DSM Lookalike

1.0 General

Comm port #1 is used to interface to the outside world. Over this comm port a central polling system can routinely collect data, status and calibration results. A user can also communicate with the CPP and check the units operation. Upon command, the CPP will enter a mode that allows the user to communicate directly with any instrumentation connected over the serial ports.

The CPP supports various types of data transfers. These are listed below.

- DSM Look Alike
- Operator readable
- Comma Delimited, ASCII Data
- Modbus (Optional)

The following sections detail the various operating modes. When interfacing to a CPP, section 9 should be read in its entirety, as all features are simultaneously available. Due to the number of options available over comm port #1, to an unfamiliar user the features offered by the CPP over comm port #1 can be confusing. Please do not hesitate to contact H2NS to clarify questions.

2.0 DSM Lookalike

H2NS offers an output protocol option that emulates that of the Odessa DSM-3260, and the Odessa SPLC series of data loggers. This feature allows Envicom or programs that emulate Envicom to collect and process data in a DSM format.

Two bytes (16 bits) are allocated in the DSM to store each data point. Twelve bits (12) of this is used to represent the data value and the other four bits are used to present status information regarding the data value. One of the twelve data bits is used as a sign bit, leaving eleven bits to represent the data value. Since most sensors employed in air quality systems use only positive voltages, the eleven bits provide 1 part out of 2000, or 0.05% accuracy. One of sixteen bits of status information is prioritized and encoded into the four-bit status nibble.

In contrast, the CPP allocates five bytes (40 bits) to store each data point. Two bytes (16 bits) are used to present sixteen simultaneous status conditions for each data point. The data value is saved in the other three bytes. Data is collected and stored in the CPP in scientific notation, *i.e.*, sign, exponent and four significant digits. This provides 1 part in 10,000 or 0.01% accuracy, a five time's improvement over the DSM.

The CPP uses the entered full scale and zero values to convert this stored data into the twelve-bit binary number that is required by Envicom. Status information stored with each data byte is also prioritized and mapped into a DSM format for transmittal to Envicom. The CPP can control, capture and return calibration results and calculate and return both sigma theta and sigma W calculations. The DSM uses lower case letters as commands to return data strings tagged with identifying characters and terminated with checksums and EOT characters. Table 1.0 below lists the DSM data collection commands supported by the CPP.

Table 1.0
DSM Commands Supported By The CPP

1. t - Time and date/instantaneous status
2. p - Preliminary average status
3. s - Interval average status
4. r - Latest calibration results
5. o - Last ten preliminary averages
6. j - Current hours interim averages
7. k - Previous hours interim averages
8. l - Next previous hours interim averages
9. a - Current final averages
10. b - h Previous final averaging periods
11. > - Time download
12. ? - Date download

3.0 Communications

The CPP is designed to communicate with the central poller over communications port #1 (comm port #1). Due to the number of options available over comm port #1, to an unfamiliar user the features offered by the CPP over comm port #1 can be confusing. Please do not hesitate to contact H2NS to clarify questions.

The CPP option is designed to use a dollar sign, unit ID to gain access to the unit. This can be changed in the initialization from the Main Menu. A rear panel dip switch setting also allows the Baud rate to be set from 300 to 38.4K Baud. The factory defaults are 9600 Baud and an ID of 01H. Table 2.0 below defines the Baud rate settings for comm ports #1 and #4.

The CPP supports all normal features covered in the manual over the local or monitor port comm port #4. In addition, depending on the manufacturer drivers installed in the unit, serial communications can be used to interface to the instrumentation.

Table 2.0
Rear Panel Baud Rate Settings

Switch Position 1 2 3	4 5 6	7 8 9 10	Function	
0 0 0	x x x	x x x x	300 Baud	Comm Port #4 Baud rate
1 0 0	x x x	x x x x	1200 Baud	
0 1 0	x x x	x x x x	2400 Baud	
1 1 0	x x x	x x x x	4800 Baud	
0 0 1	x x x *	x x x x	9600 Baud	
1 0 1	x x x	x x x x	19.2K Baud	
0 1 1	x x x	x x x x	38.4K Baud	
1 1 1	x x x	x x x x	38.4K Baud	

Switch Position 1 2 3	4 5 6	7 8 9 10	Function	
x x x	0 0 0	x x x x	300 Baud	

x	x	x	1	0	0	x	x	x	x	1200 Baud		
x	x	x	0	1	0	x	x	x	x	2400 Baud		
x	x	x	1	1	0	x	x	x	x	4800 Baud	Comm Port #1	
x	x	x	0	0	1*	x	x	x	x	9600 Baud	Baud rate	
x	x	x	1	0	1	x	x	x	x	19.2K Baud		
x	x	x	0	1	1	x	x	x	x	38.4K Baud		
x	x	x	1	1	1	x	x	x	x	38.4K Baud		

Positions number 1, 2, and 3 of switch #1 are decoded to select the baud rate for Comm Port #4 or on site port. Factory default is 9600 baud. Switch positions 4, 5, and 6, are used to set the Baud rate of comm port #1. The factory default for both Baud rates is 9600 Baud.

Switch position number 10 is used to disconnect the battery from the CPP. Switch position number 9 allows comm port #1 to operate in a half duplex, 2 wire mode.

Switch positions number 7 and 8 are not defined.

4.0 Differences Between The CPP And A DSM

A CPP emulating an eight channel DSM, returns the first 8 channels with the little keys. The CPP routinely sends three channels of sigma theta data whether there are any sigma theta channels set up or not. If none are set up the CPP returns zeroes in the data field. If sigma thetas are being calculated, using channels other than the first 8 to calculate the sigma thetas will maximize the channel utility.

When emulating a sixteen channel DSM, the CPP returns channels 1 - 8 in the first data group and channels 11- 18 in the second data group. Channels 9 and 10, and channels 19 and 20 are not returned in the data field. The CPP routinely sends three channels of sigma W data whether there are any sigma W channels set up or not. If none are set up the CPP returns zeroes in the data field.

The CPP allocates two bytes or 16 bits for assigning and storing status. The DSM allocates 4 bits for storing status. Sixteen bits of status information are encoded into the 4 bit nibble. The CPP maps the sixteen independent status bits into an encoded 4

bit nibble to be compatible with the DSM. The priority for this mapping is given below in Table 3.0.

Table 3.0
Mapping CPP Status into DSM Status

CPP	CPP Flag	DSM Flag(H)	DataLink Flag	Effect on Data
Missing	M	B (0)	B	Not used
Good	None	None(8)	None	Used
Pwr Fail	F	F (4)	F	Not used
Bad	B	B (0)	B	Not used
Not All Smps, But > Required	>	None(8)	None	Used
Not All Smps, And < Required	<	< (2)	F	Not used
Downed	D	D (5)	F	Not used
In Cal	C	C (1)	C	Not used
Cal Alarm	*	* (C)	F	Not used
Low Alarm	a	a (B)	-	Used
High Alarm	A	A (A)	+	Used
Status	S	S (3)	F	Not used
Smp Delay	P	P (9)	F	Not used
Holding	H	None (8)	None	Used
Remainder	Several	B (0)	B	Not used

It should be noted that the CPP stores data in a scientific format, *i.e*., number plus exponent. If data is being received from the A/D, then the full scale and zero values set up for each channel are used to calculate the number and exponent. When these stored numbers are being scaled into binary numbers, the full scale and zero values are used to conduct this scaling. If a stored number is too large to be represented in the binary number, then the data point is flagged with a bad status.

5.0 Operator Keyins

The CPP is set up in a regular fashion, over comm port #4, as described in the CPP manual. To make the use of comm port #1 more useful, the CPP also supports a number of operator interactions over comm port #1. In keeping with the DSM format, small characters are used for automatic data collections, and capital characters are used for operator interaction. The functions supported by the CPP are similar to that of the DSM, but the keyins and the data presentation are different-matching that of a standard CPP. Table 4.0 below provides a list of the operator keyins supported by the CPP over comm port #1.

All of the features of comm #4 can be accessed via comm port #1. This is accomplished by entering **SETQ<cr>.** This command internally connects comm port #4 signal lines into comm port #1 signal lines, and comm port #1 takes itself off line.

When in comm port #4, entering **RESQ<cr>** disconnects comm port #4 signal lines form comm port #1 signal lines. If left connected, the signal lines will automatically disconnect in one hour. Detecting a **$** on comm port #1 also disconnects the signal lines. Reference section 3.0 of the CPP manual for a more detailed description of the CPP functions.

Table 4.0
Comm Port #1 Keyins Supported By The CPP

A<cr>	Print the last power fail.
A5<cr>	Print the last 5 power fails.
E<cr>	Print the communications error log.
EC<cr>	Clear the error log.
L<cr>	Print a listing of how the CPP is configured.
PD<cr>	Print the existing digital I/O settings.
PC<cr>	Print the last calibration results.
PI<cr>	Print the current hours interim averages.
PI1<cr>	Print the previous hours interim averages.
PI2<cr>	Print the next previous hours interim averages.
PI3<cr>	Print the next previous hours interim averages.
PF<cr>	Print the current hours final averages.
PF1<cr>	Print the previous hours final averages.
PFx<cr>	Go back x periods and print final averages.
PM<cr>	Print the last 10 preliminary averages. The CPP also responds to **PP<cr>**.
PMxxx<cr>	Print the last **xxx** preliminary averages, where **xxx**
PMxx<cr>	is limited to 180. The CPP also responds to the **PMx,cr>** **PPxxx<cr>** format.
PV<cr>	Print the instantaneous data values.
PVS<cr>	Selects either scientific or engineering print format
SETQ<cr>	Connects Comm #1 signal lines to Comm #4. Any received $ switches the comm ports back.
T<cr>	Print the current time & date.

6.0 Communicating With The Instruments

General

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.

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 **SETQ<cr>.** Then enter **D43<cr>** and the CPP will respond with a list of instruments connected to the unit. Select the channel of the desired instrument and begin communications. An example is provided below.

```
Chn#      Manu          Mdl#          Type
01 =   THERMO       SO2-43C       - SO2
02 =   API          NOX-200A      - NOX
03 =   API          NO2-200A      - NOX
04 =   API          NO- 200A      - NOX
04 =    CLI         WS            - WS
05 =    CLI         WD            - WD

Selection = 2<cr>
02 =  API  NOX-200A  -  NOX  Selected

CMD> ?<cr>
T =  Test Measurement
D =  Diagnostics
W = Warnings
V  = Variables
C  = Calibrations

CMD> = NT? <cr>
#NT01 =  NOx Reading (%T NOX)
#NT02 =  NO2 Reading (%t NO2)
#NT03 =  NO Reading (%T NO)
#NT04  =  Sample Input Range (%T RANGE)
MORE>
```

Note - this is an example, the actual list is much longer. The MORE> indicates that more commands are available. At this point commands can be sent to the instruments as well as from the CMD> prompt. A carriage return input presents more commands. The N is required as it ensures that the system is communicating with an NOx instrument. This would be as S when communicating with an SO2 instrument, an O when communicating with an Ozone analyzer, and would be a C when communicating with a CO Analyzer. This is covered in more detail in the appendix available for each manufacturer from H2NS.

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.

Commands

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

Table 1
Pass Through Commands

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

To pass through to an instrument, select the appropriate comm port, *e.g.* **SET0<cr>.** 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.

7.0 Unit $ID Definitions

The CPP offers a multi faceted interface over comm port #1. The unit will respond to a series of DSM commands, including all commands the are required by ENVICOM to control and collect data from the unit, in a DSM format. These commands are presented in section 2.0. Comm port #1 also offers access to a limited number of the CPP commands. These commands are presented in section 3.0. The number ID given to a CPP is mapped into a small alphabetical character that is used to place the CPP in a computer to computer data transfer mode. Table 5.0, presents the ID mapping.

Table 5.0
CPP $ID Addresses

DSM ID	CPP #2ID
$1	$h
$2	$i
$3	$j
$4	$k

$5	$l
$6	$m
$7	$n
$8	$o
$9	$p
$A	$q
$B	$r
$C	$s
$D	$t
$E	$u
$1	$h

Using the second $ID allows the CPP to respond to a better, more capable set of commands for collecting data from the CPP. The CPP responds to commands and returns data in a comma or space delimited, ASCII format. In addition, CPP configurations can be uploaded and saved into computer files, and then downloaded to this or other CPP's. The data collection commands are available in Tech Note TN19, and a step by step approach to using the upload/download feature is presented in Tech Note TN26. H2NS provides an upload/download program as part of the DataLink software package.

8.0 Envicom Retrieving RamPack Data

The CPP will also emulate a DSM and return data stored in the RamPack or the excess internal memory in a DSM format. To do this the CPP must be set up to record 20 channels of data to the cartridge.