September 12, 2002

Temet Calcmet Interface

1.0 General

The CPP has a driver for interfacing to a Temet instrument using the Calcmet software. The interface protocol is Modbus RTU. The CPP polls the Calcmet software every fifteen seconds and stores all received data fields. No responses and incorrect responses to polls are logged in an error file. Entering an **E** commands the CPP to print the error file. In normal operation errors should be infrequent. AN **EC** clears the error log. The NR stands for No Response, the AE for Address Error, the CE for command Error, and the NE for a Count Error.

2.0 Connections

The PC should be connected to comm port #3-0 on the CPP/TCI. Parameters are set up and reviewed over comm port #4. The default for both comm ports is 9600 Baud, 8 data bits, no parity and one stop bit. Comm port #1 can be used to collect data using H2NS DataLink software. Comm port #2 can be configured to output data collected in the CPP in a Modbus format.

3.0 Analog Outputs

Using either the internal DAC board or the external DigiMux unit, the digital values received from the Calcmet software can be converted to isolated voltage or 4-20mA signals. The setup procedure for an external DAC is described in the CPP manual. The setup procedure using the internal DAC is presented below. From the Main Menu select the Temet Set Up Option. The CPP responds as follows. Select the analog setup.

```
Analog(A), Digital(D) = A<cr>

DAC#(1-12) = 1<cr>
Par# = 3<cr>

DAC #(1-12) = <cr>

Return to Main Menu
```

Every fifteen seconds, after a successful poll, the data received in the parameter #3 field will be output as an analog signal over DAC channel #1.

4.0 Operator Set of DAC outputs

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

```
OBD 01 50<cr>
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<cr>
DAC# 01 Operator Set Cleared
>
```

5.0 Using Digital Output Bits To Set Analog Output Values

The values output by the 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 Internal DAC Settings. The CPP responds as follows.

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

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.

6.0 CPP Channels to DAC

Any channel's data can also be directed to a DAC channel. This is presented in the CPP manual.

7.0 Digital I/O Mapping

The state of selected I/O bits in the CPP can be mapped into the relay settings of the Temet. In a like manner, the relay settings in the Temet can be mapped into the CPP output bits.

7.1 CPP I/O to Temet

From the Main Menu select the Temet set up. At the first question enter a D, and then at the second question, enter a 2 to select CPP I/O to Temet.

```
Analog(A), Digital(D) = D<cr>
Temet Relays to CPP Output Bits(1), CPP I/O to Temet Inputs(2) = 2<cr>

Into Temet Bit # = 02<cr>
Map CPP Bit # = I01<cr>

Into Temet Bit # = 03<cr>
Map CPP Bit # = O02<cr>

Into Temet Bit # = 05<cr>
Map CPP Bit # = I04<cr>

Into Temet Bit # = <cr>


Return to main Menu
```

The CPP has provisions for 40 bits that can be mapped into the Temet, and are numbered from 1 to 40. Each bit to be mapped can be either an input bit or an output bit of the CPP, but not both. In the first example presented above, CPP input bit number 1 is to be mapped into Temet relay position number 2. Once this is setup, CPP digital output bit number 1 can not be mapped into the Temet because mapping bit position number 1 has already been setup for an input bit. If CPP output bit number 1 needs to be mapped into the Temet, then the Boolean function of the CPP must first be used to map CPP output bit number 1 into another bit position, for example bit number 10. Then bit number 10 can be mapped into the Temet.

Mapping the CPP I/O bits into the Temet works as follows. Once a true CPP I/O bit is detected, the CPP sends the command to the Temet to set the associated input bit set up in the configuration. The CPP sends this command only once, and will not send the command again until the CPP I/O bit resets and again sets.

Entering a Temet bit of 00 or a C clears this particular bit mapping. CPP input bit mapping must use the Ixx to indicate that it is an input bit. Mapping the output bits can include or not include the Oxx indicating and output bit. With no indicator, the CPP assumes and output bit.

7.2 Temet Relays to CPP Output

At the first question enter a D, and then at the second question, enter a 1 to select Temet relays to CPP output bits.

Analog(A), Digital(D) = **D<cr>**
Temet Relays to CPP Output Bits(1), CPP I/O to Temet Inputs(2) =**1<cr>**

Into CPP Output Bit # = **25<cr>**
Map Temet Relay # =**1<cr>**

Temet Relays to CPP Output Bits(1), CPP I/O to Temet Inputs(2) =**<cr>**

Return to Main Menu

This set up is such that anytime the CPP detects that Temet relay #1 is true or set, the CPP sets its output bit #25. Bit #25 will remain set as long as the Temet bit is set. When the Temet bit resets, the CPP bit will also reset.

8.0 Initializing Temet into a standard CPP channel

The sections presented above describe how to set up the CPP to poll the Temet, to convert the received data into analog signals, and to interchange digital I/O bits. To use the data display and logging capabilities of the CPP, the Temet must be set up in the standard initialization of the CPP. This is presented below. From the Main Menu, select Channel Set Up.

Channel # = **01<cr>**
Name = **SO2<cr>**
Units = **ppm<cr>**
Full Scale = **1000<cr>**
Zero = **0<cr>**

Instrument Manufacturer
00 = Other
01 = API
02 = M&C
03 = R&P
04 = Sick
05 = Siemens
06 = Thermo
07 = Temet

Selection = **07<cr>**
Temet Comp#
01 = Comp#01
02 = Comp#02
03 = Comp#03
04 = Comp#04

05 = Comp#05
06 = Comp#06
07 = Comp#07
08 = Comp#08

Selection = **01<cr>**

Comm Port # (3-0 thru 3-8) = 3-0 = **<cr>**

At this point the CPP will ask a series of question regarding Baud rate, bipolar or unipolar data, or questions regarding the USEPA. Simply carriage return through these questions, until the CPP asks for the next channel number. If more Temet parameters are being monitored, they can be initialized into other CPP channels.

Temet channels set up in the CPP will be stored and can be printed out by commands issued to the CPP. Data can also be displayed on the front panel LCD.

9.0 Saving Configurations

After the CPP has been set up as desired, the configuration should be saved in the on board EEPROM. This is done as follows over comm port #4.

CFGUL<cr>
This will overwrite any configuration stored in the RamPack
Continue Y/N **Y<cr>**

No RamPack installed, OK to write to EEPROM Y/N **Y<cr>**

This can take several minutes

A list of data being stored in the EEPROM will scroll across the screen. A prompt will be printed when the transfer is complete.

At any power up the CPP checks the configuration RAM for parity errors. If any are detected, the CPP clears the associated RAM error, and if a configuration has been stored in the CPP, automatically reloads the configuration.

The stored configuration can be downloaded into the CPP with a CFGDL command and a Yes answer to the question.

10.0 Initial Set Up Helpers

Over comm port #4, enter an M keyin. The CPP will print a number of mnemonics and memory address locations. Two that can be of interest are the transmit (TX3BP) and receive (RX3BP) buffers for comm port #3, which is the comm port used to interface to the Temet instrument. Assmue that RX3BP = 8D24 is printed. Entering M 8D24 04 results in the CPP printing four lines of hexadecimal data starting at memory address $8D24_H$. The first two bytes are the receive buffer pointer, and the bytes that follow are the bytes that were last received.

More detail on this feature is presented in the CPP manual.

The CPP also provides as a standard feature two custom screens. The set up of these screens is presented in Appendix C of the CPP manual.

11.0 Operating Parameters

The CPP polls the Temet instrument every fifteen seconds, starting at 4 seconds after the minute. The CPP sends an 04 command to request data, delays 800ms, and then checks the receive buffer for returned data. If data has not been returned the CPP logs the miss in the error log. If data has been returned, the CPP process the returned data field.

The CPP then checks to see if bits are to be mapped between the Temet and the CPP. If so, the CPP processes the set up. If a CPP bit is to be mapped into the Temet, the state of the CPP bit is sent to the Temet every fifteen seconds. The CPP inserts 800ms between transmissions to the Temet. Bit mapping from the Temet to the CPP is processed every fifteen seconds.

Data processed by the CPP is processed every fifteen seconds starting at 00 minutes. This data can be viewed on the front panel LCD. After an hour of inactivity, the CPP turns off the front panel. This can be set up to not turn off if desired.