July 2002

CPP Modbus Interfacing

1.0 General

The CPP offers a Modbus interface for the exchange of analog and digital information. The CPP can be configured as a slave device in either a Modbus RTU or a Modbus ASCII protocol. The CPP operates only as a polled slave remote. The Modbus interface is implemented using comm port #2, which can be configured as an RS-232 or as an RS-422/485 comm port. The factory default is RS-422/485. To change comm port #2 to an RS-232 port reference section 6.0. Section 5.0 describes a table printout that prints the last Modbus queries and responses of the CPP. The following sections describe the CPP/Modbus interface.

Tech Note 35 describes the formats of Modbus RTU, Modbus ASCII, generation of CRC and LRC error check codes, and exception or error responses returned by the CPP.

2.0 Setting Up the CPP

The CPP Modbus interface is set up from the Main Menu. From the Main Menu select the Modbus set up. The CPP responds as follows. The factory defaults are presented in the example. A factory default of 1 stop bit is standard.

Modbus Setup
Address = 000 **10<cr>**
Baud Rate
300(1), 1200(2), 2400(3), 4800(4), 9600(5), 19.2K(6), 38.4K(7) = 5 <**cr>**
Parity
Even(E), Odd(O), None(N) = E **<cr>**
ASCII(A), RTU(R), None(N) = N **R <cr>**
Inactivity Timeout (0-120 Sec) = **40<cr>**
Timeout Action
Reset(R), Set Relay (Rel#) = **40<cr>**

Return to Main Menu

Selecting None in the ASCII/RTU question disables the Modbus interface.

The Modbus inactivity operates as follows. Inputting anything other than zero for the Inactivity Timeout question results in the CPP implementing a watchdog timer on the Modbus communications. Once the CPP responds to a Modbus query, the timer is started with the operator entered time. Each time the CPP responds to the Modbus, the timer is reset to the timeout value. Should the timer be counted to zero (No responses to Modbus queries) the CPP will initiate the set up action. The CPP will either set a relay or execute a power reboot. If a relay is being set, the relay will be automatically reset when Modbus communications resume.

3.0 Command Examples

The commands that will be presented are given in table 1.0 below.

Table 1.0
Modbus commands

Command	Function code
Read coil status	01
Read input status	02
Read holding register	03
Read input register	04
Force single coil	05
Force multiple coils	15
Preset multiple registers	16
Report slave ID	17

3.1 Read Coil Status (0*x* references, coils)

Reads the On/Off status of the discrete outputs, (Digital Outputs). This command is uses to read the present setting of the coils in the unit. Addresses 00000-00047 correspond to coils 1 … 48.

Query

The query message specifies the starting coil and quantity of coils to be read. Coils are addressed starting at zero-coils. Sixteen relays are addressed as 0 … 15. Here is an example of a query to read relays 20 … 56 from slave device 17:

Field Name	Example	Example with CRC
Slave Address	11	01
Function Code	01	01
Starting Address Hi	00	00
Starting Address Lo	13	00
Number of Points Hi	00	00
Number of Points Lo	25	28
CRC Characters	LRC or CRC	3C14

Response

The coil status in the response message is packed as one coil per bit of the data field. Status is indicated as:1 = On; 0 = Off. The LSB of the first byte contains the coil addressed in the query. The other coils follow toward the high order end of the byte, and from low order to high order in subsequent bytes.

If the returned coil quantity is not a multiple of eight, the remaining bits in the final data byte will be padded with zeroes (toward the high order end of the byte). The Byte Count field specifies the quantity of complete bytes of data. Here is an example of a response to the query.

Field name	Example
Slave Address	11
Function Code	01
Byte Count	05
Data (coils 27…20)	CD
Data (coils 35…28)	6B
Data (coils 43…36)	B2
Data (coils 51…44)	0E
Data (coils 56…52)	1B
Error Check	LRC or CRC

The status of coils 27 … 20 is shown as the byte value CD_H, or binary 1100 1101. Coil 27 is the MSB of this byte, and coil 20 is the LSB. Left to right the status of coils 27 … 20 is On-On-Off-Off-On-On-Off-On.

By convention, bits within a byte are shown with the MSB to the left, and the LSB to the right. Thus the coils in the first byte are 27 …20, from left to right. The next byte has coils 35 … 28, left to right. As the bits are transmitted serially, they flow from LSB to MSB: 20 …27, 28 …30, and so on.

In the last data byte, the status of coils 56 …52 is shown as the byte value $1B_{H,}$ or binary 0001 1011. Coil 56 is in the fourth bit position from the left, and coil 52 is the LSB of this byte. The status of coils 56 … 52 is On-On-Off-On-On. The remaining three bits (towards the high order end) are zero filled.

3.2 Read Input Status (1*x* references)

Reads the On/Off status of the discrete inputs, (Digital Inputs). This command is used to read the incoming digital inputs. Addresses 10000-10015 correspond to inputs number 1 … 16.

Query

The query message specifies the starting input and quantity of inputs to be read. Inputs are addressed starting at zero-inputs. Sixteen inputs are addressed as 0 … 15. Here is an example of a query to read inputs 10009 … 10016 from slave device 17:

Field Name	Example
Slave Address	11
Function Code	02
Starting Address Hi	00
Starting Address Lo	08
Number of Points Hi	00
Number of Points Lo	08
CRC Characters	LRC or CRC

Response

The input status in the response message is packed as one input per bit of the data field. Status is indicated as:1 = On; 0 = Off. The LSB of the first byte contains the input addressed in the query. The other inputs follow toward the high order end of the byte, and from low order to high order in subsequent bytes.

If the returned input quantity is not a multiple of eight, the remaining bits in the final data byte will be padded with zeroes (toward the high order end of the byte). The Byte Count field specifies the quantity of complete bytes of data. Here is an example of a response to the query.

Field name	Example
Slave Address	11
Function Code	02
Byte Count	01
Data (inputs 10016 …100009)	CD
Error Check	LRC or CRC

The status of inputs 16 … 09 is shown as the byte value CD_H, or binary 1100 1101. Input 16 is the MSB of this byte, and input 09 is the LSB. Left to right the status of coils 16 … 09 is On-On-Off-Off-On-On-Off-On.

By convention, bits within a byte are shown with the MSB to the left, and the LSB to the right. Thus the inputs in the byte are 16 …09, from left to right.

3.3 Read Channel Values from CPP

The Read Holding Registers (3*x* references) is used to implement this function. The CPP responds to a Read Holding Registers (03) query to channel values. The CPP channels are mapped into the registers one to one. Channel #1 of the CPP is transmitted in register zero, CPP channel #2 is transmitted in register one, and so on. CPP channels 1 … 40 are addressed as registers 0 . . . 39 respectively. The values returned with this request, starting at register zero, are the latest instantaneous values.

Query

An example of a request to read CPP channels # 1-4 (registers 0-3) is presented below.

Field Name	Example
Slave Address	01
Function	03
Starting Add Hi	00
Starting Add Lo	00
Number Reg’s Hi	00
Number Reg’s Lo	04
Error Check	CRC or LRC

Response

The register data in the response message are packed as two bytes per register, with the binary counts right justified. For each register, the first byte contains the high-order bits and the second contains the low order bits. Data is returned as a signed, two’s complement binary number. Full scale equals 16,384 counts. The binary number represents percent of the full scale value set up in the CPP. The CPP can be set up channel by channel to always set negative values to zero, or to set negative values to zero except during calibrations, or never set negative values to zero.

Field Name	Example
Slave Address	01
Function	03
Byte count	08
Data Hi Ch#1	1A
Data Lo Ch#1	DC
Data Hi Ch#2	00
Data Lo Ch#2	0B
Data Hi Ch#3	05
Data Lo Ch#3	C0
Data Hi Ch#4	20
Data Lo Ch#4	00
Error Check	CRC or LRC

CPP Ch #	Register Reference	Value (Hex)	%Full Scale
1	30000	1ADC	41.97
2	30001	000B	00.07
3	30002	05C0	8.98
4	30003	2000	50.00

3.4 Data Status

The Read Input Registers (4*x* references) is used to implement this function. The CPP responds to a Read Input Registers (04) query to transmit data status values. The CPP channel status bits are mapped into the registers one to one starting at register number 00. The status of channel #1 of the CPP is transmitted in register number 00, CPP status of channel #2 is transmitted in register 01, and so on. The status of channels 1 … 40 are addressed as registers 00 . . . 39 respectively. The status values returned with this query, starting at register 00, are the status values associated with the instantaneous analog values.

Query

An example of a request to read the status of channels # 1-4 is presented below.

Field Name	Example
Slave Address	01
Function	04
Starting Add Hi	00
Starting Add Lo	00
Number Reg’s Hi	00
Number Reg’s Lo	04
Error Check	CRC or LRC

Response

The register data in the response message are packed as two bytes per register, with the binary counts right justified. For each register, the first byte contains the high-order bits and the second contains the low order bits.

Field Name	Example
Slave Address	01
Function	04
Byte count	08
Status Hi Ch#1	80
Status Lo Ch#1	00
Status Hi Ch#2	50
Status Lo Ch#2	00
Status Hi Ch#3	10
Status Lo Ch#3	00
Status Hi Ch#4	80
Status Lo Ch#4	10
Error Check	CRC or LRC

CPP Ch #	Register Reference	Value (Hex)	Meaning
1	30040	8000	Good
2	30041	5000	Bad - Off Line
3	30042	1000	Bad - Inst no Resp
4	30043	8010	Good - Holding

Status Bit definitions

Section 7.0 provides definitions of the status bits stored with data values.

3.5 Read Preliminary Values

Reference section 3.3 Read Channel values From CPP. With a starting address of 32,768 (8000_H), the last preliminary average can be retrieved from the CPP

3.6 Read Preliminary Status

Reference section 3.4 Read Data Status. With a starting address of 8000_H, the status associated with the last preliminary values can be retrieved.

3.7 Read Interim Values

Reference section 3.3 Read Channel values From CPP. With a starting address of 36,864 (9000_H), the last interim average can be retrieved from the CPP

3.8 Read Interim Status

Reference section 3.4 Read Data Status. With a starting address of 9000_H, the status associated with the last final interim values can be retrieved.

3.9 Read Final Values

Reference section 3.3 Read Channel values From CPP. With a starting address of 40,960 ($A000_H$), the last final average can be retrieved from the CPP

3.10 Read Final Status

Reference section 3.4 Read Data Status. With a starting address of $A000_H$, the status associated with the last final values can be retrieved.

3.11 Read Calibration Values TBD

3.12 Force Multiple Coils (0F reference)

Forces each coil in a sequence of coils to either On or Off. This is used to set and reset the Digital Outputs. The addresses 00000-00047 correspond to coils number 1 … 48.

Query

The query message specifies the coil references to be forced. Coils are addressed starting at zero-coil 1 is addressed as 0. The requested On/Off states are specified by contents of the query data field. A logical 1 in a bit position of the field sets the corresponding coil On. A logical 0 turns the coil Off.

The following shows an example of a query to force a series of ten coils starting at coil 1 (addressed as 0, 0r 00_H) in slave device 01. The query data contents are two bytes: CD_H 01_H (1100 1101 0000 0001 binary). The binary bits correspond to the coils in the following way:

Bit	1 1 0 0 1 1 0 1 0 0 0 0 0 0 0 1
Coil	8, 7, 6, 5, 4, 3, 2, 1 16,15,14,13,12,11,10, 9

The first byte transmitted (CD_H) addresses bits 8 … 1, with the least significant bit addressing the lowest coil. The next byte transmitted (01_H) addresses coils 10 and 9, with the least significant bit addressing coil 9 in this set. Unused bits in the last data byte should be zero filled.

Field Name	Example
Slave add	11
Function	0F
Coil add Hi	00
Coil add Lo	00
# Coils Hi	00
# Coils Lo	0A
Byte count	02
Force Coil 8 … 1	CD
Force Coil 16 … 9	01
Error Check	CRC

Response

The normal response returns the slave address, function code, starting address, and quantity of relays forced. Here is an example of a response to the above query.

Field Name	Example
Slave add	11
Function	0F
Coil add Hi	00
Coil add Lo	00
# Coils Hi	00
# Coils Lo	0A
Error Check	CRC

If a query is received that tries to set a coil above the number supported by the CPP, the following error code is returned.

Error Response

Field Name	Example
Slave add	17
Function	8F
Error code	03
Error Check	CRC

The 03 indicates an illegal data value in the query data field.

3.13 Preset Multiple Registers

TBD

3.14 Report Slave ID

Returns a description of the unit present at the slave address, along with the units capabilities.

Query

Here is an example of a request to report the ID and status of slave address 01.

Field Name	Example
Slave Address	01
Function	11
Error Check	CRC or LRC

Response

The response can take on several lengths and formats depending on the slave unit. The formats of normal responses for H2NS products are shown below.

CPP-2001

Field Name	Example	
Slave Address	01	
Function	11	
Byte Count	05	
Type	04	(CPP-2001)
Prog Version Hi	61	
Prog Version Lo	01	
Number Chn’s	30	
Chn’s Enabled	20	
Error Check	CRC or LRC	

CPP-3794

Field Name	Example	
Slave Address	01	
Function	11	
Byte Count	05	
Type	08	(CPP-3794)
Prog Version Hi	71	
Prog Version Lo	01	
Number Chn’s	30	
Chn’s Enabled	20	
Error Check	CRC or LRC	

DigiMux

Field Name	Example
Slave add	11
Function	11
Byte count	07
Type	02 (DigiMux)
Digital I/O Hi	00
Digital I/O Lo	FE
Analog Out Hi	F0
Analog Out Lo	00
Analog In Hi	0C
Analog In Lo	00
Error Check	CRC

The slot positions in the DigiMux are numbered 1 … 16, which corresponds to 0 … 15 in the Modbus addressing scheme. The information returned above provides the slot number that each type of DigiPak is installed.

In the above example;

The digital I/O modules are installed in slot positions 8, 7, 6, 5, 4, 3, and 2, corresponding to the FE_H in the LSB. This corresponds to addresses 00001 … 00007. The analog output modules are installed in slot positions 16, 15, 14, and 13, corresponding to $F0_H$ in the MSD. This corresponds to addresses 00015 …00012 Analog output modules are installed in slot positions 12 and 11 which corresponding to the 0CH in the MSD. This corresponds to addresses 00011 and 00010.

3.15 Force Single Coil

This query allows a single coil to be set and reset.

Query

The query message specifies the coil reference to be set or reset. The On/Off state is specified in the query field. A value of FF 00 hex sets the coil and a value of 00 00 hex resets the coil.

Field Name	Example
Slave add	11
Function	05
Coil add Hi	00
Coil add Lo	02
Force data Hi	FF
Force data Lo	05
Error Check	CRC

Response

The normal response is a return of the query.

4.0 Exception responses

Except for broadcast messages, when a master device sends a query to a slave device it expects a normal response. One of four possible events can occur from the master's query:

- If the slave device receives the query without a communication error, and can handle the query normally, it returns a normal response.

- If the slave does not receive the query due to a communication error, no response is returned. The master program will eventually process a timeout condition for the query.

- If the slave receives the query, but detects a communication error (parity, LRC, or CRC), no response is returned. The master program will eventually process a timeout condition for the query.

If the slave receives the query without a communication error, but cannot handle it (for example, if the request is to read a non-existent relay or register), the slave will return an exception response informing the master of the nature of the error.

5.0 Last Responses

The CPP stores the last query and its last response sent to a master. This is printed out with a LR command over comm port #4. An example is presented below

LR<cr>

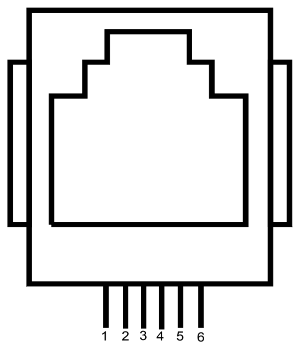
```
                    Modbuss – RTU
     Last received
Query         - 0101000000283C14
Rd Coils      - 0101000000283C14
Rd Status     -
Rd Holding    -
Set Coils     -

     Last Returned
Error         -
Rd Coils      - 01010500000000009152
Rd Status     -
Rd Holding    -
Set Coils     -

ASCIEC - 00
LRC Clears
>
```

This table could have an entry on each line, depending on what commands have been received since the table was last cleared.

6.0 Setting up Comm Port #2 as RS-232/RS-422/485

The serial interfaces (RS-232) of the CPP are offered out via RJ-11 telephone connectors. All serial interface signals are internally protected against static discharges. Table 6.1 below defines the signal pin assignments for the RJ-11connectors. The pins are defined looking into the RJ-11 connector where the mating plug is inserted.

Table 6.1
Pin Definitions for RJ-11 Serial 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

Table 6.2 below defines the switch settings for RS-232 and RS-422/485 outputs.

Table 6.2
Switch #2 Settings for RS-232/422/485

Pos	RS-232	RS-422/485	
1	On*	Off	Comm Port #1
2	On*	Off	
3	On*	Off	
4	On	Off*	Comm Port #2
5	On	Off*	
6	On	Off*	
7	On*	Off	Comm Port #3-0
8	Off*	Off	
9	Off*	Off	
10	Off*	Off	Factory Testing Only

* Factory Settings

To switch comm port #2 from RS-422/485 to RS-232, remove the LTC491 in socket #U45, and close switch positions number 4, 5, and 6 on switch #2.

7.0 Status Bit Definitions

Definition of Binary Status Codes

A definition of the binary status codes is presented below.

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 = Holding		
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

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 condition it will not be reset by the other.

A good data point has status value of 8000_H. More than one status condition may apply to any data point.

Bits 2, 1, and 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.