Watson MSS Communications Manual

Mercury Speciation System

119923-00 • 21May2019

Thermo Fisher SCIENTIFIC

Contents

Chapter 1
80*i* C-Link Protocol Commands

This chapter provides a description of the C-Link protocol commands that can be used to remotely control a Model 80*i* analyzer using a host device such as a PC or a datalogger. C-Link protocol may be used over RS-232, RS-485, or Ethernet. C-Link functions can be accessed over Ethernet using TCP/IP port 9880. Streaming data may be accessed over Ethernet using TCP/IP port 9881. Up to three simultaneous connections are allowed per protocol.

Instrument Identification Number

Each command sent to the instrument must begin with the American Standard Code for Information Interchange (ASCII) symbol or byte value equivalent of the instrument's identification number plus 128. For example, if the instrument ID is set to 25, then each command must begin with the ACSII character code 153 decimal. The instrument ignores any command that does not begin with its instrument identification number. If the instrument ID is set to 0, then this byte is not required. For more information on changing Instrument ID, see Chapter 3, “Operation.”

Commands

The instrument must be in the remote mode in order to change instrument parameters via remote. However, the command "set mode remote" can be sent to the instrument to put it in the remote mode. Report commands (commands that don’t begin with “set”) can be issued either in the remote or local mode. For information on changing modes, see Chapter 3, “Operation.”

The commands can be sent in either uppercase or lowercase characters. Each command must begin with the proper instrument identification number (ASCII) character. The command in the example that follows begins with the ASCII character code 208 decimal, which directs the command to the Model 80i, and is terminated by a carriage return “CR” (ASCII character code 13 decimal).

<ASCII 208>	T	I	M	E	<CR>

If an incorrect command is sent, a “bad command” message will be received. The example that follows sends the incorrect command “set alarm eductor pres max” instead of the correct command “set alarm eductor pres max 20.0.”

```
Send:        set alarm eductor pres max
Receive:     set alarm eductor pres max bad cmd
```

Table 1–1 provides a description of the command response errors.

Table 1–1. Command Response Error Descriptions

Command Response	Description
too high	Supplied value is higher than the upper limit
too low	Supplied value is lower than the lower limit
invalid string	Supplied string invalid (typically because a letter was detected when the value should be numeric)
data not valid	Supplied value is not acceptable for entered command
can't, wrong settings	Command not allowed for current measurement mode
can't, mode is service	Command not allowed while instrument is in service mode

The "save" and "set save params" commands store parameters in FLASH. It is important that this command be sent each time instrument parameters are changed. If changes are not saved, they will be lost in the event of a power failure.

Accessing Streaming Data

Streaming data is sent out the serial port or the Ethernet port on a user-defined periodic basis. Streaming data over Ethernet is only generated when a connection is made on TCP port 9881. Up to three simultaneous connections are allowed per protocol.

Service Mode

If the Service Mode is active, C-Link "set" commands are not allowed. This is to prevent parameters from being changed remotely while the unit is being serviced locally.

Commands List

Table 1–2 lists the 80*i* C-Link protocol commands. The interface will respond to the associated command strings.

Table 1–2. C-Link Protocol Commands

Command	Description	Page
1	Simulates pressing soft key 1 pushbutton	1-31
2	Simulates pressing soft key 2 pushbutton	1-31
3	Simulates pressing soft key 3 pushbutton	1-31
4	Simulates pressing soft key 4 pushbutton	1-31
addr dns	Reports/sets domain name server address	1-48
addr gw	Reports/sets default gateway address	1-48
addr ip	Reports/sets IP address	1-48
addr nm	Reports/sets netmask address	1-49
addr ntp	Reports the IP address for the NTP time server	1-49
alarm blow back pres max	Reports/sets blow back alarm maximum value	1-14
alarm blow back pres min	Reports/sets blow back alarm minimum value	1-14
alarm chamber temp max	Reports/sets chamber temperature alarm maximum value	1-15
alarm chamber temp min	Reports/sets chamber temperature alarm minimum value	1-15
alarm conc hg0 max	Reports/sets current Hg0 concentration alarm maximum value	1-15
alarm conc hg0 min	Reports/sets current Hg0 concentration alarm minimum value	1-15
alarm conc hg2+ max	Reports/sets current Hg2+ concentration alarm maximum value	1-15
alarm conc hg2+ min	Reports/sets current Hg2+ concentration alarm minimum value	1-15
alarm conc hgt max	Reports/sets current Hgt concentration alarm maximum value	1-15
alarm conc hgt min	Reports/sets current Hgt concentration alarm minimum value	1-15
alarm converter temp max	Reports/sets current converter temperature alarm maximum value	1-16
alarm converter temp min	Reports/sets current converter temperature alarm minimum value	1-16

Command	Description	Page
allow mode cmd	Reports/sets the current allow mode setting which configures the instrument to either accept or ignore the "set mode local" and "set mode remote" commands	1-52
analog iout range	Reports/sets analog current output range per channel	1-54
analog vin	Retrieves analog voltage input data per channel	1-55
analog vout range	Reports/sets analog voltage output range per channel	1-55
avg time Hg0	Reports/sets Hg0 averaging time	1-11
avg time Hgt	Reports/sets Hgt averaging time	1-11
avg time Cal	Reports/sets Cal averaging time	1-11
baud	Reports/sets current baud rate	1-49
bb filter	Sets system gas mode blow back filter	1-33
bb period	Reports/sets the blow back frequency	1-33
bb set pres	Reports blow back pressure in psig	1-33
bb set pres counts	Reports/sets blow back pressure in counts	1-33
bb stinger	Reports/sets system gas mode to blow back stinger	1-34
bb stinger duration	Reports/sets blow back stinger duration seconds	1-34
bb system duration	Reports/sets blow back filter duration seconds	1-34
bench set temp	Reports/sets bench temperature	1-35
cal hg0 bkg	Sets/auto-calibrates Hg0 background	1-30
cal hg0 coef	Sets/auto-calibrates Hg0 coefficient	1-29
cal hgt bkg	Sets/auto-calibrates Hgt background	1-30
cal hgt coef	Sets/auto-calibrates Hgt coefficient	1-29
cal mode O2 pct	Returns/sets O2 cal mode O2 percent	1-37
chamber temp	Reports the temperature of the optical chamber	1-35
clr lrecs	Clears away only long records that have been saved	1-22
clr records	Clears away all logging records that have been saved	1-22
clr srecs	Clears away only short records that have been saved	1-22
contrast	Reports/sets current screen contrast	1-45
conv set temp	Reports/sets converter set temperature	1-35
conv temp	Reports converter temperature	1-35
copy lrec to sp	Sets/copies current lrec selection into the scratch pad	1-27
copy sp to lrec	Sets/copies current selections in scratch pad into lrec list	1-27

Command	Description	Page
copy sp to srec	Sets/copies current selections in scratch pad into srec list	1-27
copy sp to stream	Sets/copies current selections in scratch pad into stream list	1-27
copy srec to sp	Sets/copies current srec selection into the scratch pad	1-27
copy stream to sp	Sets/copies current streaming data selection into the scratch pad	1-27
current sample	Reports current sample measurement (Hg0 or Hgt)	1-38
custom	Reports/sets defined custom range concentration	1-40
date	Reports/sets current date	1-46
default params	Sets parameters to default values	1-47
dhcp	Reports/sets state of use of DHCP	1-50
diag volt iob	Reports diagnostic voltage level for optional I/O expansion board	1-21
diag volt mb	Reports diagnostic voltage level for motherboard	1-21
diag volt mib	Reports diagnostic voltage level for measurement interface board	1-21
diag volt probe	Reports diagnostic voltage level for 82*i* measurement interface board	1-21
dig in	Reports status of the digital inputs	1-56
dilution ratio	Reports/sets dilution ratio	1-35
dilution set pres	Reports dilution pressure in psig	1-36
dilution set pres counts	Reports/sets dilution pressure in counts	1-36
din	Reports/sets digital input channel and active state	1-56
do (down)	Simulates pressing down pushbutton	1-31
dout	Reports/sets digital output channel and active state	1-56
dtoa	Reports outputs of the digital to analog converters per channel	1-57
duration time Hg0	Reports/sets the Hg0 sample duration time (minutes)	1-38
duration time Hgt	Reports/sets the Hgt sample duration time (minutes)	1-38
eductor set pres	Reports eductor pressure in psig	1-36
eductor set pres counts	Reports/sets eductor pressure in counts	1-37
en (enter)	Simulates pressing enter pushbutton	1-31
er	Returns a brief description of the main operating conditions in the format specified in the commands	1-23
erec	Returns a snapshot of the main operating conditions (measurements and status) in the specified format	1-24
erec format	Reports/sets erec format (ASCII or binary)	1-25
erec layout	Reports current layout of erec data	1-25

Command	Description	Page
flags	Reports 8 hexadecimal digits (or flags) that represent the status of the PMT, gas mode, and alarms	1-13
flow	Reports current measured flow	1-12
format	Reports/sets current reply termination format	1-50
gas mode	Reports current mode of sample, zero, or span	1-41
he (help)	Simulates pressing help pushbutton	1-31
hg0	Reports current Hg0 concentration	1-12
hg0 bkg	Reports/sets current Hg0 background	1-31
hg0 coef	Reports/sets current Hg0 coefficient	1-30
hg0 gas	Reports/sets Hg0 span gas concentration	1-30
Hg0 next time	Reports current the next time for the Hg0 channel to run while in sample mode	1-39
hgt	Reports current Hgt concentration	1-12
hgt bkg	Reports/sets current Hgt background	1-31
hgt coef	Reports/sets current Hgt coefficient	1-30
hgt gas	Reports/sets Hgt span gas concentration	1-30
host name	Reports/sets host name string	1-51
instr name	Reports instrument name	1-51
instrument id	Reports/sets instrument id	1-51
internal temp	Reports current internal instrument temperature	1-13
isc (iscreen)	Retrieves framebuffer data used for the display	1-31
lamp comp	Reports/sets lamp compensation on/off	1-37
lamp intensity	Reports lamp intensity	1-37
layout ack	Disables stale layout/layout changed indicator ('*')	1-54
le (left)	Simulates pressing left pushbutton	1-31
list din	Lists current selection for digital input	1-22
list dout	Lists current selection for digital output	1-22
list lrec	Lists current selection lrec logging data	1-22
list sp	Lists current selection in the scratchpad list	1-22
list srec	Lists current selection srec logging data	1-22
list stream	Lists current selection streaming data output	1-22
list var aout	Reports list of analog output, index numbers, and variables	1-58
list var din	Reports list of digital input, index numbers, and variables	1-58
list var dout	Reports list of digital output, index numbers, and variables	1-58

Command	Description	Page
lr	Outputs long records in the format specified in the command	1-23
lrec	Outputs long records	1-24
lrec format	Reports/sets output format for long records (ASCII or binary)	1-25
lrec layout	Reports current layout of lrec data	1-25
lrec mem size	Reports maximum number of long records that can be stored	1-26
lrec per	Reports/sets long record logging period	1-26
malloc lrec	Reports/sets memory allocation for long records	1-27
malloc srec	Reports/sets memory allocation for short records	1-27
me (menu)	Simulates pressing menu pushbutton	1-31
meas mode	Reports/sets which measurement mode is active	1-42
mode	Reports operating mode in local, service, or remote	1-53
no of lrec	Reports/sets number of long records stored in memory	1-26
no of srec	Reports/sets number of short records stored in memory	1-26
pmt supply	Reports/sets PMT supply power on/off	1-44
pmt voltage	Reports current PMT voltage	1-13
power converter	Reports/sets converter power on/off for the selected probe	1-42
power eductor	Reports/sets eductor power on/off for the selected probe	1-43
power probe	Reports/sets probe power on/off for the selected probe	1-43
power stinger	Reports/sets stinger power on/off for the selected probe	1-43
power up mode	Reports/sets the power up mode which configures the instrument to power up in either the local/unlocked mode or the remote/locked mode.	1-53
pres	Reports current optical chamber pressure	1-13
pres cal	Reports/sets pressure used for calibration	1-31
pres comp	Reports/sets pressure compensation on/off	1-44
probe no	Reports/sets the active probe (hydra only)	1-44
probe set failsafe temp	Reports/sets probe failsafe temperature	1-44
probe set temp	Sets probe temperature	1-45
program no	Reports instrument program number	1-54
push	Simulates pressing a key on the front panel	1-31
range hg0	Reports/sets current Hg0 range	1-39
range hg2+	Reports/sets current Hg2+ range	1-39
range hgt	Reports/sets current Hgt range	1-39
range mode	Reports/sets current range mode	1-40

Command	Description	Page
react temp	Reports current optical chamber temperature	1-13
ref intensity	Reports reference lamp intensity in Hz	1-45
relay	Reports relay logic status to for the designated relay(s)	1-58
relay stat	Sets relay logic status to for the designated relay(s)	1-58
ri (right)	Simulates pressing right pushbutton	1-31
ru (run)	Simulates pressing run pushbutton	1-31
s span	Sets system gas mode to system span	1-42
s zero	Sets system gas mode to system zero	1-42
sample	Sets zero/span valves to sample mode	1-41
save	Stores parameters in FLASH	1-47
save params	Stores parameters in FLASH	1-47
sc (screen)	C-series legacy command that reports a generic response (Use iscreen instead)	1-33
sp field	Reports/sets item number and name in scratch pad list	1-28
span	Sets zero/span valves to span mode	1-41
span inst	Reports/sets the instrument span level	1-41
span sys	Reports/sets the system span level	1-41
sr	Reports last short record stored	1-23
srec	Reports maximum number of short records	1-24
srec format	Reports/sets output format for short records (ASCII or binary)	1-25
srec layout	Reports current layout of short record data	1-25
srec mem size	Reports maximum number of short records	1-26
srec per	Reports/sets short record logging period	1-26
stack o2 channel	Returns/sets O2 analog input channel	1-38
stream per	Reports/sets current set time interval for streaming data	1-29
stream time	Reports/sets a time stamp to streaming data or not	1-29
time	Reports/sets current time (24-hour time)	1-47
tz	Reports the "tz" timezone string for the NTP server	1-54
up	Simulates pressing up pushbutton	1-33
zero	Sets zero/span valves to zero mode	1-41

Measurements

avg time hg0

This command reports the Hg0 averaging time (during sample mode) in seconds. The example that follows shows that the averaging time is 60 seconds, according to the following table.

```
Send:         avg time hg0
Receive:      avg time hg0 6: 60 sec
```

set avg time hg0 time *selection*

This command sets the Hg0 averaging time (during sample mode) according to **Table 1–3**. The example that follows sets the Hg0 averaging time to 30 seconds.

```
Send:         set avg time hg0 5
Receive:      set avg time hg0 5 ok
```

avg time hgt

This command reports the Hgt averaging time (during sample mode) in seconds. The example that follows shows that the averaging time is 300 seconds, according to the following table.

```
Send:         avg time hgt
Receive:      avg time hgt 11: 300 sec
```

set avg hgt time *selection*

This command sets the Hgt averaging time (during sample mode) according to **Table 1–3**. The example that follows sets the Hgt averaging time to 30 seconds.

```
Send:         set avg time hgt 5
Receive:      set avg time hgt 5 ok
```

avg time cal

This command reports the calibration averaging time (during calibration modes) in seconds. The example that follows shows that the averaging time, for both Hg0 and Hgt while in calibration mode, is 120 seconds, according to the following table.

```
Send:         avg time cal
Receive:      avg time cal 8: 120 sec
```

set avg time cal *selection*

This command sets the calibration averaging time (during calibration modes) according to **Table 1–3**. The example that follows sets the calibration averaging time to 60 seconds.

```
Send:         set avg time cal 6
Receive:      set avg time cal 6 ok
```

Table 1–3. Averaging Times

Selection	Seconds
0	1
1	2
2	5
3	10
4	20
5	30
6	60
7	90
8	120
9	180
10	240
11	300

hg0
hg2+
hgt

These commands report the measured Hg0, Hg2+, and Hgt concentrations. The example that follows shows that the Hg0 concentration is 15.35.

```
Send:         hg0
Receive:      hg0 1.535E+01 ug/m3
```

flow

This command reports the current measured flow. The example that follows reports that the flow measurement is 0.391 lpm.

```
Send:         flow
Receive:      flow 0.391 lpm
```

internal temp

This command reports the current internal instrument temperature. The example that follows shows that the internal temperature is 30 °C.

```
Send:         internal temp
Receive:      internal temp 30 deg C, actual 33.5
```

pmt voltage

This command reports the current PMT voltage. The example that follows reports that the current PMT voltage is 799.2 volts.

```
Send:         pmt voltage
Receive:      pmt voltage 799.2
```

pres

This command reports the current Hg pressure. The example that follows shows that the actual Hg pressure is 42.8 mmHg.

```
Send:         pres
Receive:      pres 42.8 mm Hg
```

react temp

This command reports the current optical chamber temperature. The example that follows reports that the current optical chamber temperature is 45.0 °C.

```
Send:         react temp
Receive:      react temp 45.0 deg C
```

flags

This reports 8 hexadecimal digits (or flags) that represent status of the pressure and temperature compensation, gas mode, and alarms. To decode the flags, each hexadecimal digit is converted to binary as shown in the **Figure 1–1**. It is the binary digits that define the status of each parameter. In the example that follows, the instrument is reporting that the password lock is ON, temperature compensation is OFF, pressure compensation is ON, measure mode is AUTO, gas mode is SAMPLE, converter power is ON, eductor power is OFF, umbilical 2 power is OFF, umbilical 1 power is OFF, probe power is OFF, and there are no alarms.

```
Send:         flags
Receive:      flags 28300000
```

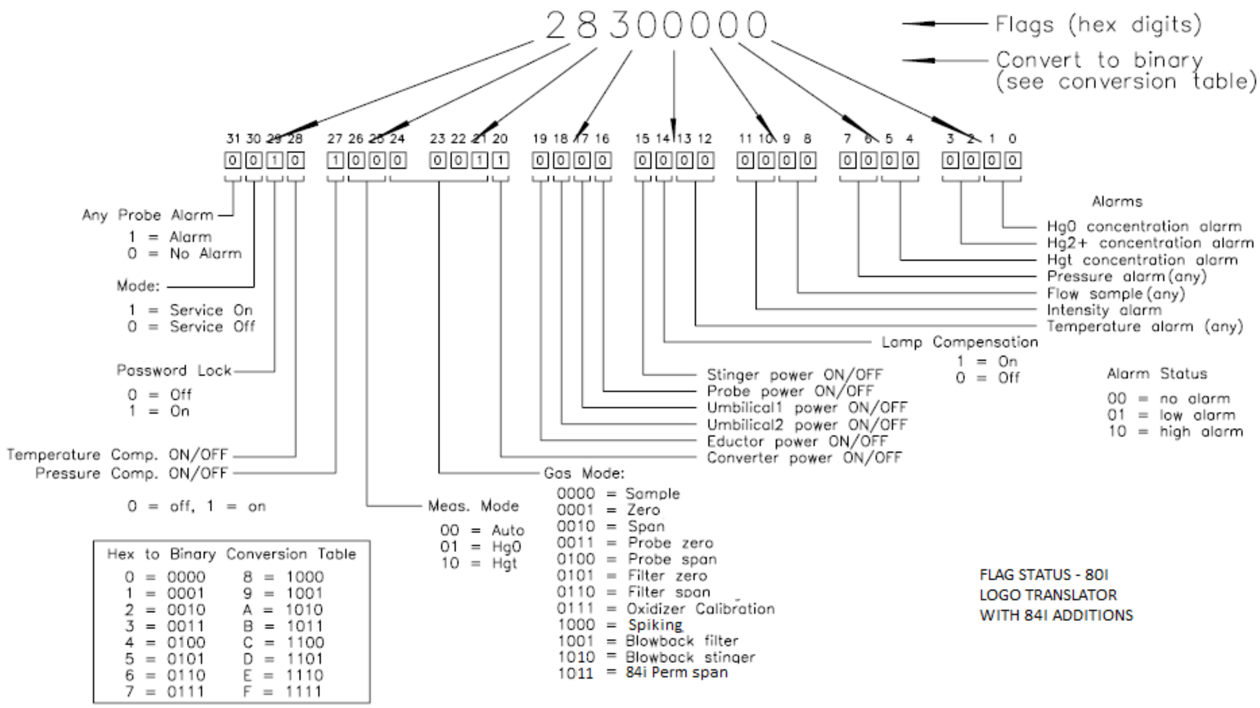

Figure 1–1. Flag Status

Alarms

alarm blow back pres min
alarm blow back pres max

These commands report the blow back pressure alarm minimum and maximum value current settings. The example that follows reports that the blow back pressure alarm minimum value is 3.0 psig.

```
Send:        alarm blow back pres min
Receive:     alarm blow back pres min 3.0 psig
```

set alarm blow back pres min *value*
set alarm blow back pres max *value*

These commands set the blow back pressure alarm minimum and maximum values to *value*, where *value* is a floating-point number representing blow back pressure alarm limits in psig. The example that follows sets the blow back pressure alarm maximum value to 65 psig.

```
Send:        set alarm blow back pres max 65
Receive:     set alarm blow back pres max 65 ok
```

alarm chamber temp min
alarm chamber temp max

These commands report the chamber temperature alarm minimum and maximum value current settings. The example that follows reports that the chamber temperature alarm minimum value is 47.0 °C.

```
Send:         alarm chamber temp min
Receive:      alarm chamber temp min 47.0 deg C
```

set alarm chamber temp min *value*
set alarm chamber temp max *value*

These commands set the chamber temperature alarm minimum and maximum values to *value*, where *value* is a floating-point number representing chamber temperature alarm limits in degrees C. The example that follows sets the chamber temperature alarm maximum value to 50.0 °C.

```
Send:         set alarm chamber temp max 50.0
Receive:      set alarm chamber temp max 50.0 ok
```

alarm conc hg0 min
alarm conc hg2+ min
alarm conc hgt min
alarm conc hg0max
alarm conc hg2+ max
alarm conc hgt max

These commands report the Hg0, Hg2+, and Hgt concentrations alarm minimum and maximum value current setting. The example that follows reports that the Hg0 concentration minimum is 5.2 $\mu g/m^3$.

```
Send:         alarm conc hg0 min
Receive:      alarm conc hg0 min 5.2 ug/m3
```

set alarm conc hg0 min *value*
set alarm conc hg2+ min *value*
set alarm conc hgt min *value*
set alarm conc hg0 max *value*
set alarm conc hg2+ max *value*
set alarm conc hgt max *value*

These commands set the Hg0, Hg2+, and Hgt concentrations alarm minimum and maximum value to *value*, where *value* is a floating-point representation of the concentration alarm limits. The example that follows sets the Hg0 concentration alarm maximum value to 6.8 $\mu g/m^3$.

```
Send:         set alarm conc hg0 max 6.8
Receive:      set alarm conc hg0 max 6.8 ok
```

alarm converter temp min *probenumber*
alarm converter temp max *probenumber*

These commands report the converter alarm minimum and maximum value settings for a specified probe, where *probenumber* is the number of the specified probe (1-4). If a probe number is not specified, the value of the currently active probe is displayed. The example that follows reports that the converter temperature alarm minimum value is 47.0 °C for probe number 3.

```
Send:        alarm converter temp min 3
Receive:     alarm converter temp min 3 47.0 deg C
```

set alarm converter temp min *value probenumber*
set alarm converter temp max *value probnumber*

These commands set the converter temperature alarm minimum and maximum values for a specified probe number, where *value* is a floating-point number representing converter temperature alarm limits in degrees C, and *probenumber* is the number of the specified probe (1-4). If a probe number is not specified, the value is applied to the currently active probe.

The example that follows sets the converter temperature alarm maximum value to 50.5 °C for probe number 2.

```
Send:        set alarm converter temp max 50.5 2
Receive:     set alarm converter temp max 50.5 2 ok
```

alarm cooler temp min
alarm cooler temp max

These commands report the lamp alarm minimum and maximum value current settings. The example that follows reports that the lamp temperature alarm minimum value is minus 10.0 °C.

```
Send:        alarm cooler temp min
Receive:     alarm cooler temp min -10.0 deg C
```

set alarm cooler temp min *value*
set alarm cooler temp max *value*

These commands set the lamp temperature alarm minimum and maximum values to *value*, where *value* is a floating-point number representing lamp temperature alarm limits in degrees C. The example that follows sets the lamp temperature alarm maximum value to minus 1.0 °C.

```
Send:        set alarm cooler temp max -1.0
Receive:     set alarm cooler temp max -1.0 ok
```

alarm dilution pres min *probenumber*
alarm dilution pres max *probenumber*

These commands report the dilution pressure alarm minimum and maximum value settings for a specified probe, where *probenumber* is the number of the specified probe (1-4). If a probe number is not specified, the value of the currently active probe is displayed. The example that follows reports that the dilution pressure alarm minimum value is 40.0 psig for probe number 2.

```
Send:         alarm dilution pres min 2
Receive:      alarm dilution pres min 2 40.0 psig
```

set alarm dilution pres min *value probenumber*
set alarm dilution pres max *value probenumber*
These commands set the dilution pressure alarm minimum and maximum values to *value*, where *value* is a floating-point number representing dilution pressure alarm limits in psig, and *probenumber* is the number of the specified probe (1-4). If a probe number is not specified, the value is applied to the currently active probe.

The example that follows sets the dilution pressure alarm maximum value to 65.0 psig for probe number 2.

```
Send:         set alarm dilution pres max 65.0 2
Receive:      set alarm dilution pres max 65.0 2 ok
```

alarm eductor pres min *probenumber*
alarm eductor pres max *probenumber*
These commands report the eductor pressure alarm minimum and maximum value settings for a specified probe, where *probenumber* is the number of the specified probe (1-4). If a probe number is not specified, the value of the currently active probe is displayed. The example that follows reports that the eductor pressure alarm minimum value is 5.0 psig for probe number 3.

```
Send:         alarm eductor pres min 3
Receive:      alarm eductor pres min 3 5.0 psig
```

set alarm eductor pres min *value probenumber*
set alarm eductor pres max *value probenumber*
These commands set the eductor pressure alarm minimum and maximum values to *value*, where *value* is a floating-point number representing dilution pressure alarm limits in psig, and *probenumber* is the number of the specified probe (1-4). If a probe number is not specified, the value is applied to the currently active probe.

The example that follows sets the eductor pressure alarm maximum value to 20.0 psig for probe number 2.

```
Send:        set alarm eductor pres max 20.0 2
Receive:     set alarm eductor pres max 20.0 2 ok
```

alarm flow min
alarm flow max

These commands report the sample flow alarm minimum and maximum value current settings. The example that follows reports that the sample flow alarm minimum value is 0.300 lpm.

```
Send:        alarm flow min
Receive:     alarm flow min 0.300 lpm
```

set alarm flow min *value*
set alarm flow max *value*

These commands set the sample flow alarm minimum and maximum values to *value*, where *value* is a floating-point number representing sample flow limits in psig. The example that follows sets the sample flow alarm maximum value to 0.600 lpm.

```
Send:        set alarm flow max 0.600
Receive:     set alarm flow max 0.600 ok
```

alarm internal temp min
alarm internal temp max

These commands report the internal temperature alarm minimum and maximum value settings. The example that follows reports that the internal temperature alarm minimum value is 15.0 °C.

```
Send:        alarm internal temp min
Receive:     alarm internal temp min 15.0 deg C
```

set alarm internal temp min *value*
set alarm internal temp max *value*

These commands set the internal temperature alarm minimum and maximum values to *value*, where *value* is a floating-point number representing internal temperature alarm limits in degrees C. The example that follows sets the internal temperature alarm maximum value to 45.0 °C.

```
Send:        set alarm internal temp max 45
Receive:     set alarm internal temp max 45 ok
```

alarm pressure min
alarm pressure max

These commands report the pressure alarm minimum and maximum value current settings. The example that follows reports that the pressure alarm minimum value is 20.0 mmHg.

```
Send:         alarm pressure min
Receive:      alarm pressure min 20.0 mm Hg
```

set alarm pressure min *value*
set alarm pressure max *value*
These commands set the pressure alarm minimum and maximum values to *value*, where *value* is a floating-point number representing pressure alarm limits in millimeters of mercury. The example that follows sets the pressure alarm maximum value to 80 mmHg.

```
Send:         set alarm pressure max 80
Receive:      set alarm pressure max 80 ok
```

alarm probe temp min *probenumber*
alarm probe temp max *probenumber*
These commands report the probe alarm minimum and maximum value settings for a specified probe, where *probenumber* is the number of the specified probe (1-4). If a probe number is not specified, the value of the currently active probe is displayed. The example that follows reports that the probe temperature alarm minimum value is 3.0 °C for probe number 3.

```
Send:         alarm probe temp min 3
Receive:      alarm probe temp min 3 3.0 deg C
```

set alarm probe temp min *value probenumber*
set alarm probe temp max *value probenumber*
These commands set the probe temperature alarm minimum and maximum values for a specified probe number, where *value* is a floating-point number representing probe temperature alarm limits in degrees C, and *probenumber* is the number of the specified probe (1-4). If a probe number is not specified, the value is applied to the currently active probe.

The example that follows sets the probe temperature alarm maximum value to 230 °C for probe number 2.

```
Send:         set alarm probe temp max 230.0 2
Receive:      set alarm probe temp max 230.0 2 ok
```

alarm trig conc hg0
alarm trig conc hg2+
alarm trig conc hgt
These commands report the Hg0, Hg2+, and Hgt concentrations alarm trigger action for minimum alarm, current setting, to either floor or ceiling. The example that follows shows the Hg0 concentration minimum alarm trigger to ceiling, according to the following table.

```
Send:        alarm trig conc hg0
Receive:     alarm trig conc hg0 1
```

set alarm trig conc hg0 *value*
set alarm trig conc hg2+ *value*
set alarm trig conc hgt *value*
These commands set the Hg0, Hg2+, and Hgt concentrations alarm minimum value, where *value* is set to either floor or ceiling, according to the following table. The example that follows sets the Hg0 concentration minimum alarm trigger to ceiling.

```
Send:        set alarm trig conc hg0 1
Receive:     set alarm trig conc hg0 1 ok
```

Table 1–4. Alarm Trigger Values

Value	**Alarm Trigger**
00	Floor
01	Ceiling

alarm umbilical temp min *probenumber*
alarm umbilical temp max *probenumber*
These commands report the umbilical alarm minimum and maximum value settings for a specified probe, where *probenumber* is the number of the specified probe (1-4). If a probe number is not specified, the value of the currently active probe is displayed. The example that follows reports that the umbilical temperature alarm minimum value is 70.0 °C for probe number 1.

```
Send:        alarm umbilical temp min 1
Receive:     alarm umbilical temp min 1 70.0 deg C
```

set alarm umbilical temp min *value probenumber*
set alarm umbilical temp max *value probenumber*
These commands set the umbilical temperature alarm minimum and maximum values for a specified probe number, where *value* is a floating-point number representing probe temperature alarm limits in degrees C, and *probenumber* is the number of the specified probe (1-4). If a probe number is not specified, the value is applied to the currently active probe.

The example that follows sets the umbilical temperature alarm maximum value to 190 °C for probe number 2.

```
Send:        set alarm umbilical temp max 190.0 2
Receive:     set alarm umbilical temp max 190.0 2 ok
```

alarm vacuum pres min *probenumber*
alarm vacuum pres max *probenumber*
These commands report the vacuum pressure alarm minimum and maximum value settings for a specified probe, where *probenumber* is the number of the specified probe (1-4). If a probe number is not specified, the value of the currently active probe is displayed. The example that follows reports that the vacuum pressure alarm minimum value is 19 in Hg for probe number 3.

```
Send:         alarm vacuum pres min 3
Receive:      alarm vacuum pres min 3 19.0 inhg
```

set alarm vacuum pres min *value probenumber*
set alarm vacuum pres max *value probenumber*
These commands set the vacuum pressure alarm minimum and maximum values for a specified probe number, where *value* is a floating-point number representing vacuum pressure alarm limits in psig, and *probenumber* is the number of the specified probe (1-4). If a probe number is not specified, the value is applied to the currently active probe.

The example that follows sets the vacuum pressure alarm maximum value to 25 in Hg for probe number 2.

```
Send:         set alarm vacuum pres max 25 2
Receive:      set alarm vacuum pres max 25 2 ok
```

Diagnostics

diag volt mb
This command reports the diagnostic voltage measurements on the motherboard. The sequence of voltages is: positive 24, positive 15, positive 5, positive 3.3, and negative 3.3. Each voltage value is separated by a space.

```
Send:         diag volt mb
Receive:      diag volt mb 24.1 14.9 4.9 3.2 -3.2
```

diag volt mib
This command reports the diagnostic voltage measurements on the measurement interface board. The sequence of voltages is: positive 24, positive 15, negative 15, positive 5, positive 3.3, and positive 15. Each voltage value is separated by a space.

```
Send:         diag volt mib
Receive:      diag volt mb 24.1 14.9 -14.9 4.9 3.2 14.9
```

diag volt iob
This command reports the diagnostic voltage measurements on the optional I/O expansion board. The sequence of voltages is: positive 24,

positive 5, positive 3.3, and negative 3.3. Each voltage value is separated by a space.

```
Send:         diag volt iob
Receive:      diag volt iob 24.1 4.9 3.2 -3.2
```

diag volt probe

This command reports the diagnostic voltage level measurements on the 82i measurement interface board. The sequence of voltages is: positive 24, positive 15, negative 15, positive 5, positive 3.3, and positive 15. Each voltage value is separated by a space.

```
Send:         diag volt probe
Receive:      diag volt probe 24.1 14.9 -14.9 4.9 3.2 14.9
```

Datalogging

clr records

This command will clear all long and short records that have been saved.

```
Send:         clr records
Receive:      clr records ok
```

set clr lrecs
set clr srecs

These commands will clear only the long records or only the short records that have been saved. The example that follows clears short records.

```
Send:         set clr srecs 1
Receive:      set clr srecs 1 ok
```

list din
list dout

These commands report the current selection for the digital outputs in the format: Output no Index number variable name active state. The active state for digital outputs is open or closed. The active state for digital inputs is high or low.

```
Send:         list dout
Receive:      list dout
output index variable state
1  2  HGO MODE open
2  3  HGT MODE open
3  4  HGO/HGT MODE open
4  5  SAMPLE MODE open
5  6  INST ZERO MODE open
```

list lrec
list srec
list stream
list sp

These commands report the list of current selections for long record logging data, short record logging data, streaming data output, or the scratch pad (sp) list. The example that follows shows the list for streaming data output.

The scratch pad is a temporary memory area which is used to set up lists of selections for lrec, srec, or streaming data items. The user can copy any of these lists to the scratch pad, modify individual elements in the list, then save the scratch pad back to the original list. Refer to the "sp field" command for information on how to edit the scratch pad.

```
Send:         list stream
Receive:      list stream
field index variable
 x   x time
 1   1 hg0
 2   2 hg2
 3   3 hgt
 4  30 dilf
```

---

er *xy*
lr *xy*
sr *xy*

x = | 0 | 1 | : Reply termination format (see "set format format" command)

y = | 0 | 1 | 2 | : Output format (see "set erec/lrec/srec format format" command)

These commands report the last long and short records stored or the dynamic data record. In the example that follows, the command requests a long record with no checksum, in ASCII format with text. For details on how to decode the flag fields within these records, see **Figure 1–1** in the "flags" command.

```
Send:         lr01
Receive:      lr01
14:21 04-13-07  flags 5B2526 hg0 15.363 hg2 -1.327 hgt 14.035 rfint
5713.950 intt 33.522 rctt 44.908 prbt 204.762 cnvt 799.621 umbt
161.447 vntp 11.251 orfp 0.472 dilp 45.258 bbkp 20.805 edup 18.668 vac
21.338 smplf 0.369 pmtv 799.201 pres 41.646 dilf 30.000 hg81 15.000
obkg 1.288 tblg 1.106 ocoef 1.104 tcoef 0.860 hg0dic 31.483 hg0dit
503.438 hgtdic 86.136 hgtdit 188.385 hg0dsc 112.080 hg0dst 148.926
lampt 43.372 oxyt 0.000
```

erec

This command returns a snapshot of the main operating conditions (measurements and status) at the time the command is issued. The example that follows shows a typical response.

The format is defined by the current settings of the "format" and "erec format" commands. For details on erec formatting, see the "Record Layout Definition" section at the end of this appendix. For details on how to decode the flag fields within these records, see **Figure 1–1** in the "flags" command.

```
Send:          erec
Receive:       erec
14:24 04-13-07 flags 5B2526 hg0 0.000 hgt 0.000 hg2+ 0.000  1 lohg0
15.380 lohgt 14.020 lohg2+ -1.360  1 pmtv 799.201 tempal 1 pres 47.938
flow 0.276 hiavgtime 60 loavgtime 60 hg0bkg 1.288 hgtbkg 1.106 hg0coef
1.000 hgtcoef 1.000 hg2+coef 1.000 lohg0coef 1.104 lohgtcoef 0.860
lohg2+coef 1.000 intt 33.498 chmbt 44.908 prbtmp 204.762 umbtmp
159.084 cnvtmp 795.684 venpr 11.232 orfpr 0.464 dilpr 45.258 blbpr
20.805 edupr 18.634 vacpr 21.327 dilfac 30.000 refint 5715.000 prbidx
1
```

lrec
srec
lrec *xxxx yy*
srec xxxx yy
lrec *aa:bb oo-pp-qq yy*
srec *aa:bb oo-pp-qq yy*
xxxx = the number of past records
yy = the number of records to return (1 to 10)
aa = hours (01 to 24)
bb = minutes (01 to 59)
oo = month (01 to 12)
pp = day (01 to 31)
qq = year

These commands output long or short records and dynamic data. The output format is determined by the "set lrec format", and "set srec format" commands. The logging time is determined by the "set lrec per" and "set srec per" commands. In Hg0 or Hgt only mode, the pertinent high value used, other concentrations are set to 0. Concentrations are stored in $\mu g/m^3$.

When the command lrec 100 2 is sent, the instrument counts back 100 records from the last record collected, and then returns 2 records. For details on how to decode the flag fields within these records, see **Error! Reference source not found.** in the "flags" command.

```
Send:          lrec 100 5
Receive:       lrec 100 5
```

```
12:46 04-13-07  flags 5B2784 hg0 2.922 hg2 -0.224 hgt 2.698 rfint
5730.030 intt 33.665 rctt 44.967 prbt 204.499 cnvt 801.197 umbt
159.084 vntp 11.232 orfp 0.464 dilp 45.258 bbkp 20.805 edup 18.634 vac
21.338 smplf 0.354 pmtv 799.201 pres 47.339 dilf 30.000 hg81 3.000
obkg 1.288 tblg 1.106 ocoef 1.104 tcoef 0.860 hg0dic 31.483 hg0dit
503.438 hgtdic 86.136 hgtdit 188.385 hg0dsc 112.080 hg0dst 148.926
lampt 43.377 oxyt 0.000
12:47 04-13-07  flags 5B2504 hg0 2.925 hg2 -0.231 hgt 2.694 rfint
5727.140 intt 33.665 rctt 45.062 prbt 204.499 cnvt 798.834 umbt
160.134 vntp 11.232 orfp 0.470 dilp 45.258 bbkp 20.805 edup 18.600 vac
21.348 smplf 0.415 pmtv 799.201 pres 41.347 dilf 30.000 hg81 3.000
obkg 1.288 tblg 1.106 ocoef 1.104 tcoef 0.860 hg0dic 31.483 hg0dit
503.438 hgtdic 86.136 hgtdit 188.385 hg0dsc 112.080 hg0dst 148.926
lampt 43.370 oxyt 0.000
```

erec format
lrec format
srec format

These commands report the output format for long and short records, and dynamic data in various formats such as ASCII without text, ASCII with text, or binary. The example that follows shows the output format for long records is ASCII with text, according to the following table.

```
Send:         lrec format
Receive:      lrec format 1
```

set erec format
set lrec format
set srec format

These commands set the output format for long and short records, and dynamic data, according to the following table. The example that follows sets the long record output format to ASCII with text.

```
Send:         set lrec format 1
Receive:      set lrec format 1 ok
```

Table 1–5. Record Output Formats

Format	**Output Format**
0	ASCII no text
1	ASCII with text
2	binary data

erec layout
lrec layout
srec layout

These commands reports the layout (string indicating the data formats) for data that is sent out in response to the erec, lrec, srec, and related commands The example that follows shows a typical response. For details on how to interpret the strings, see “Record Layout Definition” later in this appendix.

```
Send:          lrec layout
Receive:       lrec layout
lrec layout %s %s %lx %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f
%f %f %f %f %f %f %f %f %f %f %f %f %f %f %f %f
t D L f f f f f f f f f f f f f f f f f f f f f f f f f f f f f f f f
flags hg0 hg2 hgt rfint intt rctt prbt cnvt umbt vntp orfp dilp bbkp
edup vac smplf pmtv pres dilf hg81 obkg tblg ocoef tcoef hg0dic hg0dit
hgtdic hgtdit hg0dsc hg0dst lampt oxyt
```

lrec mem size
srec mem size
These commands report the number of lrecs and srecs that can be stored with the current settings and the number of blocks reserved for lrecs and srecs. The example that follows shows that 1075 blocks were reserved for lrecs and the maximum number of lrecs that can be stored in memory is 241979. Memory allocation can be changed using the malloc command.

```
Send:          lrec mem size
Receive:       lrec mem size 241979 recs, 1075 blocks
```

lrec per
srec per
These commands report the long and short records logging period. The example that follows shows that the short record logging period is 5 minutes.

```
Send:          srec per
Receive:       srec per 5 min
```

set lrec per *value*
set srec per *value*
value = | 1 | 5 | 15 | 30 | 60 |

These commands set the long and short records logging period to *value* in minutes. The example that follows sets the long record logging period to 15 minutes.

```
Send:          set lrec per 15
Receive:       set lrec per 15 ok
```

no of lrec
no of srec
These commands report the number of long and short records stored in the long and short records memory. The example that follows shows that 50 long records have been stored in the memory.

```
Send:         no of lrec
Receive:      no of lrec 50 recs
```

malloc lrec
malloc srec
These commands report the currently set memory allocation for long and short records in percent of total memory.

```
Send:         malloc lrec
Receive:      malloc lrec 70%
```

set malloc lrec *value*
set malloc srec *value*
value = 0 to 100

These commands set the percent of memory space allocated for long and short records to *value*, where *value* is a floating-point number representing percent. The example that follows sets the memory allocation for long records to 70.

Note Issuing these commands will clear all the logging data memory. All the existing records should be retrieved using appropriate commands, if required. ▲

```
Send:         set malloc lrec 70
Receive:      set malloc lrec 70 ok
```

set copy sp to lrec
set copy sp to srec
set copy sp to stream
These commands copy the current selections in scratch pad (sp) into the long record, short record, or streaming data list. The example that follows copies the current list in scratch pad into the long records list.

The scratch pad is a temporary memory area which is used to set up lists of selections for lrec, srec, or streaming data items. The user can copy any of these lists to the scratch pad, modify individual elements in the list, then save the scratch pad back to the original list. Refer to the "sp field" command for information on how to edit the scratch pad.

```
Send:         set copy sp to lrec
Receive:      set copy sp to lrec ok
```

set copy lrec to sp
set copy srec to sp
set copy stream to sp

These commands copy the current contents of the long record, short record, or streaming data list into the scratch pad (sp). These commands are useful in easy modification of current long record, short record, or streaming data lists. The example that follows copies the current list of long records into the scratch pad.

The scratch pad is a temporary memory area which is used to set up lists of selections for lrec, srec, or streaming data items. The user can copy any of these lists to the scratch pad, modify individual elements in the list, then save the scratch pad back to the original list. Refer to the "sp field" command for information on how to edit the scratch pad.

```
Send:        set copy lrec to sp
Receive:     set copy lrec to sp ok
```

---

sp field *number*

This command reports the variable *number* and name stored at index in the scratch pad list. The example that follows shows that the field 5 in the scratch pad is set to index number 21, which is for the pressure.

The scratch pad is a temporary memory area which is used to set up lists of selections for lrec, srec, or streaming data items. The user can copy any of these lists to the scratch pad, modify individual elements in the list, then save the scratch pad back to the original list. Refer to the "sp field" command for information on how to edit the scratch pad.

```
Send:        sp field 5
Receive:     sp field 5 21 pres
```

sp field *number value*

number = 1-32 is the maximum number of fields in long and short record lists.

number = 1-18 is for streaming data lists.

This command sets the scratch pad field *number* (item number in scratch pad list) to *value*, where *value* is the index number of a variable in the analog out variable list. Available variables and their corresponding index numbers may be obtained using the command "list var aout". The "set sp field" command is used to create a list of variables which can then be transferred into the long record, short record, or streaming data lists, using the "set copy sp to lrec", "set copy sp to srec", or "set copy sp to stream" commands.

```
Send:        set sp field 1 34
Receive:     set sp field 1 34 ok
```

stream per
This command reports the currently set time interval in seconds for streaming data.

```
Send:         stream per
Receive:      stream per 10 sec
```

set stream per *numbervalue*
numbervalue = | 1 | 2 | 5 | 10 | 20 | 30 | 60 | 90 |120 | 180 | 240 | 300 |
This command sets the time interval between two consecutive streaming data strings to *numbervalue* in seconds. The example that follows sets the number value to 10 seconds.

```
Send:         set stream per 10
Receive:      set stream per 10 ok
```

stream time
This command reports if the streaming data string will have a time stamp attached to it or not, according to the following table.

```
Send:         stream time
Receive:      stream time 1
```

set stream time *value*
This command enables *value*, where *value* is to attach or disable time stamp to streaming data string, according to the following table. The example that follows attaches a time stamp to streaming data.

```
Send:         set stream time 0
Receive:      set stream time 0 ok
```

Table 1–6. Stream Time Values

Value	Stream Time
00	Attaches time stamp to streaming data string
01	Disables time stamp to streaming data string

Calibration

set cal hg0 coef
set cal hgt coef
These commands will auto-calibrate the Hg0 and Hgt coefficients based on the Hg0 and Hgt span gas concentrations. The example that follows shows a successful auto-calibration of the Hg0 coefficient.

```
Send:         set cal hg0 coef
Receive:      set cal hg0 coef ok
```

set cal hg0 bkg
set cal hgt bkg
These commands will auto-calibrate the Hg0 and Hgt backgrounds. If the instrument is set to manual Hgt mode, the response to "set cal Hg0 bkg" will be "can't, wrong settings". The example that follows shows a successful auto-calibration of the Hg0 background.

```
Send:         set cal hg0 bkg
Receive:      set cal hg0 bkg ok
```

hg0 coef
hgt coef
These commands report the Hg0 and Hgt coefficients. The example that follows reports that the Hg0 coefficient is 1.000.

```
Send:         hg0 coef
Receive:      hg0 coef 1.000
```

set hg0 coef *value*
set hgt coef *value*
These commands set the Hg0 and Hgt coefficients to user-defined values, where *value* is a floating-point representation of the coefficient. The example that follows sets the Hg0 coefficient to 1.005.

```
Send:         set hg0 coef 1.005
Receive:      set hg0 coef 1.005 ok
```

hg0 gas
hgt gas
These commands report the Hg0 and Hgt span gas concentrations used to auto-calibrate Hg0 and Hgt coefficients. The example that follows shows that the Hg0 low span gas concentration is 10.0 μg/m^3.

```
Send:         hg0 gas
Receive:      hg0 gas 1.oooE+01 ug/m3
```

set hg0 gas *value*
set hgt gas *value*
These commands set the Hg0 and Hgt span gas concentrations used by the auto-calibration routine to *value*, where *value* is a floating-point representation of the gas concentration in current selected units. The gas units are the same as those chosen by the user. The example that follows sets the Hg0 span gas concentration to 15.0 μg/m^3.

```
Send:         set hg0 gas 15.0
Receive:      set hg0 gas 15.0 ok
```

hg0 bkg
hgt bkg
These commands report the current Hg0 and Hgt backgrounds. The example that follows reports that the Hg0 background is 1.3 μg/m^3.

```
Send:        hg0 bkg
Receive:     hg0 bkg 1.300E+00 ug/m3
```

set hg0 bkg *value*
set hgt bkg *value*
These commands are used to set Hg0 and Hgt backgrounds to user-defined *value*, where *value* is a floating-point representation of the background in current selected units. The example that follows sets the Hg0 background to 2.0 μg/m^3.

```
Send:        set hg0 bkg 2.0
Receive:     set hg0 bkg 2.0 ok
```

pres cal
This command reports the pressure recorded at the time of calibration. The example that follows shows that the pressure at calibration is 150.0 mmHg.

```
Send:        pres cal
Receive:     pres cal 150.0 mm Hg
```

set pres cal
This command automatically sets the current pressure as the calibration pressure. The example that follows successfully sets the calibration pressure to 120.5 mmHg.

```
Send:        set pres cal 120.5
Receive:     set pres cal 120.5 ok
```

Keys/Display

push *button*
button = | do | down | en | enter | he | help | le | left | me | menu | ri | right | ru | run | up | 1 | 2 | 3 | 4 |
These commands simulates pressing the front panel pushbuttons. The numbers represent the front-panel soft keys, from left to right.

```
Send:        push enter
Receive:     push enter ok
```

isc
iscreen
This command retrieves the framebuffer data used for the display on the *i*Series instrument. It is 19200 bytes in size, 2-bits per pixel, 4 pixels per byte arranged as 320 by 240 characters. The data is sent in RLE encoded

form to save time in transmission. It is sent as a type '5' binary c_link response with no checksum.

The RLE encoding consists of a 0 followed by an 8-bit count of consecutive 0xFF bytes. The following 'c' code will expand the incoming data.

```
void      unpackDisplay ( void far* tdib, unsigned char far* rlescreen )
{
int i,j,k;
unsigned char far *sc4bpp, *sc2bpp, *screen, *ptr;

   ptr = screen = (unsigned char far *)malloc(19200);
   //RLE decode the screen
   for (i=0; i<19200 && (ptr - screen) < 19200; i++)
   {
      *(ptr++) = *(rlescreen + i);
      if (*(rlescreen + i) == 0)
      {
         unsigned char rlecount = *(unsigned char *)(rlescreen + ++i);

         while (rlecount)
         {
            *(ptr++) = 0;
            rlecount--;
         }
      }
      else if (*(rlescreen + i) == 0xff)
      {
         unsigned char rlecount = *(unsigned char *)(rlescreen + ++i);

         while (rlecount)
         {
            *(ptr++) = 0xff;
            rlecount--;
         }
      }
   }
}
```

To convert this data into a BMP for use with windows, it needs to be turned into a 4BPP as that is the smallest windows can display. Also note that BMP files are upside down relative to this data, i.e. the top display line is the last line in the BMP.

sc
screen
This command is meant for backward compatibility on the C series. Screen information is reported using the "iScreen" command described previously.

```
Send:         screen
Receive:      screen This is an I series Instrument. Screen
Information not available
```

Measurement Configuration

set bb filter
This command sets the system gas mode to filter. The example that follows sets the system gas mode to filter.

```
Send:         set bb filter
Receive:      set bb filter ok
```

bb period
These commands report the blow back frequency setting. The example that follows reports the blow back frequency is four hours.

```
Send:         bb period
Receive:      bb period 4 00
```

set bb period min *hr min*
This command sets the blow back frequency to *hours* and *minutes*. The example that follows sets the blow back frequency to 6 hours.

```
Send:         set bb period 6 00
Receive:      set bb period 6 00 ok
```

bb set pres *probenumber*
This command reports the blow back pressure in psig, where *probenumber* is the number of the specified probe (1-4). If a probe number is not specified, the value of the currently active probe is displayed. The example that follows reports that the blow back pressure is 8.0 for probe number 3.

```
Send:         bb set pres 3
Receive:      bb set pres 3 8.0 psig
```

bb set pres counts *probenumber*
This command reports the blow back pressure in counts and *probenumber* is the number of the specified probe (1-4). If a probe number is not specified, the value is applied to the currently active probe.

The example that follows reports the the blow back pressure is 2000 counts.

```
Send:         bb set pres counts
Receive:      bb set pres counts 2000
```

set bb set pres counts *value probenumber*
This command sets the blow back pressure to *value*, where *value* represents blow back pressure in counts and *probenumber* is the number of the specified probe (1-4). If a probe number is not specified, the value is applied to the currently active probe.

The example that follows sets the blow back pressure to 2100 counts for probe 2.

```
Send:         set bb set pres counts 2100 2
Receive:      set bb set pres counts 2100 2 ok
```

set bb stinger
This command sets the system gas mode to blow back stinger. The example that follows sets the system gas mode to blow back stinger.

```
Send:         set bb stinger
Receive:      set bb stinger ok
```

bb stinger duration
This command reports the blow back stinger duration (seconds). The example that follows reports the blow back stinger duration as 15 seconds.

```
Send:         bb stinger duration
Receive:      bb stinger duration 15 sec
```

set bb stinger duration *seconds*
This command sets the blow back stinger duration in seconds. The example that follows sets the blow back stinger duration to 25 seconds.

```
Send:         set bb stinger duration 25
Receive:      set bb stinger duration 25 ok
```

bb system duration
This command reports the blow back filter duration (seconds). The example that follows reports the blow back filter duration is 15 seconds.

```
Send:         bb system duration
Receive:      bb system duration 15 sec
```

set bb system duration *seconds*
This command sets the blow back filter duration in seconds. The example that follows sets the blow back filter duration to 40 seconds.

```
Send:         set bb system duration 40
Receive:      set bb system duration 40 ok
```

bench set temp

This command reports the temperature of the optical bench in degrees C. The example that follows reports the bench temperature is 45.0 degrees C.

```
Send:          bench set temp
Receive:       bench set temp 45.0 deg C
```

set bench set temp *value*

This command sets the optical bench temperature to *value*, where *value* is the temperature in degrees C. The example that follows sets the optical bench temperature to 47 degrees C.

```
Send:          set bench set temp 47
Receive:       set bench set temp 47.0 ok
```

chamber temp

This command reports the chamber temperature in degrees C. The example that follows reports the chamber temperature is 116.6 degrees C.

```
Send:          chamber temp
Receive:       chamber temp 116.6 deg C
```

conv set temp

This command reports the converter set temperature in degrees C. The example that follows reports the converter temperature is 760 degrees C.

```
Send:          conv set temp
Receive:       conv set temp 760.0 deg C
```

set conv set temp *value*

This command sets the converter set temperature to *value*, where *value* is the temperature in degrees C. The example that follows sets the converter temperature to 800 degrees C.

```
Send:          set conv set temp 800
Receive:       set conv set temp 800.0 ok
```

conv temp

This command reports the converter temperature in degrees C. The example that follows reports the converter temperature is 760 degrees C.

```
Send:          conv temp
Receive:       conv temp 760.0 deg C
```

dilution ratio

This command reports the dilution ratio. The example that follows reports the dilution ratio is 40.

```
Send:         dilution ratio
Receive:      dilution ratio 40
```

set dilution ratio
This command sets the dilution ratio. The example that follows sets the dilution ratio to 50.

```
Send:         set dilution ratio 50
Receive:      set dilution ratio 50 ok
```

---

dilution set pres *probenumber*
This command reports the dilution pressure in psig, where *probenumber* is the number of the specified probe (1-4). If a probe number is not specified, the value of the currently active probe is displayed. The example that follows reports the dilution pressure is 40.0 psig for probe 3.

```
Send:         dilution set pres 3
Receive:      dilution set pres 3 40.0 psig
```

---

dilution set pres counts *probenumber*
This command reports the dilution pressure in counts, where *probenumber* is the number of the specified probe (1-4). If a probe number is not specified, the value is applied to the currently active probe. The example that follows reports the dilution pressure is 2050 counts for probe 1.

```
Send:         dilution set pres counts 1
Receive:      dilution set pres counts 1 2050
```

set dilution set pres counts *value probenumber*
This command sets the dilution pressure *value*, where *value* is a number representing dilution pressure in counts and *probenumber* is the number of the specified probe (1-4). If a probe number is not specified, the value is applied to the currently active probe. The example that follows sets the dilution pressure to 2030 counts for probe 4.

```
Send:         set dilution set pres counts 2030 4
Receive:      set dilution set pres counts 2030 4 ok
```

---

eductor set pres *probenumber*
This command reports the eductor pressure in psig, where *probenumber* is the number of the specified probe (1-4). If a probe number is not specified, the value is applied to the currently active probe. The example that follows reports the eductor pressure is 1.6 psig for probe 3.

```
Send:         eductor set pres 3
Receive:      eductor set pres 3 1.6 psig
```

eductor set pres counts *probenumber*

This command reports the eductor pressure in counts, where *probenumber* is the number of the specified probe (1-4). If a probe number is not specified, the value is applied to the currently active probe. The example that follows reports the eductor pressure counts are 2100 for probe 3.

```
Send:        eductor set pres counts 3
Receive:     eductor set pres counts 3 2100
```

set eductor set pres counts *value probenumber*

This command sets the eductor pressure to *value*, where *value* is a number representing eductor pressure in counts and *probenumber* is the number of the specified probe (1-4). If a probe number is not specified, the value is applied to the currently active probe.

The example that follows sets the eductor set pressure to 2150 counts for probe 1.

```
Send:        set eductor set pres counts 2150 1
Receive:     set eductor set pres counts 2150 1 ok
```

lamp comp

This command reports the whether the lamp compensation is on or off. The following example reports that the lamp compensation is on.

```
Send:        lamp comp
Receive:     lamp comp on
```

set lamp comp *on/off*

This command sets the lamp compensation on or off. The example that follows sets the lamp compensation on.

```
Send:        set lamp comp on
Receive:     set lamp comp on ok
```

lamp intensity

This command reports the lamp intensity in Hz. The following example reports that the lamp intensity is 59043 Hz.

```
Send:        lamp intensity
Receive:     lamp intensity 59043.0 Hz
```

cal mode o2 pct

This command reports the cal mode O_2 percent. The example that follows shows that the cal mode O_2 percent is 1.50%.

```
Send:        cal mode o2 pct
Receive:     cal mode o2 pct 1.50%
```

set cal mode o2 pct *selection*
This command sets the cal mode O_2 pct. The example that follows sets the cal mode O_2 percent to 1.60%.

```
Send:       set cal mode o2 pct 1.60
Receive:    set cal mode o2 pct 1.60 ok
```

stack o2 channel
This command returns the current stack O_2 analog input channel.

```
Send:       stack o2 channel
Receive:    stack o2 channel 3
```

set stack o2 channel *X*
This command sets the stack O_2 analog input channel to *X*. The allowed range is 1 to 8.

```
Send:       set stack o2 channel 3
Receive:    set stack o2 channel 3 ok
```

current sample
This command reports the current measurement when in sample mode. The example that follows shows the current measurement is Hgt.

```
Send:       current sample
Receive:    current sample hg(t)
```

duration time hg0
This command reports the Hg0 duration time (during sample mode) in minutes. The example that follows shows that the Hg0 duration time (during sample mode) is 2 minutes.

```
Send:       duration time hg0
Receive:    duration time hg0 2 minutes
```

set duration time hg0 *selection*
This command sets the Hg0 duration time during sample mode. The example that follows sets the Hg0 duration time (during sample mode) to 3 minutes.

```
Send:       set duration time hg0 3
Receive:    set duration time hg0 3 ok
```

duration time hgt
This command reports the Hgt duration time (during sample mode) in minutes. The example that follows shows that the Hgt duration time during sample mode is 28 minutes.

```
Send:        duration time hgt
Receive:     duration time hgt 28 minutes
```

set duration time hgt *selection*
This command sets the Hgt duration time during sample mode. The example that follows sets the Hgt duration time (during sample mode) to 27 minutes.

```
Send:        set duration time hgt 27
Receive:     set duration time hgt 27 ok
```

hg0 next time
This command reports the next time the 80*i* Analyzer will read the Hg0 concentration when in sample mode. The example that follows shows that the Hg0 next time is at 15:00.

```
Send:        hg0 next time
Receive:     hg0 next time 15:00
```

range hg0
range hg2+
range hgt
These commands report the Hg0, Hg2+, and Hgt ranges according to the following table. The example that follows reports that the Hg0 range is 600.0 µg/m³.

```
Send:        range hg0
Receive:     range hg0 0: 6.000E+2 ug/m3
```

set range hg0 *selection*
set range hg2+ *selection*
set range hgt *selection*
These commands select the Hg0, Hg2+, and Hgt full-scale ranges, according to the following table. The example that follows sets the Hgt full-scale to 15.0 µg/m³.

```
Send:        set range hgt 3
Receive:     set range hgt 3 ok
```

Table 1–7. Standard Ranges

Selection	µg/m³
0	1.5
1	3.0
2	6.0
3	15.0

Selection	µg/m³
4	30.0
5	60.0
6	150.0
7	300.0
8	600.0
9	C1
10	C2
11	C3

custom *range*
range = | 1 | 2 | 3 |
This command reports the user-defined value of custom *range* 1, 2, or 3. The example that follows reports that custom range 1 is set to 600.0 µg/m³.

```
Send:          custom 1
Receive:       custom 1 6.000E+2 ug/m3
```

set custom *range value*
range = | 1 | 2 | 3 |
This command is used to set the maximum concentration for any of the three custom ranges 1, 2, or 3 to range *value*, where *value* is a floating-point number representing concentration in µg/m³. The example that follows sets the custom 1 range to 600.0 µg/m³.

```
Send:          set custom 1 600.0
Receive:       set custom 1 600.0 ok
```

range mode
This command reports the current range mode in single, dual, or auto. The example that follows reports the range mode is set to single.

```
Send:          range mode
Receive:       range mode single
```

set range mode *mode*
mode = | single | dual | auto |
This command sets the current range mode to single, dual, or auto. The example that follows sets the range mode to single.

```
Send:          set range mode single
Receive:       set range mode single ok
```

gas mode
This command reports the current mode of sample, zero, or span. The example that follows reports that the gas mode is sample.

```
Send:         gas mode
Receive:      gas mode sample
```

set sample
set zero
set span
These commands set the current gas mode to sample, zero, or span. The example that follows sets the instrument to span mode, that is, the instrument is sampling span gas.

```
Send:         set sample
Receive:      set sample ok
```

span inst
This command reports the instrument span level currently in effect. The example that follows reports the instrument span level is 3.

```
Send:         span inst
Receive:      span inst 3
```

set span inst *level*
This command sets the instrument span level to *level* where *level* is a number from 1-6. This command is used to tell the Model 81*i* which span level to use. The example that follows sets the instrument span level to 2.

```
Send:         set span inst 2
Receive:      set span inst 2 ok
```

span sys
This command reports the system span level currently in effect. The example that follows reports the system span level is 4.

```
Send:         span sys
Receive:      span sys 4
```

set span sys *level*
This command sets the system span level to *level* where *level* is a number from 1-6. This command is used to tell the Model 81*i* which span level to use. The example that follows sets the system span level to 6.

```
Send:         set span sys 6
Receive:      set span sys 6 ok
```

meas mode
This command reports which measurement mode (Hg0/Hgt, Hg0, Hgt) is active. The example that follows reports that the measurement mode is set to Hg0.

```
Send:       meas mode
Receive:    meas mode hg0
```

set meas mode *mode*
mode = | hg0/hgt | hg0 | hgt |
This command sets the instrument to Hg0/Hgt (auto) mode, manual Hg0 mode, or manual Hgt mode. The example that follows sets the instrument to the manual Hg0 mode.

```
Send:       set meas mode hg0
Receive:    set meas mode hg0 ok
```

set s span
This command sets the system gas mode to system span as shown in the following example.

```
Send:       set s span
Receive:    set s span ok
```

set s zero
This command sets the system gas mode to system zero as shown in the following example.

```
Send:       set s zero
Receive:    set s zero ok
```

power converter *probenumber*
This command reports the whether the converter power is on or off for the selected probe. The probe number must be included in the command. The following example reports that the converter power is off for probe 2.

```
Send:       power converter 2
Receive:    power converter 2 off
```

set power converter *on/off probenumber*
This command sets the converter power on or off for the selected probe. The probe number must be included in the command. The example that follows sets the converter power on for probe 1.

```
Send:       set power converter on 1
Receive:    set power converter on 1 ok
```

power eductor *probenumber*
This command reports the whether the eductor power is on or off for the selected probe. The probe number must be included in the command. The following example reports that the eductor power is on for probe 3.

```
Send:         power eductor 3
Receive:      power eductor 3 on
```

set power eductor *on/off probenumber*
This command sets the eductor power on or off for the selected probe. The probe number must be included in the command. The example that follows sets the eductor power on for probe 2.

```
Send:         set power eductor on 2
Receive:      set power eductor on 2 ok
```

power probe *probenumber*
This command reports the whether the probe power is on or off for the selected probe. The probe number must be included in the command. The following example reports that the probe power is off for probe 1.

```
Send:         power probe 1
Receive:      power probe 1 off
```

set power probe *on/off probenumber*
This command sets the probe power on or off for the selected probe. The probe number must be included in the command. The example that follows sets the probe power on for probe 3.

```
Send:         set power probe on 3
Receive:      set power probe on 3 ok
```

power stinger *probenumber*
This command reports the whether the stinger power is on or off for the selected probe. The probe number must be included in the command. The following example reports that the stinger power is on for probe 2.

```
Send:         power stinger 2
Receive:      power stinger 2 on
```

set power stinger *on/off probenumber*
This command sets the stinger power on or off for the selected probe. The probe number must be included in the command. The example that follows sets the stinger power off for probe 1.

```
Send:         set power stinger off 1
Receive:      set power stinger off 1 ok
```

pmt supply

This command reports whether the PMT supply is on or off as shown in the example that follows.

```
Send:         pmt supply
Receive:      pmt supply on
```

set pmt supply *on/off*

This command sets the PMT supply on or off. The example that follows sets the PMT supply on.

```
Send:         set pmt supply on
Receive:      set pmt supply on ok
```

pres comp

This command reports whether pressure compensation is on or off. The example that follows shows that pressure compensation is on.

```
Send:         pres comp
Receive:      pres comp on
```

set pres comp *onoff*

This command turns the pressure compensation on or off. The example that follows turns pressure compensation off.

```
Send:         set pres comp off
Receive:      set pres comp off ok
```

probe no

This command reports the number of the active probe (hydra only). The example that follows shows that probe 3 is the currently active probe.

```
Send:         probe no
Receive:      probe no 3
```

set probe no *probenumber*

This command selects the probe to use (hydra only). The example that follows selects probe 2 to be the active probe.

```
Send:         set probe no 2
Receive:      set probe no 2 ok
```

probe set failsafe temp *probenumber*

This command reports the probe failsafe temperature in degrees C for the specified probe (*probenumber)*. The example that follows reports the probe failsafe temperature is 760 degrees C for probe 1.

```
Send:         probe set failsafe temp 1
Receive:      probe set failsafe temp 1 760.0 deg C
```

set probe set failsafe temp *value probenumber*
This command sets the probe failsafe temperature to *value*, where *value* is the temperature in degrees C and *probenumber* is the number of the specified probe (1-4). The example that follows sets the probe failsafe temperature to 800 degrees C for probe 3.

```
Send:         set probe set failsafe temp 800 3
Receive:      set probe set failsafe temp 800.0 3 ok
```

---

probe set temp *probenumber*
This command reports the probe temperature in degrees C for the specified probe (*probenumber)*. The example that follows reports the probe temperature is 760 degrees C for probe 2.

```
Send:         probe set temp 2
Receive:      probe set temp 2 760.0 deg C
```

set probe set temp *value probenumber*
This command sets the probe temperature to *value*, where *value* is the temperature in degrees C and *probenumber* is the number of the specified probe (1-4). The example that follows sets the probe temperature to 800 degrees C for probe 3.

```
Send:         set probe set temp 800 3
Receive:      set probe set temp 800.0 3 ok
```

---

ref intensity
This command reports the reference intensity in Hz. The following example reports that the lamp intensity is 59061 Hz.

```
Send:         lamp intensity
Receive:      lamp intensity 59061 Hz
```

Hardware Configuration

contrast
This command reports the screen's level of contrast. The example that follows shows the screen contrast is 45%, according to the following table.

```
Send:         contrast
Receive:      contrast 9: 45%
```

set contrast *level*
This command sets the screen's *level* of contrast, according to the following table. The example that follows sets the contrast level to 50%.

```
Send:          set contrast 10
Receive:       set contrast 10 ok
```

Table 1–8. Contrast Levels

Level	**Contrast Level**
0	0%
1	5%
2	10%
3	15%
4	20%
5	25%
6	30%
7	35%
8	40%
9	45%
10	50%
11	55%
12	60%
13	65%
14	70%
15	75%
16	80%
17	85%
18	90%
19	95%
20	100%

date

This command reports the current date. The example that follows reports the date as April 18, 2007.

```
Send:          date
Receive:       date 04-18-07
```

set date *mm-dd-yy*
mm = month
dd = day
yy = year

This command sets the date of the instrument's internal clock. The example that follows sets the date to March 19, 2005.

```
Send:        set date 03-19-05
Receive:     set date 03-19-05 ok
```

---

set default params

This command sets all the parameters to their default values. This does not affect the factory-calibrated parameters.

```
Send:        set default params
Receive:     set default params ok
```

---

save

set save params

These commands store all current parameters in FLASH memory. It is important that this command be sent each time instrument parameters are changed. If changes are not saved, they will be lost in the event of a power failure. The example that follows saves the parameters to FLASH memory.

```
Send:        set save params
Receive:     set save params ok
```

---

time

This command reports the current internal time (24-hour time). The example that follows reports that the internal time is 2:15:30 pm.

```
Send:        time
Receive:     time 14:15:30
```

set time *hh:mm:ss*
hh = hours
mm = minutes
ss = seconds

This command sets the internal clock (24-hour time). The example that follows sets the internal time to 2:15 pm.

Note If seconds are omitted, the seconds default to 00. ▲

```
Send:        set time 14:15
Receive:     set time 14:15 ok
```

Communications Configuration

addr dns
This command reports the TCP/IP address for the domain name server.

```
Send:         addr dns
Receive:      addr dns 192.168.1.1
```

set addr dns address
This command sets the domain name server address, where address consists of four numbers ranging from 0-255 inclusive, separated by ".".

```
Send:         set addr dns 192.168.1.1
Receive:      set addr dns 192.168.1.1 ok
```

addr gw
This command reports the default TCP/IP gateway address.

Note This command cannot be used when DHCP is on. Refer to the DHCP command that follows for additional information. ▲

```
Send:         addr gw
Receive:      addr gw 192.168.1.1
```

set addr gw *address*
This command sets the default gateway *address*, where *address* consists of four numbers ranging from 0-255 inclusive, separated by ".".

```
Send:         set addr gw 192.168.1.1
Receive:      set addr gw 192.168.1.1 ok
```

addr ip
This command reports the IP address of the instrument.

Note This command cannot be used when DHCP is on. Refer to the DHCP command that follows for additional information. ▲

```
Send:         addr ip
Receive:      addr ip 192.168.1.15
```

set addr ip *address*

This command sets the instrument's IP *address*, where *address* consists of four numbers ranging from 0-255 inclusive, separated by ".".

```
Send:          set addr ip 192.168.1.15
Receive:       set addr ip 192.168.1.15 ok
```

addr nm
This command reports the TCP/IP netmask address.

Note This command cannot be used when DHCP is on. Refer to the DHCP command that follows for additional information. ▲

```
Send:          addr nm
Receive:       addr nm 255.255.255.0
```

set addr nm *address*
This command sets the netmask address, where address consists of four numbers ranging from 0-255 inclusive, separated by ".".

```
Send:          set addr nm 255.255.255.0
Receive:       set addr nm 255.255.255.0 ok
```

addr ntp
This command reports the IP address for the NTP time server. See "Network Time Protocol Server" in the "Communications Settings" section of the "Operation" chapter for more information.

```
Send:          addr ntp
Receive:       addr ntp 192.168.1.2
```

set addr ntp *address*
This command sets the NTP time server *address*, where *address* consists of four numbers ranging from 0-255 inclusive, separated by ".".

```
Send:          set addr ntp 192.168.1.2
Receive:       set addr ntp 192.168.1.2 ok
```

baud
This command reports the current baud rate for the serial port (RS232/RS485). The example that follows reports that the current baud rate is 9600.

```
Send:          baud
Receive:       baud 9600
```

set baud *rate*
rate = | 1200 | 2400 | 4800 | 9600 | 19200 | 38400 | 57600 | 115200 |

This command sets the instrument baud *rate*. The example that follows sets the instrument's baud rate to 115200.

Note After the command is sent, the baud rate of the sending device must be changed to agree with the instrument. ▲

```
Send:         set baud 115200
Receive:      set baud 115200 ok
```

dhcp

This command reports the current state of use of DHCP on or off. DHCP is used to assign an IP address to the instrument automatically. The example that follows shows that DHCP is on.

```
Send:         dhcp
Receive:      dhcp on
```

set dhcp *onoff*

This command enables (*on*) and disables (*off*) the DHCP service. When DHCP is set to on, the instrument gets the IP address, the netmask address, and the gateway address from a DHCP server. When DHCP is set to off, the instrument gets these addresses from system memory.

Note When changing the IP address, the netmask address, or the gateway address, you must cycle power to the instrument before the change takes effect. Until you cycle power, the address assigned by the DHCP server will still be used and reported as the current address. ▲

```
Send:         set dhcp on
Receive:      set dhcp on ok
```

format

This command reports the current reply termination format. The example that follows shows that the reply format is 00, which means reply with no checksum, according to the following table.

```
Send:         format
Receive:      format 00
```

set format *format*

This command sets the reply termination *format*, where *format* is set according to the following table. The example that follows sets the reply termination format to checksum.

```
Send:         set format 01
Receive:      set format 01 ok
```

Table 1–9. Reply Termination Formats

Format	Reply Termination
00	<CR>
01	<NL> sum xxxx <CR>

where xxxx = 4 hexadecimal digits that represent the sum of all the characters (bytes) in the message

host name
This command reports the host name string.

```
Send:          host name
Receive:       host name analyzer01
```

set host name *string*
This command sets the host name *string*, where *string* is 1-13 alphanumeric characters.

```
Send:          set host name analyzer01
Receive:       set host name analyzer01 ok
```

instr name
This command reports the instrument name.

```
Send:          instr name
Receive:       instr name
Hg0-Hg2+-HgT Analyzer
Hg0-Hg2+-HgT Analyzer
```

instrument id
This command reports the instrument id.

```
Send:          instrument id
Receive:       instrument id 80
```

set instrument id *value*
This command sets the instrument id to *value*, where *value* is a decimal number between 0 and 127 inclusive.

Note Sending this command via RS-232 or RS-485 will require the host to use the new id for subsequent commands. ▲

```
Send:          set instrument id 50
Receive:       set instrument id 50 ok
```

allow mode cmd

This command reports the current allow mode setting: 1 = allow "set mode local" or "set mode remote" commands; 0 = ignore "set mode local" or "set mode remote" commands. Refer to **Table 1–10.** Allow Mode Command Values. The default value is 0; ignore the commands. The example that follows shows that the instrument is configured to ignore "set mode local" or "set mode remote" commands.

```
Send:         allow mode cmd
Receive:      allow mode cmd 0
```

set allow mode cmd *value*

This command is used to configure the instrument to *value,* where *value* is either 1 = accept or 0 = ignore the "set mode local" or "set mode remote" commands. Refer to **Table 1–10.** Allow Mode Command Values.

If the instrument is set to accept the commands (*value* = 1), the "set mode local" command will unlock the instrument and the keypad can be used to make changes via the front panel.

If the instrument is set to ignore the commands (*value* = 0), the instrument will respond with "ok" as if the command has been accepted and acted upon, **but will not change the instrument lock status** (this is for compatibility with systems expecting an "ok" response).

Note The instrument will always respond to the command "mode" with the status of the password lock as "mode local" or "mode remote" regardless of the above setting. ▲

The example that follows sets the instrument to accept the "set mode local" or "set mode remote" commands.

```
Send:         set allow mode cmd 1
Receive:      set allow mode cmd 1 ok
```

Table 1–10. Allow Mode Command Values

Value	**Allow Mode Command**
0	Ignore (default)
1	Accept

mode

This command reports what operating mode the instrument is in: local, service, or remote. The example that follows shows that the instrument is in the remote mode.

```
Send:        mode
Receive:     mode remote
```

set mode local
set mode remote

These commands set the instrument to local or remote mode. The example that follows sets the instrument to the local mode.

```
Send:        set mode local
Receive:     set mode local ok
```

power up mode

This command reports the current power up mode setting, where *value,* is either 0 = local/unlocked or 1 = remote/locked, as shown in the following table. The default value is 0; power up in local/unlocked mode. The example that follows shows that the instrument is configured to power up in the remote/locked mode.

```
Send:        power up mode
Receive:     power up mode 1
```

set power up mode *value*

This command is used to configure the instrument to power up in the local/unlocked mode (*value* = 0) or the remote/locked mode (*value* = 1), as shown in the following table.

If the instrument is set to power up in the local/unlocked mode, the keypad can be used to make changes via the front panel. If the instrument is set to power up in the remote/locked mode, changes can not be made from the front panel. The example that follows sets the instrument to power up in remote/locked mode.

```
Send:        set power up mode 1
Receive:     set power up mode 1 ok
```

Table 1–11. Power Up Mode Command Values

Value	Power Up Mode Command
0	Local/Unlocked (default)
1	Remote/Locked Mode

program no

This command reports the instrument's model information and program version number, which depends on the current version.

```
Send:        program no
Receive:     program no iSeries 80i 00.05.68.192
```

set layout ack *value*

This command disables the stale layout/layout change indicator (*) that is attached to each response if the erec layout has changed since the last time erec layout was requested, where *value* represents the function. Refer to **Table 1–12**.

```
Send:        set layout ack 0
Receive:     set layout ack 0 ok
```

Table 1–12. Set Layout Ack Values

Value	Function
0	Do nothing (default)
1	Append "*"

tz

This command reports the "tz" timezone string for the NTP server. See "Network Time Protocol Server" in the "Communications Settings" section of the "Operation" chapter for more information.

```
Send:        tz
Receive:     tz EST+5EDT
```

set tz *string*

This command sets the timezone *string* for the instrument for use with the NTP time server, where *string* is a standard timezone string. Common strings are listed in the timezone screen description in "Chapter 3."

```
Send:        set tz EST+5EDT
Receive:     set tz EST +5 EDT ok
```

I/O Configuration

analog iout range *channel*

This command reports the analog current output range setting for *channels*, where *channel* must be between 1 and 6, inclusive. The example that follows reports current output channel 4 to the 4-20 mA range, according to the following table. This command responds with "feature not enabled" if the optional I/O expansion board is not detected.

```
Send:         analog iout range 4
Receive:      analog iout range 4 2
```

set analog iout range *channel range*
This command sets analog current output *channel* to the *channel range* where *channel* is between 1 and 6 inclusive and range is set according to the following table. The example that follows sets current output channel 4 to the 0-20 mA range. This command responds with "feature not enabled" if the optional I/O expansion board is not detected.

```
Send:         set analog iout range 4 1
Receive:      set analog iout range 4 1 ok
```

Table 1–13. Analog Current Output Range Values

Range	Output Range
1	0-20 mA
2	4-20 mA
0 [cannot be set to this, but may report]	Undefined

analog vin *channel*
This command retrieves the analog voltage input *channel* data, both the calculated value and the actual voltage. In the example that follows, the "calculated" value of channel 1 is 75.325 degrees F, volts are 2.796. This command responds with "feature not enabled" if the optional I/O expansion board is not detected.

```
Send:         analog vin 1
Receive:      analog vin 1 75.325 2.796
```

analog vout range *channel*
This command reports the analog voltage output *channel* range, where *channel* is between 1 and 6 inclusive, according to the following table.

```
Send:         analog vout range 2
Receive:      analog vout range 2 3
```

set analog vout range *channel range*
This command sets analog voltage output *channel* to the range, where *channel* is between 1 and 6 inclusive, and *range* is set according to the following table. The example that follows sets channel 2 to the 0-10 V range.

```
Send:         set analog vout range 2 3
Receive:      set analog vout range 2 3 ok
```

Table 1–14. Analog Voltage Output Range Values

Range	Output Range
1	0-1 V
2	0-100 mV
3	0-10 V
4	0-5 V
0 [cannot be set to this, but may report]	Undefined

dig in

This command reports the status of the digital inputs as a 4-digit hexadecimal string with the most significant bit (MSB) being input 16.

```
Send:         dig in
Receive:      dig in 0xff7f
```

din *channel*

This command reports the action assigned to input *channel* and the corresponding active state. The example that follows reports the input 5 to be assigned an index number 5 corresponding to action of "instrument span mode" with the active state being high.

```
Send:         din 5
Receive:      din 5 5 INST SPAN MODE high
```

set din *channel index state*

This command assigns digital input *channel* (1-16) to activate the action indicated by *index* (1-35), when the input transitions to the designated *state* (high or low). Use "list din var" command to obtain the list of supported *index* values and corresponding actions.

```
Send:         set din 1 3 high
Receive:      set din 1 3 high ok
```

dout *channel*

This command reports the index number and output variable and the active state assigned to output *channel*. The example that follows reports the input 4 to be assigned an index number 5 corresponding to "sample mode" with the active state being open.

```
Send:        dout 4
Receive:     dout 4 5 SAMPLE MODE open
```

set dout *channel index state*
This command assigns digital output *channel* to be assigned to the action associated with *index*, and assigns it an active *state* of state (open or closed).

```
Send:        set dout 4 11 open
Receive:     set dout 4 11 open ok
```

dtoa *channel*
This reports the outputs of the 6 or 12 digital to analog converters, according to the following table. The example that follows shows that the DAC 1 is 97.7% full-scale.

```
Send:        dtoa 1
Receive:     dtoa 1 97.7%
```

Note All channel ranges are user definable. If any customization has been made to the analog output configuration, the default selections may not apply. ▲

Table 1–15. Default Output Assignment

D to A	Function	Single Range
1	Voltage Output	Hg0
2	Voltage Output	Hg2+
3	Voltage Output	Hgt
4	Voltage Output	Not Assigned
5	Voltage Output	Not Assigned
6	Voltage Output	Not Assigned
7	Current Output	Hg0
8	Current Output	Hg2+
9	Current Output	Hgt
10	Current Output	Not Assigned
11	Current Output	Not Assigned
12	Current Output	Not Assigned

list var aout
list var dout
list var din

These commands report the list of index numbers and the variables (associated with that index number) available for selection in the current mode for analog output, digital output and digital inputs. The index number is used to insert the variable in a field location in a list using "set sp field index". The example that follows reports the list of analog output, index numbers, and variables.

```
Send:          list var aout
Receive:       list var aout
                1 hg0       17 edup       33 ain3
                2 hg2       18 vac        34 ain4
                3 hgt       19 smplf      35 ain5
                4 rghg0     20 pmtv       36 ain6
                5 rghg2     21 pres       37 ain7
                6 rghgt     22 pbspn      38 ain8
                7 rfint     23 anspn      40 hg0dic
                8 intt      24 anspn      41 hg0dit
                9 rctt      25 obkg       42 hgtdic
               10 prbt      26 tblg       43 hgtdit
               11 cnvt      27 ocoef      44 hg0dsc
               12 umbt      28 tcoef      45 hg0dst
               13 vntp      29 fsafe      46 hgtdsc
               14 orfp      30 dilf       47 hgtdst
               15 dilp      31 ain1       48 hg81
               16 bbkp      32 ain2       49 lampt
                                          50 oxyt
```

relay
relay stat

This command reports the current relay logic normally "open" or normally "closed," if all the relays are set to same state, that is all open or all closed.

Note The relay stat command is report only. ▲

The example that follows shows the status when all the relays logic are set to normally "open".

```
Send:          relay
Receive:       relay open
```

Note If individual relays have been assigned different logic then the response would be a 4-digit hexadecimal string with the least significant byte (LSB) being relay no 1. ▲

For example:

```
Receive:   relay stat 0x0001 (indicates relay no 1 is set to normally
open logic, all others are normally closed)
Receive:   relay stat 0x0005 (indicates relay no 1 and 3 are set to be
normally open logic, all others are normally closed)
```

set relay *action*:*relaynumber*
action = open or closed
relaynumber = number of the selected relay
These commands set the relay logic to normally open or closed for relay number *relaynumber*, where *relaynumber* is the relay between 1 and 16. The example that follows sets the relay no 1 logic to normally open.

Note If the command is sent without an appended relay number then all the relays are assigned the set logic of normally open/closed. ▲

```
Send:          set relay open 1
Receive:       set relay open 1 ok
```

Record Layout Definition

The Erec, Lrec, and Srec Layouts contain the following:

- A format specifier for parsing ASCII responses
- A format specifier for parsing binary responses

In addition to these, the Erec Layout contains:

- A format specifier for producing the front-panel displays

In operation, values are read in using either the ASCII or binary format specifiers and converted to uniform internal representations (32-bit floats or 32-bit integers). These values are converted into text for display on the screen using the format specifier for the front-panel display. Normally, the specifier used to parse a particular datum from the input stream will be strongly related to the specifier used to display it (e.g., all of the floating point inputs will be displayed with an 'f' output specifier, and all of the integer inputs will be displayed with a 'd' specifier).

Format Specifier for ASCII Responses

The first line of the Layout response is the scanf-like parameter list for parsing the fields from an ASCII ERec response. Parameters are separated by spaces and the line is terminated by a \n (the normal line separator character). Valid fields are:

%s - parse a string

%d - parse a decimal number

%ld - parse a long (32-bit) decimal number

%f - parse a floating point number

%x - parse a hexadecimal number

%lx - parse a long (32-bit) hex number

%* - ignore the field

Note Signed versus unsigned for the integer values does not matter; it is handled automatically. ▲

Format Specifier for Binary Responses

The second line of the Layout response is the binary parameter list for parsing the fields from a binary response. Parameters MUST be separated by spaces, and the line is terminated by a '\n'. Valid fields are:

t - parse a time specifier (2 bytes)

D - parse a date specifier (3 bytes)

i - ignore one 8-bit character (1 byte)

e - parse a 24-bit floating point number (3 bytes: n/x)

E - parse a 24-bit floating point number (3 bytes: N/x)

f - parse a 32-bit floating point number (4 bytes)

c - parse an 8-bit signed number (1 byte)

C - parse an 8-bit unsigned number (1 byte)

n - parse a 16-bit signed number (2 bytes)

N - parse a 16-bit unsigned number (2 bytes)

m - parse a 24-bit signed number (3 bytes)

M - parse a 24-bit unsigned number (3 bytes)

l - parse a 32-bit signed number (4 bytes)

L - parse a 32-bit unsigned number (4 bytes)

There is an optional single digit d which may follow any of the numeric fields which indicates that after the field has been parsed out, the resulting value is to be divided by 10^d. Thus the 16-bit field 0xFFC6 would be interpreted with the format specifier 'n3' as the number -0.058.

Format Specifier for Front-Panel Layout

The subsequent lines in the ERec Layout response describe the appearance of the full panel. The full instrument panel as it appears on the screen has two columns of lines. Each line is composed of three major components: (1) a text field, (2) a value field, and (3) a button. None of these three components is required. The text field contains statically displayed text.

The value field displays values which are parsed out of the response to a DATA/ERec command. It also displays, though background changes, alarm status. The button, when pressed, triggers input from either a dialog box or a selection list. There are five kinds of buttons, B, I, L, T, and N.

Each line in the layout string corresponds to one line on the display. The layout string describes each of the three major fields as well as translation mechanisms and corresponding commands.

Text

The first field in the layout string is the text. It is delimited by a ':'. The string up to the first ':' will be read and inserted in the text field of the line.

Value String

This is followed by a possible string, enclosed in quotes. This is used to place a string into the value field.

Value Source

The value source, which is the item (or word) number in the DATA/ERec response, appears next. This is followed by an optional bitfield designator. The datum identified by the value source can be printed as a string 's', hexadecimal 'x', decimal 'd', or floating point 'f', or binary 'b' number. Typically, bitfield extractions are only done for decimal or hexadecimal numbers.

Floating-point numbers can be followed with an optional precision specifier which will be used as an argument to printf's %f format (e.g., a field of '4' would be translated into the printf command of '%.3f'). Alternately, the special character '*' can precede the precision specifier; this causes an indirection on the precision specifier (which now becomes a field number).

This is useful when formatting, for example, numbers which have varying precision depending on the mode of the instrument.

Binary numbers can also have an optional precision specifier which is used to determine how many bits to print. For example, the specifier 'b4' will print the lowest four bits of the parsed number.

There are serious restrictions on where an 's' field may appear: currently sources 1 and 2 must be 's', and no others may be 's'.

Alarm Information

The value source is followed by optional alarm information, indicated by a commercial at sign '@' with a source indicator and a starting bit indicator. All alarm information is presumed to be two bits long (low and high). The bitfield extraction is performed on the integer part of the source. Typical alarm information would appear as '@6.4'.

Translation Table

Then, there appears an optional translation table within braces '{}'. This is a string of words separated by spaces. An example translation table would be '{Code_0 Code_1 Code_2 Code_3}'. The value, once extracted is used as a zero-based index into the translation table to determine the string to display.

Selection Table

Then there appears an optional selection table within parentheses '(...)'. This is a string of numbers separated by spaces '(0 1)'. The selection table lists the translation table entries which the user may select from when setting the parameter. This is not necessarily the same as the entries which may be displayed.

Button Designator

Then there appears an optional button designator. This will be one of 'B', 'I', 'L', 'T', or 'N'.

B- Indicates a button which pops up an input dialog prompting the user for a new value using the designated input format. The input format is specified from the 'B' through the subsequent semicolon.

I—Indicates a button which pops up a selection list with input translation. That is, the values read are translated before they are compared to the selection list options.

L—Indicates a button which pops up a selection list without any translation. The output value is number of the selected option.

T—Indicates a button which pops up a selection list with output translation. The number of the option selected is used as an index into the translation table to generate an output string.

N—Indicates a button which only sends the subsequent command to the instrument. No user-prompting happens.

The following string through an optional '|' or the end of the line is the command which is to be sent to the instrument upon the completion of the button selection. The command string should normally contain print-style formatting to include the user input. If a '|' is present, it indicates a command which is sent to the instrument upon successful completion of the button command to update the value field.

Examples

Some examples ('\n' is the C syntax for an end-of-line character):

'Concentrations\n'

This is a single text-only line.

'\n'

This is a single blank line.

' hg0:3s\n'

This is a line which appears slightly indented. The text field is 'Hg0', the value is taken from the third element of the data response, and interpreted as a string.

' hg0:18sBd.ddd;set hg0 coef %s\n'

This is a line which also appears slightly indented. The next field is also 'Hg0', but the value is taken from the eighteenth element of the data response, again interpreted as a string. A button appears on this line which, when pressed, pops up an input dialog which will state "Please enter a new value for Hg0 using a d.ddd format." The string entered by the user is used to construct the output command. If the user enters, for example, '1.234', the constructed command will be 'set Hg0 coef 1.234'.

' hg0:21f{Code_0 Code_1 Code_2 Code_3 Code_4 Code_5 Code_6 Code_7 Code_8 Code_9 Code_10 Code_11}Lset range hg0 %d\n'

This is a line which appears slightly indented, the title is again 'Hg0', and the value the twenty-first element of the data response, interpreted as a floating-point number. There is a no-translation button which creates a selection list of twelve "Code nn" options. The number of the user selection is used to create the output command.

'Mode:6.12-13x{local remote service service}(0 1)Tset mode %s\n'

This is a line which has a title of 'Mode', and value taken from the sixth field of the data response. There is a bitfield extraction of bits 12 through 13 from the source (the value type is not important here because the value is being translated to an output string). Once the bits have been extracted, they are shifted down to the bit-zero position. Thus, the possible values of this example will be 0 through 3. The translation list shows the words which correspond to each input value, the zeroth value appearing first (0 -> local, 1 -> remote, etc.). The selection list shows that only the first two values, in this case, are to be shown to the user when the button is pressed. The 'T' button indicates full translation, input code to string, and user selection number to output string.

'\xC'

This is a line that starts a new column (the \xC or ^L),

' Comp:6.11x{off on}Tset temp comp %s\n'

This shows that the bitfield end (the second part of a bitfield specification) is optional. The bitfield will be one bit long, starting in this case at the eleventh bit.

'Background:7f*8Bd.ddd;set o3 bkg %s\n'

This shows the use of indirect precision specifiers for floating point displays. The background value is taken from the 7th element, and the precision specifier is taken from the 8th. If the asterisk were not present, it would indicate instead that 8 digits after the decimal point should be displayed.

Chapter 2
80*i* Modbus Protocol

This appendix provides a description of the MODBUS Protocol Interface and is supported both over RS-232/485 (RTU protocol) as well as TCP/IP over Ethernet.

The MODBUS Commands that are implemented are explained in detail in this document. The MODBUS protocol support for the *i*Series enables the user to perform the functions of reading the various concentrations and other analog values or variables, read the status of the digital outputs of the instrument, and to trigger or simulate the activation of a digital input to the instrument. This is achieved by using the supported MODBUS commands listed that follows.

For details of the Model 80*i* MODBUS Protocol specification, see the following topics:

- "Serial Communication Parameters" on page 2-1
- "TCP Communication Parameters" on page 2-2
- "Application Data Unit Definition" on page 2-2
- "Function Codes" on page 2-3
- "MODBUS Addresses Supported" on page 2-8

Additional information on the MODBUS protocol can be obtained at http://www.modbus.org. References are from MODBUS Application Protocol Specification V1.1a MODBUS-IDA June 4, 2004.

Serial Communication Parameters

The following are the communication parameters that are used to configure the serial port of the *i*Series to support MODBUS RTU protocol.

Number of Data bits	: 7 or 8
Number of Stop bits	: 1 or 2
Parity	: None, Odd, or Even
Data rate	: 1200 to 115200 Baud (9600 is default)

TCP Communication Parameters

*i*Series Instruments support the MODBUS/TCP protocol. The register definition is the same as for the serial interface. Up to three simultaneous connections are supported over Ethernet.

TCP connection port for MODBUS: 502

Application Data Unit Definition

Here are the MODBUS ADU (Application Data Unit) formats over serial and TCP/IP:

Serial:	Slave Address	Function Code	Data	Error Check
TCP/IP:	MBAP Header	Function Code	Data	

Slave Address

The MODBUS slave address is a single byte in length. This is the same as the instrument ID used for C-Link commands and can be between 1 and 127 decimal (i.e. 0x01 hex to 0x7F hex). This address is only used for MODBUS RTU over serial connections.

Note Device ID '0' used for broadcast MODBUS commands, is not supported. Device IDs 128 through 247 (i.e. 0x80 hex to 0xF7 hex) are not supported because of limitations imposed by C-Link. ▲

MBAP Header

In MODBUS over TCP/IP, a MODBUS Application Protocol Header (MBAP) is used to identify the message. This header consists of the following components:

Transaction Identifier	2 Bytes	0x0000 to 0xFFFF (Passed back in response)
Protocol Identifier	2 Bytes	0x00 (MODBUS protocol)
Length	2 Bytes	0x0000 to 0xFFFF (Number of following bytes)
Unit Identifier	1 Byte	0x00 to 0xFF (Passed back in response)

A Slave address is not required in MODBUS over TCP/IP because the higher-level protocols include device addressing. The unit identifier is not used by the instrument.

Function Code

The function code is a single byte in length. The following function codes are supported by the instrument:

Read Coils	:	0x01
Read Inputs	:	0x02
Read Holding Registers	:	0x03
Read Input Registers	:	0x04
Force (Write) Single Coil	:	0x05
Read Exception Status	:	0x07

If a function code is received that is not in this list, and invalid function exception is returned.

Data

The data field varies depending on the function. For more description of these data fields, see "Function Codes" that follows.

Error Check

In MODBUS over Serial an error check is included in the message. This is not necessary in MODBUS over TCP/IP because the higher-level protocols ensure error-free transmission. The error check is a two-byte (16-bit) CRC value.

Function Codes

This section describes the various function codes that are supported by the Model 80*i*.

(0x01/0x02) Read Coils / Read Inputs

Read Coils/Inputs reads the status of the digital outputs (relays) in the instrument. Issuing either of these function codes will generate the same response.

These requests specify the starting address, i.e. the address of the first output specified, and the number of outputs. The outputs are addressed starting at zero. Therefore, outputs numbered 1–16 are addressed as 0–15.

The outputs in the response message are packed as one per bit of the data field. Status is indicated as 1 = Active (on) and 0 = Inactive (off). The LSB of the first data byte contains the output addressed in the query. The other outputs follow toward the high order end of this byte, and from low order to high order in subsequent bytes. If the returned output quantity is not a multiple of eight, the remaining bits in the final data byte will be padded with zeros (toward the high order end of the byte). The Byte Count field specifies the quantity of complete bytes of data.

Note The values reported may not reflect the state of the actual relays in the instrument, as the user may program these outputs for either active closed or open. ▲

Request

Function code	1 Byte	0x01 or 0x02
Starting Address	2 Bytes	0x0000 to maximum allowed by instrument
Quantity of outputs	2 Bytes	1 to maximum allowed by instrument
Unit Identifier	1 Byte	0x00 to 0xFF (Passed back in response)

Response

Function code	1 Byte	0x01 or 0x02
Byte count	1 Byte	N (N = Quantity of Outputs / 8, if the remainder not equal to zero, then N=N+1)
Output Status	n Byte	N = N or N+1

Error Response

Function code	1 Byte	0x01 or 0x02
Exception code	1 Byte	01=Illegal Function, 02=Illegal Address, 03=Illegal Data, 04=Slave Device Failure

Here is an example of a request and response to read outputs 2–15:

Request

Field Name	*(Hex)*
Function	0x01
Starting Address Hi	0x00
Starting Address Lo	0x02
Quantity of Outputs Hi	0x00
Quantity of Outputs Lo	0x0D

Response

Field Name	*(Hex)*
Function	0x01
Byte Count	0x03
Output Status 2–10	0xCD
Output Status 11–15	0x0A

The status of outputs 2–10 is shown as the byte value 0xCD, or binary 1100 1101. Output 10 is the MSB of this byte, and output 2 is the LSB. By convention, bits within a byte are shown with the MSB to the left, and the LSB to the right. Thus the outputs in the first byte are '10 through 2', from left to right. In the last data byte, the status of outputs 15-11 is shown as the byte value 0x0A, or binary 0000 1010. Output 15 is in the fifth bit position from the left, and output 11 is the LSB of this byte. The four remaining high order bits are zero filled.

(0x03/0x04) Read Holding Registers / Read Input Registers

Read holding/input registers reads the measurement data from the instrument. Issuing either of these function codes will generate the same response. These functions read the contents of one or more contiguous registers.

These registers are 16 bits each and are organized as shown below. All of the values are reported as 32-bit IEEE standard 754 floating point format. This uses 2 sequential registers, least significant 16 bits first.

The request specifies the starting register address and the number of registers. Registers are addressed starting at zero. Therefore registers numbered 1–16 are addressed as 0–15. The register data in the response message are packed as two bytes per register, with the binary contents right justified within each byte. For each register, the first byte contains the high order bits and the second contains the low order bits.

The status of outputs 2–10 is shown as the byte value 0xCD, or binary 1100 1101. Output 10 is the MSB of this byte, and output 2 is the LSB. By convention, bits within a byte are shown with the MSB to the left, and the LSB to the right. Thus, the outputs in the first byte are '10 through 2', from left to right. In the last data byte, the status of outputs 15-11 is shown as the byte value 0x0A, or binary 0000 1010. Output 15 is in the fifth bit position from the left, and output 11 is the LSB of this byte. The four remaining high order bits are zero filled.

The request specifies the starting register address and the number of registers. Registers are addressed starting at zero. Therefore, registers numbered 1–16 are addressed as 0–15. The register data in the response message are packed as two bytes per register, with the binary contents right justified within each byte. For each register, the first byte contains the high order bits and the second contains the low order bits.

Request

Function code	1 Byte	0x03 or 0x04
Starting Address	2 Bytes	0x0000 to maximum allowed by instrument
Quantity of Registers	2 Bytes	1 to maximum allowed by instrument

Response

Function code	1 Byte	0x03 or 0x04
Byte count	1 Byte	2 x N (N = quantity of registers)
Register value	N* x 2 Bytes	N = N or N+1

Error Response

Function code	1 Byte	Function code + 0x80
Exception code	1 Byte	01=Illegal Function, 02=Illegal Address, 03=Illegal Data, 04=Slave Device Failure

Here is an example of a request and response to read registers 10–13:

Request

Field Name	*(Hex)*
Function	0x03
Starting Address Hi	0x00
Starting Address Lo	0x09
No. of Registers Hi	0x00
No. of Registers Lo	0x04

Response

Field Name	*(Hex)*
Function	0x03
Byte Count	0x06
Register value Hi (10)	0x02
Register value Lo (10)	0x2B

Register value Hi (11)	0x00
Register value Lo (11)	0x00
Register value Hi (12)	0x00
Register value Lo (12)	0x64
Register value Hi (13)	0x00
Register value Lo (13)	0x64

The contents of register 10 are shown as the two byte values of 0x02 0x2B. Then contents of registers 11–13 are 0x00 0x00, 0x00 0x64 and 0x00 0x64 respectively.

(0x05) Force (Write) Single Coil

The force (write) single coil function simulates the activation of the digital inputs in the instrument, which triggers the respective action.

This function code is used to set a single action to either ON or OFF. The request specifies the address of the action to be forced. Actions are addressed starting at zero. Therefore, action number 1 is addressed as 0. The requested ON/OFF state is specified by a constant in the request data field. A value of 0xFF00 requests the action to be ON. A value of 0x0000 requests it to be OFF. All other values are illegal and will not affect the output. The normal response is an echo of the request, returned after the state has been written.

Note This function will not work if the instrument is in service mode. ▲

Request

Function code	1 Byte	0x05
Starting Address	2 Bytes	0x0000 to maximum allowed by instrument
Output Value	2 Bytes	0x0000 or 0xFF00

Response

Function code	1 Byte	0x05
Starting Address	2 Bytes	0x0000 to maximum allowed by instrument
Output Value	2 Bytes	0x0000 or 0xFF00

Error Response

Function code	1 Byte	Function code + 0x80

Exception code	1 Byte	01=Illegal Function, 02=Illegal Address, 03=Illegal Data, 04=Slave Device Failure

Here is an example of a request to write Coil 5 ON:

Request

Field Name	*(Hex)*
Function	05
Output Address Hi	00
Output Address Lo	05
Output Value Hi	FF
Output Value Lo	00

Response

Field Name	*(Hex)*
Function	05
Output Address Hi	00
Output Address Lo	05
Output Value Hi	FF
Output Value Lo	00

MODBUS Addresses Supported

Table 2–1 through **Table 2–3** list the MODBUS addresses supported for the Model 80*i*.

IMPORTANT NOTE The addresses in the following tables are Protocol Data Unit (PDU) addresses. Verify the coil number on your MODBUS master to ensure that it matches the coil number on the instrument. ▲

Note Coil status 1 indicates active state. ▲

Table 2–1. Read Coils for 80*i* (Digital Output)

Coil Number	Status
0	Invalid
1	SERVICE
2	Hg^0 MODE
3	Hgt MODE

Coil Number	Status
4	Hg^0/Hg^t MODE
5	SAMPLE MODE
6	INSTRUMENT ZERO MODE
7	INSTRUMENT SPAN MODE
8	ORIFICE ZERO MODE
9	ORIFICE SPAN MODE
10	SYSTEM ZERO MODE
11	SYSTEM SPAN MODE
12	BLOWBACK MODE
13	GENERAL ALARM
14	INTERNAL TEMPERATURE ALARM
15	CHAMBER TEMPERATURE ALARM
16	CHAMBER PRESSURE ALARM
17	SAMPLE FLOW ALARM
18	INTENSITY ALARM
19	Hg^0 CONCENTRATION, MAXIMUM ALARM
20	Hg^0 CONCENTRATION, MINIMUM ALARM
21	Hg2+ CONCENTRATION, MAXIMUM ALARM
22	Hg^{2+} CONCENTRATION, MINIMUM ALARM
23	Hg^t CONCENTRATION, MAXIMUM ALARM
24	Hg^t CONCENTRATION, MINIMUM ALARM
25	MOTHERBOARD STATUS ALARM
26	MEASUREMENT INTERFACE BOARD STATUS ALARM
27	I/O BOARD STATUS ALARM
28	81*i* STATUS
29	ZERO CHECK/CAL ALARM
30	SPAN CHECK/CAL ALARM
31	PROBE DILUTION ALARM
32	SYSTEM DILUTION ALARM
33	PROBE ZERO CHECK/CAL ALARM
34	SYSTEM ZERO CHECK/CAL ALARM
35	GENERAL PROBE ALARM
36	PROBE 1 SELECTED
37	Unused

Coil Number	Status
38	Unused
39	Unused
40	LOCAL/REMOTE
41	OXIDIZER CAL MODE
42	HYDRATOR
43	PROBE 1 STATUS
44	CURRENT SAMPLE Hg^0
45	CURRENT SAMPLE Hgt
46	O2 QUENCH STATUS
47	EXT ALARM
48	OXIDIZER Hg
49	OXIDIZER Cl2
50	OXIDIZER PURGE
51	THC ZERO MODE
52	THC SPAN MODE
53	PERMEATION SPAN MODE
54	84*i* CONNECT A
55	84*i* CONNECT B
56	84*i* GAS TEMP
57	84*i* OVEN TEMP
58	84*i* CAPILLARY TEMP
59	84*i* FLOW
60	84*i* PRESSURE
61	84*i* STATUS

IMPORTANT NOTE The addresses in the following tables are Protocol Data Unit (PDU) addresses. Verify the register number on your MODBUS master to ensure that it matches the register number on the instrument. ▲

Note For additional information on how to read registers and interpret the data, refer to the "(0x03/0x04) Read Holding Registers / Read Input Registers" section in this appendix. ▲

Table 2–2. Read Registers for 80*i* (Analog Output)

Register Number	Variable
0	Invalid
1&2	Hg^0
3&4	Hg^{2+}
5&6	Hg^t
7&8	Hg^0 RANGE
9&10	Hg^{2+} RANGE
11&12	Hg^t RANGE
13&14	INTENSITY
15&16	INTERNAL TEMPERATURE
17&18	CHAMBER TEMPERATURE
19&20	PROBE TEMPERATURE
21&22	CONVERTER TEMPERATURE
23&24	UMBILICAL TEMPERATURE
25&26	VENTURI PRESSURE
27&28	ORIFICE PRESSURE
29&30	DILUTION AIR PRESSURE
31&32	BLOWBACK PRESSURE
33&34	EDUCTOR PRESSURE
35&36	VACUUM PRESSURE
37&38	FLOW
39&40	PMT VOLTS
41&42	CHAMBER PRESSURE
43&44	PROBE SPAN
45&46	Hg^0 SPAN
47&48	Hg^t SPAN
49&50	Hg^0 BACKGROUND
51&52	Hg^t BACKGROUND
53&54	Hg^0 COEFFICIENT
55&56	Hg^t COEFFICIENT
57&58	PROBE FAILSAFE TEMPERATURE
59&60	DILUTION FACTOR
61&62	ANALOG IN 1
63&64	ANALOG IN 2

Register Number	Variable
65&66	ANALOG IN 3
67&68	ANALOG IN 4
69&70	ANALOG IN 5
71&72	ANALOG IN 6
73&74	ANALOG IN 7
75&76	ANALOG IN 8
77&78	PROBE NUMBER
79&80	Hg^0 INSTRUMENT DRIFT CONCENTRATION
81&82	Hg^0 INSTRUMENT DRIFT TIME
83&84	Hg^t INSTRUMENT DRIFT CONCENTRATION
85&86	Hg^t INSTRUMENT DRIFT TIME
87&88	Hg^0 SYSTEM DRIFT CONCENTRATION
89&90	Hg^0 SYSTEM DRIFT TIME
91&92	Hg^t SYSTEM DRIFT CONCENTRATION
93&94	Hg^t SYSTEM DRIFT TIME
95&96	CALIBRATOR ACTUAL CONCENTRATION
97&98	LAMP TEMPERATURE
99&100	OXIDIZER TEMPERATURE
101&102	OXIDATION
103&104	INTEGRITY
105&106	UMBILICAL TEMP 2
107&108	EXT ALARMS
109&110	84*i* PERM GEN RATIO
111&112	84*i* PERM GAS TEMP
113&114	84*i* PERM OVEN HEATER TEMP
115&116	84*i* CAPILLARY TEMP
117&118	84*i* PRESSURE
119&120	CURRENT O2 %

IMPORTANT NOTE The addresses in the following tables are Protocol Data Unit (PDU) addresses. Verify the coil number on your MODBUS master to ensure that it matches the coil number on the instrument. ▲

Note Writing 1 to the coil number shown in the following table will initiate the "action triggered" listed in the table. This state must be held for at least 1 second to ensure the instrument detects the change and triggers the appropriate action. ▲

Note The coils within each coil group in the following table are mutually exclusive and will not be triggered if there is a conflict. Before you assert (1) one coil in a group, make sure the other coils in the group are de-asserted (0). ▲

Note If an item from the "System Span" Coil Group is triggered during an oxidation sequence, the span level will be changed to the corresponding span number. The span level will remain at this setting for the remainder of the System Integrity Test unless otherwise changed again. This is to facilitate multi-level integrity checks. ▲

Table 2–3. Write Coils for 80*i* (Digital Input)

Coil Number	Action Triggered	Coil Group
100	Invalid	
101	Hg^0 MODE	Measure Mode
102	Hg^t MODE	Measure Mode
103	Hg^0/Hg^t MODE	Measure Mode
104	INSTRUMENT ZERO MODE	Zero Span Mode
105	INSTRUMENT SPAN MODE	Zero Span Mode
106	Unused	
107	Unused	
108	SYSTEM ZERO MODE	Zero Span Mode
109	SYSTEM SPAN MODE	Zero Span Mode
110	BLOWBACK SYSTEM MODE	Zero Span Mode
111	BLOWBACK STINGER MODE	Zero Span Mode
112	SET BACKGROUND	
113	SET SPAN COEF	

Coil Number	Action Triggered	Coil Group
114	PROBE SELECT 1	Probe Select
115	PROBE SELECT 2	Probe Select
116	PROBE SELECT 3	Probe Select
117	PROBE SELECT 4	Probe Select
118	SYSTEM SPAN 1	System Span
119	SYSTEM SPAN 2	System Span
120	SYSTEM SPAN 3	System Span
121	SYSTEM SPAN 4	System Span
122	SYSTEM SPAN 5	System Span
123	SYSTEM SPAN 6	System Span
124	OXIDIZER CAL MODE	Zero Span Mode
125	SPIKING	
126	INSTRUMENT SPAN 1	Instrument Span
127	INSTRUMENT SPAN 2	Instrument Span
128	INSTRUMENT SPAN 3	Instrument Span
129	HYDRATOR ALARM	
130	EXT ALARM	
131	Unused	
132	Unused	
133	Unused	
134	PERM SPAN	84*i* Perm
135	Unused	
136	84*i* CONNECT B	84*i* Perm
137	SET Hg^0 SPAN	Measure Mode
138	SET Hg^t SPAN	Measure Mode

Chapter 3
81*i* C-Link Protocol Commands

This appendix provides a description of the C-Link protocol commands that can be used to remotely control a Model 81*i* analyzer using a host device such as a PC or a datalogger. C-Link protocol may be used over RS-232, RS-485, or Ethernet. C-Link functions can be accessed over Ethernet using TCP/IP port 9880. Streaming data may be accessed over Ethernet using TCP/IP port 9881. Up to three simultaneous connections per protocol may be made over Ethernet.

Instrument Identification Number

Each command sent to the instrument must begin with the American Standard Code for Information Interchange (ASCII) symbol or byte value equivalent of the instrument's identification number plus 128. For example, if the instrument ID is set to 25, then each command must begin with the ACSII character code 153 decimal. The instrument ignores any command that does not begin with its instrument identification number. If the instrument ID is set to 0, then this byte is not required. For more information on changing Instrument ID, see Chapter 3, "Operation".

Commands

The instrument must be in the remote mode in order to change instrument parameters via remote. However, the command "set mode remote" can be sent to the instrument to put it in the remote mode. Report commands (commands that don't begin with "set") can be issued either in the remote or local mode. For information on changing modes, see Chapter 3, "Operation."

The commands can be sent in either uppercase or lowercase characters. Each command must begin with the proper instrument identification number (ASCII) character. The command in the example that follows begins with the ASCII character code 208 decimal, which directs the command to the Model 81*i,* and is terminated by a carriage return "CR" (ASCII character code 13 decimal).

<ASCII 208>	T	I	M	E	<CR>

If an incorrect command is sent, a "bad command" message will be received. The example that follows sends the incorrect command "set sp field 1" instead of the correct command "`set sp field 1 34`".

```
Send:         set sp field 1
Receive:      set sp field 1 bad cmd
```

Table 3–1 provides a description of the command response errors.

Table 3–1. Command Response Error Descriptions

Command Response	Description
too high	Supplied value is higher than the upper limit
too low	Supplied value is lower than the lower limit
invalid string	Supplied string invalid (typically because a letter was detected when the value should be numeric)
data not valid	Supplied value is not acceptable for entered command
can't, wrong settings	Command not allowed for current measurement mode
can't, mode is service	Command not allowed while instrument is in service mode

The "save" and "set save params" commands "stores parameters in FLASH. It is important that each time instrument parameters are changed, that this command be sent. If changes are not saved, they will be lost in the event of a power failure.

Accessing Streaming Data

Streaming data is sent out the serial port or the Ethernet port on a user-defined periodic basis. Streaming data over Ethernet is only generated when a connection is made on TCP port 9881. Up to three simultaneous connections are allowed per protocol.

Service Mode

If the Service Mode is active, C-Link "set" commands are not allowed. This is to prevent parameters from being changed remotely while the unit is being serviced locally.

Commands List

Table 3–2 lists the 81*i* C-Link protocol commands. The interface will respond to the associated command strings.

Table 3–2. C-Link Protocol Commands

Command	Description	Page
addr dns	Reports/sets domain name server address	3-28
addr gw	Reports/sets default gateway address	3-28
addr ip	Reports/sets IP address	3-28
addr nm	Reports/sets netmask address	3-29
addr ntp	Reports the IP address for the NTP time server	3-29
alarm ambient temp max	Reports/sets ambient alarm maximum value	3-10
alarm ambient temp min	Reports/sets ambient alarm minimum value	3-10
alarm conc hg max	Reports/sets Hg concentration alarm maximum value	3-10
alarm conc hg min	Reports/sets Hg concentration alarm minimum value	3-10
alarm cooler temp max	Reports/sets current cooler alarm maximum value	3-11
alarm cooler temp min	Reports/sets current cooler alarm minimum value	3-11
alarm dilution flow max	Reports/sets current dilution flow alarm maximum value	3-11
alarm dilution flow min	Reports/sets current dilution flow alarm minimum value	3-11
alarm hg flow max	Reports/sets current Hg flow alarm maximum value	3-12
alarm hg flow min	Reports/sets current Hg flow alarm minimum value	3-12
alarm pressure max	Reports/sets current pressure alarm maximum value	3-12
alarm pressure min	Reports/sets current pressure alarm minimum value	3-12
alarm trig conc hg	Reports/sets current trig conc Hg alarm trigger sense	3-13
allow mode cmd	Reports/sets the current allow set mode remote” commands.	3-32
ambient temp	Reports current ambient temperature	3-8

Command	Description	Page
analog iout range	Reports/sets analog current output range per channel	3-35
analog vin	Retrieves analog voltage input data per channel	3-35
analog vout range	Reports/sets analog voltage output range per channel	3-35
baud	Reports/sets current baud rate	3-29
clr lrecs	Clears away only long records that have been saved	3-14
clr records	Clears away all logging records that have been saved	3-14
clr srecs	Clears away only short records that have been saved	3-14
contrast	Reports/sets current screen contrast	3-25
cooler setpoint	Reports/sets current cooler setpoint for 81*i* and 81*i*-L options	3-8
cooler temp	Reports current cooler temperature	3-8
copy lrec to sp	Sets/copies current lrec selection into the scratch pad	3-20
copy sp to lrec	Sets/copies current selections in scratch pad into lrec list	3-20
copy sp to srec	Sets/copies current selections in scratch pad into srec list	3-20
copy sp to stream	Sets/copies current selections in scratch pad into stream list	3-20
copy srec to sp	Sets/copies current srec selection into the scratch pad	3-20
copy stream to sp	Sets/copies current streaming data selection into the scratch pad	3-20
data treatment lrec/srec	Reports/sets the current selection of data treatment for concentrations in the lrecs or srecs	3-19
date	Reports/sets current date	3-26
default params	Sets parameters to default values	3-26
dhcp	Reports/sets state of use of DHCP	3-30
diag volt iob	Reports diagnostic voltage level for optional I/O expansion board	3-13
diag volt mb	Reports diagnostic voltage level for motherboard	3-13
diag volt mib	Reports diagnostic voltage level for measurement interface board	3-13
diag volt probe	Reports diagnostic voltage level for 82*i* measurement interface board	3-14
dig in	Reports status of the digital inputs	3-36
dilution flow	Reports the current dilution flow	3-8
din	Reports/sets digital input channel and active state	3-36
do (down)	Simulates pressing down pushbutton	3-22
dout	Reports/sets digital output channel and active state	3-37
dtoa	Reports outputs of the digital to analog converters per channel	3-37
en (enter)	Simulates pressing enter pushbutton	3-22
er	Returns a brief description of the main operating conditions in the format specified in the commands	3-15

Command	Description	Page
erec	Returns a snapshot of the main operating conditions (measurements and status) in the specified format	3-16
erec format	Reports/sets erec format (ASCII or binary)	3-16
erec layout	Reports current layout of erec data	3-17
flags	Reports 8 hexadecimal digits (or flags) that represent the status of the PMT, gas mode, and alarms	3-9
format	Reports/sets current reply termination format	3-30
gas mode	Reports/sets current gas mode	3-24
he (help)	Simulates pressing help pushbutton	3-22
hg	Reports current measured Hg concentration	3-8
hg flow	Reports current measured Hg flow	3-9
hg span	Reports/sets Hg span values	3-25
hg span conc	Reports/sets Hg span concentration	3-25
hg span range	Reports/sets Hg span range	3-24
host name	Reports/sets host name string	3-31
instr name	Reports instrument name	3-31
instrument id	Reports/sets instrument id	3-31
isc (iscreen)	Retrieves framebuffer data used for the display	3-22
layout ack	Disables stale layout/layout changed indicator (‘*’)	3-32
le (left)	Simulates pressing left pushbutton	3-22
list din	Lists current selection for digital input	3-14
list dout	Lists current selection for digital output	3-14
list lrec	Lists current selection lrec logging data	3-15
list sp	Lists current selection in the scratchpad list	3-15
list srec	Lists current selection srec logging data	3-15
list stream	Lists current selection streaming data output	3-15
list var aout	Reports list of analog output, index numbers, and variables	3-38
list var din	Reports list of digital input, index numbers, and variables	3-38
list var dout	Reports list of digital output, index numbers, and variables	3-38
lr	Outputs long records in the format specified in the command	3-15
lrec	Outputs long records	3-17
lrec format	Reports/sets output format for long records (ASCII or binary)	3-16
lrec layout	Reports current layout of lrec data	3-17
lrec mem size	Reports maximum number of long records that can be stored	3-18

Command	Description	Page
lrec per	Reports/sets long record logging period	3-18
malloc lrec	Reports/sets memory allocation for long records	3-18
malloc srec	Reports/sets memory allocation for short records	3-18
me (menu)	Simulates pressing menu pushbutton	3-22
mode	Reports the operating mode in local, service, or remote	3-33
no of lrec	Reports/sets number of long records stored in memory	3-18
no of srec	Reports/sets number of short records stored in memory	3-18
power up mode	Reports/sets the power up mode which configures the instrument to power up in either the local/unlocked mode or the remote/locked mode.	3-33
pres	Reports current optical chamber pressure	3-9
program no	Reports instrument program number	3-34
push	Simulates pressing a key on the front panel	3-22
r cooler setpoint	Reports/sets cooler setpoint according to range (20, 30, 50, 300) for 81*i*-H only	3-19
ri (right)	Simulates pressing right pushbutton	3-22
ru (run)	Simulates pressing run pushbutton	3-22
save	Stores parameters in FLASH	3-27
save params	Stores parameters in FLASH	3-27
sc (screen)	C-series legacy command that reports a generic response (Use iscreen instead)	3-23
sp field	Reports/sets item number and name in scratch pad list	3-21
sr	Reports last short record stored	3-15
srec	Reports maximum number of short records	3-17
srec format	Reports/sets output format for short records (ASCII or binary)	3-16
srec layout	Reports current layout of short record data	3-17
srec mem size	Reports maximum number of short records	3-18
srec per	Reports/sets short record logging period	3-18
stream per	Reports/sets current set time interval for streaming data	3-21
stream time	Reports/sets a time stamp to streaming data or not	3-22
time	Reports/sets current time (24-hour time)	3-27
tz	Reports the "tz" timezone string for the NTP server	3-34
up	Simulates pressing up pushbutton	3-22

Measurements

ambient temp

This command reports the current ambient temperature. The example below shows that the ambient temperature is 30 °C.

```
Send:         ambient temp
Receive:      ambient temp 30 deg C
```

cooler setpoint

This command reports the current setpoint of the cooler. The example below shows that the cooler setpoint is 9 °C.

```
Send:         cooler setpoint
Receive:      cooler setpoint 9 deg C
```

set cooler setpoint *temp*

temp = integer value between 5 and 10 °C for 81*i* and 7 and 12 °C for 81*i*-L.

This command sets the current setpoint of the cooler. The example below sets the cooler setpoint to 9 °C.

```
Send:         set cooler setpoint 9
Receive:      set cooler setpoint 9 ok
```

Note The cooler setpoint and set cooler setpoint commands are only available for the standard 81*i* and 81*i*-L option. To view and set cooler setpoints for the 81*i*-H option, see the command r*xxx* cooler setpoint below. ▲

cooler temp

This command reports the current cooler temperature. The example below shows that the cooler temperature is 13.9 °C.

```
Send:         cooler temperature
Receive:      cooler temperature 13.9 deg C
```

dilution flow

This command reports the current dilution flow. The example below shows that the dilution flow is 12.625 lpm.

```
Send:         dilution flow
Receive:      dilution flow 12.625 lpm
```

hg

This command reports the measured Hg concentration. The example below shows that the Hg concentration is 10 μg/m^3.

```
Send:       hg
Receive:    hg 1.000E+01
```

hg flow

This command reports the measured Hg flow. The example below shows that the Hg flow is 9.250 sccm.

```
Send:       hg
Receive:    hg 9.250 sccm
```

pres

This command reports the Hg source pressure. The example below shows that the actual Hg source pressure is 48.7 mmHg.

Troubleshooting Note The pres command reports the Hg source pressure not to exceed 1300 mmHg (high) or 750 mmHg (low). A low pressure reading indicates a possible leak in the Hg source path. ▲

```
Send:       pres
Receive:    pres 48.7 mm Hg
```

flags

This reports 8 hexadecimal digits (or flags) that represent status of gas mode, service mode, password locked, and alarms. To decode the flags, each hexadecimal digit is converted to binary as shown in the **Figure 3–1**. It is the binary digits that define the status of each parameter. In the example below, the instrument is reporting that there is an Hg Concentration alarm (high) and a Pressure alarm (low).

```
Send:       flags
Receive:    flags 00000042
```

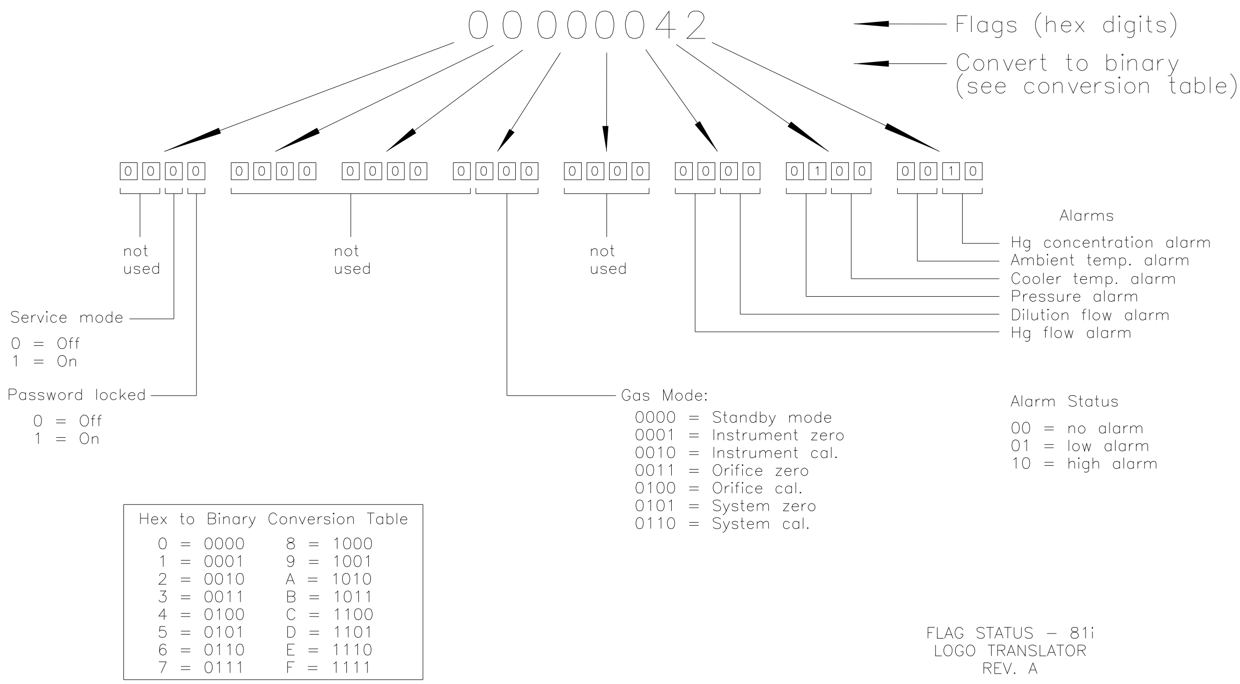

Figure 3–1. Flag Status

Alarms

alarm ambient temp min
alarm ambient temp max

These commands report the ambient temperature alarm minimum and maximum value current settings. The example below reports that the ambient temperature alarm minimum value is 15.0 °C.

```
Send:        alarm ambient temp min
Receive:     alarm temp min 15.0 deg C
```

set alarm ambient temp min *value*
set alarm ambient temp max *value*

These commands set the ambient temperature alarm minimum and maximum values to value, where value is a floating-point number representing ambient temperature alarm limits in degrees C. The example below sets the ambient temperature alarm maximum value to 35.0 °C.

```
Send:        set alarm ambient temp max 35
Receive:     set alarm ambient temp max 35 ok
```

alarm conc hg min
alarm conc hg max

These commands report the Hg concentration alarm minimum and maximum value current setting. The example below reports that the Hg concentration minimum is 2.5 µg/m^3.

```
Send:        alarm conc hg min
Receive:     alarm conc hg min 2.5 ug/m3
```

set alarm conc hg min *value*
set alarm conc hg max *value*

These commands set the Hg concentration alarm minimum and maximum value to value, where value is a floating-point representation of the concentration alarm limits. The example below sets the Hg concentration alarm maximum value to 55 µg/m^3.

```
Send:        set alarm conc hg max 55
Receive:     set alarm conc hg max 55 ug/m3 ok
```

alarm cooler temp min
alarm cooler temp max

These commands report the cooler temperature alarm minimum and maximum value current settings. The example below reports that the cooler temperature alarm minimum value is 13.0 °C.

```
Send:        alarm cooler temp min
Receive:     alarm cooler temp min 13.0 deg C
```

set alarm cooler temp min *value*
set alarm cooler temp max *value*

These commands set the cooler temperature alarm minimum and maximum values to value, where value is a floating-point number representing cooler temperature alarm limits in degrees C. The example below sets the cooler temperature alarm maximum value to 17.0 °C.

```
Send:        set alarm cooler temp max 17
Receive:     set alarm cooler temp max 17 deg C ok
```

alarm dilution flow min
alarm dilution flow max

These commands report the dilution flow alarm minimum and maximum value current settings. The example below reports that the dilution flow alarm minimum value is 1.0 lpm.

```
Send:        alarm dilution flow min
Receive:     alarm dilution flow min 1.000 lpm
```

set alarm dilution flow min *value*
set alarm dilution flow max *value*

These commands set the dilution flow alarm minimum and maximum values to *value*, where *value* is a floating-point number representing dilution flow limits in psi. The example below sets the dilution flow alarm maximum value to 25 lpm.

```
Send:         set alarm dilution flow max 25
Receive:      set alarm dilution flow max 25 ok
```

alarm hg flow min
alarm hg flow max

These commands report the hg flow alarm minimum and maximum value current settings. The example below reports that the hg flow alarm minimum value is 2.5 sccm.

```
Send:         alarm hg flow min
Receive:      alarm hg low min 2.500 sccm
```

set alarm hg flow min *value*
set alarm hg flow max *value*

These commands set the hg flow alarm minimum and maximum values to *value*, where *value* is a floating-point number representing hg flow limits in psi. The example below sets the hg flow alarm maximum value to 52.5 sccm.

```
Send:         set alarm hg flow max 52.5
Receive:      set alarm hg flow max 52.500 ok
```

alarm pressure min
alarm pressure max

These commands report the pressure alarm minimum and maximum value current settings. The example below reports that the pressure alarm minimum value is 800 mmHg.

```
Send:         alarm pressure min
Receive:      alarm pressure min 800.0 mm Hg
```

set alarm pressure min *value*
set alarm pressure max *value*

These commands set the pressure alarm minimum and maximum values to value, where value is a floating-point number representing pressure alarm limits in millimeters of mercury. The example below sets the pressure alarm maximum value to 800 mmHg.

```
Send:         set alarm pressure max 800
Receive:      set alarm pressure max 800.0 ok
```

alarm trig conc hg

This command reports the Hg concentration alarm trigger action for minimum alarm, current setting, to either floor or ceiling. The example below shows the Hg concentration minimum alarm trigger to ceiling, according to **Table 3–3**.

```
Send:         alarm trig conc hg
Receive:      alarm trig conc hg 1
```

set alarm trig conc hg *value*

This command sets the Hg concentrations alarm minimum value, where value is set to either floor or ceiling, according to **Table 3–3**. The example below sets the Hg concentration minimum alarm trigger to ceiling.

```
Send:         set alarm trig conc hg 1
Receive:      set alarm trig conc hg 1 ok
```

Table 3–3. Alarm Trigger Values

Value	**Alarm Trigger**
00	Floor
01	Ceiling

Diagnostics

diag volt mb

This command reports the diagnostic voltage measurements on the motherboard. The sequence of voltages is: positive 24, positive 15, positive 5, positive 3.3, and negative 3.3. Each voltage value is separated by a space.

```
Send:         diag volt mb
Receive:      diag volt mb 24.1 14.9 4.9 3.2 -3.2
```

diag volt mib

This command reports the diagnostic voltage measurements on the measurement interface board. The sequence of voltages is: positive 24, positive 15, negative 15, positive 5, positive 3.3, and positive 15. Each voltage value is separated by a space.

```
Send:         diag volt mib
Receive:      diag volt mb 24.1 14.9 -14.9 4.9 3.2 14.9
```

diag volt iob

This command reports the diagnostic voltage measurements on the optional I/O expansion board. The sequence of voltages is: positive 24, positive 5, positive 3.3, and negative 3.3. Each voltage value is separated by a space.

```
Send:         diag volt iob
Receive:      diag volt iob 24.1 4.9 3.2 -3.2
```

diag volt probe

This command reports the diagnostic voltage measurements on the 82i measurement interface board. The sequence of voltages is: positive 24, positive 15, negative 15, positive 5, positive 3.3, and positive 15. Each voltage value is separated by a space.

```
Send:         diag volt probe
Receive:      diag volt iob 24.1 14.9 -14.8 5.1 3.2 15.1
```

Datalogging

clr records

This command will clear all long and short records that have been saved.

```
Send:         clr records
Receive:      clr records ok
```

set clr lrecs
set clr srecs

These commands will clear only the long records or only the short records that have been saved. The example below clears short records.

```
Send:         set clr srecs
Receive:      set clr srecs ok
```

list din
list dout

These commands report the current selection for the digital outputs in the format: output no | index number | variable name | active state. The active state for digital outputs is open or closed. The active state for digital inputs is high or low.

```
Send:         list dout
Receive:      list dout
output index variable state
 1  3 HG SPAN BIT 2 open
 2  4 HG SPAN BIT 2 open
 3  5 GEN ALARM open
 4  6 STANDBY MODE open
 5  7 INST ZERO MODE open
 6  8 INST SPAN MODE open
 7  9 SYST ZERO MODE open
 8  2 HG SPAN BIT 1 open
 9 15 PRESSURE open
```

list lrec
list srec
list stream
list sp

These commands report the list of current selections for long record logging data, short record logging data, streaming data output, or the scratch pad (sp) list.

The scratch pan is a temporary memory area which is used to set up lists of selections for lrec, srec, or streaming data items. The user can copy any of these lists to the scratch pad, modify individual elements in the list, then save the scratch pad back to the original list. Refer to the "sp field" command for information on how to edit the scratch pad.

The example below shows the list for streaming data output.

```
Send:         list stream
Receive:      list stream
field index variable
x x time
1 1 conc
2 2 syssp
3 3 hgflo
4 4 dlflo
5 5 ctemp
```

er *xy*
lr *xy*
sr *xy*

x = | 0 | 1 | : Reply termination format (see "set format format" command)
y = | 0 | 1 | 2 | : Output format (see "set erec/lrec/srec format format" command)

These commands report the last long and short records stored or the dynamic data record. In the example below, the command requests a long record with no checksum, in ASCII format with text. For details on how to decode the flag fields within these records, see **Figure 3–1** in the "flags" command.

```
Send:         lr01
Receive:      lr01
lr01
09:59 04-13-07 flags 000000 conc 0.000 syssp 2.951 hgflo 17.939 dlflo
10145.800 ctemp 14.020
```

erec
This command returns a snapshot of the main operating conditions (measurments and status) at the time the command is issued. The example that follows shows a typical response.

The format is defined by the current settings of the "format" and "erec format" commands. For details on erec formatting, see the "Record Layout Definition" section at the end of this appendix. For details on how to decode the flag fields within these records, see the "flags" command.

```
Send:         erec
Receive:      erec
05:07 04-10-07 flags 00000068 dilf 0.000 hgf 0.000 tcolr 99.000 tamb
112.190 press 0.000 colrsp 15 tanco 3.000 syslv 0 hgout 0.000
```

erec format
srec format
lrec format
These commands report the output format for long and short records, and dynamic data in various formats such as ASCII without text, ASCII with text, or binary. The example below shows the output format for long records is ASCII with text, according to **Table 3–4**.

```
Send:         lrec format
Receive:      lrec format 1
```

set erec format
set srec format
set lrec format
These commands set the output format for long and short records, and dynamic data, according to **Table 3–4**. The example below sets the long record output format to ASCII with text.

```
Send:         set lrec format 1
Receive:      set lrec format 1 ok
```

Table 3–4. Record Output Formats

Format	**Output Format**
0	ASCII no text
1	ASCII with text
2	binary data

erec layout
lrec layout
srec layout

These commands reports the layout (string indicating the data formats) for data that is sent out in response to the erec, lrec, srec, and related commands The example below shows a typical response. For details on how to interpret the strings, see “Record Layout Definition” later in this appendix.

```
Send:          lrec layout
Receive:       lrec layout %s %s %lx %f %f %f %f %f %f
               t D L ffffff
               flags conc syssp hgflo dlflo ctemp
```

lrec
srec
lrec *xxxx yy*
srec *xxxx yy*
lrec *aa:bb oo-pp-qq yy*
srec *aa:bb oo-pp-qq yy*
xxxx = the number of past records
yy = the number of records to return (1 to 10)
aa = hours (01 to 24)
bb = minutes (01 to 59)
oo = month (01 to 12)
pp = day (01 to 31)
qq = year

These commands output long or short records and dynamic data. The output format is determined by the “set lrec format”, and “set srec format” commands. The logging time is determined by the “set lrec per” and “set srec per” commands.

In the following example, there are 740 long records currently stored in memory. When the command lrec 100 2 is sent, the instrument counts back 100 records from the last record collected (record 740), and then returns two records. For details on how to decode the flag fields within these records, see **Figure 3–1** in the “flags” command.

```
Send:          lrec 100 2
Receive:       lrec 100 2
lrec 100 2
08:27 04-13-07  flags 0000 conc 0.000 syssp 2.951 hgflo 17.939 dlflo
10151.200 ctemp 14.018
08:28 04-13-07  flags 0000 conc 0.000 syssp 2.951 hgflo 17.939 dlflo
10151.200 ctemp 14.047
```

lrec mem size
srec mem size
These commands report the number of lrecs and srecs that can be stored with the current settings and the number of blocks reserved for lrecs and srecs. The example that follows shows that 1075 blocks were reserved for lrecs and the maximum number of lrecs that can be stored in memory is 241979. Memory allocation can be changed using the malloc command.

```
Send:         lrec mem size
Receive:      lrec mem size 241979 recs, 1075 blocks
```

lrec per
srec per
These commands report the long and short records logging period. The example below shows that the short record logging period is 5 minutes.

```
Send:         srec per
Receive:      srec per 5 min
```

set lrec per *value*
set srec per *value*
value = | 1 | 5 | 15 | 30 | 60 |

These commands set the long and short records logging period to *value* in minutes. The example below sets the long record logging period to 15 minutes.

```
Send:         set lrec per 15
Receive:      set lrec per 15 ok
```

no of lrec
no of srec
These commands report the number of long and short records stored in the long and short records memory. The example below shows that 50 long records have been stored in the memory.

```
Send:         no of lrec
Receive:      no of lrec 50 recs
```

malloc lrec
malloc srec
These commands report the currently set memory allocation for long and short records in percent of total memory.

```
Send:         malloc lrec
Receive:      malloc lrec 10 %
```

set malloc lrec *value*
set malloc srec *value*
value = 0 to 100

These commands set the percent of memory space allocated for long and short records to *value*, where *value* is a floating-point number representing percent. The example below sets the memory allocation for long records to 10.

Note Issuing these commands will clear all the logging data memory. All the existing records should be retrieved using appropriate commands, if required. ▲

```
Send:        set malloc lrec 10
Receive:     set malloc lrec 10 ok
```

---

data treatment lrec
data treatment srec
These commands report the current selection of data treatment for concentrations in the long records (lrecs) or short records (srecs). The example that follows reports the data treatment for concentrations in lrec is minimum.

```
Send:        data treatment lrec
Receive:     data treatment lrec min
```

set data treatment lrec *string*
set data treatment srec *string*
string = | cur | avg | min | max |
These commands set the data treatment to *string*, where *string* is current, average, minimum, or maximum for the concentration values recorded in the long records (lrecs) or short records (srecs). The example that follows sets the data treatment for concentrations in lrec to minimum.

```
Send:        set data treatment lrec min
Receive:     set data treatment lrec min ok
```

---

r*xxx* **cooler setpoint**
This command reports the current setpoint of the cooler for the 81*i*-H option. The 81*i*-H has 4 ranges, each with its own cooler setpoint. The example below shows that the cooler setpoint for range 20 is 8 °C.

```
Send:        r20 cooler setpoint
Receive:     r20 cooler setpoint 8 deg C
```

set r*xxx* cooler setpoint *temp*

xxx= | 20 | 30 | 50 | 300 |

temp = integer value between 0 and 19 °C

This command sets the cooler setpoint for the 81*i*-H option according to *xxx*, where *xxx* is one of the 4 ranges, each with its own cooler setpoint. The example below sets the cooler setpoint for range 300 to 18 °C.

```
Send:         set r300 cooler setpoint 18
Receive:      set r300 cooler setpoint 18 ok
```

set copy sp to lrec
set copy sp to srec
set copy sp to stream

These commands copy the current selections in scratch pad (sp) into the long record, short record, or streaming data list.

The scratch pan is a temporary memory area which is used to set up lists of selections for lrec, srec, or streaming data items. The user can copy any of these lists to the scratch pad, modify individual elements in the list, then save the scratch pad back to the original list. Refer to the "sp field" command for information on how to edit the scratch pad.

The example below copies the current list in scratch pad into the long records list.

```
Send:         set copy sp to lrec
Receive:      set copy sp to lrec ok
```

set copy lrec to sp
set copy srec to sp
set copy stream to sp

These commands copy the current contents of the long record, short record, or streaming data list into the scratch pad (sp). These commands are useful in easy modification of current long record, short record, or streaming data lists.

The scratch pan is a temporary memory area which is used to set up lists of selections for lrec, srec, or streaming data items. The user can copy any of these lists to the scratch pad, modify individual elements in the list, then save the scratch pad back to the original list. Refer to the "sp field" command for information on how to edit the scratch pad.

The example below copies the current list of long records into the scratch pad.

```
Send:         set copy lrec to sp
Receive:      set copy lrec to sp ok
```

sp field *number*
This command reports the variable *number* and name stored at index in the scratch pad list.

The scratch pan is a temporary memory area which is used to set up lists of selections for lrec, srec, or streaming data items. The user can copy any of these lists to the scratch pad, modify individual elements in the list, then save the scratch pad back to the original list.

The example below shows that the field 5 in the scratch pad is set to index number 1, which is for the variable for concentration.

```
Send:        sp field 5
Receive:     sp field 5 1 conc
```

set sp field *number value*
number = 1-32 is the maximum number of fields in long and short record lists.

number = 1-18 is for streaming data lists.

This command sets the scratch pad field *number* (item number in scratch pad list) to *value*, where *value* is the index number of a variable in the analog out variable list. Available variables and their corresponding index numbers may be obtained using the command "list var aout". The "set sp field" command is used to create a list of variables which can then be transferred into the long record, short record, or streaming data lists, using the "set copy sp to lrec", "set copy sp to srec", or "set copy sp to stream" commands.

```
Send:        set sp field 1 34
Receive:     set sp field 1 34 ok
```

stream per
This command reports the currently set time interval in seconds for streaming data.

```
Send:        stream per
Receive:     stream per 10 sec
```

set stream per *number value*
number value = | 1 | 2 | 5 | 10 | 20 | 30 | 60 | 90 |120 | 180 | 240 | 300 |
This command sets the time interval between two consecutive streaming data strings to *number value* in seconds. The example below sets the number value to 10 seconds.

```
Send:        set stream per 10
Receive:     set stream per 10 ok
```

stream time
This command reports if the streaming data string will have a time stamp attached to it or not, according to **Table 3–5**.

```
Send:          stream time
Receive:       stream time 1
```

set stream time *value*
This command enables value, where value is to attach or disable time stamp to streaming data string, according to **Table 3–5**.The example below attaches a time stamp to streaming data.

```
Send:          set stream time 0
Receive:       set stream time 0 ok
```

Table 3–5. Stream Time Values

Value	**Stream Time**
00	Attaches time stamp to streaming data string
01	Disables time stamp to streaming data string

Keys/Display

push *button*
button = | do | down | en | enter | he | help | le | left | me | menu | ri | right | ru | run | up | 1 | 2 | 3 | 4 |
These commands simulates pressing the front panel pushbuttons. The numbers represent the front-panel soft keys, from left to right.

```
Send:          push enter
Receive:       push enter ok
```

isc
iscreen
This command retrieves the framebuffer data used for the display on the iSeries instrument. It is 19200 bytes in size, 2-bits per pixel, 4 pixels per byte arranged as 320 by 240 characters. The data is sent in RLE encoded form to save time in transmission. It is sent as a type '5' binary c_link response with no checksum.

The RLE encoding consists of a 0 followed by an 8-bit count of consecutive 0xFF bytes. The following 'c' code will expand the incoming data.

```
void      unpackDisplay ( void far* tdib, unsigned char far* rlescreen )
{
int i,j,k;
unsigned char far *sc4bpp, *sc2bpp, *screen, *ptr;

   ptr = screen = (unsigned char far *)malloc(19200);
   //RLE decode the screen
   for (i=0; i<19200 && (ptr - screen) < 19200; i++)
   {
      *(ptr++) = *(rlescreen + i);
      if (*(rlescreen + i) == 0)
      {
         unsigned char rlecount = *(unsigned char *)(rlescreen + ++i);

         while (rlecount)
         {
            *(ptr++) = 0;
            rlecount--;
         }
      }
      else if (*(rlescreen + i) == 0xff)
      {
         unsigned char rlecount = *(unsigned char *)(rlescreen + ++i);

         while (rlecount)
         {
            *(ptr++) = 0xff;
            rlecount--;
         }
      }
   }
}
```

To convert this data into a BMP for use with windows, it needs to be turned into a 4BPP as that is the smallest windows can display. Also note that BMP files are upside down relative to this data, i.e. the top display line is the last line in the BMP.

---

sc
screen
This command is meant for backward compatibility on the C series. Screen information is reported using the “iScreen” command above.

```
Send:         screen
Receive:      screen This is an I series instrument. Screen
information not available
```

Measurement Configuration

hg span range
This command reports the current span range. The example below reports that the range is 10 μg/m^3, where "10.0" is the range in μg/m^3 and "1" is the first range available for that 81*i* variant according to **Table 3–6**.

```
Send:         hg span range
Receive:      hg span range 10.0 1
```

set hg span range *range*
range = | 1 | 2 | 3 | 4 | 5 | 6 |

This command sets the hg span range, according to **Table 3–6**. The example below sets the hg span range to 20 μg/m^3 for a standard 81*i*.

```
Send:         set hg span range 3
Receive:      set hg span range 3 ok
```

Table 3–6. 81*i* Variant Span Ranges

Standard 81*i*	Low Level 81*i*	High Level 81*i*
Span Range 1 = 5 μg/m^3	Span Range 1 = 1 μg/m^3	Span Range 1 = 20 μg/m^3
Span Range 2 = 10 μg/m^3	Span Range 2 = 2 μg/m^3	Span Range 2 = 30 μg/m^3
Span Range 3 = 20 μg/m^3	Span Range 3 = 5 μg/m^3	Span Range 3 = 50 μg/m^3
Span Range 4 = 30 μg/m^3	Span Range 4 = 10 μg/m^3	Span Range 4 = 300 μg/m^3
Span Range 5 = 40 μg/m^3	Span Range 5 = 20 μg/m^3	
Span Range 6 = 50 μg/m^3		

gas mode
This command reports the current mode. The example below reports that the gas mode is standby.

```
Send:         gas mode
Receive:      gas mode standby
```

set gas mode *mode*
mode = | a cal | a zero | o ca | o zero | ox cal | s cal | s zero | standby |

This command sets the gas mode to a cal, a zero, 0 cal, 0 zero, ox cal, s cal, s zero or standby. The example below sets the gas mode to a zero.

```
Send:        set gas mode a zero
Receive:     set gas mode a zero ok
```

hg span

This command reports the current hg span selection. The example below reports that the hg span selection is span 4.

```
Send:        hg span
Receive:     hg span 4
```

set hg span

This command sets the hg span selection (1-6). The example below sets the hg span selection to span 3.

```
Send:        set hg span 3
Receive:     set hg span 3 ok
```

hg span conc

This command reports the current hg span concentration value followed by the span selection number (0-5). The example below reports that the hg span concentration is 10.0 µg/m^3 and the current span selection is 4 (n+1).

```
Send:        hg span conc
Receive:     hg span conc 10.0 3
```

set hg span conc

This command sets the hg span concentrations. The example below sets span 4 to a hg span concentration to 10.0 µg/m^3.

```
Send:        set hg span conc 10 4
Receive:     set hg span conc 10 4 ok
```

Hardware Configuration

contrast

This command reports the screen's level of contrast. The example below shows the screen contrast is 40%, according to **Table 3–7**.

```
Send:        contrast
Receive:     contrast 8: 40%
```

set contrast level

This command sets the screen's level of contrast, according to **Table 3–7** . The example below sets the contrast level to 50%.

```
Send:        set contrast 10
Receive:     set contrast 10 ok
```

Table 3–7. Contrast Levels

Level	Contrast Level
0	0%
1	5%
2	10%
3	15%
4	20%
5	25%
6	30%
7	35%
8	40%
9	45%
10	50%
11	55%
12	60%
13	65%
14	70%
15	75%
16	80%
17	85%
18	90%
19	95%
20	100%

date

This command reports the current date. The example below reports the date as May 9, 2007.

```
Send:          date
Receive:       date 05-09-07
```

set date *mm-dd-yy*

mm = month
dd = day
yy = year

This command sets the date of the instrument's internal clock. The example below sets the date to March 19, 2007.

```
Send:        set date 03-19-07
Receive:     set date 03-19-07 ok
```

set default params

This command sets all the parameters to their default values. This does not affect the factory-calibrated parameters.

```
Send:        set default params
Receive:     set default params ok
```

save

set save params

This command stores all current parameters in FLASH memory. It is important that each time instrument parameters are changed, that this command be sent. If changes are not saved, they will be lost in the event of a power failure. The example below saves the parameters to FLASH memory.

```
Send:        set save params
Receive:     set save params ok
```

time

This command reports the current time (24-hour time). The example below reports that the internal time is 2:15:30 pm.

```
Send:        time
Receive:     time 14:15:30
```

set time *hh:mm:ss*

hh = hours
mm = minutes
ss = seconds

This command sets the internal clock (24-hour time). The example below sets the internal time to 2:15 pm.

Note If seconds are omitted, the seconds default to 00. ▲

```
Send:        set time 14:15
Receive:     set time 14:15 ok
```

Communications Configuration

addr dns
This command reports the TCP/IP address for the domain name server. The default address is 192.168.1.201, however this can be changed to another address.

```
Send:         addr dns
Receive:      addr dns 192.168.1.201
```

set addr dns address
This command sets the domain name server address, where address consists of four numbers ranging from 0-255 inclusive, separated by ".".

```
Send:         set addr dns 192.168.1. 201
Receive:      set addr dns 192.168.1. 201 ok
```

addr gw
This command reports the default TCP/IP gateway address.

Note This command cannot be used when DHCP is on. Refer to the DHCP command that follows for additional information. ▲

```
Send:         addr gw
Receive:      addr gw 192.168.1. 201
```

set addr gw *address*
This command sets the default gateway *address*, where *address* consists of four numbers ranging from 0-255 inclusive, separated by ".".

```
Send:         set addr gw 192.168.1. 201
Receive:      set addr gw 192.168.1. 201 ok
```

addr ip
This command reports the IP address of the instrument.

Note This command cannot be used when DHCP is on. Refer to the DHCP command that follows for additional information. ▲

```
Send:         addr ip
Receive:      addr ip 192.168.1. 201
```

set addr ip *address*

This command sets the instrument's IP *address*, where *address* consists of four numbers ranging from 0-255 inclusive, separated by ".".

```
Send:          set addr ip 192.168.1. 201
Receive:       set addr ip 192.168.1. 201 ok
```

---

addr nm

This command reports the TCP/IP netmask address.

Note This command cannot be used when DHCP is on. Refer to the DHCP command that follows for additional information. ▲

```
Send:          addr nm
Receive:       addr nm 255.255.255.0
```

set addr nm *address*

This command sets the netmask address, where address consists of four numbers ranging from 0-255 inclusive, separated by ".".

```
Send:          set addr nm 255.255.255.0
Receive:       set addr nm 255.255.255.0 ok
```

---

addr ntp

This command reports the IP address for the NTP time server. See "Network Time Protocol Server" in the "Communications Settings" section of the "Operation" chapter for more information.

```
Send:          addr ntp
Receive:       addr ntp 192.168.1.2
```

set addr ntp *address*

This command sets the NTP time server *address*, where *address* consists of four numbers ranging from 0-255 inclusive, separated by ".".

```
Send:          set addr ntp 192.168.1.2
Receive:       set addr ntp 192.168.1.2 ok
```

---

baud

This command reports the current baud rate for the serial port (RS232/RS485). The example below reports that the current baud rate is 9600.

```
Send:          baud
Receive:       baud 9600
```

set baud *rate*
rate = | 1200 | 2400 | 4800 | 9600 | 19200 | 38400 | 57600 | 115200 |

This command sets the instrument baud *rate*. The example below sets the instrument's baud rate to 115200.

Note After the command is sent, the baud rate of the sending device must be changed to agree with the instrument. ▲

```
Send:        set baud 115200
Receive:     set baud 115200 ok
```

---

dhcp
This command reports the current state of use of DHCP on or off. DHCP is used to assign an IP address to the instrument automatically. The example below shows that DHCP is on.

```
Send:        dhcp
Receive:     dhcp on
```

set dhcp *onoff*
This command enables (*on*) and disables (*off*) the DHCP service. When DHCP is set to on, the instrument gets the IP address, the netmask address, and the gateway address from a DHCP server. When DHCP is set to off, the instrument gets these addresses from system memory.

Note When changing the IP address, the netmask address, or the gateway address, you must cycle power to the instrument before the change takes effect. Until you cycle power, the address assigned by the DHCP server will still be used and reported as the current address. ▲

```
Send:        set dhcp on
Receive:     set dhcp on ok
```

---

format
This command reports the current reply termination format. The example below shows that the reply format is 00, which means reply with no checksum, according to **Table 3–8**.

```
Send:        format
Receive:     format 00
```

set format *format*

This command sets the reply termination *format*, where *format* is set according to **Table 3–8**. The example below sets the reply termination format to checksum.

```
Send:         set format 01
Receive:      set format 01 ok
```

Table 3–8. Reply Termination Formats

Format	Reply Termination
00	<CR>
01	<NL> sum xxxx <CR>

where xxxx = 4 hexadecimal digits that represent the sum of all the characters (bytes) in the message

host name

This command reports the host name string.

```
Send:         host name
Receive:      host name 81i
```

set host name *string*

This command sets the host name *string*, where *string* is 1-13 alphanumeric characters.

```
Send:         set host name 81i
Receive:      set host name 81i ok
```

instr name

This command reports the instrument name.

```
Send:         instr name
Receive:      instr name
              Calibrator
              Calibrator
```

instrument id

This command reports the instrument id.

```
Send:         instrument id
Receive:      instrument id 81
```

set instrument id *value*
This command sets the instrument id to value, where value is a decimal number between 0 and 127 inclusive.

Note Sending this command via RS-232 or RS-485 will require the host to use the new id for subsequent commands. ▲

```
Send:         set instrument id 50
Receive:      set instrument id 50 ok
```

set layout ack
This command disables the stale layout/layout change indicator ('*') that is attached to each response if the layout has changed since the last time erec layout was requested. Refer to **Table 3–9**.

```
Send:         set layout ack
Receive:      set layout ack ok
```

Table 3–9. Set Layout Ack Values

Value	**Function**
0	Do nothing (default)
1	Append "*"

allow mode cmd
This command reports the current allow mode setting: 1 = allow "set mode local" or "set mode remote" commands; 0 = ignore "set mode local" or "set mode remote" commands. Refer to **Table 3–10**. The default value is 0; ignore the commands. The example that follows shows that the instrument is configured to ignore "set mode local" or "set mode remote" commands.

```
Send:         allow mode cmd
Receive:      allow mode cmd 0
```

set allow mode cmd *value*
This command is used to configure the instrument to *value,* where *value* is either 1 = accept or 0 = ignore the "set mode local" or "set mode remote" commands. Refer to **Table 3–10**.

If the instrument is set to accept the commands (*value* = 1), the "set mode local" command will unlock the instrument and the keypad can be used to make changes via the front panel.

If the instrument is set to ignore the commands (*value* = 0), the instrument will respond with "ok" as if the command has been accepted and acted upon, **but will not change the instrument lock status** (this is for compatibility with systems expecting an "ok" response).

Note The instrument will always respond to the command "mode" with the status of the password lock as "mode local" or "mode remote" regardless of the above setting. ▲

The example that follows sets the instrument to accept the "set mode local" or "set mode remote" commands.

```
Send:          set allow mode cmd 1
Receive:       set allow mode cmd 1 ok
```

Table 3–10. Allow Mode Command

Value	**Allow Mode Command**
0	Ignore (default)
1	Accept

mode

This command reports what operating mode the instrument is in: local, service, or remote. The example below shows that the instrument is in the remote mode.

```
Send:          mode
Receive:       mode remote
```

set mode local
set mode remote

These commands set the instrument to local or remote mode. The example below sets the instrument to the local mode.

```
Send:          set mode local
Receive:       set mode local ok
```

power up mode

This command reports the current power up mode setting, where *value,* is either 0 = local/unlocked or 1 = remote/locked. The default value is 0; power up in local/unlocked mode. The example that follows shows that the instrument is configured to power up in the remote/locked mode.

```
Send:         power up mode
Receive:      power up mode 1
```

set power up mode *value*
This command is used to configure the instrument to power up in the local/unlocked mode (*value* = 0) or the remote/locked mode (*value* = 1) as indicated in **Table 3–11**.

If the instrument is set to power up in the local/unlocked mode, the keypad can be used to make changes via the front panel. If the instrument is set to power up in the remote/locked mode, changes can not be made from the front panel. The example that follows sets the instrument to power up in remote/locked mode.

```
Send:         set power up mode 1
Receive:      set power up mode 1 ok
```

Table 3–11. Power Up Mode Values

Value	**Power Up Mode Command**
0	Local/Unlocked (default)
1	Remote/Locked Mode

program no
This command reports the instrument's model information and program version number, which will be dependant on the current version.

```
Send:         program no
Receive:      program no iSeries 81i 00.04.54.058
```

tz
This command reports the "tz" timezone string for the NTP server. See "Network Time Protocol Server" in the "Communications Settings" section of the "Operation" chapter for more information.

```
Send:         tz
Receive:      tz EST+5EDT
```

set tz *string*
This command sets the timezone *string* for the instrument for use with the NTP time server, where *string* is a standard timezone string. Common strings are listed in the timezone screen description in the "Operation chapter".

```
Send:        set tz EST+5EDT
Receive:     set tz EST+5EDT ok
```

I/O Configuration

analog iout range *channel*

This command reports the analog current output range setting for *channels*, where *channel* must be between 1 and 6, inclusive. The example below reports current output channel 4 to the 4-20 mA range, according to **Table 3–12**. This command responds with "feature not enabled" if the optional I/O expansion board is not detected.

```
Send:        analog iout range 4
Receive:     analog iout range 4 2
```

set analog iout range *channel range*
This command sets analog current output *channel* to the *channel range* where *channel* is between 1 and 6 inclusive, and range is set according to **Table 3–12**. The example below sets current output channel 4 to the 0-20 mA range. This command responds with "feature not enabled" if the optional I/O expansion board is not detected.

```
Send:        set analog iout range 4 1
Receive:     set analog iout range 4 1 ok
```

Table 3–12. Analog Current Output Range Values

Range	Output Range
1	0-20 mA
2	4-20 mA
0 [cannot be set to this, but may report]	Undefined

analog vin *channel*
This command retrieves the analog voltage input *channel* data, both the calculated value and the actual voltage. In the example below, the "calculated" value of channel 1 is 75.325 degrees F, volts are 2.796. This command responds with "feature not enabled" if the optional I/O expansion board is not detected.

```
Send:        analog vin 1
Receive:     analog vin 1 75.325 2.796
```

analog vout range *channel*

This command reports the analog voltage output *channel* range, where *channel* is between 1 and 6 inclusive, according to **Table 3–13**.

```
Send:         analog vout range 2
Receive:      analog vout range 2 3
```

set analog vout range *channel range*

This command sets analog voltage output *channel* to the range, where *channel* is between 1 and 6 inclusive, and *range* is set according to **Table 3–13**. The example below sets channel 2 to the 0-10 V range.

```
Send:         set analog vout range 2 3
Receive:      set analog vout range 2 3 ok
```

Table 3–13. Analog Voltage Output Range Values

Range	Output Range
1	0-1 V
2	0-100 mV
3	0-10 V
4	0-5 V
0 [cannot be set to this, but may report]	Undefined

dig in

This command reports the status of the digital inputs as a 4-digit hexadecimal string with the most significant bit (MSB) being input 16.

```
Send:         dig in
Receive:      dig in 0xffff
```

din *channel*

This command reports the action assigned to input *channel* and the corresponding active state. The example below reports the input 5 to be assigned an index number 5 corresponding to action of "filter zero" with the active state being high.

```
Send:         din 5
Receive:      din 5 5 FILTER ZERO high
```

set din *channel index state*

This command assigns digital input *channel* (1-16) to activate the action indicated by *index* (1-35), when the input transitions to the designated *state* (high or low). Use "list din var" command to obtain the list of supported *index* values and corresponding actions.

```
Send:         set din 1 3 high
Receive:      set din 1 3 high ok
```

dout *channel*

This command reports the index number and output variable and the active state assigned to output *channel*. The example below reports the input 4 to be assigned an index number 11 corresponding to "general alarm" with the active state being open.

```
Send:         dout 4
Receive:      dout 4 11 GEN ALARM open
```

set dout *channel index state*

This command assigns digital output *channel* to be assigned to the action associated with *index*, and assigns it an active *state* of state (open or closed).

```
Send:         set dout 4 11 open
Receive:      set dout 4 11 open ok
```

dtoa *channel*

This reports the outputs of the 6 or 12 digital to analog converters, according to **Table 3–14**. The example below shows that the DAC 1 is 97.7% full-scale.

```
Send:         dtoa 1
Receive:      dtoa 1 97.7%
```

Note All channel outputs are user definable. If any customization has been made to the analog output configuration, the default selections may not apply. ▲

Table 3–14. Default Output Assignment

D to A	Assignment	Function
1	Hg CONC	Voltage Output
2	Hg SPAN	Voltage Output
3	Hg FLOW	Voltage Output
4	DIL FLOW	Voltage Output
5	COOLER TEMP	Voltage Output
6	NONE	Voltage Output
7	Hg CONC	Current Output
8	Hg SPAN	Current Output
9	Hg FLOW	Current Output
10	DIL FLOW	Current Output
11	COOLER TEMP	Current Output

D to A	Assignment	Function
12	NONE	Current Output

list var aout
list var dout
list var din
These commands report the list of index numbers, and the variables (associated with that index number) available for selection in the current mode for analog output, digital output and digital inputs. The index number is used to insert the variable in a field location in a list using "set sp field index". The example below reports the list of analog output, index numbers, and variables.

```
Send:        list var aout
Receive:     list var aout
```

Record Layout Definition

The Erec, Lrec, and Srec Layouts contain the following:

- A format specifier for parsing ASCII responses
- A format specifier for parsing binary responses

In addition to these, the Erec Layout contains:

- A format specifier for producing the front-panel displays

In operation, values are read in using either the ASCII or binary format specifiers and converted to uniform internal representations (32-bit floats or 32-bit integers). These values are converted into text for display on the screen using the format specifier for the front-panel display. Normally, the specifier used to parse a particular datum from the input stream will be strongly related to the specifier used to display it (e.g., all of the floating point inputs will be displayed with an 'f' output specifier, and all of the integer inputs will be displayed with a 'd' specifier).

Format Specifier for ASCII Responses

The first line of the Layout response is the scanf-like parameter list for parsing the fields from an ASCII ERec response. Parameters are separated by spaces and the line is terminated by a \n (the normal line separator character). Valid fields are:

%s - parse a string

%d - parse a decimal number

%ld - parse a long (32-bit) decimal number

%f - parse a floating point number

%x - parse a hexadecimal number

%lx - parse a long (32-bit) hex number

%* - ignore the field

Note Signed versus unsigned for the integer values does not matter; it is handled automatically. ▲

Format Specifier for Binary Responses

The second line of the Layout response is the binary parameter list for parsing the fields from a binary response. Parameters MUST be separated by spaces, and the line is terminated by a '\n'. Valid fields are:

t - parse a time specifier (2 bytes)

D - parse a date specifier (3 bytes)

i - ignore one 8-bit character (1 byte)

e - parse a 24-bit floating point number (3 bytes: n/x)

E - parse a 24-bit floating point number (3 bytes: N/x)

f - parse a 32-bit floating point number (4 bytes)

c - parse an 8-bit signed number (1 byte)

C - parse an 8-bit unsigned number (1 byte)

n - parse a 16-bit signed number (2 bytes)

N - parse a 16-bit unsigned number (2 bytes)

m - parse a 24-bit signed number (3 bytes)

M - parse a 24-bit unsigned number (3 bytes)

l - parse a 32-bit signed number (4 bytes)

L - parse a 32-bit unsigned number (4 bytes)

There is an optional single digit d which may follow any of the numeric fields which indicates that after the field has been parsed out, the resulting value is to be divided by 10^d. Thus the 16-bit field 0xFFC6 would be interpreted with the format specifier 'n3' as the number -0.058.

Format Specifier for Front-Panel Layout

The subsequent lines in the ERec Layout response describe the appearance of the full panel. The full instrument panel as it appears on the screen has two columns of lines. Each line is composed of three major components: (1) a text field, (2) a value field, and (3) a button. None of these three components is required. The text field contains statically displayed text.

The value field displays values which are parsed out of the response to a DATA/ERec command. It also displays, though background changes, alarm status. The button, when pressed, triggers input from either a dialog box or a selection list. There are five kinds of buttons, B, I, L, T, and N.

Each line in the layout string corresponds to one line on the display. The layout string describes each of the three major fields as well as translation mechanisms and corresponding commands.

Text

The first field in the layout string is the text. It is delimited by a ':'. The string up to the first ':' will be read and inserted in the text field of the line.

Value String

This is followed by a possible string, enclosed in quotes. This is used to place a string into the value field.

Value Source

The value source, which is the item (or word) number in the DATA/ERec response, appears next. This is followed by an optional bitfield designator. The datum identified by the value source can be printed as a string 's', hexadecimal 'x', decimal 'd', or floating point 'f', or binary 'b' number. Typically, bitfield extractions are only done for decimal or hexadecimal numbers.

Floating-point numbers can be followed with an optional precision specifier which will be used as an argument to printf's %f format (e.g., a field of '4' would be translated into the printf command of '%.3f'). Alternately, the special character '*' can precede the precision specifier; this causes an indirection on the precision specifier (which now becomes a field number).

This is useful when formatting, for example, numbers which have varying precision depending on the mode of the instrument.

Binary numbers can also have an optional precision specifier which is used to determine how many bits to print. For example, the specifier 'b4' will print the lowest four bits of the parsed number.

There are serious restrictions on where an 's' field may appear: currently sources 1 and 2 must be 's', and no others may be 's'.

Alarm Information

The value source is followed by optional alarm information, indicated by a commercial at sign '@' with a source indicator and a starting bit indicator. All alarm information is presumed to be two bits long (low and high). The bitfield extraction is performed on the integer part of the source. Typical alarm information would appear as '@6.4'.

Translation Table

Then, there appears an optional translation table within braces '{}'. This is a string of words separated by spaces. An example translation table would be '{Code_0 Code_1 Code_2 Code_3}'. The value, once extracted is used as a zero-based index into the translation table to determine the string to display.

Selection Table

Then there appears an optional selection table within parentheses '(...)'. This is a string of numbers separated by spaces '(0 1)'. The selection table lists the translation table entries which the user may select from when setting the parameter. This is not necessarily the same as the entries which may be displayed.

Button Designator

Then there appears an optional button designator. This will be one of 'B', 'I', 'L', 'T', or 'N'.

B- Indicates a button which pops up an input dialog prompting the user for a new value using the designated input format. The input format is specified from the 'B' through the subsequent semicolon.

I—Indicates a button which pops up a selection list with input translation. That is, the values read are translated before they are compared to the selection list options.

L—Indicates a button which pops up a selection list without any translation. The output value is number of the selected option.

T—Indicates a button which pops up a selection list with output translation. The number of the option selected is used as an index into the translation table to generate an output string.

N—Indicates a button which only sends the subsequent command to the instrument. No user-prompting happens.

The following string through an optional '|' or the end of the line is the command which is to be sent to the instrument upon the completion of the button selection. The command string should normally contain print-style formatting to include the user input. If a '|' is present, it indicates a command which is sent to the instrument upon successful completion of the button command to update the value field.

Examples Some examples ('\n' is the C syntax for an end-of-line character):

'Concentrations\n'

This is a single text-only line.

'\n'

This is a single blank line.

' hg0:3s\n'

This is a line which appears slightly indented. The text field is 'Hg0', the value is taken from the third element of the data response, and interpreted as a string.

' hg0:18sBd.ddd;set hg0 coef %s\n'

This is a line which also appears slightly indented. The next field is also 'Hg0', but the value is taken from the eighteenth element of the data response, again interpreted as a string. A button appears on this line which, when pressed, pops up an input dialog which will state "Please enter a new value for Hg0 using a d.ddd format." The string entered by the user is used to construct the output command. If the user enters, for example, '1.234', the constructed command will be 'set Hg0 coef 1.234'.

' hg0:21f{Code_0 Code_1 Code_2 Code_3 Code_4 Code_5 Code_6 Code_7 Code_8 Code_9 Code_10 Code_11}Lset range hg0 %d\n'

This is a line which appears slightly indented, the title is again 'Hg0', and the value the twenty-first element of the data response, interpreted as a floating-point number. There is a no-translation button which creates a selection list of twelve "Code nn" options. The number of the user selection is used to create the output command.

'Mode:6.12-13x{local remote service service}(0 1)Tset mode %s\n'

This is a line which has a title of 'Mode', and value taken from the sixth field of the data response. There is a bitfield extraction of bits 12 through 13 from the source (the value type is not important here because the value is being translated to an output string). Once the bits have been extracted, they are shifted down to the bit-zero position. Thus, the possible values of this example will be 0 through 3. The translation list shows the words which correspond to each input value, the zeroth value appearing first (0 -> local, 1 -> remote, etc.). The selection list shows that only the first two values, in this case, are to be shown to the user when the button is pressed. The 'T' button indicates full translation, input code to string, and user selection number to output string.

'\xC'

This is a line that starts a new column (the \xC or ^L),

' Comp:6.11x{off on}Tset temp comp %s\n'

This shows that the bitfield end (the second part of a bitfield specification) is optional. The bitfield will be one bit long, starting in this case at the eleventh bit.

'Background:7f*8Bd.ddd;set o3 bkg %s\n'

This shows the use of indirect precision specifiers for floating point displays. The background value is taken from the 7th element, and the precision specifier is taken from the 8th. If the asterisk were not present, it would indicate instead that 8 digits after the decimal point should be displayed.

Chapter 4
81*i* MODBUS Protocol

This appendix provides a description of the MODBUS Protocol Interface and is supported both over RS-232/485 (RTU protocol) as well as TCP/IP over Ethernet.

The MODBUS Commands that are implemented are explained in detail in this document. The MODBUS protocol support for the iSeries enables the user to perform the functions of reading the various concentrations and other analog values or variables, read the status of the digital outputs of the instrument, and to trigger or simulate the activation of a digital input to the instrument. This is achieved by using the supported MODBUS commands listed below.

For details of the Model 81*i* MODBUS Protocol specification, see the following topics:

- "Serial Communication Parameters" on page 4-1
- "TCP Communication Parameters" on page 4-2
- "Application Data Unit Definition" on page 4-2
- "Function Codes" on page 4-3
- "MODBUS Addresses Supported" on page 4-8.

Additional information on the MODBUS protocol can be obtained at http://www.modbus.org. References are from MODBUS Application Protocol Specification V1.1a MODBUS-IDA June 4, 2004.

Serial Communication Parameters

The following are the communication parameters that are used to configure the serial port of the *i*Series to support MODBUS RTU protocol.

Number of Data bits : 7 or 8

Number of Stop bits : 1 or 2

Parity : None, Odd, or Even

Data rate : 1200 to 115200 Baud (9600 is default)

TCP Communication Parameters

*i*Series instruments support the MODBUS/TCP protocol. The register definition is the same as for the serial interface. Up to three simultaneous connections are supported over Ethernet.

TCP connection port for MODBUS: 502

Application Data Unit Definition

Here are the MODBUS ADU (Application Data Unit) formats over serial and TCP/IP:

Serial:	Slave Address	Function Code	Data	Error Check
TCP/IP:	MBAP Header	Function Code	Data	

Slave Address

The MODBUS slave address is a single byte in length. This is the same as the instrument ID used for C-Link commands and can be between 1 and 127 decimal (i.e. 0x01 hex to 0x7F hex). This address is only used for MODBUS RTU over serial connections.

Note Device ID ‘0’ used for broadcast MODBUS commands, is not supported. Device IDs 128 through 247 (i.e. 0x80 hex to 0xF7 hex) are not supported because of limitations imposed by C-Link. ▲

MBAP Header

In MODBUS over TCP/IP, a MODBUS Application Protocol Header (MBAP) is used to identify the message. This header consists of the following components:

Transaction Identifier	2 Bytes	0x0000 to 0xFFFF (Passed back in response)
Protocol Identifier	2 Bytes	0x00 (MODBUS protocol)
Length	2 Bytes	0x0000 to 0xFFFF (Number of following bytes)
Unit Identifier	1 Byte	0x00 to 0xFF (Passed back in response)

A Slave address is not required in MODBUS over TCP/IP because the higher-level protocols include device addressing. The unit identifier is not used by the instrument.

Function Code

The function code is a single byte in length. The following function codes are supported by the instrument:

Read Coils	:	0x01
Read Inputs	:	0x02
Read Holding Registers	:	0x03
Read Input Registers	:	0x04
Force (Write) Single Coil	:	0x05
Read Exception Status	:	0x07

If a function code is received that is not in this list, and invalid function exception is returned.

Data

The data field varies depending on the function. For more description of these data fields, see "Function Codes" below.

Error Check

In MODBUS over Serial an error check is included in the message. This is not necessary in MODBUS over TCP/IP because the higher-level protocols ensure error-free transmission. The error check is a two-byte (16-bit) CRC value.

Function Codes

This section describes the various function codes that are supported by the Model 81*i*.

(0x01/0x02) Read Coils / Read Inputs

Read Coils/Inputs reads the status of the digital outputs (relays) in the instrument. Issuing either of these function codes will generate the same response.

These requests specify the starting address, i.e. the address of the first output specified, and the number of outputs. The outputs are addressed starting at zero. Therefore, outputs numbered 1–16 are addressed as 0–15.

The outputs in the response message are packed as one per bit of the data field. Status is indicated as 1 = Active (on) and 0 = Inactive (off). The LSB of the first data byte contains the output addressed in the query. The other outputs follow toward the high order end of this byte, and from low order to high order in subsequent bytes. If the returned output quantity is not a multiple of eight, the remaining bits in the final data byte will be padded with zeros (toward the high order end of the byte). The Byte Count field specifies the quantity of complete bytes of data.

Note The values reported may not reflect the state of the actual relays in the instrument, as the user may program these outputs for either active closed or open. ▲

Request

Function code	1 Byte	0x01 or 0x02
Starting Address	2 Bytes	0x0000 to maximum allowed by instrument
Quantity of outputs	2 Bytes	1 to maximum allowed by instrument
Unit Identifier	1 Byte	0x00 to 0xFF (Passed back in response)

Response

Function code	1 Byte	0x01 or 0x02
Byte count	1 Byte	N*
Output Status	n Byte	N = N or N+1

*N = Quantity of Outputs / 8, if the remainder not equal to zero, then N=N+1

Error Response

Function code	1 Byte	0x01 or 0x02
Exception code	1 Byte	01=Illegal Function, 02=Illegal Address, 03=Illegal Data, 04=Slave Device Failure

Here is an example of a request and response to read outputs 2–15:

Request

Field Name	*(Hex)*
Function	0x01
Starting Address Hi	0x00
Starting Address Lo	0x02
Quantity of Outputs Hi	0x00
Quantity of Outputs Lo	0x0D

Response

Field Name	*(Hex)*
Function	0x01
Byte Count	0x03
Output Status 2–10	0xCD
Output Status 11–15	0x0A

The status of outputs 2–10 is shown as the byte value 0xCD, or binary 1100 1101. Output 10 is the MSB of this byte, and output 2 is the LSB. By convention, bits within a byte are shown with the MSB to the left, and the LSB to the right. Thus the outputs in the first byte are '10 through 2', from left to right. In the last data byte, the status of outputs 15-11 is shown as the byte value 0x0A, or binary 0000 1010. Output 15 is in the fifth bit position from the left, and output 11 is the LSB of this byte. The four remaining high order bits are zero filled.

(0x03/0x04) Read Holding Registers / Read Input Registers

Read holding/input registers reads the measurement data from the instrument. Issuing either of these function codes will generate the same response. These functions read the contents of one or more contiguous registers.

These registers are 16 bits each and are organized as shown below. All of the values are reported as 32-bit IEEE standard 754 floating point format. This uses 2 sequential registers, least significant 16 bits first.

The request specifies the starting register address and the number of registers. Registers are addressed starting at zero. Therefore registers numbered 1–16 are addressed as 0–15. The register data in the response message are packed as two bytes per register, with the binary contents right justified within each byte. For each register, the first byte contains the high order bits and the second contains the low order bits.

The status of outputs 2–10 is shown as the byte value 0xCD, or binary 1100 1101. Output 10 is the MSB of this byte, and output 2 is the LSB. By convention, bits within a byte are shown with the MSB to the left, and the LSB to the right. Thus, the outputs in the first byte are '10 through 2', from left to right. In the last data byte, the status of outputs 15-11 is shown as the byte value 0x0A, or binary 0000 1010. Output 15 is in the fifth bit position from the left, and output 11 is the LSB of this byte. The four remaining high order bits are zero filled.

The request specifies the starting register address and the number of registers. Registers are addressed starting at zero. Therefore, registers numbered 1–16 are addressed as 0–15. The register data in the response message are packed as two bytes per register, with the binary contents right justified within each byte. For each register, the first byte contains the high order bits and the second contains the low order bits.

Request

Function code	1 Byte	0x03 or 0x04
Starting Address	2 Bytes	0x0000 to maximum allowed by instrument
Quantity of Registers	2 Bytes	1 to maximum allowed by instrument

Response

Function code	1 Byte	0x03 or 0x04
Byte count	1 Byte	2 x N*
Register value	N* x 2 Bytes	N = N or N+1

*N = Quantity of Registers

Error Response

Function code	1 Byte	Function code + 0x80
Exception code	1 Byte	01=Illegal Function, 02=Illegal Address, 03=Illegal Data, 04=Slave Device Failure

Here is an example of a request and response to read registers 10–13:

Request

Field Name	*(Hex)*
Function	0x03
Starting Address Hi	0x00
Starting Address Lo	0x09
No. of Registers Hi	0x00
No. of Registers Lo	0x04

Response

Field Name	*(Hex)*
Function	0x03
Byte Count	0x06
Register value Hi (10)	0x02

Register value Lo (10)	0x2B
Register value Hi (11)	0x00
Register value Lo (11)	0x00
Register value Hi (12)	0x00
Register value Lo (12)	0x64
Register value Hi (13)	0x00
Register value Lo (13)	0x64

The contents of register 10 are shown as the two byte values of 0x02 0x2B. Then contents of registers 11–13 are 0x00 0x00, 0x00 0x64 and 0x00 0x64 respectively.

(0x05) Force (Write) Single Coil

The force (write) single coil function simulates the activation of the digital inputs in the instrument, which triggers the respective action.

This function code is used to set a single action to either ON or OFF. The request specifies the address of the action to be forced. Actions are addressed starting at zero. Therefore, action number 1 is addressed as 0. The requested ON/OFF state is specified by a constant in the request data field. A value of 0xFF00 requests the action to be ON. A value of 0x0000 requests it to be OFF. All other values are illegal and will not affect the output. The normal response is an echo of the request, returned after the state has been written.

Note This function will not work if the instrument is in service mode. ▲

Request

Function code	1 Byte	0x05
Starting Address	2 Bytes	0x0000 to maximum allowed by instrument
Output Value	2 Bytes	0x0000 or 0xFF00

Response

Function code	1 Byte	0x05
Starting Address	2 Bytes	0x0000 to maximum allowed by instrument
Output Value	2 Bytes	0x0000 or 0xFF00

Error Response

Function code	1 Byte	Function code + 0x80

Exception code	1 Byte	01=Illegal Function, 02=Illegal Address, 03=Illegal Data, 04=Slave Device Failure

Here is an example of a request to write Coil 5 ON:

Request

Field Name	*(Hex)*
Function	05
Output Address Hi	00
Output Address Lo	05
Output Value Hi	FF
Output Value Lo	00

Response

Field Name	*(Hex)*
Function	05
Output Address Hi	00
Output Address Lo	05
Output Value Hi	FF
Output Value Lo	00

MODBUS Addresses Supported

Table 4–1 through **Table 4–3** list the MODBUS addresses supported for the Model 81*i*.

IMPORTANT NOTE The addresses in the following tables are Protocol Data Unit (PDU) addresses. Verify the coil number on your MODBUS master to ensure that it matches the coil number on the instrument. ▲

Note Coil status 1 indicates active state. ▲

Table 4–1. Read Coils for 81*i* (Digital Output)

Coil Number	Status	Used Exclusively In
0	Invalid	
1	SERVICE	
2	HG SPAN BIT 1	
3	HG SPAN BIT 2	

Coil Number	Status	Used Exclusively In
4	HG SPAN BIT 3	
5	GENERAL ALARM	
6	STANDBY MODE	
7	INSTRUMENT ZERO MODE	
8	INSTRUMENT SPAN MODE	
9	ORIFICE ZERO MODE	
10	ORIFICE SPAN MODE	
11	SYSTEM ZERO MODE	
12	SYSTEM SPAN MODE	
13	COOLER TEMPERATURE ALARM	
14	AMBIENT TEMPERATURE ALARM	
15	PRESSURE ALARM	
16	Hg FLOW ALARM	
17	DILUTION FLOW ALARM	
18	CONCENTRATION ALARM	
19	MOTHERBOARD STATUS ALARM	
20	MEASUREMENT INTERFACE BOARD STATUS ALARM	
21	I/O BOARD STATUS ALARM	I/O Expansion Board Option
22	LOCAL/REMOTE	
23	EXT ALARM 1	
24	EXT ALARM 2	
25	EXT ALARM 3	

IMPORTANT NOTE The addresses in the following tables are Protocol Data Unit (PDU) addresses. Verify the register number on your MODBUS master to ensure that it matches the register number on the instrument. ▲

Note For additional information on how to read registers and interpret the data, refer to the "(0x03/0x04) Read Holding Registers / Read Input Registers" section in this appendix. ▲

Table 4–2. Read Registers for 81*i* (Analog Output)

Register Number	Variable	Used Exclusively In
0	Invalid	
1&2	Hg CONCENTRATION	
3&4	Hg SPAN	
5&6	Hg FLOW	
7&8	DILUTION FLOW	
9&10	COOLER TEMPERATURE	
11&12	AMBIENT TEMPERATURE	
13&14	PRESSURE	
15&16	COOLER SET TEMPERATURE	
17&18	ANALOG IN 1	I/O Expansion Board Option
19&20	ANALOG IN 2	I/O Expansion Board Option
21&22	ANALOG IN 3	I/O Expansion Board Option
23&24	ANALOG IN 4	I/O Expansion Board Option
25&26	ANALOG IN 5	I/O Expansion Board Option
27&28	ANALOG IN 6	I/O Expansion Board Option
29&30	ANALOG IN 7	I/O Expansion Board Option
31&32	ANALOG IN 8	I/O Expansion Board Option
33&34	EXT ALARMS	
35&36	Hg RANGE	

IMPORTANT NOTE The addresses in the following tables are Protocol Data Unit (PDU) addresses. Verify the coil number on your MODBUS master to ensure that it matches the coil number on the instrument. ▲

Note Writing 1 to the coil number shown in the following table will initiate the “action triggered” listed in the table. This state must be held for at least 1 second to ensure the instrument detects the change and triggers the appropriate action. ▲

Note The coils within each coil group in the following table are mutually exclusive and will not be triggered if there is a conflict. Before you assert (1) one coil in a group, make sure the other coils in the group are de-asserted (0). ▲

Table 4–3. Write Coils for 81*i* (Digital Input)

Coil Number	Action Triggered	Coil Group	Used Exclusively In
100	Invalid		
101	INSTRUMENT ZERO MODE	Zero Span Mode	
102	INSTRUMENT CAL MODE	Zero Span Mode	
103	ORIFICE ZERO MODE	Zero Span Mode	
104	ORIFICE CAL MODE	Zero Span Mode	
105	SYSTEM ZERO MODE	Zero Span Mode	
106	SYSTEM CAL MODE	Zero Span Mode	
107	STANDBY		
108	AOUTS TO ZERO	Analog Out Test	I/O Expansion Board Option
109	HG SPAN BIT 1	Span Level	
110	HG SPAN BIT 2	Span Level	
111	HG SPAN BIT 3	Span Level	
112	AOUTS TO FS	Analog Out Test	I/O Expansion Board Option
113	EXT ALARM 1		
114	EXT ALARM 2		
115	EXT ALARM 3		
116	RANGE 20	81*i*-H Range	81*i*-H
117	RANGE 30	81*i*-H Range	81*i*-H
118	RANGE 50	81*i*-H Range	81*i*-H
119	RANGE 300	81*i*-H Range	81*i*-H

USA
27 Forge Parkway
Franklin, MA 02038
Ph: (508) 520-0430
Fax: (508) 520-2800
orders.aqi@thermofisher.com

India
C/327, TTC Industrial Area
MIDC Pawane
New Mumbai 400 705, India
Ph: +91 22 4157 8800
india@thermofisher.com

China
Beijing Silver Tower
#2 DongSanHuan North Rd
Beijing, China, 100020
Ph: +86 10 84193588
info.eid.china@thermofisher.com

Europe
Ion Path, Road Three,
Winsford, Cheshire CW73GA UK
Ph: +44 1606 548700
Fax: +44 1606 548711
sales.epm.uk@thermofisher.com

Find out more at **thermofisher.com/EUmercury**

ThermoFisher SCIENTIFIC