



Integrated Systems Engineering & Products

PROTOCOL CONVERTER

MODBUS RTU TO PPI

Technical Manual

DOCUMENTED BY

ISEP

Updated: 4 April 2002

Table of Content

1	INTRODUCTION	2
2	SYSTEM CONFIGURATION	2
3	CONVERTER DESCRIPTION	2
4	OVERVIEW OF COMMUNICATION PROCESS	3
5	DESCRIPTION OF PPI PROTOCOL	ERROR! BOOKMARK NOT DEFINED.
6	DESCRIPTION OF MODBUS RTU	5
6.1	Cyclic Redundancy Check (CRC-16) Calculation	5
6.2	Read Registers Query	5
6.3	Read Registers Response	6
7	CONVERSION SOFTWARE OVERVIEW	6
7.1	Serial ports configuration in the protocol converter	6
7.2	State machine diagram	7
7.3	Flow Chart of the Protocol Converter	8
7.4	Data Array Maps	Error! Bookmark not defined.

1 Introduction

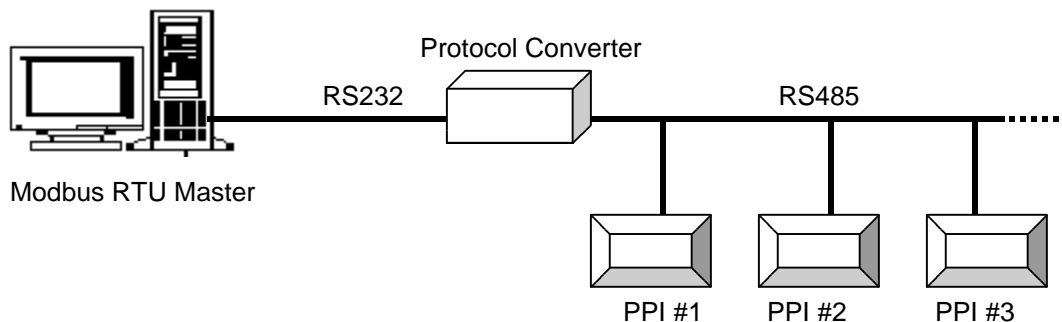
The MODBUS RTU to PPI Protocol Converter by ISEP facilitates data communications between a MODBUS RTU master and one or more PPI monitor. It does this by converting MODBUS RTU queries to equivalent PPI commands. The PPI commands are transmitted on a RS-232 bus and only the PPI monitor whose address matches responds. Upon receipt of the PPI response, the converter will translate it to MODBUS RTU recognizable by the MODBUS RTU master device.

2 System Configuration

Converter is user configurable. The Hyper Terminal of a PC connected to port 1 can be used to enter the configuration for IP address (IP address of the Converter itself), Remote IP address (IP Address of the PPI Slave, to which the Converter wants to establish a connection), Gateway address, Net-mask address, for the Ethernet port and Baud rate, Data bits, Parity and Mode(Client/Server) for the serial port 2 through which the data communications takes place.

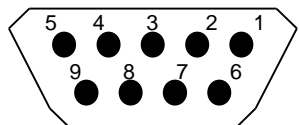
In the monitoring system, MODBUS RTU or PPI could be monitored. The converter can be configured to act as Client or Server based on which device (MODBUS RTU or PPI) is acting as Master or Slave. If the PPI is acting as Master then configure the converter to act as server and if MODBUS RTU is acting as Master then configure the converter to act as client. (See the Configuration Setup for details) .

The converter is connected to MODBUS RTU via the Port 2 (RS232 port). The converter is connected to PPI via the LAN port.. The system configuration is shown below.

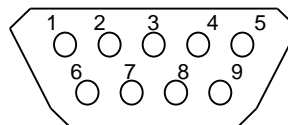


3 Converter Description

The converter is designed to sit on a shelf or platform. The pin description for the 9 pin D connector is shown below.



9 Pin Female Connector (PPI Port)	
Pin #	Description
2	RS485+
3	RS485-
5	Ground
rest	No connection



9 Pin Female Connector (MODBUS Port)	
Pin #	Description
2	Tx
3	Rx
5	Ground
7	CTS
8	RTS
rest	No connection

4 Overview of Communication Process

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

The protocol converter begins a series of events upon receiving a valid MODBUS RTU query from the MODBUS RTU master. Currently, the valid query is Read Registers. The converter will ignore any queries other than those stated above.

On power up, the converter monitors the MODBUS RTU port continuously for queries. The converter will only respond to queries with the function codes 3 which correspond to the query mentioned above.

Upon receipt of a valid query, the converter will convert the MODBUS RTU query to a PPI command. After which, it will send out the command and waits for a response from the PPI monitor..The converter can send multiple Query (five function codes of PPI) to PPI depending on the starting no. of the register and the number of registers requested by the MODBUS RTU. The maximum amount of time, that is allowed between the time the last byte of the command is sent from the converter to the time the PPI monitor responds, is 1000 milliseconds That is around 8000 milliseconds, that is the maximum amount of time that a MODBUS RTU master can wait for the response from the converter after a query for maximum no.of registers have been sent. If no response is received from the PPI monitor, the converter will abort the communication process of the current query and wait for a new MODBUS RTU query. The whole communication process of the current query could be finished before the MODBUS RTU master send next query.

If an error- free response is received from the PPI monitor, the converter will convert it to a MODBUS response. The converter then sends this response back to the MODBUS RTU master.

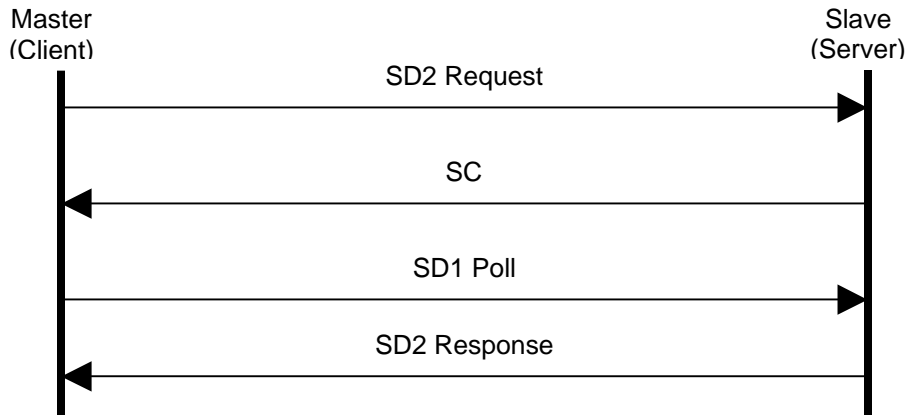
For both communication ports of the converter, the converter will do the error check first after a packet is received. The CRC-16 code of the MODBUS RTU query and the checksum byte of PPI response are used for error check. If a response from PPI monitor with errors in its checksum, the converter will ignore this response and wait for next query from the MODBUS RTU master. The converter will not re-transmitted the PPI commands to the PPI monitors and send response to the MODBUS RTU master if no error-free response is received from the PPI monitors. If a query with errors is received, the converter will ignore this query and wait for next query.

If no response is received from the converter after the MODBUS RTU master sent the query, check the converter first. If the Tx LED of the PPI port still flash after a query is received on the MODBUS port, that means the converter work well and the PPI monitor should be check. Otherwise, power off the converter and power up again. The converter will start the communication process described above automatically.

5 Description of PPI

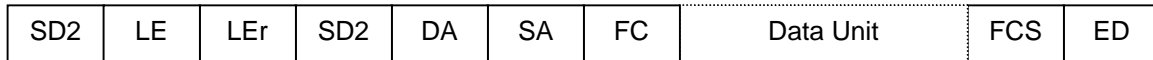
The PPI of the PLC is the slave device and the converter issues commands as a master. Upon receipt of a valid frame (with no errors), the PLC responds with corresponding data.

This section describes the handshake of the PPI protocol used in protocol converter. The handshake use are shown below:

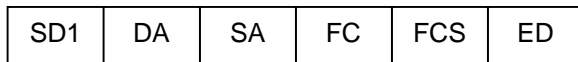


This section describes the various message formats and definition of the PPI protocol used in protocol converter. The message fields for a typical message are shown below:

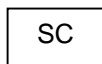
SD2 message format:



SD1 message format:



SC message format:

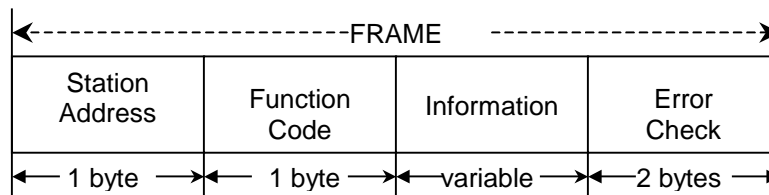


- SD2 -- Start Delimiter 2 (068H)
- SD1 -- Start Delimiter 1 (010H)
- LE -- Length Byte
- LEr -- Length Byte repeated
- DA -- Destination Address
- SA -- Source Address
- FC -- Frame Control
- Data Unit -- Message Data and Control
- FCS -- Frame Check Sequence (From DA to before FCS)
- ED -- End Delimiter (016H)

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 1 function (Read Registers) is implemented in this protocol converter.

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 than 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 Registers Query

Byte Number	Field
0	Address
1	Function Code(3L)

2	Starting Register No. <i>Hi Byte</i>
3	Starting Register No. <i>Lo Byte</i>
4	Number of Registers <i>Hi Byte</i>
5	Number of Registers <i>Lo Byte</i>
6	CRC-16 <i>Hi Byte</i>
7	CRC-16 <i>Lo Byte</i>

Example (This query is for Get System Information):

0x01	0x03	0x9C	0x40	0x00	0x0B	0x2B	0x89
------	------	------	------	------	------	------	------

0x01: the PPI monitor address is 1.
 0x03: function code.
 0x9C40: the starting point No. is 30001.
 0x000B: the number of registers
 0x2B89: CRC-16

6.3 Read Registers Response

Byte Number	Field
0	Address
1	Function Code(3L)
2	Byte Count
3 to 3+ByteCount -1	<i>Register Hi Byte</i>
4 to 4+ByteCount -1	<i>Register Lo Byte</i>
4+Byte Count	CRC-16 <i>Hi Byte</i>
5+Byte Count	CRC-16 <i>Lo Byte</i>

7 Conversion Software Overview

7.1 Serial ports configuration in the protocol converter

The converter is designed to sit on a 35 mm DIN Rail. There are two serial ports on the converter .

Port 1 : Configuration port & Modbus port .

(Note :- Once the converter is configured the same port can be used for connecting the Modbus device.)

Port 2: PPI port.

On power up, Converter enters into Configuration mode and requests for the set up configuration of the Modbus port through the port1. These requests can be viewed through the Hyper-terminal of the PC connected to port 1 of the Converter. Hyper-terminal should be connected at 19200 baud, 8 databit ,1 stop bit and parity None. HyperTerminal displays the existing configuration

Port 1

Baud-rate = 9600 baud

data-bit = 8

parity = None

Port 2

Baud-rate = 9600 baud

data-bit = 8
 parity = None

Do you wish to continue configuration ?

Press “Y” or “y” and wait for few seconds until the Converter enters into configuration mode. DO not enter “Enter Key” after pressing “Y” or “y” on the keyboard.

If the user doesn’t wish to change any configuration simply enter “any key” other than “Y” or “y” to exit out of configuration mode.

Once in the configuration mode, Converter requests for the following values :

User can enter the corresponding new values followed by “Enter Key” or simply enter the “Enter Key “ to use the existing values

Enter the baud rate.

Please use 6 digits to enter baud rate followed by “Enter Key”.

(e.g :- “019200” followed by “Enter Key”)

Enter the Databit.

Please enter “8” or “7” followed by “Enter Key”.

Enter the Parity.

Please use 1 Capital letter to enter Parity.

(e.g :- “E” for Even, “O” for Odd & “N” for None followed by “Enter Key”).

Once the user finishes the configuration, the Converter will prompt again to correct any of the values

Do you wish to continue the configuration ?

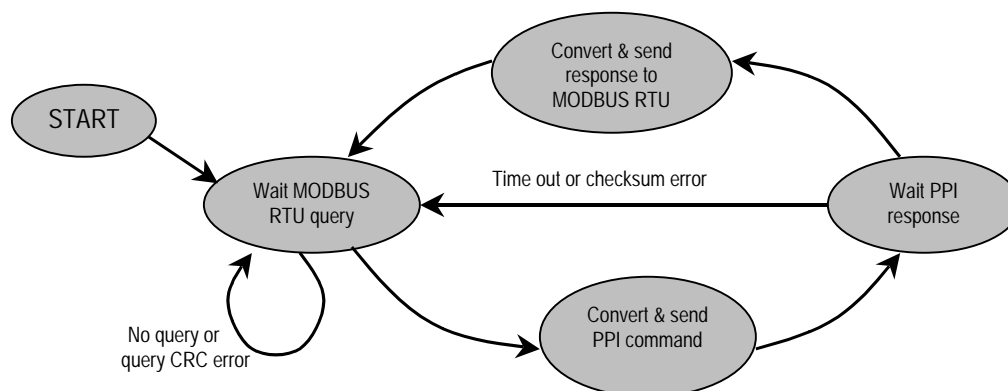
Enter “any key” other than “Y” or “y” to exit out of configuration mode.

Converter then displays the last entered values.

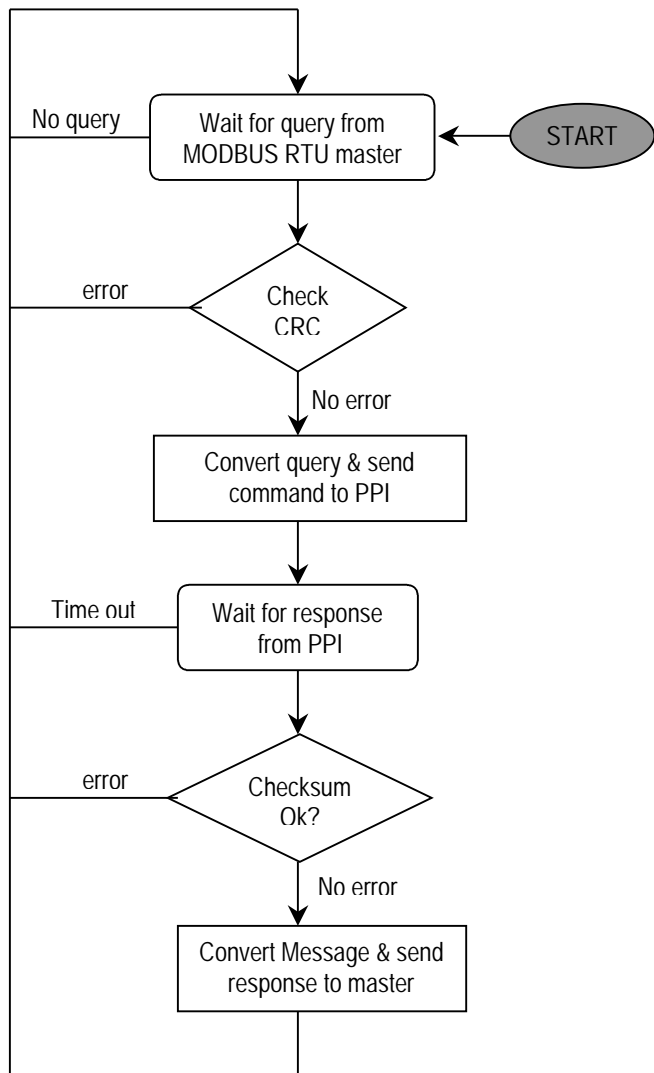
Converter will take the last entered values for setting up the Modbus port. Please disconnect the Hyper-terminal of the PC and connect the Modbus device to Port 1. Converter establishes a connection with Modbus device and enters into communication mode.

During the communication mode, for any reason if the user wants to change any configuration of the Modbus port or PPI port, A power off and Power up of the Converter is required. The Converter will start the communication process automatically. During normal operation, If there is a power failure and recovery, the Converter will automatically start-up and will use the last set communication parameters to establish communication automatically.

7.2 State machine diagram



7.3 Flow Chart of the Protocol Converter



Flowchart of the Converter