ISEP

Integrated Systems Engineering & Products

PROTOCOL CONVERTER

MODBUS RTU TO BACNET

Technical Manual

DOCUMENTED BY
ISEP

Updated: 8 April 2002

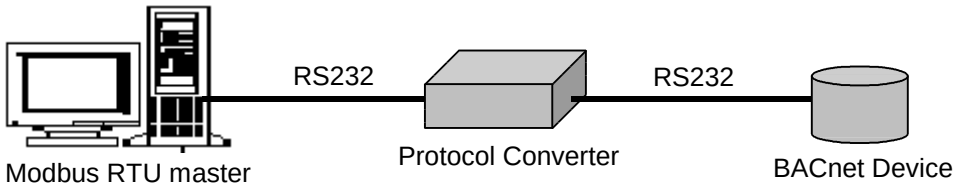
Table of Content

1 Introduction

The MODBUS RTU to BACnet Protocol Converter by ISEP facilitates data communications between a MODBUS RTU master and a BACnet device. It transmits BACnet requests to the BACnet device continuously. Upon receipt of a BACnet response, the converter decodes it and stores the values into the internal data buffers. It waits for MODBUS RTU queries and return a response encoded with values in data buffers.

2 System Configuration

In the monitoring system, a BACnet device is monitored. A PC is used as the MODBUS RTU master. The BACnet device is the slave device. One RS232 port of the converter is connected to MODBUS RTU master and the converter responds to MODBUS RTU queries as a slave. The other RS232 port is connected to BACnet device and the converter send BACnet requests as a master. The system configuration is shown below.



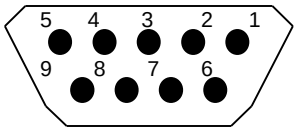
3 Converter Serial Port Description

The converter is designed to sit on a shelf or platform.

One RS232 channel is connected to the Modbus device and the other RS232 channel is connected to BACnet device. The configuration of both channels is shown below.

Parameters	Baud Rate	Data bit	Stop bit	Parity
RS232 channel	9600 bps	8	1	None
RS485 channel	9600 bps	8	1	None

The pin description for the 9 pin D connector is shown below. The configuration of BACnet port and MODBUS port are the same.



9 Pin Female Connector	
Pin #	Description
2	Tx
3	Rx
5	Ground
rest	No connection

4 Overview of Communication Process

This section provides an overview of the communication processes that takes place within the BACnet device, the MODBUS RTU master and the protocol converter.

On power up, the converter will send the seven character ANSI X3.4 trigger sequence “BACnet<CR>”, where “<CR>” denotes the ANSI X3.4 character X’0D’, to inform the BACnet device that it wishes to establish a BACnet PTP connection. The BACnet device will then send a Connect Request frame. The converter will respond by transmitting a Connect Response frame.

Upon completion of above procedure, the converter will send a Heartbeat XON frame to the BACnet device to inform that the converter is ready to receive Data Frames (the details about Data Frames are shown in part 4). The converter will wait a Heartbeat XON frame from the BACnet device. Upon receipt of Heartbeat XON frame, the converter knows the BACnet device is ready to receive Data Frames. The converter will send 3 requests that are Data Frames sequentially to request data. After a request is transmitted, a time counter is started and the converter will wait for a response for this request. The BACnet device should send a data acknowledgement frame to inform the converter that it have received the request. Then, it should send a Data Frame encoded with requested data. On receipt of this response, data are decoded and values are stored into internal data buffer. If no response is received from the BACnet device the time counter is expired and the converter will abort the communication process of the current request and send the next request. The maximum waiting time for the response is currently set to 100 millsesecond. If the converter sends requests without receiving response for more than 1 min , the timeout bit (Refer the Data array Maps for Modbus RTU) will go high (1) . Once the converter receives any message from Bacnet the timeout bit will go low(0).

If the converter finds the BACnet device is disconnected, it will restart the whole communication process described above. Successful receipt of the response with requested data will reset the failure counter to zero. Values stored in data buffer are updated continuously due to sending requests to the BACnet device continuously.

On the MODBUS RTU side, the valid request is Read Input Table , Read Registers, Force Single Output and Preset Multiple Registers . The converter will ignore any queries other than those stated above. The converter will monitor the MODBUS RTU port continuously for queries. It will only respond to messages with the function codes 2 ,3 ,5 and 16 that correspond to the request mentioned above. Upon receipt of a valid request, the converter will generate a response with values stored in buffer and transmits it to the MODBUS RTU master.

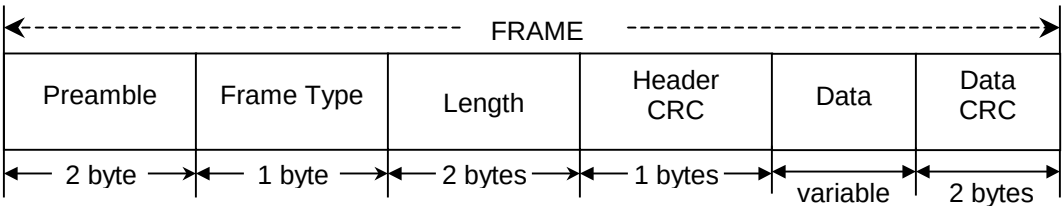
For both communication ports of the converter, the converter will do the error check first after a packet is received. The CRC are used for error check. If a response from the BACnet device with errors, the converter will ignore this response and send next request. If a query with error is received from the MODBUS RTU master, this query will be ignored and the converter will wait for next query.

If the converter can't respond to the MODBUS RTU query, power off the converter and power up again. The converter will start the communication process described above automatically.

5 Description of BACnet Protocol

The BACnet device is the slave device and the converter issues requests as a master. Upon receipt of an error-free frame, the BACnet device responds with corresponding data. They will be decoded and stored by the converter.

This section describes the various message formats and definition of the CLEANSORB protocol used in protocol converter. All codes are in ASCII format. The message fields for a typical message are shown below:



Preamble: two bytes preamble X'55FF'

Length: length of data field no including CRC, most significant byte first

Data: varies with frame type

Data CRC: (if data is present) least significant byte first

The Preamble, Frame Type, Length, and Header CRC are collectively known as the header segment of the frame. The Data and Data CRC are collectively known as the data segment of the frame. The Frame Type is used to distinguish between different types of MAC frames. All defined types are shown below. Details of some frame types are also shown.

Frame Type	Description	Frame Type	Description
X'00'	Heartbeat XOFF	X'09'	Data Nak 1 XOFF

X‘01’	Heartbeat XON	X‘0A’	Data Nak 0 XON
X‘02’	Data 0	X‘0B’	Data Nak 1 XON
X‘03’	Data 1	X‘0C’	Connect Request
X‘04’	Data Ack 0 XOFF	X‘0D’	Connect Response
X‘05’	Data Ack 1 XOFF	X‘0E’	Disconnect Request
X‘06’	Data Ack 0 XON	X‘0F’	Disconnect Response
X‘07’	Data Ack 1 XON	X‘14’	Test Request
X‘08’	Data Nak 0 XOFF	X‘15’	Test Response

Description	Preamble		Frame Type	Length		Header CRC
X’00’	0x55	0xFF	0x00	0x00	0x00	0xEE
X’01’	0x55	0xFF	0x01	0x00	0x00	0x76

Heartbeat Frames X’00’ and X’01’: A frame of one of these types is transmitted by each device periodically when no other data are ready to transmit, to indicate to the peer device that the data link is still active. They contain no data segment. A type X’00’ frame is transmitted to indicate to the peer device that the local device is not ready to accept Data frames. A type X’01’ frame is transmitted to indicate readiness to receive Data frames.

Description	Preamble		Frame Type	Length		Header CRC	Data	Data CRC	
X’02’	0x55	0xFF	0x02	?	?	?	?	?	?
X’03’	0x55	0xFF	0x03	?	?	?	?	?	?

Data Frames X’02’ and X’03’: A frame of one of these types is transmitted to convey data (NPDUs) to the peer device. The length of the data field of a Data frame may range from 0 to 501 types. Successive transmissions alternate frame types.

Description	Preamble		Frame Type	Length		Header CRC
X’04’	0x55	0xFF	0x04	0x00	0x00	0x88
X’05’	0x55	0xFF	0x05	0x00	0x00	0x10
X’06’	0x55	0xFF	0x06	0x00	0x00	0xBB
X’07’	0x55	0xFF	0x07	0x00	0x00	0x23

Data Ack Frames X'04' through X'07': A frame of these types is transmitted to acknowledge a correctly received Data frame. They contain no data segment. Frame types X'04' and X'06' acknowledge receipt of Data frames whose frame type is X'02'. Frame types X'05' and X'07' acknowledge receipt of Data frames whose frame type is X'03'. Frame types X'04' and X'05' indicate that the device is not ready to receive additional Data frames (XOFF). Frame types X'06' and X'07' indicate that the device is ready to receive additional Data frames (XON).

Data Nak Frames X'08' through X'0B': A frame of one of these types is used to reject and incorrectly received Data frame. These frames are not implemented in our protocol converters. So the details of message fields are not shown here.

Description	Preamble		Frame Type	Length		Header CRC
X'0C'	0x55	0xFF	0x0C	0x00	0x00	0x44

Connect Request Frame X'0C': The Connect Request frame is issued by answering device in an attempt to establish a BACnet connection. It contains no data segment.

Description	Preamble		Frame Type	Length		Header CRC
X'0D'	0x55	0xFF	0x0D	0x00	0x00	0xDC

Connect Response Frame X'0D': The Connect Response frame is issued by a device in response to a received Connect Request frame.

Description	Preamble		Frame Type	Length		Header CRC
X'0E'	0x55	0xFF	0x0E	0x00	0x01	0x89
	Data	Data CRC		Disconnect Reason		
	0x00	0x78	0xF0	No more data needs to be transmitted.		
	0x01	0xF1	0xE1	The peer process is being preempted		
	0x02	0x6A	0xD3	The received password is invalid		
	0x03	0xE3	0xC2	Other		

Disconnect Request Frame X'0E': The Disconnect Request frame may be issued by either device when it wishes to discontinue the BACnet PTP dialogue. The data field of the frame conveys the reason for requesting a disconnect and shall be one byte in length.

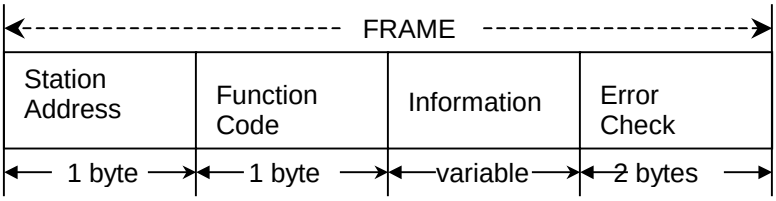
Description	Preamble		Frame Type	Length		Header CRC
X'0F'	0x55	0xFF	0x0F	0x00	0x00	0xEF

Disconnect Response Frame X'0F': The Disconnect Response frame is used to acknowledge a previously received Disconnect Request frame. The disconnect Response frame indicates that the responding device accepts the request to disconnect. It contains no data segment.

6 Description of MODBUS RTU

RTU protocol is a query – response protocol for communication between an RTU device and a host computer capable of communicating using protocol. The host computer is the master device and it transmits a query to a RTU slave, in this case the protocol converter, which responds to the master. The protocol converter as the RTU slave is not allowed to query but only respond to the master.

The RTU data transferred consists of 8-bit binary characters with an optional parity bit. No control characters are added to the data block; however, an error check (Cyclic Redundancy Check) included as the final field of each query and response to ensure accurate transmission of data. The message fields for a typical message are shown below:



Only 3 functions are implemented in this protocol converter. The details of queries and responses are shown below.

6.1 Cyclic Redundancy Check (CRC-16) Calculation

The pseudo code for calculation of the CRC-16 is given below.

Preset byte count for data to be sent.

Initialize the 16-bit remainder (CRC) register to all ones.

XOR the first 8-bit data byte with the high order byte of the 16-bit CRC register. The result is the current CRC.

INIT SHIFT: Initialize the shift counter to 0.

SHIFT: Shift the current CRC register 1 bit to the right.

Increment shift count.

Is the bit shifted out to the right (flag) a 1 or a 0?

If it is a 1, XOR the generating polynomial with the current CRC.

If it is a 0, continue.

Is shift counter equal to 8?

If NO, return to SHIFT.

If YES, increment byte count.

Is byte count greater that the data length?

If NO, XOR the next 8-bit data byte with the current CRC and go to INIT SHIFT.

If YES, add current CRC to end of data message for transmission and exit.

6.2 Read Input Table Query

This query is used to get the binary input and output variable values stored in data buffer.

Byte Number	Field
0	Address
1	Function Code(2)
2	Starting Point No. *Hi Byte*
3	Starting Point No. *Lo Byte*
4	CRC-16 *Hi Byte*
5	CRC-16 *Lo Byte*

6.3 Read Input Table Response

Byte Number	Field
0	Address
1	Function Code(2)
2	Byte Count
3 to 3+ByteCount -1	*Data*
3+Byte Count	CRC-16 *Hi Byte*
4+Byte Count	CRC-16 *Lo Byte*

6.4 Read Registers Query

This query is used to get the analog input and output variable values stored in data buffer.

Byte Number	Field
0	Address
1	Function Code(3)

2	Starting Register No. *Hi Byte*
3	Starting Register No. *Lo Byte*
4	Number of Registers *Hi Byte*
5	Number of Registers *Lo Byte*
6	CRC-16 *Hi Byte*
7	CRC-16 *Lo Byte*

6.5 Read Registers Response

Byte Number	Field
0	Address
1	Function Code(3)
2	Byte Count
3 to 3+ByteCount –1	*Register Hi Byte*
4 to 4+ByteCount –1	*Register Lo Byte*
4+Byte Count	CRC-16 *Hi Byte*
5+Byte Count	CRC-16 *Lo Byte*

6.6 Force Single Output Query

This query is used to write the binary output values into the system.

Byte Number	Field
0	Address
1	Function Code(5)
2	Point No. *Hi Byte*
3	Point No. *Lo Byte*
4	Data *Hi Byte*
5	Data (always zero) *Lo Byte*
6	CRC-16 *Hi Byte*
7	CRC-16 *Lo Byte*

6.7 Force Single Output Response

Byte Number	Field
0	Address

1	Function Code(5)
2	Point No. *Hi Byte*
3	Point No. *Lo Byte*
4	Data *Hi Byte*
5	Data (always zero) *Lo Byte*
6	CRC-16 *Hi Byte*
7	CRC-16 *Lo Byte*

6.8 Preset Multiple Registers Query

This query is used to write the analog output values in the data buffer.

Byte Number	Field
0	Address
1	Function Code(16)
2	Starting Register No. *Hi Byte*
3	Starting Register No. *Lo Byte*
4	Number of Registers *Hi Byte*
5	Number of Registers *Lo Byte*
6	Byte Count
7	Data
8	CRC-16 *Hi Byte*
9	CRC-16 *Lo Byte*

6.9 Preset Multiple Registers Response

Byte Number	Field
0	Address
1	Function Code(16)
2	Starting Register No. *Hi Byte*
3	Starting Register No. *Lo Byte*
4	Number of Registers *Hi Byte*
5	Number of Registers *Lo Byte*
6	CRC-16 *Hi Byte*
7	CRC-16 *Lo Byte*

7 Data Array Maps for the MODBUS RTU master

Analog Input & Output Present Values

Starting Address	900
Data Array Offset	Description
0	AIP 1: CH-1 CHWS Temp
2	AIP 2: CH-1 CHWR Temp
4	AIP 3: CH-1 CWR Temp
6	AIP 4: CH-1 CWS Temp
8	AIP 5: CH-1 % RLA
10	AIP 6: CH-1 Evap Refrg Pressure
12	AIP 7: CH-1 Cond Refrg Pressure
14	AIP 8: CH-1 Comp Disch Rdfrg Temp
16	AIP 9: CH-1 Evap Refrg Temp
18	AIP 10: CH-1 Cond Refrg Temp
20	AIP 11: CH-2 CHWS Temp
22	AIP 12: CH-2 CHWR Temp
24	AIP 13: CH-2 CWR Temp
26	AIP 14: CH-2 CWS Temp
28	AIP 15: CH-2 % RLA
30	AIP 16: CH-2 Evap Refrg Pressure
32	AIP 17: CH-2 Cond Refrg Pressure
34	AIP 18: CH-2 Comp Disch Rdfrg Temp
36	AIP 19: CH-2 Evap Refrg Temp
38	AIP 20: CH-2 Cond Refrg Temp
40	AIP 21: CH-3 CHWS Temp
42	AIP 22: CH-3 CHWR Temp
44	AIP 23: CH-3 CWR Temp
46	AIP 24: CH-3 CWS Temp
48	AIP 25: CH-3 % RLA
50	AIP 26: CH-3 Evap Refrg Pressure
52	AIP 27: CH-3 Cond Refrg Pressure
54	AIP 28: CH-3 Comp Disch Rdfrg Temp
56	AIP 29: CH-3 Evap Refrg Temp
58	AIP 30: CH-3 Cond Refrg Temp
60	AOP 1: CH-1 Chilled Water Setpoint
62	AOP 2: CH-1 Demand Limit Setpoint
64	AOP 3: CH-2 Chilled Water Setpoint
66	AOP 4: CH-2 Demand Limit Setpoint
68	AOP 5: CH-3 Chilled Water Setpoint
70	AOP 6: CH-3 Demand Limit Setpoint

Binary Input & Output Present Values

Starting Address	972
Data Array Offset	Function
0	BIP 1: CH-1 Status
1	BIP 2: CH-1 CW Flow
2	BIP 3: CH-1 CHW Flow
3	BIP 4: CH-1 Manual Reset Alarm
4	BIP 5: CH-1 Auto Reset Alarm
5	BIP 6: CH-1 Communication Status
6	BIP 7: CH-2 Status
7	BIP 8: CH-2 CW Flow
8	BIP 9: CH-2 CHW Flow
9	BIP 10: CH-2 Manual Reset Alarm
10	BIP 11: CH-2 Auto Reset Alarm
11	BIP 12: CH-2 Communication Status
12	BIP 13: CH-3 Status
13	BIP 14: CH-3 CW Flow
14	BIP 15: CH-3 CHW Flow
15	BIP 16: CH-3 Manual Reset Alarm
16	BIP 17: CH-3 Auto Reset Alarm
17	BIP 18: CH-3 Communication Status
18	BOP 1: CH-1 Start/Stop Control
19	BOP 2: CH-2 Start/Stop Control
20	BOP 3: CH-3 Start/Stop Control
21	Time Out