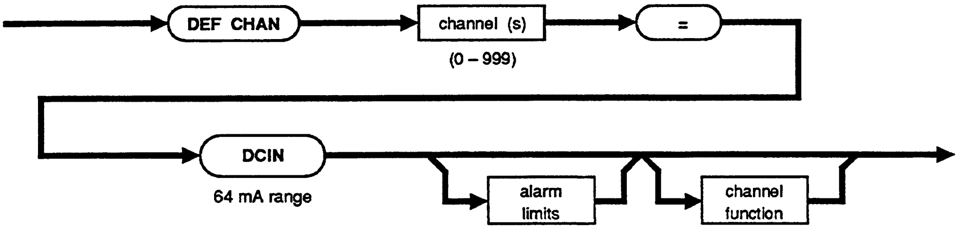
Section 6, Measurement Reference: Digital/Status Input, Status Output

DEF CHAN Direct Current Input DCIN
Define Direct Current Input Channel

Format

```
DEF CHAN(<channels>) = DCIN [<alarm limits>]&
[<channel function>]
```

Syntax Diagram



Description

This command defines channel(s) of analog input for direct current input measurements.

Direct current measurements can be made with either of two sets of hardware. If the -161 High Performance A/D Converter is used, measurements are made on the 64 mA range, using inputs through the -171 Current Input Connector attached to the -162 Thermocouple/DC Volts Scanner.

DEF CHAN Direct Current Input DCIN

Define Direct Current Input Channel

If the Fast A/D Converter is used, inputs are attached to a modified -176 Voltage Input Connector attached directly to the a/d converter. The connector is modified with shunt resistors across the inputs. This procedure is fully described in Section 6B.

Note that if Burst Scan Mode is used with the Fast A/D Converter, direct current input channels cannot be changed during operation. If Continuous Scan Mode is used with this converter, channels can be changed during operation.

Example

To define channels 0 through 5, and channels 8 and 9 as direct current input, enter

```
DEF CHAN(0..5,8,9) = DCIN
```

See Also

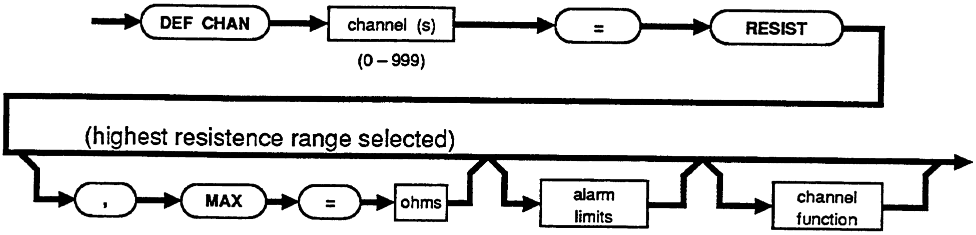
Section 6, Measurement Reference: Current Measurement

DEF CHAN Resistance Input
RESIST
Define Resistance Input Channel

Format

```
DEF CHAN(<channels>) = RESIST[, MAX = <ohms>]
[<alarm limits>] [<channel function>]
```

Syntax Diagram



Description

This command defines designated channel(s) as resistance input channels.

Resistance measurements can be made more accurately by specifying the range of expected measurement values with the MAX parameter. To do this, set MAX to the highest expected measurement value, in ohms. Lower ranges yield higher measurement accuracy. If MAX is not set, Helios Plus defaults to the highest range.

If Burst Scan Mode is used with the Fast A/D Converter, resistance input channels cannot be changed during operation. If Continuous Scan Mode is used with this converter, channels can be changed during operation.

Three hardware configurations are possible:

Configuration A
- -161 High Performance A/D Converter
- -163 RTD/Resistance Scanner
- -177 RTD/Resistance Input Connector

Configuration B
- -161 High Performance A/D Converter
- -162 Thermocouple/DC Volts Scanner
- -175 or -176 Input Connector
- -164 Transducer Excitation Module
- -174 Transducer Excitation Connector

Configuration C
- -165 Fast A/D Converter
- -175 or -176 Input Connector
- -164 Transducer Excitation Module
- -174 Transducer Excitation Connector

Configuration A provides three ranges: 256, 2048, and 64000 ohms. Configurations B and C provide two ranges: 64 and 512 ohms. For all configurations, lower ranges yield higher resolution.

Examples

```
DEF CHAN(803) = RESIST
DEF CHAN(30..37) = RESIST, MAX = 300
```

See Also

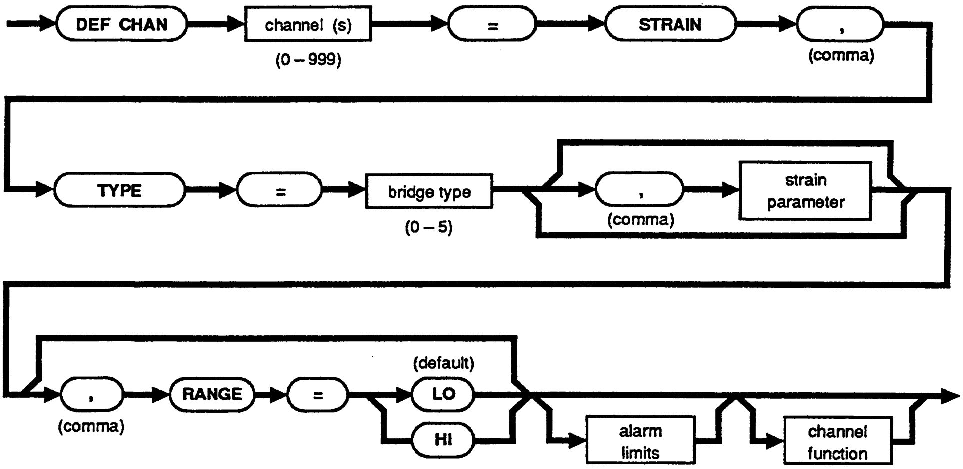
Section 6, Measurement Reference: Resistance Measurement

DEF CHAN Strain Input
STRAIN
Define Strain Input Channel

Format

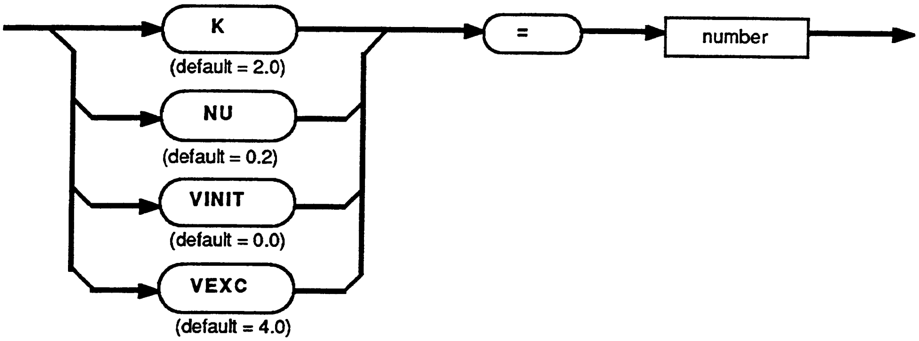
```
DEF CHAN(<channels>) = STRAIN, TYPE = <bridge type>
[,<strain parameter>][,RANGE = HI|LO]
[<alarm limits>] [<channel function>]
```

Syntax Diagram



DEF CHAN Strain Input
STRAIN
Define Strain Input Channel

Strain Parameter:

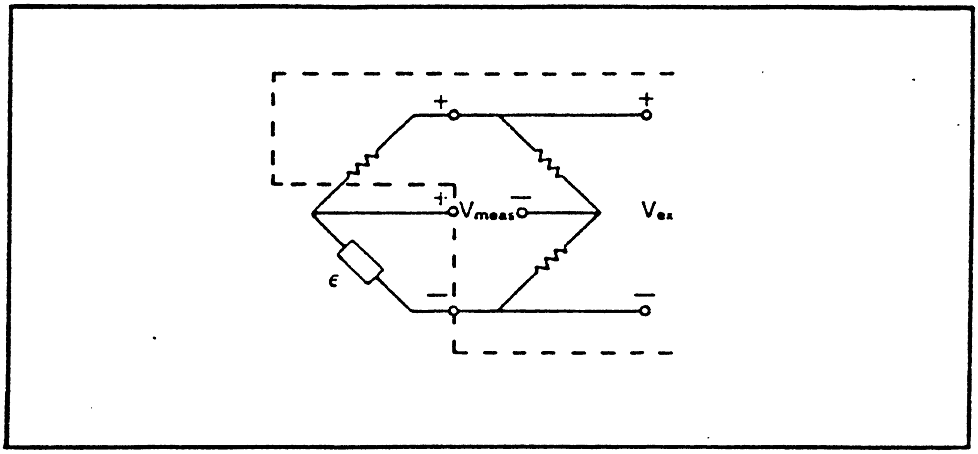


Description

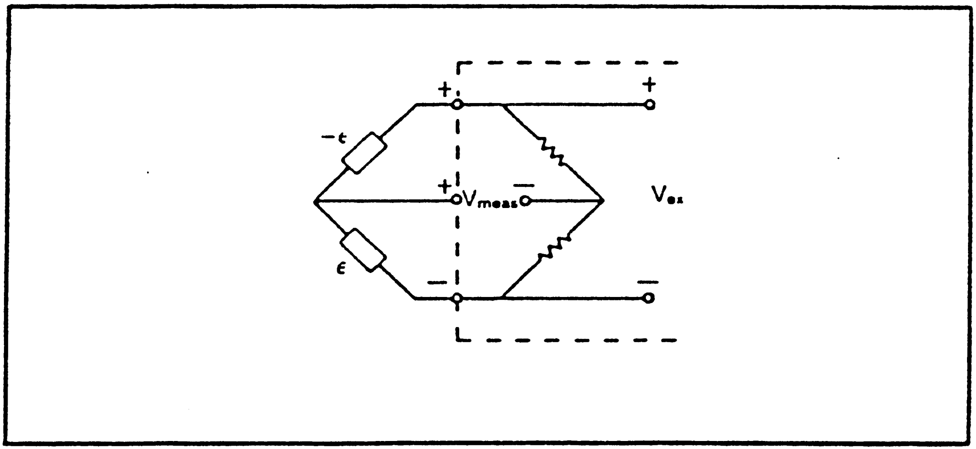
This command defines specified analog input channels as strain gauge measurement channels.

All readings are automatically converted into units of microstrain.

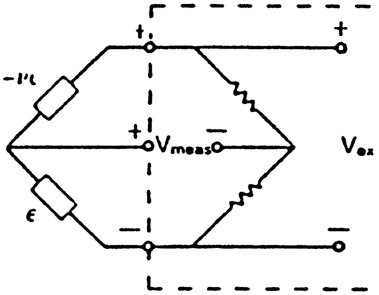
The "TYPE = <bridge type>" expression defines the bridge configuration to be used. Six bridge types, illustrated below, are available.



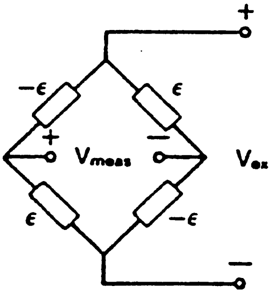
Type 0. Quarter Bridge



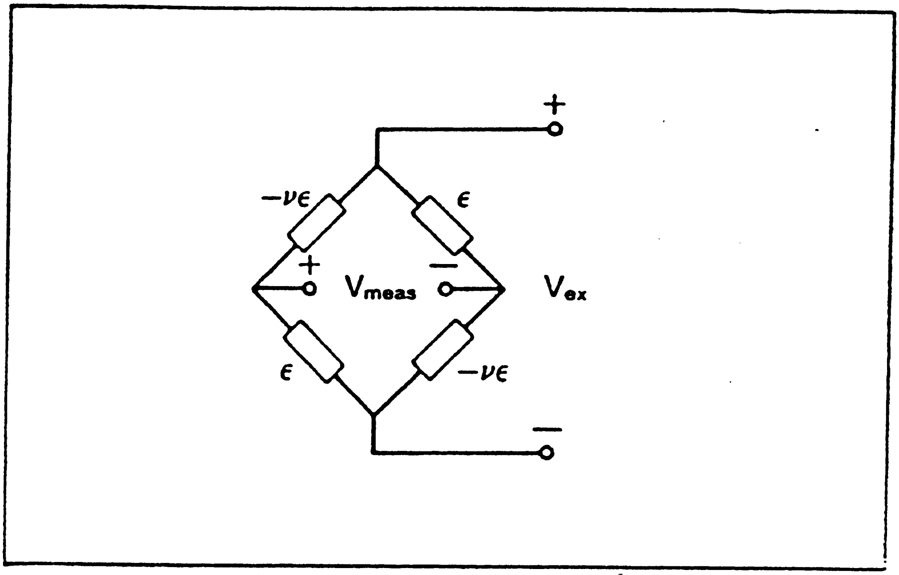
Type 1. Half Bridge with Equal and Opposite Strains



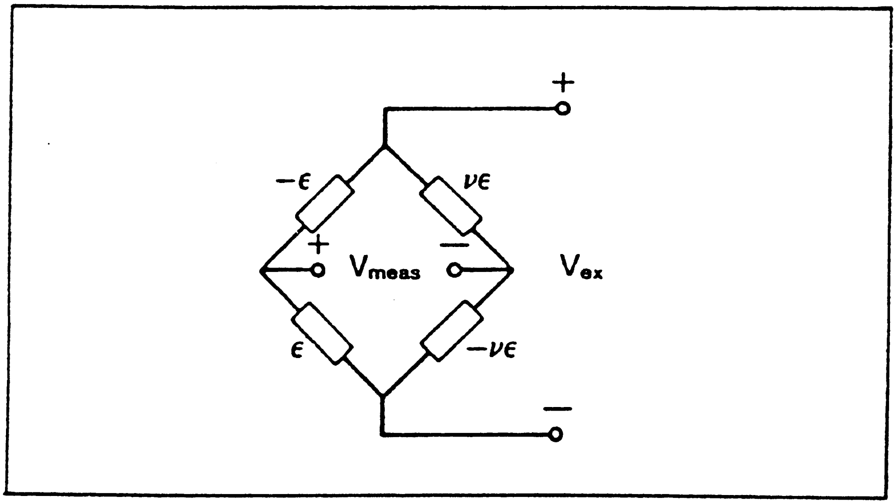
Type 2. Half Bridge with One Principle and One Poisson Strain



Type 3. Full bridge with Two Pairs of Equal and Opposite Strains



Type 4. Full Bridge with Two Principle and Two Poisson Strains



Type 5. Full Bridge with Two Principle and Two Poisson Strains

DEF CHAN Strain Input
STRAIN
Define Strain Input Channel

The value of one to four strain parameters can be set per channel. If a strain parameter value is not specified, a default value is assigned.

When a strain parameter is changed on a strain channel, a subsequent redefinition of that channel will use the same value, unless the constant in question is redefined or a RESET command is executed.

HI or LO ranges are available. If no range is specified, "range" defaults to LO, allowing the most sensitive voltage measurement. If the output of the strain measurement bridge is expected to exceed 60mV, then HI range must be specified to prevent saturation.

Example

To set up channel 0 to measure strain with bridge type 1 (half bridge with equal and opposite strains) and an unstrained voltage of 100 microvolts, enter

```
DEF CHAN(0) = STRAIN, TYPE = 1, VINIT = 0.0001
```

See Also

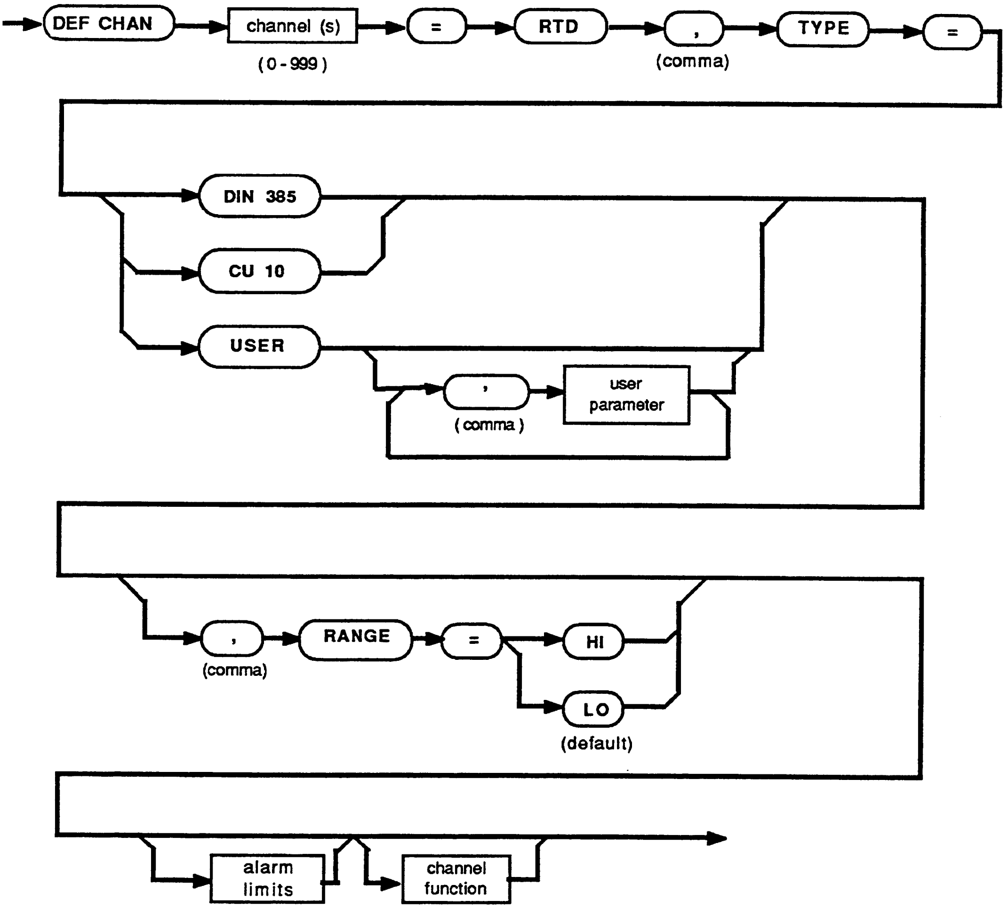
Section 6, Measurement Reference: Strain Measurement

Define RTD Input Channel

Format

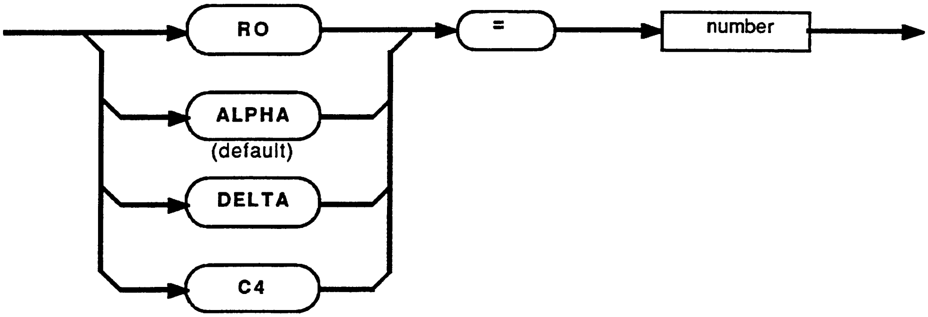
```
DEF CHAN(<channels>) = RTD, TYPE = <rtd type>
[<alarm limits>] [<channel function>]
```

Syntax Diagram



DEF CHAN Temperature Input, RTD
RTD
Define RTD Input Channel

User Parameter:



Description

This command defines designated analog input channel(s) as Resistance Temperature Detector (RTD) measurement channels.

Measurements are in the units specified by the temperature unit selection (TUNIT) statement.

The "TYPE =" expression specifies the kind of RTD being used. An RTD type must always be designated. Two standard types are offered:

- DIN385 for 100 ohms platinum
- CU10 for 10 ohms copper

In addition, a user-selected type, other than those listed above, can be specified. With the user type, four RTD constants must be specified. If a value is not assigned to an RTD constant, the Front End assigns a default value. RTD constants and defaults are:

RO	=	100.0
Alpha	=	3.850E-3
Delta	=	1.45
C4	=	1.19619E-13

When a constant is changed on an RTD channel, a subsequent redefinition of that channel will use the same value, unless the constant in question is redefined or a RESET command is executed.

If you are using a -161 a/d converter with a -163 scanner, two ranges can be selected. But range selection is only significant if the RTD Scanner is used with a 100 ohm platinum RTD. In the LO range, the maximum temperature is limited to 425 degrees Celsius. In the HI range, the maximum becomes 600 degrees Celsius, but the resolution is reduced.

If you are using either the -165 a/d converter or the -161 a/d converter with -164 module, only one range is available. With either a/d converter, excitation current (1 mA) is provided by a Transducer Excitation Module, and the resulting voltage is computed (ohms * 10E-3.) See Section 6H for detailed information on making RTD temperature measurements with either hardware configuration.

If Burst Scan Mode is used with the Fast A/D Converter, RTD temperature input channels cannot be changed during operation. If Continuous Scan Mode is used with this converter, channels can be changed during operation.

RTD

Define RTD Input Channel

Examples

To define channel 30 to measure a DIN 43760 100-ohm platinum RTD, enter

```
DEF CHAN(30) = RTD, TYPE = DIN385
```

To define channels 31 through 35 to measure 100-ohm platinum RTD, with user-defined constants for temperatures around 500 degrees Celsius, using an RTD Scanner, enter

```
DEF CHAN(31..35) = RTD, TYPE = USER,
                   ALPHA = 3.92E-3, RANGE = HI
```

See Also

Section 6, Measurement Reference:
Temperature Measurement Using RTDs

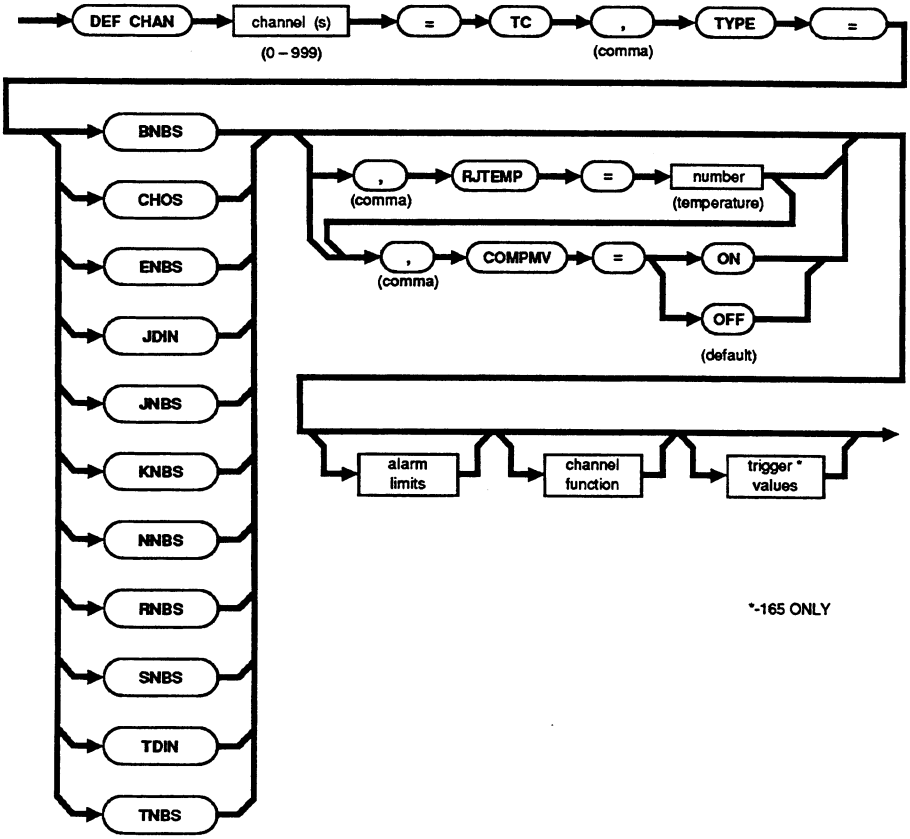
DEF CHAN Temperature Input, TC

Define Thermocouple Input Channel

Format

```
DEF CHAN(<channels>) = TC, TYPE = <tc type>
[, RJTEMP = <reference junction temperature>]
[, COMPMV = ON|OFF] [<alarm limits>] [<channel function>]
```

Syntax Diagram



*-165 ONLY

DEF CHAN Temperature Input, TC
TC
Define Thermocouple Input Channel

Description

This command defines designated analog input channel(s) as thermocouple measurement channels.

The expression, "TYPE = <tc type>", indicates the type of thermocouple to be used. Eleven types of thermocouples are supported:

TYPE	DESCRIPTION
BNBS	B Thermocouple, NBS
CHOS	C Thermocouple, Hoskins Wo-5% Re vs Wo-26% Re
ENBS	E Thermocouple, NBS
JDIN	J Thermocouple, DIN
JNBS	J Thermocouple, NBS
KNBS	K Thermocouple, NBS
NNBS	N Thermocouple, NBS
RNBS	R Thermocouple, NBS
SNBS	S Thermocouple, NBS
TDIN	T Thermocouple, DIN
TNBS	T Thermocouple, NBS

Normally, thermocouples are measured with the Thermocouple/DC Volts Scanner (option -162) combined with the Isothermal Input Connector (option -175).

Thermocouple temperature measurements can be made with either the -165 Fast A/D Converter or the -161 High Performance A/D Converter. Refer to Section 6J for detailed information about both hardware configurations.

Note that if Burst Scan Mode is used with the Fast A/D Converter, thermocouple temperature input channels cannot be changed during operation. If Continuous Scan Mode is used with this converter, channels can be changed during operation.

Thermocouples may be terminated in an external reference junction, from which copper wires extend to a Voltage Input Connector (option -175, -176, or -160). If an external reference junction is used, specify the external reference junction temperature using the RJTEMP parameter. When specifying the reference junction temperature, Helios Plus assumes that the temperature is specified in the temperature units of the TUNITS system variable; so care should be taken to set TUNITS to the desired value before defining the channel.

Readings are usually returned in temperature units. However, compensated millivolts (COMPMV) can also be selected on channels where the user does not want built in linearization.

Examples

To define channels 10 through 15 for J (NBS) type thermocouple measurement with temperature to be returned in degrees C, enter

```
TUNIT = CELSIUS
DEF CHAN(10..15) = TC, TYPE =JNBS
```

To define channels 40 through 59 as T (DIN) type thermocouple inputs, with an external reference junction temperature of 23.4 degrees Celsius, enter

```
DEF CHAN(40..59) = TC, TYPE = TDIN, RJTEMP = 23.4
```

DEF CHAN Temperature Input, TC
TC
Define Thermocouple Input Channel

See Also

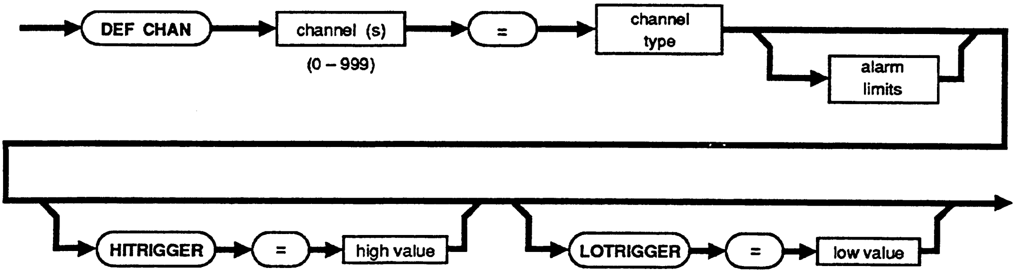
Section 6, Measurement Reference: Temperature Measurement Using Thermocouples

DEF CHAN Trigger Value
LOTRIGGER HITRIGGER
Define Trigger Value(s)

Format

```
    DEF CHAN(<channels>) = <channel type>[<alarm limits>]
                           [,LOTRIGGER=<low trigger value>]
                           [,HITRIGGER=<high trigger value>]
```

Syntax Diagram



DEF CHAN Trigger Value
LOTRIGGER HITRIGGER
Define Trigger Value(s)

Description

A Fast A/D Converter that is in Burst Scan Mode can generate a trigger input when the channel reading passes specified high and/or low trigger values. This type of trigger input can be used to modify the burst scanning activity of the Fast A/D. New trigger values are used immediately if the Fast A/D is in Burst Scan Mode. If the Fast A/D is in Continuous Scan Mode, triggering is disabled, and new trigger values are used the next time Burst Scan Mode is activated.

NOTE

Trigger values do not interact with alarm limit values. Detection of a trigger input (as defined by this command) by the Fast A/D does not result in any alarm processing (as defined with the DEF CHAN Alarm Limits command). For alarm processing to take place, the channel value must be either measured by a SEND CHAN command or logged by a scan task.

Refer to the DEF CHAN Alarm Limits subsection for a recommendation on initiating alarm processing during burst scan operations.

Values defined with HITRIGGER and LOTRIGGER must be within the range accepted by the channel(s) mentioned in the same command string. An error is returned if this rule is not observed.

Ranges for each function are defined in the Helios Plus Specifications. Based on this range information, some HITRIGGER/LOTRIGGER examples would be:

- A KNBS thermocouple, which Helios Plus expects to measure in degrees C., cannot have a HITRIGGER above +1350 or a LOTRIGGER lower than -273.

- A dc voltage input HITRIGGER value cannot exceed the channel's range value. If the MAX parameter is used in the dc voltage channel definition, the lowest range acceptable to the MAX value is used. Therefore, if MAX = 0.8, 8V is the range and HITRIGGER cannot exceed 8.0. If MAX = 0.4, the range is 512 mV and HITRIGGER cannot exceed 0.512. If MAX is not used, the highest dc voltage range is used.

Triggering and Channel Functions

Although HITRIGGER/LOTRIGGER and channel functions (TABLE, POLY, or SQR) can be defined for the same channel, the channel function must follow some special rules so that a value expected by the trigger definition remains available. Mathematical reversals of the channel function are used for this purpose and are fully described in the DEF CHAN (Channel Function) area of this section.

Triggering and Thermocouples

When a trigger is assigned to a thermocouple channel, the raw direct volts trigger value used by the Fast A/D Converter may need to change as the temperature of the isothermal block on the a/d drifts. Refer to the DEF BSCAN area in this section for a discussion of maintaining accurate direct voltage trigger values for thermocouples during such temperature variations.

DEF CHAN Trigger Value
LOTRIGGER HITRIGGER
Define Trigger Value(s)

Trigger Accuracy

The actual trigger point may vary somewhat from the HITRIGGER and LOTRIGGER values. This mathematical variation is function and range dependent, as follows:

FUNCTION	RANGE	TRIGGER ACCURACY
TC	all	±0.2 °C
RTD	all	±0.05°C
DVIN	64 mV	±1.0 uV
	512 mV	±5.0 uV
	8V	±10 uV
	10.5V	±1 mV
DCIN	64 mA	±1.0 uA
RESIST	64 ohm	±2 milliohms
	256 ohm	±3 milliohms
	512 ohm	±6 milliohms
STRAIN	all	±1 micro strain

Example

The following example defines a trigger if the temperature falls below 100.7 or rises above 123.5 degree C.

```
TUNIT = CELSIUS
DEF CHAN(10) = TC, TYPE = KNBS, LOTRIGGER = 100.7,&
HITRIGGER = 123.5
```

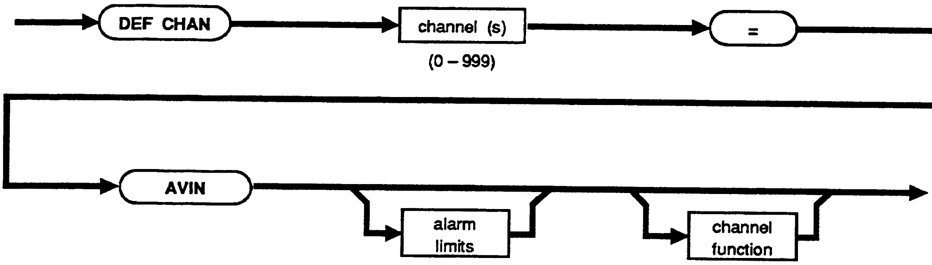
Note that actual trigger points may range from 100.5 to 100.9 for LOTRIGGER and from 123.3 to 123.7 for HITRIGGER.

DEF CHAN Voltage Input, Alternating
AVIN
Define Alternating Voltage Input Channel

Format

```
DEF CHAN(<channels>) = AVIN [<alarm limits>]
[<channel function>]
```

Syntax Diagram



DEF CHAN Voltage Input, Alternating
AVIN
Define Alternating Voltage Input Channel

Description

NOTE

> Alternating voltage inputs cannot be measured with the -165 Fast A/D Converter. Error 3 (Incompatible Channel) is returned if such an attempt is made.

This command defines designated analog input channel(s) as alternating voltage inputs. Measurement results are in volts. Alternating voltage input is selected by entering the keyword AVIN on the define channel(s) (DEF CHAN) command line. Measurements are in volts. 250V (rms) is the only range.

Although each analog input module accommodates 20 channels, only the first 10 of each block can be used for alternating voltage input. This means that the corresponding channel numbers must fall with a decade that begins with an even multiple of 10 (for example: 0..9, 20..29, 40..49, etc.).

Example

```
DEF CHAN(40..49, 60..69) = AVIN
```

See Also

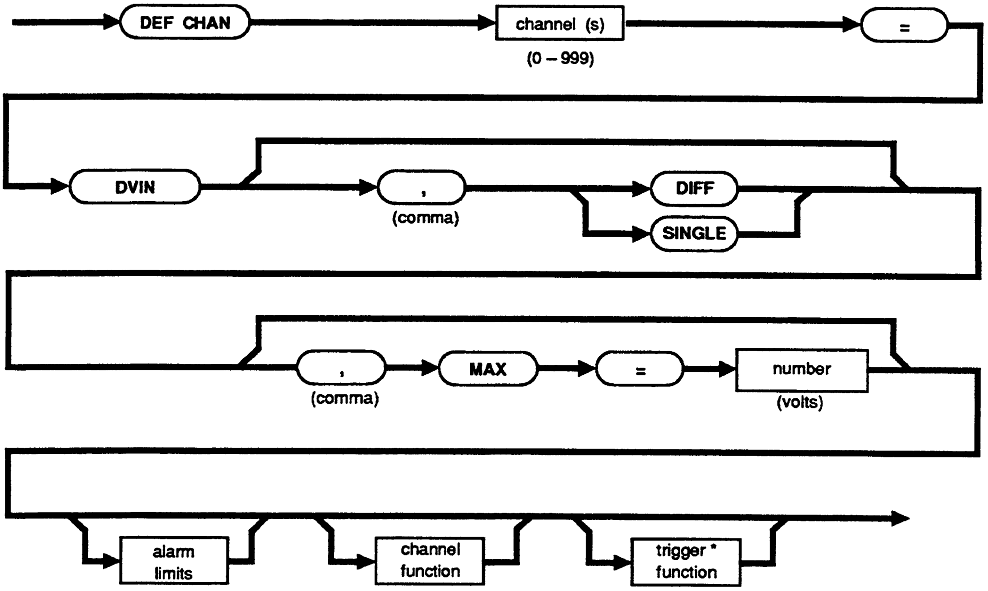
Section 6, Measurement Reference: Voltage Measurement

DEF CHAN Voltage Input, Direct
DVIN
Define Direct Voltage Input Channel

Format

```
DEF CHAN(<channels>) = DVIN [, DIFF | SINGLE]&
[,MAX = <volts>] [<alarm limits>] [<channel function>]
```

Syntax Diagram



* -165 ONLY

DEF CHAN Voltage Input, Direct
DVIN
Define Direct Voltage Input Channel

Description

This command defines designated analog input channel(s) as direct voltage inputs. Measurement results are in volts. Direct voltage input is selected by entering the keyword DVIN on the define channel(s) (DEF CHAN) command line. Ranges available for the Fast A/D Converter include: 64 mV, 512 mV, 8V, and 10.5V (or 64V for the High Performance A/D Converter.)

When measuring direct voltage (DVIN), more accurate measurements can be made by specifying the range of expected measurement values with the MAX parameter. To do this, set MAX to the highest expected measurement value, in volts. Lower ranges yield higher measurement accuracy. If MAX is not set, Helios Plus defaults to the highest range (64V for the High Performance A/D Converter, 10.5V for the Fast A/D Converter).

The DIFF and SINGLE parameters can be used only for channels associated with a Fast A/D Converter. If DIFF and SINGLE are not specified for such channels, a differential measurement is performed. A mixture of differential and single-ended channels can be associated with a single Fast A/D Converter.

o DIFF is specified for a differential measurement of a pair of inputs. A pair of inputs that are to be measured differentially must be connected on input channels numbered n and n+20, where n is between 0 and 19. For example, with the Fast A/D Converter address block 100..139, the differential pairs could range from 100/120 to 119/139. If either DIFF channel is specified in the DEF CHAN command, that channel is measured differentially.

- o SINGLE is specified for a single-ended measurement. The a/d converter measures the voltage difference between an input and the reference common. Any of the first 40 channels in a 50-channel group (0..39, 150..189, etc) that is specified with the DEF CHAN command as SINGLE will be measured single-ended. It is also possible to have an input measured as a member of a differential pair and measured single-ended at the same time.

If Burst Scan Mode is used with the Fast A/D Converter, voltage input channels cannot be changed during scanning. If Continuous Scan Mode is used with this converter, channels can be changed during operation.

Examples

```
DEF CHAN(0) = DVIN

DEF CHAN(20..29) = DVIN, MAX = 0.8
  (The system selects the 8V range because the 0.512V
  range cannot accommodate 0.8V)

DEF CHAN(2..10,12..18) = DVIN, SINGLE
  (Channels 2 through 10 and channels 12 through 18 are
  defined for direct voltage single-ended measurement)
```

See Also

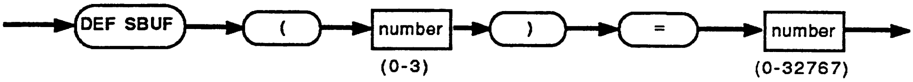
Section 6, Measurement Reference: Voltage Measurement

DEF SBUF
Define Scan Buffer

Format

```
DEF SBUF(<scan task number>)=<buffer size>
```

Syntax Diagram



Description

This command defines a scan buffer for storage of scan data. When a scan buffer is defined, space for <buffer size> records of scan data is allocated in Helios Plus memory. Each scan record contains data about a single scan, which includes: the date and time of the scan, the scan task number, the number of channels scanned, the channel numbers of the channels scanned, the channel measurements, and the status of each of the channels at the time of the scan. Scan data is stored in a scan buffer whenever a scan task is executing, which sends its output to a scan buffer.

Before you define a scan buffer, first use the DEF SCAN command to define the scan task associated with the scan buffer; for example, define scan task zero before scan buffer zero. If a scan task is defined after its associated scan buffer is defined, the scan buffer definition is erased. A scan buffer definition may also be erased by defining a scan buffer with a buffer size of zero.

DEF SBUF
Define Scan Buffer

Scan records may be read from the scan buffer by using the SEND SBUF command or a version of the SHOW SBUF command. SEND SBUF removes scan records from the scan buffer, while the SHOW SBUF commands do not. Generally, scan data is read from the scan buffer in the same order as it is entered. Exceptions occur with SHOW FIRST SBUF, which reads the oldest record, SHOW AGAIN SBUF, which repeats the most recently shown record, and SHOW LAST SBUF, which reads the newest record.

If a scan buffer is filled, scan task data will continue to be logged to the scan buffer, with the oldest records being overwritten by the newest records. The Helios Plus system variable STATUS will also be set to indicate that a buffer overrun has occurred.

Example

Define a 60-record scan buffer.

```
DEF SCAN(1) = CHAN(0..19,40..119)
DEF SBUF(1) = 60
```

With the SCAN definition of 100 channels (20+80), the buffer reserves room for 60 * 100 = 6000 measurement values.

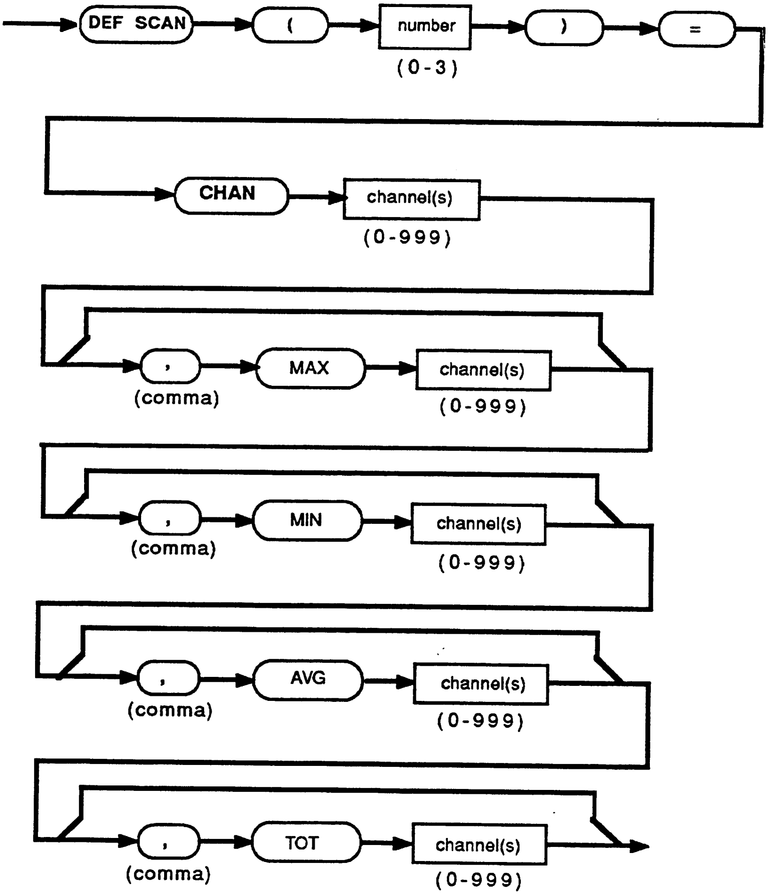
See Also

SEND SBUF
SHOW SBUF
SHOW FIRST|LAST|AGAIN SBUF
RESET SBUF
DEF SCAN
START SCAN
STOP SCAN

Format

```
DEF SCAN(<scan task number>)=CHAN (<channel(s)>)
[,MAX(<channel(s)>)] [,MIN(<channel(s)>)]
[,AVG(<channel(s)>)] [,TOT(<channel(s)>)]
```

Syntax Diagram



DEF SCAN
Define Scan Task

Description

This command defines a scan task. A scan task is a program that executes on Helios Plus and measures a set of channels on a periodic basis. Measurement data from a scan task is sent to an output device. The output device that is permissible depends on math function definitions, as follows:

- Scan task measurement data that does not have MAX, MIN, AVG, or TOT defined may be sent to a host computer, the printer port, or a scan buffer.

- Measurement data from a scan task defined with math functions (MAX, MIN, AVG, or TOT) can be send only to a scan buffer.

Scan tasks can be defined for channels associated with either the High Performance A/D Converter or the Fast A/D Converter. However, a special consideration must be made for use with the Fast A/D Converter. DEF SCAN cannot send meaningful data for channels that have been configured for external trigger input/output (via a jumper on the Fast A/D Converter). Only the 1st and 21st channel within an address block reserved by a Fast A/D Converter can be so configured. Error 55 is returned if DEF SCAN is used for these channels.

Channel values can be represented in different ways using the following math functions:

- MAX

  The value shown for each channel represents the highest sampled value for that channel during the specified scan period.

o MIN

The value shown for each channel represents the lowest sampled value for that channel during the specified scan period.

o AVG

The value shown for each channel represents the average of all sampled values for that channel during the specified scan period.

o TOT

The value shown for each channel represents the total of all sampled values for that channel during the specified scan period.

Three commands control the use of scan tasks. The DEF SCAN command described here assigns a set of channels to a scan task. START SCAN starts the execution of a scan task and specifies the period (in number of scans) required by the MAX, MIN, AVG, and TOT functions. STOP SCAN terminates the execution of a scan task. When scan data is output to a scan buffer, use commands DEF SBUF, SEND SBUF, and SHOW SBUF to define and read from the scan buffer.

Scan tasks must be defined before they can be executed. When using a scan buffer to store scan data, the scan task must be defined before the scan buffer is defined; defining a scan task erases any existing definition of the associated scan buffer. Up to four scan tasks may be defined and executed. Scan tasks are identified by their scan task number, which may be 0,1,2, or 3. When executing multiple scan tasks simultaneously, scheduling conflicts are resolved by scan task priority. The scan task priority is determined by the scan task number (lower numbers have higher priority.)

DEF SCAN
Define Scan Task

Examples

The definition for a single scan of channels 0 through 13, 20, and 300 through 349 would appear as:

```
DEF SCAN(1) = CHAN(0..13, 20, 300..349)
DEF SBUF(1) = 10
START SCAN(1), OUTPUT=SBUF(1), INTERVAL=600
```

Values from these channels are placed in scan buffer number 1 over a 10-minute (600-second) interval.

Using some math functions, a single scan of channels 10 through 14 and 100 through 104, with overall average, maximum, and minimum values would appear as:

```
DEF SCAN(0) = CHAN(10..14, 100..104), AVG(10..14,
100..104), MAX(10..14, 100..104), MIN(10..14, 100..104)
DEF SBUF(0) = 10
START SCAN(0), OUTPUT=SBUF(0), INTERVAL=600, PERIOD=6
```

It is also important to note the additional PERIOD parameter in this START SCAN statement. This parameter is essential for meaningful assessment of math function (MAX, MIN, AVG, TOT) values in any associated scan definition statement. In this example, a period of six scan intervals of ten minutes each is scanned, for a total of one hour.

See Also

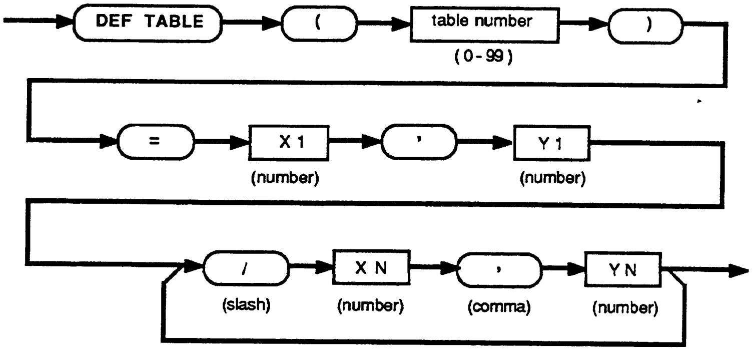
DEF SBUF
START SCAN
STOP SCAN
SHOW SBUF
SHOW FIRST|LAST|AGAIN SBUF
SEND SBUF

DEF TABLE
Define Interpolation Table

Format

```
DEF TABLE(<table number>) = <x1>,<y1>/<x2>,
<y2> [.../<xn>,<yn>]
```

Syntax Diagram



Description

This command creates a user-defined interpolation table.

Up to 100 tables (numbered 0 - 99) can be created. These tables provide the user a transfer function which, when used on input channels, allows the measured value to be transformed to a user-specified value. On output channels, the transformation takes place before the output channel is set.

The contents of a table are entered as pairs of numbers, representing X,Y coordinates in a Cartesian system. Input values are along the X-axis and output values are along the Y-axis.

At least two pairs must be entered to define a valid table. Table size is limited only by available memory. Each pair consists of an input value and an output value, separated by a comma. Pairs are separated by a slash (/). The pairs must be entered in ascending order for the input values.

If the input falls between input value points in the table, a linear interpolation is used to determine its output. If the input falls outside the table, then a linear extrapolation is performed.

If a table definition is too long to fit on a single line, the definition can be continued on the next line, beginning with "DEF TABLE (<table number>) = /".

All input/output values for a given table can be viewed with the LIST TABLE (<table number>) command. The first line returned indicates the number of input/output entries in the table. Each subsequent line shows one or more input/output pair of values.

Examples

Let's measure pressure using a transducer (output 0.1 to 0.2 volts, corresponding to 1 to 20 kilopascals).

The relationship between pressure and output voltage is not linear. The Front End will perform a linearization if a table is defined and the channel function option (CHFN) is included on the DEF CHAN command line.

A legal table for this linearization could be:

```
DEF TABLE(0) = 0.1,1/0.12,3/0.14,6/0.16,10
DEF TABLE(0) = /0.18,15/0.2,20
```

The accompanying channel definition would be:

```
DEF CHAN(0) = DVIN, MAX = 0.2, CHFN = TABLE(0)
```

Suppose we want to use the unipolar voltage output of an analog output channel so that the output reflects some linear relationship with the value we want to apply. Assume these values range from -100 to +500 and the actual output ranges 0 through 10. The table must be defined as follows:

```
DEF TABLE(3) = -100,0/500,10
```

The accompanying channel definition would be:

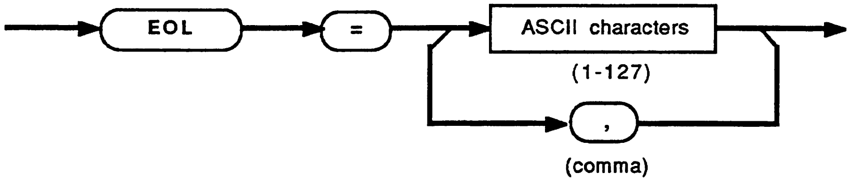
```
DEF CHAN(100) = UNIPOLV, CHFN = TABLE(3)
```

EOL
Set End-of-Line System Variable

Format

```
EOL = <character value>[,<character value>...]
```

Syntax Diagram



(five characters maximum)
(default = 13, 10=<CR>,<LF>)

Description

The EOL system variable is appended to each line sent from the Front End (in computer or terminal mode) to the host computer.

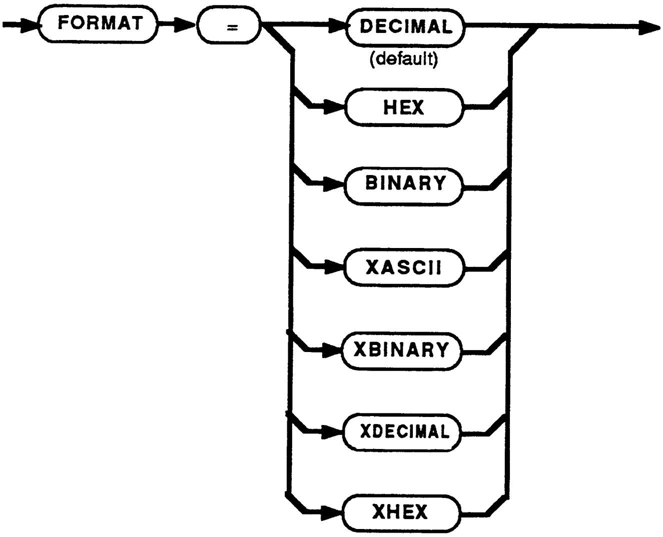
The <character value> parameters are the decimal values for the ASCII characters, in order, that are appended to each line. A maximum of five characters, separated by commas, is allowed.

The default EOL value is 13, 10 (representing carriage return, line feed). Characters 1 through 127 can be set; NUL (decimal 0) cannot be set.

Format

FORMAT = DECIMAL|HEX|BINARY|XASCII|XBINARY|XDECIMAL|XHEX

Syntax Diagram



There is no relationship between the FORMAT system variable and the FORMAT parameter of the LABEL CHAN statement.

FORMAT
Set Format System Variable

Description

This variable determines the format in which various measurements are returned in response to commands. The available formats are: decimal, binary, hexadecimal, extended ASCII, extended decimal, extended binary, and extended hexadecimal. The system default is decimal.

DECIMAL FORMAT

The decimal format (DECIMAL) is a general purpose format and should be supported by most hosts. It consists of an ASCII representation of a measurement in the following generic form,

```
<s><x>.<xxxxx><E><+ | -><yy><EOL>
```

where <s> is a space or minus, <x> is a digit between 0 and nine, and <yy> is the exponent. Each measurement requires 12 bytes plus the end-of-line sequence.

HEXADECIMAL FORMAT

In hexadecimal format, each measurement value is sent as 8 hexadecimal characters, followed by the end-of-line character sequence. Measurement values are represented in Helios Plus as 32 bit floating-point numbers, which conform to the IEEE standard for single-precision numbers. Each hexadecimal character represents 4 bits of the floating point number. The first hexadecimal character represents bit 0 to 3, etc. When the Helios Plus system variable EOL has its default value (EOL=13,10), each measurement sent requires 10 bytes of output as illustrated below.

Character	Bits Represented
1	0 - 3
2	4 - 7
3	8 - 11
4	12 - 15
5	16 - 19
6	20 - 23
7	24 - 27
8	28 - 31
9	<CR>
10	<LF>

(characters 1 - 8 take the value 0 - F)

BINARY FORMAT

Binary format (BINARY) is also a direct representation of the IEEE standard format for single-precision floating-point numbers. Four bytes are sent directly to the host, allowing for the quickest transfer of data to the host. When sent, each byte is comprised of the following bits:

Byte 1 = bits 0 through 7
Byte 2 = bits 8 through 15
Byte 3 = bits 16 through 23
Byte 4 = bits 24 through 31

- Bit 0 is the sign bit (S). If set to 1, the number is negative.
- Bits 1 through 8 represent the binary exponent (E) offset by +127.

- Bits 9 through 31 represent the 23-bit mantissa (Y). Note that the mantissa is normalized (shifted to the left to eliminate leading zeros and the first one bit).

Example

Illustrated below is the translation of an IEEE binary-formatted number to decimal format.

SIGN BIT	OFFSET EXPONENT	MANTISSA (without hidden bit)
0	10000010	0100000 00000000 00000000
0	1 8	9 . 31

Since the sign bit is zero, the number is positive. The binary exponent is given by:

Binary Exponent = Offset Exponent - Offset

= 10000010 (binary) - 127

= 130 - 127

= 3

The mantissa is given by:

hidden bit (=1), plus bits 9-31

Mantissa = 1.01000000000000000000000 (binary)

= .625

Finally the decimal representation of the number is:

$$\text{Decimal Number} = .625 * 2^3$$

$$= 5$$

When sending binary formatted data to the IBM-PC, care should be taken to account for the differences in floating-point representations. There are two common floating-point formats used on the IBM-PC: the Microsoft floating-point format and the IEEE floating-point format. In each case, conversion of floating-point representations is required.

The IBM-PC stores the least significant bits in the first byte, while just the opposite is true for Helios Plus. So, for IBM-PC computer languages that use the IEEE standard, convert Helios Plus binary formatted numbers to IBM-PC formatted numbers by reversing the byte order. Some of the computer languages that use the IEEE standard are Microsoft C, Microsoft QuickBASIC 3.0 (if the PC has a math coprocessor), Turbo Pascal, and Turbo BASIC.

The byte order must also be reversed for the Microsoft floating-point format. Also, the Microsoft floating-point format uses an exponent offset of +128, instead of +127; the sign bit is bit 8 instead of bit 0; and the exponent is in bits 0 to 7, instead of 1 to 8. Some of the computer languages which use the Microsoft floating-point format are IBM compiled BASIC and Microsoft QuickBASIC 2.0.

* IBM and IBM PC are registered trademarks of IBM Corporation

* Microsoft and QuickBASIC are registered trademarks of Microsoft Corporation

XASCII FORMAT

The XASCII format is a general purpose format intended to be used in TERMinal mode. The following example shows the text output by Helios Plus when the XASCII format has been selected:

```
chan (<nnn>) <value> <status><EOL>
where<value> is <s><x>.<xxxxx>E<+|-><yy>
```

<nnn> is the channel number in decimal, <s> is a space or minus,<x> and <y> are digits between 0 and 9, and <status> is a status message that indicates the status of the measurement value.

XBINARY, XDECIMAL, and XHEX FORMATS

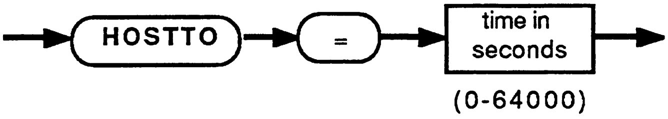
These formats represent numbers the same way as BINARY, DECIMAL, and HEX formats. For measurements, the channel number precedes the channel value, followed by the channel status.

HOSTTO
Set Host Timeout System Variable

Format

HOSTTO = <time interval>

Syntax Diagram



Description

This variable allows the user to set the Host Timeout value in seconds. If the host did not send any character during the specified time, then all information on alarms that change status is sent to the printer port. A timeout value of zero indicates that this feature is turned off (default condition).

Examples

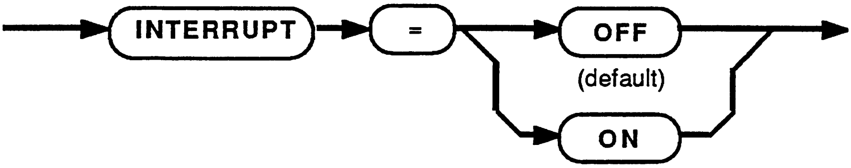
HOSTTO = 120 <CR>	Timeout after 2 minutes of no reception.
HOSTTO = 0 <CR>	Timeout feature is disabled.

INTERRUPT
Set Interrupt System Variable

Format

```
INTERRUPT = ON | OFF
```

Syntax Diagram



Description

This variable is used to enable or disable reporting of changes of Helios Plus status to the host. If the variable is set ON, any change of system status is automatically reported to the host. This is accomplished by asynchronously transmitting the new value of the STATUS system variable (in decimal format), followed by the end-of-line sequence. Default is off.

Example

Interrupt the host for any change in Helios Plus status:

```
INTERRUPT = ON <CR>
```

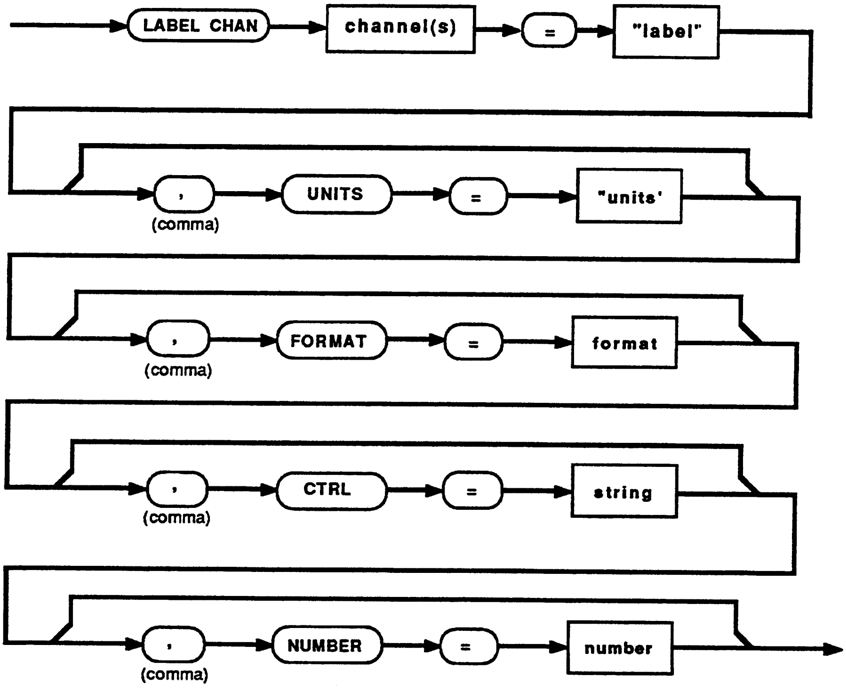
NOTE

Take care when using this function in a multidrop mode. Multiple Helios Plus Front Ends could try to return status information at the same time, resulting in lost data.

Format

```
LABEL CHAN(<channels>) ="<label>"[, UNITS="<units>"]
[, FORMAT=<format>] [, CTRL=<string>]
[, NUMBER=<number>]
```

Syntax Diagram



LABEL CHAN
Label Printer Port Output Format

Description

The LABEL CHAN statement permits customization of the printer port output value. For example, a non-customized output of

```
CHAN 9    9.50000E02
```

could be transformed into

```
HEATER 1 950 DEG C
```

The output sent to the printer port is constructed as follows:

```
<ctrl><label><value><units><status><eol>
```

Unique features of the LABEL CHAN statement are:

o "<label>"

The label string can be used to identify a channel with a name describing that channel's function. For example, a channel measuring operating temperature of an engine could be labeled "ENGINE 9". The label string is also used for any alarm data sent to the printer port.

The label string must be composed of printable characters, all of which are enclosed in quotation marks. The string may total nine characters or less, but the output space reserved for the string will always be exactly ten characters in length. Strings of less than nine characters will be left-justified in the output, with space characters filling out the field.

Finally, where multiple channels are specified in the LABEL CHAN statement, a "#" symbol can be used in the string to further customize each output to a specific channel. Each "#" symbol causes substitution of one channel number digit, beginning from the left. Therefore, three "#" symbols must be used to substitute three-digit channel numbers. For example, LABEL CHAN(9,14)="ENGINE ##" produces labels:

```
ENGINE 9
ENGINE 14
```

- UNITS

  The UNITS string can be a maximum of six characters in length and must be enclosed in quotation marks. The output space reserved for this string is always six characters with spaces filling out the right side of smaller strings. The UNITS string is appended to both scan and alarm information sent to the printer port.

- FORMAT

  This parameter specifies output value decimal point placement in fixed (F) or engineering (E) notation. The number following the F or E (0 .. 5) specifies the number of digits to the right of the decimal point. The formatted value is printed right-justified (as shown below) in a 12-character field. With the fixed (F) formats, leading zeros are not displayed.

LABEL CHAN
Label Printer Port Output Format

The following specifications can be used:

Specification	Output Format
F0	sXXXXXXX
F1	sXXXXXX.X
F2	sXXXXX.XX
F3	sXXXX.XXX
F4	sXXX.XXXX
F5	sXX.XXXXX
E0	sX.EsXX
E1	sX.XEsXX
E2	sX.XXEsXX
E3	sX.XXXEsXX
E4	sX.XXXXEsXX
E5	sX.XXXXXEsXX

Initial "s" = sign, only "-" printed
"Es" = engineering notation and sign

o CTRL

This parameter consists of up to eight decimal codes. Each code represents an ASCII character in a standard escape sequence used for output cursor positioning on a crt. A semi-colon (;) separates decimal character codes.

Five cursor-positioning escape sequences used with the Fluke 1020 terminal are presented below. Escape sequences required by other terminals may differ, and additional terminal-specific sequences may be available. Refer to documentation provided with your terminal for more information.

```
ESCAPE SEQUENCE  |  EQUIVALENT HELIOS PLUS DECIMAL STRING
```

Move Cursor to Absolute Position

```
<ESC>[<pl>;<pc>H |  CTRL=27;91;53;59;48;52;48;72
                         (moves to line 0, column 40)
```

Move Cursor Forward

```
<ESC>[<pn>C      |  CTRL=27;91;49;48;67
                        (move 10 characters forward)
```

Move Cursor Backward

```
<ESC>[<pn>D      |  CTRL=27;91;53;68
                        (move back 5 characters)
```

Move Cursor Up

```
<ESC>[<pn>A      |  CTRL=27;91;48;50;49;65
                        (move up 21 lines)
```

Move Cursor Down

```
<ESC>[<pn>B      |  CTRL=27;91;49;50;66
                        (move down 12 lines)
```

```
<pl> = line number
<pc> = column number
<pn> = number of characters to move
```

Refer to Appendix 9c for a complete listing of ASCII decimal codes.

- NUMBER

  The NUMBER parameter can be used in conjunction with the label string's "#" symbol. NUMBER overrides the related channel number, specifying a unique starting point for the series of channel labels. For example, LABEL CHAN(9,14)="ENGINE #", NUMBER=1 yields:

  ```
  ENGINE 1
  ENGINE 2
  ```

Examples

The following example returns a two-decimal place reading in degrees C, labeled for CHAMBER 1.

```
LABEL CHAN(19)="CHAMBER 1", UNITS="DEG C", FORMAT=F2
```

A response from this channel could be:

```
CHAMBER 1   23.44 DEG C
```

The next example returns a series of four two-decimal place readings in degrees C, labeled for HEATER 1 through HEATER 4.

```
LABEL CHAN(0..3)="HEATER #", UNITS="DEG C", FORMAT=F2,&
NUMBER =1
```

Possible responses are:

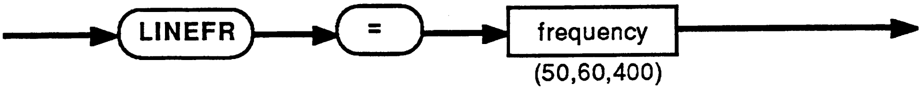
```
HEATER 1   90.33 DEG C
HEATER 2   91.09 DEG C
HEATER 3   92.82 DEG C
HEATER 4   90.98 DEG C
```

LINEFR
Set Line Frequency System Variable

Format

```
LINEFR = 50 | 60 | 400
```

Syntax Diagram



Description

The AC power line frequency is set in order for the A/D converters to provide the maximum noise rejection. A frequency of 50, 60 or 400 Hz can be assigned. The default frequency is determined by the 50/60 Hz switch on the Computer Interface Assembly. The line frequency does not affect the internal clock, which is crystal controlled. A valid LINEFR value causes the A/D converters to be calibrated correctly.

LIST ABUF
List Alarm Buffer Definition

Format

```
LIST ABUF
```

Syntax Diagram

Description (Computer Mode)

This command returns the size of the alarm buffer in number of records. Note that no information is given about the number of records actually stored in the buffer.

Description (Terminal Mode)

In Terminal Mode the size of the alarm buffer is returned in the following form.

```
abuf = 25
```

This means that an alarm buffer capable of storing 25 alarm records has been defined.

LIST ABUF
List Alarm Buffer Definition

Example (Computer Mode)

After receiving the commands

```
MODE = COMP <CR>
DEF ABUF=25 <CR>
LIST ABUF <CR>
```

Helios Plus returns

```
25
```

Example (Terminal Mode)

After receiving the additional commands

```
MODE = TERM <CR>
LIST ABUF <CR>
```

Helios Plus returns

```
abuf = 25
```

See Also

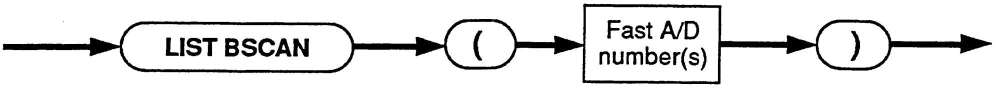
SEND ABUF
SHOW ABUF
SHOW FIRST|LAST|AGAIN ABUF

LIST BSCAN
List Burst Scan Definition

Format

LIST BSCAN (<Fast A/D Converter number>)

Syntax Diagram



Description

LIST BSCAN retrieves the burst scan characteristics (as established with the DEF BSCAN command) for the specified Fast A/D Converter. If no scan characteristics have been defined, the default burst scan characteristics are returned.

Computer Mode Response

In computer mode, a listing with the following fields separated by commas is returned.

Field 1 = Fast A/D Converter number

Field 2 =

0 (1st and 21st channels configured for measurement)

1 (1st and 21st channels configured for external trigger I/O)

```
Field 3 =
      0 (no external trigger type)
      1 (HILO external trigger type)
      2 (LOHI external trigger type)
      4 (HI external trigger type)
      8 (LO external trigger type)

      (Field 3 can also be the OR'd value of more
      than one external trigger type.)

Field 4 = number of scans to filter trigger inputs

Field 5 = number of scans to acquire after a
          trigger event

Field 6 = time between scan starts in milliseconds

Field 7 = 0 (calibration off)
          1 (calibration on)

<EOL>
```

Each of these fields is an ASCII string. Formats and ranges for fields 3 through 5 are the same as specified in the DEF BSCAN command.

Terminal Mode Response

In terminal mode the listing of the information is returned in a more readable form.

```
bscan(<Fast A/D number>),
jumper = measure|xtrig
xtrigtype = lo|hi|lohi|hilo,
filtercnt = <number of scans>,
trigpos = <number of scans>,
scaninterval = <time in milliseconds>
cal = on|off
```

Example 1

The following command example defines burst scanning with external triggering and assumes that the Fast A/D Converter is configured (via hardware jumper) for external trigger inputs and outputs:

```
DEF BSCAN = FAD(15), XTRIGTYPE = HILO,&
XTRIGTYPE = LOHI, FILTERCNT = 12, TRIGPOS = 60,&
SCANINTERVAL = 30, CAL = OFF

LIST BSCAN(15)
```

The returned information in computer mode is:

```
15,1,3,12,60,30,0
```

The returned information in terminal mode is:

```
bscan(15), jumper = xtrig, xtrigtype = lohi,hilo
filtercnt = 12, trigpos = 60, scaninterval = 30,
cal = off
```

Example 2

The next example assumes that the following channel definition has already been made:

```
DEF CHAN(150..154) = DVIN, MAX = 0.5,&
HITRIGGER = 0.3, LOTRIGGER = 0.2
```

LIST BSCAN
List Burst Scan Definition

Example 2 builds on this channel definition with a burst scan definition that uses channel value triggering (rather than external triggering); the Fast A/D Converter is configured via hardware jumper for measurement rather than external trigger input/output.

```
DEF BSCAN = FAD(15), FILTERCNT = 5, TRIGPOS = 30,&
SCANINTERVAL = 75, CAL = ON

LIST BSCAN(15)
```

The response in computer mode would be:

```
15,0,0,5,30,75,1
```

The response in terminal mode would be:

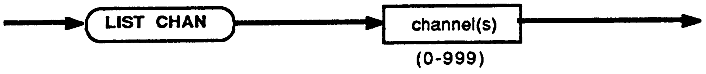
```
bscan(15), jumper = measure, xtrigtype = none
filtercnt = 5, trigpos = 30, scaninterval = 75,
cal = on
```

LIST CHAN
List Channel Definition

Format

LIST CHAN (<channels>)

Syntax Diagram



Description

This command returns a listing of the definitions of designated channels.

Each listing describes both hardware configurations and software channel definitions. The form in which a listing is returned depends on whether the front end is operating in the Terminal (TERM) or Computer (COMP) mode. Explanations of channel listing formats in Terminal and Computer Mode follow on separate pages.

Computer Mode

In the Computer Mode, the LIST CHAN command returns a single line with the number of channels in the list. This is followed by one or more lines for each channel in the list. The first line returned for each channel consists of six numeric fields. Fields are separated by commas. Each field represents a hardware configuration or element of a channel definition. Additional lines of data, when present, represent user-defined parameters, alarm limits parameters, channel function parameters or combinations.

The data returned in the fields of the first line are:

Field 1 = Channel Number
Field 2 = I/O Card
Field 3 = I/O Sub-type
Field 4 = Analog Input/External Trigger Definition
Field 5 = Sensor/Range or Mode Selection
Field 6 = Channel Function and Limits Flags

Field 1. The number in this field identifies the channel number whose definition is listed on the same line.

Field 2. Represents the installed hardware associated with that channel. A number from 0 to 4 will be returned.

Field 3. Represents a further definition of the installed hardware defined by Field 2. The meaning of the number in this field varies, according to the installed hardware option, as shown in the following table.

Field 4. This field is meaningful for analog input channels or for channels configured (via hardware jumper on the Fast A/D Converter) for external trigger input/output. Otherwise, the value in the field is always zero (0). The meaning of the value in this field is shown in the following table.

Field 5. Has meaning only if the channel is an Analog Input or Analog Output channel. Field 5 provides additional information about the analog channel. Fields 5 and 6 are always 0 for XTRIGIN and XTRIGOUT.

FIELD 2	FIELD 3	FIELD 4
0 (none)		
1 Analog Input	0 -161,-162/-175	0 undefined
(-161 A/D)	1 -161,-162/-176	1 DVIN
	2 -161,-163	2 RTD
	3	3 STRAIN
	4 -161,Y2056 Cal	4 RESIST
	5 -161,-164	5 DCIN
		6 AVIN
		7 CAL
		8 TC
		9 TC, COMPMV
		10 TC, RJTEMP
		11 TC, COMPMV/R
6 Analog Input	0 -165/-175	0 undefined
(-165 A/D)	1 -165/-176	1 DVIN
	2 -165,-164	2 RTD
		3 STRAIN
		4 RESIST
		5 DCIN
		6 undefined
		7 undefined
		8 TC
		9 TC, COMPMV
		10 TC, RJTEMP
		11 TC, COMPMV/R
		12 XTRIGIN
		13 XTRIGOUT
2 Digital I/O	0 STATOUT	0
	1 STATIN	
	2 BINARY	
	3 BCD	

3	Analog Output	0		0
4	Counter/ Totalizer	0 1	TOTAL FREQ	0

FIELD 5

(Field 5 is not used if Field 4 = CAL - 7, XTRIGIN - 12, or XTRIGOUT - 13.)

TC		DVIN		Analog Output	
(Field 4= 8,9,10,11)		(Field 4=1)		(Field 2=2)	
		A/D Converter			
		-161	-165		
0	JNBS	64 mV	64 mV differential	0	UNIPOLV
1	KNBS	512 mV	512 mV differential	1	BIPOLV
2	RNBS	8V	8V differential	2	DCOUT
3	SNBS	64V	n/a	3	PVOUT
4	TNBS	n/a	10.5V differential		
5	BNBS	n/a	64 mV single-ended		
6	CHOS	n/a	512 mV single-ended		
7	ENBS	n/a	8V single-ended		
8	NNBS	n/a	10.5V single-ended		
9	JDIN				
10	TDIN				
11	-				

	RESIST	RTD	STRAIN	DCIN	AVIN	
Field 4:	4	2	3	5	6	
	*	**	***			
0	256/64	CU10/LO	0/LO	64mA	250V	-
1	2048/512	CU10/HI	0/HI			
2	64000	DIN385/LO	1/LO			
3	-	DIN385/HI	1/HI			
4	-	USER/LO	2/LO			
5	-	USER/HI	2/HI			
6	-	-	3/LO			
7	-	-	3/HI			
8	-	-	4/LO			
9	-	-	4/HI			
10	-	-	5/LO			
11	-	-	5/HI			

* Range with -163 Option/Range with -164 and -162 Option

** RTD Type/Range

*** Bridge Type/Range

FIELD 6

This field indicates whether user-defined alarm limits and/or a channel function have been assigned to a channel, as follows:

VALUE	MEANING (n = 1..100)
0	No interpolation table selected No polynomial function defined No alarm limits defined XTRIGIN or XTRIGOUT
n < 128	Interpolation table n-1 selected No alarm limits defined
128	Polynomial function only
256	Alarm limits only
512	Square root function only
1024	Additional trigger information. (For -165 Fast A/D Converter only, 1024 means external trigger input or trigger value exceeded. No other FIELD 6 information is returned.)
256+n	Alarm limits and interpolation table n-1 are selected.
256+128 (384)	Alarm limits and polynomial function are selected.
256+512 (768)	Alarm limits and square root function are defined.
1024+256 (1280)	Alarms limits and additional trigger information.

User-defined Parameter Line. The next line returned is the user parameter line. This line is returned only when the user specifies an RTD type 'USER', or a strain type, or when a reference junction temperature (RJTEMP) parameter is specified in a thermocouple (TC) definition.

The RTD constants RO, ALPHA, DELTA and C4 are returned for the user-defined RTD.

Gauge factor (K), Poisson factor (NU), initial voltage (VINIT), and excitation voltage (VEXC) are returned for STRAIN.

Only reference junction temperature (RJTEMP) is returned for a TC with a specified reference junction temperature.

If FIELD 6 returns a value of 1024, an additional line of trigger information is returned as the last response in Computer Mode. This information is:

XTRIGIN, XTRIGOUT	
1 = HILO	<lotrigger value>,<hitrigger value>
2 = LOHI	<hitrigger value>,<lotrigger value>
4 = HI	<hitrigger value>
8 = LO	<lotrigger value>

n = Logical OR of multiple external trigger types

Polynomial Coefficients Line
If no polynomial has been defined, this line is not sent. The line consists of three fields separated by commas.

Field 1 = A coefficient
Field 2 = B coefficient
Field 3 = C coefficient

Example 1 (Computer Mode)

The following example assumes that the Fast A/D Converter is configured for measurement (and not external trigger input/output):

```
DEF CHAN(52,53) = DVIN, SINGLE, MAX = 0.4,&
HI = .312
LIST CHAN(52,53)
```

In computer mode, Helios Plus responds with:

```
2
52,6,1,1,5,256
9.99999E+36,-9.99999E+36,3.12000E-01,-9.99999E+36,1.00000E+01,0
53,6,1,1,5,256
9.99999E+36,-9.99999E+36,3.12000E-01,-9.99999E+36,1.00000E+01,0
```

The first line signifies that 2 channels are being listed. Then, three lines are returned for each of two channels (52 and 53), as follows:

- Line 1

  Channel number (52 or 53)
  Fast Analog Input (6)
  -176 Voltage Input Connector in use (1)
  DVIN measurements (1)
  512 mV range (5)
  Alarm limits information exists (256)

- Line 2

  Limits (9.99999E+36,-9.99999E+36,3.12000E-01,
  -9.99999E+36)
  Hysteresis (1.00000E+01,0)

Remember that default hysteresis is 10% of the span between highest and lowest alarm limit values. In line 2, the highest limit is 9.99999E+36 and the lowest limit is -9.99999E+36; the resulting large hysteresis is not meaningful. In cases such as this, two solutions are available:

- Set hysteresis to zero, or
- Specify more reasonable, smaller values for all four alarm limits.

Example 2 (Computer Mode)

The next example assumes that the Fast A/D Converter is jumper-configured for external trigger input/output (and not measurement).

```
LIST CHAN(50,70)
```

The Helios Plus response is:

```
2
50,6,1,12,0,1024
3
70,6,1,13,0,1024
3
```

Channel 50 is a fast analog input (6) on a -165 Fast A/D Converter using a -176 Voltage Input Connector (1). This channel is configured as an external trigger input (12). Note that Field 5 is unused (0) to indicate external trigger use of this channel. Field 6 (1024) indicates that there is additional trigger information. The next line (3) identifies the trigger information. Here, 3 means 2+1 (HILO + LOHI external triggers.)

Channel 70 is a fast analog input (6) on a Fast A/D Converter using a -176 hardware option (1). The channel is configured as an external trigger output (13). Field 5 is unused (0) to indicate external trigger use of this channel. Field 6 (1024) indicates that there is additional trigger information. The next line (3) identifies the trigger information. Here, 3 means 2+1 (HILO + LOHI external triggers.)

Example 3 (Computer Mode)

```
DEF CHAN(99) = TC, TYPE = KNBS, HI = 100,&
LO = 90, HYST = 0, CHFN = POLY (1.234E-03, 1, 5.67)
LIST CHAN(99)
```

The returned listing is:

```
1
99,1,0,8,1,384
9.99999E+36,-9.99999E+36,1.00000E+02,9.00000E+01,
  0.00000E+00, 0
1.23400E-03,1.000000E+00,5.67000E+00
```

This first line indicates the following:

99	channel number
1	-161 High Performance A/D Converter
0	-162 Thermocouple/DC Volts Scanner and -175 Isothermal Input Connector.
8	thermocouple measurements
1	KNBS type thermocouple
384	(256+128) indicates that both alarm limits and a polynomial function have been defined. Limit values not defined in the limit list of the channel definitions are replaced by their default value (+/- 9.99999E+36).

The second line then indicates that a HI alarm limit of 100 degrees and a LO alarm limit of 90 degrees with a 0% hysteresis has been specified (units of degrees are determined by the TUNIT command and are not reflected in the channel listing). The third line indicates that the following polynomial has been specified:

$$.001234x^2 + x + 5.67$$

Terminal Mode

In the Terminal Mode, channel definition data are returned as alphabetic, rather than numeric, strings, making them more readily understandable. All responses begin with an I/O device category.

o All responses begin with an I/O device category.

```
aichan:   -161                  (High Performance A/D)
     iso:      -162/-175
     v/i:      -162/-176
     r:        -163

faichan:  -165                  (Fast A/D)
     iso:      -175
     v/i:      -176
```

NOTE

If the Fast A/D Converter is jumpered for external trigger input/output and the related 1st or 21st channel is listed, the next response is:

```
,def=xtrigin|xtrigout
,xtrigtype=none|hi|lo|lohi|hilo
```

If a trigger value has been specified for this channel, the following additional trigger information line is returned:

```
lotrigger=<value>,hitrigger=<value>
```

If only one trigger value has been specified, the default value is returned for the other trigger value. The defaults are:

```
lotrigger: -9.99999E+36
hitrigger: +9.99999E+36
```

If no trigger value has been specified, the lotrigger/hitrigger response line is not returned.

Other I/O device categories are:

```
dgchan:   -168
     statin:   -179 as status inputs
     statout:  -169
     binary:   -179 as binary input
     bcd:      -179 as BCD input

aochan:   -170
     unipolv:  as 0 to 10V output
     bipolv:   as -5 to 5V output
     dcout:    as 4 to 20 mA output
     pvout:    as 0 to 100% output

ctchan:   -167
     total     as totalizer input
     freq      as frequency input
```

o The channel number is identified in the parentheses associated with the device category.

o The hardware type is identified for the specified channel. Type is defined by the installed option assembly and, in several cases, by switch settings on those assemblies.

o The analog input measurement is listed.

- The external trigger definition for this channel is listed, as follows:

```
,xtrigtype=none

     or

,xtrigtype=lo|hi|lohi|hilo
```

- For the Fast A/D Converter, the following measurement range and type information is available:

```
range = 64 mV differential
range = 512 mV differential
range = 8V differential
range = 10.5V differential
range = 64 mV single-ended
range = 512 mV single-ended
range = 8V single-ended
range = 10.5V single-ended
```

- In Terminal Mode, the trigger definition for the Fast A/D Converter is appended onto the other channel limits information. This specification appears as:

```
lotrigger=<value>
hitrigger=<value>
```

- Additional parameters identifying any further items specified by the definition are given.

Example 1 (Terminal Mode)

The following commands assume that channels 0 and 20 on the Fast A/D Converter are configured to support measurement channels:

```
DEF  CHAN(52,53) = DVIN, SINGLE, MAX = 0.4,&
HI = .300, HYST = 0, HITRIGGER = .312
LIST CHAN(52,53)
```

Helios Plus could respond in terminal mode with:

```
faichan(52) = V/I, def = dvin, range = 512 mV single-ended
   hihi = 9.99999E+36, lolo = -9.99999E+36,
   hi = 3.00000E-01, lo = -9.99999E+36
   hyst = 0.00000E+00
   lotrigger=-9.99999E+36,hitrigger=3.12000E-01
faichan(53) = V/I, def = dvin, range = 512 mV single-ended
   hihi = 9.99999E+36, lolo = -9.99999E+36,
   hi = 3.00000E-01, lo = -9.99999E+36
   hyst = 0.00000E+00
   lotrigger=-9.99999E+36,hitrigger=3.12000E-01
```

Two channels are listed here. Channels 52 and 53 are fast analog measurement inputs on a -165 Fast A/D Converter using a -176 Voltage Input Connector, measuring direct voltage, using the 512mV single-ended range, with alarm limits, default lotrigger, and hitrigger of 0.312V.

Example 2 (Terminal Mode)

This example assumes that the Fast A/D Converter is configured for external trigger I/O (rather than measurement) and the following command is sent:

```
LIST CHAN(50,70)
```

In terminal mode, Helios Plus responds with:

```
faichan(50) = V/I, def = xtrigin, xtrigtype = lohi,hilo
faichan(70) = V/I, def = xtrigout, xtrigtype = lohi,hilo
```

Again, two channels are listed. Channel 50 is on a -165 Fast A/D Converter using a -176 Voltage Input Connector. The channel is configured as an external trigger input with external trigger types of HILO and LOHI.

Channel 70 is on a -165 Fast A/D Converter using a -176 Voltage Input Connector. The channel is configured as an external trigger output. Its corresponding external trigger input on channel 50 has trigger types of HILO and LOHI.

Note that additional information normally associated with the LIST CHAN response (measurement range, limit values, etc) is not returned when the channel is used for external triggering.

LIST ERROR
List Error Log

Format

LIST ERROR

Syntax Diagram

Description (General)

This command lists the error log, which contains the errors detected while making measurements with the SEND CHAN command. The error log may hold up to 20 errors. If there are more than 20 errors, only the last 20 errors detected will be remembered. Once errors are listed, they will be removed from the error log.

Description (Computer Mode)

In the Computer Mode, the error log is returned as a series of lines in decimal format. The first line consists of a single number, which indicates the number of errors returned. Each subsequent line represents a single channel. Each channel line contains two numbers, separated by a comma: the first indicates the number of the channel being reported; the second indicates the error code number. The error code number can then be looked up in the Error Code List in Section 8 of this manual.

LIST ERROR
List Error Log

Description (Terminal Mode)

In the Terminal Mode, the error log is returned in a more easily readable form, shown below.

```
chan(x) <error message>
```

The "x" field indicates the channel number.

Example (Computer Mode)

The command

```
LIST ERROR
```

would return an error log looking something like:

```
3
30,19
66,15
71,15
```

This error log reports errors on three channels: error 19 on channel 30, error 15 on channels 66 and 71. In the Error Code List, Error 15 reads "Out of range," meaning that an input was measured that exceeded the programmed range. Error 19 reads "Over Temperature," which means the temperature is outside the range for the thermocouple or RTD being used.

The Terminal Mode command

```
LIST ERROR
```

would return an error log looking something like:

```
chan(30) - Illegal BCD

chan(68) - Open TC
```

This means that on channel 30, a BCD digit greater than 9 was present in one or more digit fields of a digital input; and that on channel 68, a thermocouple is broken or damaged.

See Also

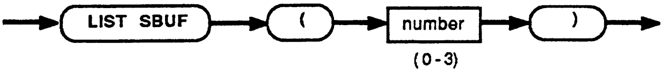
Section 8, Error Messages

LIST SBUF
List Scan Buffer Definition

Format

```
LIST SBUF(<scan task number>)
```

Syntax Diagram



Description

This command returns the number of records that can be stored in the scan buffer. It does not return the number of records that are stored in the buffer or any scan data. To read scan records from the scan buffer, use the SEND SBUF command or one of the SHOW SBUF commands.

The format in which the scan buffer definition is returned depends on the setting of the Helios Plus system variable MODE. Refer to the following examples.

Example (Computer Mode)

After receiving the commands

```
MODE = COMP
DEF SBUF(1)=60
LIST SBUF(1)
```

Helios Plus returns

```
60
```

Example (Terminal Mode)

After receiving the commands

```
MODE = TERM
LIST SBUF(1)
```

Helios Plus returns

```
sbuf(1) = 60
```

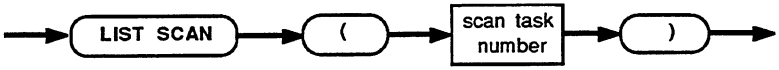
This means that scan buffer 1 had been defined to reserve memory for 60 records.

LIST SCAN
List Scan Definition

Format

```
LIST SCAN(<scan task number>)
```

Syntax Diagram



Description (General)

This command returns a listing of the definition of the designated SCAN task (the SCAN task identification, number of channel groups and all channel numbers associated with that particular SCAN task.) The order in which the channel numbers appear in the list is identical to the order in the SCAN definition.

Description (Computer Mode)

In computer mode, a listing with the following fields will be returned.

```
Field 1 = scan task number
Field 2 = number of channel groups
<EOL>
Field 3 = first channel number of a group
Field 4 = last channel number of a group
<EOL>
```

Fields 3, 4 and <eol> are repeated [Field 2] times. Fields are separated by commas.

Description (Terminal Mode)

In Terminal Mode, the listing of the definition of the designated SCAN task will be returned in a more readable form, as shown below. The order in which the channel numbers appear in the list is identical to the order in the SCAN definition.

Example (Computer Mode)

The commands:

```
MODE = COMP

DEF SCAN(1)=CHAN(0..5,10..15,17,20..39) <CR>
LIST SCAN(1)
```

result in the following listing:

```
1,4
0,5
10,15
17,17
20,39
```

The Terminal Mode command

```
MODE = TERM

LIST SCAN(1)
```

would return scan information looking like

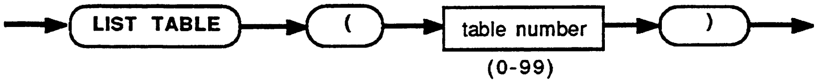
```
scan(1) = chan(0..5,10..15,17,20..39)
```

LIST TABLE
List Interpolation Table

Format

```
LIST TABLE(<table number>)
```

Syntax Diagram



Description (General)

This command returns the contents of a user-defined interpolation table from the Front End memory. One table is returned with each command.

The form in which the interpolation table is returned will be determined by whether the system is operating in Computer or Terminal Mode.

LIST TABLE
List Interpolation Table

Description (Computer Mode)

In the Computer Mode, the designated table is returned as a series of (at least) three data lines in the following format.

```
n
x0,y0
x1,y1
.
.
xn-1,yn-1
```

The first line consists of a single field, "n", which indicates the number of pairs in the table.

Each subsequent line consist of pairs of numbers, separated by a comma (,). These numbers are the actual contents of the table. The first number of each pair is the input and the second is the corresponding output. The numbers are represented in scientific notation.

Example (Computer Mode)

```
MODE = COMP
LIST TABLE(19)
```

returns table 19, which has four pairs, in the following form

```
4
1.00000E+00, 1.12345E+00
2.20000E+00,-3.22453E+00
5.50000E+00,-8.45987E+00
1.00000E+01,-1.23453E+01
```

Description (Terminal Mode)

In the Terminal Mode, the command, LIST TABLE(table number), returns the contents of a user-defined interpolation table in the following form.

```
table(t) = x0,y0 / x1,y1 / .. / xn-1,yn-1
```

"t" indicates the table number, while x0,y0 through xn-1,yn-1 are the contents. Numbers are represented in scientific notation.

Example (Terminal Mode)

```
MODE = TERM
LIST TABLE(25)
```

returns Table 25 in the following form:

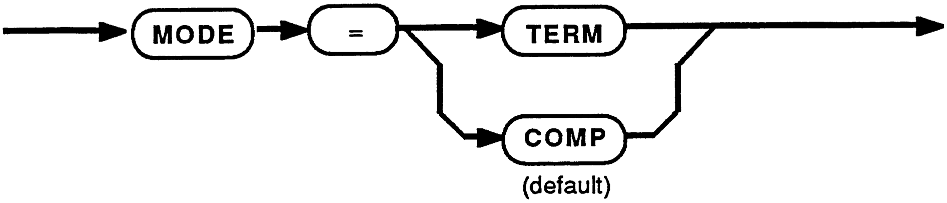
```
table(25)=1.20000E+00,1.00000E+00/1.20000E+01,3.33333E+00
```

MODE
Set Mode System Variable

Format

MODE = COMP | TERM

Syntax Diagram



Description

This variable determines whether Helios Plus operates in terminal mode (MODE=TERM) or computer mode (MODE=COMP). Setting MODE affects the way Helios Plus responds to commands. When MODE=COMP, Helios Plus responses are compact and easily read by computers. When MODE=TERM, Helios Plus responses are verbose and easily read by humans. Computer mode is the default mode.

See Also

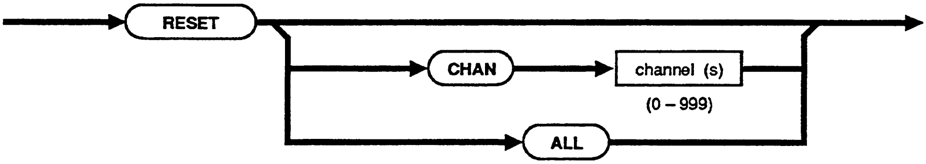
Section 4

RESET RESET ALL RESET CHAN
Reset System or Channel(s)

Format

```
RESET [ALL | CHAN(<channel[s]>)]
```

Syntax Diagram



Description

RESET causes the Front End to:

- o Perform a system self test. If errors are detected, the highest priority error message is sent to the host.
- o Clear all channel definitions, interpolation tables, labels, and the channel list.
- o Reset and initialize all I/O devices. The output of all status output channels are set to zero and analog output channels are set to -5V, 0V, or 4 mA, depending on the output channels.
- o Build a list containing all channels.
- o Calibrate all system A/D converters.
- o Erase the scan task, scan buffer, and alarm buffer definitions.

RESET ALL causes the Front End to:

- Reset all system variables except TIME$ and DATE$ to default values.
- Clear the error log.
- Execute all of the items in the RESET description.
- Reset alarms.

RESET CHAN(channel[s]) causes the Front End to:

- Reset the channel descriptions of the named channel(s) to the initial state.
- Clear channel labels defined for the named channel(s).

NOTE

Use of RESET CHAN during an active burst scan results in an error (52). Changes in channel definitions are not allowed during burst scanning.

Examples

To reset the entire system, enter: RESET ALL

To reset channels 15 through 30, enter

```
RESET CHAN(15..30)
```

See Also

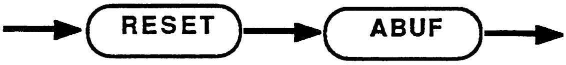
Appendix 9d

RESET ABUF
Reset Alarm Buffer

Format

```
RESET ABUF
```

Syntax Diagram



Description

This command removes records that have been previously read with the 'SHOW ABUF' command.

Example

```
RESET ABUF <CR>
```

See Also

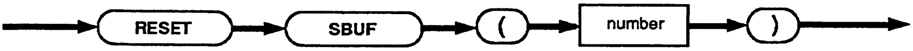
```
SHOW ABUF
SHOW FIRST|LAST|AGAIN ABUF
SEND ABUF
```

RESET SBUF
Reset Scan Buffer

Format

```
RESET SBUF(<scan task number>)
```

Syntax Diagram



Description

This command removes those records from the designated buffer that have been previously read with the 'SHOW SBUF' command.

Example

```
RESET SBUF(1) <CR>
```

See Also

SHOW SBUF
SHOW FIRST|LAST|AGAIN SBUF
SEND SBUF

SEND ABUF
Send Alarm Buffer Data

Format

SEND ABUF

Syntax Diagram

Description

This command returns the oldest record in the alarm buffer. The alarm record is then deleted from the alarm buffer.

Each alarm record has six fields of information:

Field 1: Date
Field 2: Time
Field 3: Alarm Type
Field 4: Alarm Parameter 1
Field 5: Alarm Parameter 2
Field 6: Alarm Parameter 3

Fields 4, 5, and 6 are only relevant to certain alarm types (see the table below) and have undefined values elsewhere.

SEND ABUF
Send Alarm Buffer Data

Condition	Field 3 (Alarm Type)	Field 4 (Alarm Parameter 1)	Field 5 (Alarm Parameter 2)	Field 6 (Alarm Parameter 3)
End-of-Buffer	0	N/A	N/A	N/A
Cold Start	1	N/A	N/A	N/A
Warm Start	2	N/A	N/A	N/A
Status Change	3	Channel #	New Status	New Value
Host Off-line	4	N/A	1	N/A
Host On-line	5	N/A	0	N/A
Abuf Overflow	6	N/A	N/A	N/A
Abuf Cleared*	7	N/A	N/A	N/A
Sbuf Overflow	8	Task #	1	N/A
Sbuf Cleared*	9	Task #	0	N/A

Where:

'new value' is the channel value at the time of the channel status change.

'channel #' is the number of the channel that had a status change.

'new status' is the most recent status of the channel (see table for channel status codes).

'task #' is the scan task number of the buffer that changed status.

* Buffer overflow condition was cleared

Channel status codes are composed of a set of flags encoded in the bits of a two-byte number:

Bit	Meaning When Set
0	lo alarm
1	hi alarm
2	lolo alarm
3	hihi alarm
4..7	unused
8..15	error code

where 'error code' is one of the error codes listed in Section 8.

The format in which the alarm buffer data is returned depends on the value of the system variable FORMAT.

FORMAT is XASCII:

```
dd-Mmm-yy hh:mm:ss : Alarm message   <EOL>
```

where:

dd-Mmm-yy — is the date the alarm condition was detected; e.g., 21-Jan-84

hh:mm:ss — is the time the alarm condition was detected in hours, minutes, and seconds; e.g., 20:12:23

alarm message — contains the information in Fields 3 through 6 in a self-explanatory format

End-of-Buffer Format:

```
[EOB]
```

SEND ABUF
Send Alarm Buffer Data

FORMAT is DECIMAL or XDECIMAL:

<day>, <month>, <year>, <time>, <alarm type>,
<alarm parameter 1>, <alarm parameter 2>,
<alarm parameter 3> <EOL>

where:

<day> is the day of the month

<month> is the month number (1 to 12)

<year> is a number representing the last two digits of the year; e.g., 1987 would be returned as '87'

<time> is a number representing the number of milliseconds past midnight

<alarm type> *
is the number indicating either the alarm type or the end-of-buffer condition.

<alarm parameter 1> *
<alarm parameter 2> *
<alarm parameter 3> *
are the parameters related to particular alarm types. Alarm parameter 3 is represented in scientific notation.

* See previous page for fields 3 through 6.

FORMAT is HEX or XHEX:

This form resembles DECIMAL and XDECIMAL except:

- o Each field is represented as a single-precision IEEE floating-point value encoded as eight hexadecimal digits.
- o End-of-line sequences, instead of commas, separate fields.

FORMAT is BINARY or XBINARY:

This representation is the same as HEX and XHEX except:

- Each field is represented as a single-precision IEEE floating-point value encoded as four bytes.
- Fields are not separated by anything.
- An end-of-line sequence does not follow the record.

Examples

The following are valid responses to SEND ABUF:

In XASCII format:

```
05-Jun-87   11:05:16  :  Cold start
05-Jun-87   11:05:16  :  Warm start
05-Jun-87   11:05:16  :  Status change  chan(003)
                         4.91761E+00 lo alarm
05-Jun-87   11:05:16  :  Host off-line
05-Jun-87   11:05:16  :  Host on-line
05-Jun-87   11:05:16  :  ABUF overflow
05-Jun-87   11:05:16  :  ABUF cleared
05-Jun-87   11:05:16  :  SBUF overflow  sbuf(0)
05-Jun-87   11:05:16  :  SBUF cleared   sbuf(0)
05-Jun-87   11:05:16  :  Status change  chan(003)
                         5.91761E+00
05-Jun-87   11:05:16  :  Status change  chan(003)
                         9.99999E+37  ERROR  5
```

In DECIMAL format:

```
     15, 6,87,4546321,3,8,1, 0.00000E+00
```

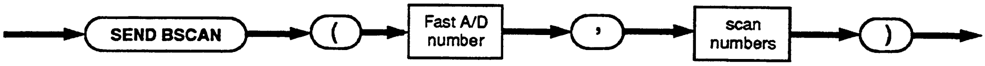
which is a channel status change alarm for channel 8 where the status changed to a lo alarm.

SEND BSCAN
Send Burst Scan Data

Format

SEND BSCAN(<Fast A/D Converter number>,<scan numbers>)

Syntax Diagram



Description

SEND BSCAN returns all specified burst scan records for the specified Fast A/D Converter. Scan numbers, or positions in the burst scan buffer, are listed in a syntax similar to channel numbers. One or more scan numbers separated by commas, a range of scan numbers, or multiple ranges separated by commas are allowed. Negative, zero, and positive scan numbers are valid.

Scan record data is returned in the order in which it appears in the scan number list. Scan number 0 is the newest record in the buffer (if the Fast A/D Converter is still burst scanning) or the record at the time of a trigger event (if the Fast A/D Converter has stopped scanning after the trigger event.)

Each scan record contains one reading for each channel that was defined with a DEF CHAN command when the START BSCAN command was received. The channel reading data is arranged in ascending channel order. Refer to SEND CHAN for a description of response format.

NOTE

> Channel readings returned with the SEND BSCAN command are not checked for alarm limits.

In Computer Mode (with the system variable COUNT ON), the first value returned indicates the number of scan records to be returned. The second value returned then indicates the number of channel measurements to be returned per scan record. In Terminal Mode, no record or channel counts are returned.

NOTE

> Before changing channel definitions for a Fast A/D Converter that has stopped burst scanning, always use the SEND BSCAN command to retrieve burst scan information. All burst scan data is discarded as soon as a channel definition command is received by a stopped Fast A/D Converter.

If SEND BSCAN is used during burst scanning, the response is as follows:

- o Only the newest scan record is available. Even if a scan number other than 0 is specified, the newest scan record is sent.
- o To retrieve other records, send the STOP BSCAN command or wait until scanning stops due to a trigger event.

Note, however, that the Fast A/D Converter may continue scanning after a trigger event. Use the SEND BSCANIPS command to verify that burst scanning has stopped.

No alarm processing is performed on measurement values returned with the SEND BSCAN command. To check for out of limits inputs during burst scanning, use DEF SCAN and START SCAN to scan the channel(s) or use the SEND CHAN command.

Example (Computer Mode)

After receiving the commands:

```
DEF CHAN(307..309) = DVIN
COUNT = ON
FORMAT = DECIMAL
START BSCAN(30)
STOP BSCAN(30)
SEND BSCAN(30,-1,-2)
```

Helios Plus responds with:

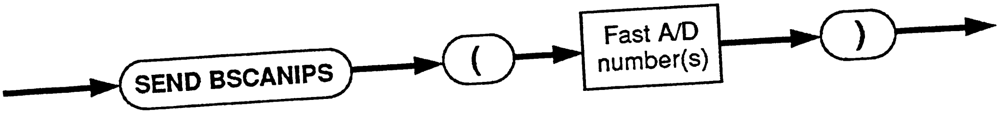
```
2.00000E+00         (2 scan records)
3.00000E+00         (2 channel measurements/record)
1.23456E+00         (reading value)
2.34567E+00         (reading value)
3.45678E+00         (reading value)
7.89012E+00         (reading value)
8.90123E+00         (reading value)
9.01234E+00         (reading value)
```

SEND BSCANIPS
Send Burst Scan in Progress Status

Format

```
SEND BSCANIPS(<Fast A/D Converter numbers>)
```

Syntax Diagram



Description

SEND BSCANIPS identifies the burst scanning status of the specified Fast A/D Converters. General burst scan activity in the Fast A/D Converter system can thereby be monitored. Retrieval of data from a burst scan buffer can occur only when burst scanning for the respective Fast A/D Converter is halted.

Status is returned as the ASCII string "on" (burst scanning in progress for at least one of the specified Fast A/D Converters) or "off" (burst scanning not in progress), followed by an end-of-line sequence. If no Fast A/D Converter number is specified, the status for all Fast A/D Converters in the system is returned.

Example

Send the following command:

```
SEND BSCANIPS(35,60)
```

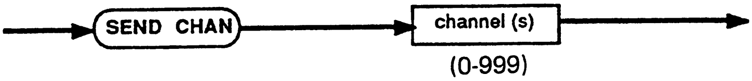
If burst scanning is in progress for either or both Fast A/D Converter 35 or 60, the response is: on

SEND CHAN
Send Channel Measurement(s)

Format

```
SEND CHAN(<channels>)
```

Syntax Diagram



Description

This command returns a channel measurement (in appropriate units). In contrast, the LIST CHAN command returns the definition of the designated channel(s).

A single channel, multiple channels, a range of channels, or multiple ranges can be specified. Valid channel numbers are 0 through 999.

Channels are returned in the order in which they appear in the SEND list. However, they are not necessarily measured in that order. A channel is measured only once, regardless of the number of times it may (inadvertently) appear in the SEND list.

NOTE

If the system variable COUNT is ON, the first value of a returned measurement indicates the number of channel measurements to be returned.

USE WITH FAST A/D CONVERTER

For channels associated with a Fast A/D Converter (-165), the source for measurement data depends on the scan mode in use. For Continuous Scan Mode, measurement data stored by the mainframe is sent. This process is the same as used with the High Performance A/D Converter (-161). For Burst Scan Mode, the most recent measurement data in the Fast A/D Converter buffer is sent.

SEND CHAN cannot return meaningful data for channels that have been configured for external trigger input/output (via a jumper on the Fast A/D Converter). (Only the 1st and 21st channel within an address block reserved by a Fast A/D Converter can be so configured.) Error 55 is returned if SEND CHAN is used for these channels.

The response to a SEND CHAN command depends on the FORMAT system variable setting. The following descriptions show the response format for the different system variable settings:

o XASCII Format

```
chan(nnn)  sf.fffffE±ee  <status message>
```

o DECIMAL Format

```
sf.fffffE±ee
```

o XDECIMAL Format

```
nnn,sf.fffffE±ee,<channel status code>
```

o HEX Format

```
hhhhhhhh <EOL>
```

- XHEX Format

```
iiiiiiii <EOL>
hhhhhhhh <EOL>
jjjjjjjj <EOL>
```

- BINARY Format

```
bbbb
```

- XBINARY Format

```
aaaa bbbb cccc
```

where:

aaaa represents the channel number in IEEE single-precision floating-point format.

bbbb represents the channel value in IEEE single-precision floating-point format.

cccc represents the status code as a binary IEEE single-precision floating-point format.

hhhhhhhh represents the channel value as eight hexadecimal digits in IEEE single-precision floating-point format

iiiiiiii represents the channel number as eight hexadecimal digits in IEEE single-precision floating-point format.

jjjjjjjj represents the channel status message as eight hexadecimal digits in IEEE single-precision floating-point format.

nnn represents the channel number.

sf.fffffE+-ee represents the channel value in scientific notation.

If an error is detected as a result of a SEND CHAN command, the value

```
9.99999E+37
```

is returned for that channel. By listing the error log (LIST ERROR), the last 20 errors can be inspected.

Examples

After receiving the commands:

```
COUNT = OFF
FORMAT = DECIMAL
SEND CHAN(3)
```

Helios Plus responds with:

```
1.23456E+00
```

After receiving the commands:

```
MODE = COMP
COUNT = ON
SEND CHAN(5,7,9)
```

Helios Plus responds with:

```
3.00000E+00
1.23456E+00
3.45678E+00
4.56789E+00
```

After receiving the commands:

```
MODE = TERM
FORMAT=XASCII
SEND CHAN(3)
```

Helios Plus responds with:

```
CHAN(003)  1.23567E+00  hi alarm
```

After receiving the commands:

```
FORMAT=DECIMAL
SEND CHAN(3)
```

Helios Plus responds with:

```
1.23567E+00
```

After receiving the commands:

```
FORMAT=XDECIMAL
SEND CHAN(3)
```

Helios Plus responds with:

```
3, 1.23567E+00, 2
```

After receiving the commands:

```
FORMAT=HEX
SEND CHAN(3)
```

Helios Plus responds with:

```
3A000008
```

After receiving the commands:

```
FORMAT =XHEX
SEND CHAN(3)
```

Helios Plus responds with:

```
00000000
41A00000
40000000
```

The following are valid SEND commands:

```
SEND CHAN(4)

SEND CHAN(7,9,14)

SEND CHAN(3..6,9..15)

SEND CHAN(12..20,34..38,42..48)

SEND CHAN(13..19,17..27)
```

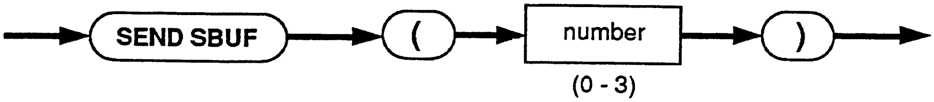
Note: In the last command, channels 17 through 19 are measured once but reported twice.

SEND SBUF
Send Scan Buffer Data

Format

SEND SBUF(<scan buffer number>)

Syntax Diagram



Description

This command returns the oldest record in the designated scan buffer. The scan record is then deleted from the scan buffer.

Each scan record incorporates a scan header subrecord and zero or more channel measurement subrecords. Alternately, a user-defined header can be specified as part of the START SCAN statement for the printer port. This feature is fully described in the START SCAN area of this section.

The scan header is as follows:

- FORMAT is XASCII

  Header format for records that contain measurement data:

  ```
  dd-Mmm-yyy, hh:mm:ss, scan(n)<EOL>
  ```

  Header format for indicating end of buffer: [EOB]

- FORMAT is DECIMAL or XDECIMAL

  <day>,<month>,<year>,<time>,<buffer identification>

  <day> is the day of the month.

  <month> is the month number (1-12).

  <year> is a number representing the last two digits of the year; e.g., 1989 would be returned as "89".

  <time> is a number representing the number of milliseconds past midnight.

  <buffer identification> is the scan buffer number plus one for records that contain measurement data and is zero (0) for indicating the end of buffer.

- FORMAT is HEX or XHEX

  The header format is the same as for DECIMAL or XDECIMAL except:

  Each number is represented as a single-precision IEEE floating-point value encoded as eight hexadecimal digits.

  End-of-line sequences separate numbers instead of commas.

- o FORMAT is BINARY or XBINARY

  The header format is the same as for HEX or XHEX except:

  > Each single-precision IEEE floating-point value is encoded as four bytes.
  >
  > Numbers are not separated by anything.
  >
  > An end-of-line sequence does not follow the record.

The format in which the optional count and scan buffer measurement data is returned also depends on the value of the system variable FORMAT. This format is the same as that shown for the SEND CHAN command.

Math Function Response in Extended Format

Channels that are defined for scan tasks (with math functions MAX, MIN, AVG, or TOT) provide an encoded extended format response. This definition is discussed under DEF SCAN.

For XASCII format, the following self-explanatory text format is returned:

```
<math function> (<channel number>), <value>, <status>
```

SEND SBUF
Send Scan Buffer Data

For XDECIMAL, XHEX, and XBINARY formats, the encoded response is a single number representing the channel number plus a sum for the math functions activated (through the scan task) for that channel. In XDECIMAL format, these math function numbers are:

o	MAX	2048	(Bit 11)
o	MIN	4096	(Bit 12)
o	AVG	8192	(Bit 13)
o	TOT	16384	(Bit 14)

Divide the response in XDECIMAL format by 2048; the remainder represents the channel number. Then subtract the channel number from the response to obtain a number representing the sum of activated math functions. Finally, derive the actual activated math functions using the numbers listed above.

Examples

```
SEND SBUF(2) <CR>
```

If FORMAT is XASCII, the following record may be output:

```
10-Oct-86, 11:42:06, scan(2)
chan(000) -3.66211E-04
chan(001) -8.54493E-02 lo alarm
     |         |
chan(019) -9.99999E+37 ERROR 10
```

If FORMAT is XDECIMAL, the output may look like:

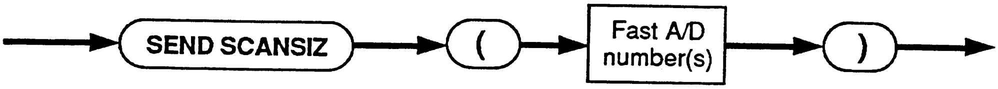
```
10,10,86,15962120,3
0,-3.66211E-04,0
1,-8.54493E-04,1
```

SEND SCANSIZ
Send Maximum Number of Burst Scans in Buffer

Format

```
SEND SCANSIZ(<Fast A/D number>)
```

Syntax Diagram



Description

SEND SCANSIZ returns the scan record size of the burst scan buffer of the specified Fast A/D Converter. This number is dependent on the number of channels defined for scanning and the amount of storage available in Fast A/D Converter memory. If no channels for this Fast A/D Converter are defined, the SEND SCANSIZ response is zero.

The response format is an ASCII string of 1 to 7 digits, ranging from 0 through 1,048,575, followed by an end-of-line (EOL) sequence.

Example

With the following command sent:

```
SEND SCANSIZ(75)
```

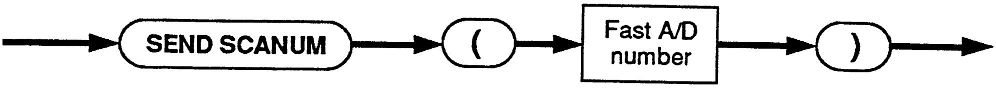
A response of '324'signifies that Fast A/D Converter 75 can hold a maximum of 324 scan records.

SEND SCANUM
Send Scan Number Information

Format

```
SEND SCANUM(<Fast A/D Converter number>)
```

Syntax Diagram



Description

SEND SCANUM requests two pieces of information from the designated Fast A/D Converter. The response comprises two numbers that can subsequently be used in determining the scan buffer records to retrieve with the SEND BSCAN command. These two numbers are described below:

o The first number identifies the oldest scan record in the burst scan buffer, indicating the number of scans that occurred before the trigger event. This number can be either negative or positive, as follows:

Negative The number of scans that took place before a trigger event. If there never was a trigger event, or there were no scans made after a trigger event, the number is always negative, and its absolute value is always less than the maximum number of records that can fit into the scan buffer.

Positive A trigger event happened, and there were more scans made after the trigger event than the number of scan records that could fit in the burst scan buffer. This number can be as large as 1,048,575 minus the size of the scan buffer.

- The second number identifies the newest scan record in the burst scan buffer, indicating the number of scans that occurred after the trigger event. This number can be either positive or zero, as described below:

Zero Either a trigger event never happened, or no scans were made after a trigger event that did happen.

Positive The number of scans that took place after the trigger event is shown. This number can be as large as 1,048,575.

Usually, a trigger event can be identified to a specific record in the burst scan buffer. An exception occurs when the value of TRIGPOS exceeds the number of records allowed in the buffer, in which case the trigger event record is written over.

The SEND SCANSIZ command retrieves the maximum number of scan records that can fit in the memory available for the burst scan buffer.

Identify the trigger event by using the SEND SCANUM command. The following examples examine the four most probable burst scan buffer configurations. Records positioned lower in these examples occur earlier than higher records.

Remember that each scan record in the burst scan buffer automatically includes all channels that have been defined (with DEF CHAN) on the associated Fast A/D Converter.

The position of the trigger event within the burst scan buffer can be determined from:

SCANSIZ = |SCANUM #1| + SCANUM #2 + 1

In each of the following four examples, the assumed value retrieved by SCANSIZ is 500.

A: TRIGPOS = 0. The burst scan buffer contains only the trigger event and data recorded prior to the trigger event.

n = (SCANSIZ - 1 - TRIGPOS)

SEND SCANUM would retrieve a negative number, followed by 0. This response can range from -1,0 (meaning that the buffer contains only one record occurring before the trigger event record) to -499,0 (signifying that the buffer size of 500 is filled.)

Scan Record 0 (newest record) **TRIGGER EVENT**

Scan Record -1

Scan Record -2

Scan Record -3

.
.
.

Scan Record -n (oldest record)

B: TRIGPOS = SCANSIZ. The burst scan buffer contains only the trigger event and scan records occurring immediately after the trigger event.

n = (SCANSIZ - 1 - TRIGPOS) = -1

For example, SEND SCANUM retrieves 0,499.

Scan Record TRIGPOS (newest record)

Scan Record (TRIGPOS -1)

Scan Record (TRIGPOS -2)

Scan Record (TRIGPOS -3)

.
.
.

Scan Record 2

Scan Record 1 (first Scan Record after Trigger Event)

Scan Record 0 **TRIGGER EVENT**

SEND SCANUM
Send Scan Number Information

C: 0 < TRIGPOS < SCANSIZ.	The burst scan buffer contains the trigger event and data recorded both before and after the trigger event. n = (SCANSIZ - 1 - TRIGPOS) For example, with TRIGPOS = 200, SEND SCANUM retrieves -299,200.

Scan Record TRIGPOS	(newest record)
Scan Record (TRIGPOS -1)	
Scan Record (TRIGPOS -2)	
Scan Record (TRIGPOS -3)	
•	
•	
Scan Record 2	
Scan Record 1	
Scan Record 0	**TRIGGER EVENT**
Scan Record -1	
Scan Record -2	
Scan Record -3	
•	
•	
Scan Record -n	(oldest record)

D: TRIGPOS > SCANSIZ. The burst scan buffer contains only data recorded after the trigger event (not the trigger event itself).

For example, with TRIGPOS set to 779, SEND SCANUM retrieves 280,779, indicating that the trigger event record has been written over by subsequent scan data and that the buffer contains scan records 280 through 779.

```
Scan Record TRIGPOS                          (newest record)

Scan Record TRIGPOS -1

Scan Record TRIGPOS -2

Scan Record TRIGPOS -3

       .
       .
       .

Scan Record (TRIGPOS - (SCANSIZ - 1))  (oldest record)

- - - - - - - - -          (end of buffer)
       .
       .
                                              TRIGGER EVENT
```

SCANSIZ can be 0 only if no channels have been defined.

When a Fast A/D Converter starts burst scanning in response to a START BSCAN command, both numbers returned by SEND SCANUM are zero. These numbers are also cleared by a RESET ALL command.

During burst scanning, the host can retrieve only the newest burst data record with the SEND BSCAN command.

Examples

After receiving the command:

```
SEND SCANUM(65)
```

Helios Plus responds with:

```
-19
33
```

This response signifies that Fast A/D Converter 65 has 53 scan records available (19 before the trigger event, 33 after the trigger event, and 1 taken at the time of the trigger event.)

NOTE

Before changing channel definitions for a Fast A/D Converter that has stopped burst scanning, always use the SEND BSCAN command to retrieve burst scan information. All burst scan data is discarded as soon as a channel definition command is received by a Fast A/D Converter that has stopped burst scanning.

SEND SINTERVALOVR
Send Burst Scan Interval Overrun Information

Format

```
SEND SINTERVALOVR
```

Syntax Diagram

SEND SINTERVALOVR

Description

SEND SINTERVALOVR retrieves the number of the Fast A/D Converter that had a scan interval overrun. These overruns occur when a burst scanning Fast A/D Converter cannot acquire scan records within the interval specified by the SCANINTERVAL parameter of the DEF BSCAN command.

The response to each SEND SINTERVALOVR is a Fast A/D Converter number, followed by an EOL (end-of-line) sequence. For multiple converters with overruns, the command must be reissued. Each response then represents the next Fast A/D Converter in ascending numerical order (not in the order of overrun occurrence.) A first response of only an EOL sequence signifies that there were no overruns. EOL-only responses following valid converter number responses signify that there are no additional converter overruns.

The Fast A/D Converter number is returned as an ASCII string of 1 or 2 digits, 0 through 95, identifying a valid converter, followed by an end-of-line sequence.

NOTE

Before changing channel definitions for a Fast A/D Converter that has stopped burst scanning, always use the SEND BSCAN command to retrieve burst scan information. All burst scan data is discarded as soon as a channel definition command is received by a Fast A/D Converter that has stopped burst scanning.

Example

After receiving the command:

```
SEND SINTERVALOVR
```

The following response means that Fast A/D Converter 50 could not acquire a scan record in the specified time.

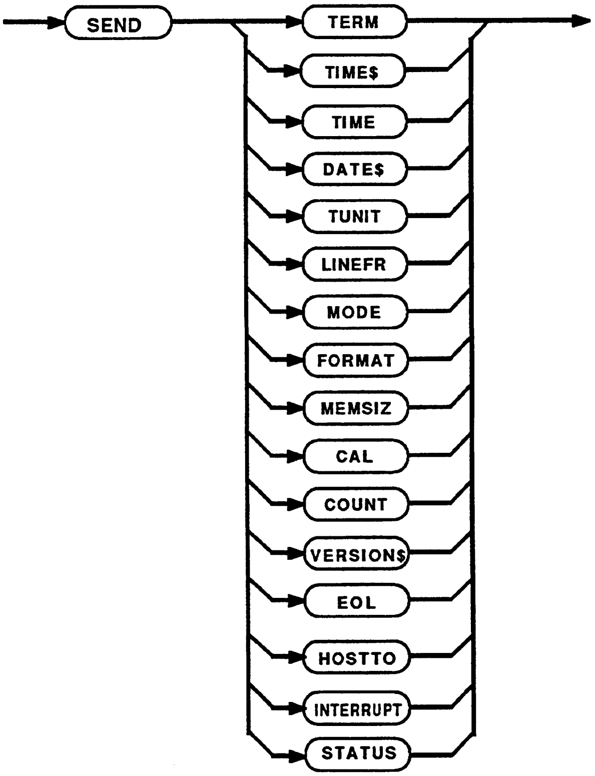
```
50
```

SEND System Variable
Send Value of System Variable

Format

```
SEND <system variable>
```

Syntax Diagram



SEND System Variable
Send Value of System Variable

Description

The value contained in <system variable> is returned to the host.

The system variables whose values can be returned are shown in the syntax diagram (above). The format in which each variable is returned is shown below. Each returned value is followed by an end-of-line sequence.

CAL (A/D auto-calibration)	"on" or "off"
COUNT (Channel count)	"on" or "off"
DATE$ (In day-Month-year)	dd-Mmm-yy
EOL (End-Of-Line Character(s))	ASCII character code(s)
FORMAT (Of Measurement Values)	"decimal", "hex" or "binary" "xascii", "xbinary", "xdecimal", "xhex"
HOSTTO (Host Time Out Interval) (in seconds)	sss
INTERRUPT (Host Alarm Interrupt)	"on" or "off"
LINEFR (Line Frequency in Hz)	"50","60", or "400"
MEMSIZ (Memory Size in Bytes)	1 to 5 digits
MODE (Operating Mode)	"term" or "comp"

STATUS (System Status Code) (see information below)	1 to 5 digits
TERM (End-Of-Response Indicator)	"on" or "off"
TIME$ (In hours:minutes:seconds)	hh:mm:ss
TIME (In ms since midnight)	8 digits
TUNIT (Temperature Units)	"kelvin", "celsius", "fahrenheit", "rankine"
VERSION$ (Firmware version)	

```
Helios Plus  Version 1.0 Software by John Fluke Mfg. Inc.
```

System Status Codes

The system status code (STATUS) returns 1 to 5 digits that represent a 16-bit number. Refer to the SEND STATUS example presented later in this subsection. Bit descriptions are as follows:

NOTE

> Normally, status information can be retrieved only by using the SEND STATUS command. If you need more immediate notification of a status change, set the system variable INTERRUPT to ON. Any change in the value of STATUS is then sent via an interrupt. As appropriate, other commands can then be used to retrieve additional information.

Bit Description

0 Host off line. When this bit is set, the timeout set with HOSTTO has expired. Alarms that occurred subsequent to this timeout were sent to the Printer Port.

1 Alarm line set. This bit is set when the alarm line is set.

2 Alarm buffer contains information. This bit is set when a new alarm is recorded in the alarm buffer and cleared when all alarms are removed.

3 Alarm buffer overflow. This bit is set on an alarm buffer overflow and cleared when alarms are removed from the buffer.

4 Trigger event status (Burst Scan Mode.) This bit is set when a trigger event occurs on a Fast A/D Converter. Additional information about the trigger event can be determined with the following commands:

SEND TRIGFAD — Determine the Fast A/D Converter that encountered the trigger event.

SEND BSCANIPS — Determine if scanning is continuing for that Fast A/D Converter. This is possible after a trigger event depending on the value of TRIGPOS set in the DEF BSCAN command.

SEND TRIGCHAN	Identify the Fast A/D Converter channel that caused the trigger event.

Bit 4 is cleared in any of the following ways:

- o All Fast A/D Converters having had a trigger event are identified (in response to the SEND TRIGFAD command).
- o All Fast A/D Converters having had a trigger event have started a new burst scan.
- o A RESET ALL command is sent.

5 Burst scan timing problem. This bit is set during burst scanning if a Fast A/D Converter encounters an overrun (cannot acquire scan records in the time allowed.) Use the SEND SINTERVALOVR command to identify the Fast A/D Converter that encountered this problem.

Bit 5 is cleared in any of the following three ways:

- o All Fast A/D Converters having had an overrun have started a new burst scan.
- o The response to the SEND SINTERVALOVR command is complete (all Fast A/D Converters having had an overrun are identified).
- o RESET ALL command is sent.

6,7 Spare.

8 Scan buffer 0 contains information. This bit is set whenever scan buffer 0 contains one or more records and cleared when all records have been removed from scan buffer 0.

9 Scan buffer 0 overflow. This bit is set when scan buffer 0 overflows and cleared as soon as records are removed from the buffer.

10 Scan buffer 1 contains information. This bit is set whenever scan buffer 1 contains one or more records and cleared when all records have been removed from scan buffer 1.

11 Scan buffer 1 overflow. This bit is set when scan buffer 1 overflows and cleared as soon as records are removed from the buffer.

12 Scan buffer 2 contains info. This bit is set whenever scan buffer 2 contains one or more records and cleared when all records have been removed from scan buffer 2.

13 Scan buffer 2 overflow. This bit is set when scan buffer 2 overflows and cleared as soon as records are removed from the buffer.

14 Scan buffer 3 contains info. This bit is set whenever scan buffer 3 contains one or more records and cleared when all records have been removed from scan buffer 3.

15 Scan buffer 3 overflow. This bit is set when scan buffer 3 overflows and cleared as soon as records are removed from the buffer.

Examples

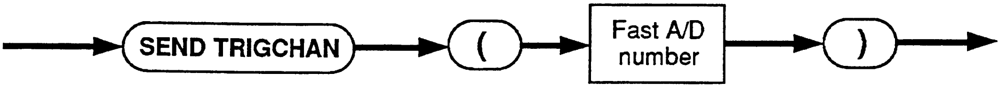
SEND DATE$	Response: 25-Apr-86
SEND TIME	Response: 42423459
SEND TIME$	Response: 15:31:27
SEND FORMAT	Response: xascii
SEND STATUS	A response of '1028' means, for example, that bits 2 and 10 are set, indicating that both the alarm buffer and scan buffer 1 contain information. A response of '16' means that bit 4 is set, indicating that a trigger event has occurred on a Fast A/D Converter.

SEND TRIGCHAN
Send Channel Causing Trigger Event

Format

```
SEND TRIGCHAN(<Fast A/D Converter number>)
```

Syntax Diagram



Description

SEND TRICHAN retrieves the number of the channel that caused the trigger event on the specified Fast A/D Converter. This information is cleared when a START BSCAN or RESET ALL command is sent. Otherwise, the information is saved and can be retrieved with subsequent SEND TRIGCHAN commands.

Normally, the response is a 1, 2, or 3-digit ASCII string, ranging from 0 through 989, followed by an EOL (end-of-line) sequence. If an external trigger input causes the trigger event, a special identifying response of -1 is returned.

NOTE

Before changing channel definitions for a Fast A/D Converter that has stopped burst scanning, always use the SEND BSCAN command to retrieve burst scan information. All burst scan data is discarded as soon as a channel definition command is received by a Fast A/D Converter that has stopped burst scanning.

Example

After receiving the command:

```
SEND TRIGCHAN(30)
```

Helios Plus responds with '308' if the channel that caused the trigger event on Fast A/D Converter was number 308.

SEND TRIGFAD
Send Trigger Event Information

Format

SEND TRIGFAD

Syntax Diagram

Description

SEND TRIGFAD retrieves the number of the Fast A/D Converter that encountered a trigger event. If multiple Fast A/D Converters experienced multiple trigger events, each response is in ascending order of Fast A/D Converter number (not the order in which the trigger events occurred.)

If a START BSCAN command has been issued for a Fast A/D Converter, SEND TRIGFAD does not return a number for that converter until a related trigger event occurs.

Each response comprises a Fast A/D Converter number and an EOL (end-of-line) sequence. An EOL response with no number signifies that no trigger event occurred, that no Fast A/D Converters exist, or that all trigger event information has been retrieved.

NOTE

Before changing channel definitions for a Fast A/D Converter that has stopped burst scanning, always use the SEND BSCAN command to retrieve burst scan information. All burst scan data is discarded as soon as a channel definition command is received by a Fast A/D Converter that has stopped burst scanning.

Example

After receiving the command:

```
SEND TRIGFAD
```

Helios Plus identifies Fast A/D Converter 30 as having had a trigger event by responding with:

```
30
```

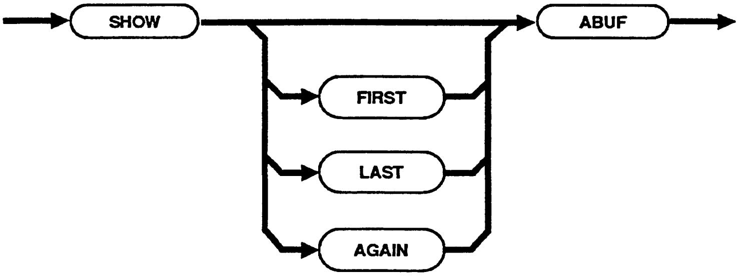
SHOW ABUF SHOW FIRST ABUF SHOW LAST ABUF SHOW AGAIN ABUF

Show Alarm Buffer Data

Format

```
    SHOW ABUF
    SHOW FIRST|LAST|AGAIN ABUF
```

Syntax Diagram



SHOW ABUF SHOW FIRST ABUF SHOW LAST ABUF SHOW AGAIN ABUF
Show Alarm Buffer Data

Description

This command can be used in any of the following ways:

- SHOW ABUF

  This is a relative command that sends the next newer record in the alarm buffer, depending on the position of the previously sent record. The oldest record is sent if no other SHOW ABUF command has been used or if all the records have already been sent. If a SHOW command has been used already, the starting point for the SHOW ABUF next record may have changed. For example, if SHOW ABUF follows SHOW FIRST ABUF, the next record will be one newer than the oldest.

  The SHOW ABUF command is identical to the SEND ABUF command, except that it does not remove the retrieved record from the alarm buffer. The format of the returned record depends on the setting of the system variable FORMAT.

  If the last buffer record has been transmitted, the SHOW command returns an end-of-buffer response.

- SHOW FIRST ABUF

  Send the oldest record in the ABUF.

- SHOW LAST ABUF

  Send the newest record in the ABUF.

- o SHOW AGAIN ABUF

  Repeat the most recently shown alarm buffer record. If the applicable record has already been overwritten with new data, the following error message is returned:

  "No previous record available"

The following chart represents an alarm buffer containing ten records and illustrates which records are accessed with the various SHOW commands.

```
                           Alarm
                           Buffer
                           (ABUF)
                        ________________
SHOW LAST ABUF-------------> Record                  .
    (actual newest)        Record                  .
                             .    <---------------SHOW ABUF
                             .              (relative,
                             .              next newer)
                             .                       .
                             .    <---------SHOW AGAIN ABUF
                             .              (relative,
                             .              repeat previous)
SHOW FIRST ABUF------------> Record                  .
    (actual oldest)     ________________
```

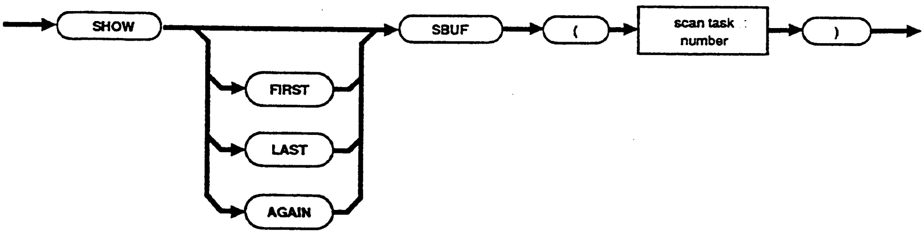
SHOW SBUF SHOW FIRST SBUF SHOW LAST SBUF SHOW AGAIN SBUF

Show Scan Buffer Data

Format

```
SHOW SBUF(<scan buffer number>)
SHOW FIRST|LAST|AGAIN SBUF(<scan buffer number>)
```

Syntax Diagram



Description

Various forms of the SHOW SBUF command are listed below. Note that scan buffer number (1) is shown as an illustration, but numbers (0), (2), or (3) could also be used.

SHOW SBUF SHOW FIRST SBUF
SHOW LAST SBUF SHOW AGAIN SBUF
Show Scan Buffer Data

- SHOW SBUF(1)

  Send the next record in the scan buffer. The record sent is the next newer record (beginning with the oldest record) if no other SHOW command has been used. The record accessed may also be relative, in that a record previously accessed with one of the SHOW SBUF commands becomes the starting point in defining the next newer record. For example, if SHOW SBUF follows SHOW FIRST SBUF, the next record will be one newer than the oldest.

  The SHOW SBUF command does not remove the retrieved record from the scan buffer, but is otherwise identical to the SEND SBUF command

  If the last buffer record has been transmitted, the SHOW SBUF command returns an end-of-buffer response. Returned record format depends on the system variable FORMAT setting.

- SHOW FIRST SBUF(1)

  Send the oldest record in SBUF(1).

- SHOW LAST SBUF(1)

  Send the newest record in SBUF(1).

o SHOW AGAIN SBUF(1)

Repeat the most recently shown scan buffer record. If the applicable record has already been overwritten with new data, the following error message is returned:

"No previous record available"

The following chart represents a scan buffer containing ten records and illustrates which records are accessed with the various SHOW commands.

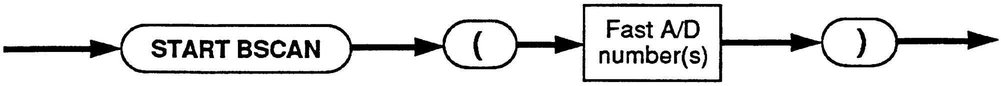
```
                                   Scan
                                  Buffer
                                  (SBUF)
                              _______________
SHOW LAST SBUF-------------> Record                .
    (actual newest)          Record                .
                               .    <--------------SHOW SBUF
                               .          (relative,
                               .          next newer)
                               .                   .
                               .    <--------SHOW AGAIN SBUF
                               .          (relative,
                               .          repeat previous)
SHOW FIRST SBUF------------> Record                .
    (actual oldest)           _______________
```

START BSCAN
Start Burst Scanning

Format

```
START BSCAN(<Fast A/D Converter numbers>)
```

Syntax Diagram



Description

START BSCAN instructs one or more Fast A/D Converters to start scanning in the Burst Scan Mode, according to the parameters specified in the DEF CHAN and DEF BSCAN commands. If START BSCAN is sent before a DEF BSCAN command, the Fast A/D Converter uses the default parameters (see DEF BSCAN command description.)

Multiple Fast A/D Converters are instructed to start burst scanning in the numerical order listed in the START BSCAN command.

START BSCAN may refer to a list of Fast A/D Converters, some of which are already in Continuous Scan Mode, Burst Scan Mode, or some combination of both modes. If START BSCAN refers to a Fast A/D Converter that is in continuous scan mode, burst scanning is started. If START BSCAN refers to a Fast A/D Converter that is already burst scanning, no change is made to that converter.

Example

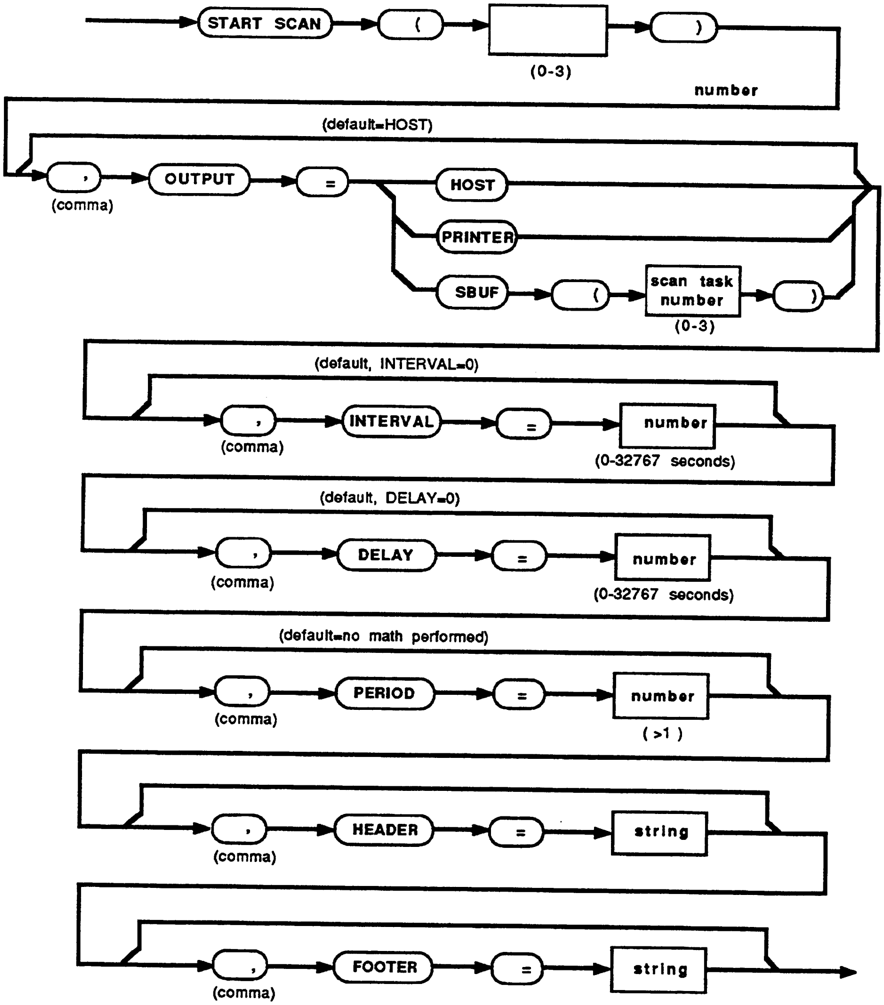
The following example instructs Fast A/D Converter 35 (channels 350 through 389) to start burst scanning:

```
START BSCAN(35)
```

Format

```
START SCAN(<scan task number>) [,OUTPUT=HOST | PRINTER |
SBUF(<scan task number>)] [,INTERVAL=<seconds>]
[,DELAY=<seconds>] [,PERIOD=<number of scans>]
[,HEADER=<string>] [,FOOTER=<string>]
```

Syntax Diagram



START SCAN
Start Scan Task

Description

This command starts execution of a scan task. A scan task is a program that executes on Helios Plus and measures a set of channels on a periodic basis. Measurement data from a scan task is sent to an output device, which may be the Helios Plus host computer port, the Helios Plus printer port, or a scan buffer in Helios Plus memory.

NOTE

> If the scan task has been defined using math functions (MAX, MIN, AVG, TOT), the OUTPUT parameter can only to set equal to SBUF(<scan task number>).

Parameters includable in the START SCAN statement are:

- INTERVAL

  This parameter establishes the time (in seconds) between successive scans. If no INTERVAL is specified, continuous scanning occurs.

- DELAY

  This parameter specifies the time in seconds before the initial scan is executed.

- PERIOD

  This parameter sets the number of scan cycles to be used by the math functions MIN, MAX, TOT, and AVG. To allow for meaningful computations, the number of scan cycles specified should always be greater than 1. If no PERIOD parameter is defined, no MIN, MAX, TOT, or AVG functions are performed.

- HEADER
  FOOTER

  Header and footer strings are used with the printer port only. Each string contains a maximum of 128 characters formed from any combination of:

  1. Quoted character strings. Quote marks do not count in the 128-character limit.

  2. ASCII character decimal codes (not in quotes). Each code counts as one character.

  In addition, the header only can be appended with current date and time with:

  /t

  /d

  A semi-colon (;) must separate all character strings, character codes, and time/date specifiers. Semi-colons are not part of the 128-character limit.

  If HEADER is not specified in the START SCAN statement, the default header remains in effect. The default shows scan group, time, and date, as described under SEND SBUF.

There are three commands for using scan tasks: DEF SCAN, START SCAN and STOP SCAN. The DEF SCAN command assigns a set of channels to a scan task; START SCAN starts the execution of a scan task; and STOP SCAN terminates the execution of a scan task.

For a scan buffer, definitions are handled with the DEF SBUF command, and reads are handled with the SEND SBUF command or one of the SHOW SBUF commands.

Scan tasks must be defined before they can be executed. When using a scan buffer to store scan data, the scan task must be defined before the scan buffer is defined, because defining a scan task erases any existing definition of the associated scan buffer.

Up to four scan tasks may be defined and executed. Scan tasks are identified by their scan task number, which may be 0, 1, 2, or 3. When executing multiple scan tasks simultaneously, scheduling conflicts are resolved by scan task priority. The scan task priority is determined by the scan task number, with lower numbers having higher priority.

The priority of concurrent scans is based on the scan task number. Only one task runs at a time, with lower numbers having precedence.

NOTE

> SCAN TASK(0), with no interval specified, runs continuously and will prevent execution of any other tasks.

Example

The command string

```
START SCAN(1),OUTPUT=SBUF(1),INTERVAL=60,
HEADER="GENERATOR NUMBER 1";/t;/d
```

starts scan task(1), sends data to scan buffer(1), schedules scans every 60 seconds, and includes a header showing GENERATOR NUMBER 1, the time, and the date.

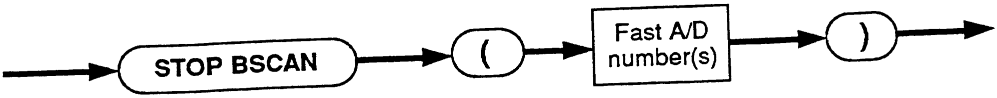
See Also

DEF SCAN
DEF SBUF
SHOW SBUF
SEND SBUF
STOP SCAN

Format

```
STOP BSCAN(<Fast A/D Converter numbers>)
```

Syntax Diagram



Description

When Helios Plus is in Burst Scan mode, this command can be used with single or multiple Fast A/D Converters. If a single converter number is mentioned, the mainframe instructs that Fast A/D Converter to stop burst scanning and freeze burst data. If multiple Fast A/D Converter numbers are specified, the mainframe notifies each Fast A/D Converter (in the order listed, from left to right) to stop burst scanning and freeze burst data.

STOP BSCAN is ignored if Helios Plus is not in Burst Scan mode.

Example

The following example shows how to stop burst scanning on several Fast A/D Converters:

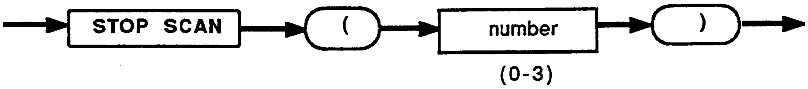
```
STOP BSCAN(30,45,80)
```

STOP SCAN
Stop Scan Task

Format

```
STOP SCAN(<scan task number>)
```

Syntax Diagram



Description

This command allows the user to stop a previously started scan task.

Example

The command:

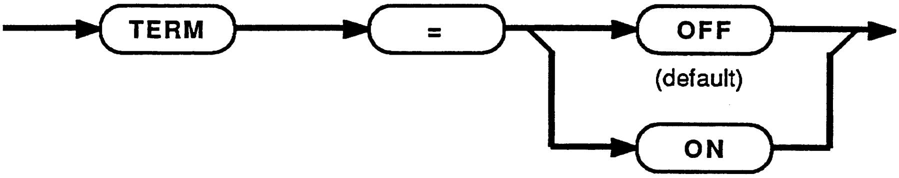
```
STOP SCAN(2) <CR>
```

stops scan task number 2.

Format

```
TERM = ON | OFF
```

Syntax Diagram



Description

The setting of this system variable is only effective in computer mode. When TERM is set to ON, a "!" character and <EOL> are appended to the data returned by the SEND and LIST commands.

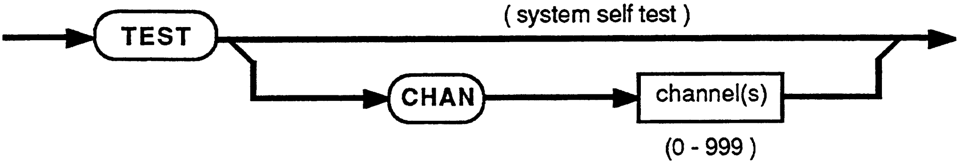
NOTE

All examples in this manual assume that the system variable TERM is in the default setting of OFF.

Format

```
TEST [CHAN(<channels>)]
```

Syntax Diagram



Description (General)

This command causes the Front End to perform a system-wide self test or a self test on designated channels. The Front End will operate if a self-test error has been reported; however, operation may be unreliable.

If specific channels are not indicated, a system-wide self test is performed. When a system-wide self test is performed, the interface hardware (i.e., the read only memory (ROM), random access memory (RAM) and the serial link universal asynchronous receiver transmitter (UART)) are checked.

If a self test is performed on a channel list, the designated channels are tested, implied hardware assemblies are put through a test procedure, and an error code or message is returned.

The format in which error messages are returned depends on the MODE system variable setting.

Description (Computer Mode)

In computer mode, self-test failures are reported in the format:

```
? <error number><EOL>
```

where <error number> is one of the error numbers listed in Section 8.

Description (Terminal Mode)

In terminal mode, self-test failures are reported in the format:

```
? <error message><EOL>
```

where <error message> is one of the error messages listed in Section 8.

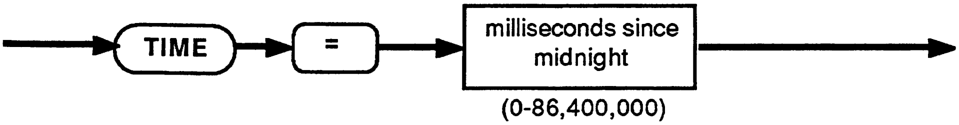
See Also

Section 8: Error Messages

Format

```
TIME = <number of milliseconds since midnight>
```

Syntax Diagram



Description

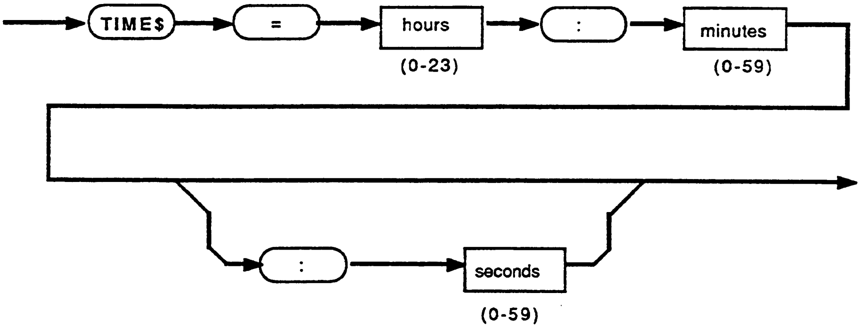
This variable allows the user to assign a starting time to the internal clock representing the number of milliseconds since midnight.

Since the system has only one internal clock, setting TIME changes the TIME$ setting.

Format

```
TIME$ = <hours>:<minutes>[:<seconds>]
```

Syntax Diagram



Description

This variable is used to set the internal time in the more conventional form of hours (0-23), minutes (0-59) and seconds (0-59). If <seconds> is omitted, it is assumed to be 0.

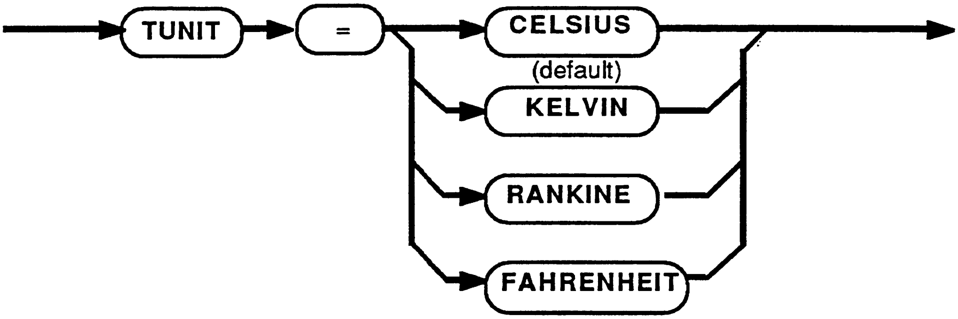
Since the system has only one internal clock, setting TIME$ changes the TIME setting.

TUNIT
Set Temperature Units System Variable

Format

```
TUNIT = CELSIUS | FAHRENHEIT | KELVIN | RANKINE
```

Syntax Diagram



Description

This variable selects the temperature units in which temperature readings (from thermocouples and RTDs) are returned. Four options are available: Celsius, Fahrenheit, Rankine and Kelvin. The instrument defaults to Celsius.

Section 6
Measurement Reference

CONTENTS

Section 6A
Analog Output
UNIPOLV, BIPOLV, DCOUT, PVOUT

INTRODUCTION

The Front End can produce two types of analog outputs: direct voltage or dc current. This section explains use of the Front End mainframe and required option assemblies in generating these two outputs. Additional information can be found in Section 3B in the discussion of the Analog Output (-170) assembly.

Direct Voltage Output

A direct voltage output is available to control devices requiring a continuously variable voltage. The Front End Analog Output can supply a unipolar source (0V to 10V) or a bipolar source (-5V to 5V). Applications might include voltage controlled power supplies, process controllers, or 0 to 5 volt actuators.

Current Output

A current output allows for control of devices requiring a continuously variable current. Outputs from 4 mA to 20 mA are available. Applications could include current-controlled power supplies, process controllers, or 4 to 20 mA actuators.

REQUIRED HARDWARE

Applications involving analog voltage or current outputs require the Analog Output (-170) assembly. This assembly can be installed in either the Front End mainframe or the 2281A Extender Chassis. Each -170 assembly provides four independent output channels with three sets of output terminals per channel.

Installation for any of the required assemblies is detailed in Section 3B of this manual. Reference each assembly by its option number (-170).

As an overview, installation requires the following actions:

o Address Selection

Address selection switches establish the hundreds and tens designations for the first channel in a range of four successive channels. These switches are accessible through the rear panel of the Analog Output and can be set at any time.

o Wiring Connections

For each channel, the connections determine the type of output (direct current control, current control as a process variable, bipolar voltage, or unipolar voltage).

o Physical Installation

The Analog Output interfaces with the serial link and, therefore, can be installed in either the Front End mainframe or the 2281A Extender Chassis.

USING THE COMMANDS

General

The DEF CHAN statement is used in defining a channel (or group of channels) for one of four types of analog output. In this statement, the channels must fall within the range set by the address selection switches. Also, the defined channel type should correspond to the Analog Output assembly wiring configuration.

For example, if the channel selection switches are set for 1 and 0 (meaning 100), and unipolar voltage wiring connections have been made, the following definition can be made for a range of four channels:

```
DEF CHAN(100..103) = UNIPOLV
```

Using the commands requires two steps. First, the channel number(s) and the associated channel type (BIPOLV, UNIPOLV, DCOUT, or PVOUT) are defined. A DEF CHAN statement is used for this purpose.

NOTE

UNIPOLV is the default setting.

Second, the output value is assigned separately using the CHAN statement. The assigned output value corresponds directly with the output voltage or current when any of the following modes of operation are selected:

- o BIPOLV (bipolar voltage from -5 to +5V)
- o UNIPOLV (unipolar voltage from 0 to +10V)
- o DCOUT (current from 4 to 20 mA)

When the PVOUT mode is selected, the assigned value represents percent of range (0 to 100), corresponding to a span from 4 mA to 20 mA. Some examples follow:

Assigned	Actual
0	4 mA
50	12 mA
100	19.995 mA

Any assigned value less than 0 causes an illegal value error message (code 01). Any assigned value greater than 100 causes 19.995 mA to be set.

Command Examples

Use the following examples to familiarize yourself with the Analog Output commands. First, define channels 100 and 101 for dc current output:

```
DEF CHAN(100..101) = DCOUT
```

Next, set the output level for these channels to 19.5 mA:

```
CHAN(100..101) = 0.0195
```

Now, change the output level to 16 mA:

```
CHAN(100..101) = 0.016
```

Set the remaining channels for this Analog Output for process variable output. This specifies output current or voltage as a percentage of the available range.

```
DEF CHAN(102..103) = PVOUT
```

Next, set the output level to 60%. For current, 60% of range is 13.6 mA. For voltage, 60% is 6V for UNIPOLV and 1V for BIPOLV.

```
CHAN(102..103) = 60
```

Finally, return the set output for any of these channels.

```
SEND CHAN(103)
```

The response for this channel would be:

```
6.00000E+01
```

PUTTING IT ALL TOGETHER

The following example can be typed in directly from a terminal (TERM mode) or included in a computer program (COMP mode). Refer to Section 4 under "Operating the Front End from a Terminal/Computer" for more detailed information. Also, Appendix 9A provides guidelines for use with specific types of host computers.

```
MODE = TERM
FORMAT = DECIMAL
DEF CHAN (0..3) = BIPOLV
CHAN (0..3) = 4.0
SEND CHAN (0..3)
```

This example assumes that the Analog Output address is set to 00. It commands the Front End to operate in terminal mode, sets system variable FORMAT, and defines channels 0 through 3 for an analog output voltage in the range of -5V to +5V. The CHAN command then sets the output to +4.0V.

The SEND CHAN command is used to monitor the set value received by the Analog Output assembly. However, this command does not indicate the output actually being made from the Analog Output. To verify the output, connect a measuring device (such as a DMM) to the actual outputs.

If the SEND CHAN response is "9.99999E+37", an error has occurred. Enter the LIST ERROR command to retrieve the associated error message. If necessary, refer to Section 8 for additional error message information.

Section 6B
Current Measurement
DCIN

INTRODUCTION

About This Section

This section explains use of the Front End and associated option assemblies for obtaining current readings. Detailed information about the physical installation of the Front End mainframe and related options is provided in Section 3.

Current Measurement

Current input measurements use an 8-ohm shunt resistor and cover the range from 0 to 64 mA on a maximum of 20 channels. This range covers the useful range of standard current transmitters. Since this type of transmitter is often used to translate the output of another transducer (pressure, fluid flow, etc) to a current output, the current measurement function can serve a multitude of applications.

Standard current transmitters output a current value proportional to the physical parameter being measured. This sensor type usually outputs a value of from 1 to 5 mA, 4 to 20 mA, or 10 to 50 mA across its measurement range. Scaling and conversion used by the Front End are optimized to measure current values within these ranges.

REQUIRED HARDWARE

Using the High Performance A/D Converter

Current measurements can be made with the following configuration of option assemblies:

-161	High Performance A/D Converter
-162	Thermocouple/DC Volts Scanner
-171	Current Input Connector

The High Performance A/D Converter (-161) provides high accuracy analog to digital conversion of scanner inputs. At least one a/d converter must be installed in the Front End when analog measurements are being made.

The Thermocouple/DC Volts Scanner (-162) is a plug-in, 20-channel thermocouple and multi-voltage range relay scanner. The scanner links the High Performance A/D Converter to external measurement points.

The Current Input Connector (-171) routes a maximum of 20 current input channels to the Thermocouple/DC Volts Scanner.

Installation for any of these option assemblies is detailed in Section 3B of this manual. Reference each option assembly by its option number (-161, -162, or -171).

As an overview, installation of the High Performance A/D Converter current measurement configuration requires the following actions:

- Addressing

  Addresses are determined by the both the position of the Thermocouple/DC Volts scanner relative to the High Performance A/D Converter (-161) and the address switch settings on that a/d converter.

- Wiring Connections

  For each channel, HI and LO connections must be made so that current flows into the HI terminal and out of the LO terminal.

- Physical Installation

  Each Current Input Connector attaches to a Thermocouple/DC Volts Scanner. Each connector/scanner interfaces with the serial link and can be installed in either the Front End mainframe or the 2281A Extender Chassis.

Using the Fast A/D Converter

Current measurements can also be made with the Fast A/D Converter. The required hardware is then:

-165	Fast A/D Converter
-176	Voltage Input Connector
641449	8-ohm current shunt resistors, Fluke Part Number

The Fast A/D Converter provides accurate, high speed analog to digital conversion of inputs from the connector. The shunt resistors are placed across input terminals on the connector as required for differential measurements. A maximum of 20 differential inputs can be handled by the Fast A/D Converter.

The Fast A/D Converter current measurement configuration requires the following actions:

o Addressing

Set the addresses using the switches on the -165 Fast A/D Converter. Each differential pair must include a second channel that is exactly 20 addresses higher than the first. Therefore, differential inputs can be set up for address pairs 0/20 through 19/39. A differential measurement is performed when either channel is addressed.

o Wiring Connections

Make differential connections to the -176 Voltage Input Connector. Specific information is provided later in this section.

- Physical Installation

  Each -176 Voltage Input Connector attaches directly onto a -165 Fast A/D Converter. Each -165 Fast A/D Converter/-176 Voltage Input Connector pair can be installed in a 2287A Front End or a 2281A Extender Chassis. Each channel that is not used for dc current measurement remains available for dc voltage measurement connections (-175 and -176) and or thermocouple measurement connections (-175 only).

INPUT CONNECTIONS

Current transmitter outputs referenced to ground or some other potential can be used, with the limitation noted below.

NOTE

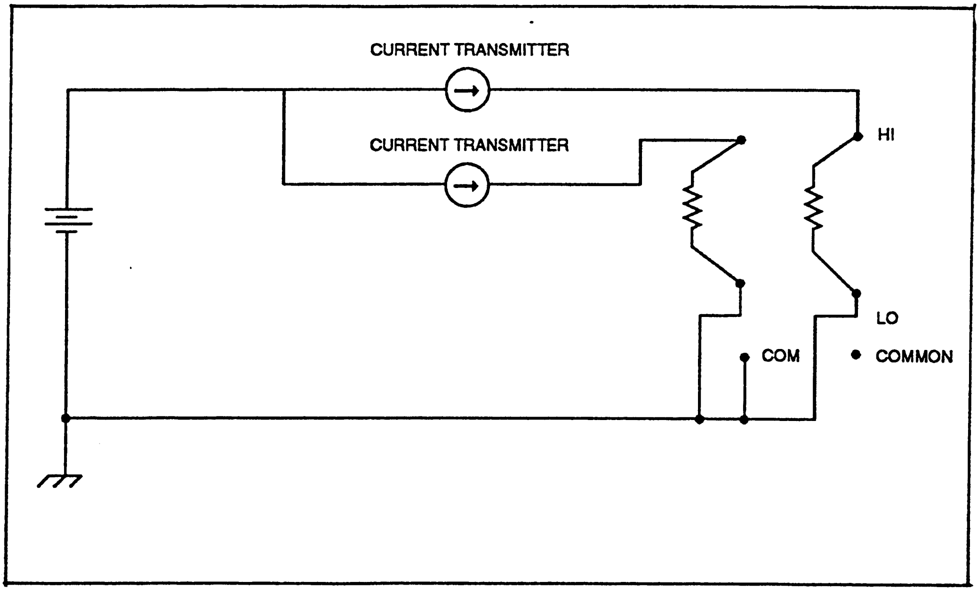
> The -165 Fast A/D Converter can be used only with a COMMON voltage level that is within +/-10V of both HI and LO voltage levels. If no such common level is available, the High Performance A/D Converter configuration must be used. If such a common level is available, proceed with the following steps.

1. Connect the common level to one of the COMMON terminals on the Voltage Input Connector.

2. Connect shunt resistors between HI and LO inputs of each channel on the Voltage Input Connector.

3. Attach current transmitter outputs to HI and LO as shown below:

Current Measurement



USING THE COMMANDS

General

One input range (64 mA) is available for current inputs. Each current input channel (or group of channels) is specified with the DCIN parameter in a DEF CHAN statement.

```
DEF CHAN(0..5, 8, 9) = DCIN
```

An interpolation table, relating dc current input values to output values in other units, can be called from this definition, allowing for conversion from units of current to other engineering units.

The measurement is made with the SEND CHAN command.

Command Examples

Let's augment the previous DEF CHAN statement with an interpolation table. First, establish the table as a sequence of input/output value pairs. An actual output is linearly interpolated (or extrapolated) from values found in the table.

```
DEF TABLE(1)=<input>,<output>/<input>,<output>/  ....
```

Next, call the interpolation table from a channel definition statement.

```
DEF CHAN(0..5, 8, 9) = DCIN, CHFN=TABLE(1)
```

For example, a pressure transmitter may have a dc current output range from 4 mA (at 0 psi) to 20 mA (at 500 psi). Interpolation table values can be entered to correlate current to psi.

```
DEF TABLE(1) = 0.004,0/0.02,500
```

The table can be called for channels measuring direct voltage:

```
DEF CHAN(0..5, 8, 9) = DCIN, CHFN = TABLE(1)
```

Now, measure these channels with the SEND CHAN command:

```
SEND CHAN(0..5, 8, 9)
```

The actual unit-of-measurement (psi in this case) can be appended to the returned output value within the computer's program.

PUTTING IT ALL TOGETHER

The following example can be typed in directly from a terminal (TERM mode) or included in a computer program (COMP mode). Refer to Section 4 under "Operating the Front End from a Terminal/Computer" for more detailed information. Also, Appendix 9A provides guidelines for use with specific types of host computers.

```
MODE = TERM
FORMAT = DECIMAL
DEF CHAN(0..19) = DCIN
SEND CHAN(0)
```

This example assumes that the High Performance A/D Converter (-161) address is set to 0 and that a direct current source is connected to the HI and LO terminals for channel 0 on the Current Input Connector (-171). It commands the Front End to operate in terminal mode, defines system variable FORMAT, and defines analog input channels 0 through 19 for direct current measurement. The resulting measurement is made on channel 0 with the SEND CHAN command.

If 60 mA is measured, the returned reading is:

```
6.00000E-02
```

If the response to the SEND CHAN command is "9.99999E+37", an error has occurred. Enter the LIST ERROR command to retrieve the associated error message. If necessary, refer to Section 8 for additional error message information.

Section 6C
Digital/Status Inputs
BINARY, BCD/STATIN

INTRODUCTION

About This Section

This section explains use of the Front End and associated option assemblies in obtaining digital or status inputs. Detailed information about the physical installation of the Front End mainframe and related options is provided in Section 3.

Digital Inputs

The digital input function is a logic input for reading BCD (binary coded decimal) or straight binary data from sensors and instruments. Digital inputs can be provided by simple system instruments and by such things as position sensors and event counters.

Status Inputs

The status input function is a logic input for reading status (on/off) conditions from sensors, switches, and instruments. Status inputs are used to sample the state of one-bit inputs. They are used to determine if something is on or off, open or closed, enabled or disabled.

REQUIRED HARDWARE

Applications involving digital or status inputs require the following option assemblies:

-168	Digital I/O Assembly
-179	Digital/Status Input Connector

The Digital I/O Assembly allows the Front End to exchange information with a digital peripheral device. Four types of data exchange are possible: Alarm or Status Output; Status Input; BCD Input; and Binary Input.

The Digital/Status Input Connector can be used for the mutually exclusive functions providing BCD digital input, binary digital input, or status input information to the Front End. This connector functions as a plug-in appendage to the Digital I/O Assembly, providing up to 20 input lines.

Installation for any of the required assemblies is detailed in Section 3B of this manual. Reference each option assembly by its option number (-168 or -179).

As an overview, installation requires the following actions:

o Address Selection

With the connector detached, set the address switches on the Digital I/O Assembly. See Section 3B (-168).

o Input Format

Make jumper connections for status, binary, or BCD inputs. See Section 3B (-179).

- Handshake Connections

  For digital inputs, make handshake terminal connections for the desired handshake method. Refer to Section 3B (-179).

- Wiring Connections

  Make necessary external connections to SIGNAL and RETURN terminals.

- Physical Installation

  Attach the Digital/Status Input Connector to a Digital I/O Assembly installed in the Front End or an Extender Chassis.

USING THE COMMANDS

General

Digital/status inputs require a DEF CHAN command line to specify the channel (or range of channels) and the input type (STATIN, BCD, or BINARY).

Command Examples

Use a DEF CHAN command to define a group of twenty status input channels:

```
DEF CHAN  (200..239) = STATIN
```

Use a SEND CHAN command to monitor the status inputs:

```
SEND CHAN(200..239)
```

PUTTING IT ALL TOGETHER

The following example can be typed in directly from a terminal (TERM mode) or included in a computer program (COMP mode). Refer to Section 4 under "Operating the Front End from a Terminal/Computer" for more detailed information. Also, Appendix 9A provides guidelines for use with specific types of host computers.

```
MODE = TERM
FORMAT = DECIMAL
DEF CHAN(0..19) = STATIN
SEND CHAN (0..19)
```

This example assumes that a Digital/Status Input Connector (-179) has been installed on a Digital I/O Assembly (-168) set for address 00. It commands the Front End to operate in terminal mode, sets system variable FORMAT, and defines digital input channels 0 through 19 for one-bit status inputs. The resulting measurements are monitored by the SEND CHAN command.

Verify status input responses by shorting some otherwise disconnected input channels. For example, short Signal and Return terminals on channels 7 through 13. Also, short terminal 21, Signal to Return. Leave channels 0 through 6 and 14 through 19 open.

For channels 7 through 13, the response should be:

0.00000E+00

The response for the remaining channels should be:

1.00000E+00

If the response is "9.99999E+37", an error has occurred. Enter the LIST ERROR command to retrieve the associated error message. If necessary, refer to Section 8 for additional error message information.

Section 6D
Frequency Measurements
FREQ

INTRODUCTION

About This Section

This section explains the use of the Front End for measuring frequency. Detailed information on installing the Front End and its options is found in Section 3.

Frequency Measurement

Frequency is defined as the number of events that occur in a given time period. Frequency is expressed in hertz. Frequency measurements can be used to determine the linear or rotational speed of objects, the flow rate of fluids, and the oscillation rate of electrical signals.

Frequency is measured by counting events for a known time period. The measurement represents the average frequency observed during the sampling time. In the Front End, the measurement time is about two-thirds of a second.

A transition between one voltage state and another constitutes an event. The boundary between a high and low voltage state may be different for different types of signals. The Front End counts high-to-low transitions.

REQUIRED HARDWARE

For frequency measurements, the following is needed:

-167 Counter/Totalizer

Each Counter/Totalizer option supports six frequency measurement channels.

The installation instructions for the Counter/Totalizer are found in Section 3B of this manual. Consult these instructions as necessary.

Hardware Preparation

The Counter/Totalizer adjustments allow for measurement of a variety of signal types. The reference voltage and input deadband are adjustable. These adjustments define the high and low voltage thresholds of the input.

The Counter/Totalizer also supports an event counting (totalizing) function. Debouncers and input pull-up adjustments are provided for event counting measurements. These adjustments are normally not used when measuring frequency.

To prepare a Counter/Totalizer channel for frequency measurement, perform the following steps. Refer to Figure 6d-1.

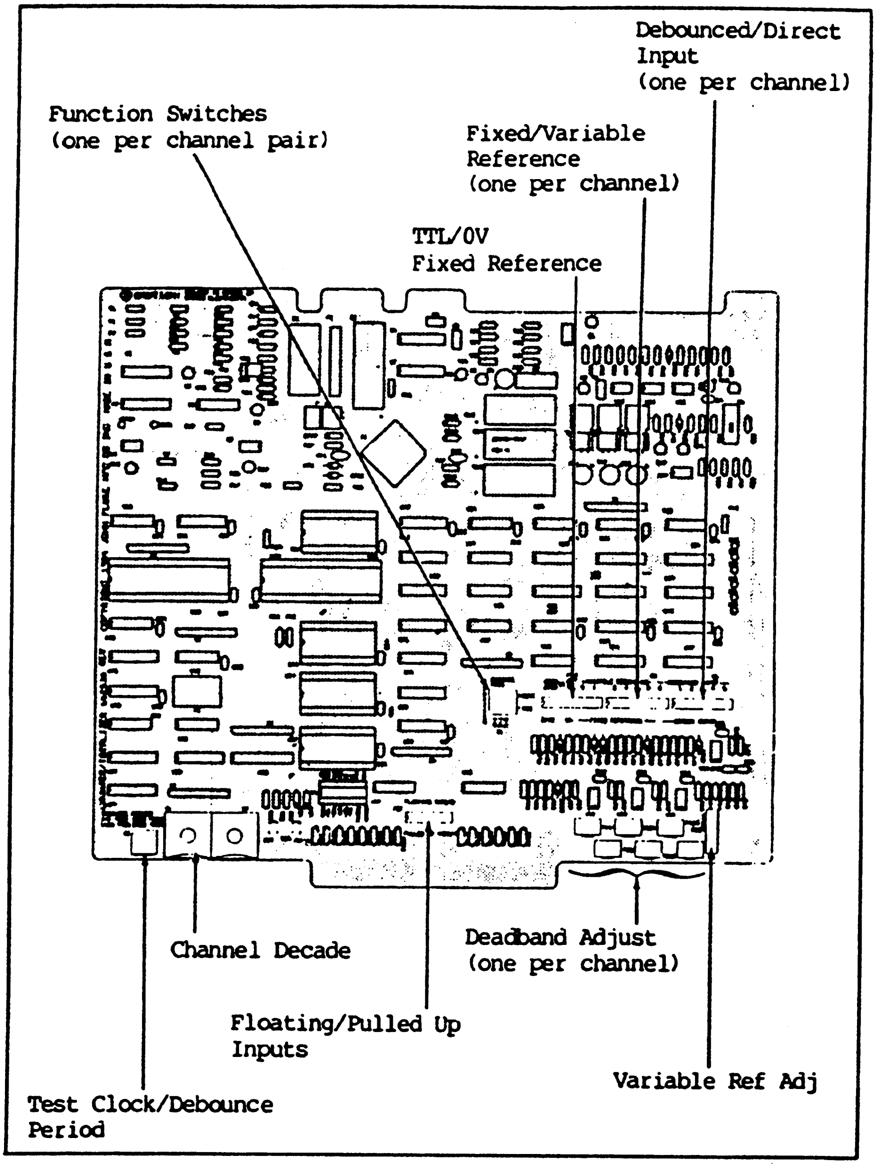


Figure 6d-1. Counter/Totalizer Adjustments

Frequency Measurements

- FREQ Function

  Move the channel function switch to the FREQ position. Notice that the channels are grouped in pairs. There are three pairs of channels: channels 0 and 1, channels 2 and 3, and channels 4 and 5. Both channels in a pair must have the same function.

- Direct Input

  Move the debounced/direct input switch to the DIRECT INPUT position. This turns off the debouncer. Since it alters the shape of the input signal, the debouncer can introduce errors during frequency measurements. Also, frequencies above 125 Hz cannot be measured with the debouncer on.

- Pulled-Up Inputs

  Move the floating/pulled-up resistor network to the FLOATING position. Pulled-up inputs are necessary when counting contact closures. They are usually not needed for frequency measurements.

- Addressing

  Set the hundreds and tens address switches to the first address for the group of six channels.

Adjustments

Figure 6d-2 shows how input voltage levels are detected by the Counter/Totalizer. The reference voltage defines the boundary between the high and low voltage states of the input signal. Deadband prevents input noise from being detected. As shown in the figure, the reference voltage and deadband define the high and low input voltage thresholds.

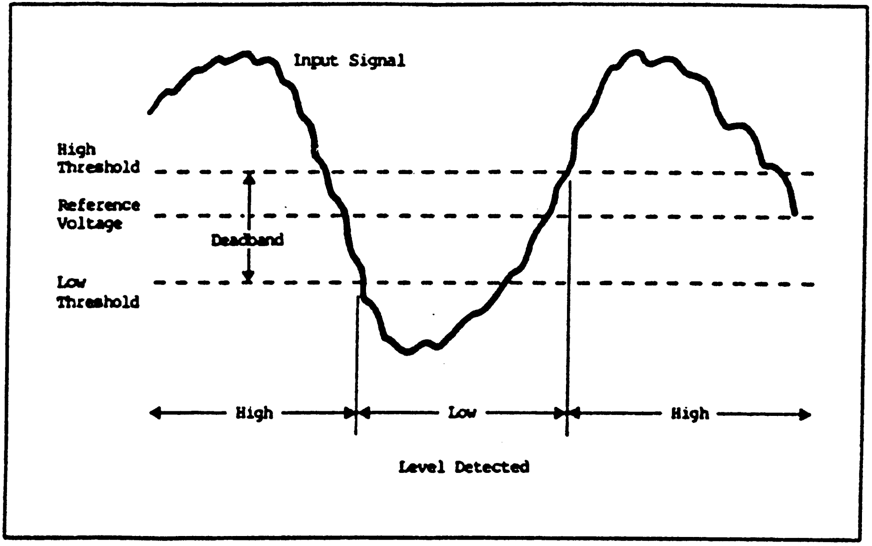


Figure 6d-2. Input Level Detection

On the Counter/Totalizer assembly, the reference voltage and deadband are adjustable. For each channel, a switch selects either a fixed or variable reference voltage. There is one variable reference on the assembly, adjustable from -10 to +10 volts. There is also one fixed reference, which is selectable for 0 volts or 1.4 volts (TTL level). The deadband is adjustable from 0 to 3 volts for each channel.

ADJUSTING REFERENCE VOLTAGE

To choose the proper reference voltage, follow these guidelines:

o If the input signal is centered around 0 volts, select the 0 volt fixed reference.

o If the input signal is TTL compatible, select the TTL fixed reference.

- If the input signal is neither of the above, select the variable reference. To adjust the variable reference voltage, connect a voltmeter to the variable reference terminal and one of the return terminals on the rear panel connector. Using a small screwdriver, turn the variable reference adjustment screw until the desired voltage is displayed by the voltmeter.

ADJUSTING DEADBAND

To adjust the input deadband, there are several approaches. Use the method most suitable for your application.

- Method 1. For TTL signals, turn the deadband adjustment screw counterclockwise until it stops. Then turn the screw clockwise to the position marked TTL on the rear panel.

- Method 2. Connect the signal to be measured to one of the Counter/Totalizer input terminals, as for normal operation. Turn the appropriate deadband adjustment screw counterclockwise until it stops. Sample the channel and turn the deadband screw clockwise until a stable reading is obtained.

- Method 3. Temporarily select the 0 volt fixed reference for the channel in question. Remove any connections to the input terminal. Connect a voltmeter to one of the return terminals and to the threshold output terminal for this channel. Disregarding the polarity of the threshold voltage, turn the appropriate deadband adjustment screw until the threshold is one-half the desired deadband voltage. Restore the reference voltage to its original setting.

To ensure accurate measurements, adjust the deadband for each channel as high as possible.

CHECKING THRESHOLD VOLTAGES

The combination of reference voltage and deadband determines the high and low threshold voltages. To check the threshold levels, use the following procedure:

o Connect a voltmeter to one of the return terminals and to the threshold output terminal for the channel in question. Connect the test clock output to the channel input terminal.

o Turn the test clock switch to position 0 (+14 volt output) and read the low threshold on the voltmeter.

o Turn the test clock switch to position 1 (-15 volt output) and read the high threshold on the voltmeter.

Connections

The terminal assignments for the Counter/Totalizer connector are listed on the rear panel. The maximum input voltage is ±15V dc or ac peak.

The six channels on a Counter/Totalizer assembly are isolated from the Front End chassis and from ground but not from each other. The return lines on the input connector are common. All return lines must be connected to the same voltage.

USING THE COMMANDS

General

The DEF CHAN statement is used in defining pairs of frequency measurement channels. A conflict with some hardware settings on the Counter/Totalizer Board must not be introduced.

- First, each pair must consist of consecutive even and odd channels, corresponding to those selected on the function switch (Figure 6d-1).
- Second, the channels defined should fall within the range set by the channel decade switches (Figure 6d-1). Note that these switches establish the hundreds and tens designations for a range of channels.

Command Examples

Keeping these two rules in mind, let's define a pair of frequency measurement channels. If the function switch is set for FREQ on channels 0 and 1, and the channel decade switches are set for 1 and 0 (meaning 100), you can make the following definition:

```
DEF CHAN(100, 101) = FREQ
```

To measure these channels, enter a SEND CHAN command:

```
SEND CHAN(100, 101)
```

To ensure that the input signals are detected accurately, the Counter/Totalizer input thresholds must be properly adjusted. These and other adjustments are described earlier in this section.

Out-of-Range Conditions

The Counter/Totalizer is guaranteed to measure frequencies down to 2 hertz. Frequencies below 2 hertz cause a response of "9.99999E+37", signifying that an error has occurred.

The Counter/Totalizer is guaranteed to measure frequencies up to 400 kilohertz. Overrange conditions are not detected. When the input frequency exceeds 400 kilohertz, erroneous readings may be returned.

PUTTING IT ALL TOGETHER

The following example can be typed in directly from a terminal (TERM mode) or included in a computer program (COMP mode). Refer to Section 4 under "Operating the Front End from a Terminal/Computer" for more detailed information. Also, Appendix 9A provides guidelines for use with specific types of host computers.

```
MODE = TERM
FORMAT = DECIMAL
DEF CHAN (0..5) = FREQ
SEND CHAN (0)
```

This example assumes that the Counter/Totalizer (-167) address is set for 00, its channel function switches set to the FREQ position, the debounce/direct input switches set to Direct Input, and the floating/pulled-up resistor network moved to the Floating position. It commands the Front End to operate in terminal mode, sets the system variable FORMAT, and defines channels 0 through 5 for frequency measurement. The resulting measurement is made on channel 0 with the SEND CHAN command.

To verify a returned measurement, wire the test clock output on the Counter/Totalizer rear panel connector to the channel zero input terminal. The terminal assignments for the connector are listed on the rear panel. Since the test clock output and the channel inputs use the same ground, it is not necessary to wire a return line.

Using a small screwdriver, move the test clock switch to position 2 (100 kHz). Measure the channels input (SEND CHAN (0)) and verify a reading of:

```
1.00000E+05
```

If the response is "9.99999E+37", an error has occurred. Enter the LIST ERROR command to retrieve the associated error message. If necessary, refer to Section 8 for additional error message information.

Section 6E
Resistance Measurements
RESIST

INTRODUCTION

About This Section

This section explains the use of the Front End and associated option assemblies for obtaining resistance readings. Detailed information regarding the physical installation of the Front End mainframe and options is provided in Section 3.

Resistance Measurement

Resistance measurements, with or without conversion to other measurement units such as position or temperature, are directly supported by the Front End.

Measurements can be made with the Front End to determine resistance, or to determine the value of another directly related parameter. Many slide wire pots, non-standard RTDs, and other sensors with variable resistance outputs can be used to indicate temperature, position, and other physical parameters.

Several types of variable resistance transducers are discussed in greater detail in:

- o 6G: Strain Measurements
- o 6H: Temperature Measurement Using RTDs
- o 6I: Temperature Measurement Using Thermistors

The Front End measures resistance by passing a stable current through the resistor or sensor, and measuring the voltage that results. Appropriate conversions are then made to the sensed voltage and the output is displayed as a resistance reading.

REQUIRED HARDWARE

Resistance measurement applications require one of three option assembly configurations (A, B, and C).

The -161 High Performance A/D Converter, which provides high accuracy analog-to-digital conversion of scanner inputs, is used in Configurations A and B. The -165 Fast A/D Converter is used in Configuration C.

Installation of the required assemblies is detailed in Section 3B of this manual, where the options can be referenced by their three-digit numbers.

Configuration A

The first configuration employs the following two option assemblies in addition to the High Performance A/D Converter:

-163	RTD/Resistance Scanner
-177	RTD/Resistance Input Connector

This configuration provides the most accurate and repeatable resistance readings with the lowest cost per channel. Each RTD/Resistance Scanner and Connector set provides current excitation and measurement for 20 resistance channels. This configuration is intended for applications more exclusively involving resistance measurements; all 20 channels must be used for resistance measurement.

The RTD/Resistance Scanner configuration also provides lead-wire compensation for performing accurate 3-Wire resistance measurements.

Installation is covered in Section 3B for each option assembly being used. Briefly, these steps include:

- Select the Measurement Mode

  Set jumpers W1 and W2 on the RTD/Resistance Scanner for the desired measurement mode.

- Addressing

  Addresses are determined by the both the position of the RTD/Resistance Scanner (-163) relative to the High Performance A/D Converter (-161) and the address switch settings on that a/d converter.

- Physical Installation

  The RTD/Resistance Input Connector attaches to the RTD/Resistance Scanner.

- External Connections (see Section 3B, -177)

  4-Wire (4W) Measurement Mode
  Two sense and two excitation wires must be connected to each channel. Lead-wire or reed switch resistances do not affect readings.

  3-Wire Accurate (3WA) Measurement Mode
  One sense and two excitation wires must be connected to each channel. Equal lead-wire resistances are compensated for. Reed switch resistances do not affect readings. Ten channel returns are internally connected together.

  3-Wire Isolated (3WCM) Measurement Mode
  One sense and two excitation wires must be connected to each channel. Equal lead-wire resistances are compensated for. One reed switch resistance error affects readings.

Configuration B

The second configuration supports a wider mix of both resistance and other measurement applications. Excitation and measurement can be provided for several types of 4-Wire resistance output sensors by the following assemblies:

-162	Thermocouple/DC Volts Scanner
-176	Voltage Input Connector*

* Any other input connector supporting direct voltage measurement can be used here. These include: -160 AC Volts Input Connector and -175 Isothermal Input Connector.

-164	Transducer Excitation Module
-174	Transducer Excitation Connector

The -162 Thermocouple/DC Volts Scanner is a plug-in, 20 channel, thermocouple and direct voltage, reed relay scanner contained on a single pcb. This selects and conditions one channel at a time for conversion by the High Performance A/D Converter. Connections to the -162 scanner can be made through a Voltage Input, Isothermal Input, or AC Voltage Input Connector. This combination (High Performance A/D Converter, scanner, and connector) measures the direct voltage generated across the resistance to be measured.

The -164 Transducer Excitation Module is used to energize either the resistance sensors with an excitation current or the resistance bridge configurations with an excitation voltage. Used in conjunction with the -174 Transducer Excitation Connector, each -164 provides five current sources that can excite up to four channels per source, and one voltage source that can excite bridge configurations.

Installation is covered in Section 3B for each option assembly being used. Briefly, these steps include:

- Addressing

  The addresses for resistance measurement channels are determined by the measurement hardware. Therefore, both the position of the Thermocouple/DC Volts Scanner (-162) relative to the High Performance A/D Converter (-161) and the address switch settings on that a/d converter set the address.

- Voltage/Current Excitation Jumper

  Each group of four channels must be configured for voltage or current output by correctly positioning the respective jumper on the Transducer Excitation Connector (-174). Refer to the connector decal for details.

- Physical Installation

  The Transducer Excitation Module/Connector (-164/-174) and the Thermocouple/DC Volts Scanner/connector (-162/-175,-176, or -160) should be installed one-above-the-other to facilitate interconnections. Overall installation considerations are discussed in Section 3B.

- External Connections

  Configuration B resistance measurements are performed with the system connections shown in Section 3B for the -174 connector.

Configuration C

Configuration C is similar to Configuration B in that it supports a mixture of measurement types. Configuration C differs in that it uses the -165 Fast A/D Converter in place of the -161 High Performance A/D Converter and -162 scanner. The full hardware requirements are:

-165	Fast A/D Converter
-176	Voltage Input Connector, or
-175	Isothermal Input Connector
-164	Transducer Excitation Module
-174	Transducer Excitation Connector

The -165 Fast A/D Converter is a plug-in measurement card that supports a maximum of 20 differential voltage measurement inputs. The Fast A/D Converter must be used in differential mode with the -164 Transducer Excitation Module. The -176 or -175 connector provides convenient screw terminals for inputs to the -165 a/d converter.

The -164 Transducer Excitation Module provides 1 mA current sources used in making resistance measurements. This option also provides voltage excitation for bridge-type measurements. The -174 connector provides screw terminals.

o Addressing

Resistance measurement channel addresses are determined by the measurement hardware. In this configuration, the address switches on the -165 Fast A/D Converter must be set. This setting identifies the first 40 addresses of an address block.

Since differential measurement inputs must be used in conjunction with the -164 Transducer Excitation

Module, differential pair addressing rules must be observed. Each pair must include a second channel that is exactly 20 addresses higher than the first. Therefore, differential inputs can be set up for address pairs 0/20 through 19/39.

- Voltage/Current Excitation Jumper

  Each group of four channels must be configured for voltage or current output by correctly positioning the respective jumper on the Transducer Excitation Connector (-174). Refer to the connector decal for details.

- Physical Installation

  The Transducer Excitation Module/Connector (-164/-174) and the Fast A/D Converter/connector (-165/-175,-176) should be installed one-above-the-other to facilitate interconnections. Overall installation considerations are discussed in Section 3B.

- External Connections

  Configuration C resistance measurements are performed with the system connections shown in Section 3B for the -174 connector.

USING THE COMMANDS

General

The DEF CHAN statement is used to set-up a channel or group of channels for measuring resistance transducers.

CONFIGURATION A

For Configuration A, three ranges (256, 2048 and 64000 ohms) are available. The range selected is again determined by the MAX parameter, with the default range being 64 kilohms. The channel definition uses the following format:

```
DEF CHAN(channels)=RESIST, MAX=(anticipated high value)
```

CONFIGURATIONS B AND C

For Configuration B or C, two ranges (64 and 512 ohms) are available. In specifying the highest anticipated reading, the optional MAX parameter also determines the appropriate range. If the anticipated high reading is within the lower range, that range is selected. If no MAX parameter is specified, the default (512 ohm) range is used. The same channel definition format is used.

Interpolation Tables

At times, the resistance value returned by the Front End may not be in the units or in the form that will eventually be used by the computer. In such a case, a transfer function must be used to convert the reading.

This function can be provided by the computer program. Where the formula is known, this method may be all that is necessary.

But the transfer formula may not be known. Some types of transducers are only provided with a table of input and output values. The Front End's interpolation table function can then be used as a transfer function.

An interpolation table allows for user-specified input-to-output value relationships. When an actual input falls between two table input values, a linear interpolation is used to determine the output. If the input falls outside table values, a linear extrapolation is used to determine the output.

A complete discussion of interpolation tables is found in Section 5.

Command Examples

Let's define channels 30 through 37 for resistance measurement with an anticipated maximum reading of 300 ohms. For Configuration A, the 2048 ohm range is used. For Configuration B or C, the 512 ohm range is used. In the same statement, call the interpolation table previously defined.

```
DEF CHAN(30..39) = RESIST, MAX = 300
```

Now, measure the resistance on the defined channels.

```
SEND CHAN(30..39)
```

If the reading is greater than allowed in the specified range (as determined by the maximum anticipated value), "9.99999E+37" is returned. This signifies that an out of range error has occurred.

PUTTING IT ALL TOGETHER

The following example can be typed in directly from a terminal (TERM mode) or included in a computer program (COMP mode). Refer to Section 4 under "Operating the Front End from a Terminal/Computer" for more detailed information. Also, Appendix 9A provides guidelines for use with specific types of host computers.

```
MODE = TERM
FORMAT = DECIMAL
DEF CHAN (0..19) = RESIST
SEND CHAN (0)
```

This example assumes that the High Performance A/D Converter (-161) address is set to 0 and that a resistance is connected to either the RTD/Resistance Input Connector (-177) (Configuration A used) or the Transducer Excitation Connector (-174) (Configuration B used). It commands the Front End to operate in terminal mode, sets system variable FORMAT, and defines analog input channels 0 through 19 for resistance measurement. Since no MAX parameter has been specified, the highest resistance range (Configuration A = 64 kilohm range, Configuration B or C = 512 ohm range) is used. The resulting measurement is made on channel 0 with the SEND CHAN command.

For example, a 330 ohm resistor applied to the inputs should be measured as:

```
3.30000E+02
```

If the response is "9.99999E+37", an error has occurred. Enter the LIST ERROR command to retrieve the associated error message. If necessary, refer to Section 8 for additional error message information.

Section 6F
Status Outputs
STATOUT

INTRODUCTION

About This Section

This section explains use of the Front End and associated option assemblies for making status outputs. Detailed information about the physical installation of the Front End mainframe and related options is provided in Section 3.

Status Output

The Status Output control function is a single-bit logic state output. Status outputs are used for such activities as opening and closing relays, turning indicator lights on and off, or opening and closing valves. Status outputs can also be wired to provide logic inputs to other devices.

REQUIRED HARDWARE

Applications involving status outputs require the following option assemblies:

-168	Digital I/O Assembly
-169	Status Output Connector

The Digital Input/Output Assembly allows the Front End to exchange information with a digital peripheral device. Four types of data exchange are possible: Alarm or Status Output; Status Input; BCD Input; and Binary Input.

Status Outputs

The Status Output Connector attaches to the Digital I/O Assembly (-168). It can send 20 single-bit output signals from the Digital I/O to external control points or terminals. Each output is individually selectable and can be used either to drive lamps and relays or change logic levels.

Installation for either assembly is detailed in Section 3B of this manual. Reference each option assembly by its option number (-168 or -169).

As an overview, installation requires the following actions:

- Address Selection

  With the connector detached, set the address switches on the Digital I/O Assembly. See Section 3B (-168).

- Wiring Connections

  On the Status Output Connector, make necessary external connections to OUTPUT and RETURN terminals and, if used as a relay driver, to the appropriate CLAMP terminal.

- Physical Installation

  Attach the Status Output Connector to a Digital I/O Assembly installed in the Front End or an Extender Chassis.

USING THE COMMANDS

General

Setting up status output channels involves the DEF CHAN command to specify the channel (or group of channels) and type. The CHAN statement is used separately to define the output value (1 or 0). Any non-zero value is interpreted as "1".

The output can be checked with the SEND CHAN command.

Command Examples

Let's define channels 20 through 29 as status outputs.

```
DEF CHAN(20..29) = STATOUT
```

Now, set the outputs for this block of channels to logic "1".

```
CHAN(20..29) = 1
```

If you want to change the output value, enter:

```
CHAN(20..29) = 0
```

Use the SEND CHAN command to check the last value set for a specified group of status outputs.

```
SEND CHAN(20..29)
```

PUTTING IT ALL TOGETHER

The following example can be typed in directly from a terminal (TERM mode) or included in a computer program (COMP mode). Refer to Section 4 under "Operating the Front End from a Terminal/Computer" for more detailed information. Also, Appendix 9A provides guidelines for use with specific types of host computers.

```
MODE = TERM
DEF CHAN(0..19) = STATOUT
CHAN(0..19) = 0
SEND CHAN(0..19)
```

This example assumes that a Status Output Connector (-169) is wired so that its terminals can be accessed with a digital multimeter. The connector is installed on a Digital I/O Assembly set for address 00. The example commands the Front End to operate in terminal mode, defines channels 0 through 19 for status outputs, and sets the channel outputs for logic zero. The SEND CHAN command verifies only the output value last received by the Front End through the "CHAN(channels)=0/1" command.

To verify actual status outputs, a digital multimeter should be used at the output terminals. Use a DMM set for either the diode checking range or the ohms range. Connect the DMM Volt/Ohm (red) lead to the channel OUTPUT terminal and the common (black) lead to the RETURN terminal. When the output is set for zero, a high resistance (greater than 20 megohms) should be found on each channel.

Now send the commands:

```
CHAN(0..19) = 1
SEND CHAN(0..19)
```

Verify that the returned value is a "one" and that the resistance measured by the DMM is lower (less than 9 megohms).

If either response is "9.99999E+37", an error has occurred. Enter the LIST ERROR command to retrieve the associated error message. If necessary, refer to Section 8 for additional error message information.

Section 6G
Strain Measurement
STRAIN

INTRODUCTION

About This Section

This subsection explains the concept of strain and its measurement by the Front End. Detailed information regarding the physical installation of the Front End mainframe and options is provided in Section 3.

Strain Measurement

Strain is the measurement of the deformity of an object subjected to an external force. This type of measurement is valid up to the elastic limit of the object being stressed. Strain gauges are widely used in such measurements as:

- Weight: load cells are commonly used to weigh heavy objects.
- Pressure: pressure gauges are frequently strain gauges mounted on a diaphragm that deforms in a predictable manner under pressure.
- Fluid Level: the fluid level in a tank can be measured using a strain gauge mounted on a support leg of the tank.

STRAIN MEASUREMENT

General

The force deforming a body can cause an increase or a decrease in a dimension of that body. Any increase is considered positive and is termed tensile strain. A decrease is negative and is called compressive strain. Strain is normally expressed as the ratio of the change in dimension over the original dimension (the ratio itself is a dimensionless number.) See Figure 6g-1. Strain is defined by the following equation:

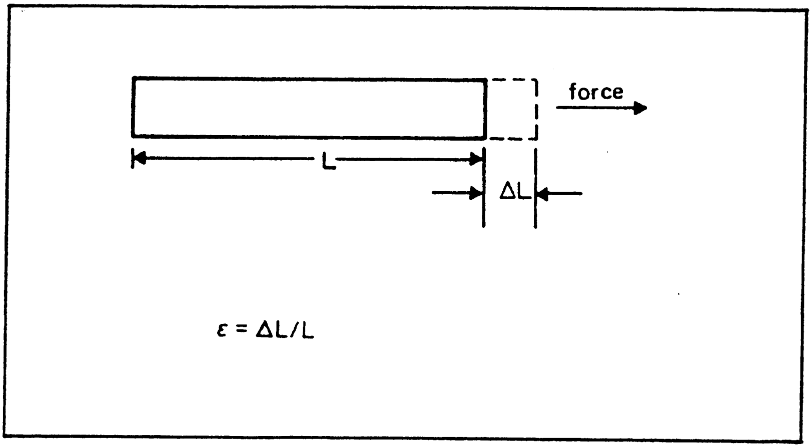


Figure 6g-1. Strain

Poisson's Ratio

When a material is stressed, it expands along the axis of the applied force and it contracts along an axis at a right angle to the applied force. The negative ratio of axial strain to perpendicular strain is called the Poisson ratio. The Poisson ratio is a constant that is a characteristic of the material. See Figure 6g-2.

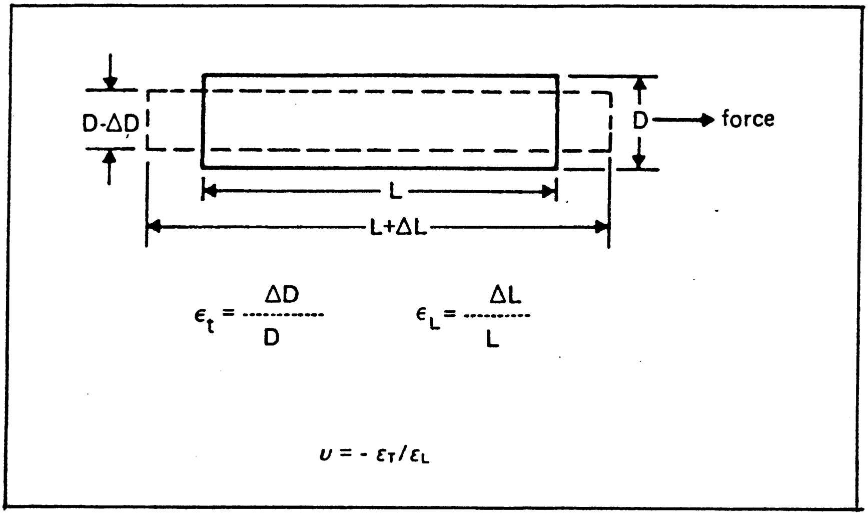

D-ΔD
D
force
L
L+ΔL
$\epsilon_t = \frac{\Delta D}{D}$
$\epsilon_L = \frac{\Delta L}{L}$
$\upsilon = - \varepsilon_T / \varepsilon_L$

Figure 6g-2. Poisson Ratio

Resistive Strain Gauges

The resistive strain gauge is the most common variety. It is composed of a serpentine of metal film on a thin piece of plastic. This arrangement is highly sensitive to strain in the longitudinal direction, but relatively insensitive to strain in the perpendicular direction. The resistive strain gauge, when bonded to the material under test, changes resistance as the test material undergoes deformation.

Gauge Factor

The degree of change (resistance to strain) is expressed as the sensitivity of the gauge (or "gauge factor"). It is the calibration constant for the strain gauge. The gauge factor, K, varies with the material used in the gauge, the temperature, and the stress. K is defined as:

$$K=(\Delta R/R)\ /\ (\Delta L/L)$$

Substituting ϵ for $\Delta L/L$, the general equation for a resistive strain gauge becomes:

$$\epsilon = \Delta R/R\ *\ (1/K)$$

Typical gauge factor values range from 2.0 to 4.0.

Micro Strain

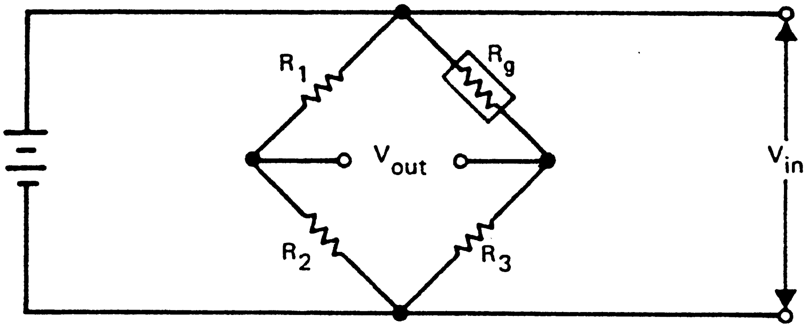
Strain is typically measured with a resolution of 1.0E-6 (one micro strain). Remember that strain is expressed as a ratio of the change in length divided by the original length. It is a dimensionless measurement.

If a strain gauge has a resistance of 120 ohms and a gauge factor of 2, one micro strain amounts to 0.24 milliohms change in resistance. Resolving 0.2 milliohms out of 120 ohms requires very precise instrumentation.

Measurement Techniques

Special techniques are used to measure the incremental change in resistance caused by strain.

A Wheatstone bridge (Figure 6g-3) offers one method of measuring these small resistance changes. As strain occurs, Vout changes in proportion to the change in gauge resistance. The relationship of this resistance change to Vout is non-linear. However, within the range of interest, a linear equation can very closely approximate accurate results.



Where: Vin is the input voltage to the bridge
Rg is the resistance of strain gauge
R1, R2, R3 are bridge completion resistor
Vout is the bridge output voltage

Figure 6g-3. Wheatstone Bridge

Strain Measurement

A 1/4 bridge configuration exists when one arm of the bridge is active (Rg) and the other arms are either fixed value resistors or unstrained gauges. Vout is a function of Vin, R1, R2, R3, and Rg as seen in the following equation:

$$V_{out} = V_{in} * (R3/(R3 + R_g) - R2/(R1 + R2))$$

When R1/R2 = Rg/R3, the bridge is balanced and Vout becomes zero. Referring back to the gauge factor equation, it can be seen that the quantity to be measured is the fractional change in gauge resistance from unstrained to strained).

In a manual strain measurement system, the bridge circuitry is usually modified by the addition of a balancing resistor. Stresses induced in the strain gauge when it is bonded to the tested material are nulled (balanced) out by the balancing resistor when the gauge is in its unstrained condition. While this method is often satisfactory for single-point measurements, the number of manual adjustments for balanced bridge methods becomes cumbersome in multichannel systems.

A two-measurement system solves the manual-adjustment problem. The first (initial) measurement is made after the strain gauge has been mounted, but before it has been subjected to strain. Subsequent measurements are then subtracted from the initial measurement.

Anatomy of a Measurement

The equation for Vout can be rewritten as the ratio Vout to Vin:

$$V_{out}/V_{in} = (R3/(R3 + R_g) - R2/(R1 + R2))$$

This equation is relevant for both unstrained and strained gauge conditions. With the unstrained value of gauge resistance defined as Rg and the change due to strain defined as ΔR_g, the strained value of gauge resistance is Rg $+\Delta R_g$. The effective value of resistance in each bridge arm is the sum of all resistances in that arm. Lead wires, printed circuit board traces, and interconnects may all contribute to this effective resistance. As long as these resistances remain constant between the strained and unstrained readings (and are relatively small compared to the gauge resistance), the measurement is not affected. Later, we will introduce a term to compensate for these resistances.

The difference between the Vout/Vin ratio in the unstrained state and in the strained state requires a new term, Vr.

$$V_r = (V_{out}/V_{in})_{\text{strained}} - (V_{out}/V_{in})_{\text{unstrained}}$$

By substituting the resistor values that correspond to the two Vout/Vin terms into this equation, an equation for $\Delta R_g/R_g$ can be derived.

$$\Delta R_g/R_g \quad = -4V_r/(1 + 2V_r)$$

When combined with the equation for gauge factor (K), strain can be defined in terms of Vr and K. This equation describes the ideal behavior of a strain gauge over a wide range of conditions:

$$\varepsilon = -4V_r/K(1 + 2\,V_r)$$

This equation can be modified as follows to compensate for conditions where there are significant lead wire and termination resistances.

$$\varepsilon = -4V_r/K(1 + 2\,V_r) * (1 + R_L/R_g)$$

where: RL = total of all lead resistances
Rg = gauge resistance

In practice, factors such as gauge hysteresis and "creep" limit restrict the useful range of most resistive strain gauges. In such cases, the Vr term in the denominator is very small and can be ignored. The 1/4 bridge equation can thereby be simplified to:

$$\varepsilon = -4V_r/K * (1 + R_L/R_g)$$

REQUIRED HARDWARE

For maximum resolution and accuracy, applications involving strain measurement can use the following option assemblies:

-164	Transducer Excitation Module
-174	Transducer Excitation Connector
-161	High Performance A/D Converter
-162	Thermocouple/DC Volts Scanner
-176	Voltage Input Connector

For higher reading rates, where accuracy and resolution are less critical, use the following option assemblies:

-164	Transducer Excitation Module
-174	Transducer Excitation Connector
-165	Fast A/D Converter
-176	Voltage Input Connector

These assemblies are documented in Section 3B. They can be categorized within two logical functions: excitation and measurement.

Excitation Hardware

The Transducer Excitation Module (-164) is used to energize resistive strain gauges with an excitation voltage. It is used in conjunction with the Transducer Excitation Connector (-174). Together, each 164/174 set provides 20 channels of output.

Measurement Hardware

HIGH ACCURACY HARDWARE

The High Performance A/D Converter (-161) provides high accuracy analog to digital conversion of scanner input voltages. At least one a/d converter must be installed in the Front End.

The Thermocouple/DC Volts Scanner (-162) is a plug-in, 20-channel thermocouple and multi-voltage range relay scanner contained on a single pwb. The scanner operates as a self-calibrating analog data multiplexer, linking the High Performance A/D Converter to external measurement points. It accepts a variety of analog inputs, depending on the type of connector in use.

The Voltage Input Connector (-176) or Isothermal Input Connector (-175) routes a maximum of 20 direct voltage input channels to the Thermocouple/DC Volts Scanner. The AC Voltage Input Connector (-160) can also be used to route 10 direct voltage input channels.

HIGHER SPEED HARDWARE

The Fast A/D Converter (-165) provides higher reading rates than the -161, with some compromise in accuracy and resolution. The -165 a/d converter is self-calibrating. It measures 20 differential direct voltage inputs directly.

The Voltage Input Connector (-176) (or the Isothermal Input Connector (-175)) routes a maximum of 20 differential direct voltage inputs to the Fast A/D Converter.

Installation Summary

A constant voltage power supply (the Transducer Excitation Module) furnishes Vin. The Thermocouple/DC Volts Scanner is used to measure Vout. The voltage output of the unbalanced gauge is measured before and after the strain is applied: the Front End performs the required calculations.

Installation is covered in Section 3B for each option assembly being used. Briefly, these steps include:

o Addressing

The addresses for strain measurement channels are determined by the measurement hardware.

With High Accuracy Hardware, both the position of the Thermocouple/DC Volts Scanner (-162) relative to the High Performance A/D Converter (-161) and the address switch settings on that a/d converter set the address.

For Higher Speed Hardware, only the address switch settings on the -165 Fast A/D Converter set the address of a measurement channel.

- Voltage Excitation Jumper

  Each group of four channels must be configured for voltage output by correctly positioning the respective jumper on the Transducer Excitation Connector (-174). Refer to the connector decal for details.

- Physical Installation

  The excitation hardware (Transducer Excitation Module/Connector, -164/-174) and the measurement hardware should be installed adjacent to each other to facilitate interconnections. High Accuracy measurement hardware includes the High Performance A/D Converter, the Thermocouple/DC Volts Scanner, and a connector (-176, -175, or -160). Higher Speed measurement hardware includes the Fast A/D Converter and a connector (-176 or -175).

  Overall installation considerations are discussed in Section 3B.

- External Connections

  Strain measurements are performed with the system connections shown in the following wiring diagrams (Figures 6g-4 through 6g-6). Note that the LO to SHIELD connection differs between types of measurement hardware (High Accuracy or Higher Speed). With High Accuracy hardware, connect LO to SHIELD for each channel. With Higher Speed hardware, connect LO to SHIELD on only one channel per Fast A/D Converter.

NOTE

On the input connector, tie LO to SHIELD on each channel when the -161 High Performance A/D Converter and -162 Thermocouple/DC Volts Scanner are used. If the input connector is used directly on the -165 Fast A/D Converter, tie LO to SHIELD on one channel only (per a/d converter.)

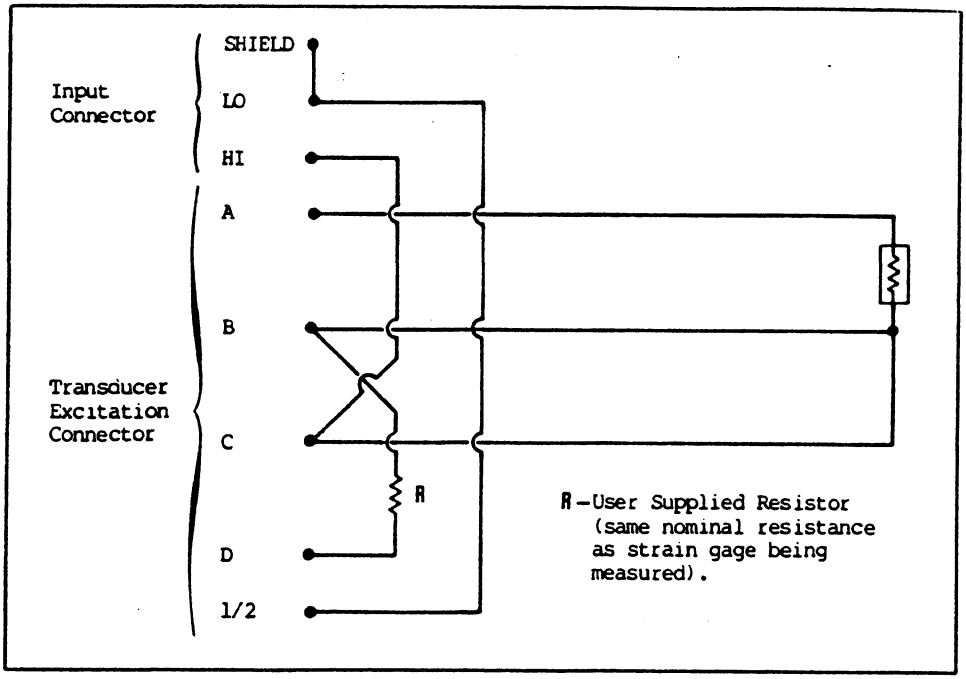

Input Connector
SHIELD
LO
HI
Transducer Excitation Connector
A
B
C
D
1/2
R
R–User Supplied Resistor (same nominal resistance as strain gage being measured).

Figure 6g-4. Quarter Bridge Connections

NOTE

On the input connector, tie LO to SHIELD on each channel when the -161 High Performance A/D Converter and -162 Thermocouple/DC Volts Scanner are used. If the input connector is used directly on the -165 Fast A/D Converter, tie LO to SHIELD on one channel only (per a/d converter.)

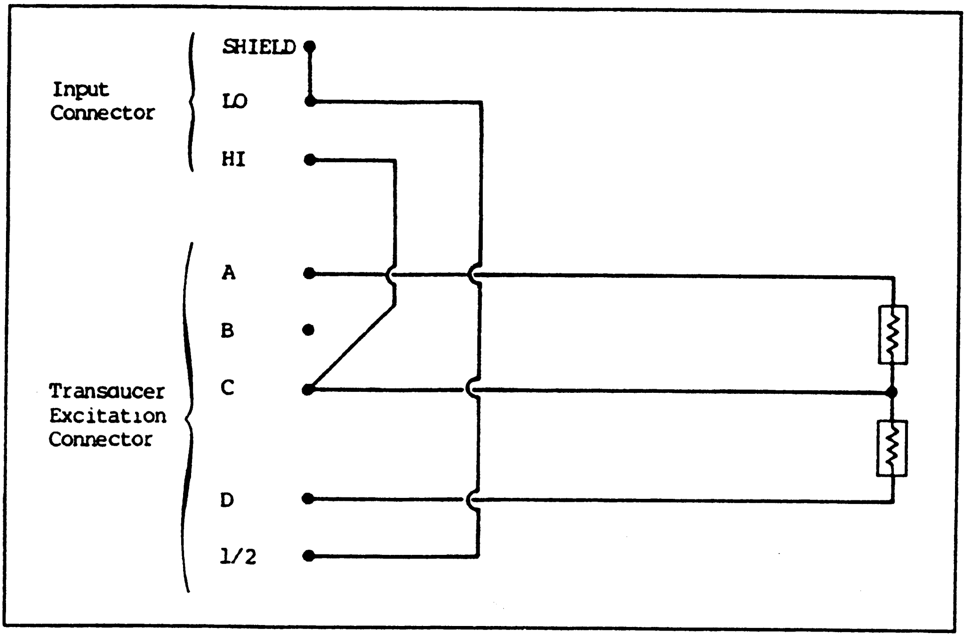


Figure 6g-5. Half Bridge Connections

NOTE

On the input connector, tie LO to SHIELD on each channel when the -161 High Performance A/D Converter and -162 Thermocouple/DC Volts Scanner are used. If the input connector is used directly on the -165 Fast A/D Converter, tie LO to SHIELD on one channel only (per a/d converter.)

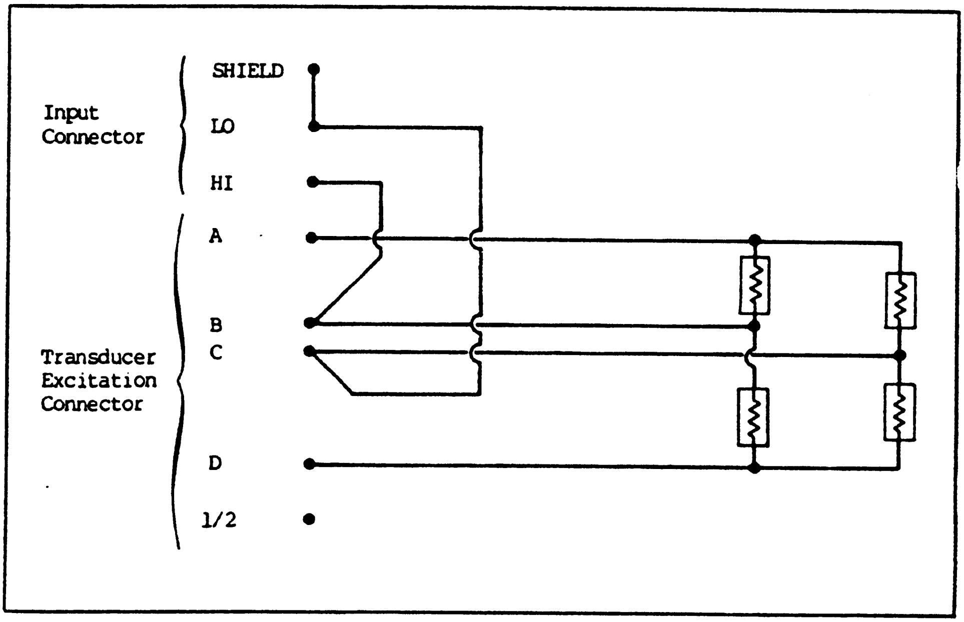


Figure 6g-6. Full Bridge Connections

USING THE COMMANDS

General

Channel definitions for strain measurement can be more involved than those used for other types of measurement. There are six parameters to define (STRAIN, TYPE, K, NU, VINIT, VEXC, RANGE). The STRAIN type must be defined, but settings for each of the remaining five parameters are necessary only if values other than the defaults are required. Table 6g-1 describes these five parameters.

Table 6g-1. Strain Parameters

Parameter	Description	Default
K	Gauge Factor	2.0
NU	Poisson ratio	0.2
VINIT	Initial voltage measured	0.0
VEXC	Excitation voltage	4.0
RANGE	Measurement range	LO

If no default values are used, the definition statement must be entered in the following format:

```
DEF CHAN(channels) = STRAIN, TYPE = (0,1,2,3,4, or 5),
K = (number), NU = (number), VINIT = (number), VEXC =
(number), RANGE = HI
```

If the default values for K, NU, VINIT, VEXC, and RANGE are correct, the following syntax is adequate:

```
DEF CHAN(channels) = STRAIN, TYPE = (0,1,2,3,4, or 5)
```

And any combination of parameters between these two extremes is possible.

Strain Measurement

Strain Initialization

The first step in any strain measurement involves a direct voltage reading of the unstrained configuration. This establishes the value of VINIT, one of the parameters mentioned above. For this measurement, make a separate direct voltage definition for the channel in question. Since the unstrained voltage is a small value, specify the most sensitive range. Use the following command line:

```
DEF CHAN(channel) = DVIN, MAX = .06
```

Now, monitor the unstrained reading with SEND CHAN(channel). Then, specify any response other than 0 (the Vinit default) in the strain definition (VINIT = (response)).

Bridge Configuration TYPE

The six bridge configurations available for strain measurement are summarized in Table 6g-2 and illustrated in Figures 6g-7 through 6g-12.

NOTE

The following symbols are used in Figures 6g-7 through 6g-12:

ϵ	measured micro strain
V_{meas}	measured gauge voltage (mV)
V_{init}	measured gauge voltage in unstrained condition (mV)
υ	Poisson's ratio
V_{ex}	excitation voltage
K	gauge factor

Table 6g-2. Strain Bridge Configurations

TYPE	Description
0	Quarter Bridge (Figure 6g-7) The quarter bridge configuration can be used where there is a single active strain gauge.
1	Half Bridge with Equal and Opposite Strains (Figure 6g-8) This half bridge arrangement involves two active strain gauges, both with equal and opposite strains.
2	Half Bridge with One Principle and One Poisson Strain (Figure 6g-9) This half bridge arrangement involves two active strain gauges, one aligned with maximum principle strain, one aligned with Poisson strain.
3	Full Bridge with Two Pairs of Equal and Opposite Strains (Figure 6g-10) The full bridge configuration allows for four active strain gauges with pairs subjected to equal and opposite strains.
4	Full Bridge with Two Principle Strains and Two Poisson Strains (Figure 6g-11) This configuration allows for four active gauges, two aligned with maximum principle strain, two aligned with Poisson strain.
5	Full Bridge with Two Principle Strains and Two Poisson Strains (Figure 6g-12) This configuration also allows for four active gauges, two aligned with maximum principle strain, two aligned with Poisson strain.

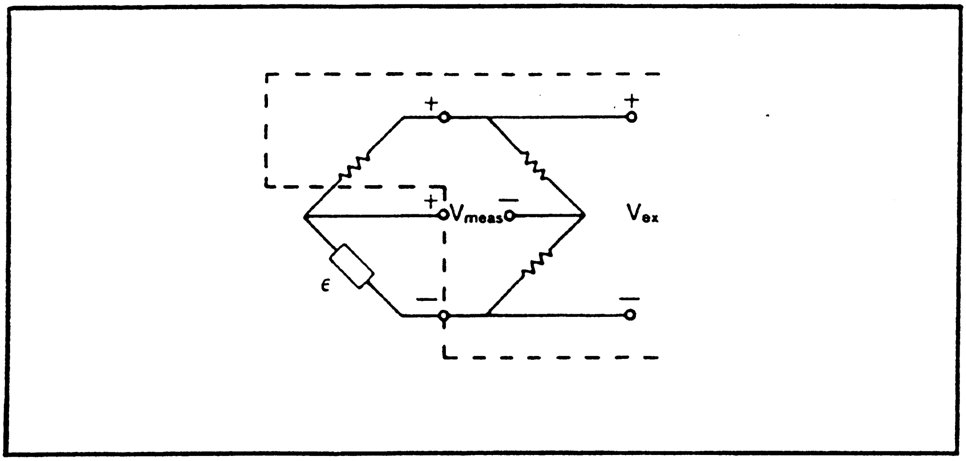


Figure 6g-7. Quarter Bridge Configuration 0

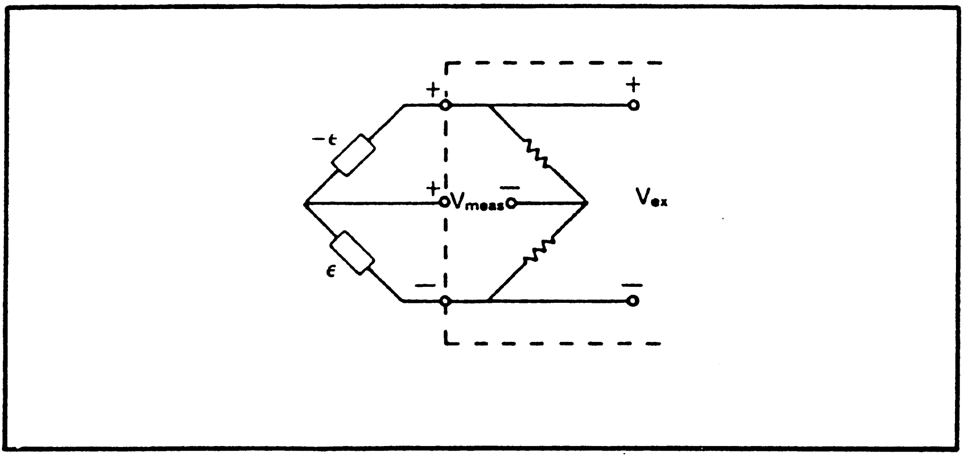


Figure 6g-8. Half Bridge Configuration 1

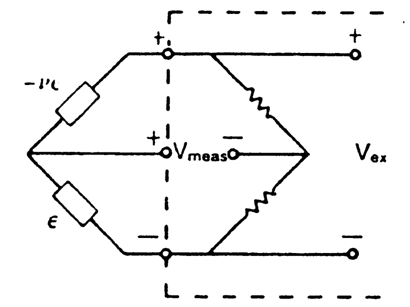


Figure 6g-9. Half Bridge Configuration 2

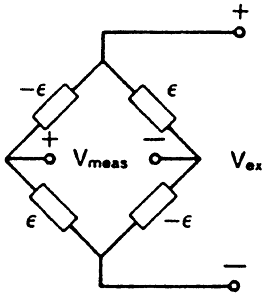


Figure 6g-10. Full Bridge Configuration 3

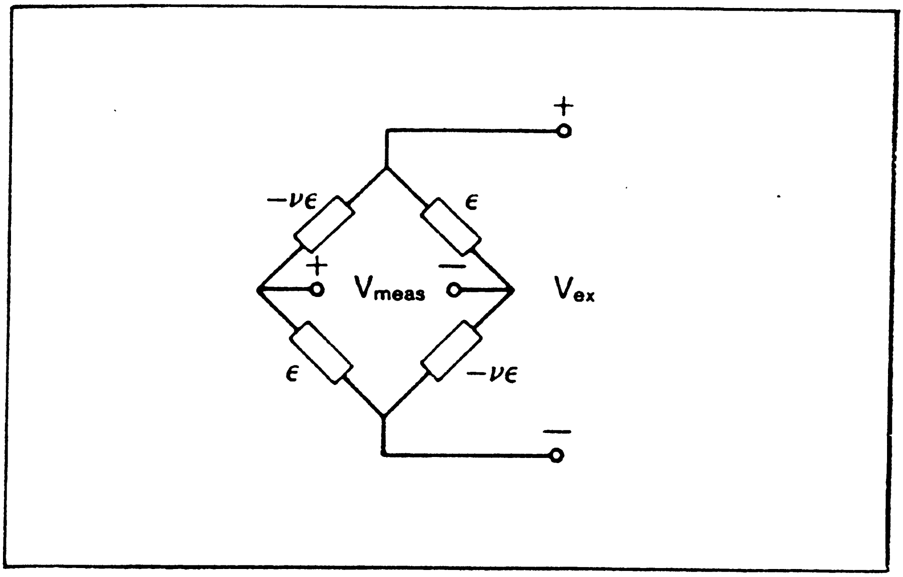


Figure 6g-11. Full Bridge Configuration 4

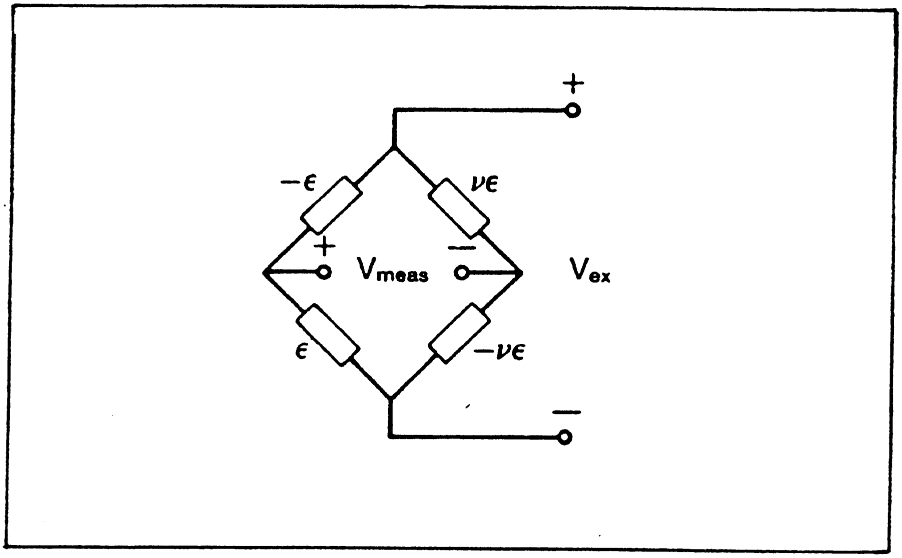


Figure 6g-12. Full Bridge Configuration 5

Command Examples

Let's set up a strain measurement in a half-bridge configuration for channel 0. The first step involves a direct voltage measurement of this channel in the unstrained state.

```
DEF CHAN(0) = DVIN, MAX = .06
```

Now, identify the initialization voltage, VINIT, by monitoring this reading.

```
SEND CHAN(0)
```

Let's assume that the response is 1.00000E-4, or 0.0001 volts. Since this differs from the initialization voltage default value (0), VINIT must be specified in the strain definition statement as follows:

```
DEF CHAN(0) = STRAIN, TYPE = 2, VINIT = 0.0001
```

Monitor the strain measurement with:

```
SEND CHAN(0)
```

If other strain parameters differ from the default values, a strain definition might look as involved as this:

```
DEF CHAN(0) = STRAIN, TYPE = 2, K = 2.1, NU =
0.25, VINIT = 0.01, VEXC = 4.5, RANGE = HI
```

PUTTING IT ALL TOGETHER

The following example can be typed in directly from a terminal (TERM mode) or included in a computer program (COMP mode). Refer to Section 4 under "Operating the Front End from a Terminal/Computer" for more detailed information. Also, Appendix 9A provides guidelines for use with specific types of host computers.

```
MODE = TERM
FORMAT = DECIMAL
DEF CHAN(0..19)=STRAIN, TYPE=1, VEXC=2.0, VINIT=0.025
SEND CHAN(0)
```

This example assumes that the High Performance A/D Converter (-161) or Fast A/D Converter (-165) address is set to 0 and that a half-bridge (Configuration 1) strain circuit is wired to the Transducer Excitation Connector (-174) and channel 0 of the Voltage Input Connector (-176). It commands the Front End to operate in terminal mode, sets system variable FORMAT, and defines analog input channels 0 through 19 for configuration 1 strain measurement.

Additional parameters in the DEF CHAN statement set the excitation voltage (VEXC) and initial voltage measured (VINIT) to non-default values.

The resulting measurement is made on channel 0 with the SEND CHAN command.

If the measured strain is 5000 microstrain, the response would be:

```
5.00000E+03
```

If the response is "9.99999E+37", an error has occurred. Enter the LIST ERROR command to retrieve the associated error message. If necessary, refer to Section 8 for additional error message information.

Section 6H
Temperature Measurement Using RTDs

INTRODUCTION

About This Section

This section explains the use of the Front End and associated option assemblies for obtaining RTD temperature readings. Detailed information regarding the physical installation of the Front End mainframe and related options is provided in Section 3.

RTD Temperature Measurement

This type of temperature measurement employs a resistance-temperature detector (RTD). RTDs, which are usually larger and more expensive than thermocouples, are frequently used where accuracy and repeatability are important because they exhibit greater accuracy and stability.

The resistance of an RTD varies directly with the RTD sensor temperature. Passing a current through this resistance generates a proportional voltage that can be accurately translated into a temperature reading.

Most RTD types have temperature sensing elements that are made of platinum. Several other materials can be used in RTDs, but platinum remains the most popular and accurate type. Each type of RTD requires a unique algorithm for converting the measured resistance into temperature. This capability is provided by the Front End operating software which contains conversion algorithms for platinum and copper RTDs. Conversion algorithms for other platinum RTD types can be implemented using the Front End's user-defined RTD functions.

REQUIRED HARDWARE

RTD temperature measurement applications require one of three option assembly configurations (A, B, or C).

The High Performance A/D Converter (-161) is used in Configuration A and Configuration B to provide high accuracy analog-to-digital conversion of RTD, thermistor, or resistance measurements. At least one a/d converter must be installed in the Front End or the 2281A Extender Chassis.

The Fast A/D Converter is used in Configuration C to provide higher reading rates at somewhat lower accuracy and resolution.

Configuration A

The first configuration employs the following two option assemblies:

-163	RTD/Resistance Scanner
-177	RTD/Resistance Input Connector

If a large number of channels is to be measured, this configuration provides the most accurate and repeatable RTD readings, the fewest restrictions, and the lowest cost per channel. Each RTD/Resistance Scanner/Connector set provides current pair excitation and signal multiplexing for 20 RTD, thermistor, and/or resistance channels. These two assemblies are intended for use in applications more exclusively involving RTD, thermistor, and resistance measurements.

The RTD/Resistance Scanner configuration also provides lead-wire compensation for performing accurate 3-Wire resistance measurements.

Installation is covered in Section 3B for each option assembly being used. Briefly, these steps include:

o Select the Measurement Mode

Set jumpers W1 and W2 on the RTD/Resistance Scanner for the desired measurement mode.

o Addressing

Addresses are determined by the both the position of the RTD/Resistance Scanner (-163) relative to the High Performance A/D Converter (-161) and the address switch settings on that a/d converter.

o Physical Installation

The RTD/Resistance Input Connector attaches to the RTD/Resistance Scanner.

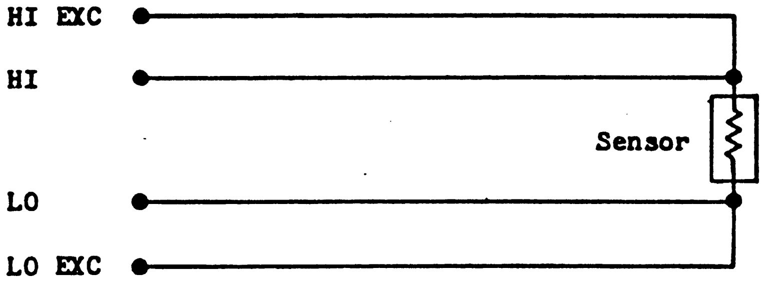
o External Connections

4-Wire (4W) Measurement Mode
Two sense and two excitation wires must be connected to each channel. Lead-wire or reed switch resistances do not affect readings. See Figure 6h-1.

3-Wire Accurate (3WA) Measurement Mode
One sense and two excitation wires must be connected to each channel. Equal lead-wire resistances are compensated for. Reed switch resistances do not affect readings. Ten channel returns are internally connected together; therefore, the RTDs must be electrically isolated. See Figure 6h-2.

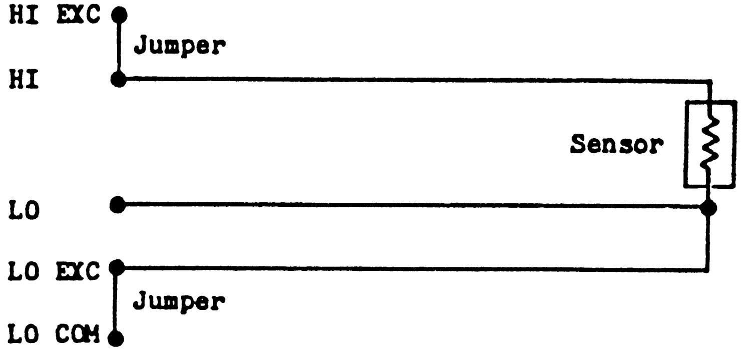
3-Wire Isolated (3WCM) Measurement Mode
One sense and two excitation wires must be connected to each channel. Equal lead-wire resistances are compensated for. One reed switch resistance error affects readings. See Figure 6h-3.

Where lead-wire errors are negligible, 2-Wire measurements can be made by setting the scanner to operate in 4-Wire mode, and wiring the -177 input connector properly.



This is the most accurate measurement mode.

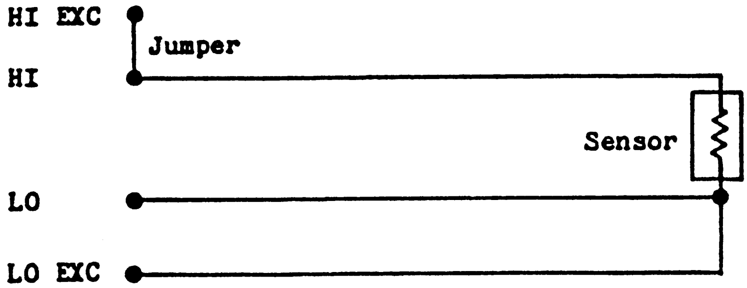
Figure 6h-1. Wiring: 4-Wire, Configuration A



Ensure that the HI and LO EXC lead wires have the same length, composition, and resistance.

Use this 3-Wire configuration if at all possible. Note that the LO COM's of ten channels, (e.g., 0 to 9) are connected together inside the connector pca.

Figure 6h-2. Wiring: 3-Wire Accurate, Configuration A



Ensure that the HI and LO EXC lead wires have the same length, composition, and resistance.

This is much less accurate than the 3WA mode for all but large (high-kilohm) resistance measurements.

Figure 6h-3. Wiring: 3-Wire Isolated, Configuration A

Configuration B

This configuration supports a wider mix of applications, providing both current and voltage excitation to support the measurement of 4-wire RTDs and transducers in bridge arrangements (such as strain gauges). The following option assemblies are used:

-162	Thermocouple/DC Volts Scanner
-176	Voltage Input Connector (or other direct voltage measuring connector)
-164	Transducer Excitation Module
-174	Transducer Excitation Connector

The -162 Thermocouple/DC Volts Scanner is a 20 channel, thermocouple and voltage, signal multiplexing and conditioning scanner. This scanner allows the High Performance A/D Converter to make external direct voltage measurements on four ranges. Connections to external inputs are made through a Voltage Input, Isothermal Input, or AC Voltage Input Connector. The scanner and connector measure the voltages generated across the RTDs in this configuration.

The Transducer Excitation Module, -164, is used to energize the RTDs with an excitation current or the bridge configurations with a voltage. The -174 Transducer Excitation Connector provides connections for accessing the five current sources for exciting up to 4 channels per source, or the one voltage source, for exciting the resistances to be measured.

Installation is covered in Section 3B for each option assembly being used. Briefly, these steps include:

- Addressing

  The addresses for RTD measurement channels are determined by the measurement hardware. Both the position of the Thermocouple/DC Volts Scanner (-162) relative to the High Performance A/D Converter (-161) and the address switch settings on the a/d converter set the address.

- Voltage/Current Excitation Jumper

  Each group of four channels must be configured for voltage or current output by correctly positioning the respective jumper on the Transducer Excitation Connector (-174). Refer to the connector decal for details.

- Physical Installation

  The Transducer Excitation Module/Connector (-164/-174) and the Thermocouple/DC Volts Scanner/connector (-162/-176, -176, or -160) should be installed in adjacent slots to facilitate interconnections. Overall installation considerations are discussed in Section 3B.

- External Connections

  Configuration B RTD measurements are performed with the system connections shown in the in Figures 6h-4 through 6h-6. Note that for Configuration B, the LO to SHIELD connection should be made on each channel.

Configuration C

This configuration supports the same wider mix of applications as Configuration B. The following set of option assemblies is used:

-165	Fast A/D Converter
-176	Voltage Input Connector
-164	Transducer Excitation Module
-174	Transducer Excitation Connector

The -165 Fast A/D Converter supports the same measurement capabilities as the -161 High Performance A/D Converter and -162 scanner do in Configuration B, but at higher reading rates and somewhat lower resolution and accuracy. Connections to external inputs are made through a -176 Voltage Input Connector attached directly to the -165 a/d converter.

The Transducer Excitation Module, -164, is used to energize the RTDs with an excitation current or the bridge configurations with a voltage. The -174 Transducer Excitation Connector provides connections for accessing the five current sources for exciting up to 4 channels per source, or the one voltage source, for exciting the resistances to be measured.

Installation is covered in Section 3B for each option assembly being used. Briefly, these steps include:

o Addressing

The addresses for RTD measurement channels are determined by the measurement hardware. In this configuration, only the address switch settings on the Fast A/D Converter set the address.

- Voltage/Current Excitation Jumper

  Each group of four channels must be configured for voltage or current output by correctly positioning the respective jumper on the Transducer Excitation Connector (-174). Refer to the connector decal for details.

- Physical Installation

  The Transducer Excitation Module/Connector (-164/-174) and the Fast A/D Converter (-165) with connector (-176) should be installed in adjacent slots to facilitate interconnections. Overall installation considerations are discussed in Section 3B.

- External Connections

  Configuration C RTD measurements are performed with the system connections shown in the in Figures 6h-4 through 6h-6. Note that for Configuration C, the LO to SHIELD connection must be made on only one channel per a/d converter.

NOTE

On the input connector, tie LO to SHIELD on each channel when the -161 High Performance A/D Converter and -162 Thermocouple/DC Volts Scanner are used. If the input connector is used directly on the -165 Fast A/D Converter, tie LO to SHIELD on one channel only (per a/d converter.)

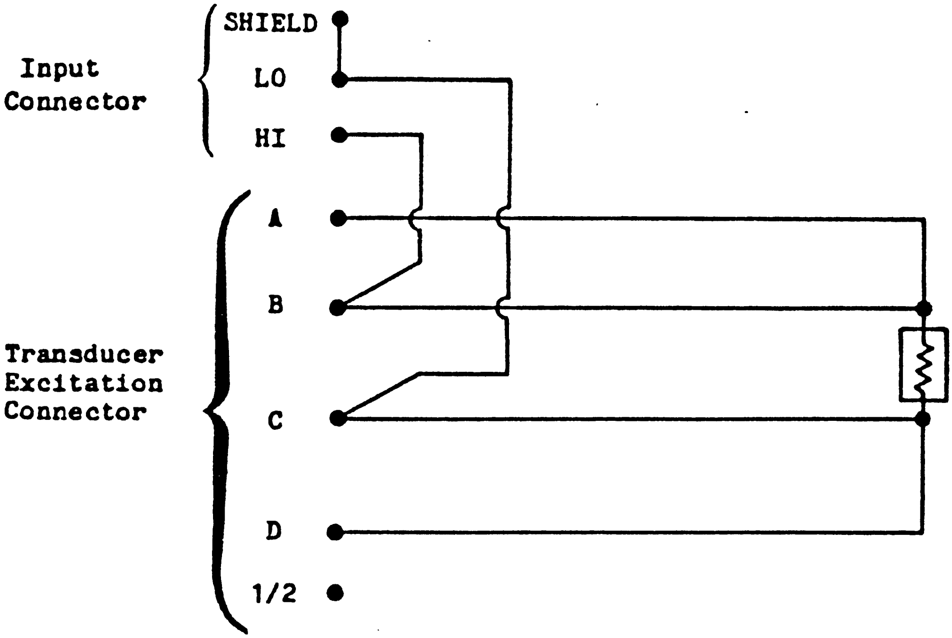


Figure 6h-4. Wiring: 4-Wire, Configuration B, C

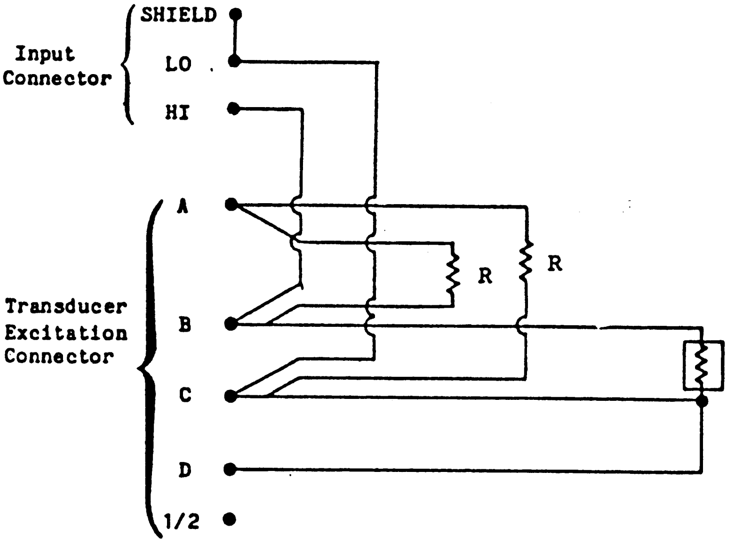
The 3-wire method in Figure 6h-5 uses the +4V, constant voltage supply present on the Transducer Excitation Module. This configuration uses two fairly large resistors as pseudo current sources to both excite the RTD and compensate for lead wire resistance.

- For 100 ohm Pt RTDs, suggested resistors would be 40 kilohms ±0.1% with 5ppm temperature coefficient wire-wound resistors. For other resistors, the resistance value should be sufficiently high so that the effect of the RTD in one of the legs is relatively small. For example, it is desirable to maintain a ratio of about 200:1 between the fixed resistors and the resistor to be measured.

- For 10 ohm Cu RTDs, 4 kilohm resistors should be used.

NOTE

On the input connector, tie LO to SHIELD on each channel when the -161 High Performance A/D Converter and -162 Thermocouple/DC Volts Scanner are used. If the input connector is used directly on the -165 Fast A/D Converter, tie LO to SHIELD on one channel only (per a/d converter.)

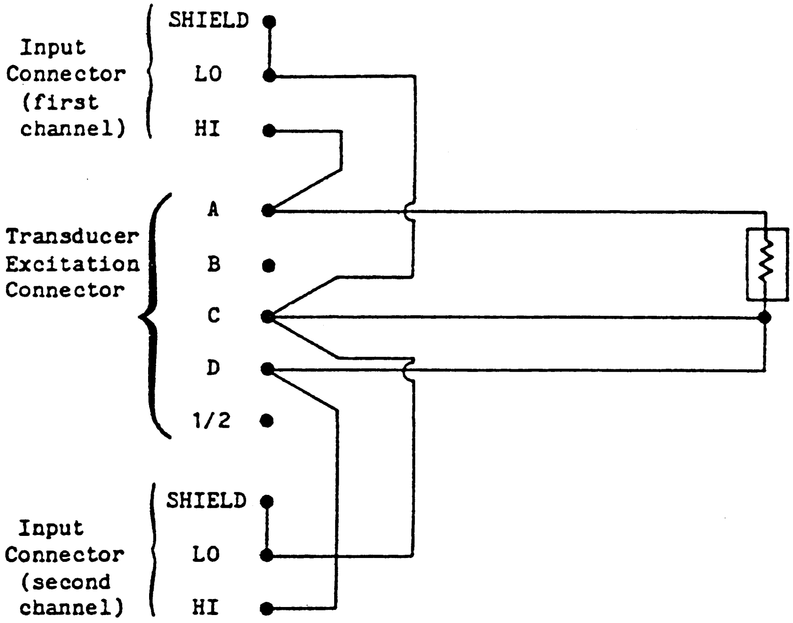


R = User-Supplied Resistor

Figure 6h-5. Wiring: 3-Wire, Constant Voltage, Configuration B, C

NOTE

On the input connector, tie LO to SHIELD on each channel when the -161 High Performance A/D Converter and -162 Thermocouple/DC Volts Scanner are used. If the input connector is used directly on the -165 Fast A/D Converter, tie LO to SHIELD on one channel only (per a/d converter.)


Input Connector (first channel)
SHIELD
LO
HI
Transducer Excitation Connector
A
B
C
D
1/2
Input Connector (second channel)
SHIELD
LO
HI

This method uses the constant current supply in the Transducer Excitation Module. Two voltage measurement channels are employed to remove the effects of lead resistance. One channel measures both the lead resistance and the RTD, and another input channel is used to measure only the lead resistance. The difference between the two channels represents the resistance of the RTD, which is then linearized.

Figure 6h-6. Wiring: 3-Wire, Constant Current, Configuration B, C

USING THE COMMANDS

General

The channel definition (DEF CHAN) statement establishes the TYPE of RTD being measured on a channel or group of channels. TYPE can be set to DIN385 (100 ohms platinum), CU10 (10 ohms copper), or USER (user defined).

```
DEF CHAN(channels) = RTD, TYPE = [DIN385, CU10, or USER]
```

- For DIN385 and CU10, The Front End performs calculations automatically, and no further parameters need be set.

- For USER, constants R0 (R "zero"), ALPHA, DELTA, and C4 can be set. Default values are used for any constants not so specified. The default constants are:

  R0 = 100.00
  Alpha = 3.85E-3
  Delta = 1.45
  C4 = 1.19619E-13

  Once set, user-defined constants persist through re-definition of the original channel. Only a re-definition of the constant or use of the RESET command changes a user-defined constant.

For Configuration A, HI range (maximum 600 °C) or LO range (maximum 425°C) can be specified. With 100-ohm platinum RTDs, HI covers the entire RTD range, but yields less measurement resolution. If HI is used, the channel definition would look like:

```
DEF CHAN(channels) = RTD, TYPE = [DIN385 or USER],&
RANGE = HI
```

For Configurations B and C, the RANGE parameter cannot be changed; one range covers the full span of the RTD.

The SEND CHAN command can be used to obtain an RTD reading in any configuration. Measurements are returned in units set by the TUNIT system variable (Celsius, Fahrenheit, Kelvin, or Rankine).

```
SEND CHAN(channels)
```

Command Examples

Let's define a group of 20 channels for measurements using a 100-ohm platinum RTD. Anticipated temperatures approach 600°C.

```
DEF CHAN(0..19) = RTD, TYPE = DIN385, RANGE = HI
```

Next, define another group of 20 channels for user-defined RTD measurements. The ALPHA constant differs from the default ALPHA value.

```
DEF CHAN(20..39) = RTD, TYPE = USER, ALPHA = 3.92E-3,
```

Note that the four RTD constants remain at their system default values until specifically changed in the channel definition statement. In this example, Alpha is changed, but R0, DELTA, and C4 remain at respective default settings of 100.0, 1.45, and 1.19619E-13.

Now, monitor measurements for all 40 channels:

```
SEND CHAN(0..39)
```

If necessary, check the TUNIT setting with:

```
SEND TUNIT
```

PUTTING IT ALL TOGETHER

The following example can be typed in directly from a terminal (TERM mode) or included in a computer program (COMP mode). Refer to Section 4 under "Operating the Front End from a Terminal/Computer" for more detailed information. Also, Appendix 9A provides guidelines for use with specific types of host computers.

```
MODE = TERM
FORMAT = DECIMAL
TUNIT = CELSIUS
DEF CHAN(0..19) = RTD, TYPE = DIN385
SEND CHAN(0)
```

This example assumes that the High Performance A/D Converter (-161) address is set to 0, that the jumpers on the RTD/Resistance Scanner (-163) are positioned to 4-Wire mode, and that a variable resistor (substituting for an RTD) is connected to channel 0 on the RTD/Resistance Input Connector (-177). The example commands the Front End to operate in terminal mode, sets system variables FORMAT and TUNIT, and defines channels 0 through 19 for DIN 385 RTD measurements. The resulting measurement is made on channel 0 with the SEND CHAN command.

If the variable resistor is set to 100 ohms (representing 0°C), the SEND CHAN(0) command yields:

```
0.00000E+00
```

With the resistor set to 212 ohms (representing 300°C), SEND CHAN(0) yields:

```
3.00000E+02
```

If the response is "9.99999E+37", an error has occurred. Enter the LIST ERROR command to retrieve the associated error message. If necessary, refer to Section 8 for additional error message information.

Section 6I
Temperature Measurement Using Thermistors
RESIST

INTRODUCTION

About this Section

Thermistors are a commonly-used type of variable resistance transducer. The resistance of thermistors varies greatly with temperature, thereby providing a means of making reliable temperature measurements over a range of -100 to +300°C.

This subsection explains the use of the Front End and associated option assemblies for obtaining thermistor temperature readings. Additional information can be found in:

- Section 3: Installation of the Front End mainframe and option assemblies (-161, -163, -177).

- Section 6E: Resistance Measurements (Configuration A only) describes techniques employed for Thermistor Temperature Measurement.

Thermistor Temperature Measurements

Thermistors that are to be measured in bridge configurations, while not specifically covered in this subsection, can be implemented using techniques presented here and in the resistance measurements subsection.

Thermistors exhibit a large temperature coefficient of resistance when compared to other resistance temperature sensors. A large decrease in resistance usually results from a small increase in thermistor temperature, providing a measurement resolution not available with other temperature transducers.

A thermistor's resistance is often determined by passing a known current through it and measuring the resulting voltage drop, as is done by the Front End. The Front End converts this resistance measurement into a temperature reading.

REQUIRED HARDWARE

Thermistor temperature measurement applications require the following option assemblies:

-161	High Performance A/D Converter
-163	RTD/Resistance Scanner
-177	RTD/Resistance Input Connector

Each RTD/Resistance Scanner/Connector selects, excites, and conditions 20 channels. In addition to supporting 4-Wire measurements, this configuration provides lead-wire compensation for performing accurate 3-Wire resistance measurements.

Installation is covered in Section 3B for each option assembly being used. Briefly, these steps include:

- Select the Measurement Mode

  Set jumpers W1 and W2 on the RTD/Resistance Scanner for the desired measurement mode.

- Addressing

  Addresses are determined by the both the position of the RTD/Resistance Scanner (-163) relative to the High Performance A/D Converter (-161) and the address switch settings on that a/d converter.

- Physical Installation

  The RTD/Resistance Input Connector attaches to the RTD/Resistance Scanner.

- External Connections

  4-Wire (4W) Measurement Mode
  Two sense and two excitation wires must be connected to each channel. Lead-wire or reed switch resistances do not affect readings.

  3-Wire Accurate (3WA) Measurement Mode
  One sense and two excitation wires must be connected to each channel. Equal lead-wire resistances are compensated for. Reed switch resistances do not affect readings. Ten channel returns are internally connected together.

  3-Wire Isolated (3WCM) Measurement Mode
  One sense and two excitation wires must be connected to each channel. Equal lead-wire resistances are compensated for. One reed switch resistance error affects readings.

Where lead-wire resistances are very small in comparison with the resistance being measured, as may be true with high resistance thermistors, 2-Wire measurements can be made by setting the scanner to operate in 4-Wire mode, and wiring the -177 input connector properly.

Refer to the -177 option in Section 3 for input wiring diagrams and instructions.

USING THE COMMANDS

General

Thermistor temperature is measured as a resistance that is converted into meaningful temperature units by either:

- The host computer

  A mathematical equation can used in the computer program when a conversion formula is provided by the thermistor manufacturer.

- The Front End

  The Front End interpolation table capabilities can be employed when a table of temperature versus resistance is available.

Use a channel definition to specify thermistor measurement on a channel or group of channels. Since thermistors vary greatly in resistance, the scanner ranges to be used must be chosen to match. Three ranges are provided:

256 ohms
2048 ohms
64 kilohms

Refer to the temperature versus resistance table for the thermistor to be used, and examine the overall possible resistance range for that thermistor as well as the anticipated temperature range of the environment to be measured. Considering both of these variables, set an appropriate MAX parameter value. MAX represents the maximum anticipated value. Its value causes automatic selection of the lowest possible of the three ranges.

```
DEF CHAN(10..12) = RESIST, MAX = 300
```

When known, a conversion formula for the thermistor type can be entered as part of the computer program. When a formula is not known, the following interpolation table approach must be used.

An interpolation table can be used to convert resistance measurements into temperature readings. The table comprises a set of data points that describe the non-linear resistance versus temperature behavior of the thermistor being used.

A maximum of 100 interpolation tables can be defined by the user. Contents for each table are entered as pairs of numbers. Each pair comprises an input value (entered in ascending order), followed by an output value (order determined by related input). Although a minimum of two pairs must be entered, the maximum number is limited only by available memory.

A sampled input falling between two table input values results in an linearly interpolated output. An input falling outside table input values results in a linearly extrapolated output.

The DEF TABLE command is used for entering interpolation table values. The form (input),(output)/(input),(output)/... is used. Start each new line for the same table with "DEF TABLE(n) = /(input),(output)/..."

For thermistors, input is defined as the resistance of the thermistor in resistance range units (e.g., kilohms on the 64 kilohm range). Output is in temperature units. For example, a resistance of 10.25 kilohms (measured on 64 kilohm range) that corresponds to a 25°C temperature reading requires that one table entry be 10.25,25.

Interpolation tables are fully described in Section 5.

Command Examples

A thermistor temperature measurement definition always requires specifying items (1) and (2) below, and may require items (3) and (4).

1. The channel, or group of channels, performing the measurement. E.g., DEF CHAN(0..9)

2. The type of measurement (=RESIST)

3. If necessary, specify the range. Use the MAX parameter. If no MAX parameter is used, the default range (64 kilohms) is used. Use MAX = 200 (specifying the 256 ohm range).

4. If an interpolation table is used, call that table. CHFN=TABLE(0)

If used, an interpolation table must be defined prior to making the channel definition. Enter the thermistor's resistance versus temperature characteristics into the specified interpolation table. Each table entry represents an input resistance figure (in ohms) and a corresponding output temperature figure (in any temperature unit desired). Note that the input figures must be entered in ascending order. To avoid confusion, it is probably best to specify the output temperature in the same units as established for the system (Celsius, Fahrenheit, Rankine, or Kelvin).

Now, let's define an interpolation table, using eleven resistance input to temperature output correlations:

```
DEF TABLE(0) = /132.26,105 /136.09,104 /140.07,103
DEF TABLE(0) = /144.20,102 /148.41,101 /152.75,100
DEF TABLE(0) = /157.35,99 /161.99,98 /166.87,97
DEF TABLE(0) = /171.89,96 /177.16,95
```

Now, we can make the full channel definition:

```
DEF CHAN(0..9) = RESIST, MAX = 200, CHFN = TABLE(0)
```

To measure these channels, use:

```
SEND CHAN(0..9)
```

PUTTING IT ALL TOGETHER

The following example can be typed in directly from a terminal (TERM mode) or included in a computer program (COMP mode). Refer to Section 4 under "Operating the Front End from a Terminal/Computer" for more detailed information. Also, Appendix 9A provides guidelines for use with specific types of host computers.

```
MODE = TERM
FORMAT = DECIMAL
TUNIT = CELSIUS
DEF TABLE(1)=200,40/300,37/500,30/800,20/1000,15
DEF CHAN(0..19)=RESIST,MAX=2000,CHFN=TABLE(1)
SEND CHAN(0)
```

This example assumes that the High Performance A/D Converter (-161) address is set to 0, that the jumpers on the RTD/Resistance Scanner (-163) are positioned to 4-Wire mode, and that a variable resistor (substituting for a thermistor) is connected to channel 0 on the RTD/Resistance Input Connector (-177).

The example commands the Front End to operate in terminal mode and sets system variables FORMAT and TUNIT. The DEF TABLE command creates a table for the conversion of measured inputs to returned readings. The DEF CHAN command defines channels 0 through 19 for resistance measurement on the 2048 ohm range and assigns table 1 to linearize the measured input. The resulting measurement and conversion is accomplished with the SEND CHAN command.

If the variable resistor is set to 1000 ohms, the SEND CHAN(0) command should yield a reading of:

```
1.500000E+01
```

If the variable resistor is set to 200 ohms, the SEND CHAN(0) command should return a reading of:

```
4.00000E+01
```

If the response is "9.99999E+37", an error has occurred. Enter the LIST ERROR command to retrieve the associated error message. If necessary, refer to Section 8 for additional error message information.

Section 6J
Temperature Measurement Using Thermocouples
TC

INTRODUCTION

About This Section

This section explains use of the Front End and associated option assemblies for obtaining thermocouple temperature readings. Detailed information about the physical installation of the Front End mainframe and related options is provided in Section 3.

Thermocouple Temperature Measurement

Thermocouple temperature measurement is frequently used. Thermocouples are relatively rugged and can measure a wide range of temperatures.

This application can use linearizations for any of the following 11 standard thermocouples: JNBS, KNBS, RNBS, SNBS, TNBS, BNBS, CHOS, ENBS, NNBS, JDIN, TDIN.

Thermocouple measurements involve an input from a thermocouple-type temperature sensor. This input is converted to degrees Fahrenheit, Celsius, Kelvin, or Rankine (depending on the TUNIT system variable).

THERMOCOUPLE TEMPERATURE MEASUREMENTS

Thermocouples are very useful temperature measurement sensors. Various types are available for temperatures from cryogenic to above the melting point of steel.

A thermocouple is made of two dissimilar metal conductors. The thermocouple loops shown in Figure 6j-1 are each made up of two such conductors (A and B). Loop (a) shows a thermocouple loop in its simplest form. Conductors A and B are joined twice, creating a thermocouple. The current flow in this loop is related to the temperature difference between reference temperature TR and measurement temperature TM. Measuring this current presents a problem: any connection between a measurement circuit (usually made of copper) and the thermocouple loop would form additional thermocouples.

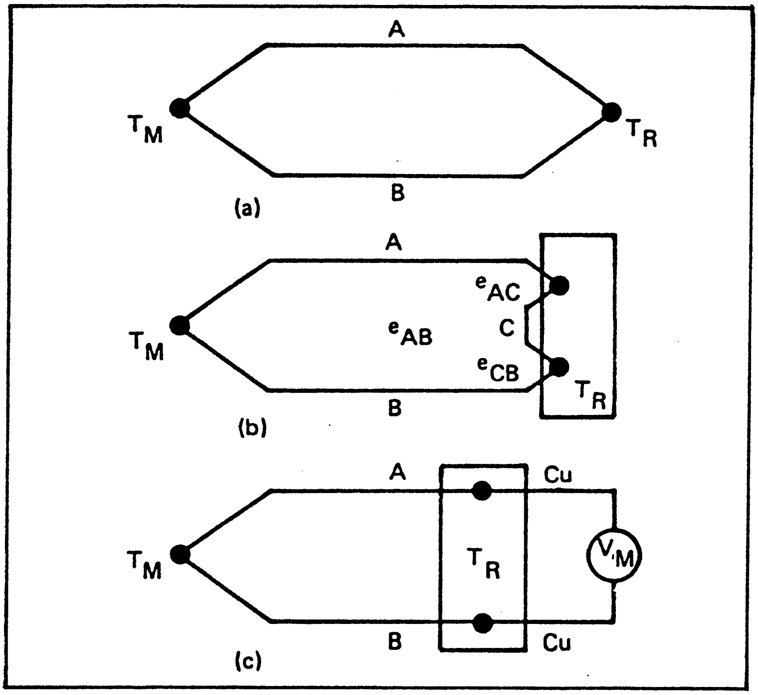


Figure 6j-1. Thermocouple

Loop (b) shows a circuit with connections to an additional material (C). As long as junctions C-A and C-B remain at the same temperature, no additional thermocouples are created. Electrical connections can thereby be made to measure the thermal emf. This is shown in loop (c). The Front End reads a voltage (VM) related to the difference between TM and TR; temperatures and gradients in the copper (C) have no effect. If TR and the emf per degree are known, TM can be determined. The measurement function is nonlinear and unique for each combination of metals used in thermocouple construction.

The Front End automatically performs the thermocouple measurement process. This process includes the following steps:

- Two measurements (VM and TR) are taken.

- Since the thermocouple is usually terminated at a temperature different from zero degrees C, the measured voltage (VM) is compensated using the reference temperature (TR) measurement.

- This compensated voltage is converted to temperature using the voltage/temperature characteristics for the type of thermocouple.

The Front End satisfies these requirements in one of two ways. First, if the thermocouple is terminated at the Isothermal Input Connector (-175), compensation and linearization are performed automatically. Second, if the thermocouple is externally terminated, the external reference junction temperature can be defined to achieve the same result.

For either method, the returned reading can be specified in temperature or compensated millivolts. For compensated millivolts, the built-in thermocouple linearizations are not applied.

REQUIRED HARDWARE

Two hardware configurations can be used for thermocouple temperature measurement.

High Accuracy Hardware

Applications involving thermocouple readings can use the following set of option assemblies for high accuracy:

-161	High Performance A/D Converter
-162	Thermocouple/DC Volts Scanner
-175	Isothermal Input Connector

The High Performance A/D Converter (-161) provides high accuracy analog-to-digital conversion of scanner input voltages. At least one a/d converter must be installed in the Front End or the 2281A Extender Chassis to provide thermocouple input capabilities.

The Isothermal Input Connector contains the necessary hardware to ensure that the thermocouple connections and reference temperature sensor are at the same temperature. The reference temperature sensor portion of the connector is used to measure the temperature of the thermocouple terminations.

The Thermocouple/DC Volts Scanner (-162) is a plug-in, 20-channel thermocouple and multi-voltage range relay scanner. The scanner links the High Performance A/D Converter to external measurement points through the connector.

Installation for any of the option assemblies is detailed in Section 3B of this manual. Reference each option assembly by its option number (-161, -162, or -175). Briefly, installation procedures include:

- Addressing

  Addresses are determined by the both the position of the Thermocouple/DC Volts Scanner (-162) relative to the High Performance A/D Converter (-161) and the address switch settings on that a/d converter.

- Physical Installation

  The Isothermal Input Connector mates with the Thermocouple/DC Volts Scanner. This 20-channel, multi-range relay scanner selects the proper input channel for the High Performance A/D Converter.

Higher Speed Hardware

Applications involving thermocouple readings can use the following set of option assemblies for higher speed measurements:

-165	Fast A/D Converter
-175	Isothermal Input Connector

The Fast A/D Converter (-165) provides faster analog to digital conversion of voltages from the Isothermal Input Connector (-175). The connector attaches directly to the a/d converter in this configuration; no separate scanner is required.

The Isothermal Input Connector contains the necessary hardware to ensure that the thermocouple connections and reference temperature sensor are at the same temperature. The reference temperature sensor portion of the connector is used to measure the temperature of the thermocouple terminations.

Temperature Measurement Using Thermocouples

Installation for any of the option assemblies is detailed in Section 3B of this manual. Briefly, installation procedures for the faster hardware configuration include:

- Addressing

  Addresses are determined only by address switch settings on the Fast A/D Converter.

- Physical Installation

  The Isothermal Input Connector attaches to the Fast A/D Converter.

Thermocouple Connections

Each input channel uses three lines (HIGH, LOW, SHIELD) on the input connector. Both hardware configurations (High Accuracy or Higher Speed) use the same thermocouple connection guidelines for LOW and HIGH.

- LOW

  The RED wire on the thermocouple is the negative lead and must be attached to the LOW terminal.

- HIGH

  The wire color of the positive lead varies with the type of thermocouple. The color appropriate for a given thermocouple is defined in Table 6j-1.

The SHIELD connection differs between High Accuracy and Higher Speed hardware configurations. With High Accuracy hardware, first determine if the thermocouple has a shield; thermocouples with only two connecting wires do not have a shield. For these unshielded thermocouples, connect a jumper between LOW and SHIELD on the Isothermal Input Connector.

If the thermocouple used with High Accuracy hardware is shielded, next determine if the shield is electrically connected to the thermocouple.

1. Connect shields that are electrically connected to the thermocouple to the SHIELD terminal on the Isothermal Input Connector.

2. Connect shields that are floating (no electrical connection to the thermocouple) to an earth ground at the thermocouple.

3. Then connect LO to SHIELD at the input terminals on the -175 connector.

For Higher Speed Hardware (-165 A/D Converter), SHIELD is treated as COMMON, and the following additional considerations must be made:

1. If the thermocouple is completely isolated from ground, connect LOW to COMMON (SHIELD) on the Isothermal Input Connector. This arrangement yields the best noise rejection.

2. If the thermocouple is electrically connected to some potential with respect to ground, connect only one of the COMMON (SHIELD) terminals on the Isothermal Input Connector to a potential that is within ±10V of both HIGH and LOW terminals on all channels.

USING THE COMMANDS

Internal Reference Junction Measurements

Usually, the Isothermal Input Connector is used as the reference junction. Each connector supports 20 sets of thermocouple terminals.

Thermocouples terminated at the Isothermal Input Connector use permanently stored temperature compensation and voltage/temperature linearization algorithms.

The Front End only needs to know what type of thermocouple is connected to the channel. The conversion algorithms to support 11 thermocouple types are handled internally. These thermocouples are further described in Table 6j-2.

Use the channel definition statement to define the channels and the thermocouple TYPE. For example, designate channel 100 for automatic compensation and linearization of a "K" type thermocouple.

```
DEF CHAN(100) = TC, TYPE = KNBS
```

To read the channel 100 measurement, enter:

```
SEND CHAN(100)
```

The returned reading is in the system temperature units TUNIT (Celsius, Fahrenheit, Kelvin, or Rankine). To check the TUNIT value in effect:

```
SEND TUNIT
```

To change to a different value:

```
TUNIT = FAHRENHEIT (or other)
```

External Reference Junction Measurements

Sometimes, the Isothermal Input Connector reference junction will not be used. The thermocouples are then attached to the HI and LOW terminal pairs with copper wires from the external reference junction.

Setting up an externally terminated measurement requires an additional definition for the reference junction temperature (RJTEMP) parameter. This must be done in the same units as specified by the system temperature unit (TUNIT). If necessary, use the SEND TUNIT command for verification.

Use the channel definition statement to define channels 40 through 59 for external termination, measuring temperature with T-type thermocouples. The external junction temperature is defined as 23.4 degrees (in system temperature units).

```
DEF CHAN(40..59) = TC, TYPE = TDIN, RJTEMP = 23.4
```

PUTTING IT ALL TOGETHER

The following example can be typed in directly from a terminal (TERM mode) or included in a computer program (COMP mode). Refer to Section 4 under "Operating the Front End from a Terminal/Computer" for more detailed information. Also, Appendix 9A provides guidelines for use with specific types of host computers.

```
MODE = TERM
FORMAT = DECIMAL
TUNIT = FAHRENHEIT
DEF CHAN(0..19) = TC, TYPE = KNBS
SEND CHAN(0)
```

This example assumes that the High Performance A/D Converter (-161) or the Fast A/D Converter (-165) address is set to 0 and that a KNBS type thermocouple is connected to channel 0 on the Isothermal Input Connector. It commands the Front End to operate in terminal mode, sets system variables FORMAT and TUNIT, and defines analog input channels 0 through 19 for KNBS type thermocouple measurement. The resulting measurement is accomplished on channel 0 by the SEND CHAN command and converted into a degrees Fahrenheit reading.

For example, if the measured temperature is 23°C, the returned reading would be:

```
7.34000E+01
```

If the response is "9.99999E+37", an error has occurred. Enter the LIST ERROR command to retrieve the associated error message. If necessary, refer to Section 8 for additional error message information.

Table 6j-1. Thermocouple Lead Wire Color Code

ANSI Designation	Lead Material Positive (+)	 Negative (-)	Color of Positive Lead
J	Iron	Constantan	White
K	Chromel	Alumel	Yellow
T	Copper	Constantan	Blue
E	Chromel	Constantan	Purple
R	Platinum	Platinum (13% Rhodium)	Black
S	Platinum	Platinum (10% Rhodium)	Black
C (W5Re/ W26Re)	Tungsten (5% Rhenium)	Tungsten (26% Rhenium)	White
B	Platinum (30% Rhodium)	Platinum (6% Rhodium)	Gray

Table 6j-2. Thermocouple Temperature Ranges and Applications

Select	Type	Parameter	Usable Range (^{o}C)
J	JNBS (Can be used in oxidizing, reducing, inert, or vacuum atmospheres.)	NBS J	-200 to 760
K	KNBS (Should not be used in reducing or sulphurous atmospheres. Can only be used in vacuum for short time until calibration shifts.)	NBS K	-225 to 1350
T	TNBS (Can be used in oxidizing, reducing, inert, or vacuum atmospheres.)	NBS T	-230 to 400
E	ENBS (Can be used in oxidizing or inert atmospheres. Should not be used in reducing or vacuum atmospheres.)	NBS E	-250 to 838
R	RNBS	NBS R	0 to 1767
S	SNBS	NBS S	0 to 1767
B	BNBS	NBS B	200 to 1820
N	NNBS	NISIL-NICROSIL	-250 to 400
C	CNBS	W5Re/W26Re	0 to 2315
V	TDIN	DIN T	-200 to 600
H	JDIN	DIN J	-200 to 900

Section 6K
Totalizing Measurement
TOTAL

INTRODUCTION

About this Section

This section explains how to use the Front End for counting events. Detailed information on installing the Front End and its options is found in Section 3.

Totalizing Measurement

To the electronic circuitry in the Front End an event is a transition between one voltage state and another. Event-counting is commonly used on production lines for counting items, and in monitoring the flow of gasses or liquids.

The Front End counts events by detecting the transitions from a high to a low voltage state. The boundary between these states may vary for different types of signals, so the hardware includes adjustments that accomodate different signal levels.

HARDWARE DESCRIPTION

To count events, the following option is needed:

-167 Counter/Totalizer

Each Counter/Totalizer option supports six event counting channels.

The installation instructions for the Counter/Totalizer are found in Section 3B of this manual. Consult these instructions as necessary.

Hardware Preparation

The Counter/Totalizer adjustments allow for measurement of a variety of signal types. The reference voltage and input deadband adjustments define the high and low voltage thresholds of the input. Debouncers and input pull-ups allow accurate counting of contact closures.

The terminal assignments for the Counter/Totalizer connector are listed on the rear panel. The maximum input voltage is plus or minus 15 volts dc or ac peak.

CAUTION

The six channels on a Counter/Totalizer assembly are isolated from the Front End chassis and from ground but not from each other. The return lines on the input connector are common. All return lines must be connected to the same voltage.

To prepare a Counter/Totalizer for event-counting, refer to Figure 6k-1 and perform the following steps for event-counting channels:

- COUNT Function

  Move the channel function switch to the COUNT position. Notice that the channels are grouped into three pairs: channels 0 and 1, channels 2 and 3, and channels 4 and 5. Both channels in a pair must have the same function.

- Debounced Input

  Set the position of the debounced/direct input switch. For counting contact closures, move the switch to the DEBOUNCED INPUT position. This turns the debouncer on.

  For counting events signaled other than by the closing of a contact, move the switch to the DIRECT INPUT position. This turns the debouncer off. In this configuration, the input signal is fed directly into the counter.

- Floating/Pulled-Up Input

  Note the position of the floating/pulled-up resistor network. For counting contact closures, move the resistor network to the PULLED UP position. In this configuration, each channel input is connected through a pull-up resistor to the positive power supply on the Counter/ Totalizer assembly. If you are using DIRECT INPUT, move the resistor network to the FLOATING position.

- Addressing

  Set the hundreds and tens address switches to the first address for the group of six channels.

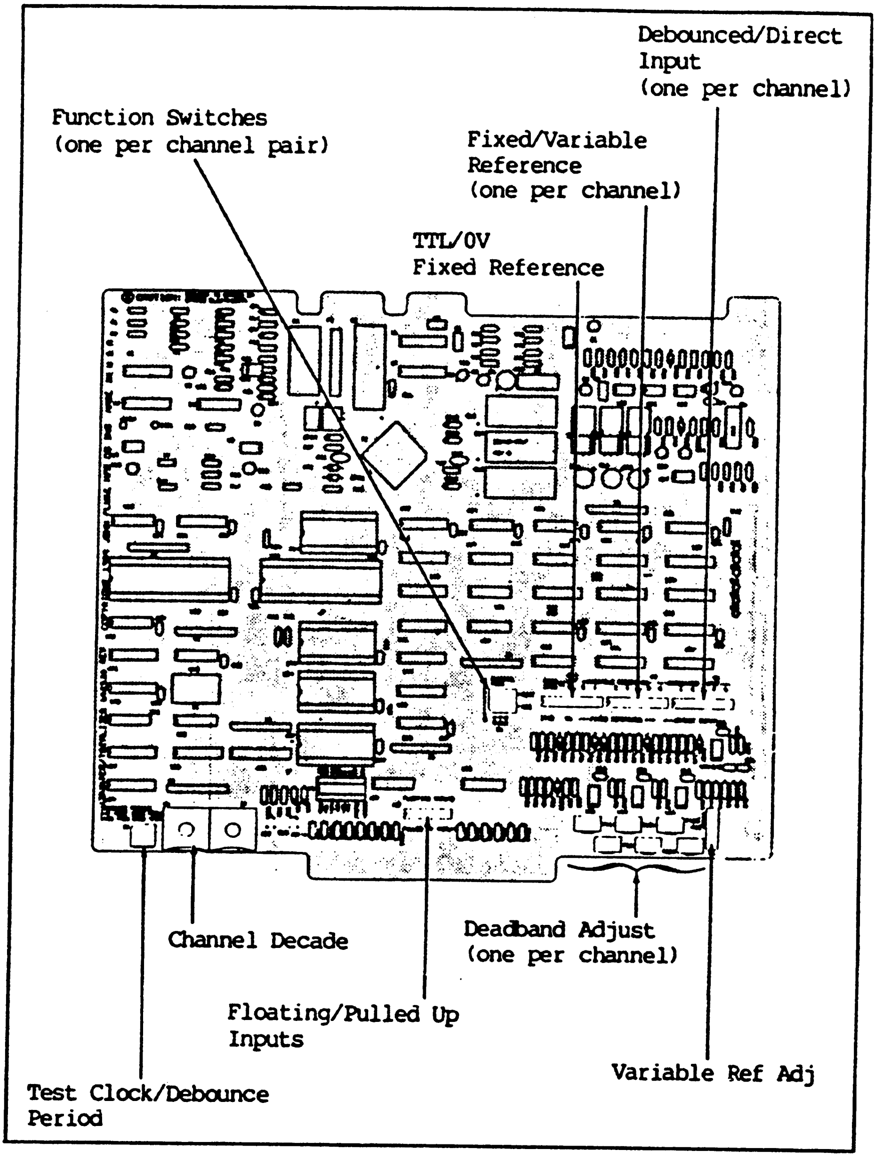

Debounced/Direct Input (one per channel)
Function Switches (one per channel pair)
Fixed/Variable Reference (one per channel)
TTL/0V Fixed Reference
Channel Decade
Deadband Adjust (one per channel)
Floating/Pulled Up Inputs
Variable Ref Adj
Test Clock/Debounce Period

Figure 6k-1. Counter/Totalizer Adjustments

Adjusting Input Thresholds

To ensure that the input signals are detected accurately, the Counter/Totalizer input thresholds must be properly adjusted. Figure 6k-2 shows how input voltage levels are detected by the Counter/Totalizer. The reference voltage defines the boundary between the high and low voltage states of the input signal. Deadband prevents input noise from being counted as events. As shown in the figure, the reference voltage and deadband define the high and low input voltage thresholds.

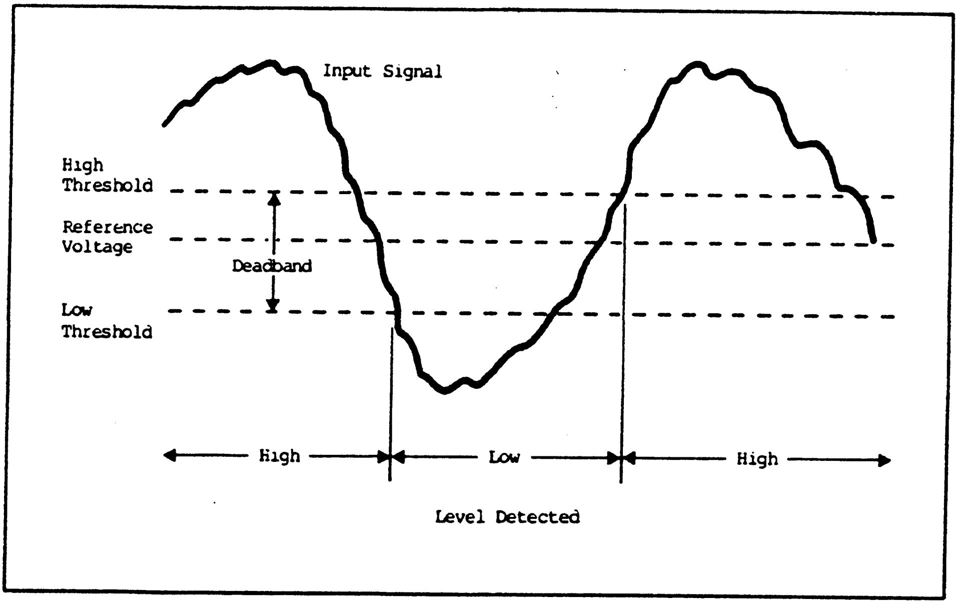


Figure 6k-2. Input Level Detection

Totalizing Measurement

On the Counter/Totalizer assembly, the reference voltage and deadband are adjustable. For each channel, a switch selects either a fixed or variable reference voltage. There is one variable reference on the assembly, adjustable from -10 to +10 volts. There is also one fixed reference, which is selectable for either 0 volts or 1.4 volts (TTL level). The deadband is adjustable from 0 to 3 volts for each channel.

Adjusting Reference Voltage

To choose the proper reference voltage, follow these guidelines:

- If the input signal is centered around 0 volts, select the 0 volt fixed reference.

- If the input signal is TTL compatible, select the TTL fixed reference.

- If the input signal is neither of the above, select the variable reference. To adjust the variable reference voltage, connect a voltmeter to the variable reference terminal and one of the return terminals on the rear panel connector. Using a small screwdriver, turn the variable reference adjustment screw until the desired voltage is displayed by the voltmeter.

- If the input is a contact closure, and the input pull-ups are being used to pull the input voltage to a high level, select the variable reference and adjust it to 7 volts. This level is half way between the high and low input voltages. (When the contacts are closed, the input is 0 volts. When the contacts are open, the input is pulled to +14 volts.)

Adjusting Deadband

Use the input deadband adjustment method most suitable for your application.

- Method 1. For TTL signals, turn the deadband adjustment screw counterclockwise until it stops. Then turn the screw clockwise to the position marked TTL on the rear panel.

- Method 2. When counting contact closures, the deadband adjustment is not important. Instead, the debouncer must be used. See the paragraphs that follow on using the debouncer.

- Method 3. Temporarily select the 0 volt fixed reference for the channel in question. Remove any connections to the input terminal. Connect a voltmeter to one of the return terminals and to the threshold output terminal for this channel. Disregarding the polarity of the threshold voltage, turn the appropriate deadband adjustment screw until the threshold is one-half the desired deadband voltage. Restore the reference voltage to its original setting.

- Method 4. This method can be used when the input signal has a steady frequency. Set the function switch for the channel in question to FREQ. Set-up the channel to the frequency function. Connect the signal to the channel input terminals, as for normal operation. Turn the appropriate deadband adjustment screw counterclockwise until it stops. Sample the channel and turn the deadband screw clockwise until a stable frequency reading is obtained. Return the channel to the COUNT function.

To ensure accurate measurements, adjust the deadband for each channel as high as possible.

Checking Threshold Voltages

The combination of reference voltage and deadband determines the high and low threshold voltages. To check the threshold levels, use the following procedure:

- Connect a voltmeter to one of the return terminals and to the threshold output terminal for the channel in question. Connect the test clock output to the channel input terminal.

- Turn the test clock switch to position 0 (+14 volt output), and read the low threshold on the voltmeter.

- Turn the test clock switch to position 1 (-15 volt output), and read the high threshold on the voltmeter.

Using the Debouncer

Relays, switches, and other mechanical devices frequently exhibit contact bounce when their contacts are opened or closed. In an event-counting application, contact bounce can cause extra, unwanted events to be counted. To prevent this, each Counter/Totalizer channel supports a debounce feature. The debouncer eliminates counting errors due to mechanical contact bounce.

Figure 6k-3 illustrates debouncer operation. When a debounced input is selected, the input signal must remain stable longer than the debounce period before a new input level will be recognized. There are three debounce periods available: 4 ms, 20 ms, and 80 ms. The debounce period is selected by turning a switch.

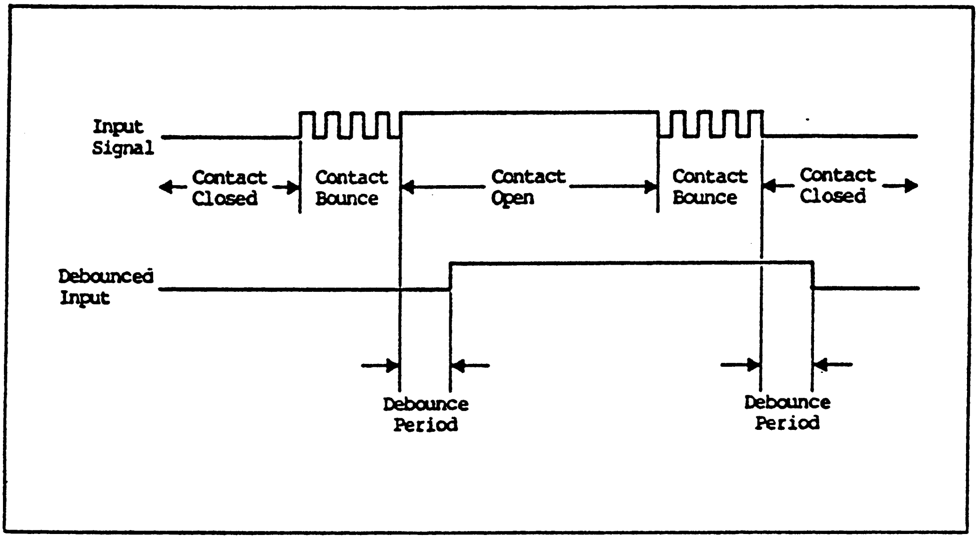


Figure 6k-3. Debouncer Operation

To set up the debouncer, use the following steps:

- Verify that the Counter/Totalizer has been configured properly for counting contact closures. The debounced/direct input switch for the channel in question should be in the DEBOUNCED INPUT position. The floating/pulled-up resistor network should be in the PULLED UP position. The reference voltage switch for the channel should be set to VARIABLE REFERENCE, with the reference adjusted to 7 volts.

- Using a small screwdriver, turn the debounce period switch to position 0 (4 MSEC).

- Move the channel function switch to the COUNT position.

- Connect the channel input terminal to one contact. Connect the return terminal to the other contact.

Totalizing Measurement

- o Set-up the channel to the event-counting function and sample the channel. Open and close the contact and notice the change in the sampled channel value. Increase the debounce period, if necessary, until each contact closure is counted only once.

USING THE COMMANDS

General

The DEF CHAN statement is used in defining pairs of totalizing measurement channels. A conflict with some hardware settings on the Counter/Totalizer Board must not be introduced.

- o First, each pair must consist of consecutive even and odd channels, corresponding to those selected on the function switch (Figure 6k-1).

- o Second, the channels must fall within the range set by the channel decade switches (Figure 6k-1). Note that these switches establish the hundreds and tens designations for a range of channels.

For example, if the function switch is set for TOTAL on channels 2 and 3, and the channel decade switches are set for 1 and 0 (meaning 100), the following definition can be made:

```
DEF CHAN(102, 103) = TOTAL
```

An event count can be obtained by entering two successive SEND CHAN commands:

```
SEND CHAN(102)
SEND CHAN(102)
```

The first command (at t1) returns the count total at that point. Since the time of any SEND CHAN command prior to this may not be known, ignore this initial count. The second command (at t2) returns the count accumulated since the first command. Each additional command returns only the count since the previous command. These incremental counts can then be manipulated within the host computer program.

To prevent overrange readings, keep the interval between readings short. At its maximum counting rate of 400 kilohertz, a Counter/Totalizer channel must be monitored at least once every 20 seconds to ensure valid readings.

Command Examples

Let's set up a totalizing channel and take some counts. First, use a channel definition to set up one or more pairs of channels for event counting.

```
DEF CHAN(100..103)=TOTAL
```

Now, since the totalizer is already running, clear the count for one channel:

```
SEND CHAN(100)
```

Ignore this initial response (t0).

Then, after the desired interval (less than 20 seconds), enter:

```
SEND CHAN(100)
```

Totalizing Measurement

This time (t1), the response equals only the counts since the previous (in this case - initial) measurement. Assuming 230 counts were made in this t0-t1 interval, the t1 response is:

```
2.30000E+02
```

At t2, use SEND CHAN(100) again. This interval (t1-t2) yields 738 counts, and the t2 response is:

```
7.38000E+02
```

The t1 and t2 responses can then be used to find the total count from t0 to t2 (= t1 + t2).

```
---------|-----|----------------|----------------------
         t0    t1               t2

-discard-|-230-|    - 738 -     |
```

PUTTING IT ALL TOGETHER

The following example can be typed in directly from a terminal (TERM mode) or included in a computer program (COMP mode). Refer to Section 4 under "Operating the Front End from a Terminal/Computer" for more detailed information. Also, Appendix 9A provides guidelines for use with specific types of host computers.

```
MODE = TERM
FORMAT = DECIMAL
DEF CHAN(0..5) = TOTAL
SEND CHAN(0)
```

This example assumes that the Counter/Totalizer (-167) address is set for 00, the channel function switches are set to the COUNT position, the debounce/direct input switches are set to Debounce, and the floating/pulled-up resistor network is moved to the Floating position.

The example commands the Front End to operate in terminal mode, sets the system variable FORMAT, and defines channels 0 through 5 for totalizing measurements. The resulting measurement is made on channel 0 with the SEND CHAN command.

A sample reading can be obtained by wiring the test clock output on the Counter/Totalizer rear panel connector to the channel 0 input terminal. The terminal assignments for the connector are listed on the rear panel. Since the test clock output and the channel inputs use the same ground, it is not necessary to wire a return line.

Using a small screwdriver, move the test clock switch to position 3 (50HZ W/BOUNCE). Make repeated measurements of the channel 0 input by entering a SEND CHAN(0) command every five seconds. Ignore the first response. Each successive response represents the accumulated count since the preceding SEND CHAN(0) command. In this example, responses of approximately 250 counts should be returned. This response appears as:

```
2.50000E+02
```

If the response is "9.99999E+37", an error has occurred. Enter the LIST ERROR command to retrieve the associated error message. If necessary, refer to Section 8 for additional error message information.

Section 6L
Voltage Measurement - Alternating
AVIN

INTRODUCTION

About This Section

This section explains use of the Front End and associated option assemblies for making alternating voltage readings. Detailed information about the physical installation of the Front End mainframe and related options is provided in Section 3.

Alternating Voltage Measurement

The alternating voltage capability of the Front End is suitable for measuring low frequency (45 to 450 Hz) sine wave voltage signals. Power line sampling is the most common use of the alternating voltage measurement function.

REQUIRED HARDWARE

In addition to the standard Front End mainframe, the following three option assemblies are required for alternating voltage measurements:

-161	High Performance A/D Converter
-162	Thermocouple/DC Volts Scanner
-160	AC Volts Input Connector

The High Performance A/D Converter (-161) provides high accuracy analog-to-digital conversion of scanner input voltages.

Voltage Measurement - Alternating

The Thermocouple/DC Volts Scanner (-162) is a plug-in, 20-channel thermocouple and multi-voltage range relay scanner. The scanner links the High Performance A/D Converter to external measurement points.

The AC Voltage Input Connector provides terminal connections for routing 20 voltage input channels to the scanner. Of these 20 channels, ten are designated for alternating voltage inputs. The remaining ten are designated for direct voltage inputs. The AC Voltage Input Connector attaches to the scanner; it can be installed or removed without removing the scanner.

Installation for any of the option assemblies is detailed in Section 3B of this manual.

As an overview, installation requires the following actions:

- Addressing

  Addresses are determined by the both the position of the Thermocouple/DC Volts Scanner relative to the High Performance A/D Converter (-161) and the address switch settings on that a/d converter.

- Wiring Connections

  For each channel, make connections to the HI and LO terminals.

- Physical Installation

  Each AC Volts Input Connector attaches to a Thermocouple/DC Volts Scanner. Each connector/scanner interfaces with the serial link and can be installed in either the Front End mainframe or the 2281A Extender Chassis.

USING THE COMMANDS

General

Alternating voltage measurement set-up requires no more than assigning the appropriate function to the desired channel or group of channels. The range is preset at 250V rms. Any combination of channels can be monitored with the SEND CHAN command.

Command Example

For example, the following command would be used to define a block of four alternating voltage input channels.

```
DEF CHAN(40..43) = AVIN
```

NOTE

> Alternating voltage can be measured on those channels distinguished by an even-numbered tens digit (21, 49, 84, etc). Thus, any of the first 10 channels in a block of 20 can be defined as "AVIN".

Sample any of these channels with the SEND CHAN command.

```
SEND CHAN(40..43)
```

PUTTING IT ALL TOGETHER

The following example can be typed in directly from a terminal (TERM mode) or included in a computer program (COMP mode). Refer to Section 4 under "Operating the Front End from a Terminal/Computer" for more detailed information. Also, Appendix 9A provides guidelines for use with specific types of host computers.

```
MODE = TERM
FORMAT = DECIMAL
DEF CHAN(0..9) = AVIN
SEND CHAN(0)
```

This example assumes that the High Performance A/D Converter (-161) address is set to 0 and that an alternating voltage source is connected to the HI and LO terminals for channel 0 on the AC Voltage Input Connector (-160). It commands the Front End to operate in terminal mode, sets system variable FORMAT, and defines analog input channels 0 through 9 for an alternating voltage measurement. The resulting measurement is made on channel 0 with the SEND CHAN command.

For example, if an rms value of 132V at a frequency of 60 Hz is measured, the returned value is:

```
1.32000E+02
```

If the response is "9.99999E+37", an error has occurred. Enter the LIST ERROR command to retrieve the associated error message. If necessary, refer to Section 8 for additional error message information.

Section 6M
Voltage Measurement - Direct
DVIN

INTRODUCTION

About this Section

Sensors outputting a dc voltage are commonly encountered in the measurement environment.

This section explains use of the Front End and associated option assemblies for obtaining dc voltage readings. Detailed information about the physical installation of the Front End mainframe and related options is provided in Section 3.

Direct Voltage Measurement

Direct voltage measurement is a relatively simple use of the Front End. Numerous applications require dc voltage measurement. For example, many industrial sensors output 0 to 5 volts to indicate 0% to 100% of the measured parameter.

REQUIRED HARDWARE

Two hardware configurations can be used for direct voltage measurement.

High Accuracy Hardware

High accuracy direct voltage measurements can be made with the following hardware:

-161	High Performance A/D Converter
-162	Thermocouple/DC Volts Scanner

One of the following connector assemblies is required:

-176	Voltage Input Connector (recommended)
-160	AC Voltage Input Connector (10 direct voltage channels)
-175	Isothermal Input Connector (20 direct voltage channels)

The High Performance A/D Converter (-161) provides high accuracy analog-to-digital conversion of scanner input voltages.

The Thermocouple/DC Volts Scanner (-162) is a plug-in, 20-channel thermocouple and multi-voltage range relay scanner. The scanner links the High Performance A/D Converter to external measurement points.

The input connector provides terminal connections for routing 20 (with -175 or -176 option) or 10 (with -160 option) direct voltage input channels to the scanner. This assembly attaches to the scanner: it can be installed or removed without removing the scanner.

Installation for any of the option assemblies is detailed in Section 3B of this manual. Reference each option assembly by its option number (-161, -162, or -176/-160/-175).

As an overview, installation requires the following actions:

- Addressing

  Addresses are determined by the both the position of the Thermocouple/DC Volts scanner relative to the High Performance A/D Converter (-161) and the address switch settings on that a/d converter.

- Physical Installation

  The input connector (-175, -176, or -160) attaches to a Thermocouple/DC Volts Scanner.

- Wiring Connections

  For inputs to the Voltage Input Connector or the Thermocouple Input Connector, each input channel includes three lines (HIGH, LOW, SHIELD). The voltage is measured across HIGH and LOW.

  The following considerations apply to use of SHIELD:

  1. To provide maximum protection from common-mode noise voltages, attach SHIELD to LOW at the point of measurement. Use a shield line from the Front End connector to the measurement point.

  2. To provide maximum rejection of radio frequency interference (RFI) or electro-magnetic interference (EMI), attach SHIELD to LOW at the connector with as short a wire as possible.

  3. SHIELD must never be left disconnected or connected to HIGH or the chassis ground. For more information, see Fluke Application Bulletin AB-20 (guarded signal measurements.)

Higher Speed Hardware

Higher speed direct voltage measurements can be made with the following hardware:

-165	Fast A/D Converter

One of the following connector assemblies is required:

-176	Voltage Input Connector (recommended)
-175	Isothermal Input Connector (also allows thermocouple inputs)

(the -160 connector is not suitable for use with the Fast A/D Converter)

The -165 Fast A/D Converter provides higher reading rates than the -161 High Performance A/D Converter, but with somewhat lower accuracy and noise rejection.

The input connector provides terminal connections for routing 20 differential or 40 single-ended direct voltage input channels to the a/d converter. The connector attaches directly to the a/d converter in this configuration; no separate scanner is required. The connector can be installed or removed without removing the a/d converter.

Single-ended measurements are a cost effective way to measure many inputs with a minimal amount of hardware. However, this configuration works well only with signals of greater than 1V in relatively noise-free applications. Single-ended measurements also require that all signals share a common LOW connection.

Installation for any of the option assemblies is detailed in Section 3B of this manual.

As an overview, installation requires the following actions:

- Addressing

  Addresses are determined only by the address switch settings on the Fast A/D Converter.

- Physical Installation

  The input connector (-175 or -176) attaches directly to the Fast A/D Converter.

- Wiring Connections

  For inputs to the Voltage Input Connector or the Thermocouple Input Connector, each input channel includes three lines (HIGH, LOW, SHIELD). The voltage is measured across HIGH and LOW.

  For Higher Speed Hardware (-165 A/D Converter), SHIELD is treated as COMMON, and the following additional considerations must be made:

  1. If the input is completely isolated from ground, connect LOW to COMMON (SHIELD) on the input connector. This arrangement yields the best noise rejection.

  2. If the input is electrically connected to some potential with respect to ground, connect only one of the COMMON (SHIELD) terminals on the input connector to a potential that is within ±10V of both HIGH and LOW terminals.

  3. If it is uncertain whether case (1) or (2) above applies, assume case (2).

USING THE COMMANDS

General

A direct voltage input channel is set-up with a DEF CHAN command line. This specifies the channel number (or group of channel numbers).

Optionally, the range can be set-up with the MAX parameter (at the end of the DEF CHAN statement). MAX specifies the maximum expected value; the Front End then selects the lowest possible range that will accommodate that expected value. Ranges available include:

- High Accuracy Hardware (-161 A/D Converter)

  64 mV, 512 mV, 8V, 64V

- Higher Speed Hardware (-165 A/D Converter)

  64 mV, 512 mV, 8V, 10.5V

If no MAX designation is made, the highest range (64V or 10.5V) is selected.

Any combination of channels can be monitored with the SEND CHAN command.

Command Example

Let's set channels 20 through 29 for dc voltage input, using the 8V range. We'll set the range with the MAX parameter, specifying a maximum anticipated voltage of 0.8 volts. Since this voltage is out-of-range for the 512 mV range, the lowest range available is 8V.

```
DEF CHAN(20..29) = DVIN, MAX = 0.8
```

Now, make a measurement on any of these channels using the SEND CHAN command.

```
SEND CHAN(20..29)
```

PUTTING IT ALL TOGETHER

The following example can be typed in directly from a terminal (TERM mode) or included in a computer program (COMP mode). Refer to Section 4 under "Operating the Front End from a Terminal/Computer" for more detailed information. Also, Appendix 9A provides guidelines for use with specific types of host computers.

```
MODE = TERM
FORMAT = DECIMAL
DEF CHAN(0..19) = DVIN, MAX = 0.4
SEND CHAN (0)
```

This example assumes that the High Performance A/D Converter (-161) or Fast A/D Converter (-165) address is set to 0 and that a dc voltage source (350 mV) is connected to the HIGH and LOW terminals for channel 0 on the Voltage Input Connector (-176). Also, it is assumed that the SHIELD terminal is shorted to the LOW terminal and that proper polarity has been observed. The example commands the Front End to operate in terminal mode, sets the system variable FORMAT, and defines analog input channels 0 through 19 for a direct voltage measurement on the 512 mV range.

The resulting measurement is made on channel 0 with the SEND CHAN command. If 350 mV is measured, the returned reading will be:

```
3.50000E-01
```

If the response is "9.99999E+37", an error has occurred. Enter the LIST ERROR command to retrieve the associated error message. If necessary, refer to Section 8 for additional error message information.

Section 7
Maintenance

CONTENTS

INTRODUCTION

Most operator maintenance does not require direct access to the interior of the Front End. Only the procedure to change the line voltage setting requires the unit to be opened. Do not perform this procedure unless you are qualified to do so and thoroughly understand the task before beginning. All other maintenance that requires access to the Front End interior is covered fully in the Service Manual.

SETTING THE LINE VOLTAGE

NOTE

Setting the line voltage may not be necessary on your unit. Some units use power supplies that automatically adjust to line power voltage. Check the rear panel to identify these units. If there is no specific line voltage marked by the input power connection, your unit uses an auto-switching power supply; no separate line voltage setting is necessary, and the following procedure need not be followed.

WARNING

THE FOLLOWING PROCEDURE REQUIRES ACCESS TO THE INTERIOR OF THE FRONT END. DO NOT PERFORM THIS PROCEDURE UNLESS YOU ARE QUALIFIED TO DO SO. LETHAL VOLTAGES MAY EXIST WITHIN THE UNIT.

The power input setting is normally marked on the rear panel of the Computer Interface Module, just above the power input connector. If there is no mark in either box, or if the box is marked for a voltage other than the one you will be using, use the following procedure to gain access to the internal setting.

1. Turn the Front End power switch to OFF.

2. Remove the ac input line cord from the power source and from the Front End.

3. Remove the four Phillips-head screws indicated in Figure 7-1 below.

4. See Figure 7-2. Slide the Module out by grasping the finger indentation in the fan filter hole and slide it straight back and out.

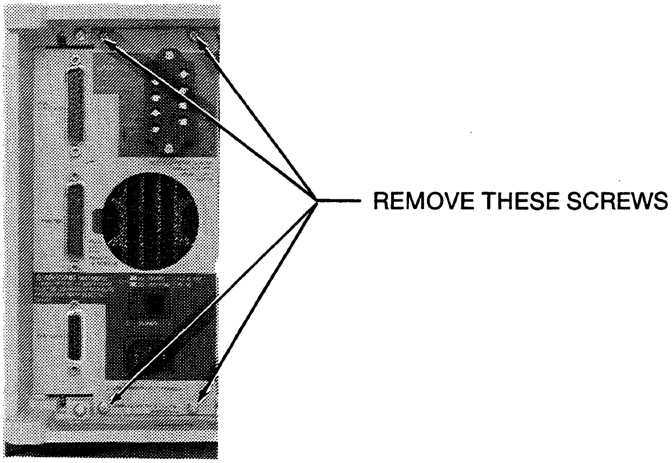


Figure 7-1. Power Supply Removal Screws

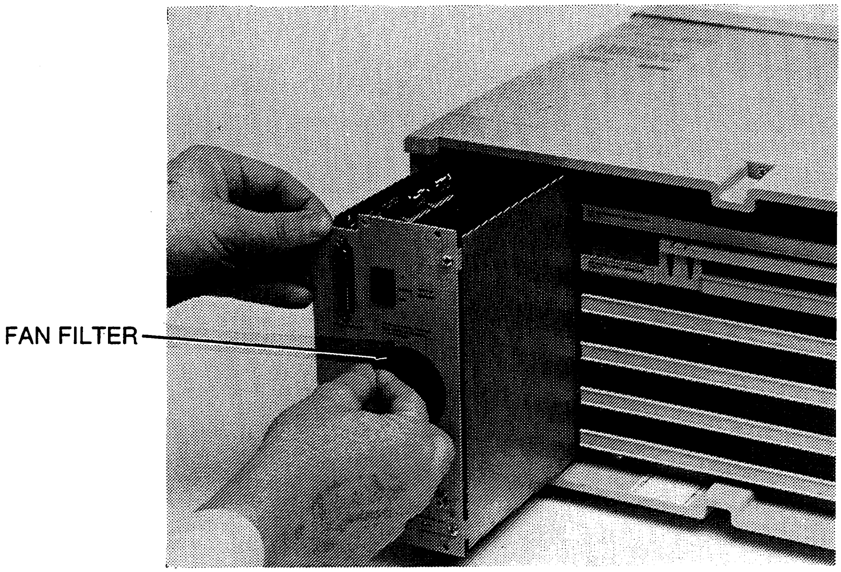


Figure 7-2. Removing the Computer Interface Module

5. Refer to Figure 7-3 to locate the Line Power Voltage Pins. To select 180-264V operation, connect the wire to the pin marked 220V. For 90-132 operation, place the wire on the pin labeled 110V.

NOTE

It is not necessary to change the power input fuse when changing the power supply operating voltage.

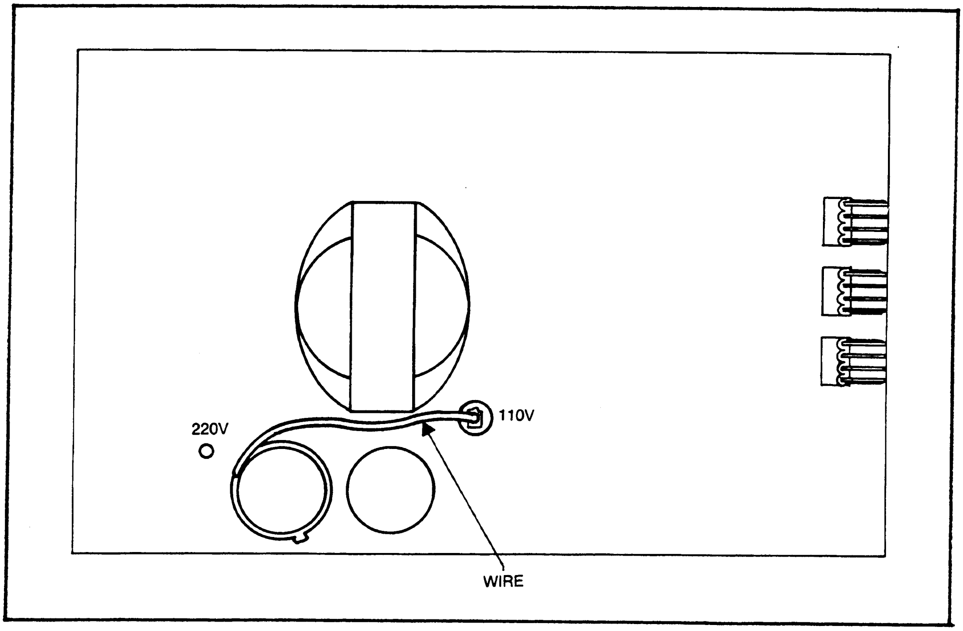


Figure 7-3. Line Power Voltage Pins

6. While the unit is open, locate the Line Frequency Selection Switch (S3-8) and ensure that it is in the proper position to match the local line frequency. If it is not, change it's position before continuing. For 50 Hz operation, place S3-8 in the 0 position (toward card edge). For 60 Hz Operation, place it in the 1 position.

7. Properly mark the power setting on the outside of the unit after changing the line power voltage.

8. Slide the Computer Interface Module back into the Front End, and reinstall the Phillips-head screws.

FUSE REPLACEMENT

The fuse is located on a clip-type holder on the power supply assembly, in the corner to the right of the 110V voltage selection pin. When replacing the fuse use the same value (2.0A @ 250V).

To check or replace the fuse, first turn the power OFF. Then remove the Computer Interface Module as described previously in this section. Use a slotted screwdriver or adjustment tool to remove the fuse. After checking or replacing the fuse, reassemble the unit and test its operation.

FAN FILTER CLEANING

Clean the fan filter any time it is visibly contaminated. If the unit is operated too long with a dirty air filter, heat buildup inside could damage sensitive electronic components.

To clean the filter, pinch the center and pull directly out. Clean with warm soapy water and rinse thoroughly before replacing.

GENERAL CLEANING

CAUTION

Disconnect the battery before cleaning the Front End. Also, do not use aromatic hydrocarbons (such as naphthalene) or chlorinated solvents (such as carbon tetrachloride) for cleaning. They may damage plastic materials used in the instrument.

o Handle the assemblies by their edges rather than by their connector pins. Skin oils can contaminate the assembly and degrade measurement accuracy.

- Improper handling can also cause instantaneous or delayed electrostatic discharge damage. The yellow "Static Awareness" sheet in the front of the Service Manual explains some of the hazards of static electricity to sensitive components.

- Do not use a static-inducing vacuum brush to clean assemblies. Possible electrostatic discharge can damage sensitive components.

Clean the Front End periodically to remove dust, grease, and other contamination. The Front End case may be cleaned using a soft cloth dampened with a mild solution of detergent and water. Dry the case after cleaning.

If visual inspection reveals significant contamination on printed circuit board surfaces, the assembly can be cleaned with low pressure (<20 psi) air. If air is not available, clean the assembly with commercial water-based cleaning equipment.

If commercial water-based cleaning equipment is not available, clean the assembly by holding it under warm, running water. Observe the following precautions when using water-based cleaning equipment:

1. Read all precautions listed above under General Cleaning.

2. Remove all assembly shielding covers and separate any relay piggy-back assemblies.

3. In areas with exceptionally hard water, use either deionized or distilled water for a final wash to remove ions left by the hard water wash.

4. Dry all assemblies thoroughly. Use a low-temperature drying chamber or an infrared drying rack with a temperature range between 100 and 120 ^{o}F (38 to 46 ^{o}C) if available.

5. If a drying chamber or infrared drying rack is not available, air-dry the assembly at room temperature for a minimum of 48 hours before reassembling.

6. Use a mixture of 70% isopropyl alcohol and 30% water and a lint-free cloth to clean edge-connector contacts. Never use an eraser to clean connector contacts; it might generate static or abrade the gold plating on the contacts.

SERVICE INFORMATION

For additional maintenance and servicing information, refer to the Helios Front End Service Manual.

Section 8
Error Messages

CONTENTS

INTRODUCTION

If the command line you have sent cannot be carried out, the Front End returns error information. Usually, this means that some part of the command line must be changed to match the format expected by the Front End. It may also mean that command line instructions conflict with the Front End's hardware configuration, or that an illegal value has been entered. A complete listing of error codes and explanations is presented at the end of this section.

If Terminal Mode is activated, a brief message describing the error condition is returned. For example, if a channel number not falling between 0 and 999 is specified, this is returned:

```
?Not a valid channel
```

This information is returned as a numerical error code when Computer Mode is in use. The same error would then be:

```
?02
```

If an option assembly is not installed, or is incorrectly installed or wired, SEND CHAN(channels) evokes the following response:

```
9.99999E+37
```

When this general error indication occurs, gather more error information by sending: LIST ERROR.

WHAT TO DO ABOUT AN ERROR

Most errors result from a command line that is either syntactically wrong or attempts to do something that is just not possible. You may need additional description of the error code (Computer Mode) or error message (Terminal Mode). Full descriptions are provided at the end of this section.

Suppose the problem cannot be resolved by changing the command line. This situation suggests a hardware problem: a measurement connection to the Front End may be faulty, a measurement sensor may be defective or trying to sense something out of its range, or a problem may exist with the Front End and its connections to the Terminal/Computer.

At this point, it's generally best to gather all available error history. Do this in Terminal Mode: the returned information will be in descriptive English. Use the LIST ERROR command to return errors that occurred during execution of a SEND command.

Use the TEST command to check out any suspected channel or group of channels.

ERROR INFORMATION

The following list relates the two-digit error code (as seen in Computer Mode) to an error message (as seen in Terminal Mode). Further description is then provided for each type of error.

01 Illegal value

The output value set for an analog output channel is out of range. Select a value permissible in the selected analog output mode:

BIPOLV	-5 to +5
UNIPOLV	0 to +10
DCOUT	0.004 to 0.020
PVOUT	0 to 100

02 Not a valid channel

A channel number other than 0 through 999 was specified.

03 Incompatible channel

The channel is not capable of doing what you command. For example, an output value cannot be assigned to an analog input channel, a thermocouple input cannot be defined on a Counter/Totalizer channel, and a -165 Fast A/D channel cannot be used for alternating voltage (AVIN) inputs. Check that the command line does not conflict with either other programming instructions or with hardware settings.

04 Channel not available

A definition or operation was attempted on a channel for which no hardware is installed in the system. Check that related hardware is correctly installed.

05 No device reply

An error occurred while trying to communicate with an I/O device (or no I/O device is installed).

06 Illegal bridge type

In a strain definition statement, a bridge type number other than 0 through 5 was specified.

07 System error

The system could not handle an unexpected error condition.

08 Bad frequency

A LINEFR value assignment other than 50, 60, or 400 was attempted.

09 Device error

A channel has failed the self test routine initiated with the TEST command. This is usually caused by a hardware problem.

10 Channel not defined.

A reading from an undefined channel was attempted. Define that channel, or read a properly defined channel.

11 Illegal channel function

An illegal table number was specified. A maximum of 100 tables (0-99) can be established.

12 **Bad table**

An illegal entry was made during a table definition. This error will occur if only one entry pair is specified, or if the required format for entering pairs is not followed.

13 **No channel function**

An attempt was made to use a channel function that was not defined.

14 **Not enough memory**

The amount of memory for channel or table definitions is insufficient. Use SEND MEMSIZ to verify memory remaining.

15 **Out of range**

The measured input exceeds the programmed range. If possible, specify a higher range in the channel definitions.

16 **Open TC**

A temperature measurement was attempted with an open or improperly connected thermocouple.

17 **Illegal BCD**

A BCD digit greater than 9 was present on one or more digit fields of a digital input.

18 **No convergence**

The user-defined RTD constants do not allow for resolution of the temperature calculation.

19 **Over temperature**

The temperature is outside the specified range for the thermocouple or RTD being used.

20 **ROM failed**

A bad ROM or faulty ROM control circuit was detected during the system self test. The system will continue operating, but results may be unpredictable.

21 **RAM failed**

During the system self test, a bad RAM or faulty RAM control circuit was detected. The system will continue operating, but results may be unpredictable.

22 **SL UART failed**

A problem with the serial link UART was detected during the system self test. The system will continue operating, but results may be unpredictable.

23 **Bad parameter**

An unexpected or illegal parameter was specified. Check command reference information in Section 5.

24 **Too many parameters**

More than 20 items were specified in a channel list.

25 **Bad date**

An assignment to DATE$ did not meet the syntax specification.

26 **Bad time**

An assignment to TIME$ or TIME did not meet the syntax specification.

27 **No such command**

An unknown command was detected.

28 **Bad syntax**

The command line does not meet syntax requirements. Check command reference information in Section 5.

29 **Bad channel range**

A channel range was incorrectly specified. For example, this error occurs if CHAN(20..10) is sent.

30 **Table not defined**

A reference to an undefined interpolation table was attempted.

31 **Illegal poly definition**

A polynomial channel function has not been properly defined.

32 **Bad scan task**

An illegal scan task number has been specified.

33 **Bad buffer**

An illegal buffer number has been specified.

34 **Scan task not defined**

A reference has been made to an undefined scan task.

35 **Buffer not defined**

A reference has been made to an undefined buffer.

36 **Bad alarm channel**

An illegal alarm channel has been specified.

37 **OVERLAY failed**

A problem with the overlay manager was detected during the system selftest. The Helios system will continue operating; however, results might be unpredictable.

38 **No previous record available**

This error occurs when a SHOW AGAIN ABUF or SHOW AGAIN SBUF command has been used, but the previous record has already been overwritten and is no longer available.

39 **Illegal channel label**

The channel label string defined in a LABEL CHAN statement exceeds 12 characters, contains one or more non-printable characters, or is not enclosed in quotation marks.

40 **Illegal channel units in label**

The UNITS string defined in a LABEL CHAN statement exceeds six characters.

41 **Illegal channel format in label**

Legal channel formats, specified in a LABEL CHAN statement, are: F0, F1, F2, F3, F4, F5, E0, E1, E2, E3, E4, and E5.

42 **Illegal channel ctrl sequence in label**

The CTRL string defined in a LABEL CHAN statement exceeds eight decimal codes, or a semi-colon (;) has been omitted between decimal codes.

43 **Illegal number in label**

This error pertains to the use of the NUMBER parameter with the LABEL CHAN command. Make sure that the number of pound signs (#) used in the label string is compatible with the maximum number of digits required by NUMBER.

44 **Illegal scan header**

The HEADER defined in a START SCAN statement exceeds 128 characters or is incorrectly stated. A header may consist of any combination of strings (enclosed by quotation marks) and ASCII decimal codes. Each character within the quotation marks and each decimal code (one-, two-, or three-digit) counts in the 128-character total. Further, decimal codes and quoted strings must be separated with semicolons (;).

45 **Illegal scan footer**

The FOOTER defined in a START SCAN statement exceeds 128 characters or is incorrectly stated. Error code 44 syntax rules also apply to this error code.

46 **Illegal scan period count**

The PERIOD parameter established in a START SCAN statement is less than 2. For meaningful computation of MAX, MIN, AVG, or TOT functions, two or more scan cycles (periods) must be specified.

47 **SQR less than zero**

The square root function (SQR) has been applied to an input value less than 0.

48 **Channel list requires scan buffer**

Channel math cannot be performed if the scan task specifies OUTPUT=HOST or OUTPUT=PRINTER; channel math requires the use of a scan buffer. If no scan buffer is defined, this error results when a scan task is started and any of the channels in the channel list specifies channel math (MAX, MIN, AVG, or TOTAL).

49 **Interval time specified between scans too small**

The SCANINTERVAL parameter of the DEF BSCAN command has been set too small for the Fast A/D to scan when operating in Burst Scan Mode. The host can send the LIST BSCAN command to get the minimum interval time.

50 **Scanning in progress, cannot change burst scan definition**

The DEF BSCAN command has been received while the Fast A/D is in the process of burst scanning. The Fast A/D must be stopped for this command to change the burst scan definition.

51 **Burst scan number does not exist**

The scan number specified in the SEND BSCAN command does not represent a valid position in the burst scan buffer. Use the SEND SCANUM command to determine the valid scan number range.

52 **Burst scanning is in progress, cannot change which channels scanned**

A DEF CHAN command has been sent that defines different channels than those that are currently being measured (while a Fast A/D is in the process of doing burst scanning) or a RESET CHAN command has been sent for a channel that is burst scanning.

53 **Burst scanning is in progress, cannot specify DIFF or SINGLE**

A DEF CHAN command has been issued that attempted to change the type of channel measurement (differential or single-ended) for a channel on a Fast A/D. The measurement type can only be changed when the a/d is not burst scanning.

54 **Can only specify DIFF or SINGLE when the channel is on a Fast A/D**

You have probably tried to define DIFF or SINGLE for a channel that is not associated with a -165 Fast A/D Converter.

55 **First and twenty-first channel reserved for external trigger I/O**

The command cannot be used when the first and twenty-first channels are configured by hardware jumper on the Fast A/D Converter for use as external trigger input and output, respectively. With this configuration, these channels are not available for measurements.

56 **The filter count is too large**

A SEND BSCAN command was sent with a FILTERCNT parameter specified larger than 15. The SEND BSCAN command has no effect.

57 **The number of scans to do after a trigger event is too large**

A SEND BSCAN command was sent with a TRIGPOS parameter specified larger than 1,048,575. The SEND BSCAN command has no effect.

58 **The burst scan interval is too large**

A SEND BSCAN command was sent with a SCANINTERVAL parameter specified larger than 32,767. The SEND BSCAN command has no effect.

59 **No channels have been defined on this A/D, no burst scanning started**

A START BSCAN command was sent to start burst scanning an a Fast A/D, but no channels were defined on that A/D. The START BSCAN command has no effect.

60 **A high limit cannot be higher than a high-high limit**

A DEF CHAN command was sent with the HI limit specified greater than the HIHI limit. The DEF CHAN command has no effect.

61 **A low limit cannot be lower than a low-low limit**

A DEF CHAN command was sent with the LO limit specified less than the LOLO limit. The DEF CHAN command has no effect.

62 **Invalid scan number range**

A scan number range in the SEND BSCAN command was specified with the lower end greater than the higher end.

63 **Fast A/D is not jumpered for external trigger I/O**

A DEF BSCAN command was sent with an external trigger type (XTRIGTYPE) specified, but the Fast A/D is not jumpered correctly for external trigger I/O. The DEF BSCAN command has no effect.

64 **Fast A/D number does not exist**

A command was sent with a Fast A/D number that does not exist in the current Helios-Plus Data Acquisition System. The command has no effect.

65 **Uncalibrated A/D requires all 3 reference values**

An attempt was made to calibrate a previously uncalibrated A/D without providing all three reference values. Until an A/D has been calibrated at least once, all three reference values are unknown and must be specified explicitly.

66 **Can only specify a trigger when the channels are on a Fast A/D**

You have tried to specify a trigger for a channel that is not associated with a -165 Fast A/D Converter.

67 **Cannot define a mix of channels from a Fast A/D and a high accuracy A/D**

In a single DEF CHAN command, the channels must all reside on the same type of I/O option. For example, you cannot mix Fast A/D Converter channels with High Performance A/D Converter channels in the same DEF CHAN command. More than one DEF CHAN command is required if channels on both types of A/Ds are to be defined.

68 **A calibration reference point value is out of range**

A calibration reference point value specified in the CALFAD command is not within the range that is permitted. See the table in the CALFAD command description (2287A Service Manual) for the range of values that are allowed for each of the three reference points.

69 **Can't trigger using POLY() unless a typical X value has been specified**

For reverse computing the polynomial

$$y = aX^2 + bX + c$$

an "X" representing a typical sensor value in the range of interest for the polynomial is needed. Use a value of "X" that is near the center of the expected range of sensor values. Without "X", the reverse solution cannot determine which of two possible sensor values to use as the trigger (Y) value.

70 **Trigger is outside of measurement range**

A trigger value specified by the host, when converted from engineering units to an equivalent direct voltage for measurement, results in a value outside of the voltage range used by the Fast A/D Converter. This trigger value is rejected, and the trigger for this channel is cleared.

71 chan(XXX) - Device self test error N

The -165 Fast A/D Converter containing channel XXX has detected the self-test error(s) indicated by failure code N.

NOTE

Once error 71 has been detected for a Fast A/D Converter, Front End system power must be cycled OFF-ON to reestablish proper communication between the Front End and that Fast A/D Converter.

The following failure code values are used:

N	MEANING
1	ROM checksum error.
2	RAM error.
4	Calibration memory checksum error.
8	Analog control latch error.
16	Self-calibration error.
32	Open thermocouple detection error.
64	<not used>
128	Other self-test error

N may also represent a sum of the individual errors listed above. For example, 33 would signify error 1 (ROM checksum error) and error 32 (Open thermocouple detection error). See the Helios Plus Service Manual for more information.

NOTE

The additional information identifying the channel number (XXX) and failure code (N) are only available when operating in terminal mode. When in computer mode, only the error value ?71 is returned.

72 **Host stalled output for too long, output dropped**

The host stalled output from Helios Plus continually for more than 10 seconds. To allow the continuation of other operations (such as scan tasks and alarm processing), output intended for the host has been dropped.

73 **Printer stalled output for too long, output dropped**

The printer stalled output from Helios Plus continually for more than 10 seconds. To allow the continuation of other operations (such as scan tasks and alarm processing), output intended for the printer has been dropped.

Section 9
Appendices

CONTENTS

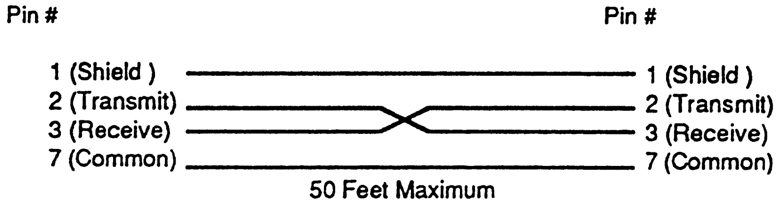
Appendix 9a
Configuring Your Computer

CONTENTS

*IBM PC is a trademark of IBM Corporation.
*TRS-80 is a trademark of The Tandy Corporation.

Cable Connections

1722A	Computer Front End
DB-25S (sockets)	DB-25S (sockets)



Note: Use Fluke Y1702, Y1703 Or Y1705 Null Modem Cables

1. Verify that the Front End is configured correctly to factory settings (see Section 3).

2. Connect the Front End to the KB1: port of the Fluke 1722A or 1752A using a null modem cable, Fluke Y1702 or equivalent, as illustrated above.

3. Power up the Front End.

4. Power up the 1722A or 1752A. Insert the System Disk and enter FDOS (the FDOS> prompt displays).

5. Enter the SET Utility Program by typing:

```
SET <RETURN>
```

6. When the SET> prompt appears, type in the following:

```
KB1:,BR 9600,DB 8,SB 1,EOL 10, EOF 26, SI E,SO E
 TO 10 <RETURN>
```

7. Verify the configuration of KB1: by typing:

```
LI <RETURN>.
```

The display should show the following list:

```
DEVICE               KB1:
BAUD RATE            9600
DATA BITS            8
PARITY               NONE
STOP BITS            1
END OF LINE          10
END OF FILE          26
STALL INPUT          ENABLED
STALL OUTPUT         ENABLED
TIME OUT             10
```

8. To exit SET, type EX <RETURN>. When the FDOS> prompt returns, enter BASIC by typing BASIC <RETURN>.

9. When the BASIC prompt **Ready** appears, enter the interactive editor by typing EDIT <RETURN>. Type in the following program:

```
10  PRINT CHR$(27)+"[2J";\C1OSE ALL
20  OPEN "KB1:" AS NEW FILE 1
30  OPEN "KB1:" AS OLD FILE 2
40  PRINT #1, "MODE=COMP"
50  INPUT #2,A$
60  IF A$<>"!" THEN GOTO 50
70  PRINT #1, "SEND TIME$"
80  INPUT #2,A$
90  PRINT CPOS(7,30),"The time is ";A$
100 END
```

10. Type <CTRL/C> to exit the BASIC editor. To save the program on disk, type:

```
SAVE 'HCLI.TST' <RETURN>
```

11. To execute the program, type RUN <RETURN>. The time will display on the center of the screen.

Here are line-by-line comments of this program:

10 Clears the screen and closes all files.

20 Opens RS-232 port for output

30 Opens port for input

40 Puts the Front End in computer mode.

50 Flushes the buffer until...

60 ...it contains "!"

70 Commands the Front End to send the time.

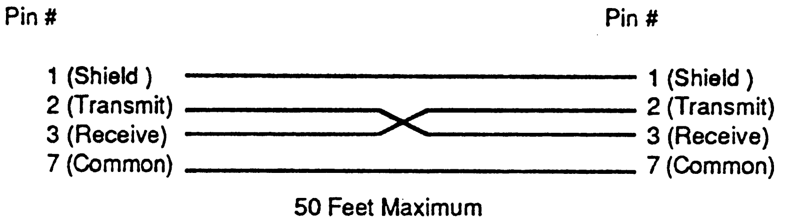
80 Inputs TIME$ to the buffer.

90 Prints the time information in the middle of the display.

IBM PC and PC-AT Personal Computers

Cable Connections

IBM PC Or XT (COM)	Computer Front End
DB-25S (sockets)	DB-25S (sockets)



Note: Use Fluke Y1702, Y1703 Or Y1705 Null Modem Cables

1. Verify that the Front End is configured correctly to factory settings (see Section 3).

2. If you have an IBM PC, an optional RS-232 serial interface must be installed and selected as COM1. The PC-XT comes standard with COM1.

3. Connect the Front End to the personal computer with a null modem cable, Fluke Y1702 or equivalent, as illustrated above.

4. Power up the Front End.

5. Insert a DOS disk into drive A and power up the personal computer. When you see the A> prompt, type BASIC and enter the following program. The program establishes communication with the Front End, retrieves the variable VERSION$ (the software version) from the Front End, and displays it on the IBM PC screen.

```
10  CLEAR
20  CLS
30  CLOSE
40  OPEN "COM1:9600,N,8,1,CS,DS,CD" AS #1
50  PRINT #1, "MODE=COMP"
60  INPUT #1, A$
70  IF A$<>"!" THEN GOTO 60
80  PRINT #1, "SEND VERSION$"
90  INPUT #1, A$
100 PRINT "The firmware is ";A$
101 END
```

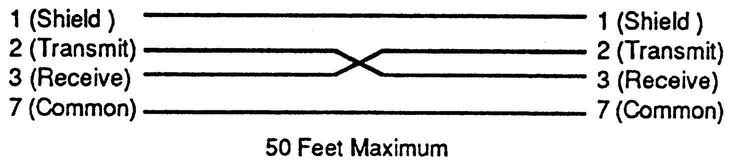
Here are line-by-line comments of this program:

10 Initializes memory.
20 Clears the screen
30 Closes all files
40 Opens RS-232 port for output
(9600 baud, 8 data bits, no parity, etc.)
50 Puts the Front End in computer mode.
60 Flushes the buffer until...
70 ...it contains "!"
80 Commands Front End to send the firmware version.
90 Inputs VERSION$ to the buffer.
100 Prints the version.

TRS-80 Model 100 Lap-Top Computer

Cable Connections

TRS-80-100	Computer Front End
DB-25P (pins)	DB-25S (sockets)
Pin #	Pin #



Note : There is no standard Fluke cable for this computer.

Communicating in the Terminal Mode

1. Verify that the Front End is configured correctly to factory settings (see Section 3).

2. Connect the Front End to the TRS-80-100 with a cable as illustrated.

3. Power up the Front End.

4. Power up the TRS-80-100 and press <F8> as required to arrive at the Main Menu. A BASIC prompt will appear in the upper left corner of the screen.

5. Use the cursor keys to move the cursor to **TELCOM** and press <enter>.

6. Press <F3> and type 88N1E <enter> to set the TRS-80-100 port characteristics. Press <F4> to enter the terminal mode.

7. Type mode=term <enter> on the TRS-80-100 keyboard. The Front End should respond by displaying the prompt **HCLI>** on the TRS-80-100 screen.

The HCLI> prompt indicates that the TRS-80-100 and the Front End are communicating bi-directionally. The Front End commands described in Section 5 of this manual can now be typed in at the keyboard.

Communicating from BASIC

The following program establishes communication with the Front End, retrieves the variable TIME$ from the Front End, and displays it on the Model 100 screen.

```
10  CLS:CLEAR
20  OPEN "COM:88N1E" FOR OUTPUT AS 1
30  OPEN "COM:88N1E" FOR INPUT AS 2
40  PRINT #1, "MODE=COMP"
50  INPUT #2, A$
60  IF A$<>"!" THEN GOTO 50
70  PRINT #1, "SEND TIME$"
80  INPUT #2, A$
90  PRINT @ 125, "The time is ";A$
100 END
```

Here are line-by-line comments of this program:

10 Clears display, initializes memory.
20 Opens the RS-232 port for output (9600 baud, 8 data bits, no parity, etc.)
30 Opens the port for input - same parameters as output.
40 Puts the Front End in computer mode.
50 Flushes the buffer until...
60 ...it contains "!"
70 Commands the Front End to send the time.
80 Inputs TIME$ to the buffer.
90 Positions the cursor and prints the time information.

Appendix 9b
Serial Data Reference

CONTENTS

WHAT IS SERIAL DATA?

Bit serial data is digital information represented as groups of bits (high and low states, representing ones and zeros), sent over one line, one bit at a time.

TIMING AND SPEED

Serial data can be transmitted in two different timing formats, synchronous and asynchronous. Synchronous, or message framed data, uses start and stop characters to mark the beginning and end of each message block. Individual characters in the message are sent one after the other, without any special bits separating them. Asynchronous data is also called character framed data, because it has start and stop bits separating each character, so each character is transmitted as an independent unit. The Computer Front End serial interface operates only using asynchronous timing.

The speed of serial data is expressed in bits per second, or baud. Although baud means the rate of actual state changes per second, it is often used interchangeably with bit rate. The Computer Front End serial interface can send and receive data at all the following bit rates:

110
300
600
1200
2400
4800
9600 (Factory Setting)
19200

CHARACTER FORMAT

Information is sent to and from the Computer Front End serial interface in the form of ASCII (American Standard Code for Information Interchange) coded characters. Figure 9b-1 shows the format of a single asynchronous serial character. Each character begins with one start bit (low), then seven or eight data bits (the ASCII character), then an optional parity bit, and finally one or two stop bits (high).

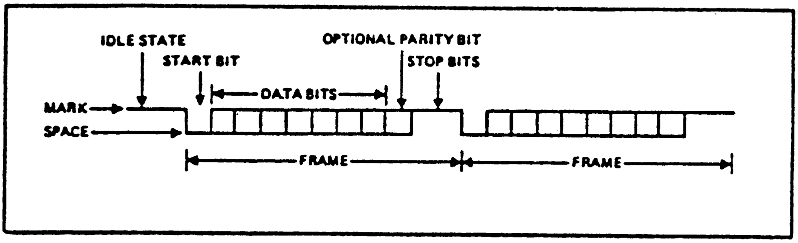


Figure 9b-1. An Asynchronous Serial Character

Reception of the start bit tells the receiving station that the next seven (or eight, depending on a switch setting) bits represent an ASCII coded character (transmitted least significant bit first). The start bit is always a single low bit (zero).

The stop bit tells the receiving station that it has reached the end of the character. The stop bit is high, and so is the idle state of the line. A stop bit may be immediately followed by the start bit for the next character, or may just return the line to its idle state (if it belongs to the last character to be transmitted). You can select a stop bit of one or two bits in length, to match the Serial Interface to the requirements of various host computers and interfaces. Instructions for setting the CHARACTER FORMAT switch (S2) are given in Section 3, Installation and Setup.

PARITY

The parity bit follows the seven or eight data bits of the ASCII-coded character. The parity bit is an error detection device, and can be configured in three ways:

- Parity Off

  When parity is turned off, the interface does not transmit a parity bit, and does not expect to receive one. The stop bit immediately follows the last data bit of the character.

- Even Parity

  Even parity means that the parity bit, when appended to the binary digits of the ASCII-coded character, yields an even number of one-bits.

- Odd Parity

  Odd parity means that the parity bit, when appended to the binary digits of the ASCII-coded character, yields an odd number. For example, the ASCII code for the letter "a" is:

  ```
  0010 0001
  ```

  If odd parity were being used, the code transmitted would be:

  ```
  0010 0001 1
  ```

Instructions for setting the parity switch segments of the CHARACTER FORMAT switch (S2) are given in Section 3, Installation and Setup.

The information in this section is an introduction to serial data communication using the Computer Front End serial interface. Detailed instructions regarding system circuit connections, interface configuration, and operation are found in Sections 3 and Appendix 9a, Configuring Your Computer.

CIRCUIT CONNECTION

The first step in data transfer between two or more computers involves the electrical circuit connection. The connection can be either direct-connect (over a dedicated hard-wired line) or switched service (over a telephone line). The network can be two-point, with just the host computer and the Computer Front End or multipoint (also called multi-drop), with one host computer and several Front Ends.

The Computer Front End can be remotely accessed by a host computer in the following types of networks:

- RS-232-C direct-connect, two-point
- RS-232-C with modems over a dedicated line, two-point
- RS-232-C with auto-answer modems (switched service)
- RS-422 direct-connect, two-point
- RS-422 direct-connect, multipoint (up to ten Front Ends)

The RS-232-C interface can be used in either switched service or direct-connect networks. However, the RS-422 interfaces can only be direct-connected. The same interface connector is used for both RS-232-C and RS-422 communications.

Using RS-232-C, a direct-connect circuit is made with a Null-Modem Cable. In systems using RS-422, a direct-connect circuit is made with twisted-pair shielded cable.

Multipoint switched-service networks with auto-answer modems (RS-232-C only), are really switched two-point networks, as each modem remains disconnected from the circuit until its own telephone number is dialed. Connection occurs in a switched-service system (RS-232-C interface) with auto-answer modems when the line attached to the modem rings. This happens as follows:

1. The modem at the host end dials the number for a particular Front End to establish a link with that Computer Front End.

2. The line at the Front End rings, and the interface detects signal CE (Ring Indicator) high.

3. The Serial interface responds by asserting CD (Data Terminal Ready) high. This tells the auto-answer modem to answer the line, establishing a connection.

This is an overview of the RS-232-C switched service connection procedure. In RS-422 multipoint systems without modems, addresses are set on the ADDRESS switch (S3) on the Computer Interface assembly. Switch settings and circuit connections are covered in Section 3, Installation and Setup.

MULTIPOINT

In a multipoint direct-connect system (RS-422), each Front End in the system has a unique address. Up to ten Front Ends may be connected in such a system. Each Front End in the system is assigned an address from zero to nine, as set on the ADDRESS switch (S3) on the Computer Interface Module. After one Computer Front End is addressed, the other Front Ends remain idle until there is another address sequence.

In a two-point network, there is no establishment procedure, since there is only one Computer Front End. In a multipoint network, however, a logical connection must be established between the host and one Front End.

The host establishes an RS-422 communication link between itself and a particular Computer Front End in a multipoint network as follows:

1. The host sends an address character followed by the Enquiry character, ENQ. The address character is the mathematical sum of the address (0 - 9) and ASCII-coded character "0" (ASCII decimal 48).

2. When a Front End detects its address sequence, it responds by echoing its address character followed by the Acknowledge character, ACK, causing establishment to occur.

NOTE

When using a multipoint system, make sure that only one Front End is addressed at a time. Perform the termination sequence prior to addressing any Front End.

An addressed Front End can be unaddressed by sending a <DLE> <EOT> character sequence or an <EOT> character. A single <EOT> disconnects the Front End from the host, but any pending output is retained until reconnected.

MESSAGE TRANSFER

Introduction

Once a connection has been established, messages can be exchanged between the Computer Front End and the host, using various data channel protocols and message transfer protocols.

Data Channel Protocols

The RS-422 interface has dedicated lines for transmitting and receiving messages. The data transfer lines for the RS-232-C interface, however, can be configured in different ways: direct-connect or with modems, and full duplex or half duplex. Half-duplex communication is not implemented in the Computer Front End.

In full-duplex mode, each modem transmits data on a carrier wave of a different frequency, allowing two data channels to exist on the same line. Figure 9b-2 shows how the available bandwidth of a voice-grade telephone line is occupied by the two data channels in full duplex.

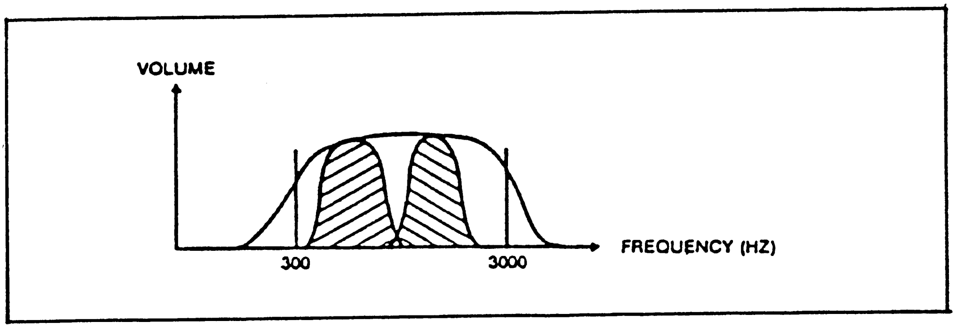


Figure 9b-2. Full-Duplex Data Channels

Message Transfer Protocols

This describes, starting with the simplest communication method, each message transfer protocol used by the Computer Front End. This information is included to introduce the user who is new to serial communication to the available alternatives in message transfer protocols. MESSAGE TRANSFER switch (S2) settings are given in Section 3, Installation and Setup.

Table 9b-1 shows the ASCII characters mentioned in this manual, with their corresponding decimal values. For a complete table of ASCII characters, see Appendix 9c.

Table 9b-1. ASCII Mnemonics and Decimal Equivalents

ASCII Mnemonic	Decimal Value
ACK	6
DLE	16
ENQ	5
EOT	4
XOFF (CTRL/S)	19
XON (CTRL/Q)	17
CTRL/C	3
DEL	127

Simple Terminal 1

During communication between two devices, one of them, device A, transmits characters to device B, and device B transmits characters to device A. This is called simple terminal protocol. With simple terminal protocol there is no data flow control mechanism and no mechanism for the recovery of character errors detected by the parity bit. This communication method is selected when the XON-XOFF switch (S3-7) on the Computer Interface Module is set to position 0 (DISABLED). This communication method is called ST1, for Simple Terminal 1.

Simple Terminal 2

At low transmission speeds, Simple Terminal 1 is adequate. However, if device B is unable to receive and process characters as fast as A can send them, information is lost. Device B avoids this problem by sending a flow control character to tell device A to stop transmitting until it catches up.

In Simple Terminal 2 message transfer protocol, special characters (XON and XOFF) are transmitted by the receiver to control the flow of data. When the host computer is receiving data using Simple Terminal 2 protocol, and its buffer becomes almost full, it sends the XOFF (transmit off) character to the Front End. The reception of XOFF causes the Front End to halt. This is done to avoid overwriting, and thereby losing, characters. When the computer has processed incoming characters until its buffer is nearly empty, it tells the transmitter to resume the flow of data by sending the XON (transmit on) character. This protocol is selected when S3-7 on the Computer Interface Module is set to position 1, ENABLE. When XON-XOFF flow control is enabled, the protocol is called ST2, for Simple Terminal 2.

Appendix 9c
ASCII Character Set

ASCII CHAR.	DECIMAL	OCTAL	HEX	BINARY	DEV. NO.
				7654 3210	
NUL	0	000	00	0000 0000	
SQH	1	001	01	0000 0001	
STX	2	002	02	0000 0010	
ETX	3	003	03	0000 0011	
EOT	4	004	04	0000 0100	
ENQ	5	005	05	0000 0101	
ACK	6	006	06	0000 0110	
BELL	7	007	07	0000 0111	
BS	8	010	08	0000 1000	
HT	9	011	09	0000 1001	
LF	10	012	0A	0000 1010	
VT	11	013	0B	0000 1011	
FF	12	014	0C	0000 1100	
CR	13	015	0D	0000 1101	
SO	14	016	0E	0000 1110	
SI	15	017	0F	0000 1111	
DLE	16	020	10	0001 0000	
DC1	17	021	11	0001 0001	
DC2	18	022	12	0001 0010	
DC3	19	023	13	0001 0011	
DC4	20	024	14	0001 0100	
NAK	21	025	15	0001 0101	
SYN	22	026	16	0001 0110	
ETB	23	027	17	0001 0111	
CAN	24	030	18	0001 1000	
EM	25	031	19	0001 1001	
SUB	26	032	1A	0001 1010	
ESC	27	033	1B	0001 1011	
FS	28	034	1C	0001 1100	
GS	29	035	1D	0001 1101	
RS	30	036	1E	0001 1110	
US	31	037	1F	0001 1111	

ASCII CHAR.	DECIMAL	OCTAL	HEX	BINARY 7654 3210	DEV. NO.
SPACE	32	040	20	0010 0000	0
!	33	041	21	0010 0001	1
"	34	042	22	0010 0010	2
#	35	043	23	0010 0011	3
$	36	044	24	0010 0100	4
%	37	045	25	0010 0101	5
&	38	046	26	0010 0110	6
'	39	047	27	0010 0111	7
(	40	050	28	0010 1000	8
)	41	051	29	0010 1001	9
*	42	052	2A	0010 1010	10
+	43	053	2B	0010 1011	11
,	44	054	2C	0010 1100	12
-	45	055	2D	0010 1101	13
.	46	056	2E	0010 1110	14
/	47	057	2F	0010 1111	15
0	48	060	30	0011 0000	16
1	49	061	31	0011 0001	17
2	50	062	32	0011 0010	18
3	51	063	33	0011 0011	19
4	52	064	34	0011 0100	20
5	53	065	35	0011 0101	21
6	54	066	36	0011 0110	22
7	55	067	37	0011 0111	23
8	56	070	38	0011 1000	24
9	57	071	39	0011 1001	25
:	58	072	3A	0011 1010	26
;	59	073	3B	0011 1011	27
<	60	074	3C	0011 1100	28
=	61	075	3D	0011 1101	29
>	62	076	3E	0011 1110	30
?	63	077	3F	0011 1111	

ASCII CHAR.	DECIMAL	OCTAL	HEX	BINARY 7654 3210	DEV. NO.
@	64	100	40	0100 0000	0
A	65	101	41	0100 0001	1
B	66	102	42	0100 0010	2
C	67	103	43	0100 0011	3
D	68	104	44	0100 0100	4
E	69	105	45	0100 0101	5
F	70	106	46	0100 0110	6
G	71	107	47	0100 0111	7
H	72	110	48	0100 1000	8
I	73	111	49	0100 1001	9
J	74	112	4A	0100 1010	10
K	75	113	4B	0100 1011	11
L	76	114	4C	0100 1100	12
M	77	115	4D	0100 1101	13
N	78	116	4E	0100 1110	14
O	79	117	4F	0100 1111	15
P	80	120	50	0101 0000	16
Q	81	121	51	0101 0001	17
R	82	122	52	0101 0010	18
S	83	123	53	0101 0011	19
T	84	124	54	0101 0100	20
U	85	125	55	0101 0101	21
V	86	126	56	0101 0110	22
W	87	127	57	0101 0111	23
X	88	130	58	0101 1000	24
Y	89	131	59	0101 1001	25
Z	90	132	5A	0101 1010	26
[	91	133	5B	0101 1011	27
\	92	134	5C	0101 1100	28
]	93	135	5D	0101 1101	29
^	94	136	5E	0101 1110	30
_	95	137	5F	0101 1111	

ASCII CHAR.	DECIMAL	OCTAL	HEX	BINARY 7654 3210	DEV. NO.
	96	140	60	0110 0000	0
a	97	141	61	0110 0001	1
b	98	142	62	0110 0010	2
c	99	143	63	0110 0011	3
d	100	144	64	0110 0100	4
e	101	145	65	0110 0101	5
f	102	146	66	0110 0110	6
g	103	147	67	0110 0111	7
h	104	150	68	0110 1000	8
i	105	151	69	0110 1001	9
j	106	152	6A	0110 1010	10
k	107	153	6B	0110 1011	11
l	108	154	6C	0110 1100	12
m	109	155	6D	0110 1101	13
n	110	156	6E	0110 1110	14
o	111	157	6F	0110 1111	15
p	112	160	70	0111 0000	16
q	113	161	71	0111 0001	17
r	114	162	72	0111 0010	18
s	115	163	73	0111 0011	19
t	116	164	74	0111 0100	20
u	117	165	75	0111 0101	21
v	118	166	76	0111 0110	22
w	119	167	77	0111 0111	23
x	120	170	78	0111 1000	24
y	121	171	79	0111 1001	25
z	122	172	7A	0111 1010	26
{	123	173	7B	0111 1011	27
|	124	174	7C	0111 1100	28
}	125	175	7D	0111 1101	29
~	126	176	7E	0111 1110	30
	127	177	7F	0111 1111	

Appendix 9d
Power-On and Reset Routines

	Battery Power Lost	RESET ALL	RESET	Power-Up
Self test, report errors		X	X	
Set-up per I/F switches	X			X
Calendar to 01-Jan-84	X			
Time to 00:00:00 hours	X			
Clear error log	X	X		X
Clear interpolation table	X	X	X	
Reset/initialize options	X	X	X	X
Calibrate A/D Converters (default line freq)	X	X		
Calibrate A/D Converters (line freq)			X	X
Set system variables defaults	X	X		
Build/rebuild system channel list	X	X	X	X
Return prompt to host computer (HCLI or !)		X	X	

Appendix 9e
Service Centers

TECHNICAL SERVICE CENTERS

U.S. SERVICE LOCATIONS

California
Fluke Technical Center
16969 Von Karman Avenue
Suite 100
Irvine, CA 92714
Tel: (714) 863-9031

Fluke Technical Center
46610 Landing Parkway
Fremont, CA 94538
Tel: (415) 651-5112

Colorado
Fluke Technical Center
14180 East Evans Avenue
Aurora, CO 80014
Tel: (303) 695-1171

Florida
Fluke Technical Center
550 S. North Lake Blvd.
Altamonte Springs,
FL 32701-5227
Tel: (407) 331-4881

Illinois
Fluke Technical Center
1150 W. Euclid Ave.
Palatine, IL 60067
Tel: (312) 705-0500

Maryland
Fluke Technical Center
5640 Fishers Lane
Rockville, MD 20852
Tel: (301) 770-1576

New Jersey
Fluke Technical Center
East 66 Midland Avenue
Paramus, NJ 07652-0930
Tel: (201) 599-9500

Texas
Fluke Technical Center
1801 Royal Lane, Suite 307
Dallas, TX 75229
Tel: (214) 869-2848

Washington
Fluke Technical Center
John Fluke Mfg. Co., Inc.
1420 75th St. S.W. M/S 6-30
Everett, WA 98203
Tel: (206) 356-5560

INTERNATIONAL

Argentina
Coasin S.A.
Virrey del Pino 4071 DPTO E-65
1430 CAP FED
Buenos Aires
Tel: 54 1 522-5248

Australia
Philips Customer Support
Scientific and Industrial
23 Lakeside Drive
Tally Ho Technology Park
East Burwood
Victoria 3151

Australia
Philips Customer Support
Scientific & Industrial
25-27 Paul St. North
North Ryde, N.S.W. 2113
Tel: 61 02 888 8222

Austria
Oesterreichische Philips Industrie
Unternehmensbereich Prof. Systeme
Triesterstrasse 66
Postfach 217
A-1101 Wein
Tel: 43 222-60101, x1388

Belgium
Philips & MBLE Associated S.A.
Scientific & Industrial Equip. Div
Service Department
80 Rue des deux Gares B-1070
Brussels
Tel: 32 2 525 6111

Brazil
Hi-Tek Electronica Ltda.
Al. Amazonas 422, Alphaville
CEP 06400 Barueri
Sao Paulo
Tel: 55 011 421-5477

Canada
Fluke Electronics Canada Inc.
400 Britannia Rd. East, Unit #1
Mississauga, Ontario L4Z 1X9
Tel: 416-890-7600

Chile
Intronsa Inc.
Casilla 16158
Santiago 9
Tel: 56 2 232-1886, 232-4308

China
Fluke International Corp.
P.O. Box 9085
Beijing
Tel: 86 01 512-3436

Colombia
Sistemas E Instrumentacion, Ltda.
Carrera 13, No. 37-43, Of. 401
Ap. Aereo 29583
Bogota
Tel: 57 232-4532

Denmark
Philips A/S
Technical Service I & E
Strandlodsveij 1A
PO Box 1919
DK-2300
Copenhagen S
Tel: 45 1 572222

Ecuador
Proteco Coasin Cia., Ltda.
P.O. Box 228-A
Ave. 12 de Octubre
2285 y Orellana
Quito
Tel: 593 2 529684

Egypt
Philips Egypt
10, Abdel Rahman el Rafei st.
el. Mohandessin
P.O. Box 242
Dokki Cairo
Tel: 20-2-490922

England
Philips Scientific
Test & Measuring Division
Colonial Way
Watford
Hertfordshire WD2 4TT
Tel: 44 923-240511

Finland
Oy Philips AB
Central Service
Sinikalliontie 1-3
P.O. Box 11
02631 ESPOO
Tel: 358-0-52572

France
S.A. Philips Industrielle
et Comerciale,
Science et Industry
105 Rue de Paris BP 62
93002 Bobigny, Cedex
Tel: 33-1-4942-8040

Greece
Philips S.A. Hellenique
15, 25th March Street
177 78 Tavros
10210 Athens
Tel: 30 1 4894911

TECHNICAL SERVICE CENTERS (cont)

Hong Kong
Schmidt & Co (H.K.) Ltd.
18/FL., Great Eagle Centre
23 Harbour Road
Wanchai
Tel: 852 5 8330222

India
Hinditron Services Pvt. Ltd
1st Floor, 17-B,
Mahal Industrial Estate
Mahakali Road, Andheri East
Bombay 400 093
Tel: 91 22 6300043

Hinditron Services Pvt. Inc.
33/44A Raj Mahal Villas Extn.
8th Main Road
Bangalore 560 080
Tel: 91 812 363139

Hinditron Services Pvt. Ltd.
Field Service Center
Emerald Complex 1-7-264
5th Floor
114 Sarojini Devi Road
Secunderabad 500 003
Tel: 08 42-821117

Hindtron Services Pvt. Ltd.
15 Community Centre
Panchshila Park
New Delhi 110 017
Tel: 011-6433675

Indonesia
P.T. Lamda Triguna
P.O. Box 6/JATJG
Jakarta 13001
Tel: (021) 8195365

Israel
R.D.T. Electronics Engineering, Ltd.
P.O. Box 43137
Tel Aviv 61430
Tel: 972 3 483211

Italy
Philips S.p.A.
Sezione I&E / T&M
Viale Elvezia 2
20052 Monza
Tel: 39-39-363-5342

Japan
John Fluke Mfg. Co., Inc.
Japan Branch
Sumitomo Higashi Shinbashi Bldg.
1-1-11 Hamamatsucho
Minato-ku
Tokyo 105
Tel: 81 3 434-0181

Korea
Myoung Corporation
Yeo Eui Do P.O. Box 14
Seoul 150
Tel: 82 2 784-9942

Malaysia
Mecomb Malaysia Sdn. Bhd.
P.O. Box 24
46700 Petaling Jaya
Selangor
Tel: 60 3 774-3422

Mexico
Mexel Servicios en Computacion
Instrumentacion y Perifericos
Blvd. Adolfo Lopez Mateos No. 163
Col. Mixcoac
Mexico D.F.
Tel: 52-5-563-5411

Netherlands
Philips Nederland
Test & Meetapparaten Div.
Postbus 115
5000 AC Tilburg
Tel: 31-13-352455

New Zealand
Philips Customer Support
Scientific & Industrial
2 Wagener Place
Mt. Albert
Auckland
Tel: 64 9 894-160

Norway
Norsk A/S Philips
I&E Service
Sandstuveien 70
Postboks 1 Manglerud
N 0680 OSLO 6
Tel: 47-2-680200

Pakistan
International Operations (PAK) Ltd.
505 Muhammadi House
I.I. Chundrigar Road
P.O. Box 5323
Karachi
Tel: 92 21 221127, 239052

Peru
Importaciones & Representaciones
Electronicas S.A.
Avad Franklin D. Roosevelt 105
Lima 1
Tel: 51 14 288650

Philippines
Spark Radio & Electronics Inc.
Greenhills, P.O. Box 610
San Juan, Metro-Manila Zip 3113
Tel: 63-2-775192

Portugal
Philips Portuguese S.A.
I&E Division
Estrada de Outurela-Carnaxide
2795 Linda-A-Velha
Tel: 418 00 71

Singapore
Rank O'Connor's Singapore Pte Ltd
98 Pasir Panjang Road
Singapore 0511
Tel: 65 4737944

South Africa
South African Philips (Pty) Ltd.
Service Department
195 Main Rd
Martindale, Johannesburg 2092
Tel: 27 11 470-5255

Spain
Philips Iberica Sae
Depto. Tecnico Instrumentacion
c/Martinez Villergas 2
28027 Madrid
Tel: 34 1 4042200

Sweden
Philips Kistaindustrier AB
I&E Technical Customer Support
Borgarfjordsgatan 16
S 164 93 Kista
Tel: 46-8-703-1000

Switzerland
Philips A.G.
Technischer Kundendienst
Postfach 670
Allmendstrasse 140
CH-8027 Zurich
Tel: 41 1 482211

Taiwan, R.O.C.
Schmidt Electronics Corp.
5th Floor, Cathay Min Sheng
Commercial Building,
344 Min Sheng East Road
Taipei
Tel: 886 2 501-3468

Thailand
Measuretronix Ltd.
2102/31 Ramkamhaeng Rd.
Bangkok 10240
Tel: 66 2 375-2733, 375-2734

Turkey
Turk Philips Ticaret A.S.
Inonu Caddesi 78/80
Posta Kutusu 504-Beyoglu
Istanbul
Tel: 90 1 1435891

TECHNICAL SERVICE CENTERS (cont)

Uruguay
Coasin Uruguaya S.A
Casilla de Correo 1400
Libertad 2525
Montevideo
Tel: 598-2-789015

Venezuela
Coasin C.A.
Calle 9 Con Calle 4, Edif. Edinurbi
Apartado de Correos Nr-70-136
Los Ruices
Caracas 1070-A
Tel: 58 2 241-0309, 241-1248

West Germany
Philips GmbH
Service VSF
Unternehmensbereich Elektronik
fur Wissenschaft und Industrie
Oskar-Messter-Strasse 18
8045 Ismaning
Tel: 49-89-9605-260

Appendix 9f
Glossary

---

a/d converter

Analog-to-Digital Converter. A circuit used to convert information in analog form (such as voltage or current) into digital form.

ADC

See a/d converter.

address

The channel number that represents an I/O channel.

analog

The representation of numerical quantities that do not have discrete values but that are continuously variable. Electrical measurements are analog in nature until converted into digital signals by an a/d converter. See digital.

ANSI

American National Standards Institute; an industry supported U.S. organization that primarily serves to coordinate and publish technical standards requested by its members.

ASCII

American Standard Code for Information Interchange; ASCII is a standardized code set of 128 characters, including upper and lower case alphabet, numerals, and a set of control characters (line feed, carriage return, etc.)

baud

Originally used to express the capabilities of a telegraph transmission facility in terms of "modulation rate per unit of time." For practical purposes, it is now used interchangeably with "bits per second" as a unit of measure for data flow.

BCD

Binary Coded Decimal; BCD is a digital data format in which each digit of a decimal (base 10) number is represented in order, by its binary equivalent. For example, the decimal number 597 is represented as (0101 1001 0111) in BCD. See binary.

binary

The base two number system; binary numbers are made up of ones and zeros. The least significant digit represents one, the next digit represents twos, the next digit represents fours, and so on with each digit of the binary number corresponding to the next higher exponent of two. See also BCD.

bit

The smallest element of a binary number; a bit (contraction of "binary digit") either represents a high or a low state (one or zero, also mark or space).

bridge

In a measuring system, an instrument in which part or all of its circuit measures one or more electrical quantities. Quarter-, half-, and full-bridge strain gauges can be used. See microstrain, strain gauge.

Burst Scan Mode

A scanning mode used by the -165 Fast A/D Converter to fill its local buffer with channel measurement data at a rate of 1000 measurements per second. No line power frequency filtering is used in this mode of operation. Burst scanning is started with an explicit command and stopped by either a command or a trigger event.

CCITT

Comite Consulatif International Telegraphic et Telephonique; an international organization concerned with devising and proposing standards for international telecommunications.

CD

Carrier Detect; RS-232-C, line 8. This signal is sent by DCE to tell DTE that a suitable carrier is present at the DCE. See also DTE, DCE.

Continuous Scan Mode

A scanning mode used by the -165 Fast A/D Converter to make measurement channel readings available to the host computer at a continuous rate of 70 readings per second. Filtering is used to reduce line power frequency noise. Continuous scanning is started when channels on the Fast A/D Converter are defined and stopped by an explicit command.

Counter/Totalizer

An option assembly that measures frequency and totalizes event occurrences.

CTS

Clear to Send; RS-232-C line 5. Sent by DCE to DTE when it is ready to accept data. See also DCE, DTE.

current shunt

Traditionally, a precision, low-value resistor placed across an ammeter's terminals to increase its range. The Front End uses a current shunt resistor in the input circuitry of each channel of the Current Input Connector. The shunt resistor allows the current input to develop a voltage that the Analog to Digital Converter can measure.

DCE

Data Communication Equipment; the RS-232-C term for computer equipment that sends or receives data. Modems are examples of DCE. The other type of computer equipment specified by RS-232 is DTE, Data Terminal Equipment.

default

The value or state selected by system software (or firmware) when the user doesn't specify one.

digital

A circuit or device concerned only with high or low states (or pulses), generating a logical or numerical result. Digital computers, calculators, and digital watches use digital circuitry. See analog.

DSR

Data Set Ready; RS-232-C line 6. Indicator from DCE to DTE that a suitable line connection has been made, that all preliminary line protocol is complete, and that it is ready for data transfer. See also DCE, DTE.

DTE

Data Terminal Equipment; the RS-232-C Standard term for computer equipment where digital information originates or terminates. The Front End, terminals, and computers are examples of DTE. The other type of computer equipment specified by RS-232-C is DCE, Data Communication Equipment.

DTR

Data Terminal Ready; RS-232-C line 20. Sent by DTE to DCE when it is ready to transfer data.

EIA

Electronic Industries Association; the organization that devises electrical and data communication standards including RS-232-C and RS-422.

External Trigger

External hardware trigger generates four types of trigger inputs: high, low, rising edge, and falling edge. These external trigger inputs are set true if they are in the appropriate logic state and triggering is enabled for that state.

Fast A/D Converter

The -165 option provides an a/d converter that can operate in continuous scan mode (up to 70 dc voltage readings per second available continuously to the host computer) or burst scan mode (1000 readings per second stored in an on-board buffer).

floating point

A numerical notation useful for computer calculations. In the Front End, the resolution of a floating-point number is 7 digits, and its range is up to 38 places either side of the decimal. When the decimal point is out of range of the displayed digits, the number is displayed in exponential form (E+mn or E-mn, where mn represents the number of places the decimal point must be moved; + for right and - for left).

frequency

The number of cycles-per-second (Hertz) of an alternating signal. Can be measured using the Counter/Totalizer Option (-167).

gauge factor

A strain gauge parameter, gauge factor specifies the ratio of the change in resistance to the strain imposed on the strain gauge being used. A typical gauge factor would be 2.0.

GND

Ground; RS-232-C line 1. A chassis connection intended for the cable shield. Not to be used for signal reference.

High Performance A/D Converter

The -161 option can be used as an alternative to, or in conjunction with the -165 Fast A/D Converter in the same system. The High Performance A/D Converter is slower and somewhat more accurate than the Fast A/D Converter. It operates at about 16 dc voltage readings per second.

Host Computer

Any computer that uses a RS-232 serial link to send commands to, and receive data from Helios Plus. Typically, a host computer would be a PC.

interface

A hardware and software connection of a device to a system.

interference

The presence of undesirable energy in a circuit, caused by electrostatically or electromagnetically coupled external circuits.

I/O

Input/Output.

isothermal

An area of equal temperature. The Thermocouple Input Connector (-175) uses an alloy isothermal terminal block to stabilize and equalize input lead temperatures.

mainframe

The controller portion of the Helios Plus Data Acquisition System. The mainframe controls the measurement subsystem, takes commands from the host computer, and passes measurement data to the host computer.

MICROSTRAIN

The unit of strain measured by a strain gauge.

modem

A transmitter and receiver of digital data. One modem receives data from an attached computer terminal and sends the data on a modulated carrier signal to another modem. The other modem demodulates the received carrier and sends the recovered data to another computer terminal.

overrange

The state indicated by a measurement instrument when an input signal is greater (or lower) than the range that the instrument can measure.

parity

A method of error detection that uses one extra bit for each unit of information (such as a character). The parity bit is set to one or zero so that the total number of one-bits in the byte is even or odd, depending on the type of parity in use.

pca

Printed circuit assembly. A circuit board and attached components.

Poisson's ratio

A strain gauge parameter; the ratio of the lateral strain to the longitudinal strain in a specimen subjected to a longitudinal stress.

port

A connection point used for data transfer. See interface.

protective ground

The common chassis ground that is connected to earth ground through the ground wire in the line power cord. See also signal ground.

REC

Received Data, RS-232-C line 3. This line carries the stream of serial data from DCE to DTE. See also serial data, DCE, DTE.

RET

Return; RS-232-C line 7. A signal return line that serves as a zero voltage reference point for all other signals.

RFI

Radio Frequency Interference; see interference.

RS-232-C

A digital communications standard agreed upon by participating manufacturers of data communication equipment for the transfer of serial digital data between data communication equipment (DCE) and data terminal equipment (DTE). The Front End is a DTE device. The standard is published and maintained by the Electronic Industries Association, 2001 Eye Street, N.w., Washington, D.C. 20006. See also DCE, DTE, serial data.

RTD

Resistance Temperature Detector; a device with a resistance that varies predictably with changes in temperature. Made of various materials, with platinum being both the most popular and the most accurate.

RTS

Request To Send; RS-232-C line 4. The signal from DTE to DCE when it has data to transmit. See also DTE, DCE.

scanner

A type of option used by the Front End to select a desired analog input channel for measurement.

serial data

Information transmitted one bit at a time over a single line at a predefined bit rate (baud). See also baud.

serial link

Internal Front End bus between controller and option assemblies.

serial link address

Address for a serial link device. The address range is manually set on the a/d converter, Analog Output, and Digital I/O Assemblies.

serial link device

An assembly performing an output or measurement input function. There are positions (horizontal slots) for six such devices in each the Front End.

serial port

A connection point on a computer that is used to transfer information in a serial manner. Data transmission through a serial port in the Front End is in the form of asynchronous ASCII codes.

shield input

The input lead on various Front End options that can be connected to the low lead at a measurement point to reduce interference. The shield input may be attached to the braided shield wire surrounding the conductor in the cable.

signal ground

A conductor establishing electrical ground reference potential for all transmitting circuits in a communications network.

strain gauge

A resistive transducer the electrical output of which is proportional to the amount it is deformed due to strain in the material to which it is bonded. Strain gauge circuits use fixed precision resistors and resistors bonded to a surface to be mechanically loaded, in quarter-, half-, or full-bridge configuration. See also bridge, gauge factor, microstrain.

thermocouple

A pair of dissimilar conductors joined together, forming a junction that generates a voltage that is proportional to temperature.

The Front End provides for 12 different types of thermocouples.

transducer

A device that converts energy from one form to another. An example of a transducer is a strain gauge.

Trigger Event

This term is used in conjunction with the Fast A/D Converter's Burst Scanning Mode. A trigger event occurs either when an external trigger input meets the specified enabling and filtering criteria (allowing the trigger input to affect the burst scanning activity of the Fast A/D Converter) or when a hitrigger or lotrigger value is exceeded for the number of consecutive scans specified by the filter count variable. The trigger event causes a Fast A/D Converter to stop capturing data in its buffer after the additional number of scans specified by the trigger position variable.

Trigger Filter

Trigger inputs are filtered by the trigger filter, which counts the number of consecutive scans each trigger condition is true. A trigger event is generated if the number of scans equals the programmed trigger filter value plus one. If the filter value = 0, the trigger event occurs on the first scan on which the trigger condition is true. If the filter value = 1, a one-scan delay is used. The trigger condition may be an external trigger input or a channel HITRIGGER or LOTRIGGER value that has been exceeded.

Trigger Input

A trigger input refers to the external trigger input on the Fast A/D Converter. This input is used in conjunction with Burst Scan Mode. Once level/edge and filtering criteria are met, trigger inputs generate a trigger event.

XMT

Transmitted Data, RS-232-C line 2. The line that carries the stream of serial data from DTE to DCE. See also RS-232-C, serial data, DCE, DTE.

X-OFF

CTRL/S or DC3 ASCII code (decimal 19); sent by the receiving station to halt transmission from the sending station when information is coming too fast for the receiver to process. When able to receive more information, the receiving station sends X-ON to restart the transmitting station. See also X-ON.

X-ON

CTRL/Q or DC1 ASCII code (decimal 17); sent by the receiving station to cause the transmitting station to resume transmission of data after being halted by an X-OFF command. See also X-OFF.

Appendix 9g
Programming Examples

GENERAL PROGRAMMING EXAMPLE ---1

This example can be used with the -161 High Performance A/D Converter or the -165 Fast A/D Converter.

```
RESET ALL                     Resets the whole system.

MODE = TERM                   Select the TERMinal Mode.

SEND MEMSIZ                   Check how much memory is
                              available.

DEF CHAN(0..9)=TC,TYPE=KNBS,CHFN=POLY(0,5,20)

                              Define channels 0..9 as
                              thermocouple inputs with
                              polynomial coefficients 0,
                              5 and 20.

DEF CHAN(10..19)=DVIN, HIHI=1.5, LOLO=0.8, HI=1.30,
                              LO=1.20, HYST=15

                              Define channels 10..19 as
                              Direct Volt Inputs with
                              four limits and hysteresis.
```

```
DEF SCAN(1)=CHAN(0..9)
DEF SCAN(2)=CHAN(10..19)

                                   Define two scan tasks, one
                                   for each group of channels.

DEF SBUF(1)=10                     Define buffer for
                                   measurement data. Reserve
                                   memory for 10 records.


DEF SBUF(2)=20                     Define buffer for
                                   measurement data. Reserve
                                   memory for 20 records

DEF ABUF=15                        Define an alarm buffer.
                                   Reserve memory for 15 alarm
                                   messages.

LIST SCAN(1)                       Check if everything is corre
LIST SCAN(2)

LIST SBUF(1)
LIST SBUF(2)

LIST ABUF
```

```
START SCAN(1),OUTPUT=SBUF(1),INTERVAL=60

                                   Start Scan (1) with an
                                   interval of 60 sec.

START SCAN(2),OUTPUT=SBUF(2),INTERVAL=10,DELAY=120

                                   Start Scan (2) with an
                                   interval of 10 sec. but
                                   wait for two minutes before
                                   starting the second scan.
                                   Store data from both scans
                                   in their respective
                                   buffers.

SEND ABUF                          Check if alarm conditions have
                                   been detected.

SEND SBUF(1)                       Send the first buffer record of
                                   scan task 1 to the terminal and
                                   remove it from the buffer.

SHOW SBUF(2)                       Send the first record of scan
                                   task 2 to the terminal but do
                                   not remove it from the buffer

STOP SCAN(1)                       Stop the scan tasks.
STOP SCAN(2)
```

FAST A/D CONVERTER PROGRAMMING EXAMPLES

Continuous Scan Mode Example

The following example, sent from a host computer, retrieves Helios Plus measurement data from a Fast A/D Converter operating in Continuous Scan Mode.

	Since no specific Fast A/D Converter address is mentioned, 0 is assumed.
MODE = COMP	Operate in computer mode.
FORMAT = DECIMAL	Return data to the host in decimal format.
DEF CHAN(0..19) = DVIN, MAX=.06	Define analog input channels 0 through 19 for direct voltage measurement on the 64 mV range.
SEND CHAN(0)	Retrieve the resulting measurement on channel 0.
	If 60 mV is measured, the returned reading is:
	6.00000E-02

NOTE

This example could also be used to retrieve measurement data from the High Performance A/D Converter. The required syntax is identical: SEND CHAN either retrieves the Fast A/D Converter channel information or tells the High Performance A/D Converter to start operating and return measured values as they are acquired. The Fast A/D Converter, which scans channels continuously, supplies the host computer with readings at a faster rate.

Burst Scan Mode Example 1

```
MODE = COMP
FORMAT = DECIMAL
DEF CHAN(0..3) = DVIN, SINGLE, MAX=.5
START BSCAN(0)
```

This example assumes that the Fast A/D address is set to 0.

MODE = COMP — Helios Plus to operate in computer mode

FORMAT = DECIMAL — Return data to host in decimal format.

DEF CHAN(0..3) = DVIN, SINGLE, MAX=.5

Define analog input channels 0 through 3 for single ended direct voltage measurement on the 512 mV range.

START BSCAN(0) — Start burst scanning channels 0 through 3.

STOP BSCAN(0) — Stop burst scanning.

```
SEND BSCAN(0,0)
```

Retrieve the most recent scan buffer record for Fast A/D Converter 0. For example, measured values of 60 mV, 180 mV, 100 mV, and 120 mV on channels 0 through 3, respectively, are returned as:

```
6.00000E-02
1.80000E-01
1.00000E-01
1.20000E-01
```

SEND SCANUM could also be used first to determine the complete range of records in the scan buffer. SEND BSCAN could then be used to retrieve all or some of the scan records.

Burst Mode Example 2

```
MODE = COMP
FORMAT = DECIMAL
TUNIT = FAHRENHEIT
DEF CHAN(55) = TC, TYPE = KNBS, HITRIGGER = 120,&
LOTRIGGER =100
DEF BSCAN = FAD(5), FILTERCNT = 9, TRIGPOS = 18
INTERRUPT = ON
START BSCAN(5)
```

This example assumes that the Fast A/D Converter address is set to 5 and that a KNBS type thermocouple is connected to channel 5 on that a/d converter.

MODE = COMP	Operate in computer mode.
FORMAT = DECIMAL	Return data to the host in decimal format.
TUNIT = FAHRENHEIT	Define system variable FAHRENHEIT.
DEF CHAN(55) = TC, TYPE = KNBS,	Define analog input channel 55 for KNBS type thermocouple measurement
HITRIGGER = 120, LOTRIGGER = 100,	Trigger condition occurs if input temperature goes above 120° or below 100° Fahrenheit.
DEF BSCAN = FAD(5),	Define Fast A/D Converter 5 for Burst Scan Mode.
FILTERCNT = 9,	Enable trigger event after 9 additional consecutive out-of-limits scans on channel 55.
TRIGPOS = 18	Measure and store 18 additional scans after trigger event, then stop burst scan.
INTERRUPT = ON	Report status changes to the host.
START BSCAN(5)	Start burst scanning for channel 55.

When a trigger event occurs, the STATUS system variable (bit 4) is set, indicating to the host computer that scan buffer data can be retrieved soon with the SEND BSCAN command. In this case, 18 more scans must occur before the data can be retrieved. The host computer can tell when the Fast A/D Converter has finished the 18th scan after the trigger event by using the SEND SCANIPS command.