

Interface to Liston Scientific Enviromax Model 2010

1.0 General

H2NS offers a serial interface to the Liston Scientific NDIR gas analyzer, the Enviromax model 2010.

Todd:

Here I need any set up required in the 2010, switch settings, etc, for serial communications. Any instrument ID's, etc.

2.0 Initialization

The CPP is initialized via the operator port, comm port #4. Entering an **I<cr>** results in the CPP bringing up a Main Menu selection. Selecting the channel initialization results in the CPP asking a series of questions as presented below. Operator responses are in **bold**, and **<cr>** means a carriage return. The numbers 1.), 2.) and so forth are used for discussion purposes only and are not printed in an actual system initialization.

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

- 6.) Instrument Manufacturer
 - 00 = Other
 - 01 = API
 - 02 = CLI
 - 03 = Liston
 - 04 = ML
 - 05 = Siemens
 - 06 = Thermo
- 7.) Setting = 00 **3<cr>**

- 8.) Liston Model 2010
 - 01 = Comp
 - 02 = O2

- 9.) Selection = 01 **<cr>**

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

- 12.) Instrument ID = 0000 **<cr>**

- 13.) IMD Installed = 00 <cr>
- 14.) # Points For Interim Avg = 01 04<cr>
- 15.) # Points For Final Avg = 01 9<cr>
- 16.) Unip(U)/Bip(B) = U <cr>
- 17.) Unip(U)/Bip(B) Cals = B <cr>
- 18.) Channel # = <cr>

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. 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, a DCS in a Modbus format) 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, five 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 Liston as the manufacturer. The CPP then prints a list of the model numbers that are supported for the Liston model 2010. Should the instrument have more than one parameter, the second parameter must be set up in another CPP channel on a separate comm port. To monitor oxygen, select parameter number 02.

In Lines 8 & 9, the operator input a **01** which selects the parameter being monitored by the 2010..

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 is nine comm ports.

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 on the CPP back panel. Comm ports 3-01 through 3-08 address comm ports physically located on the expansion printed circuit board 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 Liston 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 to output data in a Modbus format.

In line 11, the CPP is asking what baud rate should be used to communicate with this instrument. This factory default setting is 2400 Baud.

Line 12 asks if an instrument ID is required. An ID is not required with the Liston instruments. It is recommended that an ID not be used. If an ID is set up, Input the appropriate digits ID. In the example, the user did not input an ID.

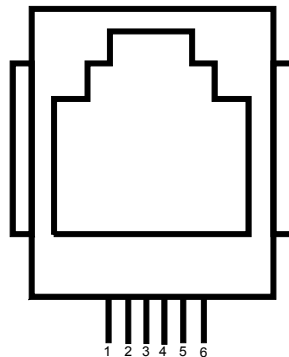
Line 13 is asking if an IMD (Intelligent Multidrop Device) is installed in this comm port cabling. The IMD, manufactured by H2NS, allows a number of different instruments to be connected to a single comm cable. The instruments can have different protocols and even different baud rates. This is very useful 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. The IMD supports addresses from 01 through 60. An address of 00 indicates that an IMD is not installed in this comm cabling.

Lines 14 & 15 are questions regarding data validity qualifiers. The first (line 14) asks how many valid one minute samples the CPP must have to qualify the interim average as valid. Question 14 asks the same in regards to the final average.

Lines 16 & 17 This allows negative data values to either be accepted or set to zero. Usually, negative values are set to zero, except during calibrations.

3.0 RJ-11 Comm Port Connections

H2NS installs a 6 pin, RJ11, telephone jack connector into the CPP rear panel. The majority of applications only use four of the 6 pins, which are pins 2,3,4,5. A null modem can be readily implemented by either twisting or not twisting the telephone cable, which is readily detectable because the cable is color-coded. **Standard, purchased telephone cables are twisted.** Presented below are the pin assignments set up by H2NS for the RJ11 connector.



1 RTS Output

- 2 TX+ Output of CPP
- 3 Gnd (TX- in RS-422/485)
- 4 Gnd (RX- in RS-422/485)
- 5 RX+ Input to CPP
- 6 CD Input

3.1 DB9 to RJ-11 Connectors

H2NS can also provide RS-232 connectors that are terminated in an RJ11 connector. One of the RS-232 connectors can be connected to an instrument and then connected with an RJ11 cable to the CPP. The wire colors for the 9 pin, DB male and female connectors are given below. These colors apply only to the connector delivered by H2NS. Other manufacturers of these type connectors may use a different color scheme.

Pin	9 Pin Male	9 Pin Female
1	Blu	Wht
2	Yel	Blk
3	Grn	Red
4	Red	Grn
5	Blk	Yel
6	Wht	Blu

3.2 Liston DB9 to RJ-11 Connector

The rear panel connector on the Liston instrument s a 9 pin male DB connector, and operates as a DCE device. The proper wiring for the mating 9 pin female DB to RJ-11 is given below. The pin assignments assume that a standard, twisted telephone cable is used. These connectors can be ordered from H2NS.

RJ-11 to 9 Pin DBM to Liston Enviromax

Liston-9 Pin	Color	RJ-11 Pin	RJ-11 Signal/Pin @ CPP
2 Tx	Blk	5	2 – Rx
3 Rx	Yel	2	5 – Tx
5 Gnd	Grn	3	3 – Gnd

4.0 Operation

The CPP polls all connected instrumentation every 15 seconds, and uses these data points to form a one minute average. The one minute averages are used to form subsequent averages. Every 15 seconds the CPP issues the following commands.

- Get data value (e1I)
- Get decimal point (eDV)
- Get Alarms (aGA)

Todd:

The data value & DP seem straightforward. How do I decode the Number received from the aGA command to determine the following:

No Alarms – data value OK – instrument in sample
Data in high alarm
Data in low alarm
Instrument in zero
Instrument in span

I break these out and flag the data values. Any other alarm results in the data value getting flagged with an I indicating a general instrument alarm.

5.0 Calibrations

Todd:

I always think of calibrations in terms of EPA requirements. I believe that your applications do not necessarily need this. I would appreciate your thoughts on best to do this. If we could recommend a consistent approach it would be best. I believe that we can accommodate any approach.

The CPP supports a variety of calibration methods. These are;

- Using the CPP digital outputs (relays) to initiate zero/span cycles
- Detecting incoming digital inputs as zero/span cycles
- Detecting incoming digital inputs to start zero/span cycles
- Using data flags from polls to detect zero/span cycles
- Sending serial commands to instruments to start zero/span cycles
- Combinations of the above

If the CPP is set up to control and initiate the zero/span cycles, then the sequencers in the CPP are configured to set the digital output bits at prescribed times. These bits can be used to switch voltages to solenoids to control the input gas to the model 2010. These bits can also be used to send serial commands to the instrument commanding it to enter the zero mode, the span mode, or to conduct a zero/span cycle.

Do a zero	(cCS0)
Do a span	(cCS1)
Do Both	(cCS2)

6.0 Communicating directly with the Enviromax

The CPP supports two modes in which an operator can communicate directly with the connected instruments. One is an on line mode, in which the CPP interleaves operator and polling commands, and the second is an off line, unabated pass through mode.

6.1 On Line Communications

Via comm port #4, the CPP has a menu that defines the appropriate commands associated with the instrument selected. Via comm port #1, enter SETQ to switch comm port #1 into comm port #4. Input **D43**<cr> and the CPP responds with a list of instrumentation connected to the unit, as shown below;

Chn#	Manu	Mdl#	Type
------	------	------	------

01 = THERMO SO2-43C - SO2
02 = API NOX-200A - NOX
03 = Liston 2010 - Comp

Selection = **3**<cr>
03 = Liston Model 2010 Selected

CMD> **%(e1X)**<cr>

This string, except for the % sign is sent to the instrument configured into channel 3 of the CPP, and then prints the response from the instrument. The CPP sends the string between the % sign and the carriage return <cr>.

The CPP remains on line collecting data even in the pass through mode. Operator commands are interlaced with the CPP normal polling or interrogation commands.

6.2 Off Line Pass Through

The CPP offers an off line, unabated pass through mode to connected instruments. Over comm port #1, a command is entered commanding the CPP to connect one of the comm port to comm port #1 and back off. Information can be passed via the instrument and the user without adhering to any CPP protocol. Manufacturer diagnostics can be conducted remotely. Only the channel being passed through is off line. Data is being collected from all other channels.

More discussion of this is provided in Tech Note 36.

7.0 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**<cr>. After some period of time (two to three minutes) check the error log (**E**<cr>) for errors. In proper operation, there should be no errors. The Error Log lists the following.

No Response	- No response from the instrument to an interrogation
Response Error	- CPP received response but detected an error in the data field
Unknown Resp	- CPP received a response but could not decode the response
Model # Error	- CPP received a response but not from the type of instrument it is configured to interrogate on this channel

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.

The proper cable pin assignments can be determined by plugging the cable into comm port #4. A small green LED by the comm port #4 RJ-11 connector will illuminate if pins 2 & 3 are correct.

8.0 Saving Configurations

The CPP has an on board EEPROM for saving all set up parameters. If a RamPack is installed in the CPP, the configuration can also be saved in the RamPack. This is done as follows.

CFGUL<cr>

This will overwrite any configuration stored in the RamPack

Are you sure Y/N **Y<cr>**

A series of strings will scroll across the screen as the configuration is being stored. This command saves the configuration in the RamPack. If a RamPack is not installed, the CPP will ask;

No RamPack – Save to EEPROM Y/N **Y<cr>**

At the first command presented above, entering an E instead of a Y will result in the configuration being stored in the EEPROM instead of the RamPack.