Preliminary Release

July 20, 2004

CPP-4794 Users Manual

H2NS Environmental Inc.

P.O. Box 500207, Austin, Texas, 78750
Phone (512) 918-8035 Fax (512) 9128-9684 email: sales@h2ns.com

Table of Contents

Table of Figures

1 Introduction

1.1 General

The CPP-4794 introduces a new data collection process to industry. The unit is a multi-user, multi-tasking device that takes advantage of today's modern instrumentation's “smart interfaces” to communicate with and collect data from them. The user is also granted access to the instrumentation in an **On Line** Mode. In an off line mode, the user can communicate directly with connected instrumentation, as if the CPP were not there. This allows manufacturer supplied diagnostic routines to be conducted remotely.

To fully support the new features provided by today's instrumentation, the CPP has departed from the traditional binary processing of data and instead processes data with floating point mathematics. It is recommended that the new user fully read this introduction section to gain an overall understanding of the capabilities of the CPP. This will make subsequent use of the CPP and systems configurations flow more smoothly. Each topic discussed in this introduction is presented in more detail elsewhere in the manual.

The CPP-4794 is a digital signal processor based Communications Protocol Processor and data acquisition system. The unit was designed to allow operators to take advantage of the serial interfaces now being offered by all major instrument manufacturers. Over the serial port, in addition to the collection of data, the instrument manufacturers offer a host of other information including status, alarming, control and diagnostic commands and features. Existing analog data loggers and PLC's can not take advantage of the features offered by newer instrumentation, and in fact restrict a user's ability to reach the smart serial port of the instruments.

As all manufacturers have different needs in controlling and setting up their instrumentation, they have all developed different protocols for interfacing to their instruments. Some device is needed to assimilate this information in the varying formats and convert it into a single format for a central processing unit. In addition, an operator should be able to access the serial port of the instrumentation to conduct diagnostics, and the like, without stopping the collection of data. To achieve this, the basic CPP contains four independent (RS-232) communications ports. With the installation of comm expansion boards, an additional 8 independent comm ports can be added to the system thereby providing a total of 12 independent comm ports.

H2NS offers a number of technical notes and application notes that describe and support various features of the CPP. These are available upon request and can be downloaded from the company web site, h2ns.com.

1.2 Interrupts

Independent comm ports are required to provide a product that can interface to the wide spectrum of instrumentation available today. Some instrumentation is designed to be polled, whereas other instrumentation operates in a broadcast mode, *i.e.,* sending out strings of data at specific intervals. Not having independent comm ports for these broadcast inputs would result in data loss. To control this number of independent comm ports required that the CPP be designed as a true multi-user, multitasking device. Every function is interrupt driven

1.3 System Initialization

The CPP is initialized over comm port #4 or comm port #1. Operating configurations can also be uploaded to a central computer, and downloaded into the CPP over comm port number 1. Configurations can also be transferred among CPP's using the Compact Flash Memory (CFM) card.

Entering an **I** key command brings up the Main Initialization Menu. An example is provided below. The numbers associated with each initialization section, along with the length of the menu, may change from system to system depending on the software options installed in the unit. If the unit supports a password, and a password has been set up, at this point the correct password must be entered before the system will proceed further.

```
I <enter>
                              Main Menu

01 - Password                        02 - Station Name I/D
03 - Averaging Intervals             04 - # Chans Enabled
05 - Channel Set Up                  06 - Data Validity
07 - Sample Delay                    08 - Init Sequencers
09 - I/O Labels                      10 - Cal Set Up
11 - Boolean Set Up                  12 - Alarm Set Up
13 - Auto Prints                     14 - Init Met Parameters
15 - Data Print Format               16 - CFM Recordings
17 - External I/O                    18 - Curve Fit
19 - Set Up Calibrators              20 - Hold Last Value
21 - DO To DAC                       22 - Modbus Set Up
23 – Poller Inactivity

Choice =
```

Figure 1-1 Main Initialization Menu

Inputting a carriage return at this point exits the initialization routine. Any time the initialization routine is exited, the CPP recalculates all interrelated functions and calculates and stores checksums for initialized parameters. Selecting one of the numbers enters the selected initialization routine. Upon exiting the initialization routine, the CPP goes to a routine that calculates and sets up related system operating conditions. If changes made in one section affect another, they are accounted for by the CPP.

1.4 Analog to Digital Section

Since all instrumentation is not yet smart, H2NS also offers an analog to digital front end for the CPP. The use of serial or A/D inputs can be mixed and matched in the setup of the CPP.

The CPP supports two types of ADC's, isolated inputs and non-isolated inputs. The front end multiplexer of the A/D is optically isolated from the CPP and switches both signal and ground lines. The operator entered full scale and zero values are important when using an A/D input as they are used to convert the binary number received from the ADC into a floating point number. Each channel is digitized once a second.

1.5 Typical System Configuration

Since the CPP can be connected to instrumentation in a variety of ways. The purpose of this discussion is to illustrate some configurations of the comm ports. The manufacturers Ma, Mb, Mc, Md, and Me represent different manufacturers. Figure 1-2 below presents a typical comm port set up.

Channel #	Parameter	Manufacturer	Comm Port
1	SO2	Ma	3-00
2	NOX	Ma	3-00
3	Opacity	Mb	3-01
4	Flow #1	Mc	3-02
5	CO - Dual Range	Md	3-03
6	O2	Md	3-04
7	NOX - Corrected	Ma	Comp
8	Temperature	Me	3-C2
9	Wind Direction	Me	3-C2
10	Wind Speed	Me	3-C2
11	Flow #2	Mc	A/D

Figure 1-2 Typical Comm Port Assignments

In this setup both instruments for channels 1 and 2 are supplied by manufacturer Ma and both channels 1 and 2 are using comm port # 3-00 to interface to the instruments. This configuration is useable only when the manufacturer supports a multidrop, or tri-stating configuration. It is always better to connect only one instrument to a comm port unless system constraints dictate otherwise. If one unit is taken off line (in the pass through mode) only the instrument connected to that comm port is affected. The CPP continues to collect data from all other instruments. If two instruments are connected to one comm port, then taking one off line results in both instruments being off line as the comm port is dedicated to the pass through.

Although the instruments used in channels 5 and 6 are provided by the same manufacturer they are connected to independent comm ports. This is a typical connection for instruments that do not support multidropping.

It should be noted that the input on channel number 5 is from a dual range CO instrument. In most A/D loggers, a dual range instrument requires two input channels and some status indication of the range the instrument is on, because only one full scale and zero setting can be given to each channel. This is not true of the CPP operating over a serial port. The instruments are providing correct readings regardless of the range, and the CPP can accept and process these readings.

The NOX corrected on channel 7 is a computed channel and is derived by a mathematical calculation based on data from other channels. As such, it does not have a comm port assigned. This is discussed in more detail in section 4 of this manual and in Tech Note TN01, Common Calculations.

Channels 8, 9 and 10 are all provided by the same manufacturer, and are all connected to comm port #2. In addition to other applications, comm port #2 can be configured to accept a broadcast input. The CPP recognizes that this manufacturer's instrumentation operates in a broadcast mode and prepares itself accordingly. Each time a broadcast is received; the CPP retrieves and averages the data. These averages are then stored in the database with the polled data.

Data from the last channel (Flow) is retrieved from the A/D converter. In the CPP initialization, the user inputs the physical channel to which this analog signal is to be connected, along with full scale and zero information. The CPP retrieves the binary data from the A/D, converts it into engineering units, then averages and stores the data.

As can be seen from the above, the CPP comm ports can be configured in many fashions. It is all dependent on the instrument selections and the general topology of the system as a whole.

1.6 Typical Comm Port Assignment

The four comm ports of the CPP are generally used as follows;

Comm port #1	- Connect to central polling system
Comm port #2	- Various (See below)
Comm port #3	- Interface to instrumentation (3-00 through 3-08)
Comm port #4	- On site connection

Comm port #1 is used to transfer data stored in the CPP to a central polling device. Data can be transferred in an operator readable format or in a computer usable comma delimited, ASCII data transfer. Comm ports #1, #2 and #3 can be converted from an RS-232 to an RS-422/485 configuration by installing line drivers.

Comm port #2 is a catchall and is used for a variety of applications. In some applications it has been used to allow access from a central to a special piece of hardware located at the site, such as a GC or another data logger. It has also been used as a second port for another central polling device, when neither central can tolerate a busy signal. One of the primary uses of comm port #2 is to accept a broadcast input from broadcasting instrumentation. The CPP also offers a Modbus interface over comm port #2, and other special protocol conversions.

Comm port #3 is designated as the port that interfaces to and collects data from the instrumentation. The comm port #3 designations should not be made overly complicated or confusing. The base CPP printed circuit board contains four comm ports, one of which is designated 3-00. With expansion boards, 8 additional comm ports can be added to the system. These are designated 3-01 through 3-08. Therefore, there can be nine independent comm ports (3-00 thru 3-08) for interfacing to the instrumentation. These comm ports can be individually setup to poll instrumentation or to accept broadcast inputs.

Comm Port #4 is normally used as on site or local port. Comm port #4 is also the comm port that is used at H2NS for program development, consequently, anything that the CPP can do, can be exercised over comm port #4. Many of these features are hidden from the end user, but it does allow comm port #4 to be used for a variety of purposes. In a pass through mode, access to comm port #4 can be achieved over comm port #1.

The most basic of an air quality monitoring system uses only the CPP PC board with its four comm ports. Comm port #1 connects to a modem for sending data to a central polling facility. Comm port #2 is used to accept broadcast meteorological data from a single level met tower. Comm Port #3 is used to poll the instrumentation. Comm port #4 is the on site port.

1.7 Channel Setup

Channels in the CPP can be set up as

- Receiving RS-232 data from instruments
- Receiving data from an analog to digital converter
- A calculated or computed channel
- A rolling average channel
- A channel to be linearized
- A digital weighted averaging channel
- A meteorological input channel
- A counter input

This is all set up in a question/answer narrative in the CPP initialization. Once initialized, the CPP polls all connected instrumentation every fifteen seconds. After the serial polling is completed, the CPP then collects data from all A/D inputs starting from channel #1 and proceeding to the last enabled channel. The A/D samples each channel every second, so the minute sample is actually an average. Following the A/D data inputs, if there are any meteorological inputs, the data that has been processed elsewhere by the CPP is transferred into the common preliminary data buffer.

Following the meteorological data, any channels that are set up to be linearized are linearized starting from channel #1 and proceeding to the last channel enabled.

The CPP then computes all channels set up as calculated channels, again starting at channel #1 and proceeding to the last channel enabled. This allows computed channels to be concatenated for complex calculations. For example, a calculation could be started in channel #4 and the results of this calculation input by channel #5 to complete the calculation. Following the computed channel calculations, any rolling averages are calculated.

Unlike traditional data loggers that use fixed-point binary mathematics and data storage, the CPP is a full floating point device, storing data in a scientific format. Since there are no data conversions (from numbers to binary and then back to numbers) there are no conversion errors possible. Five bytes of memory are used to store each data point, two bytes for status, one byte for sign & exponent, and two bytes for four digits of significance.

The CPP stores 16 bits of status (two bytes) with each data point. Two status indicators can be printed with each data point printed out by the CPP. This is presented in more detail in section 3.6 of the manual.

The CPP uses the one minute data as the basic for all other averages. The CPP supports three separate averaging and reporting periods. The first, designated as a preliminary average, which is presently a one minute average. The second, designated as an interim average, can be set up as a five, six, or ten minute average. The third, designated as a final average, can be set up as a fifteen, thirty, or sixty minute average.

In ***internal*** memory, the standard CPP stores 3 hours of preliminary data, two periods of interim data, and four periods of final data. This is four days of hourly averages, 2 days of thirty minute averages, and 1 day of fifteen minute averages. Long term data storage is provided by either ***excess*** internal memory or a Compact Flash Memory card. Reference section1.23, or tech note TN17, Long Term Data Storage.

1.8 Operational Flow

The sequence in which data is processed in the CPP can be important in some applications. Each minute the CPP starts a poll of the instruments. All channels connected to serial ports are sampled, as are the A/D channels. At this point the incoming status is sampled and relevant status bits assigned to channels.

Channels set up as meteorological channels are continually (every second) receiving and processing data. At the end of each poll, the collected meteorological data is formed into an average and stored with the one minute samples. Separate data arrays are kept for each averaging interval, i.e., preliminary, interim, and final. With the exception of the meteorological data, subsequent averaging intervals for all other channels are calculated from the one minute values.

Following the inclusion of the meteorological data, computations are performed on any channels set up to be modified with a translation table (such as linearization). Following this procedure all calculated channels are computed. The calculations start with channel number one and proceed to the last channel. Any channel set up as a calculated channel is computed. In this fashion, calculations can be concatenated from one channel to the next. A calculation can be started in some channel and continued in a higher numbered channel. Following the computed channels the rolling averages are calculated.

1.9 Sequencers

The sequencers allow the user to either initiate operational conditions using the digital I/O features of the CPP and/or the CPP clock. The CPP contains ten independent sequencers. The sequencers can be started, stopped, enabled, or disabled from the system clock, from incoming digital bits or from digital output bit settings, or a combination of both using the Boolean functions available in the CPP. In addition, instrument calibrations can be started by the control output settings. The sequencers are described in more detail in section 2.10.

1.10 Calibrations

The CPP supports the calibration of instrumentation in several fashions. The detection and control of the calibrations can be tied into the sequencers. The control output settings can be used to drive solenoids controlling the sample input to instrumentation, in which case the CPP calibration would be set up to detect the settings and break out data as calibration data. Additionally, the setting of control bits can command the calibration section of the CPP to send serial commands to instrumentation to start and stop zeroes, spans, and calibrations. System calibrations are covered in more detail in section 2.11.

1.11 Boolean Functions

The CPP supports twenty operator settable Boolean routines. Each routine can be set up with eight IF conditions based on the digital I/O conditions, and THEN set or reset other digital I/O conditions. These can be used to stop/start sequencers, detect calibrations, detect/set alarm conditions, and the like. Since the Boolean routines are processed in an ascending order, they can be concatenated like the calculated channels.

1.12 Data Validation/Invalidation

Each data point collected by the CPP, either over the serial port or with the ADC, can be validated or invalidated based on either incoming digital bits, digital output bits or a combination of both. Data can also be invalidated following the transition of an incoming status bit, a control output bit, or a combination of both. In addition, data being retrieved over the serial port is also validated by status information returned by the manufacturer regarding the data point.

1.13 Connecting an Instrument to the System

When an instrument is connected into the system, the CPP must be apprised of this. This is done in the system initialization covered in section 2 of the manual. The manufacturer and model number must be selected from a table, the baud rate selected, the comm port selected, and an instrument ID entered if it is needed. Not all instruments need or accept ID's.

Once all this is entered, proper system operation can be readily verified. Type an EC to clear the error log. Then, wait a couple of minutes for the CPP to poll the instruments. Input a PM key command to look at the last minutes data point. Data from the new instrument should be present, if not the data point will be 8.888E+2oR, indicating that the data is missing. Type an E key command to get an error log listing to determine the problem. The error log is covered in more detail in section 3.0. For proper operation, the data should be present, without error flags, and the error log should not be indicating any errors for that channel.

Note: The error log will not flag a no response error from instrumentation that broadcasts. If a response is being received, then all other types of errors will be detected and logged. The minute averages (PM) should be printed out to ensure proper data collection

1.14 Averaging Intervals

The CPP supports three independent averaging intervals, a preliminary, an interim, and a final. The preliminary average is a one minute average, and the current hour and the three previous hours are stored in memory. With the exception of the wind parameters, the one minute averages are used to from subsequent averages. The interim average can be user selected as a five minute average, a six minute average, or a ten minute average. The final average is user selectable as a fifteen minute average, a thirty minute average, or an hourly average.

The CPP stores the current hour and the previous hour of interim averages in memory. The CPP stores the current time frame and the previous 4 time frames of final averages in memory. For a fifteen minute final average the time frame is six hours. For a thirty minute final average the time frame is twelve hours. For a sixty minute final average the time frame is twenty four hours. For longer term data storage reference section 1.23.

1.15 Baud Rates

The Baud rates for comm ports #1 and #4 are determined by dip switch settings. Refer to section 2.0 to determine these Baud rates. Once set up these very rarely change, so the dip switch is read only at power up. If the settings are changed, then the power must be cycled. The factory default for comm port #1 is 9600 Baud and 9600 Baud for comm port #4. The Baud rates for all other comm ports are determined by how they are set up during the system initialization phase.

1.16 Calculated Channels

The CPP supports channel to channel calculations. This is presented in detail in section 2.6.3 in the system initialization and in tech note 01.

1.17 Rolling Averages

The CPP supports rolling average calculations. This is presented in detail in section 2.6.4 in the system initialization.

1.18 Labeling Digital I/O

The CPP allows each digital input and each digital output to be given a 15 character label. This is very useful in documenting a system configuration. The sequencers can be given labels as well.

1.19 Time/Date

The CPP supports a date format of mm/dd/yy or dd/mm/yy. At power up and at periodic rates the CPP downloads the time and date to instrumentation that supports keeping time and date. This is done at 05:27 each morning, so other tasks that may depend on having a consistent time during the entire duration of the task, should be set up to occur at another time.

1.20 Instrument Pass Through

The CPP provides two modes of communicating with attached instrumentation. One is an On Line mode where user commands are interlaced with CPP polling commands. The second mode connects comm port #1 directly to an instrument in an unabated pass through mode. The instrument is flagged as being off line. This allows the user to communicate with the instrument and run manufacturer supplied diagnostic routines without having to adhere to a CPP protocol. Section 5 provides more detail on this feature.

1.21 Long Term Data Storage

The memory elements in the CPP are divided into five separate functions, which are;

- ð Configuration Memory

ð Internal Memory
ð Excess Internal Memory or excess memory
ð Removable Memory (CFM)
ð On board EEPROM (Not used for data storage)

1.21.1 Configuration Memory

The memory designated as the configuration memory is used to store the operating configuration set up in the CPP. This configuration memory is further subdivided into partitions for each function available in the CPP, *e.g.* channel set ups, data validity, sequencers, and so on. A checksum character is calculated with each partition. A battery backup maintains the contents of memory. At any power up, each partition is checked for integrity. If and error is detected, the CPP sets a flag and clears the affected partition.

Should an error be detected, and if a CFM card that contains a set up configuration is installed in the CPP, the CPP loads the configuration from the CFM card and starts operating. If a CFM is not installed, or it does not contain a configuration, the CPP checks the on board EEPROM. If it has a configuration, the CPP downloads this configuration. If neither has a configuration, the CPP loads defaults into partitions with a checksum error.

1.21.2 Internal Memory

Internal memory is a section of memory where some number of each average being formed is stored. The number varies depending on the emulation firmware resident in the code. Each CPP stores in active memory a minimum of 24 of the latest of each average being formed. Longer term data storage is provided by the excess internal memory and the CFM.

1.21.3 Excess Internal Memory

Each CPP contains 512K of internal RAM. Some portion of this is allocated for the configuration set up and storage of the most recently formed averages, *i.e.,* the internal RAM. The remainder of this internal RAM (≈200K) is designated as excess memory and used to stored information for longer terms. The information stored in the excess memory is selected by the user, and may consist of data, calibrations, alarms, events and so forth.

Information stored in the excess internal memory is stored in a format identical to that used in the removable CFM. If a CFM is installed, all read/write commands are directed to the CFM. If a CFM is not installed, all read/write commands are directed to the excess internal memory. Data is always being recorded to the excess internal memory, whether or not a CFM is installed.

1.21.4 Compact Flash Memory (CFM)

The removable CFM is used for long term data storage. A set up configuration can be stored on the front of each CFM. The configuration set up in the CPP can be uploaded into the CFM and likewise, a configuration stored in the CFM can be downloaded into the CPP.

The CFM plugs into a slot on the CPP rear panel, and can be formatted as 8 through 64 megabytes of memory, in blocks of 8 megabytes, 8, 16, 24, 32, 40, 48, 56 or 64. Eight megabytes of memory will store about 5 days of minute averages for 20 channels, or 45 days of fifteen minute averages, or approximately ¾ years of hourly averages. Approximately 1½ years of data can be stored for 10 channels in an 8 megabyte formatted CFM.

1.21.5 Long Term Data Storage

Presented below in Table 1.3 are storage times versus number of channels being stored.

# of Channels	Average	Internal Memory	8Mb CFM
10	1 min	40 Hrs	10 Days
10	5 min	200 Hrs	50 Days
10	15 min	25 Days	150 Days
10	60 min	100 Days	600 Days
20	1 min	20 Hrs	5 Days
20	5 min	100 Hrs	25 Days
20	15 min	12 Days	75 Days
20	60 min	50 Days	300 Days

Figure 1-3 Long Term Data Storage

These times will vary depending on what other information is being recorded with the data.

1.22 Configuration CFM

As mentioned above, a single configuration can be installed at the front end of a data formatted CFM. The CFM can also be formatted as a Configuration CFM. Sixteen different configurations can be stored on a configuration formatted CFM. Data will not be recorded on a configuration formatted CFM.

1.23 Power Input & I/O Protection

The input power line along with all I/O of the CPP has internal protection against normally occurring static discharges and power surges. The CPP can operate from 120/240 volt 50/60 Hz line power, and automatically adjusts to the incoming power. In addition, the CPP can also be powered from an external DC voltage source. The DC voltage can be either 5 volts regulated, @ 3 amps, or 7 to 12 volts, capacitor filtered. H2NS should be notified of this intended operation, as the +5 regulating components are not installed unless requested.

1.24 Manufacturers Supported by the CPP

Provided below is a list of manufacturers that are supported by the CPP at the time of this publication. This list is steadily growing. To get the latest list request Tech Note TN16. TN16 also provides the reference to application notes that are available for each manufacturer. The application notes provide detailed information regarding the CPP interface to manufacturer instruments.

1.25 Operator Messages

The CPP supports the storage of 8 operator or computer messages. Each message is limited to 80 characters.

To record a message, enter OPW x, where x is the desired message number 1-8. If a message is already stored in the message bin number, the CPP prints the stored message and awaits an operator response. Entering a carriage return or escape exits without changing the existing message. Entering CLR<Enter> results in this message bin being cleared. Entering anything else erases the existing message and starts recording the new entries. A message is terminated with a carriage return or escape. The time and date is automatically stored with each message.

Messages can be printed by entering OPR x, where x is 1-8 or an A. An A instructs the CPP to print all stored messages. Messages can also be retrieved over comm port #1 or 4.

1.26 List of Keyed Commands

As these can change, reference Tech Note TN25 for the current available key commands. A list of key commands is available from the CPP by entering HELP. A list of key commands not normally used by an operator is available by entering HELP.

User's List Of Key Commands

Key Command	Definition
A	Print last 5 power fails
AC	Clear power fail log
ALM	Print channels currently in alarm
BR(X)xx	Reset bit # xx (X) = O = Outout, = I = Input, (X) not sent = Output
BS(X)xx	Set bit # xx (X) = 0 = Output, = I = Input, (X) not sent = Output
CFGUL	Upload configuration to CFM
CFGDL	Download configuration from CFM
CFGLOG	Print a log of configurations in the CFM
CFM?	List operator CFM commands
CsU	Allows clearing of setup parameters
D43	Lets comm port #4 communicate with instruments
E	Print error log
ERASEp	Erase preliminary arrays
ERASEi	Erase interim arrays
ERASEf	Erase final arrays
ERASEcE	Erase RamPack
EC	Clear error log
FORMATcx	Format CFM, x = megabytes
HELP	List key commands
I	Brings up Main Menu
L	Prints configuration set up
LC	Prints Calibrator Set Up Parameters
LCD	Toggles demo data for the bar & trend displays
Lx	Prt last averages for 20 channels, x = P = prelim, =I = interim, =F = final
LxA	Prt last average for 40 channels
MM	Brings up Main Menu on LCD
Mb aaaa cc	Prints RAM contents, b=bank, aaaa= address, cc=# 16 byte lines
PAC	Automatically print cal results
PAD	Print I/O settings on change
PARxx	Print response from instrument on channel #xx
PC	Print latest cal results
PD	Print instantaneous I/O settings
PFx	Print final averages for x periods
PIx	Print interim averages for x periods
PMxxx	Print preliminary averages for x periods
PV	Print instantaneous data values
PVA	Print instantaneous A/D values
PVR	Toggle print instantaneous data values every 10 seconds
PVAR	Toggle print instantaneous A/D Values every 10 seconds
PVS	Select scientific or engineering print formats
SET1-8	Comm #1 input - connects CP#1 to Pass Thru comm ports
SETF	Comm #1 input - Switches comm #1 signal lines to Comm #4
SEQ	Start/stop sequences
STATUS	List status flags
T	Print present time & date
TIxx	Send time & date to instrument on channel #xx
TF	Set mm/dd/yy or dd/mm/yy date format
TS	Set CPP time & date
ZS	Enter zero & span values

Figure 1-4 User's List Of Key Commands

2 Initialization – System Configuration

2.1 Getting Started

It is recommended that a new user read the Introduction, section 1, prior to using the CPP. The configuration of the CPP-4794 follows a “question - answer” script. To access this “script” you have to access the set up menu of the CPP. Therefore you must communicate with the CPP-4794.

To communicate with the CPP-4794 requires a terminal device. This device could be a computer running HyperTerminal, or Procomm, or a similar terminal program. HyperTerminal is a communication program by Microsoft and it is a standard component of Windows 95, 98 or NT. To access the set-up configuration features of the CPP, you should communicate through Comm Port 4. Comm port 4 is located on the front and rear panels of the CPP. Once we are started we will step through each of the selections on the setup menu.

The CPP will respond to certain keyboard characters, called key commands. A list of key commands is presented on the next page. As these can change, reference Tech Note TN25 for the latest available key commands. Do not confuse key commands with menu selection items. In initialization all keyed characters must be in upper case. The default settings for comm port #4 are 8 data bits, 9600 Baud, 1 stop bit, and no parity. HyperTerminal should be set for no hardware control.

The CPP is initialized with an **I** key command. Keying in an **I** results in the CPP presenting a numbered Main Menu. If a password is required then the CPP requests the password, and will proceed no farther until the correct password is input. Although the content of the Main Menu may change depending on options installed in the firmware, an example of the Main Menu is presented in Table 2.1 below.

```
                                Main Menu

    01 - Password                      02 - Station Name I/D
    03 - Averaging Intervals           04 - # Chans Enabled
    05 - Channel Set Up                06 - Data Validity
    07 - Sample Delay                  08 - Init Sequencers
    09 - I/O Labels                    10 - Cal Set Up
    11 - Boolean Set Up                12 - Alarm Set Up
    13 - Auto Prints                   14 - Init Met Parameters
    15 - Data Print Format             16 - Cartridge Recordings
    17 - External I/O                  18 - Curve Fit
    19 - Set Up Calibrators      20 - Hold Last Value
    21 - DO to DAC               22 - Modbus Set Up
    23 – Poller Inactivity
      Choice -
```

Figure 2-1 Main Initialization Menu

Key commandg in one or two digits followed by a carriage return makes a selection. A carriage return at this point exits the system initialization routine. The CPP remains fully operational in the initialization phase. However, as parameters are being changed in a real time fashion it may result in some data anomalies. The minute averages are flagged with a Z status modifier and are not used in subsequent calculations. Upon exiting the initialization routine, the CPP recomputes all system operational parameters. This eliminates the possibility of changes made in one area not getting properly conveyed to other related areas.

2.2 Password

If a password has been set in the CPP, when an **I** is entered, the CPP asks for the password. The CPP will not allow the user to proceed farther until the correct password is entered. If a password has not been set up the main menu is presented.

The CPP is shipped from the factory without a password set up. To enter a password, select Password from the main menu. Entering this routine allows the user to set up a password. Entering a carriage return at this point will exit the routine without changing the password setting. If there was no password set up, then it will exit with no password set up. If a password was set up, then the same password is still in affect upon exiting.

The CPP has two passwords, one is designated users password, and the other is designated as a managers password. The managers password allows the setting of the users password. If a managers password is required to change the user password, then the CPP prompts for the managers password input. If it is entered wrong, the CPP exits back to the Main Menu. If it is entered properly, then the CPP acknowledges the entry and awaits a new user password. The user and the managers passwords can be up to six characters long. Shorter passwords can be used and are terminated with a carriage return, thus a carriage return can not be a part of the password. The passwords are case sensitive. If more than six characters are entered the CPP reasks the question.

After the new user password is entered, the CPP asks if the managers password is to be changed also. A carriage return exits the routine with the new users password in effect and managers password the same. Any entry other than a carriage return is interpreted as the start of a new password to change the managers password.

This approach allows someone to be given the users password to operate the system, but they can not inadvertently (or intentionally) change the users password and either not know it or forget it.

To disable the password feature enter six spaces followed by a carriage return when the system asks for a new password. This is true of both passwords.

2.3 Station Name/ID

The CPP will accept a 16 character station name. Upon entry, if one has been entered, the CPP will print the existing station name. A carriage return at this point leaves the name unchanged and goes to the next question. If the name is to be changed input the station name terminated with a carriage return. If more than 16 characters are entered the CPP uses the first 16 as the station name.

The CPP then asks for a station ID number. This station ID is used to bring comm port #1 on line with the $ID.

The CPP then asks for a second station ID. This ID is used by a central processing unit to communicate with the CPP and retrieve data in a comma delimited transfer. The CPP will accept up to three characters (0-255) as the station ID. The factory default is 010 and should not be changed unless necessary.

It is understood that the dual station ID's can be confusing. For clarification please contact customer service at H2NS.

2.4 Averaging Intervals

The CPP supports three independent averaging intervals, a preliminary average, an interim average, and a final average. Entering this set up routine results in the CPP asking for the preliminary averaging interval. The preliminary average is fixed at a one minute average. The question is included should it be desirable to change this at a later date. Any enter goes to the next question, which asks for the interim averaging interval. The acceptable inputs are **5** (5 minute average), **6** (6 minute average), and **10** (ten minute average). A carriage return goes to the next question leaving the interim average unchanged.

The CPP then asks for the final average. A carriage return exit this set up routine leaving the final average unchanged. The acceptable inputs are **15** (15 minute average), **30** (30 minute average), and **60** (60 minute average). An acceptable entry gets stored as the final average and then this set up routine is exited.

With the exception of the wind averages, the interim and final averages are formed from the one minute averages. The wind averages are formed and stored independently. Only one minute averages with good status indications are used in forming the subsequent averages. The subsequent averages are checked to see if sufficient samples were present and appropriately flags the data point.

If any good samples are present, these are averaged as the data point, even if the number of good samples are insufficient to be considered a valid data point. If no good samples are present, then all samples present are averaged and presented as the data point, and flagged bad.

2.5 Number Of Channels Enabled

The CPP is delivered as a forty channel device. In many applications not all channels are used. This set up allows the user to indicate how many channels the CPP should collect data for and print out. The acceptable inputs are multiples of five.

2.6 Channel Set Up

Selecting the channel initialization results in the CPP asking a series of questions as presented below. Operator responses are in bold. The numbers 1.), 2.) and so forth are used for discussion purposes only and are not printed in an actual system initialization.

2.6.1 Example - Serial Input

1.) Channel # = **1<Enter>**
2.) Name = NAME **SO2<Enter>**
3.) Units = UNITS **PPB<Enter>**
4.) Full Scale = 100.0 **500.0<Enter>**
5.) Zero = 0.0 **0.0<Enter>**

6.) Instrument Manufacturer
00 = Other
01 = API
02 = CLI
03 = Thermo
7.) Setting = 01 **03<Enter>**

8.) Thermo Model #
01 = SO2 - 43C
02 = NOX - 42C
03 = NO2 - 42C
04 = NO - 42C
05 = CO - 48C
06 = O3 - 49C
9.) Setting = 01 **<Enter>**

10.) Comm Port (3-00 Thru 3-08) = 3-01 3- **02**
11.) Half Dplx(H), Full Dplx(F) = F **<Enter>**
12.) Baud Rate
1 = 300, 2 = 1200, 3 = 2400, 4 = 4800, 5 = 9600, 6 = 19.2K, 7 = 38.4K = 5 **<Enter>**

13.) Instrument ID = 00 **43<Enter>**
14.) IMD Installed = 00 **<Enter>**
15.) # Points For Interim Avg = 01 **04<Enter>**
16.) # Points For Final Avg = 01 **9<Enter>**
17.) Unip(U)/Bip(B) = U **<Enter>**
18.) Unip(U)/Bip(B) Cals = B **<Enter>**

19.) Channel # = **<Enter>**

Line one simply asks which channel is to be initialized. A carriage return here exits back to the Main Menu. In other steps, a carriage return input leaves the setting unchanged and the CPP goes to the next question. In cases where the operator must provide an input, the CPP asks the same question again.

Lines 2 through 5 request information that is not required for proper operation when interfaced to an instrument over the serial port, because they provide the CPP with actual numbers. It is, however, good operating procedures to set these inputs to match those in the instrument. The correct full scale and zero values are required if data stored in the CPP is being output to some other device (for example, Odessa Engineering's Envicom, or a Modbus) in a binary format. This is also true of channels set up as ADC input channels.

Channel names and the units can be up to six characters in length. The full scale and zero values can be four digits in length and the number of digits after the decimal point (if a decimal point is used) must be the same in the full scale and zero inputs. The inputs can also have a minus sign.

Line 6 lists all of the manufacturer drivers that are resident in this software version. In the example, for clarity, only three manufacturers are presented. As the manufacturers are alphabetized, the numbers delineating the various manufacturers will change depending on how many drivers are installed.

In Line 7 of our example, the operator selected Thermo as the manufacturer. The CPP then prints a list of the model numbers that are supported for the Thermo Environmental instrument line.

In Line 8 & 9, the operator input a carriage return, which selects the SO2 analyzer.

In Line 10 the CPP asks to which comm port this analyzer is going to be connected. This is the connector on the CPP into which the cable from the instrument is plugged. The allowable inputs are 00-08, which are nine comm ports. As discussed above, all Thermo instrumentation can be plugged into the same comm port. In our example, the operator selected comm port 3-02. The comm port nomenclature should not be made overly complicated or confusing. Although comm Ports 3-00 through 3-08 are independent serial ports, internally the comm port 3 designation references interfacing to the instrumentation. Comm port 3-00 addresses comm port #3 physically located on the CPP printed circuit board and labeled comm #3-00 on the CPP back panel. Comm ports 3-01 through 3-08 address comm ports physically located on the expansion printed circuit boards and are labeled comm 3-01 through comm 3-08 on the CPP back panel.

It should be noted that comm port #2 can be used to interface to some instrumentation. The Thermo line of instrumentation is **not** included in that group, and should always be assigned to a comm port number 3 designation. Comm port #2 is usually used with instruments that broadcast, which is normally meteorological instrumentation.

Line 11 allows the comm port to be set up for half duplex operation or full duplex operation. This is normally set to full duplex, but some instruments require a half duplex configuration.

In line 12, the CPP is asking what Baud rate should be used to communicate with this instrument. This setting must be the same as that set up in the instrument.

Line 13 asks if an instrument ID is required. Instrument ID's are always required with the Thermo instruments. Input the appropriate ID. In the example, the user input the SO2 default ID of **43**. This could be any other code as well, *e.g.,* **63**. An instrument ID of 00 indicates that no instrument ID is required, which is not allowed with Thermo instrumentation. If no instrument ID is entered for a Thermo instrument, the CPP applies the factory default ID.

Line 14 is asking if an IMD (Intelligent Multidrop Device) is installed in this comm port cabling. The IMD, manufactured by H2NS, allows a number of different instruments to be connected to a single comm cable. The instruments can have different protocols and even different baud rates. This is very useful when a collection of instrumentation is located remotely from the CPP. The IMD allows only one cable to be run between the CPP and the instruments, with the IMD providing the instrument multiplexing locally at the instruments. An IMD is required for each instrument connected to the line. The IMD supports addresses from 01 through 60. An address of 00 indicates that an IMD is not installed in this comm cabling.

Lines 15 & 16 are questions regarding data validity qualifiers. The first (line 15) asks how many valid one minute samples the CPP must have to qualify the interim average as valid. Question 16 asks the same in regards to the final average. Reference section 3.6 of the CPP manual for a more detailed discussion of status.

Line 17 is asking if the CPP is set negative data values to zero or not. This is quite often done as negative numbers have no meaning in some applications, and can cause calculations to go out of bounds.

Line 18 is asking if the CPP is to set negative numbers received during a calibration to zero or not. This is normally not done, so that a negative instrument drift can be detected.

2.6.2 A/D Input

Lines 1 through 6 are the same with the A/D input selection as with the serial input selection. At line 7 the other input (**0**) is selected. The CPP responds with the following selection menu. It should be remembered that setting up a channel as an A/D channel requires that the full scale and zero values entered above are now important. The CPP uses these values to convert the binary values received from the ADC to actual numbers.

A = A/D
C = Calculated
D = Digitals
R = Rolling Average
F = Counter

Input an **A** to set this channel up as an A/D input channel. The CPP then prints out an existing channel number if one has been previously assigned and asks the following;

A/D Channel = 04 **1<Enter>**
Input Range

(1=100mv, 2=1v, 3=5v, 4=10v, 5=4-20mA) = 3 **2 <Enter>**

In the example this channel was previously set up for an analog input on A/D channel number 4. The operator changed this to A/D channel number 1. The next question is asking the voltage full scale of the signal being connected to this A/D input. The operator changes this from a 5 volt full scale signal to a 1volt full scale signal. The most common ranges are offered as standard.

At this point the CPP picks up the questioning at line number 14 regarding the samples required for data validity. All A/D inputs are sampled once a second and averaged to form the one minute sample.

The CPP will also accept A/D channel # 88 as an input. This signals the CPP that this data is being supplied elsewhere in the unit, usually from the meteorological section.

2.6.3 Calculated Channels

Although not used in Air Quality Monitoring applications very often, the calculated or computed channel functionally of the CPP provides a very powerful capability. It is required in stack emissions applications where channel to channel operations are required to provide pounds per hour and pounds per MBTu values. Tech note TN01 provides more detail on setting up common calculations.

In Air Quality Monitoring applications the calculated channel feature can be used to calculate NO2 from NOx and NO values, correct wind direction values for magnetic declination, or calculate dew point from temperature and relative humidity inputs. Since the CPP conducts mathematics with numbers, the calculations are accurate enough to even provide delta temperature values if the sensors used are sufficiently accurate.

The template used for calculated channels consists of four terms and three operators and is presented below.

```
                        Math Template
(CHA)(KA)  {OP1}  (CHB)(KB)  {OP2}  (CHC)(KC)  {OP3}  (CHD)(KD)
     |        |          |        |          |        |        |
  Term 1    Op #1     Term 2    Op #2     Term 3    Op #3   Term 4
```

The math template evaluates parenthetical expressions based on algebraic rules. Terms in the template are evaluated in the following order:

1. Term 1 and Term 2 are evaluated using Op #1, (TermA)
2. Term 3 and Term 4 are evaluated using Op #3, (TermC)
3. The results of the above operations are evaluated using Op #2.

Variable Definitions

CHA/B/C/D One of the CPP's channels. This can be from a serial input, an A/D input, or from another computed channel. A channel number of 0 is valid and is considered a null channel, which results in a term whose value is given by the entered constant (K input). The default is 0.

OP1/2/3 Mathematical operators, listed below. When the CPP encounters a colon as an operator, no more terms in the expression are evaluated. For example, if OP #2 were a colon, then this calculated channel operation would only evaluate terms 1 & 2 using operator number 1, OP #1.

+ Addition
- Subtraction
* Multiplication
/ Division
: Stop
V Term validation selection (OP #2 only)

KA/B/C/D An entered constant with a fixed format of X.XXXE±X. The default is 1.000E+0. The exponent input can be plus or minus and can be a number from 0 to 9. For example an input of 2.345E+2, is a number of 234.5.

V Operator

Inputting a V for operator #2 functions as follows. If the status associated with TermA is good then the results of TermA are stored as the result of the computed channel. If the status associated with TermA is not good, then the results of TermC are stored as the result of the channel computation. In this case, the status of TermC is stored regardless of its validity.

Initialization

Upon entering this routine, the CPP prints the math template and then asks for a channel number. A carriage return at this point exits back to the Main Menu. The CPP then asks for a function code. Any channel can be set up as a calculated channel, provided the channel number is within the range set up in the number of channels enabled. The CPP then starts asking for inputs to fill the math template. This is very straightforward and an example is presented below. If the square root function is enabled, the square root of the calculated results is computed and stored as the result of the channel computation.

```
Function None(0), Sqr Root(1), = 0<Enter>
Chan # = 09<Enter>
CHA = 00 1<Enter>
KA = 1.000E+0 = 1.000E+0<Enter>
OP #1 = :  *<Enter>
CHB = 00 <Enter>
KB = 1.000E+0 = 3.704E-2<Enter>
OP #2 = : /<Enter>
CHC = 00 <Enter>
KC = 1.000E+0 = 2.033E+1
OP #3 = :  -<Enter>
```

CHD = 00 **4<Enter>**
KD = 1.000E+0**<Enter>**
Enable Y/N = N **Y**
Chan # = **<Enter>**
Back to Main Menu

In this example (which is actually a Lbs/MBtu calculation), the operator set up the following;

[(CH01)(1.000) * (1.000)(.03704)] / (20.33 - CH04)

It should be noted that in the example channel 9 was being set up as a computed channel, yet the user could get data from channel 12 if needed. All channels getting inputs from serial ports or A/D channels have been updated prior to starting the calculated channel computations. Calculated channels can be concatenated to implement more complex expressions. For example, channel 5 could pick up the results of channel 4 for additional calculations.

The CPP can save and report data values with an exponent of plus or minus 9. In the interim calculations the CPP allows the exponent to range from plus or minus 127. In the final answer, if the exponent is not in the range of plus or minus 9, then the data point is appended with a math error status flag, M.

Examples of common calculations are presented in Tech Note TN01. These examples include pounds per MBTu, pounds per hour, and diluent correction.

2.6.4 Rolling Averages

A rolling average is formed by summing all the valid averages within a given number of periods, then dividing that sum by the number of entries summed. The rolling average uses the most recent valid average and goes backward in time the specified number of periods. A rolling average channel can only accept data from a preceding channel. For example, if channel 10 is being set up as a rolling average channel, the input data can only come from channels 1 – 9.

Two Types of Rolling Averages

The CPP supports two types of rolling averages; continuous (rolling) and discrete (building). With a continuous average, the CPP updates the rolling average constantly as it receives new information. With a building average, The CPP resets the rolling average to zero after it forms the average for the final period.

The CPP supports the following rolling averages; 1, 3, 4, 6, 8, 12 and 24 hours. The CPP makes some assumptions when calculating the building averages. If the interval input is 24, the CPP assumes that this is a daily average, and discrete rolling averages are reset at midnight. An input of twelve resets at midnight and at noon. An input of three resets every 3 hours at 3, 6, 9, 12, 15, 18, 21, and 24 hours. Continuous rolling averages do not reset. The oldest reading is removed and the new reading added in.

Entering this routine results in the CPP asking:

```
Rolling Averages
Rolling ( R), Building (B), Clear( C) = R <Enter>
Input Chan # = 00  04<Enter>
P, I, F =  F  <Enter>
Interval (1,3,4,6,8,12,24,S, Hrs) = 0 24<Enter>
% Points In Final Avgs  = 45 75<Enter>
Chan # =
```

In the example, the user has set up this channel to compute a continuous rolling average from the final averages of channel #4. The rolling average must have 75% of the hourly averages used in the calculation before the rolling average is considered as valid. The value specifies the percentage of samples that must be present to make a valid rolling average.

Sometimes this rolling data validation criteria can lead to misconceptions. In the above example, a 75% validation criteria means that 18 of the hourly averages used as inputs to the rolling calculation must be present in forming the rolling average before the rolling average can be considered valid. Data points flagged as bad by the CPP are not used in the rolling average calculation. Data points flagged with other status flags indicators are evaluated by the rolling average routine and the following are not used; if in calibration, if insufficient samples were present to form the average, or a sample delay flag. Other bad status conditions are flagged as bad by the CPP. The calibration aspect has more bearing when the rolling input is from a preliminary data point or an interim data point, as status conditions can span several input samples.

In forming a continuous rolling average, several criteria are at work. Initially, the routine is adding in each new valid data point into a running sum and counting the input. Each invalid data input is stored but not summed into the running sum, nor is the count incremented. This continues until the number of data points required to form the rolling average are input (valid or invalid). This is dependent on the type input and the roll duration. After this time, with each new data input (valid or invalid) the oldest data point is removed. If it is a valid data point, then it is subtracted from the running sum and the counter decremented. If it is an invalid data point, then it is removed but not subtracted form the running sum, as it was never added in, and the counter is not decremented.

The status flagged stored with the rolling averages reflects the condition of the number of valid samples in the running sum. If no counts are present, the data point has a B modified. If there are valid samples present, but not enough to meet the validation criteria, the data point has a < modifier. If sufficient samples are present in the running sum to form a valid average, then the data point is flagged as good.

In discrete or building averages, the status uses the same criteria. If no samples are present in the building average, then the data point has a B status. If some samples are present, but not enough to meet the validation criteria, then the data point has a < status. If enough samples are present to meet the validation criteria, then the data point has a G status.

With the interim or final averages as the inputs, any of the above rolling averages can be formed. With the preliminary averages as the input, only the 1 hour, 3 hour and 4 hour rolling averages (continuous) can be set up. Any of the building averages, with any average as the input, can be set up.

The listing print (L key command) presents the channels set up to calculate rolling averages as presented below. The heading that provides the instrument manufacturer is used to indicate that this channel has been set up to from a rolling average. The 04 FR indicates that the input is the final averages (F) from channel #04, and that this is a rolling average (R). The roll time is 24 hours.

L Key Heading	Channel
Inst Man	RL AVG
MDL# or ADC#	04 FR
C-Port/Range	24 Hrs

S Option

The S feature allows the user to specify the number of values to use in the rolling average. This allows the user to set up site specific rolling averages. If the S feature is being used to create a rolling average, the the correct input is Sxxx, where xxx is either one, two or three digits followed by a carriage return. As an example, using the preliminary averages to form 6 or 20 minute rolling averages to meet some opacity measurement requirements. If it is set up as a special the last line above in the L key command example will be Sxxx.

An entry of S20 forms a rolling average that uses the last 20 inputs (preliminary averages for example) to form this rolling average. Every minute the minute average is updated with the latest 20 minute rolling average calculation.

If the S feature is used to form a building average, then the proper input at the S question is SI. This results in the CPP forming a rolling average from the preliminary values that is synchronized to the interim time boundaries. At each Interim time boundary, the rolling average is reset and builds over the Interim time period. Each minute the minute average for this channel is the rolling average of the number of preliminary averages accumulated in the time frame. If this is set up the last line above in the L key command example will be SI.

If a building average is selected, the S feature is selected, and then a number entered (not an I), the CPP automatically converts the average to a rolling average.

2.7 Data Validation

Data processed by the CPP can be validated/invalidated based on the incoming digital inputs and the digital outputs settings.

2.7.1 General

One of the methods the CPP uses to validate the collected data is to tie the data points into the incoming digital inputs and digital output bit settings. Individual digital input conditions and individual digital output settings can be defined for each channel and used to either validate or invalidate the data point. With the Boolean functions, complex data validation/invalidation schemes can be configured and the used to set or reset a single digital I/O bit to control data validity on a single channel or multiple channels. Typical examples of using input bits for data validation would be fault signals from analyzers or sample system faults such as low pressures or faulty sample conditioning equipment.

It should be pointed out that if the CPP is collecting data from an instrument over the serial port, then certain status information returned by the instrument is checked each time data is retrieved from the instrument. This varies from manufacturer to manufacturer and is covered in appendix B associated with interfacing to a given manufacturer. The use of digital I/O bits adds another degree of flexibility to the data integrity.

An example of using digital output bits for data validation would be in stream switching applications where the control outputs of the CPP are programmed to control the stream switching.

The CPP supports forty digital inputs and forty digital output bit settings regardless of how many points are actually installed in the hardware. Status inputs not implemented in the hardware are detected by the CPP and set to the open or 'zero' state when status inputs are sampled. These bits can be set to other conditions by the Boolean functions.

2.7.2 Digital I/O Set Up

The set up for the digital inputs or digital outputs are identical, therefore an example set up using only the input bits is presented. The CPP interprets a grounded input as a 'one' state, and an ungrounded input as a 'zero' state.

In the initialization function (I Key command), select the Data Validation input. The CPP then asks for a channel number. Enter the desired channel number, and the CPP then asks for status inputs that can determine the validity/invalidity of the data point.

```
Data Validity
Channel # = 01<Enter>
(1-8)    = XXXXXXXX  0XXXXXX1<Enter>
(9-16)   = XXXXXXXX  <Enter>
(17-24) = XXXXXXXX  <Enter>

(25-32) = XXXXXXXX  <Enter>
(33-40) = XXXXXXXX  <Enter>
Channel # = <Enter>
```

The CPP prints the current status input criteria for input bits 1 through 8, and then waits for an entry. An **X** entry is a "don't care," meaning that no matter what the state of that digital input, the data is not affected and is considered valid.

A digital **0** (open) entry means the data is valid as long as that digital input is not grounded. If the assigned status input is grounded, the data is not valid. If a **1** (closure) is entered, the data is valid as long as that input is grounded. If the assigned input is not grounded, the data is not valid.

The data is considered valid only if all the digital bit conditions match. In the above example, if bit number one is detected as a **1 OR** if bit number eight is detected as a **0**, then the data point is flagged as bad. If an **AND** condition is needed the digital validity should be set up in the Boolean functions and one bit set accordingly which will be detected in this data validity routine. Any entry other than **X**, **1**, or **0** translates to the CPP as an **X**. All eight entries must be made before they will be accepted by the CPP. A carriage return advances to the next eight and so on.

After all digital inputs are set up, the CPP the asks the same questions for the digital output bit settings.

Data during calibration events does not need to be expressly invalidated. The CPP does this automatically.

2.8 Delay Sample

Delay sampling implies invalidation of sampled data for a given period of time following the transition of either a digital input bit or a digital output bit.

2.8.1 General

The check for a transition occurs once per second. Typical examples of the use of the delay function are to incorporate a stabilization period after calibrations or after a stream switch. The CPP first asks for delays on input conditions and then on output conditions. A given channel can have more than one delay number assigned to it. This approach allows different delays to be assigned to different occurrences. As an example, a delay of 5 minutes could be assigned if a particular input bit changed state, and a delay of 10 minutes assigned if a different input bit or a particular output bit changed states.

2.8.2 Set Up

Sample Delay
Channel # = **2<Enter>**
Enable Y/N = N **Y<Enter>**

To disable the sample delay option, enter a **N**. To enable the sample delay option on the transition of a digital input, enter a **Y**. The CPP then asks for the input bits to check for a transition. The CPP then asks for relevant output settings.

Delay On Digital Inputs
(1-8) = **XXXXXXX1 <Enter>**
(9-16) = XXXXXXXX **<Enter>**
(17-24) = XXXXXXXX **<Enter>**
(25-32) = XXXXXXXX **<Enter>**
(33-40) = XXXXXXXX **<Enter>**

The CPP then prints the current definition of which bits are to trigger the sample delay for bits 1 through 8. An **X** entry is a "don't care," meaning that no matter what transition occurs, no delay is activated. If a **1** is entered, the delay starts when the assigned bit transitions from 0 to 1. If a **0** is entered, the delay will start when the assigned bit transitions from 1 to 0. If more than one bit is defined to start the delay, the one which occurs first must time out before the delay can be started by the transition of another bit. If more than one input is needed, it is best to use the Boolean function to set only one bit. All eight entries must be made before they are accepted by the CPP. A carriage return input goes to the next set of eight inputs. The delay is triggered on a transition, not a level. The CPP then asks for the relevant digital output settings.

Delay On Digital Outputs
(1-8) = **0XXXXXXX <Enter>**
(9-16) = XXXXXXXX <**Enter**>
(17-24) = XXXXXXXX <**Enter**>
(25-32) = XXXXXXXX <**Enter**>
(33-40) = XXXXXXXX <**Enter**>

The CPP then asks for the time to delay. The delay is in seconds.

Delay (1-9999 Sec) = **300**<**Enter**>

Enter the duration of the delay in seconds, from 1 to 9999. Once the delay is triggered, the delay timer decrements once each second until the timer goes to zero. Only after this time are samples again considered valid. In our example, the user assigned delay#1 to channel #2. Any time input bit number 8 goes to a one, or output bit number 1 goes to a zero, a 300second, or 5 minute delay or purge is conducted on channel 32. Collected data during the delay interval is flagged with a P modifier.

2.9 Labeling Digital I/O

The CPP allows each digital input and each digital output bit to be given a 15 character label or name. This is very useful in helping document a system configuration. The CPP asks for a bit number, prints a label if one has been set up, and awaits an operator entry. Entries include the number and type. As an example **I01** is input bit number 1, and **O10** is output bit number 10. Shorter labels can be given by terminating the input with a carriage return. If more than 15 characters are input, the CPP takes the first 15 characters as the label.

2.10 Control Sequencers

Control firmware for the CPP-4794 provides for timing and controlling the system's discrete (digital) outputs.

2.10.1 General

These outputs may relate to actual outputs (contact closures) or they may relate only to internal memory locations, also referred to as virtual output bits. These control functions can be started based on time (Automatic start), or they can be started based on the state of other digital output settings and/or digital inputs (Manual start). In a like manner, these control functions can be inhibited (Disabled) based on the output settings and/or the digital inputs. The CPP supports ten independent sequences each with nine output events. Each event has an operator settable signature comprised of 40 (optionally more) discrete output bits. The convention used by the CPP when printing output and input bit assignments is shown below in Figure 2-2. Each control sequence can be set up for an Automatic start and/or a Manual start, and a Disable and a repetition rate.

3 Bit Numbering Convention

Bit Number	1 - 4 - 8
Printout	XXXXXXXX

Figure 2-2 3 Bit Numbering Convention

These control sequences are very powerful and flexible. For example, if a given control function can not be completed in one sequence, sequences can be concatenated by having one bit in the last event signature of one sequence start another sequence. In a like manner, a bit or bits in one sequence can be used to enable or disable another sequence.

2.10.2 Setting an Automatic Start Sequence

From the **I** key command (Initialization) select Init Sequences. The system asks Init Sequences (1) or Label Sequences (2). Select Init Sequences. The system then responds with a request for a sequence number. At the Sequence # prompt, choose a sequence number to initialize. In Figure 4.1, the operator has selected sequence number 1. The system responds with a setup prompt. In the following discussions, operator responses are in **bold.** Although there are 40 bits available, for clarity only 16 are presented in the following discussion. All operator inputs are terminated with a carriage return **<Enter>**. Entering a carriage return in response to a question leaves the setting unchanged and the CPP proceeds to the next question. Entering an **R<Enter>** at anyone of the T? Duration questions moves the initialization procedure to the Rep Rate question. Entering a backspace results in the CPP backing up one T time and reasking the question.

A start time of 24:00:00 disables the sequence from Automatically starting. The sequence can still be started from a Manual input. In fact, this is how a manual only start or a sequence that starts if not disabled are configured, with a start time of 24:00:00. The T times, output signatures and the rep rate are set up using the A (Automatic) start key command.

A Rep Rate of all zeroes is interpreted by the CPP as 24 hours, and is really not used. The starting of a sequence is also controlled by the Day To Start entry. If a zero is entered for the Day To Start, then the sequence will start when its start time is the same as the CPP clock, which will occur once a day. If a Day to start is entered, then the sequence will not start until the start time of the selected Day To Start is encountered. After starting, sequencers run from the Rep Rate. Following a power fail or a time change, the Day To Start criteria must again be met before the sequence will restart. The last selection, NxTdy(8), allows the sequence to be started eight days from the present day (next today).

Once started, each T Duration setting is the amount of time that this particular event signature is to be output from the previous T setting. In the example below, T1 will last 1 minute from the start time, from 23:45:00 until 23:46:00, at which time the CPP will proceed to T2. The T2 signature will be output for 1 minute from the end of the T1 Duration, from 23:46:00 until 23:47:00. The T3 signature will only be held for 5 seconds, from 23:47:00 until 23:45:05. The T3 setting is used to reset the seventh output bit that was set in T2. A T Duration of all zeroes indicates that no more event signatures in this sequence are used.

Once an Automatic start sequence is started, it runs from the Rep Rate. The Rep Rate is the time from the start time until the sequence is started again. A Rep Rate of 012:00:00 will start every 12 hours.

```
        Sequence # (1-8)  1<Enter>
        (A)uto start, (M)anual start, (D)isable,  (N)ot Enabled =N A<Enter>
        Start Time     00:00:00  23:45:00<Enter>
        Restart Type  D  <Enter>
        T1 Duration   00:00:00  00:01:00<Enter>
          ( 1-8)   XXXXXXX  XXXX0001<Enter>
          (9-16)  XXXXXXX  <Enter>
        T2 Duration   00:00:00  00:01:00<Enter>
          ( 1-8)   XXXXXXX  XXXXXX10<Enter>
          (1-16)  XXXXXXXX  <Enter>
        T3 Duration  00:00:00  00:00:05<Enter>
          ( 1-8)   XXXXXXXX  XXXXXX0X<Enter>
          (1-16)  XXXXXXXX   <Enter>
        T4 Duration  00:00:00  <Enter>
          ( 1-8)   XXXXXXXX   <Enter>
          (1-16)  XXXXXXXX   <Enter>
        T5 Duration   00:00:00  R<Enter>

        Rep Rate  000:00:00  012:00:00<Enter>
Day To Start


Any(0), Sun(1), Mon(2), Tue(3), Wed(4), Thr(5), Fri(6), Sat(7), NxTdy(8) = 0 <Enter>
```

Figure 2-3 Setting Up Automatic Start Sequence

The Restart Type question lets the CPP know how to restart a sequence following a power fail or a time change. The restart criteria apply only to a sequence that has been set up to automatically start. Following a power fail or a time change, all control sequences are stopped and must be restarted to be in synchronization with the system clock.

The acceptable responses to this question are presented in Table 2.3 below. Daily, or a D input, simply means that the sequence should start at the next start time that has been set into the sequence. To eliminate any confusion, under the minutes column, an M05 entry will result in the sequence starting on the next 05, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, or 00 minute time boundary encountered.

Mxx = Minutes	Hxx = Hours	D = Days
xx	xx	
01 = Any minute	01 = Any hour	
02 = Any even minute	02 = Any even hour	
03 = Any third minute	03 = Any third hour	
05 = Any fifth minute	04 = Any fourth hour	
06 = Any sixth minute	06 = Any sixth hour	
10 = Any tenth minute	08 = Any eighth hour	
12 = Any Twelfth minute	12 = Any twelfth hour	
15 = Any fifteenth minute		
20 = Any twentieth minute		
30 = Any thirty minute		
60 = Any hour		

Figure 2-4 Restart Types

Reviewing the T event signatures set in by an operator in Figure 2.1 above, bits 1-4 and bits 9-16 are not used by this sequence. This sequence uses only bits 5-8, which are set as presented in Figure 2.2 below.

Bit	T1	State	T2	State	T3	State	T4	State
5	0	Off	X	No Change	X	No Change	X	No Change
6	0	Off	X	No Change	X	No Change	X	No Change
7	0	Off	1	On	0	Off	X	No Change
8	1	On	0	Off	X	No Change	X	No Change

T1	23:45:00	Until	23:46:00
T2	23:46:00	Until	23:47:00
T3	23:47:00	Until	23:47:05
T4	23:45:05	Until	11:45:00 of next day

Figure 2-5 States of Bits 5 thru 8

During T4 no bits are being controlled (set or reset) by this sequence. Other system routines have access to the same output bits. Attention should be paid to the assignment of outputs required for a given sequence. The use of Xs allows other routines to share common outputs and set and reset only those that apply. T times are executed in sequence until a T time with a time of all zeroes is detected or until all nine event times have been executed. The sequence then operates from the Rep Rate.

If a Day To Start had been entered, the CPP question would be as given below. The statement presented by the CPP, 2.4, indicates that the sequence is set up to start on Monday, and that the system clock is four days from Monday. The operator entered Tuesday (3) as the Day To Start. If the CPP printed this statement again, it would print

3.5, indicating that the system clock is five days from Tuesday. This same information (3.5) is presented in an L Key command following the sequence start time and restart criteria. The restart criteria should be set to a D.

```
Day To Start


Any(0), Sun(1), Mon(2), Tue(3), Wed(4), Thr(5), Fri(6), Sat(7), NxTdy(8) = 2.4
```

3<Enter>

2.10.3 Setting a Manual Start Sequence

Using the Manual start sequence, a control sequence can be started based on the state of digital outputs or digital inputs. In parallel with the Automatic start feature, the Manual start feature can also start the same control sequence. As an example, this feature is useful when the system is set up to conduct a daily instrument calibration operating from the system clock. The Manual start provides a means to conduct a calibration on demand with the use of digital output or input bits.

If the sequence is already running when the Manual start is given, it is ignored. If the sequence has been started manually and the automatic start time occurs, the automatic start is ignored. The rep rate is not affected by the Manual start feature.

In addition, a sequence can be set up to be initiated manually only. This requires that the start time of the sequence be set to 24:00:00 which disables the automatic start feature. It should be noted that even with the Manual start, the rep rate for a sequence must be appropriately set. If the Rep Rate is too short, not all of the event signatures for the sequence will be set prior to the sequence either starting again or stopping based on the condition of the Manual start bits.

At the SETUP SEQUENCE prompt, type an **M** to set up a Manual start sequence. The system responds with a start condition prompt. Figure 2.3 below provides a typical manual start set up. As before, although the CPP has 40 output and 40 input points, for clarity, only 16 are presented in Figure 2.3.

The CPP first asks for external (digital inputs) settings that will start the sequence. It then asks for internal (digital outputs) settings that will start the sequence. This is and "OR" condition. Either setting will start the sequence. If the sequence is already running, a new start condition is ignored. At the completion of the sequence timing, if a start condition still exists, the sequence will start again.

In a like fashion, the individual start bits are also "OR" condition. If any bit set up as starting the sequence is detected as true, then the sequence is started. In the Boolean section of the system initialization (I key command), a series of "AND" and "OR" conditions for both the internal bits and the external bits can be set up. This condition can then set one bit that can be used to start a sequence.

```
SEQUENCE #  (1-8)  2<Enter>
(A)uto start, (M)anual start, (D)isable,  (N)ot Enabled =M <Enter>
Start On External
( 1-8)  =  XXXXXXXX  1XXXXXX0<Enter>
(1-16) =  XXXXXXXX  <Enter>
Start On Internal
( 1-8)  =  XXXXXXXX  X1XXXXXX<Enter>
(1-16) =  XXXXXXXX  <Enter>
```

Figure 2-6 Set Up Manual Start Sequence

In the example, the user selected sequence number 2, and selected a Manual start. The CPP then asks what external conditions (digital inputs) will start the sequence. In this example, the user has set up a "1" on bit number one or a "0" on bit number eight of the digital inputs will start the sequence.

The CPP then asks what internal conditions (digital outputs) will start the sequence. In this example, the user has set up a "1" on output bit number two to start the sequence.

2.10.4 Disabling A Sequence

Sequences can be disabled based on digital inputs or digital output settings. The converse of this is also true. A sequence can be enabled if the disable conditions are not met. If conditions disabling a sequence are met, a sequence not running will not start, and a sequence running will stop and reset all of it's associated control outputs.

Figure 2.4 below provides a disable sequence set up. At set up sequence prompt the user inputs a D to set up disable conditions.

```
SEQUENCE # (1-8)  3<Enter>
(A)uto, (M)anual, (D)isable,  (N)ot Enabled  D<Enter>
Disable On External
( 1-8)  =  XXXXXXXX  <Enter>
(9-16) =  XXXXXXXX  <Enter>
Disable On Internal
( 1-8)  =  XXXXXXXX  0XXXXXXXX<Enter>
(9-16) =  XXXXXXXX  <Enter>
```

Figure 2-7 Set Up Disable Sequence

In the above example the user has set up only one internal bit to disable the sequence. Any time bit number one of the control outputs is at a zero (not set) the sequence is disabled.

2.10.5 Controlling Sequences From a Terminal

The user can exercise control over the sequences via comm port #4. A sequence can be started, stopped, or instructed to conduct a single pass. This is invoked with an SEQ input, or an SEQx input where x is the desired sequence number. An example is presented below in figure 2.5

```
SEQ<Enter>
Seq # = 3<Enter>
Start Seq    =  1
Stop Seq     =  2
Single Pass  = 3
Choice = 3<Enter>
```

Figure 2-8 Controlling Sequences from the Comm Port

Starting a sequence results in the sequence starting at the present time, and transfers the rep rate into the counter. From this time forward, sequences set up as automatically starting run from the entered rep rate. Should a power fail or time change occur, the system uses the entered start time and the restart criteria to restart the sequence. If the sequence is not set up as an automatic start sequence, then after all T times have been completed, the sequence will not restart, unless the conditions initiating a manual start are met.

Stopping a sequence results in the sequence being stopped at the present time. If the sequence is set up to automatically start, the sequence will restart at the entered start time. A sequence set up to manually start will restart when starting conditions are met.

A single pass starts a single pass through all T times. The original rep rate in the counter is not changed. Therefore this sequence continues to cycle at the rep rate related to the entered start time.

2.10.6 Labeling Control Sequences

The control sequences can also be labeled, which is very useful when service personnel are working on the system. It allows them to more readily fit the various pieces together to understand how the system is configured. Up to 40 characters can be used to label each sequence. If more than 40 characters are input, the system truncates at 40 characters. If less than 40 characters are needed, then terminate the string with a carriage return.

Invoke the system initialization by typing in an I, and selecting the INIT SEQUENCES function. The system then asks INIT SEQUENCES (1) or LABEL SEQUENCES (2) ? Select the LABEL SEQUENCES path. The system then responds with SEQUENCE #. Select the desired sequence number. The system then asks for the label string. The system stores the label and then asks for another sequence number. A carriage return at this point exits the set up routine. An example is presented in Figure 2.6 below.

```
Init Sequences (1) or Label Sequences (2)  2<Enter>
Sequence # = 1<Enter>
Label = Stream Switch NOX
Stream Switch NOx Every 6 Minutes <Enter>
Sequence #  <Enter>
```

Figure 2-9 Labeling Control Sequences

2.10.7 Summary Listing Of The Control Sequence Set Up

Presented below is a section of the summary printout obtained from the CPP using the L key command. This section of the printout represents the current control sequence set up for two sequences, although on is not enabled. The line numbers (1 through 21) at the far left hand edge do not appear in the printout and are used for reference only.

```
1   Sequence #1
2   Label = NOx 6 Minute Stream Switching
3   Sequence Setup          (1-8)          (9-16)       (17-24)       (25-32)       (33-40)
4   Start on External     XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX
5   Start on Internal     XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX
6   Disable on External   XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX
7   Disable on Internal   XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX

8   Start Time  01:00:00  M06
9         T1    00:06:00    0XXXXXXX  XXXXXXXX   XXXXXXXX  XXXXXXXX  XXXXXXXX
10        T2    00:06:00    0 1 XXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX
11        T3    00:00:00    XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX
12        T4    00:00:00    XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX
13        T4    00:00:00    XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX
14        T5    00:00:00    XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX
15        T6    00:00:00    XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX
16        T7    00:00:00    XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX
17        T8    00:00:00    XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX
18        T9    00:00:00    XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX
19        RR 000:00:00      XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX  XXXXXXXX

20        Sequence #2
21        Not Enabled
```

Figure 2-10 Listing of Control Sequence Set Up

Line 1 Identifies the sequence number (number 1)
Line 2 Presents any label the sequence has been given
Line 3 Header
Line 4 Indicates the input bits for externally starting the sequence
Line 5 Indicates the output bits for internally starting the sequence
Line 6 Indicates the input bits for externally disabling the sequence
Line 7 Indicates the output bits for internally disabling the sequence
Line 8 Indicates the start time and the restart type
Lines 9 -18 Shows each Tx time and its associated output signature
Line 19 Shows the sequence repetition rate

Line 20	Indicates the sequence number (number 2)
Line 21	Indicates that this sequence is not enabled

2.11 Calibrations

The multipoint calibration feature of the CPP supports two types of calibration data capture. Both are based on the digital I/O. One uses the output bit settings to either set relays which switch in calibration gases, or sense input bits to indicate that a particular gas has been directed to an instrument. The other is based on the serial communications with the instrumentation controlling either solenoids are circuitry inside the instrument to conduct the calibration. The serial calibration commands are sent based on digital IO settings. In addition, the I/O type supports two types of calibration capture: internal and external. The CPP also supports and detects an independent calibration conducted by the instrument or by an operator controlling the instrument.

2.11.1 I/O Calibration Capture

Both the digital inputs and the digital outputs can be used to flag the occurrence of a calibration.

2.11.1.1 Internal I/O Calibrations

Allows the CPP to actively control the instrument calibrations based on the CPP clock. Internal calibrations are accomplished by assigning zero and span functions to selected digital outputs and setting up the sequencers with start times, durations, and repetition rates for each output bit associated with the calibrations. The relay outputs of the CPP are used to close solenoid valves to switch calibration gases to the instruments.

2.11.1.2 External I/O Calibrations

Uses incoming digital bits to indicate when a zero or span is active. External calibrations are accomplished by assigning zero and span functions to selected digital inputs.

2.11.2 Serial Calibration Capture

The CPP can also be set up to send serial strings to instruments to control calibrations. The CPP can also sense status information returned by instruments indicating that a calibration is in progress.

2.11.2.1 Internal Serial Calibrations

Allows the CPP to actively control the instrument calibrations based on the CPP clock. Internal calibrations are accomplished by assigning zero and span functions to selected digital outputs and setting up the sequencers with start times, durations, and repetition rates for each control bit associated with the calibrations. Upon detecting the presence of a bit setting, the CPP queues up and sends the appropriate command to the instrument.

2.11.2.2 External Serial Calibrations

Same as the internal calibrations presented above, except the CPP senses incoming digital bits instead of output bit settings.

2.11.2.3 Independent Serial Cals

The CPP senses that an instrument is conducting an autonomous calibration and flags the collected data appropriately.

In all cases the CPP captures start times, stop times, and calibration data based on the occurrence of a digital output or digital input which has been assigned a zero or span function. Additionally, most instruments support independent, internal calibrations. Independent of the I/O settings. If the CPP detects that an instrument is in a calibration, data is flagged with a calibration modifier and the results stored in the calibration table.

2.11.3 Initialization

The calibrations are set up from the main menu. An example is presented below. The system first asks for the number of global spans that are to be enabled. This number is the same for all channels and should be set to the largest requirement in the system. If this number is changed the system prints a message relating that changing this set up parameter has system wide ramifications and asks if it is to proceed or revert back to the old setting. Following this, the system asks for a CPP channel number.

```
1.  Digital Cal Set up
2.  Number of Spans = 1<Enter>
3.  Digital (D) or Serial (S) = D<Enter>
4.  Chan # =  1<Enter>
5.  Cal on Digital I/O
6.  Outputs(1), Inputs(2), Both(3), Disable(0) = 0 1<Enter>
7.   Zero on
8.  (1-8)   = . . . . . . . .  = Z . . . . . . . <Enter>
9.  (9-16) = . . . . . . . . = <Enter>
10. Zero Value = 0.000E+0 = <Enter>
11. Span #1 on
12. (1-8)   = . . . . . . . .  = . S . . . . . . <Enter>
13. (9-16)  = . . . . . . . .  =  <Enter>
14. Span #1 Value = 0.000E+0 = 2.500E+2<Enter>
15. Chan # =  <Enter>
```

Figure 2-11 Typical Calibration Set Up

2.11.3.1 Digital Calibrations

Line 1 A statement

Line 2 Asks for the number of spans that the CPP should allow for. This is a global number and applies to all calibration channels.

Line 3 Allows the user to select either the digital set up or the serial set up.

Line 4 Asks for the CPP channel number to assign as a calibration channel.

Line 5 A statement

Line 6 Asks which I/O bits to tie into the calibration. A 1 selects the internal relay settings, a 2 selects the incoming digital signals, and a 3 selects both. If both is selected then the same internal and external bits must be used for the same functions.

Lines 7-9 Ask which bits indicate that a zero calibration is taking place. In the above example, the user set up bit number 1 to indicate that this channel is in a zero calibration mode. If the CPP is sensing the digital I/O for calibrations, then when sensing this bit the data is flagged with a "C" or calibration modifier and the start and end times and last (zero) value are saved and recorded as the zero calibration value.

If the CPP is conducting calibrations over the serial port, then sensing this bit going true results in the CPP sending a set zero calibration mode command to the instrument connected to this channel. Data from the instrument is flagged with a "C" or calibration modifier. When the bit goes away, the CPP sends a exit zero calibration mode to the instrument and records the start and stop times and the last data value as the zero calibration value.

Line 10 Asks what the expected zero value should be.

Lines 11-14 The same as lines 3 through 10 except they apply to span values instead of zero values.

Line 15 Asks for the next channel to be set up as a calibration channel. A carriage return at this point exits the routine.

2.11.3.2 Serially Controlled Calibrations

The CPP supports serial control of instrument calibrations and calibration instruments, such as mass flow controllers. Calibrations are all tied into the digital output bits of the CPP. As described in the CPP manual, digital I/O bit settings apprise the CPP when an instrument is measuring the sample input, the zero input or the span input. These bit settings can be used to drive relays, which control solenoids, which switch the various gases into the instruments.

In addition, the serial control of calibrations can be tied to the digital output bit transitions. Upon sensing the setting or resetting of digital output bits, the CPP can be set up to send command strings to instruments and to calibrators. To set up serial control of calibrations, from the Main Menu select the calibration set up. The CPP responds with:

Line 1 Number of Spans = 1**<Enter>**
Line 2 Digital(D) or Serial(S) = **S<Enter>**

As discussed above, the digital path allows bit settings to apprise the CPP of the type of input to each instrument. As an example, digital output bit #30 being set true indicates that the SO2 analyzer (on CPP channel #02) is in the zero mode. Digital output bit #31 being set true indicates that the SO2 analyzer is in the span mode. These digital outputs can be used to directly drive solenoids, or to start sequences inside calibrators that control and conduct instrument calibrations.

In addition to using the digital outputs of the CPP to drive solenoids, the CPP can be configured to sense transitions of digital output bits, and to send serial commands to instruments to close solenoids and to start calibration sequences. The calibration bit assignments in the digital path must still be set up as before.

The CPP can be set up to simultaneously control calibrations with relay closures, control calibrations with serial commands, or sense contact closures as calibration indicators. Column #1 below describes the setting up of a serially controlled zero calibration, and column #2 describes the setting up of a digitally controlled span calibration. Prior to initializing the serial calibration features, the channels must be set up in the CPP.

At the bit number question, inputting a bit number followed by a C will clear all serial conditions set up for this particular bit transition. For example, inputting **30TC<Enter>** will clear all serial conditions for bit #30 going true. The CPP responds with Bit #30 Cleared. The commands presented with each question will vary depending on the manufacturer and the type of instrument connected to the channel selected.

2.11.3.3 Example Using API Instrument - API Calibrator

In the following example, an API SO2 analyzer is initialized in CPP channel #02, and an API model 700 mass flow controller is initialized in CPP channel #20.

	Column #1 - Set Up Zero	Column #2 - Set Up Span
Line 3	Bit # = **30T<Enter>**	Bit # = **31T<Enter>**
Line 4	Cmd #1	Cmd #1
Line 5	Chan # 00 **2<Enter>**	Chan # 00 **2 <Enter>**
Line 6	API 100A SO2	API 100A SO2
Line 7	Command	Command
	01 = Start Zero	01 = Start Zero
	02 = Stop Zero	02 = Stop Zero
	03 = Set Zero	03 = Set Zero
	04 = Start Span	04 = Start Span
	05 = Stop Span	05 = Stop Span
	06 = Set Slope	06 = Set Slope
	Selection = 00 **1<Enter>**	Selection =00 **4<Enter>**
Line 8	Cmd #2	Cmd #2
Line 9	Chan # 00 **20<Enter>**	Chan # 00 **20<Enter>**
Line 10	API 700 MFC	API 700 MFC

Line 11	Command	Command
	01 = Str SO2	01 = Str SO2
	02 = Str NOx	02 = Str NOx
	03 = Str CO	03 = Str CO
	04 = Str O3	04 = Str O3
	05 = Str HC	05 = Str HC
	06 = Str GPT	06 = Str GPT
	07 = SPC01	07 = SPC01
	08 = SPC02	08 = SPC02
	09 = SPC03	09 = SPC03
	10 = SPC04	10 = SPC04
	11 = SPC05	11 = SPC05
	12 = SPC06	12 = SPC06
	13 = Standby	13 = Standby
	Selection = 00 **1<Enter>**	Selection = 00 **2<Enter>**
Line 12	Cmd #3	Cmd #3
Line 13	Chan # = 00 **<Enter>**	Chan # = 00 **00<Enter>**
Line 14	Bit # = **30F<Enter>**	Bit # = **31F<Enter>**
Line 15	Cmd #1	Cmd #1
Line 16	Chan # = 00 **2<Enter>**	Chan # = 00 **2 <Enter>**
Line 17	API 100A SO2	API 100A SO2
Line 18	Command	Command
	01 = Start Zero	01 = Start Zero
	02 = Stop Zero	02 = Stop Zero
	03 = Set Zero	03 = Set Zero
	04 = Start Span	04 = Start Span
	05 = Stop Span	05 = Stop Span
	06 = Set Slope	06 = Set Slope
	Selection = 1 **2<Enter>**	Selection = 1 **5<Enter>**
Line 19	Cmd #2	Cmd #2
Line 20	Chan # = 00 **20<Enter>**	Chan # 00 **20<Enter>**
Line 21	API 700 MFC	API 700 MFC
Line 22	Command	Command
	01 = Str SO2	01 = Str SO2
	02 = Str NOx	02 = Str NOx
	03 = Str CO	03 = Str CO
	04 = Str O3	04 = Str O3
	05 = Str HC	05 = Str HC

	06 = Str GPT	06 = Str GPT
	07 = SPC01	07 = SPC01
	08 = SPC02	08 = SPC02
	09 = SPC03	09 = SPC03
	10 = SPC04	10 = SPC04
	11 = SPC05	11 = SPC05
	12 = SPC06	12 = SPC06
	13 = Standby	13 = Standby
	Selection = 1 **13<Enter>**	Selection = 1 **13<Enter>**
Line 23	Cmd #3	Cmd #3
Line 24	Chan # = 00 **<Enter>**	Chan # = 00 **<Enter>**
Line25	Bit # = **<Enter>**	Bit # = **<Enter>**
	Return To Main Menu	Return to main Menu

Figure 2-12 Typical API Instrument Set Ups

Referencing column #1 to begin with, in the above example, sending commands to an API model 100A SO2 instrument and an API model 700 calibrator have been set up. The commands start sequence #1 in the model 700 calibrator and at the same time switch the zero air input solenoid in the model 100 SO2 analyzer. The timing control of the digital I/O bits are set up in the sequencers. Tying the digital I/O bits to the calibration is set up in the Digital pass of the above command (line 2).

Line 1 Sets up the number of spans in the system.

Line 2 Allows selection of tying the digital I/O bits to the calibration channels or setting up the serial commands to send to instruments. In this example the serial set up is selected.

Line 3 Asks which digital output bit is to be monitored. The input 30T indicates that when digital output bit # 30 goes true, the following commands are to be sent. Four commands are allowed for the setting or resetting of each bit. If more than four commands are needed then two bits can be set in the sequence set up.

> **Note** - Only one command can be sent to any instrument on a bit change. For example, a command resetting the zero mode and setting the span mode in the same instrument **can not** be sent on the same bit transition. The sequencers must be set up to send the reset zero command, and then set another bit to send the set span command.

Line 4 Indicates that command #1 is being set up. There are four commands available with each bit transition.

Line 5 Asks which channel this command is to sent out. In the example, channel #2 was selected. The CPP then prints the manufacturer and model number of the instrument initialized in this CPP channel. The CPP prints that it found an API model 100A SO2 analyzer in CPP channel #2.

Line 6 Prints the manufacturer and model number found initialized in this CPP channel.

Line 7 Presents a list of commands that are supported for this instrument. This will vary depending on the manufacturer and the type of instrument. In the example, the command to close the zero air solenoid was selected.

Line 8 Indicates that command #2 is to be set up.

Line 9 Asks which CPP channel this command is to be sent out. Channel #20 was selected. The CPP print that an API model 700 mass flow controller was found in CPP channel #20.

Line 10 Prints the manufacturer and model number found initialized in this CPP channel.

Line 11 Asks for the command to send. In this example, a command to start sequence #01 in the model 700. This sequence should have been programmed to provide a zero air output.

Line 12 Indicates that command #3 is to be set up.

Line 13 Asks which CPP channel the commands are to be sent out. Selecting a channel # of zero indicates that there are no more commands associated with this bit transition. A carriage return is input as there are no more commands to set up for this bit transition

Thus far, when the CPP detects a true transition on bit #30, it will queue up and send out comm port #3-02 (which was initialized in CPP channel #2), a set zero solenoid command to an API model 100A SO2 analyzer, and queue up and send out comm port #3-08 (which was initialized in CPP channel #20), a start SO2 Sequence command to an API model 700 calibrator. In the digital portion set up, the true state of this bit (output bit #30) should indicate to the CPP that the SO2 analyzer is in the zero air mode.

Line 14 Asks for the next bit transition. The input #30F indicates that when digital output bit #30 goes false, the following commands should be sent.

Line 15 Indicates that command #1 is to be set up,

Line 16 Asks for the CPP channel number the command is to be sent out. 02

Line 17 Prints the manufacturer and model number found on this channel. API 100A SO2

Line 18 Asks for the command. Reset zero air solenoid

Line 19 Indicates that command #2 is to be set up.

Line 20 Asks for the channel #. 20

Line 21 Prints the manufacturer and model number. API 700 MFC

Line 22 Asks for the command. Standby

Line 23 Indicates that command #3 is to be set up.

Line 24 Asks for the Channel number. 00

Line 25 Asks for the next bit transition. No More

The last section sends a reset zero air command out comm port #3-02 to an API model 100A SO2 analyzer. It also sends a return to standby command out comm port #3-08 to an API model 700 calibrator.

The digital portion should have been set up so when detecting the false transition of digital output bit #30, the CPP takes the last reading as a zero value.

Column #2 sets up a span on the SO2 analyzer. As an example, the sequencers could be set up so that at a particular time (12:00:05) the CPP sets bit #30. The CPP sends a START ZERO command to the SO2 analyzer, which closes the internal solenoid, and sends a START SEQUENCE #01 command to the M700 calibrator. Fifteen minutes later (12:15:05) the CPP resets bit #30 and sets bit #31. At 12:15:05 the CPP takes a zero reading from the SO2 analyzer and sends a STOP ZERO mode to the unit. It also sends a RETURN TO STANDBY command to the M700 calibrator.

A flag is set for each comm port that is queued up to be sent a command. The serial calibration firmware has to vie for the comm ports with the poller and possibly commands from an operator. All commands are interleaved, so it normally appears instantaneous to the user. However, some older API instrumentation can take several seconds to respond to an interrogation. In such a case it could be several seconds before the serial controller firmware is given access to a given comm port. The poller and operator initiated commands are given priority over the serial calibration control.

So given the response time of some instruments, and if they are being polled when the serial calibration is taking place, it could be 12:15:01, 02 or even 03 before the serial calibration command can be sent.

At 12:15:01 the CPP will send a START SPAN command to the SO2 analyzer, which closes the internal span solenoid, and also sends a START SEQUENCE #2 command to the M700 calibrator. As explained above, it could be 12:15:02, 03, or even 04 before the command is actually sent to the instrument. Fifteen minutes later (12:30:00) the CPP resets bit #31. The CPP takes a span reading from the SO2 analyzer and sends a STOP SPAN command to the unit. It also sends a STOP SEQUENCE command to the M700 calibrator.

The above has conducted a zero /span on the SO2 analyzer.

The L key command prints as follows.

```
Bit #   Cmd #      Chn#-CP#    Manu     Model    Command
30T      01         02   02     API      100A    01 Str Zero
30T      02         20   08     API       700    01 Str Seq #01

31T      01         02   02     API       100A   04  Str Span
31T      02         20   08     API        700   03  Str Seq #02

30F      01         02   02     API       100A   02  Stp Zero
30F      02         20   08     API        700   02  Standby

31F      01         02   02     API       100A   05  Stp Span
31F      02         20   08     API        700   04  Standby
```

Figure 2-13 Serial Calibration Commands

2.12 Boolean Functions

The CPP-4794 provides Boolean mathematics functions. The standard number of Boolean functions is sixteen (expandable). Each function can have up to eight terms. The input terms are digital input and output bits. Input bits are input as I01 through I40, and I01- through I40-, where the minus sign (-) indicates a not or false term. The output bits are specified as O01 through O40, and O01- through O40-, where the - indicates a not or false term.

An I01 input is true when the system senses input bit number 01 as a one, and is false when input bit number 01 is a zero. Conversely, an I01- input is true when the system senses input bit number 01 as a zero, and false when input bit number 01 is sensed as a one. An O01 term is true when output bit number 01 is set, and is false when output bit number 01 is reset. Conversely, term O01- is true when output bit 01 is reset and false when output bit number 01 is set.

2.12.1 Setup

The ampersand (**&**) is used to indicate the **and** condition. The plus symbol (+) is used to indicate an **or** condition. An example of the Boolean functions are provided below. Operator inputs are in **bold**.

BF #, List(L), Label(LB) **1<Enter>**

```
BF #1 This is an example
IF  I01&I02&I03-+O04
THEN SET O21
IF  I01&I02&I03-+O04- <Enter>
THEN SET  O21,I09<Enter>
```

Figure 2-14 Typical Boolean Functions

When the Boolean mathematics set up routine is entered, the system asks BF #, List (L), Label(LB). Inputting a number from 1 to 20 selects a particular Boolean function to set up. In our example, the operator has selected number 1. The system prints any terms already residing in this function, and the pauses after asking IF. A carriage return leaves this setting unchanged, and the system asks THEN SET and pauses. If the function has not been set up the system just asks IF. Inputting an L commands the system to print a listing of all functions that have been set up. Inputting an LB enters a routine that allows each Boolean function to be given a label.

Each function is limited to eight terms. There are eight terms for both the IF and the THEN SET terms. Longer Boolean functions can be implemented by concatenating the BF functions. The system starts processing at BF #1 and proceeds in sequence to BF #20.

To disable a function that was previously set up, enter a **D<Enter>** at the IF question.

Each second, the system resets the digital output working registers. It then proceeds to process the control Sequencers, followed by the BF functions. Any condition that is met is set. If no Boolean condition to set or reset a particular bit is met then it remains unchanged. Otherwise, it is set or reset as determined. After all functions have been completed the working digital output register is transferred to the actual digital output register. The digital input register is operated on before the output register.

A similar operation is used with digital input bits. First the actual digital inputs are read and stored in a working register. Then if any set or reset input bit condition is met, that particular input bit is set or reset in the working register, regardless of what actually came in from the read command. The working register is then transferred to the actual input register. The input register is serviced before the output register.

In our example, the user has corrected the IF term O04 to O04- and expanded the THEN SET points in this BF number. This function says that if digital input bits number 01 **and** 02 are true (ones) **and** digital input bit number 03 is false (zero), **or** output bit number 04 is not set, THEN SET output bit 21 and force digital input bit number 09 to a one. Multiple THEN statements may be separated with a comma or a space. A carriage return accepts the input string and returns to the BF# prompt.

The IF input can consist of nine **and/or** conditions or ten terms. If less are used, as in our example, the entry is terminated with a carriage return. The THEN SET input can consist of ten terms as well. If less are used, the entry is terminated with a carriage return.

2.12.2 Labeling Boolean Functions

Inputting an LB enters a routine that allows labels or names to be given to the Boolean functions. The CPP asks for a Boolean function number. A carriage return exits this routine. If a number is entered, any label given the entered Boolean function number is presented, and followed by New Label -. A carriage return at this point returns to the BF# question. Any other entry is accepted as the start of a New Label. Up to 60 characters can be entered as a label name.

If a label has not been set up the CPP prints None -, and waits for an entry. A carriage return exits this. Any other entry is accepted as a label. An existing label can be cleared by entering an up arrow, ^, when the CPP is awaiting a label entry.

2.13 Alarms

The CPP alarm firmware allows an operator the means of detecting, announcing and logging alarm conditions. The alarm can be generated from the one minute sample, the interim average, or the final average. If an alarm is detected multiple digital outputs can be generated, and an alarm message logged on a local terminal.

2.13.1 Alarm Set Up

An example alarm setup is presented below. The line numbers on the left margin appear for referencing only, and are not present in an actual system.

```
1.    Alarm # = 03<Enter>
2.    Chan # = 00  1<Enter>
3.    Hi(H), Lo(L), HiW(HW), LoW(W) = H <Enter>
4.    Dis(0), Instr(1), Value(2), Both(3) = 0 2<Enter>
5.    Scan(S), Prel(P) or Int(I) or Fin(F)  = I  <Enter>
6.    Print Alarm Y/N N Y<Enter>
7.    Set =  +0.000E+0  5.000E+1<Enter>
8.    On  Alarm Set
      (1 - 8)   = XXXXXXXX  XXXXXXX1<Enter>
      (9-16)  = XXXXXXXX   1XXXXXXX<Enter>
9.    Alarm # = <Enter>
```

Figure 2-15 Typical Alarm Set Up

Line 1 The CPP is asking for the alarm number. One through forty is the normal range. If requested, more alarm channels can be added to the CPP. Inputting an “A” at this point will result in the CPP outputting the set up status of the alarm channels similar to the AL key command from the Main Menu. This is very useful in determining which alarms have not been set up and are available for use.

Line 2 Asks which data channel is to be used with this alarm. A channel number of zero disables this alarm.

Line 3 The CPP asks if this is to be a high, low alarm or warning. Entering an L or an LA selects a low alarm. Entering an H or an HA selects a high alarm. Entering an LW selects a low warning, and entering an HW selects a high warning. The CPP treats high and low alarms and high and low warnings identically. However, the user can set up different actions for the two. For example, a warning could be set up to automatically print the alarm, whereas the alarm level could be set up to close a relay.

Line 4 This input selects what incoming data to check for alarms. The Dis disables this alarm. A one input selects status being received from instrumentation over a serial input. A two selects setting the alarm on the data value being received, from a serial port or an A/D converter. A three sets up an alarm on either condition. If a one or three is input and the channel selected is getting data from an A/D converter, then an error message is printed and the question asked again.

Line 5 The CPP is asking what type of data is to be checked for an alarm condition. Scan is each A/D sample, Prel is the preliminary average, Int is the interim average, and Fin is the final average. In the example the operator entered an I instructing the CPP to check the interim averages for alarm conditions.

Line 6 The operator set up an output print in an alarm condition. The output is sent to the local comm port. An example is presented below.

Line 7 Allows the alarm level to be set. In an alarm condition, data values printed locally will be appended with an 'A' modifier if it is a high alarm and with an “a” if it is a low alarm condition, and a “W” for a high warning and a “w “ for a low warning.

Line 8 Allows an operator to set up the digital outputs to activate in an alarm condition. These bits are set as long as the selected value is in an alarm condition and resets when the value returns to a non alarm condition. For clarity only 16 digital outputs are shown. In an actual system this may be forty or more digital set points.

Line 9 Asks for next alarm channel to be set up. A carriage return here returns to the Main Menu. If the auto print alarm is enabled, an alarm message will be printed when a channel goes into an alarm condition, and again when it returns to a non alarm condition. An example is presented below.

2.13.2 Alarm Prints

```
12:35:00  07-25-97  ALR #03 - CH #01 HA- On  I Value = 51.00  Set Pt =  50.00
12:40:00  07-25-97  ALR #03 - CH #01 HA- Off  I Value = 49.95  Set Pt =  50.00
```

Figure 2-16 Typical On-Site Alarm Print Out

Where;
ALR # is the alarm number
CH # is the data channel number
HA is a high alarm (LA, HW, LW)
The I indicates that the alarm used interim data values. (S,I,P,F)

2.14 Auto Prints

The CPP supports three separate automatic print routines. The first automatically prints reports each time a new preliminary, interim or final average is formed. The second prints reports based on the CPP clock, either hourly, on the hour, or when the final averaging period is completed. The third prints reports only when particular digital output bits are set. All prints can be set up concurrently. The prints are initialized as indicated below. From the Main Menu select the Auto Prints set up.

```
Auto Prints
1.0 None(0), Prelim(1), Interim(2), Final(4) = 0 <Enter>
2.0 None(0), HourlyP(1), HourlyI(2), Daily(4) = 0 5<Enter>

3.0 Digital Output Prints
4.0 Preliminary Y/N = N Y<Enter>
( 1- 8)   XXXXXXXX  = XXXXXXX1<Enter>
(9-16)    XXXXXXXX  = <Enter>
(17-24)  XXXXXXXX  = <Enter>

5.0 Interim Y/N = N <Enter>

6.0 Final Y/N = N <Enter>
Return to Main Menu.
```

Figure 2-17 Initializing Auto Prints

Line 01 The first line allows the user to have the CPP print the averages as they are formed. The sum of the inputs can be used to set up multiple prints. Table 2.4 below depicts this. This is also true of the second question. Inputting a one results in the preliminary averages printing every minute, Inputting a two results in the interim averages printing every time a new interim average is formed. Inputting a four results in the final average printing every time a new final average is formed. Inputting a three results in the preliminary and interim averages printing every time either new average is formed.

Line 02 This allows averages to be printed either on the hour or daily at midnight. Inputting a one results in the last 60-one minute averages being printed on the hour.

Input	Result
1	Print preliminary average only
2	Print interim averages only
3	Print preliminary & interim averages

```
            4        Print final averages only
            5        Print preliminary & final averages
            6        Print interim & final averages only
            7        Print preliminary, interim & final averages
```

Figure 2-18 Multiple Print Set Up Inputs

In the example, the user set up the CPP to automatically print an hourly summary of the preliminary averages and to also print a daily report. The daily report consists of the final averages. If hourly averages are being formed, the CPP prints one report daily at midnight. If thirty minute averages are being formed, the CPP prints two reports daily, one at midnight and one at noon. If fifteen minute averages are being formed, the CPP prints four reports daily at midnight, six am, noon, and six pm.

The user has also set up the CPP to automatically print the preliminary averages if digital output bits are set. In the example the user selected a print of preliminary averages whenever output bit number 8 is set. These prints are quite often used when an alarm condition is detected, which sets the bit, or by a logical (Boolean function) detection of a given monitoring scenario, where more data resolution is desirable.

2.15 Meteorological Set Up

This section is included here for continuity only. Please reference section 7.0 for the set up and initialization of meteorological parameters.

2.16 Data Print Format

The CPP provides printouts in either scientific (1.000E+1) or in engineering units (10.00). This selection asks which printout is desired. This can also be changed without entering the Initialization Mode by inputting **PVS<Enter>**. The CPP will ask the same question and respond accordingly.

2.17 Compact Flash Recordings - Long term Data Storage

The CPP has provisions for long term data storage. At the Main Menu prompt select the CFM Recordings set up. The CPP allows various types of data to be recorded to the CFM. The CPP responds with;

1. CFM Recordings
2. Chn’s 1-10 =1, 1-20=2, All = A, = 1 **2 <Enter>**
3. Prelim = N **<Enter>**
4. Interim = N **<Enter>**
5. Final = N **Y<Enter>**
6. Alarms = N **<Enter>**
7. Events = N **<Enter>**
8. Max/Min = N **<Enter>**
9. Digital I/O = N **<Enter>**
10. Calibrations = N **Y<Enter>**

Line 2 Asks how many channels of data are to be recorded on the CFM. If a one is input, then channels 1 through 10 will be recorded on the CFM. If a two is input, then channels 1 through 20 will be recorded on the CFM. If an A is entered, then all channels supported by the CPP will be recorded on the CFM. It takes one half the CFM memory to store only the first 20 channels as it does to store all channels. Equally true, it takes one half as much memory to store 10 channels as to store 20 channels.

Lines 3-5 Ask which of the averages being formed by the CPP are to be recorded on the CFM. It takes the same amount of memory to store a preliminary average as it does to store a final average. To store 24 hours, for 20 channels, of one minute data requires 180K of memory. In contrast, this same amount of memory will store 60 days of hourly averages. None, any or all of the averages being formed can be directed to the CFM.

Line 6 Asks if alarms are to be stored. If so, then the time and value of a channel entering an alarm condition is stored, along with the time and value of the channel leaving the alarm condition. Status flags stored with the data points can also be used to determine alarm conditions.

Line 7 Asks if events are to be stored on the CFM. If so, then any change in the digital I/O gets recorded along with the time.

Line 8 Asks if max/min values are to be stored.

Line 9 Asks if the digital I/O settings are to be stored. If so, along with each data average stored, a snapshot of the digital I/O is recorded as well.

Line 10 Asks if the results of calibrations are to be stored.

In the above example, the user set the CPP up to record the final averages and calibration results. The CPP supports two types of long term data storage. The first uses internal RAM memory and the second used a removable compact flash memory card.

2.17.1 Internal Ram

The CPP has 512K of internal RAM. A portion of this RAM is used to configure the CPP and for system operation. Some of the RAM is used for relatively short term data storage. The CPP stores 3 hours of preliminary averages in internal memory, 4 hours of interim averages in internal memory, and 4 periods of final averages. If hourly averages are being formed, then 4 days of data is being stored. If thirty minute averages are being formed, the 2 days are being stored. If fifteen minute averages are being formed, then 1 day is being stored.

The remainder of the internal RAM, designated as excess internal memory, can be used as longer term data storage. Referencing the above, any of the averages being formed can be directed to the Compact Flash Memory, (CFM). The excess internal memory can store 20 hours of minute averages for 20 channels, or 100 hours of five minute averages, or 8 days of ten minute averages, or 16 days of fifteen minute averages, or 64 days of hourly averages. These can vary somewhat if other activities are also being stored, and are roughly doubled if only 10 channels are being stored.

If and external CFM is not installed, then the CPP stores data in internal memory. If a CFM is installed, then the CPP stores data both in the CFM and excess internal memory.

2.17.2 Removable Long Term Data Storage Memory

The CPP supports a removable CFM as long term data storage. The CFM plugs into a slot in the rear panel, and available in 64 megabyte cards. Eight megabytes of memory will store about 5 days of minute averages for 20 channels, or 45 days of fifteen minute averages.

2.17.3 Data Recovery

The CPP can also read and output data stored in the CFM. This data can be retrieved serially over comm port 1. These transfers are in record delimited blocks. and are hexadecimal characters. H2NS offers a program to retrieve data stored on a CFM.

Reading data from the CFM is conducted in an on line mode. The CPP continues to collect, average and store data, even to the CFM. Likewise, a central computer can be simultaneously polling the CPP on another comm port.

H2NS also offers a separate CFM reader. The unit has two serial ports for data outputs and user interfacing, one printer port for printing data records, and one parallel port for data downloading.

2.17.4 Reading A CFM

TechNote TN19 describes in technical detail the reading of data stored in either the cartridge or the excess internal memory.

2.18 External I/O Of CPP

The CPP has provisions for inputting and outputting analog and digital signals to an external unit, such as a Dutec or OptoMux. The CPP is set up to send and receive external signals over a serial port. This information can also be output in a Modbus protocol over another comm port. The CPP external I/O is set up from the Main Menu. From the Main Menu select External I/O. The CPP responds with the following.

1. External I/O
2. Analog Out(AO), In (AI) - Digital Out(DO), In(DI) = **AO<Enter>**
3. Analog Output Setup
4. DAC(D), or Modbus(M) = **D<Enter>**
5. DigiMux(D), Dutec(U), Opto(O), TSX(T), Internal(I) = **D <Enter>**
6. DAC Address = **080<Enter>**
7. DAC Comm Port # 3-07**<Enter>**

8. DAC Baud Rate 1=300, 2=1200, 3=2400, 4=4800, 5=9600, 6=19.2K(6) = 2 **5<Enter>**
9. Analog Output # = **01<Enter>**
10. CPP Chan # = **3<Enter>**
11. Volt(V) or Current(I) = V = **<Enter>**
12. Hold On Cals Y/N Y = **<Enter>**
13. Hold On Bad Y/N Y = **N<Enter>**
14. Enable Y/N N = **Y<Enter>**
15. Analog Output # = **<Enter>**

Return To main Menu

Figure 2-19 External I/O Set Up

2.18.1 Analog Outputs

The CPP supports both internal DAC's and external DAC's.

2.18.1.1 External DAC's

The following describes setting up external analog outputs.

Line 1 is a heading.

Line 2 Asks for the type, either analog I/O or digital I/O.

Line 3 Indicates that the user selected the analog output set up.

Line 4 Accepts DAC input only.

Line 5 Asks for the Manufacturer of the DAC. The DigiMux, Dutec, Optomux, and TSX are all external DAC's. In this example, the user selected the DigiMux external DAC. Selection of the internal DAC is presented below.

Line 6 Asks for the module address of the DAC.

Line 7 Asks which comm port on the CPP is connected to the analog output module. Inputting a comm port of N indicates to the CPP, that actually there is not a physical DAC connected. This is used to configure the CPP to store data that can be transmitted in a Modbus format. The actual analog outputs are not needed.

Line 8 Asks at what baud rate the comm port is to operate.

Line 9 Starts the setup of the analog outputs, and asks which analog output is to be configured. Analog output #1 in the CPP corresponds to analog output #0 in Dutec and Optomux nomenclature, and analog output #16 in the CPP corresponds to analog output #15 in Dutec and Optomux nomenclature.

Line 10 Asks which channel on the CPP is to be output to this analog conversion output.

Line 11 Asks for the type of DAC output, either voltage or current. A selection of current assumes that it is a 4-20mA signal.

Line 12 Asks if the CPP is to hold the last value it has before a calibration flag is detected on this CPP channel.

Line 13 Asks a similar question regarding the detection of bad data status flags.

Line 14 Enables this analog conversion output.

2.18.1.2 Internal DAC's

In line 5 had the user selected the Internal DAC set up, the question/answer narrative skips lines 6, 7, and 8.

2.18.2 Analog Output Update Rates

Both internal and external analog outputs are updated every fifteen seconds. Even though data is being sent to the analog output module every fifteen seconds, some values, such as computed channels, and data being received over serial ports may be updated at a slower rate. The analog output number corresponds one to one with the channel designations in the analog output module. (Both Dutec and OptoMux number their outputs as 0 - 15. The CPP designates these numbers as 1 - 16). In the above example, data from CPP channel # 03 would be output over analog output #01. The serial strings will be output over comm port #7 in the CPP.

If either the Hold On Cals or Hold On Bad feature is selected, the CPP will hold the last value output prior to detecting either a calibration flag or bad data flags.

2.18.3 Setting External DAC Output Levels

A user can override the channel values and force the DAC outputs to certain levels. This is useful for testing or calibrating the analog output levels. This is done by entering the following;

```
SAO01<Enter>
Set Value (%) = 10<Enter>
>
```

The first input SAO01 selects analog output number one. The CPP then asks for the level to set the output in percent of full scale. In the example the user entered 10%. This could have been input in one string as SAO 01 10<Enter>, with or without the spaces. The CPP will output this level for 3 minutes and then revert back to outputting the channel values. The CPP can be instructed to return to outputting the channel values by inputting a value of 111%, e.g. SAO01 111<Enter>.

The CPP can also be commanded to echo the DAC strings to comm port #4. This feature is toggled by inputting SAOP<Enter>.

2.18.4 Operator Set of Internal DAC outputs

The user can force set the internal DAC analog outputs. This is entered as follows.

```
OBD 01 50<Enter>
DAC# 01 Set To 50%
>
```

The OBD is the command, the 01 is the DAC channel #, and the 50 is percent of full scale. The DAC will hold this setting until the user removes it. The set value is removed by entering,

```
OBD 01<Enter>
DAC# 01 Operator Set Cleared
>
```

2.18.5 External Analog Inputs

The analog input capability is reserved for future applications.

2.18.6 External Digital I/O

The CPP supports both external digital inputs and external digital outputs.

2.18.6.1 External Digital Outputs

In line two above inputting DO selects setting up external digital output bits. The internal output bits are mapped into the external output bit register.

3) Digital Output Setup
4) DigiMux(D), Dutec(U), Opto(O), TSX(T) = D **<Enter>**
5) DO Address = **040 <Enter>**
6) DO Comm Port # 3-08 **<Enter>**
7) DO Baud Rate 1=300, 2=1200, 3=2400, 4=4800, 5=9600, 6=19.2K(6) = 5 **<Enter>**
8) CPP Starting Bit = **31 <Enter>**
9) Number bits = **10 <Enter>**
10) Enable Y/N = **Y<Enter>**

Return to line 2

Lines 4-7 The same as above.

Line 8 Asks for the starting bit in the CPP output register.

Line 9 Asks for the number of bits in the CPP output register to send to the external device. Bits 25 through 34 of the CPP output register will be sent to the external device as bits 1 through 10. In Dutec and Opto this is bits 0 through 9.

2.18.6.2 External Digital Inputs

In line two above inputting DI selects setting up external digital input bits. The CPP internal input bits are mapped into the external input bit register.

3) Digital Input Setup
4) DigiMux(D), Dutec(U), Opto(O), TSX(T) = D **<Enter>**
5) DI Address = **040 <Enter>**
6) DI Comm Port # 3-08 **<Enter>**
7) DI Baud Rate 1=300, 2=1200, 3=2400, 4=4800, 5=9600, 6=19.2K(6) = 5 **<Enter>**
8) CPP Starting Bit = **31 <Enter>**
9) Number bits = **10 <Enter>**
10) Enable Y/N = **Y<Enter>**

Return to line 2

Lines 4-7 The same as above.

Line 8 Asks for the starting bit in the CPP input register.

Line 9 Asks for the number of bits to accept to the external device. Bits 1 through 10 will be accepted from the external device, and stored in the CPP input register as bits 25 through 35. In Dutec and Opto this is bits 0 through 9.

2.18.7 Copy Outputs To Comm Port #4

The CPP can also be commanded to echo the I/O strings to comm port #4. This is a toggle on, toggle off feature.

Toggle analog outputs
SAOP<Enter>
AO On

SAOP<Enter>
AO Off

Toggle analog inputs
SAIP<Enter>
AI On

SAIP<Enter>
AI Off

Toggle digital outputs
SDOP<Enter>
DO On

SDOP<Enter>
DO Off

Toggle digital inputs
SDIP<Enter>
DI On

SDIP<Enter>
DI Off

2.19 Curve Fit - Linearization

The CPP contains firmware that supports a curve fitting function. The method used in the curve fitting firmware is a piece wise linear approximation. This method requires that several values defining the input/output relationship (curve) be entered. Because the maximum number of values that may be entered will not completely define the curve, those values, which lie between the entered values, are approximated from a straight line between these values.

It therefore becomes the responsibility of the user to evaluate the values defining the curve, and select and enter those that best represent the curve. In those cases where there is a large change in output values for small changes in input values, the values chosen for this part of the curve should be closer together. Likewise, in parts of the curve where changes in the output values closely map those of the input values, the points may be spaced farther apart.

The only requirement for entered input values is that they start from the lowest value and proceed to larger values. Curve approximation deserves some consideration. Assume that we have a portion of a large circle (an arc) to approximate. Envision two points on the arc connected with a straight line. The two points on the arc will be exact in their input/output representation, and the maximum error, Δ, will occur at the midpoint of the straight line - where the straight line is farthest from the arc. If the two input points are moved off of the arc by Δ/2, one on one side of the arc and the other on the other side of the arc, then the midpoint of the straight line intersects the arc. This reported value would be totally correct, and the maximum error along any point on this segment approximation is Δ/2. We have improved our representation of this curve by a factor of two. By properly selecting the spacing of the input points based on the shape of the curve very good representations of a curve can be realized in this fashion.

The curve fit feature can be used to solve a number of situations, including;

curve fit or Linearization,
offset signals (4-20mA or 1-5v),
negative going signals,
negative starting signals.

Offset signals such as 4 - 20mA inputs are normally better corrected by scaling the full scale and zero settings of the channel receiving the signal, rather than curve fitting. The curve fit can be used to accept the square root of a signal, but this is better accomplished in the computed channel feature of the CPP.

The full scale and zero values must be set up in the CPP prior to setting up the curve fit channels. If the full scale and zero values of the CPP are changed, then the curve fit numbers must be reentered.

2.19.1 Set up a channel for curve fit

From the Main Menu select the curve fit option. The CPP responds with

Curve Fit Setup
Channel # = **01<Enter>**

Enable Y/N = **Y<Enter>**
Max # Points in curve = 02 **<Enter>**
01In > 0.0 **<Enter>**
01Out > 0.0 **20.0<Enter>**
02In > 0.0 **100.0<Enter>**
02Out > 0.0 **100.0<Enter>**
Channel# = **<Enter>** Ret to Main Menu

The above would remove the offset from a 4-20mA input signal, on channel #01 of the CPP. The decimal point of entered values must match those of the full scale setting of the CPP channel selected or the unit will not accept the value. A particular curve fit can be disabled by answering **N**, or no, when prompted. Entering an escape character results in the CPP reasking the previous question. Entering an E when the CPP is accepting input values, results in the CPP enabling this channel and asking for the next channel.

Although somewhat rare, instrumentation exists that starts from a zero value and increases in a negative direction for increasing readings. There also exists instrumentation the starts at a full scale negative value for a zero reading and increases positively to zero or even plus full scale as the readings increase. The curve fit feature can be used to accommodate both of this type of inputs.

2.19.2 Negative going signal

Max # points in curve 02 **<Enter>**
01In > 0.0 **-100.0<Enter>**
01Out > 0.0 **100.0<Enter>**
02In > 0.0 **<Enter>**
02Out > 0.0 **<Enter>**
Channel # = **<Enter>**

As this signal increases in a negative direction, the signal stored in the CPP channel will increase positively. Note that the least positive input points must be entered first.

2.19.3 Negative starting signal

Max # points in curve = 02 **<Enter>**
01In > 0.0 **-100.0<Enter>**
01Out > 0.0 **<Enter>**
02In > 0.0 **<Enter>**
02Out > 100.0**<Enter>**
Channel # = **<Enter>**

A full scale negative input will result in a CPP value of zero, and a zero value input will be translated into a full scale value.

2.19.4 Thermistor inputs

The negative slope feature can be used to input values straight from a thermistor temperature measuring circuit. Thermistors exhibit a negative slope coefficient. For a standard -50°C to +50°C circuit, the relative voltages (normalized to one volt) are 0.873 for -50° and .313 for +50°. Set the CPP full scale value to 100.0.

Max # points in curve = 02 **<Enter>**
01In > 0.0 **31.3<Enter>**
01Out > 0.0 **50.0<Enter>**
02In > 0.0 **87.3<Enter>**
02Out > 0.0 **-50.0<Enter>**
Channel # = **<Enter>**

This setup will provide a temperature reading from -50°C to +50°C. The readings can be presented in Fahrenheit by setting the 01Out to 122.0°Fand 02Out to -58.0°F.

2.20 Set Up Calibrators

To download configurations to calibration instruments, the calibrator must first be initialized into a CPP channel and have a comm port assigned. From the Main Menu select Set Up Calibrators. The CPP requests the channel number of the desired calibrator. Inputting the channel number selects the unit. From the manufacturer and type of unit initialized into the channel, the CPP branches to the appropriate subroutine to configure the calibrator.

2.20.1 API Model 700

The CPP supports the set up of two API calibrators. As can be seen below, the CPP defaults to a number of standard calibrator configurations. The concentration units can be ppb (parts per billion), ppm (parts per million), ppt (parts per thousand), pct (percent), ugm (micrograms per cubic meter, and mgm (milligrams per cubic meter).

Chan = **1<Enter>**
Calibrator #1 or #2 =1 **<Enter>**
API Model 700

1. Seq ZERO
2. Seq SO2
3. Seq NOx
4. Seq CO
5. Seq O3
6. Seq HC
7. Seq GPT

8. Seq SPC01
9. Seq SPC02
10. Standby
11. Send Concentrations

Selection = **2<Enter>**
Concentration = 450 ppb **400 ppb<Enter>**

1. Seq ZERO
2. Seq SO2
3. Seq NOx
4. Seq CO
5. Seq O3
6. Seq HC
7. Seq GPT
8. Seq SPC01
9. Seq SPC02
10. Standby
11. Send Concentrations

Selection = **11<Enter>**
Send Concentrations Y/N **Y<Enter>**

(Here the CPP displays all commands sent to the calibrator,)
(followed by the response from the calibrator, or a no response message.)

Response From Calibrator
ACK

Chan = **<Enter>**
Return to Main Menu

2.21 Hold Last Value

This feature allows the CPP to hold the last value that was stored prior to a digital output bit being set. This is useful for stack stream switching operations (multiplexing different signals into a common set of instruments). Rather than presenting data that is following another stack and flagging it bad, the last good data value can be held and presented. From the Main Menu select Hold Last Value. The CPP responds with a channel number request, as presented below.

Hold Last Value
Chan # = **01<Enter>**
Bit # 25 **26<Enter>**

Hold Last Value
Chan # **<Enter>**
Return to Main Menu

In the above example, the user selected channel #1. The CPP presented that channel #1 was not updating the channel value any time output bit #25 was set. The user changed the output bit to hold on to bit #26. As long as bit #26 is false, channel #1 is updating its data values. If bit #26 is true (set), then the channel register is not updated and the last data value is held. The Boolean functions can be used to map multiple I/O conditions into one output bit for the hold function. To remove a hold function from a channel, enter a bit number of zero.

2.22 Using Digital Output Bits To Set Analog Output Values

The values output by the internal DAC channels can also be controlled by the digital output bits in the CPP. The sequencers normally control these. The channels to be controlled are set up from the Main Menu as follows. From the Main Menu select the “DO To DAC” option. The CPP responds as follows.

```
Output Bits to DAC
DAC# = 2,<Enter>
OK
DAC#= <Enter>

Return To main menu
```

The above example sets up DAC channel #2 to respond to the digital output bits. Bits 25-36 are used as the twelve bits of binary data to the DAC. Bits 37, 38, and 39 are decoded to select DAC# 1-8 (0_H-7_H, respectively). Bit number 40 being true enables the decoding of a DAC channel address, and subsequent output of the binary string to the DAC. Bit 40 is the most significant bit. The user needs to be familiar with hexadecimal numbers to properly set up and use this feature. A couple of examples are given below.

```
40       36                      25
 1 0 0 1  0 0 0 1  1 0 0 1  1 0 1 0
```

Bit 40 being true enables the DAC address decode. The next three bits address a DAC channel numbered 1-8. The bit setting 001 addresses DAC# 02. A bit setting of 000 addresses DAC# 01. A bit setting of 111 addresses DAC#8.

Bits 25-36 comprise the twelve bit binary data field. A twelve bit DAC has a full scale count of 4096. In the above example, the data field setting $19A_H$ is 410 in decimal, or ten percent of full scale. Any of the 4096 values can be output in this fashion.

2.23 Modbus Set Up

A Modbus interface is an option offered in the CPP. Tech Note TN33 describes the set up and operation of this option.

2.24 Poller Inactivity – Comm Port #1

This feature allows the CPP to set a relay output in the event of inactivity from the polling computer. Selecting this option results in the CPP asking the following.

```
Poller Inactivity
Inactivity Timeout in Minutes = 10<Enter>
Activate Relay # = 40<Enter>
Auto Reset Y/N = Y <Enter>
```

The inactivity range is from 00 to 99 minutes. An input of 00 disables the inactivity feature. An entry between 01 and 99 enables the feature. If the polling goes longer that the inactivity timeout entered, the CPP will set the entered relay.

The auto reset question determines how the bit gets reset once it has been set. If auto reset is selected, the CPP will reset the bit when communications is reestablished. If the auto feature is not selected, then the bit must be reset by and operator using the BR40 command.

3 On Site Port

3.1 Description

Comm port #4 is the local or on site port and is used to set up and monitor all operational features of the CPP. This port is available on the rear panel as a RJ-11 telephone jack, and on the front panel as a DB9 connector. Presented below are features that are installed in all program versions of the CPP. As the applications of the CPP vary so do the desirable features in the monitor port. Any features that are specific to an application are documented in a special applications section.

Although comm port #1 is normally used to interface to the outside world, if desired, comm port #4 could also be connected to a modem or cellular telephone and used to interface to the CPP remotely. The configuration depends entirely on the given application and other outside operational conditions.

3.2 List of Key Commands

All inputs to the monitor port use a carriage return as a delimiter. After the character or characters presented in the following sections are entered, a carriage return is required to initiate the action. Figure 3-1 provides a list of key commands. Operator inputs are in **bold**, and a carriage return is designated as **<Enter>.** A succinct listing of this table can be obtained by inputting **HELP<Enter>.**

Input	Action
A	Print the last 5 power fail or off line times. Off Line From......12:00:00 11-15-95 Until 12:05:04 11-15-95
AC	Clear the power fail log
ALM	Print channels currently in alarm
BRxx	Allows the user to reset bits (xx) set with a BS instruction as presented below. The system response is to print the control settings when the bit is reset.
BSxx	Allows the user to set control output bits (xx), where the xx is the bit number. Once set by the user, this bit will remain set until reset by the user. The system response is to print the control settings when the bit is set.
CFGDLxx	Download configuration from CFM, xx defaults to 01. If CFM is not installed, downloads from EEPROM.

CFGLOG Print a log of configurations in the CFM

CFGULE Upload configuration to on board EEPROM

CFGULxx Upload configuration to CFM, xx defaults to 01

CFM? List operator CFM commands

CsU Allows clearing of set up parameters

D43 Allows a pass thru mode between comm ports # 4 and #3. This feature lets an on site operator communicate with the instrumentation. Reference individual manufacturers in appendix E for commands.

E Print a log of communications errors.

The error log printout is presented below in section 3.3.

EC Clear error log.
11:04:34 11-15-95 Error Log Cleared

ERASEp Erase preliminary arrays

ERASEi Erase interim arrays

ERASEf Erase final arrays

ERASEcE Erase RamPack. Not available to user.

ERASErZ Format RamPack

I Initialization - Brings up the Main system initialization menu. Refer to section 2.0 which details the system initialization.

L Print a configuration listing. Examples of the listing associated with each function is presented in that section.

LCD Toggles demo data for bar chart and trend charts

Lx Print the last average for 20 channels, where x = P = Preliminary, = I = Interim = F = Final.

LxA Print all 40 channels

Mb aaaa cc Prints RAM contents, b = bank, aaaa = address, cc = #16 byte lines

OPSET Print any operator forced settings.

OPRx Print stored operator messages, x = 1-8, or A for all

OPWx Allows writing message #x, or clearing with **CLR<Enter>**

PAC Commands the CPP to print the calibrations results after any calibration.. This continues until stopped with another PAC input. The power up default is for this to be set true.

PAD Commands the CPP to print the I/O settings on every I/O change. This continues until stopped with another PAD input.

PARxx This input instructs the CPP to print each instrument response from channel number xx. This depends on the CPP getting a response. The CPP prints 32 bytes, regardless of the length of the input string. If the CPP accepts the command the unit prints "on." To disable this feature, input only the PAR followed by a carriage return, *e.g.*, **PAR<Enter>**. The CPP will respond with "off."

PCx Print the calibration results from day x. (1=today, 2=yesterday, •••)

PD Printout the present digital I/O settings.

```
09:45:12        ( 1 - 8)  ( 9 - 16)  (17 - 24)  (25 - 32)  (33 - 40)
Digital In      00000000 00000000 00000000 00000000 00000000
Digital Out     00000000 00000000 00000000 00000000 00000000
```

PF Print the current final averages, from the last time frame.
Time frame is 00:00 for sixty minute averages.
Time frame is 00:00 or 12:00 for thirty minute averages.
Time frame is 00:00, 06:00, 12:00, or 18:00 for fifteen minute averages.

PFx Print the previous time frames final averages, where x is limited to 5.

PI Print the last current interim averages, from the hour until present time.

Pix Print the previous interim averages, where x is limited to 4.

PM Print the current minute averages. Will print the last ten.
PMxxx Print the last xxx minute averages, where xxx is limited to 180.

An example data printout is presented below in section 5.0.

PV Print instantaneous data values.

PVA Print instantaneous A/D values

PVAR Print instantaneous A/D values every ten seconds. How Stop?

PVS Allows the printout format to be toggled between a scientific format and an engineering format. This affects all printouts, but not the inputting of set up values.

STATUS List status flags

SEQ Allows the user to start and stop the sequencers. The CPP asks a series of questions as presented in section 3.4 below.

T Print the present time and date.

TIxx Allows the user to send the time and date to a particular instrument. The xx are the channel number that the instrument is connected to. If an instrument that supports time is connected to the channel selected, the CPP prints Set Time Queued Up. If an instrument that does not support time is connected to the channel selected, the CPP prints a corresponding message. Likewise, if an instrument is not connected to the channel selected. At 41 seconds after the minute the CPP sends all queued up messages and logs the event.

TF Select the data presentation format. This allows the user to set up the month, day, year printout.
MM/DD/YY = 0, DD/MM/YY = 1 **0<Enter>**

TS Set the CPP Real Time Clock - Time & Date. The CPP first prints the present time and waits for an operator input. Inputs are as indicated below. Hours, minutes, seconds, months, days and years may or may not be separated with a character. Inputting a carriage return at the start will leave a value unchanged. The CPP logs a time/date change.

14:12:15 **14:14:00**
11-15-95 **11-16-95**
Time Change From...14:12:15 11-15-95 To 14:14:05 11-16-95

U Allows the user to up down channels.

ZS Allows expected zero and span values to be entered.

Figure 3-1 Comm Port #4 Key Commands

3.3 Example Error Log

The CPP maintains a running error log that lets the user determine if communications are occurring properly. After connecting instrumentation clear the error log by inputting an **EC**. After some period of time (two to three minutes) check the error log (**E**) for errors. In proper operation, there should be no errors. An example error log printout is presented below, in Figure 3-2.

Note: The error log will not flag a no response error from instrumentation that broadcasts. If a response is being received, then all other types of errors will be detected and logged. The minute averages (**PM**) should be printed out to ensure proper data collection.

```
(1) Chan        Comm    Name  No Resp    Resp Err  Unk Resp  Mdl #   ErrComm Err
     01       00   Name#1     00         00          00             00              00
     02       01   Name#2     00         00          00             00              00
     03       02   Name#3     03*        00          00             00              00

(2) Number Incomplete Polls 00

(3)                 Ext I/O  00
Type     CP#     NoR    R Err    Err Code
         *** None Set Up ***

(4)      CFM - 08 Megabyte

         AA  55   00 01 00 01    00 01 00 01   00 00 50 00
```

Figure 3-2 Example Error Log Printout

3.3.1 Communications errors

The error log is kept independently for each channel collecting data over a serial port. The meanings of each error type are given below. A non zero location is flagged with an asterisk making it easy to spot.

No Resp	- No response from the instrument to an interrogation
Resp Err	- CPP received response but detected an error in the data field
Unk Resp	- CPP received a response but could not decode the response
Mdl # Err	- CPP received a response but not from the type of instrument it is configured to interrogate on this channel
Comm Err	- Conflict in comm port's configuration set up.

Figure 3-3 Communication Error Code Definitions

In the example presented above, channel number 3 is not getting a response from the instrument. In normal operation there should not be any errors accumulating. If there are, then some problem exists. The error log accumulates each error type up to 99 and then stops. The error log is cleared with an **EC** input.

The three most common causes of errors are, the cable does or does not need a null modem, or the cable is connected to the wrong comm port on the CPP, or the baud rate set in the CPP does not match that set in the instrument. With Thermo instrumentation, another source of potential error is that the instrument must be set for REMOTE operation (which enables the RS-232 port) and the instrument ID set up in the instrument must match that set up in the CPP.

3.3.2 Incomplete Polls

The incomplete poll indicator should always be zero. A non zero indicator means that the CPP could not poll all instruments in a one minute time frame. Rearranging the comm port assignments will correct any such problems.

3.3.3 External I/O

The Ext I/O status indicates that an error was detected when transferring data to an external I/O device. Since there are independent units communicating over a serial line, there could occasionally be an error, but it should be very infrequent. An incrementing error condition means that something is wrong and needs to be corrected.

3.3.4 Compact Flash

Each time the error log is printed the CPP checks and reports the status of the CFM. If a CFM is not installed, the CPP will provide such a message. It will also report any errors encountered when trying to write data to the CFM. A CFM must be formatted before it can be used by the CPP. The formatting process partitions the CFM into segments, and calculates and stores the next writeable memory address and the last available address in this size CFM.

Each time the CPP writes data to the CFM, these parameters are checked and any error condition logged along with the time of the error detection. This error condition will not be overwritten and is cleared with the EC command. The E response prints the contents of these error registers if an error has been detected. The CPP configuration can be stored in a reserved section of in the CFM. A CFM can be formatted as a configuration CFM to store the configurations of many CPP’s. If a configuration CFM is installed the CPP will print such a message. The CPP will not record data into a configuration CFM.

3.4 Sequencers On Line

This input allows the user to exercise control over the sequencers. The command can also be entered as SEQxx, where xx is the desired sequence number. The CPP asks the Following:

SEQ<Enter>
Seq # = **01**

The CPP asks for the sequencer number. A carriage return at this point exits the routine. If a sequence number is input, the CPP prints the sequence label and presents the following menu:

(Sequence Label)
Start Seq = 1
Stop Seq = 2
Single Pass = 3
Choice = **3<Enter>**

A Carriage return at this point exits back to asking for a sequence number. Other selections are as presented above.

Starting a sequence results in the sequence starting at the present time, and transfers the rep rate into the counter. From this time forward, sequences set up as automatically starting run from the entered rep rate. Should a power fail or time change occur, the system uses the entered start time and the restart criteria to restart the sequence. If the sequence is not set up as an automatic start sequence, then after all T times have been completed, the sequence will not restart, unless the conditions initiating a manual start are met.

Selecting a single pass operation does not change the sequence start time, but immediately starts the selected sequence manually for a single pass through all events. After the pass is completed, the sequence stops and is now ready to start at the originally set start time. This is commonly used for performing manual calibrations

If a one pass operation is in effect, should the sequence start time come up, it will be ignored. After the one pass operation has been completed, at the next start time match, the sequence will restart.

The stop command stops a sequence and resets all control output bits associated with that sequence. When the start time again matches, the sequence restarts.

3.5 Operator Message

The CPP supports the storage of 8 operator or computer messages. Each message is limited to 80 characters.

To record a message, enter OPW x, where x is the desired message number 1-8. If a message is already stored in the message bin number, the CPP prints the stored message and awaits an operator response. Entering a carriage return or escape exits without changing the existing message. Entering CLR<Enter> results in this message bin being cleared. Entering anything else erases the existing message and starts recording the new entries. A message is terminated with a carriage return or escape. The time and date is automatically stored with each message.

Messages can be printed by entering OPR x, where x is 1-8 or an A which instructs the CPP to print all stored messages.

3.6 Definition of Binary Status Codes

A definition of the binary status codes is presented in Figure 3-4.

First Status Byte				
Status Code	Bit	Meaning		
M	7	- 00 = Missing	Bits 7 & 6 are decoded	
	6	- 10 = Good	as given here	
B		- 01 = Bad		
o		- 11 = Other		
R	5	- 1 = Instrument No response		
L	4	- 0 = Instrument On Line, 1 = Off Line		
P	3	- 1 = Instrument Parameter Alarm		
A	2	- 1 = Data High Alarm		
a	1	- 1 = Data Low Alarm		
C	0	- 1 = In Cal, 0 = In Sample		
Second Status Byte				
F	7	- 1 = Power Fail		
>	6	- 1 = Not All Samples , But More Than Required		
<	5	- 1 = Not All Samples, And Less Than Required		
H	4	- 1 = Value being held		
D	3	- 1 = Downed By Operator		
	2	= Encoded	0 = U - Nothing	4 = S - Smp Delay
	1	= “	1 = Z - In Initialization	5 = Undefined
	0	= “	2 = Y - A/D Calibration	6 = Undefined
			3 = V - Validity	7 = * Cal Alarm

Figure 3-4 Binary Status Code Definitions

The first status byte contains status information that is normally provided by most instrumentation. The status conditions in byte number one can also be generated inside the CPP. For example, the high/low alarms may not be set up in the instrument, but are set up in the CPP. In this case these status bits would be provided by the CPP. Obviously both could be set up. If the status condition is set by one it will not be reset by the other.

A good data value does not have a modifier printed. The status modifiers M and B indicate Missing and Bad, respectively. If any other status bits are set, then a secondary flag indicator may be also be printed. In highest to lowest priority these are printed as; F, Y, Z, R, L, C, D, H, I, A, a, >, <.

Bits 2, 1, an 0 of the second status byte are available to be encoded to provide additional data status. If they are used they are defined in a special applications section. These bits are defined in the standard CPP in several categories.

3.7 CPP Features

The features that are resident in the CPP firmware are presented at the end of the L key command. Only acronyms are presented for the features. Figure 3-5 defines the acronyms.

CHN	xx	Number of channels
ALR	xx	Number of alarm channels
ADC	xx	Max number of analog to digital inputs
SEQ	xx	Number of sequences
MET	xx	Number of sigma theta's supported
CAL	xx	Number of calibration channels supported
SPN	xx	Number of spans/cal channel supported
BLN	xx	Number of Boolean functions supported
BTM	xx	Number of IF-THEN terms/Boolean function
I/O	xx	Number of groups of eight of digital I/O bits
SER	xx	Number of optional serial ports installed
PLM	xx	Number of hours of preliminary average storage
INT	xx	Number of hours of interim average storage
FIN	xx	Number of periods of final average storage
SMD	xx	Number of sample delays supported
DSM		DSM protocol over comm port #1
CMP		Computed channels supported
OPI		Operator interface to instrumentation supported

Figure 3-5 Definition of CPP Features In CPP

4 Front Panel Controls and Indicators

4.1 General

Although a notebook is the desired form of interfacing with the CPP, the CPP does support an LCD display and an optical switch, which allows the user a great degree of interfacing to the CPP. The interfacing is primarily for requesting data presentations, inputting normal system operating parameters, such as zero/span values, starting and stopping calibration cycles, and setting and resetting the digital I/O bits, which allows the integrity of the system operation to be verified. The CPP will also present bar charts for selected data. The features available from the front panel include;

- Displaying A/D voltages,
- Starting/stopping sequencers (calibrations),
- Displaying channel values as Scientific or Engineering units
- Setting/resetting digital input and output bits,
- Displaying calibration results,
- Inputting zero/span expected values,
- Display channel averages, selectable, "Preliminary", "Interim", or "Final"
- Setting time and date,
- Display and clear the error log. Power fail, Comm error, A/D error, Alarms, RamPack
- Setting comm port Baud rates,
- View system configuration, Password, Screens, Bar Chart, Unit ID, Firmware Version,
- Comm Port Settings
- Putting up bar displays,
- System Review, display the system configuration channel by channel

The LCD is a 16 row by 40 column display measuring 2½" x 4¾ (65 x 120mm).

4.2 Communications Bar

The top line of the display is reserved and used to show activity on the communications ports. From the left hand edge of the LCD, comm ports 1 through 4 are presented and then comm ports 3-01 through 3-08. Given the number of spaces in the Comm Bar, comm ports 3-01 through 3-08 are numbered 1 through 8, but they start about 1/3 of the way across the Comm Bar. Activity is presented as 4rt, where the comm port is number 4 and it is both receiving and transmitting. If it was just receiving, it would display as 4r and if just transmitting as 4 t.

The top line of the LCD is also used to apprise the user of some potential problem that the CPP has detected. Under such conditions, the CPP displays "E" in the far right hand corner of the top line. Any detected condition can be determined by calling up and displaying the error log. This can be done from the LCD on the front panel or from comm ports 1 or 4. Clearing the error log clears the “E” condition.

The unit will display an “A” in the far right hand corner of the top line if it has any existing alarm condition. The unit will display an “@” in the far right hand corner if the unit has entered the “Playback Mode” when emulating a DSM.

4.3 Time Bar

The bottom line of the LCD display is used to continuously display the time, Julian date and the calendar date.

4.4 Authorization Code or Password

The CPP front panel supports a numbered password feature designated as an authorization code. If a code is set up then the user must enter the code before any active commands can be invoked from the front panel. Viewing data, calibration results and other system settings can still be accomplished, however, the display format can not be changed nor can commands be sent to the CPP. The authorization code feature is covered in section 2.19, System Configuration.

The code is useful to stop well meaning, but errant fingers from upsetting the monitoring configuration, as well as malicious fingers.

4.5 Power Up checks

At power up, the CPP conducts a series of internal checks. This is displayed on the LCD and all front panel controls are disabled until the power up checks are completed. Items checked are:

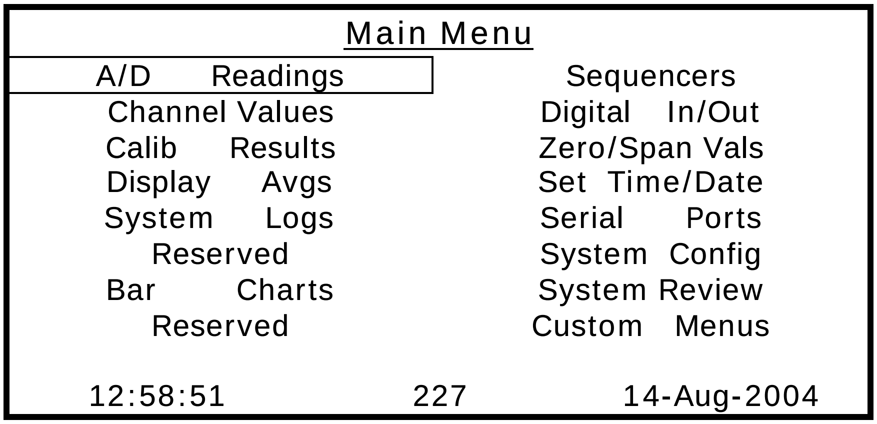
ROM
RAM
Parity
CFM

4.6 Backlight

The on/off control of the backlight can also be set up by the user. The system default is for the backlight to automatically turn off after one hour of non front panel activity. The backlight has a half life intensity of 10,000 hours. Although noticeable in a side by side comparison, a backlight operating at ½ intensity is very useable and does not present an objectionable presentation. The operating lifetime of the backlight can be extended by allowing it to automatically turn off during times of non activity. In conditions where the backlight needs to be on at all times, the user can select this option by selecting System Configuration from the front panel display menu.

4.7 Menus

All display presentations have an Exit option that returns the unit to the Main Menu. Turn the optical switch until this selection is highlighted. Press the rotary switch to activate the built in pushbutton and return to the Main Menu. Following power up the CPP presents the Main Menu as given below.



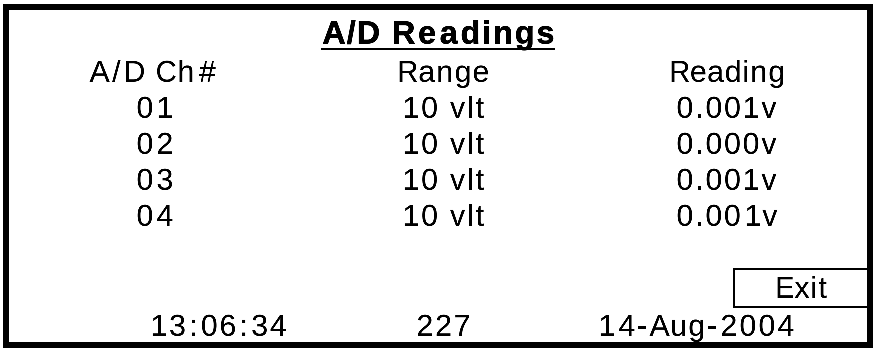
```
                    Main Menu
   A/D      Readings              Sequencers
   Channel Values                 Digital   In/Out
   Calib     Results              Zero/Span Vals
   Display     Avgs               Set  Time/Date
   System     Logs                Serial     Ports
      Reserved                    System  Config
   Bar       Charts               System Review
      Reserved                    Custom   Menus

   12:58:51            227           14-Aug-2004
```

Figure 4-1 CPP Main Menu

The first line (A/D Readings) will be highlighted (enclosed in a box). Turn the rotary switch clockwise to move the box down the menu and counterclockwise to move the box up the menu. Pressing the pushbutton selects the highlighted menu item. From the Main Menu, this normally opens a subsequent menu for selecting a particular operation. In some cases, such as the bar & trend presentations, the last configuration set up is presented. In some cases, such as the last data values, no further action is required as all values are brought up and displayed. The sub menus for each major category are presented below.

4.8 A/D Readings

Depressing the pushbutton with this category highlighted results in the CPP presenting the instantaneous reading from the analog to digital converter for the first four channels.



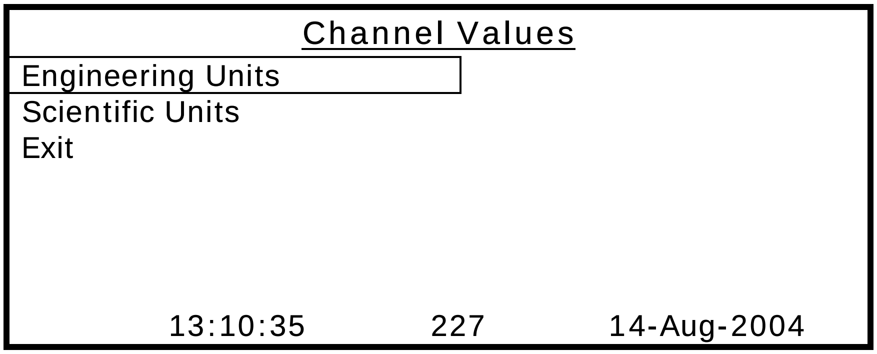
```
                  A/D Readings
   A/D Ch#           Range            Reading
     01             10 vlt            0.001v
     02             10 vlt            0.000v
     03             10 vlt            0.001v
     04             10 vlt            0.001v

                                           Exit
      13:06:34        227        14-Aug-2004
```

Figure 4-2 A/D Readings Screen

Turning the switch clockwise scrolls the channels presented in increasing A/D channel order. This may or may not correspond to channel numbers in the CPP depending on how the A/D Channels were assigned to CPP channels in the system initialization. Turning the switch counter clockwise scrolls the channels presented in decreasing order. Depressing the pushbutton returns to the Main Menu.

4.9 Channel Values

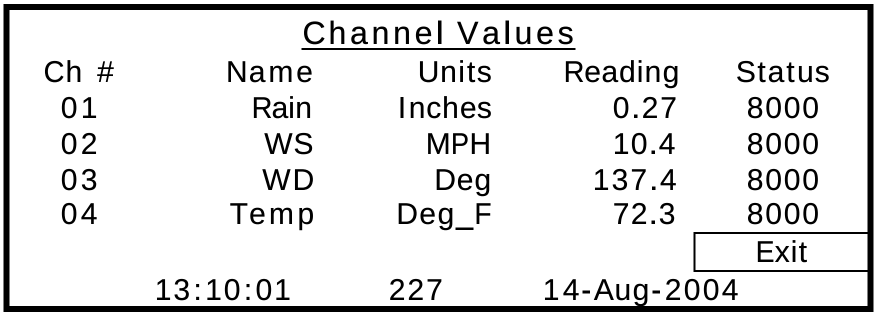
The channel values presentation is similar to the A/D Readings presented above except that data presented is organized by CPP channel number. The CPP also uses the initialized full scale and zero value to provide an instantaneous readout in the parameter units. Depressing the pushbutton results in the CPP asking the following question.



```
                    Channel Values
Engineering Units
Scientific Units
Exit



          13:10:35          227          14-Aug-2004
```

Figure 4-3 Channel Values Menu

Selecting the engineering units presentation results in the CPP displaying the first four channels data and status as presented below.



```
                    Channel Values
  Ch #        Name       Units      Reading     Status
   01         Rain      Inches         0.27       8000
   02           WS         MPH         10.4       8000
   03           WD         Deg        137.4       8000
   04         Temp       Deg_F         72.3       8000
                                                  Exit
         13:10:01       227       14-Aug-2004
```

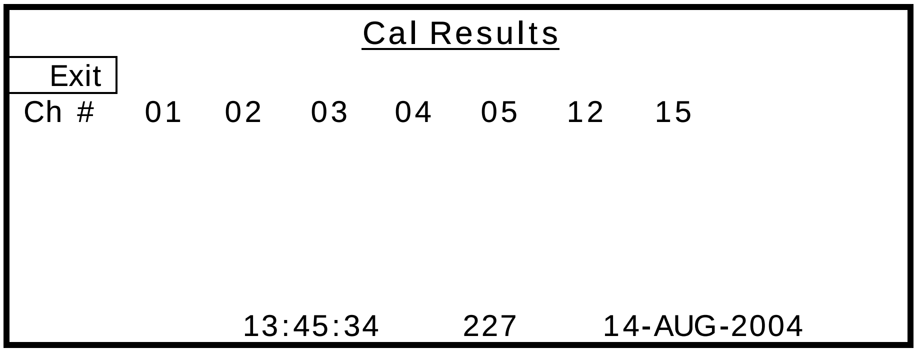
Figure 4-4 Channel Values in Engineering Units

Turning the switch clockwise scrolls the channel selection in increasing numbers, and turning the switch counterclockwise scrolls the channel selection in decreasing numbers. Depressing the pushbutton returns to the Main Menu.

The CPP captures and stores 16 independent status conditions for each data point. Since a data point could have more than one status condition existing at any time, in this presentation the status word is presented in hexadecimal so that any condition can be determined. The meaning of the codes is presented in this section under 2.2.1.

4.10 Calibration Results

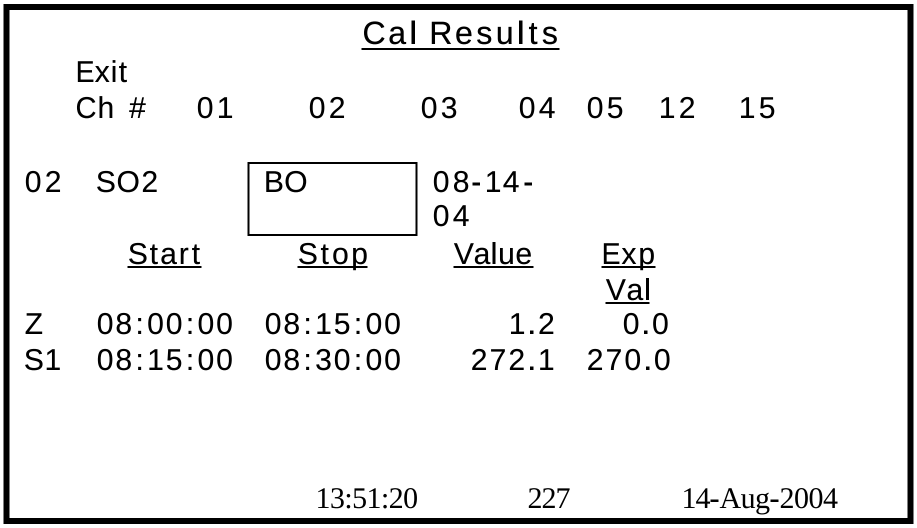
The latest calibration results can be displayed on the front panel display. Selecting this option brings up the following sub menu, which presents the channels that have been set up as calibration channels.



```
                 Cal Results
Exit
Ch #   01   02   03   04   05   12   15




          13:45:34     227     14-AUG-2004
```

Figure 4-5 Channel Calibration Result Screen

Turn the switch until the desired channel is highlighted and press the pushbutton. The CPP presents the calibration results as presented below. If no channels have been set up as calibration channels, the CPP will present such a message.



```
                  Cal Results
    Exit
    Ch #   01     02     03    04  05  12  15

02  SO2         BO        08-14-
                          04
       Start     Stop     Value    Exp
                                   Val
Z   08:00:00  08:15:00      1.2     0.0
S1  08:15:00  08:30:00    272.1   270.0



           13:51:20     227     14-Aug-2004
```

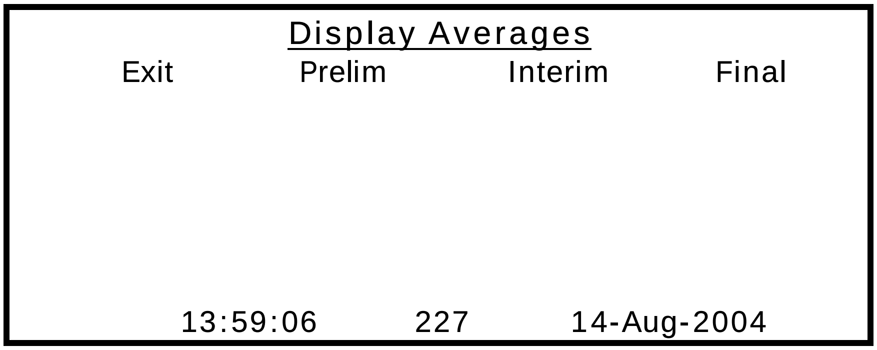
Figure 4-6 Channel Calibration Results Display

If the CPP is conducting more than one span, then the other span results would follow the S1 span presented above. The B0 will be highlighted. Turning the switch clockwise will result in the CPP going back and presenting the previous calibration results. The B0 will change to B1, B2, and so forth. The date indicates which day's calibration results are being displayed. In a like manner, turning the switch counterclockwise presents more recent calibration results. Pressing the pushbutton returns to the channel selection mode, allowing other channels to be viewed.

When a channel is selected, the CPP searches the stored calibration table looking for the occurrence of a zero cycle. When one is found, the CPP presents all zero/spans that are stored for this channel. The date presented as the date of this calibration is the date associated with the start of the zero cycle. In some instances the span cycles are conducted first and then the zero cycle. The CPP will not present any results until the zero cycle is completed.

4.11 Display Averages

The averages formed by the CPP can be displayed on the front panel. Selecting this option brings up the sub menu presented below. From this menu the user can select the preliminary, the interim, or the final averages for display.

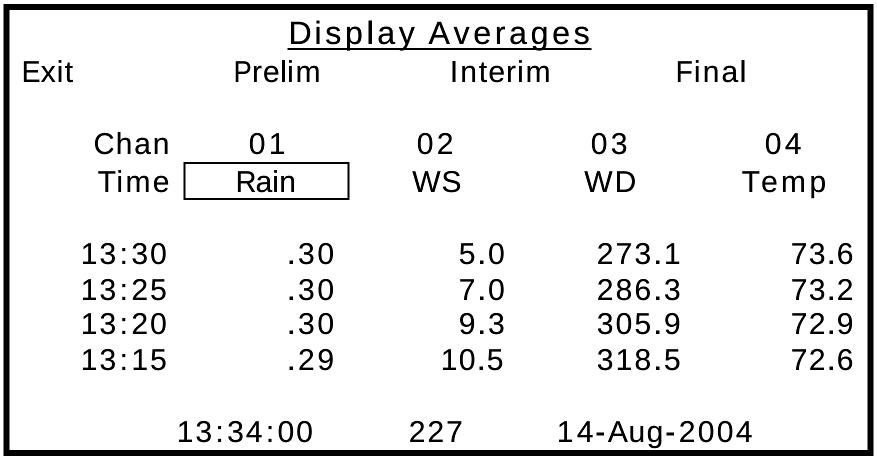


```
                    Display Averages
        Exit        Prelim        Interim        Final




           13:59:06        227        14-Aug-2004
```

Figure 4-7 Channel Display Averages Screen

Select the desired average to be displayed and press the pushbutton. The CPP brings up the last 4 values for the selected average for channels 1-4. Older averages can be viewed by turning the switch clockwise. Likewise, more recently formed averages can be viewed by turning the switch counterclockwise.

The channels being displayed can be scrolled by holding down the pushbutton and turning the switch. Data is displayed as presented below.



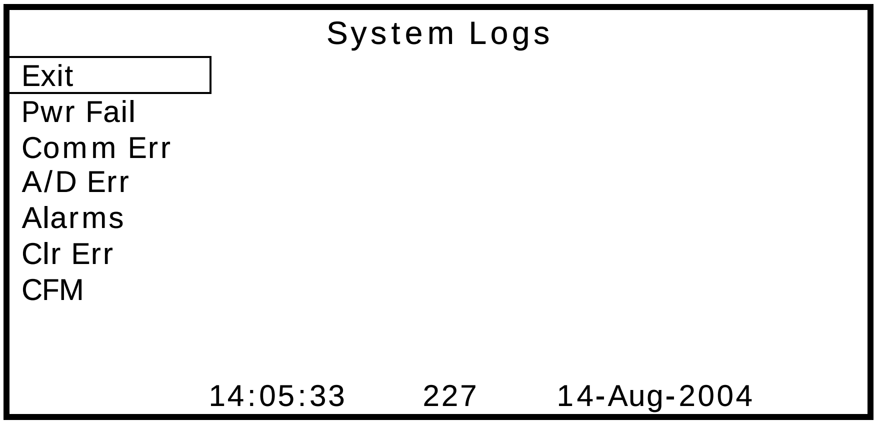
```
                    Display Averages
Exit            Prelim          Interim          Final

     Chan       01          02          03          04
     Time     [ Rain ]      WS          WD          Temp

    13:30          .30         5.0       273.1        73.6
    13:25          .30         7.0       286.3        73.2
    13:20          .30         9.3       305.9        72.9
    13:15          .29        10.5       318.5        72.6

           13:34:00        227        14-Aug-2004
```

Figure 4-8 Channel Display Averages Display

This routine is exited by turning the switch counterclockwise with the pushbutton pushed until the Exit is highlighted and pressing the pushbutton.

4.12 System Logs

Several logs providing information on system operation can be called up and reviewed on the front panel. Selecting this option brings up the following sub menu.



```
                    System Logs
Exit
Pwr Fail
Comm Err
A/D Err
Alarms
Clr Err
CFM


          14:05:33     227     14-Aug-2004
```

Figure 4-9 System Logs Screen

Selecting the power fail option brings up the power fail log. The type of log entry is identified by a two character code, where OL = Off Line, PF = Power Fail, TC = Time Change, BO = Brown Out, and DS = Time Set To DSM Time.

Selecting the Comm Err log presents any errors that were detected on comm ports used to collect serial data from instruments. To provide a compact presentation, the errors are presented as hyphenated and abbreviated mnemonics. The definitions are Cp (Comm Port), No-R (No Response), R-Err (Response Error), Un-R (Unknown Response), M-Er (Model Number Error), and C-Er (Comm Port Error). A comm port error is an improper set up condition detected by the CPP. To aid in detection, any non-zero indication is flagged with an asterisk.

This screen also presents a completed poll indication. This should always be zero. A non-zero means that the CPP could not poll all of the instruments connected in a one minute time frame. Rearranging comm port assignments will normally correct this problem.

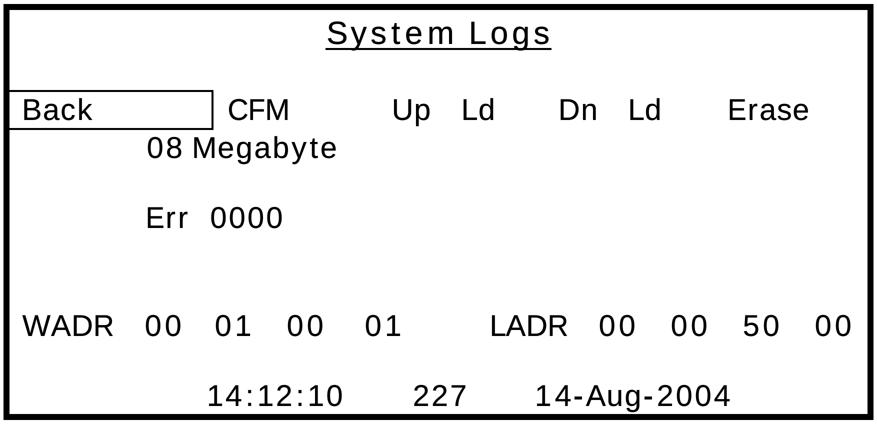
The Ext I/O status indicates that an error was detected when transferring data to an external I/O device. Since these are independent units communicating over a serial line, there could occasionally be an error, but it should be very infrequent. An incrementing error condition means that something is wrong and needs to be corrected.

The A/D Err is placed in the table for future considerations.

The Alarm log brings up any channels that are in an alarm condition.

The Clr Err clears the error log registers.

Selecting the CFM option brings up information regarding the CFM. The CPP checks and reports the status of the CFM.



```
                    System Logs

Back          CFM        Up  Ld     Dn  Ld     Erase
         08 Megabyte

         Err  0000


WADR  00  01  00  01      LADR  00  00  50  00

              14:12:10     227     14-Aug-2004
```

Figure 4-10 RamPack Status Screen

If a CFM is not installed, the CPP will present such a message. If the CFM internal battery is weak or bad, the CPP will present such information. If any errors were detected while writing to the CFM, the Err counter would not be zero. If a CFM that has been formatted as a configuration CFM is installed, the CPP will detect and report this condition. The CPP will not record data on a configuration CFM. Any error indications are cleared with the Clr Err command. The bottom line is CFM internal address registers, and are of importance only to H2NS personnel.

The CFM must be formatted before the CPP can record data. The formatting segments the CFM, and calculates and stores the next writeable address and the last writeable address. If an unformatted CFM is installed, the CPP will print such a message.

Turning the switch highlights the Up Load/Down Load/Erase functions. Pressing the pushbutton invokes the highlighted function. For up load and down load, the CPP presents any configurations stored in the CFM and pauses for a continue response from the operator. If a configuration CFM is installed, the desired stored configuration is selected with the rotary switch. Pressing the pushbutton invokes the command. If a password has been set up then the code must be entered before this command can be invoked.

Note: It is an infrequent occurrence for the need to erase a CFM from the front panel. Therefore, the CPP is delivered with the erase function from the front panel disabled. To enable this feature please contact H2NS.

The Err $0_1 0_2 0_3 0_4$ indicates the following;

0_1 = A write error was detected
0_2 = The RamPack is unformatted
0_3 = Write address bad
0_4 = Last address bad

4.13 Bar Charts

Selecting the Bar Charts icon brings up the Bar Chart #1. The CPP supports two independent bar charts. These are set up in the System Configuration section. The bar charts can be presented without entering the password.

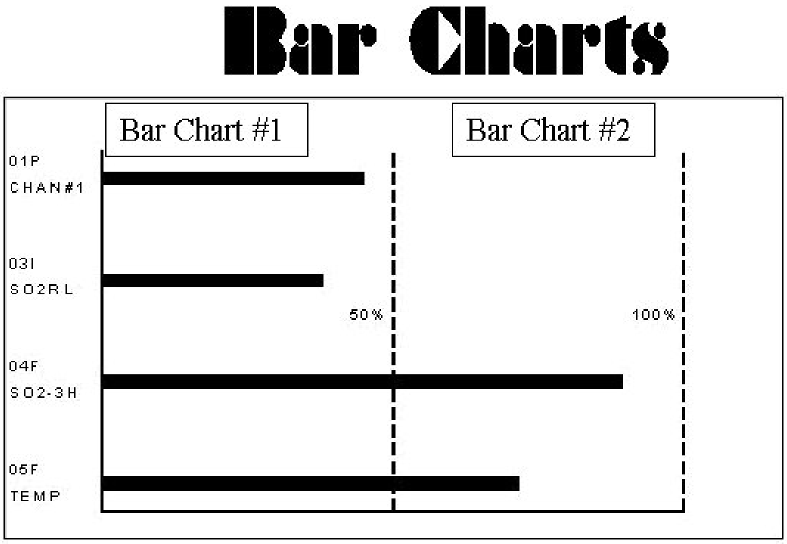


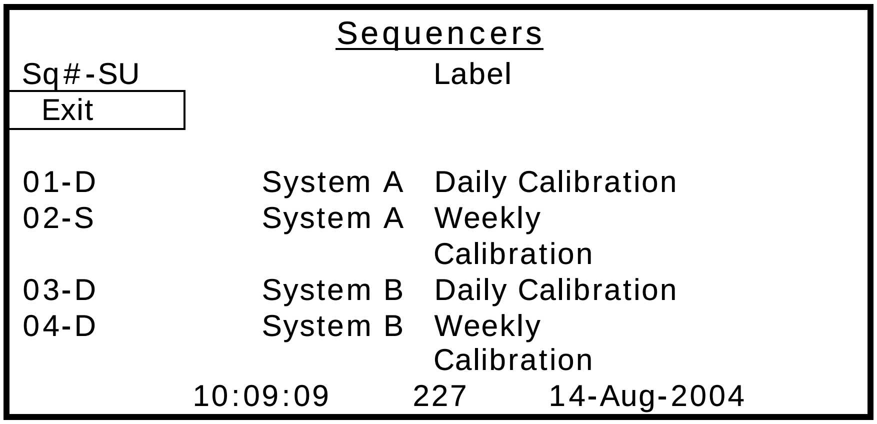
Figure 4-11 Bar Chart Screen

Use the switch to select the desired bar chart presentation. Each bar chart can have four channels presented, with the input to each channel presentation being the instantaneous, preliminary, interim, or final average data points. The channels in a bar chart can be different or the same channel with different averaging intervals.

Four bar charts are presented in the space above. To the left of the screen, the 01P indicates CPP channel number 01, and the bar chart is presenting the preliminary average. The following terms are used; P - preliminary, A - instantaneous, I - interim, and F - final. The actual value is also printed to the right of the display. This routine is exited by pressing the pushbutton.

4.14 Sequencers

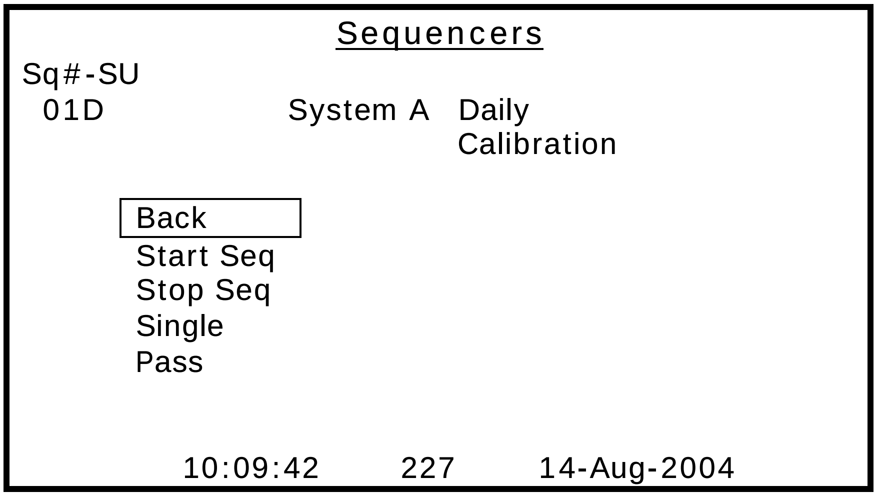
The sequencers can be started and stopped from the front panel. Making this selection brings up a subsequent menu that allows any sequence to be selected. This presentation also displays any sequencer label that has been entered. The Exit will be highlighted which if selected returns to the Main Menu.



```
                    Sequencers
Sq#-SU                    Label
 Exit

01-D          System A   Daily Calibration
02-S          System A   Weekly
                         Calibration
03-D          System B   Daily Calibration
04-D          System B   Weekly
                         Calibration
          10:09:09     227     14-Aug-2004
```

Figure 4-12 First Sequencer Screen

Turning the switch clockwise displays higher numbered sequencers. Turning the switch counterclockwise moves the numbers in the smaller direction. Depressing the pushbutton selects a particular sequence and the next menu is brought up.



```
                    Sequencers
Sq#-SU
 01D            System A   Daily
                           Calibration


      Back
      Start Seq
      Stop Seq
      Single
      Pass


         10:09:42     227     14-Aug-2004
```

Figure 4-13 Sequence Selector Screen

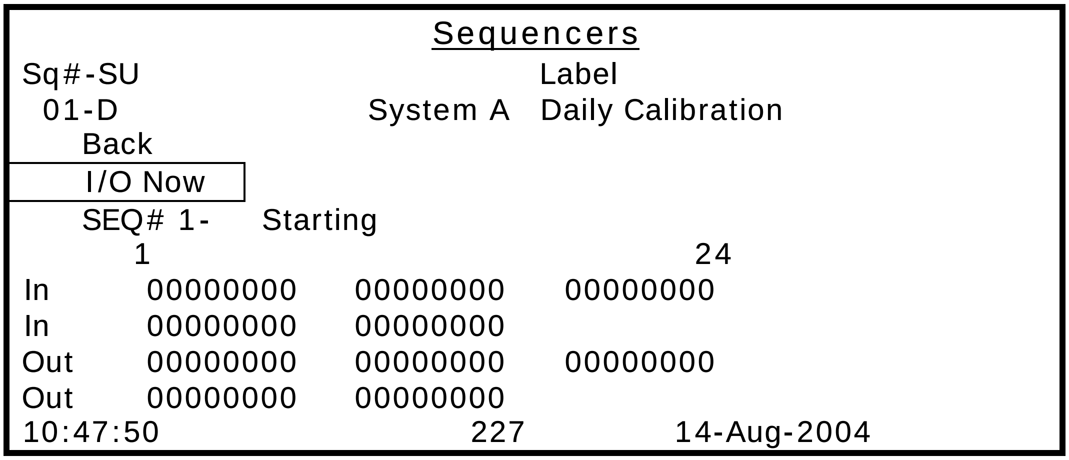
Selecting the Back option returns to the previous menu. If a sequence is active, *i.e.*, it is actively controlling the digital I/O bits, then the CPP will not accept a command to start the sequence or to start a single pass. It will provide a message that the sequence is active. If a sequence has external enabling criteria set up, then the external setting must meet the set up criteria or the CPP will not start the sequence. It will provide a message that the sequence is disabled. If a command is issued to a sequence that has not been initialized then the CPP ignores the command and provides a message.

Starting a sequence results in the sequence starting at the present time, and transfers the rep rate into the counter. From this time forward, sequences set up as automatically starting run from the entered rep rate. Should a power fail or time change occur, the system uses the entered start time and the restart criteria to restart the sequence. If the sequence is not set up as an automatic start sequence, then after all T times have been completed, the sequence will not restart, unless the conditions initiating a manual start are met.

Stopping a sequence results in the sequence being stopped at the present time. If the sequence is set up to automatically start, the sequence will restart at the entered start time. A sequence set up to manually start will restart when starting conditions are met.

A single pass starts a single pass through all T times. The original rep rate in the counter is not changed. Therefore this sequence continues to cycle at the rep rate related to the entered start time.

The CPP responds to all commands with a message in the following format.



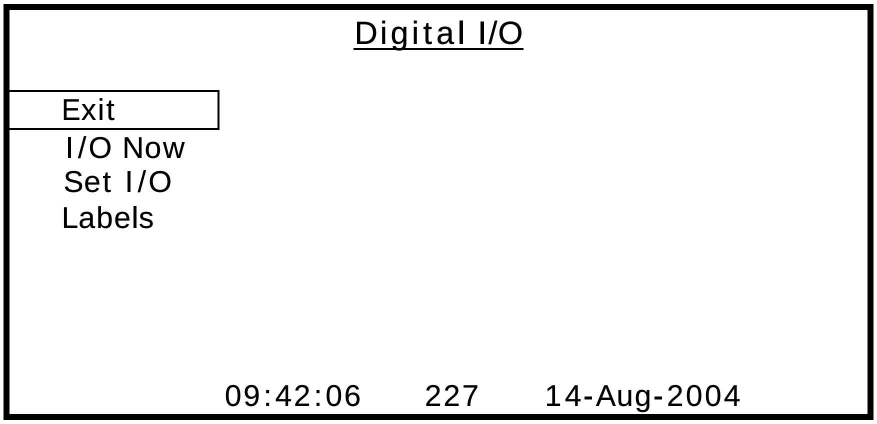
```
                        Sequencers
Sq#-SU                            Label
 01-D             System A   Daily Calibration
   Back
   I/O Now
   SEQ# 1-    Starting
      1                              24
In       00000000    00000000    00000000
In       00000000    00000000
Out      00000000    00000000    00000000
Out      00000000    00000000
10:47:50                 227         14-Aug-2004
```

Figure 4-14 Sequence Command Response Screen

Although shown above, the I/O settings are not presented until they are requested with the I/O Now selection.

4.15 Digital I/O

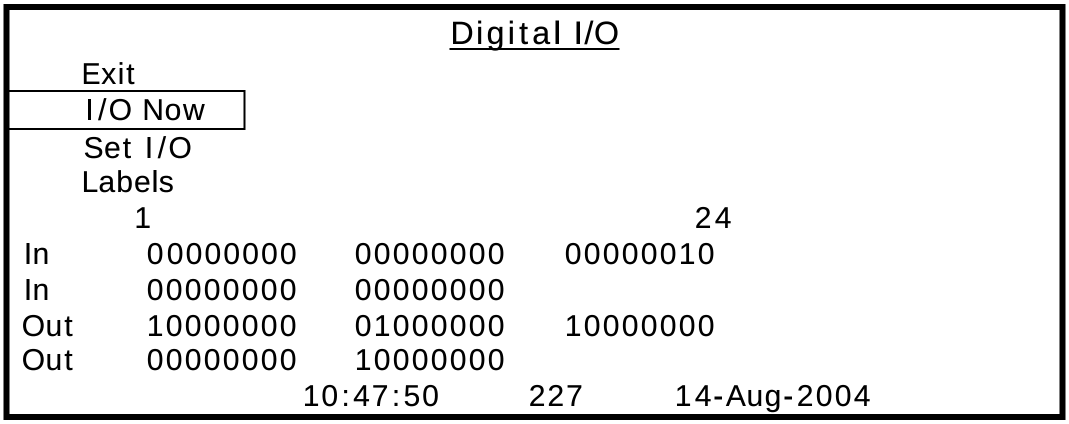
The Digital I/O selection allows the existing setting of the digital input bits and the digital output bits. The labels assigned to these bits can also be reviewed. Making this selection brings up the following sub menu.



```
                   Digital I/O

   Exit
   I/O Now
   Set I/O
   Labels




           09:42:06     227     14-Aug-2004
```

Figure 4-15 Digital I/O Screen

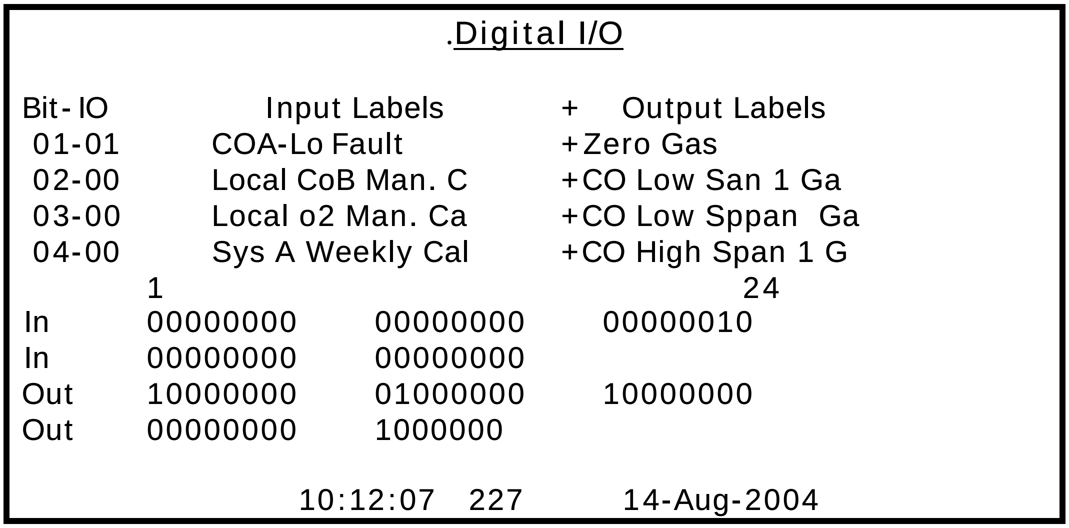
Selecting the I/O Now displays the instantaneous settings of the digital input and output bits as presented below. They update each second.



```
                         Digital I/O
     Exit
     I/O Now
     Set I/O
     Labels
        1                                  24
 In       00000000    00000000    00000010
 In       00000000    00000000
 Out      10000000    01000000    10000000
 Out      00000000    10000000
                   10:47:50      227      14-Aug-2004
```

Figure 4-16 Digital I/O Bits Display

Selecting the Labels options displays the labels that have been assigned to the digital I/O bits. Each I/O bit can have a 20 character label. For ease in reviewing on the LCD, these labels are limited to 16 characters on the front panel display. An example is presented below. The first two digits are the bit #, and the next two numbers are the digital input and digital output settings, respectively

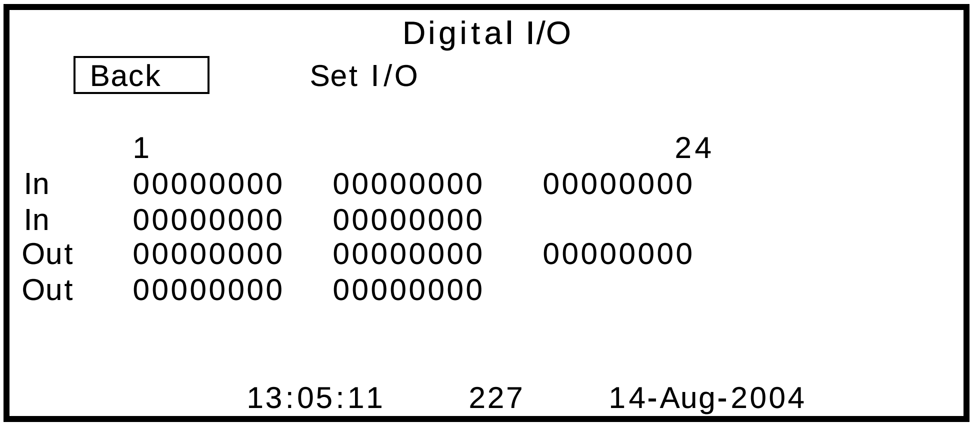


```
                        .Digital I/O

Bit-IO            Input Labels          +    Output Labels
 01-01      COA-Lo Fault                +Zero Gas
 02-00      Local CoB Man. C            +CO Low San 1 Ga
 03-00      Local o2 Man. Ca            +CO Low Sppan  Ga
 04-00      Sys A Weekly Cal            +CO High Span 1 G
        1                                      24
 In       00000000      00000000      00000010
 In       00000000      00000000
 Out      10000000      01000000      10000000
 Out      00000000      1000000

                   10:12:07   227       14-Aug-2004
```

Figure 4-17 Digital I/O Label Display

Turning the switch clockwise presents increasing numbers and turning the switch counter clockwise presents decreasing numbers. Pressing the pushbutton returns to the previous menu.

The user can also set and reset the digital I/O bits from this sub menu. Select the Set I/O option. The CPP presents the following screen.



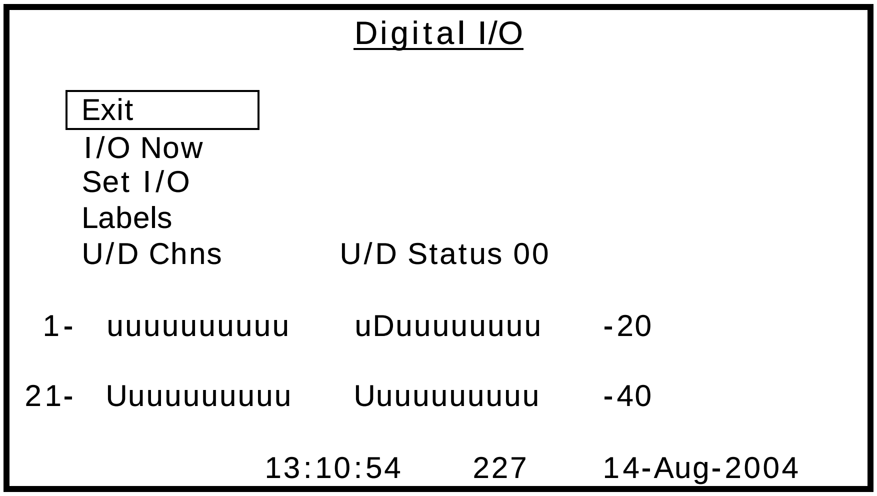
```
                    Digital I/O
   Back           Set I/O

       1                                  24
In     00000000   00000000   00000000
In     00000000   00000000
Out    00000000   00000000   00000000
Out    00000000   00000000


               13:05:11      227      14-Aug-2004
```

Figure 4-18 Set I/O Screen

Turn the switch until the In or Out that has the desired bit is highlighted with a blinking cursor. Press the pushbutton. The cursor will move to the first bit, reduce in size and blink. Turn the switch until the cursor is under the desired bit. Pressing the pushbutton will toggle the bit setting. Both input and output bits settings can be forced to a true state by the user. They can not be forced into a permanent off state. Set bits will remain in the set state until reset by the user. If a password is set into the front panel controls, this feature can not be entered without entering the password. The bit patterns presented in the screen are not the actual input/output settings. They are the operator settings registers which are “or’ed” into the I/O registers.

The CPP displays the bit number the cursor is on and any label assigned to this bit in the lower part of the display. This routine is exited by turning the switch counterclockwise until the cursor is blinking under the I or the O and depressing the pushbutton. The cursor is made larger to acknowledge the switch.

This sub menu allows the user to view the Up/Down status of all channels and to take channels up and down. Scroll to U/D Chns and press the pushbutton. The following screen is presented with the cursor blinking over the 00. Pressing the pushbutton here exits back to the Maim Menu. Turn the switch to move the cursor to any channel. The channel number is displayed where the 00 is presented below. Press the pushbutton to change the Up/Down status of the channel.

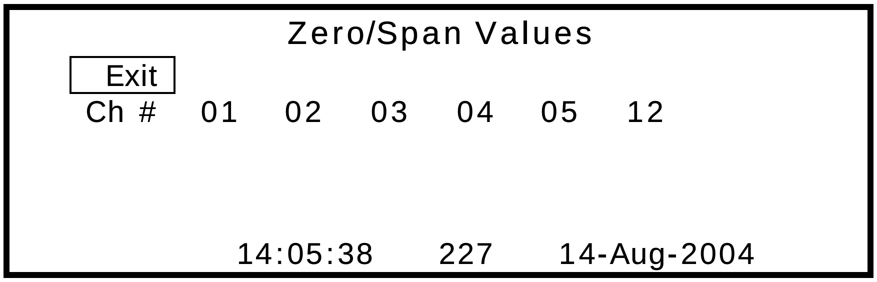


```
                    Digital I/O

   Exit
   I/O Now
   Set I/O
   Labels
   U/D Chns         U/D Status 00

  1-  uuuuuuuuuu   uDuuuuuuuu   -20

 21-  Uuuuuuuuuu   Uuuuuuuuuu   -40

             13:10:54    227    14-Aug-2004
```

Figure 4-19 Digital I/O Up/Down Channel Status

4.16 Zero/Span Values

Zero and span expected values can be viewed and entered from the front panel. Selecting this option brings up the following menu.

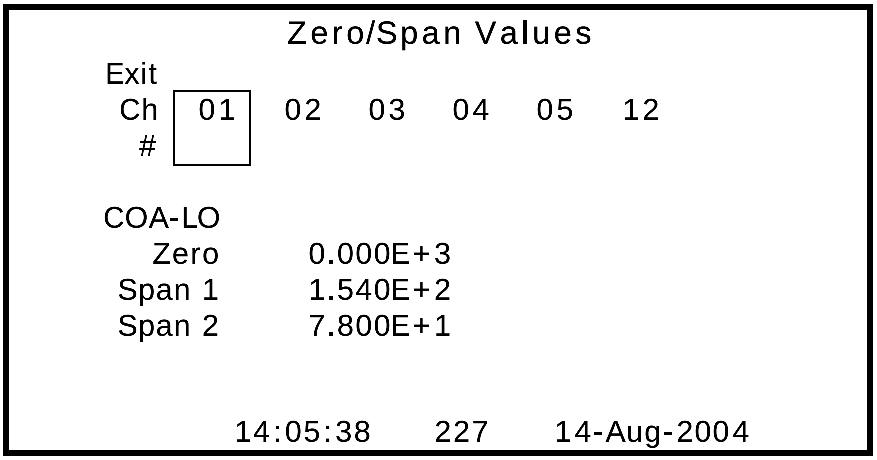


```
               Zero/Span Values
   Exit
   Ch #   01   02   03   04   05   12



           14:05:38    227    14-Aug-2004
```

Figure 4-20 Zero Span Value Menu

The CPP presents the channel numbers of the channels than have been initialized as calibration channels. If no channels have been set up as calibration channels, the CPP puts up such a message in place of the channel numbers. The only option enabled in this scenario is to exit back to the Main Menu. If channel numbers are presented, they can be selected by turning the switch. The CPP will the display the channel name and the zero and span values settings as the switch is turned. Press the pushbutton when the desired channel is highlighted to change the channel settings.

The Zero value will be highlighted. Turning the rotary switch moves the highlighted value up and down. If a value is to be changed, depress the pushbutton when the desired value is highlighted. The CPP will open another window to the right of the selected value with the selected value being displayed, +1.200E+2 R. The cursor will be blinking under the R. Turn the switch to move the cursor to the desired number or sign. Press the pushbutton to select this number. Turning the switch sequences the number 0 - 9. Press the pushbutton when the desired value is displayed. After the desired value has been entered, move the cursor to the R and press the pushbutton. The entered value will be moved into the value setting.



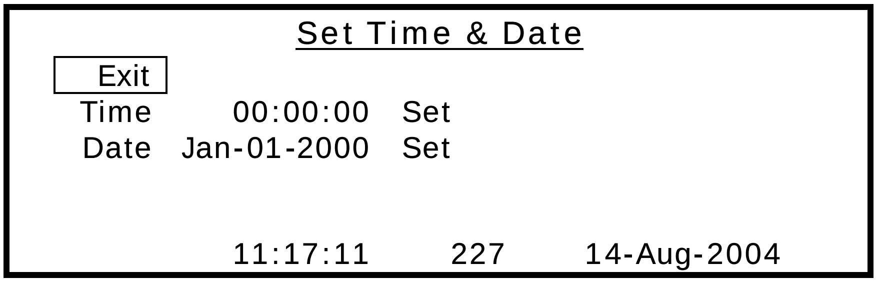
```
            Zero/Span Values
   Exit
     Ch   01    02    03    04    05    12
      #

   COA-LO
      Zero       0.000E+3
    Span 1       1.540E+2
    Span 2       7.800E+1


          14:05:38     227     14-Aug-2004
```

Figure 4-21 Zero/Span Values Channel Display

If a password has been set up, the values can not be changed without entering the password. This routine is exited by turning the switch counter clockwise until the Exit is highlighted and pressing the pushbutton.

4.17 Set Time & Date

The time and date of the CPP can be set from the front panel. Selecting this option brings up the following menu.



```
               Set Time & Date
   Exit
   Time       00:00:00   Set
   Date    Jan-01-2000   Set


            11:17:11      227      14-Aug-2004
```

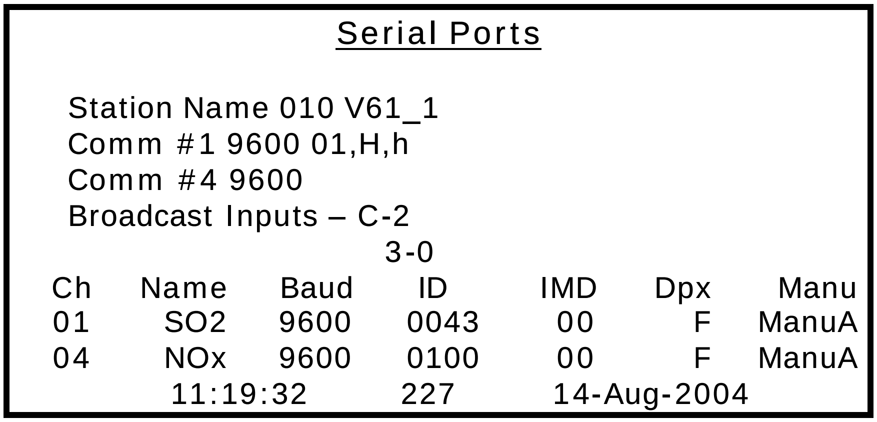
Figure 4-22 Set Time & Date Menu

To set the time or date, turn the switch until either Time or Date is highlighted. Press the pushbutton. If time is being set, the cursor starts blinking under the hours of the time. To set the hours, press the pushbutton. The cursor changes sizes. Turn the switch until the desired hour is being displayed and press the pushbutton. The cursor changes sizes again. Continue this process for the minutes and the seconds. After the time has been set turn the switch clockwise until the cursor is blinking under the Set. Press the pushbutton. The cursor reduces in size. Press the pushbutton again and the cursor disappears. This stores the setting to update the CPP clock upon exiting the routine. The Time is highlighted. Use the rotary switch and pushbutton to exit or to set the date.

If the time or date settings are not set in by moving the cursor under the Set and pressing the pushbutton, the setting will not be transferred into the CPP. Turning the switch counterclockwise until the cursor disappears does not store the settings to update the CPP clock. If a password has been set up, the password must be entered before the time or date can be changed.

4.18 Serial Ports

This option allows the user to review the serial port baud rates and assignments. Selecting this option brings up the following information.



```
                         Serial Ports

    Station Name 010 V61_1
    Comm #1 9600 01,H,h
    Comm #4 9600
    Broadcast Inputs – C-2
                          3-0
   Ch     Name     Baud     ID       IMD     Dpx     Manu
   01      SO2     9600    0043       00       F   ManuA
   04      NOx     9600    0100       00       F   ManuA
            11:19:32        227        14-Aug-2004
```

Figure 4-23 Serial Port Data Display

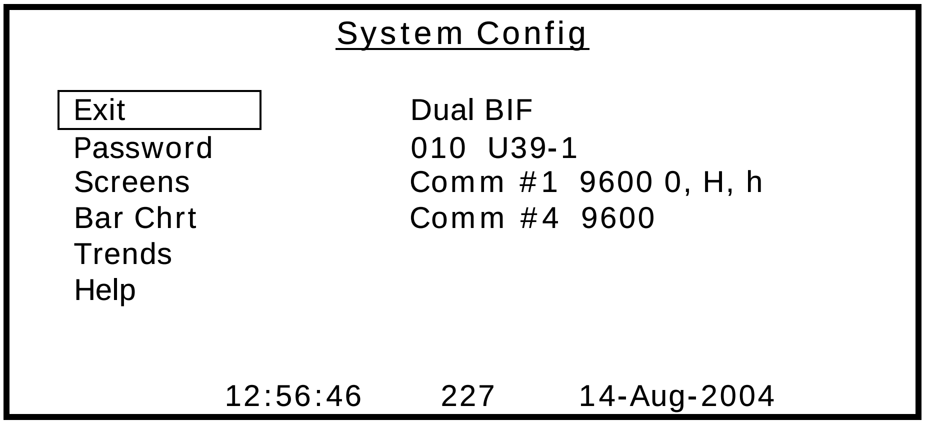
The top line gives the CPP assigned station name, the station numerical ID, and the installed program version. The second line provides information for comm port #1. In the example the Baud rate is 9600, and the DSM emulation ID is 01. The capital H has no relevant meaning to the user. The little h is the ID that invokes a quiet or computer transfer of data. This data is transferred as a comma delimited ASCII file.

The next line indicates the Baud rate of comm port #4. This is followed by a listing of any instrumentation connected that operates in the broadcast mode. This is normally meteorological sensors.

The bottom portion is used to provide information on any channels collecting data from instruments via their serial ports. When first selected, the CPP presents any channels connected to comm port number 3-0, as given in the example. Use the rotary switch to view other comm ports. This routine is exited by pressing the pushbutton.

4.19 System Configuration

Selecting this option allows the user to set up various operating parameters of the LCD display. The CPP brings up the following sub menu.



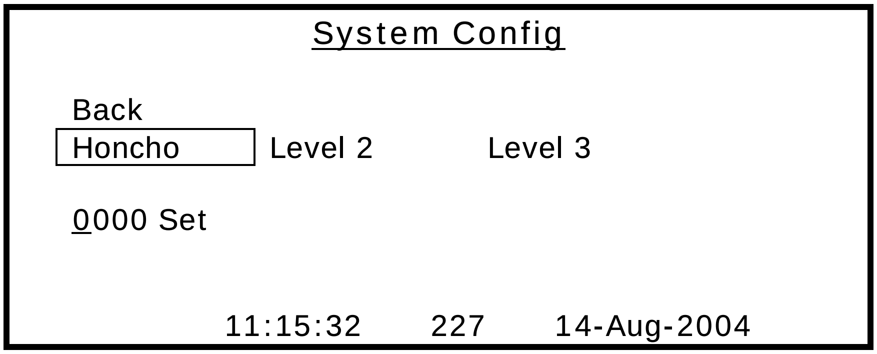
```
                    System Config

   Exit                     Dual BIF
   Password                 010  U39-1
   Screens                  Comm #1  9600 0, H, h
   Bar Chrt                 Comm #4  9600
   Trends
   Help


           12:56:46     227     14-Aug-2004
```

Figure 4-24 System Configuration Menu

4.19.1 Password

The following discussions use the term password, although it really references a hexadecimal pass code. The front panel uses numbers 0 - 9, and letters A - F. The CPP supports three levels of password regarding changing system operating parameters from the front panel controls. If a password has not been set up, making this selection brings up a sub menu for setting up the three levels of password. If a password has been set up, then the CPP requests the password which must be properly entered before further access is granted.

The CPP puts up four zeroes with the cursor blinking under the first digit. Select this digit by pressing the pushbutton. The cursor changes. Use the switch to select the desired code and press the pushbutton. Repeat this procedure until all four codes have been entered, then select Code and press the pushbutton. If the correct code has been entered, access is granted, else the request is aborted.

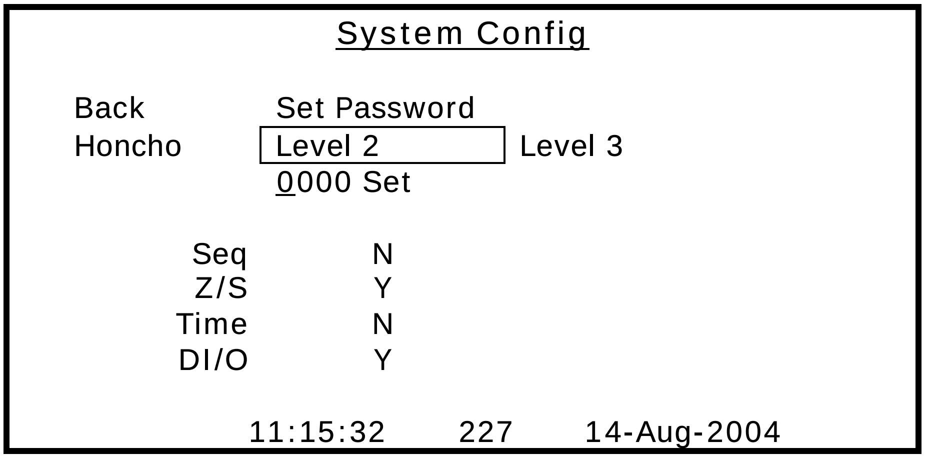


```
                  System Config

   Back
   Honcho      Level 2        Level 3

   0000 Set


         11:15:32     227     14-Aug-2004
```

Figure 4-25 System Config Password Menu

The Honcho password has the highest authority, and no other level can be set if the Honcho password is not set. Clearing the Honcho password clears levels 2 and 3 as well. The Honcho password has access to all front panel controls. Selecting the Honcho results in four zeroes being presented underneath the Honcho on the screen, with the cursor blinking under the first digit. This is shown above.

A password code of all zeroes is a null and disables the password feature. To change or set a code, turn the switch until the cursor is blinking on the desired digit and press the pushbutton. Turn the switch until the desired number or letter is being displayed, and press the pushbutton. Repeat for all digits. Zeroes are valid inputs, and unchanged zeroes are considered part of the password if any digit is set to non zero. After setting a Honcho password, passwords for levels 2 and 3 can be set. Setting a password for level 2 or level 3 brings up the following screen that allows features to be assigned to this level of password. Features not assigned to this level can not be accessed.



```
                    System Config

   Back          Set Password
   Honcho        [Level 2   ]  Level 3
                 0000 Set

           Seq        N
           Z/S        Y
          Time        N
          DI/O        Y

           11:15:32    227    14-Aug-2004
```

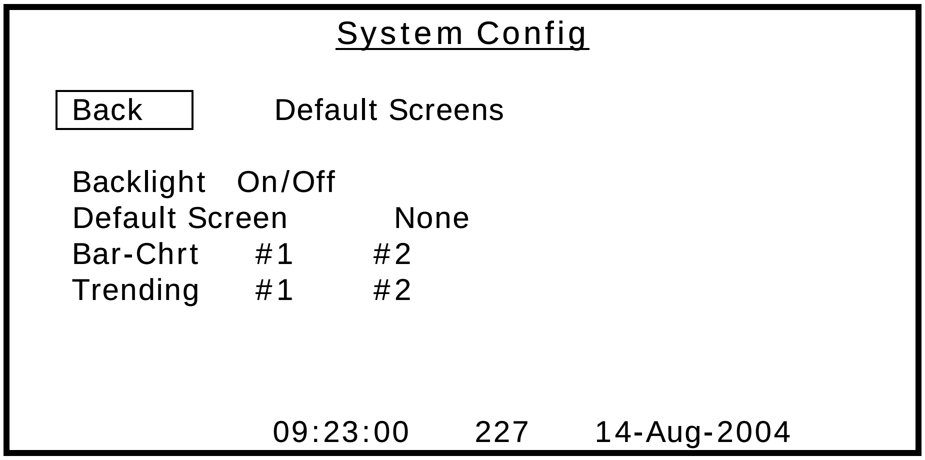
Figure 4-26 Set 2nd Level Password & Functions

4.19.2 Screens

This selection allows the user to have a default screen, such as a trend chart instead of the Main Menu, to come up following a power fail. It also allows the user to have the CPP not turn off the backlight. Any special screens provided in this firmware version will also be included in the selection menu. Selecting the screens option brings up the following sub menu.

To control the operation of the backlight, turn the switch until the On/Off is highlighted. Pushing the pushbutton toggles this setting between On, and On/Off.

To select a default screen following a power up turn the switch until the desired screen is highlighted and press the pushbutton. The selection should appear in the Default Screen box. Pressing the pushbutton when only the Default Screen box is highlighted clears any default screen, and the Main Menu is presented following a power up.



```
                    System Config

 Back          Default Screens

  Backlight   On/Off
  Default Screen         None
  Bar-Chrt     #1      #2
  Trending     #1      #2



              09:23:00     227     14-Aug-2004
```

Figure 4-27 System Config Screens Menu

The “Input” above is the CPP channel number. Under the type category, an A indicates an instantaneous input, a P a preliminary input, an I an interim input, and an F a final input.

Use the switch to select the desired Bar Chart number and press the pushbutton. The CPP will present any stored set up and the highlight the first Position #. Use the switch to select the desired input (Position #) and press the pushbutton to select. The CPP will also highlight the Input. Turning the switch allows the desired channel number to be selected. Press the pushbutton to select the displayed channel number. The CPP then highlights the Type. Use the switch the flip between the A, P, I, and F selections. Press the pushbutton to select the desired input type.

4.19.3 Trend Charts

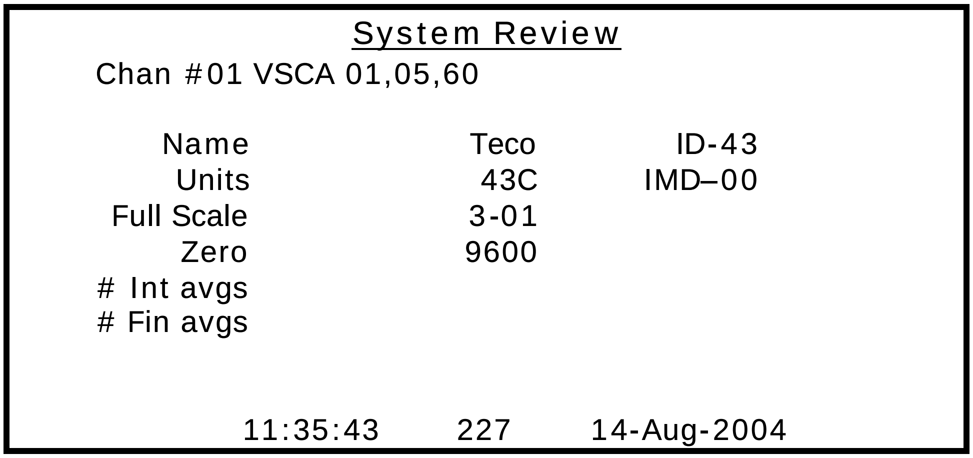
This feature is for special applications only.

4.19.4 System Review

This option allows the user to select a channel and review how it is set up. This is analogous to what is presented in the L key command via comm port #4. Select the desired channel using the rotary switch. Press the pushbutton to view the displayed channel number. The cursor changes and the CPP displays the channel’s set up. To get back to the rotary switch, press the pushbutton. To exit this routine, select channel number 00 and press the pushbutton.

If this channel is collecting data from an instrument via a serial port, the CPP presents the following information.

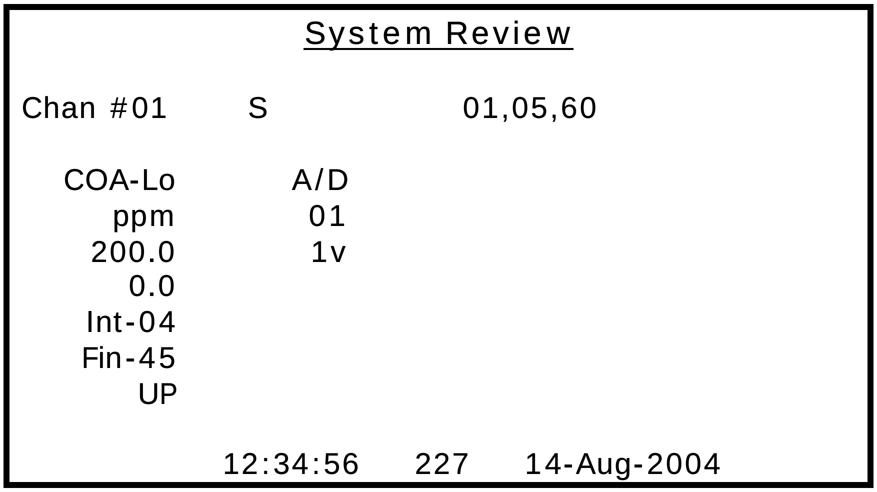
The VSCA represent the following; V - validity, S - Sample delay, C - calibration, A - alarm. These characters are presented only for the functions that have been assigned to this channel. If there were no alarms set up for this channel, but the other functions applied, then VSC would be displayed. The 01, 05, 60 are, respectively, the preliminary average, the interim average, and the final average.



```
                    System Review
   Chan #01 VSCA 01,05,60

        Name              Teco        ID-43
       Units               43C       IMD–00
  Full Scale              3-01
        Zero              9600
   # Int avgs
   # Fin avgs


          11:35:43     227     14-Aug-2004
```

Figure 4-28 System Review Display

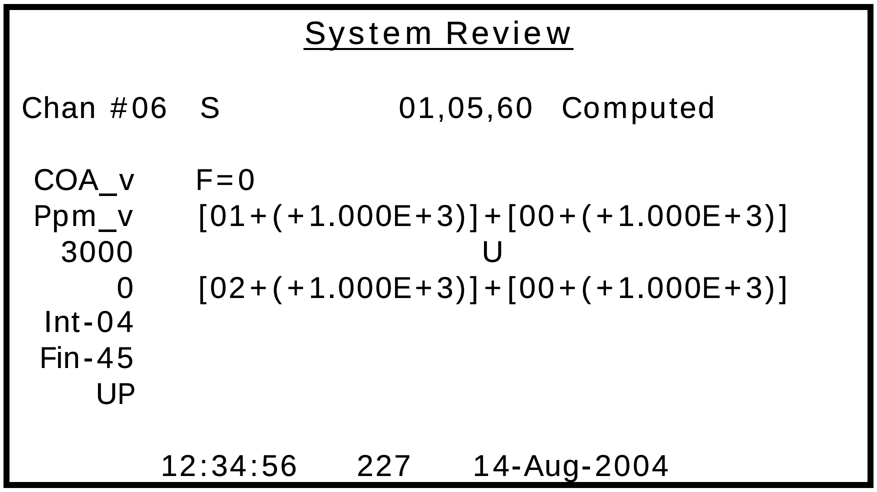
If the selected channel is getting data from the analog to digital converter, the CPP presents the following.



```
                System Review

Chan #01      S           01,05,60

   COA-Lo       A/D
      ppm        01
    200.0        1v
      0.0
   Int-04
   Fin-45
       UP

        12:34:56    227    14-Aug-2004
```

Figure 4-29 Analog to Digital Channel Display

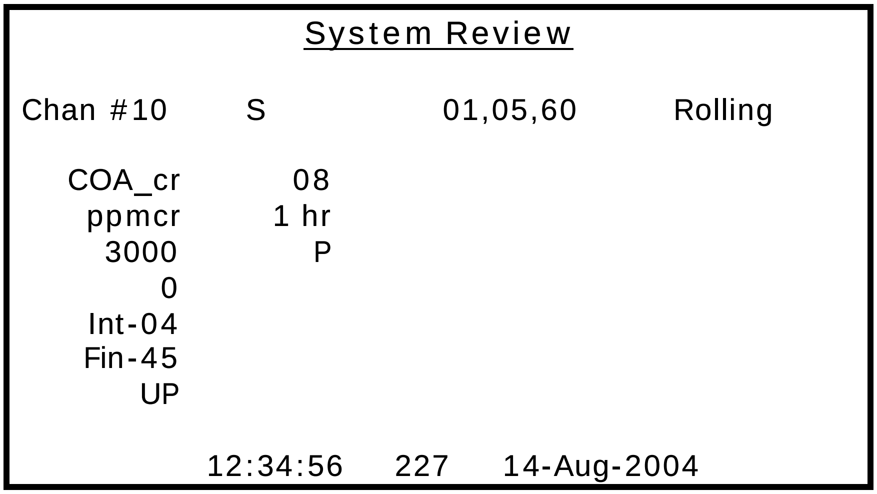
If the selected channel is a computed channel, then the CPP presents the following. The F=0 indicates the function code applied to the final computed number. Zero indicates that no other function applies. A one indicates that the square root of the final number should be taken and stored.



```
                    System Review

Chan #06   S              01,05,60   Computed

 COA_v     F=0
 Ppm_v     [01+(+1.000E+3)]+[00+(+1.000E+3)]
   3000                     U
      0    [02+(+1.000E+3)]+[00+(+1.000E+3)]
  Int-04
  Fin-45
     UP

        12:34:56    227    14-Aug-2004
```

Figure 4-30 Computed Channel Display

If the selected channel is a rolling average channel, the CPP presents the following. In the example, the 04 is the input channel data, and this channel has been set up to form a 1 hour rolling average from the preliminary averages.



```
                    System Review

Chan #10       S            01,05,60        Rolling

    COA_cr          08
    ppmcr         1 hr
     3000            P
        0
    Int-04
    Fin-45
       UP

          12:34:56    227    14-Aug-2004
```

Figure 4-31 Rolling Average Channel Display

If the selected channel has also been configured as a meteorological channel the CPP so indicates in the first line, as presented below.

System Review

Chan #01 01,05,60
Met HWD5V - 02
(HWS V - 01 - 01) other examples

The above would indicate that this channel is being treated as a 540 degree horizontal wind direction (HWD5) input, and a vector wind direction is being calculated using channel #02 as the associated wind speed. If a scalar wind direction were being calculated, the CPP would present HWD5S. If a 360 degree sensor is being used, the 5 becomes a 3. If the channel is conducting a sigma theta calculation, the mnemonic would be ST - 02, where the 02 indicates the channel being used as the horizontal wind direction channel. A sigma W would be presented as SW - 02.

The CPP supports a direct met interface which uses internal counters to determine the wind speed. If an internal counter is being used, then the second number is the number of the internal counter. If the internal counters are not being used, then nothing is printed.

4.19.5 Binary Status Indications

Each data point has sixteen independent status bits that can be set. As more than one status indication can be set for any data point, the status presented on the front panel LCD is presented in hexadecimal. This allows all status indications to be presented and reviewed. A hexadecimal character represents four binary bits. It takes two hexadecimal characters to make one 8 bit binary byte. The following table can be used to convert hexadecimal to binary.

4.19.6 Hexadecimal to Binary Conversion

Hex	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Binary	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111

Bytes	|	|	|	|
Upper Part	7	6	5	4
Lower Part	3	2	1	0

Figure 4-32 Hexadecimal to Binary Conversion

4.19.7 Binary Status

The first status byte contains status information that is normally provided by most instrumentation. The status conditions in byte number one can also be generated inside the CPP. For example, the high/low alarms may not be set up in the instrument, but are set up in the CPP. In this case these status bits would be provided by the CPP. Obviously both could be set up. If the status condition is set by one it will not be reset by the other.

A good data value does not have a modifier printed. The status modifiers M and B indicate Missing and Bad, respectively. If any other status bits are set, then a secondary flag indicator may be also be printed. In highest to lowest priority these are printed as; F, Y, Z, R, L, C, D, H, I, A, a, >, <.

First Status Byte

Status Code	Bit	Meaning		
M	7	- 00 = Missing	Bits 7 & 6 are decoded	
G	6	- 10 = Good	as given here	
B		- 01 = Bad		
o		- 11 = Other		
R	5	- 1 = Instrument No response		
L	4	- 0 = Instrument On Line, 1 = Off Line		
P	3	- 1 = Instrument Parameter Alarm		
A	2	- 1 = Data High Alarm		
a	1	- 1 = Data Low Alarm		
C	0	- 1 = In Cal, 0 = In Sample		
Second Status Byte				
F	7	- 1 = Power Fail		
>	6	- 1 = Not All Samples , But More Than Required		
<	5	- 1 = Not All Samples, And Less Than Required		
H	4	- 1 = Value being held		
D	3	- 1 = Downed By Operator		
	2	= Encoded	0 = U - Nothing	4 = S - Smp Delay
	1	= “	1 = Z - In Initialization	5 = Undefined
	0	= “	2 = Y - A/D Calibration	6 = Undefined
			3 = V - Validity	7 = * Cal Alarm

Figure 4-33 Binary Status Code Definitions

Bits 2, 1, an 0 of the second status byte are available to be encoded to provide additional data status. If they are used they are defined in a special applications section. These bits are defined in the standard CPP in several categories.

4.19.8 Common Status Indications

Some common status conditions are presented below

Status	Condition
0000	Missing
4000	Bad
8000	Good, no other status conditions
C003	Bad because of status
C001	In Initialization
C004	Sample Delay
C400	High Alarm
C200	Low Alarm
C100	In Calibration
C080	Power Fail
C040	Good - Not all samples in average, but meets set up criteria
4020	Bad - Not all samples in average, and fails set up criteria
4010	Bad - operator downed
2000	Instrument or A/D no response

Figure 4-34 Common Status Code Definitions

5 Comm #1 Pass Through

Communication port number 1 is the port that is normally connected to a central facility and is the port over which data is retrieved from the CPP. Using simple commands, comm port #1 can be electrically connected to comm port #4 or the comm #3 ports.

5.1 On Line Pass Through

The CPP supports two types of pass through operation. One is an on line mode in which commands received on comm port #1 are interleaved with CPP polling commands and sent to connected instrumentation. The instruments response is then returned to comm port #1. This mode has the benefit of the instrumentation staying on line with the CPP, but has the drawback that a certain protocol must be maintained for proper operation.

To communicate with an instrument, connect to comm port #1 with the $ID, and enter SETF<Enter>. Then enter D43<Enter> and the CPP will respond with a list of instruments connected to the unit. Select the channel of the desired instrument and begin communications. The on line pass through is discussed in more detail and in each application note for each manufacturer.

5.2 Off Line Pass Through

The second type of pass through places the instrument in an off line mode with the CPP. In this mode, commands from comm port #1 are passed unabated to the instruments comm port, and instrument responses are returned to comm port #1. The CPP does not send commands to an instrument that is this type of pass through. This allows the user to conduct manufacturer supplied diagnostics on the instrument without interference from the CPP. The disadvantage is that the instrument is off line with the CPP and data is not being collected during this time. All other instruments remain on line.

5.3 Commands

Connect with comm port #1 by entering $ID ($1 for example). The CPP echoes the ID. Figure 5-1 below presents the commands used to pass through comm port #1 to another comm port in a CPP offline mode.

To pass through to an instrument, select the appropriate comm port, *e.g.* SET0<Enter>. This connects comm port #1 to comm port #3-0. Commands and instrument responses can now be exchanged between the user and the instrument, without interference from the CPP. The CPP will remain in this mode until either a $ is received on comm port #1, or one hour elapses, at which time the CPP will switch out of the pass through mode and return to normal operation.

```
SETF<Enter> = Pass comm port #1 to comm port #4
SET0<Enter>  = Pass comm port #1 to comm port #3-0
SET1<Enter>  = Pass comm port #1 to comm port #3-1
SET2<Enter>  = Pass comm port #1 to comm port #3-2
SET3<Enter>  = Pass comm port #1 to comm port #3-3
SET4<Enter>  = Pass comm port #1 to comm port #3-4
SET5<Enter>  = Pass comm port #1 to comm port #3-5
SET6<Enter>  = Pass comm port #1 to comm port #3-6
```

Figure 5-1 Pass Through Commands

5.4 Comm Port #1 to Comm Port #4

The SETF<Enter> command internally connects comm port #1 to comm port #4. This allows the user complete control over the CPP.

6 Data Printouts and Reporting

6.1 General

The CPP offers a broad range of printouts that provide reports of collected data, status, calibrations, power fails and so forth. Reference the List of key commands in section 1.24. Some examples are provided in this section.

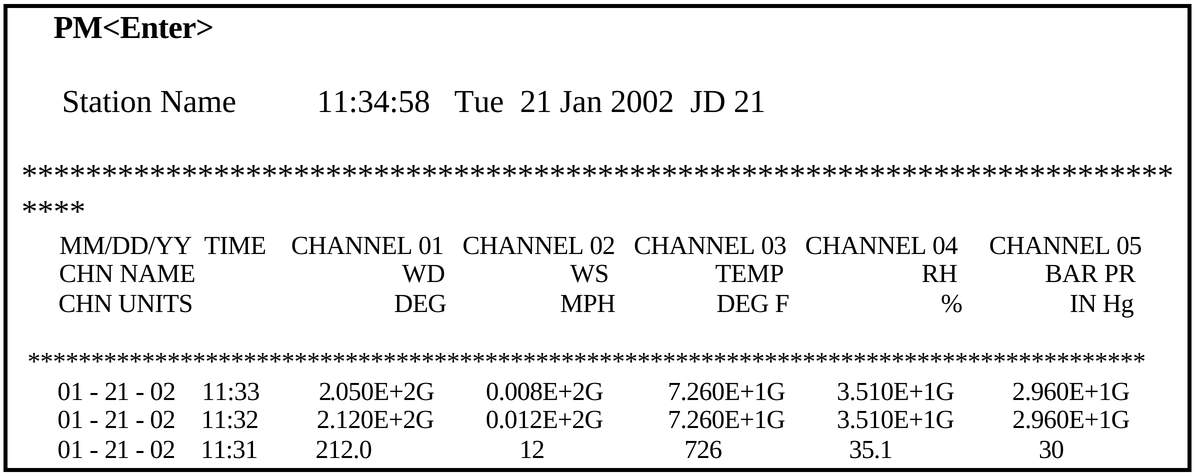
6.2 L Key Command - Listing

An **L** key command commands the CPP to print a listing of how it has been set up to operate. This is a long printout. The print out can be stopped by inputting a carriage return, and the print out continued by inputting another carriage return. Most of the printouts offered in the **L** key command are presented in the individual sections of the manual in which they are discussed. An escape key command stops the print.

The **L** key command is a very valuable tool if problems are encountered in the field. When contacting H2NS customer service, please have an L key command listing ready to Fax or e mail to H2NS.

6.3 Data printouts

All data printouts provided by the CPP are very similar, about the only difference is in the length. Presented in Figure 6.1 below is a typical data printout offered by the CPP. This particular printout is from a **PMxxx** key command, which requests a printout of the most current one minute data. The **xxx** is the number of one minute averages to print. If the **xxx** is omitted the PM input will print the ten most current one minute readings. In the example below only three time periods are shown. The CPP can be set up to present data in either scientific or engineering units. Figure 6.1 shows both types. A printout in engineering units is dependent on the number input or full scale, particularly the location of the decimal point. If a number cannot be printed in an engineering format with the given full scale and zero values, the OVRNG is printed. The data can always be viewed when converted to scientific notation.



```
 PM<Enter>

  Station Name        11:34:58   Tue  21 Jan 2002  JD 21

************************************************************************
****
  MM/DD/YY  TIME   CHANNEL 01  CHANNEL 02  CHANNEL 03  CHANNEL 04   CHANNEL 05
  CHN NAME                 WD          WS        TEMP          RH       BAR PR
  CHN UNITS               DEG         MPH       DEG F           %        IN Hg

************************************************************************************
  01 - 21 - 02   11:33     2.050E+2G   0.008E+2G    7.260E+1G   3.510E+1G    2.960E+1G
  01 - 21 - 02   11:32     2.120E+2G   0.012E+2G    7.260E+1G   3.510E+1G    2.960E+1G
  01 - 21 - 02   11:31     212.0            12          726          35.1          30
```

Figure 6-1 Typical Data Printout

Some of the data printouts can be fairly long. Inputting a carriage return will cause the printout to pause so that it can be reviewed. Inputting another carriage will resume the printout. The **PM** input will print the ten most current one minute readings back to an hour boundary.

The data is presented in either scientific notation or in engineering units. The wind direction value 2.050E+2G is 205.0 degrees with good status. The CPP will print up to two status indications with each data point. As an example X.XXXE+0oA, where the o indicates that other status bits are set and the A (high alarm) is the first bit that was encountered.

The **PIx** key command requests the interim averages from the present time back to the hour boundary. A **PI1** input request the previous hours interim averages. A **PFx** input requests the final averages from the present time back to the time frame boundary. A **PF1** input requests the previous time frames final averages. A **PF2** requests the time prior to that and so forth.

A similar format is used to print out calibration data. The key command for calibration data is the **PCx** command.

6.4 Auto Prints

The CPP can be set up to automatically print out certain reports. In addition, the reports to be automatically printed can be changed based on the setting of control output bits. The most common example would be printing minute averages in an alarm condition but not under normal operating conditions. In the alarm initialization, a particular bit is set in an alarm condition that controls the printing of the one minute averages. The set up of the automatic printouts is entered from the Main Menu.

```
Auto Prints
None(0), Prelim(1), Interim(2), Final(4) = 4  <Enter>
None(0), Hourly(1), Daily(2) = 0  <Enter>
Digital Out Print
Prelim  Y/N Y  <Enter>
Print On Digital Out
(1 - 8)  XXXXXXXX  XXXXXXX1  <Enter>
(9 -16)  XXXXXXXX  <Enter>
(17-24)  XXXXXXXX  <Enter>
(25-32)  XXXXXXXX  <Enter>
(33-40)  XXXXXXXX  <Enter>
Interim Y/N N <Enter>
Final   Y/N N  <Enter>
Main Menu
```

The first question allows the user to set up an automatic printout for immediate results. Additional choices not found in the set-up prompt are listed below in Figure 6-2. In the example, the user has selected only the final average to be automatically printed out. The CPP looks for certain bits to determine what is to be automatically printed.

Input	Result
0	None
1	Print preliminary average only
2	Print interim averages only
3	Print preliminary & interim averages
4	Print final averages only
5	Print preliminary & final averages
6	Print interim & final averages only
7	Print preliminary, interim & final averages

Figure 6-2 Multiple Print Set Up

The controlled printouts are set up similar to the automatic printouts, except that the CPP must detect a particular control bit being either a one or a zero before the selected averages will be printed out. The conditions set up in the control printouts is an **or** condition. If the printout needs to be activated by more than one control bit, the Boolean functions must be used to provide the decoding and used to set only one bit to activate the printout.

In the example, the user has entered a one in bit number eight. Anytime the CPP forms a one minute (preliminary) average and bit number eight is a one, the average will be printed out. The x's are interpreted as don't cares by the CPP.

7 Meteorological Parameters Set Up

7.1 General

A large number of meteorological parameters require special attention, and therefore a separate set up section. The parameters that are included in the meteorological set up are:

Wind Direction(HW)	
Wind Speed	(HW)
Sigma Theta	(ST)
Sigma W	(SW)
Rainfall	(RF)
Vertical Winds (VW)	

7.2 Wind Speed/Wind Direction

In the general channel initialization, channels that are wind speed (WS) and wind direction (WD) signals should be set up. Wind speed and wind direction signals can be input into the CPP in various fashions, which are covered below.

7.2.1 Signal Conditioners

Manufacturers of meteorological equipment all provide signal conditioning units that take the raw signals from the sensors and offer a linear voltage output. For example, 0 – 100 mph wind speed is equal to 0 to 1.0 volt. These voltages are input into the CPP over an A/D input channel. In the met set up, the A/D channels to which these signals are connected must be input.

7.2.2 Serial Input

Some manufacturers offer wind speed and wind direction data in a serial RS-232 string. In some units, other signals are offered as well. These signals are connected to a serial input on the CPP, for example comm port #2. All of these serial interfaces broadcast the serial strings at periodic intervals. The CPP detects the string and parses out the desired information. In the general initialization, a CPP channel is set up to accept the serial string. In the met set up, the CPP channel to which the serial string is assigned must be input.

7.2.3 Direct Met Interfacing

The CPP also offers a board for interfacing directly to the meteorological sensors. The board accepts the raw signals from the sensors and converts these signals into the appropriate readings for input into the CPP database. The direct met interface offered by H2NS has three wind speed counters, four voltage references for wind direction sensors, a battery backed rainfall counter, three interfaces for thermistors, and two signal amplifiers with an adjustable gain. The factory default setting is a gain of ten. In the met set up, the channels set up as the wind speed and wind direction channels must be input.

The rear panel connections for the direct met interface board are defined below in section 8.

To complete the WS/WD setup, from the main menu select Met Set Up. The CPP asks for the following information:

Channel # = **01<Enter>**
Channel Set Up = HW, ST, VW, SW, RF, Clr = **HW<Enter>**
WS/WD Chan = WS **<Enter>**
Vector(V), Scalar(S) = S **V<Enter>**
Associated WD Channel = 00 **02<Enter>**
Enable Y/N = N **Y<Enter>**
Ctr(#) or Other(0) = **1<Enter>**
100mph Frequency = 0.000E+0 **5.561E+1<Enter>**
Starting Threshold = 0.000E+0 **0.630E+0<Enter>**
Channel # =

Channel # = 00 **02<Enter>**
Channel Set Up = HW, ST, VW, SW, RF, Clr = **HW<Enter>**
WS/WD Chan = WS **WD<Enter>**
540(1) Or 360(0) = 0 **<Enter>**
Vector(V), Scalar(S) = S **V<Enter>**
Associated WD Channel = 00 **01<Enter>**
Enable Y/N = N **Y<Enter>**

Channel # = **<Enter>**
Return to Main Menu

Note: The Clr clears any set up on the selected channel.

The CPP assumes that the wind speed always has an associated wind direction and vice versa. In the above example, the operator set up channel #1 to calculate a vector wind speed average and to use the signal from channel #2 as the associated wind direction channel. The CPP would have asked for the associated wind direction channel even if a scalar average had been set up. The wind direction signal is just not used in the calculation of a scalar wind speed average.

The operator then set up channel #2 to calculate a vector wind direction average and to use channel #1 as the associated wind speed channel. The CPP would have asked for the associated wind speed channel even if a scalar wind direction average had been set up. The wind speed signal is just not used in the calculation of a scalar wind direction average.

For wind direction channels the CPP asks in the incoming signal is from a 360^0 or a 540^0 sensor. This is used when the signal input is analog from an A/D channel. It is asked but not used when the signal is coming from a smart sensor over the RS-232 port as the signal is provided in a 0 - 360 degree format.

The calculations are not enabled until the Enable Y/N is answered with a Y.

In the example, the wind speed is being derived from an internal counter (counter #1) that interfaces to the wind direction sensor. The CPP supports up to three wind direction counter interfaces.
The CPP then asks for the incoming frequency from the sensor for a 100 mph wind speed. Equivalent values for other units can be entered as well, e.g., meters/second. The CPP then asks for the starting threshold of the sensor.

Had the user entered a 0, the questions regarding the counter are not asked. Entering a 0 indicates that the wind speed signal is coming in on the CPP channel as an A/D input or a serial connection. These inputs are set up in the general initialization.

For a counter

```
 Chan #      Parameter          Manu     Port          Baud
   01       HW WS   V 02        C-1    5.561E+1      0.630E+0
```

Where the 01 is the channel number, HW indicates a horizontal wind, parameter, WS indicates a wind speed, V indicates a vector average is being formed, 02 indicates that channel 02 is the associated wind direction parameter, and the C-1 indicates counter #1, the first number the sensor frequency at 100mph, and the last number is the starting threshold.

The L Key printout for WS/WD channels with data coming in on a serial port is presented as shown below

```
Chan #       Parameter          Manu   Port   Baud
  01        HW WS    V 02       CLIM  3-C2   19.2K
  02        HW WD3 V 01         CLIM  3-C2   19.2K
  03        ST         R 02     CLIM  3-C2   19.2K
```

The manufacturer is Climatronics, the signal is connected to comm port #2 and operates at 19.2K Baud. Channel #1 is to calculate a horizontal wind speed, vector average using channel #2 as the associated wind direction signal. Channel #2 is to calculate a horizontal wind direction vector average using channel #1 as the associated wind speed signal. The WD3 indicates that the input is from a 360 degree sensor. This is important only if the input is in an analog format.

In the printout, CPP channel # 03 has been set up to calculate a revised, sigma theta. In the channel initialization, this channel was set up as receiving data over comm port #2, but the CPP interprets the Sigma theta calculation and retrieves data from the appropriate locations. The R 02 indicates that a revised sigma theta is to be calculated, and that CPP channel #02 is the wind direction channel.

If the signals were coming from the A/D converter then the L key in printout would look as given below.

```
Chan #       Parameter          Manu    Port   Baud
  01        HW WS    V 02        A/D    A-04
  02        HW WD3 V 01          A/D    A-05
```

```
        10              ST            S 02           A/D     A-05
```

The CPP channel number and the A/D channel number do not have to be the same. In the above example, CPP channels 01 & 02 are receiving data from A/D channels 04 & 05, respectively.

In the printout, CPP channel # 03 has been set up to calculate a standard, sigma theta. In the channel initialization, this channel was set up as receiving data from the A/D, but the CPP interprets the Sigma theta calculation and retrieves data from the appropriate locations. The S 02 indicates that a standard sigma theta is to be calculated, and that CPP channel #02 is the wind direction channel.

7.3 Set Up Of Sigma Theta

The CPP supports the computation of two types of sigma theta calculations. The first takes 3600 WD samples over the hour and calculates the sigma theta. The second breaks the hour into four fifteen minute periods, calculates the variance for each fifteen minute period, and then averages the variances as given in the USEPA guidance manual, to arrive at the sigma theta value. The CPP is set up as follows. From the Main Menu select the set up of meteorological parameters.

Channel # = **10<Enter>**
Channel Set Up = HW, ST, VW, SW, RF, Clr = **ST<Enter>**
Wind Direction Channel = 00 **02<Enter>**
Standard (S)
Revised (R) = R = **<Enter>**
Enable Y/N = N **Y<Enter>**

Channel # =<Enter>
Return to Main Menu

Note: The Clr clears any set up on the selected channel.

Sigma theta’s are calculated and reported for the preliminary, interim and final averages.

7.4 Sigma W – Vertical Wind Speed

The CPP supports a sigma w calculation. This normally operates in conjunction with a vertical wind speed channel. However, the sigma w algorithm provides a standard – standard deviation calculation - and can be used on other signals. Contact H2NS to use the calculation in this fashion.

From the Main Menu select the met set up. The vertical wind channel should be set up first. This is because there are two types of vertical wind speed sensors in common use. One outputs a unipolar signal and the other a bipolar signal. In the initialization of the vertical wind, this question is asked. When the associated sigma w channel is set up, it retrieves the unipolar/bipolar setting from the vertical wind speed channel. The default is unipolar.

The full scale and zero values for the vertical wind speed channel should be set up to match the incoming signal. The sigma w channel should be set for a full scale value of 100.0 and a zero value of 0.0.

Vertical Wind Speed Set Up
Channel # = **03<Enter>**
Channel Set Up = HW, ST, VW, SW, RF, Clr = **VW<Enter>**
Unip(U), Bip(B), = U **<Enter>**

Channel # =<Enter>
Return to Main Menu

Note: The Clr clears any set up on the selected channel.

Sigma W Set UP
Channel # = **14<Enter>**
Channel Set Up = HW, ST, VW, SW, RF, Clr = **SW<Enter>**
Vertical Wind Speed Ch = 00 **03<Enter>**

Channel # =<Enter>
Return to Main Menu

Note: The Clr clears any set up on the selected channel.

The L key command will look as follows.

```
     Chan #        Parameter            Manu   Port    Baud
       03         VWS     U             A/D    A-02
       14         SW      02            A/D    A-14
```

7.5 Rainfall

The CPP supports both tipping buckets and analog type rainfall inputs.

7.5.1 Tipping Bucket

The CPP supports a tipping bucket rainfall input. This is set up and enabled in the Met Set Up portion of the CPP initialization. The CPP has two places for connecting a tipping bucket rain gauge. If the CPP has the Direct Met Interface board installed, then the rain gauge should be connected to the RF input on this board. If a Direct Met Interface is not installed in the CPP, then the rain gauge signal lines should be connected to status input number 1 on the CPP back panel. The rainfall counter has the advantage of having a battery backup that allows rainfall ticks to be accumulated during power outages. For a tipping bucket type input the values entered as full scale and zero values as well as the units entered are not used by the system. The value of each tip is set up in the initialization sequence presented below.

1. Channel # = **10<Enter>**

2. HW, ST, VW, SW, SR, RF, Clr = **RF<Enter>**
3. Enable Y/N = N **Y<Enter>**
4. Tipping; (C)Counter or (D)Digital, (A)nalog or (W)eighing = **D<Enter>**
5. 0.01in (1), 0.1mm(2) = **1<Enter>**
6. Channel # = **<Enter>**
   Return to Main Menu

Line 1 asks which CPP channel is to be set up. In the example channel # 10 was input.

Line 2 This sets this channel up for a particular input.

Line 3 This line enables or disables this channel.

Line 4 This line asks for the type of rain gauge input. If a D is entered the CPP checks for rainfall ticks from digital input number 1. If a C is entered, the CPP checks the rainfall counter for rainfall ticks.

Line 5 For a tipping bucket type input, the CPP then asks for the value of each tip, either 0.01 inch or 0.1mm.

7.5.2 Analog Rain Gauges

This section addresses an analog input signal where the analog signal represents a tipping bucket signal that has been converted to an analog signal. These type of signals rollover (back to zero) at some predetermined rainfall amount, *e.g.*, 1.00 inch. A weighing type rain gauge, although similar, but different is presented in section 7.9.3 below. In question number 4 above if a W had been entered, the CPP would initialize this channel as accepting an analog signal as an indication of rainfall. In such a unit, the CPP calculates the difference in the analog signal from the start of a preliminary period to the end of a preliminary average. Based on this value the CPP then converts the signal to an equivalent number of 0.01in or 0.1mm counts.

With this type of input, the value entered as a full scale value is used to determine the rainfall amount. The full scale setting should have two decimal places, *e.g.*, 1.00, 5.00, 10.00, and so forth. If the full scale is entered incorrectly, the CPP will adjust the input to the best fix. The CPP will accept two digits to the left of the decimal point. Even with an incorrect full scale value, rainfall amounts can be corrected as the exact conversion factor calculated by the CPP can be determined, and a factor applied to the reported amounts.

A rain gauge calibrated in inches can be used to provide rain fall in millimeters and vice versa. This can be accomplished by setting the full scale value to the corresponding millimeters or inches full scale. For example, for a rain gauge calibrated in inches, with a full scale value of 1.00 inches, to provide readings in millimeters, the full scale value should be entered as 2.54. The CPP will then calculate and report rain fall in millimeters.

This approach has a small precision drawback. The full scale binary reading of the A/D converter in the CPP is 16,000 counts. Using the original rain gauge calculation of 1.00

inches, yields a count of 160 (16000/100 = 160) for each 0.01 inches of rainfall. As can be seen, this is an exact number. This is true for all standard full scales. If, however, a full scale of 2.54 is entered, then the resultant has a remainder, (16000/254 = 62.99). The CPP will round this number to 63. This yields a 0.015% error. For most applications, this is a very acceptable error, which is probably more accurate than the rain gauge. For larger full scale values, such as 5.00 or 10.00 inches, this error increases to 3.15% and 4.98% respectively. For a more exact conversion, the cardinal full scale for the rain gauge should be used and a computed channel set up to make the conversion. In this approach the accuracy is limited to that of the rain gauge.

7.5.3 Weighing Rainfall Inputs

Weighing rainfall instruments provide an analog signal indicating how much rainfall has been accumulated in a container. Similar to the analog input discussed above, the CPP is looking for deltas in the analog signal over a given time frame. Weighing rain gauges may reset (return to a zero voltage input) based on either weight or time. Quite often, based on either time or weight, the rain gauge switches the sample container. This allows the collected rain to be returned to a laboratory for analysis. In some cases, this switch is conducted autonomously by the instrument. In others the data logger needs to initiate the container switch. Reference the above section regarding converting reading between inches and millimeters.

7.6 Direct Met Interfaces

The CPP provides a direct interface to meteorological sensors. Tech note 41 provides full scale and starting threshold numbers for various sensors.

7.6.1 Power

On the back panel the CPP provides regulated ±12 volts and+5 volts to power external devices. Each will supply ¼ Amp.

7.6.2 Wind Direction

The CPP provides four precision 1.00 volt signals to power the potentiometers used in wind direction sensors. Any 1.00 volt reference signal can be used to power three 500 ohm wind direction potentiometers.

7.6.3 Wind Speed

The CPP has three counter inputs allowing interfacing to three wind speed sensors. When using this interface, in the channel initialization, the manufacturer should be set to A/D and the A/D channel should be set to 88. The CPP asks for the wind speed frequency for a 100mph indication. This value is normally available form the documentation supplied with the sensor, or it can be obtained from the sensor manufactures.

The CPP then requests the starting threshold for the wind speed sensor. Both of these numbers must be entered in a scientific format.

8 CPP-4794 Rear Panel & Internal Switch Settings

8.1 General

The rear panel of the CPP-4794 contains a series of slots where additional PC boards can be installed. A few slots are dedicated to a particular PC board, and the remainder can accommodate various types of boards. A picture of the rear panel is presented at the end of this tech note.

To install or replace PC boards reference the section at the end of this tech note.

8.2 Power Supply

Viewed from the rear, the leftmost slot is for the power supply PC board. This board contains the following;

+5v, ±12v supplies
Power output connector
Activity LED
Battery backup
Comm port #4 DB9 and RJ11 front panel connectors
LED indicating comm. port #4 Rx/Tx lines proper
Front/rear panel on/off switches
Fuse
Line power filter and entry

Power Output Connector

Pin	Definition
1	Ground
2	+5v
3	Ground
4	+12v
5	Ground
6	-12v
7	Ground
8	Counter Input

8.3 Controller Board

The next slot is for the controller PC board. This board contains the following

Digital signal processor
Real time clock
ROM/RAM memory
EEPROM for configuration storage
Four serial ports
Rear panel LED indicating proper Tx/Rx serial connections
Dip switch for comm ports # 1 and 4 Baud rate, and battery in/out

The serial ports on this board are designated as comm port #1, comm port #2, comm port #3-0, and comm port #4. Comm port #1 is the port used to communicate with a central processing system. Comm port #2 can be used for a variety of purposes. The most common usage is as a Modbus port or a met broadcast port. Special protocols are provided over this comm. port. Comm port # 3-0 is used to interface to instruments serially. Comm port # 4 is considered the on site operator interface.

8.4 Controller Back panel 10 position Dip switch

Switch positions 1, 2, and 3 control the Baud rate of comm port #4, and switch positions 4, 5, and 6 control the Baud rate of comm port #1, as defined below.

SW2	1	2	3	4	5	6	Baud Rate
CP#4	Off	Off	Off	-	-	-	300 Baud
CP#4	On	Off	Off	-	-	-	1200 Baud
CP#4	Off	On	OFF	-	-	-	2400 Baud
CP#4	On	On	Off	-	-	-	4800 Baud
CP#4	Off	Off	On	-	-	-	9600 Baud
CP#4	On	Off	On	-	-	-	19.2K Baud
CP#4	Off	On	On	-	-	-	38.4K Baud
CP#4	On	On	On	-	-	-	38.4K Baud
CP#1	-	-	-	Off	Off	Off	300 Baud
CP#1	-	-	-	On	Off	Off	1200 Baud
CP#1	-	-	-	Off	On	Off	2400 Baud
CP#1	-	-	-	On	On	Off	4800 Baud
CP#1	-	-	-	Off	Off	On	9600 Baud
CP#1	-	-	-	On	Off	On	19.2K Baud
CP#1	-	-	-	Off	On	On	38.4K Baud
CP#1	-	-	-	On	On	On	38.4K Baud

Rear panel switch positions 7, 8, and 9 are not used. Switch position number 10 switches the internal battery on and off to the internal memory and the real time clock.

Dip switch SW3 is used when converting comm. ports number 1, 2, and 3-0 from RS-232 to RS-422/485, as defined below.

SW3	Definition
1,2,3	On for CP#1 as RS-232 – Off for RS-422/485 operation. Install U36.
4,5,6	On for CP#2 as RS-232 – Off for RS-422/485 operation. Install U45
7,8,9	On for CP#3-0 as RS-232 – Off for RS-422/485 operation. Install U37
10	Used internally at H2NS, should always be off.

Dip switch number SW1 is also used for comm ports number 1 and 3-0. When comm port 1 is set up for RS-422/485 operation, switch position 4 allows the output to be tri-stated. With position 3 off, the RS-422/485 output is always driven. With position 3 on, the output of the RS-422/485 driver is on only when the transmitter is active.

SW1, position 3, allows a similar control of comm port 3-0 when configured as RS-422/485.

8.5 Compact Flash Memory/Printer

The third slot from the left is for the Compact Flash Memory (CFM) interface and the printer output. Compact flash memories do not have a write protect option. Switch position #4 on SW1 provides this feature. Turning this switch on, write protects the CFM. The other switch positions are not used.

The CFM is formatted with the FORMATcx command where x is 1 through 8. If the x is omitted the CFM formats as a 1meg CFM.

8.6 Expanded Comm Ports

The fourth and fifth slots are for the expanded comm. PC boards. Each board accommodates four serial ports. The comm ports are designated comm ports 3-1, 3-2, 3-3, and 3-4 on the first board, and 3-5, 3-6, 3-7, and 3-8 on the second board installed. Comm ports number 3-2 and 3-3 on board number one, and comm ports 3-6 and 3-7 on the second board, can be changed to be RS-422/485 protocols. These outputs of these comm ports are connected to the RJ11 connectors and the eight pin pluggable connectors as well. An on board 10 position Dip switch is used to change these ports from RS-232 to RS-422, as defined below. The switch settings are the same for both boards.

SW1 position 1 selects either board 1, CP# 3-1 through 3-4, or board 2, CP# 3-5 through 3-8. Position 1 off selects board 1, and on selects board 2.

Position 3 allows the transmit output of 3-3 (3-7 on second board) to be controlled with the comm ports RTS signal. On allows the RTS signal to control the output, off enables the output at all times.

Position 10 of SW1 allows the transmit output of 3-2 (3-6 on second board) to be controlled with the comm ports RTS signal. On allows the RTS signal to control the output, off enables the output at all times.

For RS-232 operation of comm port 3-3 (3-7second board) positions 3, 4, 5, and 6 should be on. For RS-422/485 operation they should be open. Reference position 3 above for controlling the RTS signal.

For RS-232 operation of comm port 3-2 (3-6 on second board) positions 7, 8, 9 and 10 should be on. For RS-422/485 operation they should be open. Reference position 10 above for controlling the RTS signal.

For RS-422/485 operation for comm port 3-2 (3-6 on second board) U29 should be installed. For RS422/485 operation for comm port 3-3 (3-7 on second board) U30 should be installed.

The eight-position connector that can be used for RS-422/485 connections is defined below.

Eight Pin Connector	Function
1	Tx2+
2	Tx2-
3	Rx2+
4	Rx2-
5	Tx3+
6	Tx3-
7	Rx3+
8	Rx3-

8.7 Remainder of Slots

The remainder of the slots can be filled with analog and digital I/O PC cards. Since the rear panel labels are generic, the connector definition of each type on board is presented below.

8.8 Analog to Digital Converter (ADC)

The ADC has 16 analog inputs. The ADC supports ±100mv, ±1.0v, 0-5v, 0-10v and 4-20mA inputs. Since the ADC switches both signal and ground, the standard ADC input can handle 4-20mA signals; however, for true 4-20mA isolation an isolated front end should be used. The analog front ends of the ADC are contained on a piggyback PC board the plugs into the ADC main board. A n on board DIP switch is used to set a board as A/D channels 1-16 or 17-32, and to select 50 or 60 Hertz operation. The switch settings are presented below.

		A/D Ch 1-16	A/D Ch 17-32
SW1	1	On	Off
SW1	2	On	Off
SW1	3	Off	On
SW1	4	Off	On

Switch position number 5 is On when only one ADC board is installed in the CPP, and Off when two ADC boards are installed.

SW1	5	On for 1 ADC, Off for 2 ADC’s.
SW1	6	Not used

Switch positions number 7 and 8 are used to select 50 or 60 hertz operation.

SW1	7	On for 60Hz operation
SW1	8	On for 50Hz operation

Dip switch SW4 is used for in house testing. For normal operation positions 1 and 2 should be on and positions 3 and 4 off.

Jumper J2 is used write enable or write disable the EEPROM used for storing calibration constants. This jumper should always be in the disabled position except when the ADC is being calibrated. All other jumpers on the board are used by H2NS in testing routines, and should not have any jumpers installed.

The rear panel connector for the ADC is defined below.

Rear Panel

Pin #	Definition	Pin #	Definition
1	Ch1+	2	Ch1-
3	Ch2+	4	Ch2-
5	Ch3+	6	Ch3-
7	Ch4+	8	Ch4-
8	Ch5+	10	Ch5-
11	Ch6+	12	Ch6-
13	Ch7+	14	Ch7-
15	Ch8+	16	Ch8-
17	Ch9+	18	Ch9-
19	Ch10+	20	Ch20-
21	Ch11+	22	Ch11-
23	Ch12+	24	Ch12-
25	Ch13+	26	Ch13-
27	Ch14+	28	Ch14-
29	Ch15+	20	Ch15-
31	Ch16+	32	Ch16-

8.9 Digital Input Board

The digital input board accommodates 16 digital inputs. The inputs are optically isolated from the CPP, and have input voltage protection and pull up resistors to +5v. A contact closure is sensed as a one, or true by the CPP. The rear panel connector of the digital input board is defined below.
{Lo equates to an isolated ground and Hi equates to a isolated floating +5v.}

Rear Panel

Pin#	Definition	Pin#	Definition
1	Input 01 Hi	2	Input 01 Lo
3	Input 02 Hi	4	Input 02 Lo
5	Input 03 Hi	6	Input 03 Lo
7	Input 04 Hi	8	Input 04 Lo
9	Input 05 Hi	10	Input 05 Lo
11	Input 06 Hi	12	Input 06 Lo
13	Input 07 Hi	14	Input 07 Lo
15	Input 08 Hi	16	Input 08 Lo
17	Input 09 Hi	18	Input 09 Lo
19	Input 10 Hi	20	Input 10 Lo
21	Input 11 Hi	22	Input 11 Lo
23	Input 12 Hi	24	Input 12 Lo

25	Input 13 Hi	26	Input 13 Lo
27	Input 14 Hi	28	Input 14 Lo
29	Input 15 Hi	30	Input 15 Lo
31	Input 16 Hi	32	Input 16 Lo

The CPP will support up 3 digital input boards providing 48 digital inputs. The board selection is made by Dip switch SW1, as defined below.

SW1	Input Selection
1 On	Inputs 1-16
2 On	Inputs 17-32
3 On	Inputs 33-48
4 On	Inputs 49-64 (Not Standard, Optional)

A switch position must be On to enable the board, but only one position can be set on any board. If more than one position is On, there will be interference among the boards.

8.10 Digital Output Board

The digital output board has 16 relays, and provides a contact closure when a one or true. The rear panel connector of the digital output board is defined below. Although there is not a positive or negative signal associated with the relay outputs, for clarity, it is shown this way below.
{the + position refers to the wiper of the relay and the – position refers to the common}

Rear Panel

Pin#	Definition	Pin#	Definition
1	Relay 01+	2	Relay 01-
3	Relay 02+	4	Relay 02-
5	Relay 03+	6	Relay 03-
7	Relay 04+	8	Relay 04-
9	Relay 05+	10	Relay 05-
11	Relay 06+	12	Relay 06-
13	Relay 07+	14	Relay 07-
15	Relay 08+	16	Relay 08-
17	Relay 09+	18	Relay 09-
19	Relay 10+	20	Relay 10-
21	Relay 11+	22	Relay 11-
23	Relay 12+	24	Relay 12-
25	Relay 13+	26	Relay 13-
27	Relay 14+	28	Relay 14-
29	Relay 15+	30	Relay 15-
31	Relay 16+	32	Relay 16-

The CPP will support up 3 digital output boards providing 48 relay outputs. The board selection is made by Dip switch SW1, as defined below.

SW1	Input Selection
1 On	Outputs 1-16
2 On	Outputs 17-32
3 On	Outputs 33-48
4 On	Outputs 49-64 (Not Standard, Optional)

A switch position must be On to enable the board, but only one position can be set on any board. If more than one position is On, there will be interference among the boards.

Relays 15 and 16 can be jumpered for normally closed operation. Jumpers J7 and J10 are used to make this selection.

Jumper J10 controls relay 15. With J10 jumpered pins 1-2, the relay output is a normally open contact. Setting to a true activates the relay and makes the contact closure. With J10 pins 2-3 jumpered, the relay output is a normally closed contact. Setting to a true activates the relay and opens the contact closure output.

Jumper J7 controls relay 16. With J7 jumpered pins 1-2, the relay output is a normally open contact. Setting to a true activates the relay and makes the contact closure. With J7 pins 2-3 jumpered, the relay output is a normally closed contact. Setting to a true activates the relay and opens the contact closure output.

8.11 Direct Met Interface

The direct met interface provides;

Four 1 volt reference outputs
Three wind speed counters (One standard)
Battery backed rainfall counter
Provisions for various thermistor connections

The rear panel connections are as follows:

Pin #	Signal	Pin #	Signal
1	Rainfall Input	17	Voltage Reference #1 Output
2	Ground	18	Ground
3	Wind Speed Input #1	19	Voltage Reference #2 Output
4	Ground	20	Ground
5	Wind Speed Input #2	21	Voltage Reference #3 Output
6	Ground	22	Ground
7	Wind Speed Input #3	23	Voltage Reference #4 Output
8	Ground	24	Ground
9	Thermistor #1 T1	25	+ 5 volts
10	Thermistor #1 T2	26	Ground
11	Thermistor #1 T3	27	Unused
12	Ground	28	+ 12 volts
13	Thermistor #2 T1	29	- 12 volts
14	Thermistor #2 T2	30	Ground
15	Thermistor #2 T3	31	Earth Ground

16	Ground	32	Earth Ground

8.12 Power

The ±12 volts are from and isolated power supply. It will provide only very small amounts of current (5mA), and if shorted to ground will damage the power supply. The ±12 volts available on the power supply board should be used to power external devices.

8.13 Thermistor Input

Some thermistors can be connected to the CPP with a single external resistor. Some require a series of precision resistors. On the direct met interface board are two DIP sockets for installing headers with the resistors installed on the headers. Please feel free to contact H2NS regarding thermistor connection.

8.14 Direct Met Board Selection

A four position dip switch is installed on the direct met interface board. Switch position number one sets the board for board number 1. Switch position number 2 sets the board for board number 2. Two direct met interface boards is an option. For one direct met interface board, switch position number one is on and switch position number 2 is off.

8.15 Wind Speed Counters

The operation of the wind speed counters can be tested using an internal switch setting. Ten position dip switch number SW1 is used for the test feature. Switch positions 1-6 are used to select various frequencies. Switch positions 7, 8, and 9 are used to direct the selected frequency into a counter input. Switch position 10 is used to turn on/off the battery power to the rainfall counter. Switch SW1 is defined below.

Switch SW1

Position	Function
1	Frequency ≈ 4,9 Hz
2	Frequency ≈ 9.8 Hz
3	Frequency ≈ 19 Hz
4	Frequency ≈ 38 Hz
5	Frequency ≈ 76 Hz
6	Frequency ≈ 152 Hz
7	Wind Speed Input #1
8	Wind Speed Input #2
9	Wind Speed Input #3
10	Rainfall Battery On/Off

8.16 Internal Amplifier

The direct met interface board contains an amplifier with a gain of approximately six. For signal amplification, and external amplifier is recommended, however the internal amplifier may prove useful in some applications. To direct the input and output of the internal amplifier, two jumpers are required. Jumper J17 directs rear panel connection Thermistor #2 T1 (Pin #13) to the input of the amplifier. Jumper J18 directs the output of the amplifier to rear panel connection Thermistor #2 T2, (pin #14).

8.17 Internal Modem/Ethernet Interface

To be added.

8.18 Removing/Installing PC Boards

The CPP-4794 has been designed so that PC boards are plugged into a motherboard and attached to the rear panel with a screw. This approach makes maintenance easier and facilitates field expansions. Power should always be shut off prior to removing or installing a PC board. Although not absolutely necessary, board removal and installation are much easier if the rear panel is loosened so that it can lean backwards, and is recommended. This is not as difficult or laborious as it may first sound.

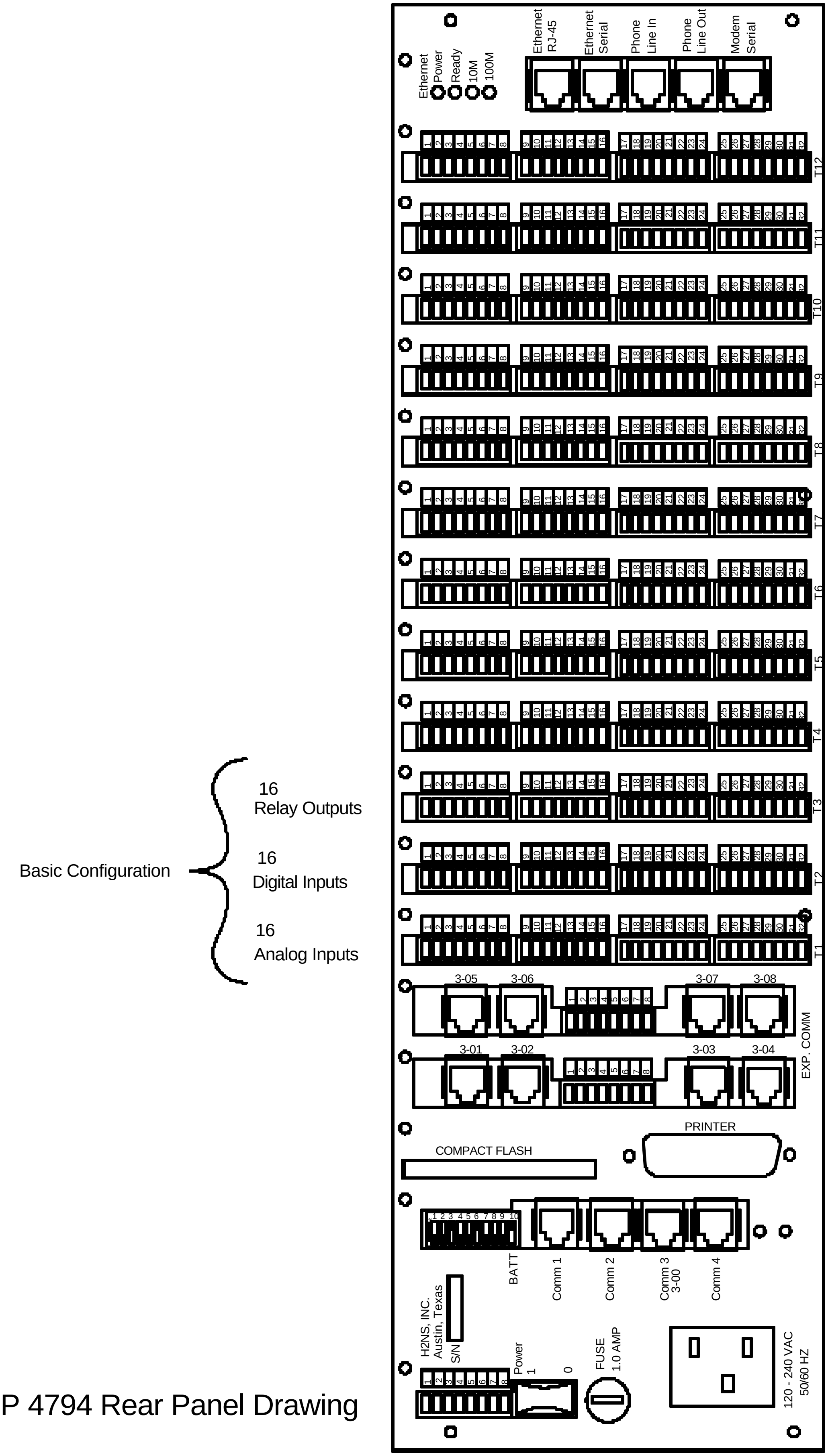
The power supply board is the most difficult to change as it is connected to both the front and rear panels. Anytime the power supply is being removed, the AC power cord should be removed.

The power supply board is connected to the incoming AC power. The AC lines are on etch on the board and run between the front and rear panel on/off switches, which allows dangerous voltages to be exposed when the board is removed. Any maintenance should be conducted only by a qualified technician familiar with high voltages.

The power supply board can be removed and installed without loosing the front panel; however, loosing the front panel is straightforward and facilitates board exchange. The outputs of the power supply can be tested at the rear panel connector defined above.

To loosen the front panel remove the two top screws and loosen all other front panel screws. The two hex nuts on the front panel DB9 connector must also be removed. Because of the connectors on the power supply rear panel, the rear panel must also be loosened so that it can lean backwards. This requires that the screws holding each PC board be removed as well. The power supply board is removed by pulling up from the front edge and sliding the connectors out of the rear panel. Installation is the reverse, position the rear panel connectors and then plug the card into the motherboard and position the front panel over the front panel connectors/switches. Reinstall all screws.

Other boards are removed or installed by removing the rear panel screw and tilting the board rearwards to mate the connectors with the rear panel cutouts and the plugging the board into the motherboard.



CPP 4794 Rear Panel Drawing

8.19 Installing Rack Mounting Brackets

The units are shipped from H2NS with 8x32, 5/16” screws for mounting the brackets to the CPP. The power supply PC board fits closely to the side panel. Although clearance holes have been drill in the power supply where the mounting bracket screws protrude, it is recommended that screws no longer than 5/16” be used.

NOTE: Screws any longer than 5/16 inch may extend inside the chassis and contact with high voltages on the power supply board. If proper length screws are needed please contact H2NS.

9 Warranty and Service

9.1 H2NS Inc. Warranty Policy

Prior to shipment, H2NS Inc. equipment is thoroughly inspected and tested. Should equipment failure occur, H2NS Inc. assures its customers that prompt service and support will be available.

9.2 Coverage

After the warranty period and during the equipment lifetime, H2NS Inc. will provide on-site or in-plant service at reasonable rates similar to those of other manufactures in the industry.

9.3 Non-H2NS Inc. manufactured equipment

Equipment provided but not manufactured and not normally offered by H2NS Inc. is warranted and will be repaired to the extent and according to the current terms and conditions of the respective equipment manufactures warranty.

9.4 General

H2NS Inc. warrants each Product manufactured by H2NS Inc. to be free of defects in material and workmanship under normal use and service for a period of one year.

If a Product fails to conform to the warranty by H2NS Inc. herein, H2NS Inc. shall correct such defect by, at H2NS Inc.'s discretion, repairing or replacing such defective Product or refunding the purchase price of said Product.

The warranties set forth in this section shall be of no force or effect with respect to any Product:(i) that has been altered or subjected to misuse, negligence or accident, or (ii) that has been used in any manner other than in accordance with the instruction provided by H2NS Inc.

THE WARRANTIES SET FORTH IN THIS SECTION AND THE REMEDIES THEREFORE ARE EXCLUSIVE AND IN LIEU OF ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR PARTICULAR PURPOSE OR OTHER WARRANTY OF QUALITY, WERE EXPRESSED OR IMPLIED. THE REMEDIES SET FORTH IN THIS SECTION ARE THE EXCLUSIVE REMEDIES FOR BREACH OF ANY WARRANTY CONTAINED HEREIN. H2NS INC. SHALL NOT BE LIABLE FOR ANY INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF OR RELATED TO THIS AGREEMENT OF H2NS INC.'S PERFORMANCE HEREUNDER WHETHER FOR BREACH OF WARRANTY OR OTHERWISE.

9.5 Terms and Conditions

All units or components returned to H2NS Inc. should be properly packed for handling and returned freight prepaid to:

H2NS Inc.
12108B Roxie Drive
Austin, TX 78729-7138

After the repair, the equipment will be returned, ground freight prepaid, unless other arrangements are made.

Phone: 512-918-8035
Fax: 512-918-9684
Home Page: http://www.h2ns.com
e-mail: sales@h2ns.com