



---

Integrated Systems Engineering & Products

PROTOCOL CONVERTER

MODBUS ROUTER

Technical Manual

DOCUMENTED BY

ISEP

Updated: 7 Nov 2003

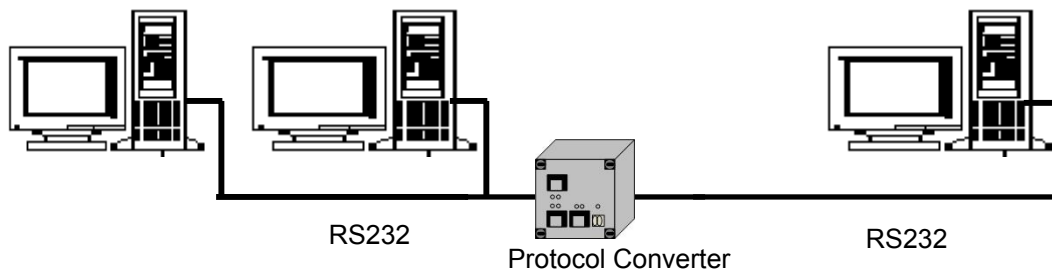
## Table of Content

<b>1</b>	<b>INTRODUCTION .....</b>	<b>2</b>
<b>2</b>	<b>SYSTEM CONFIGURATION .....</b>	<b>2</b>
<b>3</b>	<b>CONVERTER DESCRIPTION .....</b>	<b>2</b>
<b>4</b>	<b>CONFIGURATION SET-UP .....</b>	<b>3</b>
<b>5</b>	<b>CONNECTIONS DIAGRAMS .....</b>	<b>5</b>
<b>6</b>	<b>OVERVIEW OF COMMUNICATION PROCESS .....</b>	<b>5</b>
<b>7</b>	<b>DESCRIPTION OF MODBUS RTU.....</b>	<b>6</b>
<b>7.1</b>	<b>Cyclic Redundancy Check (CRC-16) Calculation.....</b>	<b>7</b>
<b>8</b>	<b>STATE MACHINE DIAGRAM.....</b>	<b>7</b>
<b>9</b>	<b>FLOW CHART OF THE PROTOCOL CONVERTER.....</b>	<b>8</b>

## 1 Introduction

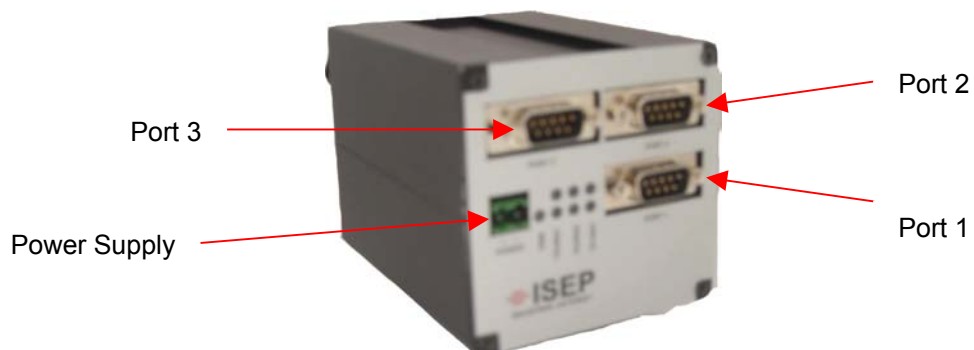
The MODBUS Router by ISEP facilitates data communications between 2 Nos. of MODBUS RTU master and one MODBUS RTU slave. It acts as a fault tolerant converter. It does this by sending MODBUS RTU queries from Modbus Master to MODBUS RTU slave. The MODBUS RTU commands are transmitted on a RS-232 cable.

## 2 System Configuration



## 3 Converter Description

The Converter is designed to sit on a 35 mm DIN Rail.



**Communication Interface**

Serial Port1,2,3 : 2 Master, 1 Slave , RS232, DB9 male connector  
Programming Port 2 : RS232, DB9 male connector

**Diagnostic LEDs**

Serial : Tx, Rx, Power

**Power**

Supply Voltage : Unregulated 10 - 30 VDC  
Power Consumption : 5 watts (max.)

**Physical Dimension**

Dimension : 105mm x 76mm x 78mm (DxWxH)  
Weight : 500g

**Environmental**

Operating Temperature : 10°C – 50°C  
Operating Humidity : 5 – 90 % non-condensing

**Installation Method**

Din-Rail, Desk Mount.

## 4 Configuration Set-Up

There are three serial ports on the Converter.

**PORT 1: Slave Port**  
**PORT 2: Master Port / configuration port**  
**PORT 3: Master Port**

To enter the configuration values or to see the existing configuration, connect a PC to the Configuration port (port 2) of the Converter and open the HyperTerminal. **Hyper Terminal should be connected at 19,200 baud, 8 data bits, 1 stop bit and parity None.** Once the HyperTerminal is connected to port 2, power off and on the converter, the HyperTerminal displays the factory settings of the following parameters:

**Port 1**

**Baud-rate = 9600 baud**

**data-bit = 8**

**parity = None**

**Port 2**

**Baud-rate = 9600 baud**

**data-bit = 8**

**parity = None**

**Port 3**

**Baud-rate = 9600 baud**

**data-bit = 8**

**parity = None**

Converter then ask if the user wants to change any configuration or use the existing one.

**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”)**

(Valid baud rates are 300,1200,2400, 9600,19200,38400,57600,115200).

**Enter the Data bit.**

**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 for the next serial port setting. After configured all the serial port setting, the converter will prompt again if the user wish to continue the configuration.

**Do you wish to continue the configuration?**

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

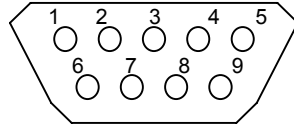
**Converter then displays the last entered values**

Once the Converter exits out of configuration mode it enters into communication mode. During the communication mode, for any reason if the user wishes to change any configuration of the ports, power off the Converter and power up again. The Converter will start the communication process described above automatically.

## 5 Connections Diagrams

### RS232 Ports

The pin description for the RS232 D9 pin cable is shown below.



9 Pin Connector (Port 1,2 and 3)	
Pin #	Description
2	Rx
3	Tx
5	Ground
rest	No connection

## 6 Overview of Communication Process

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

The protocol converter begins a series of events upon receiving a valid MODBUS RTU query from either of the two MODBUS RTU masters. Currently, the valid queries are for following function codes:

Function Code	Description
1	Read Output Table
2	Read Input Table
3	Read Registers
4	Read Analog Input
5	Force Single Output
6	Preset Single Register
7	Read Exception Status
15	Force Multiple Outputs
16	Preset Multiple Registers

The converter will ignore any function codes other than those stated above.

On power up, the converter monitors the MODBUS RTU port B and port D continuously for queries. The converter will only respond to queries with the function codes mentioned above.

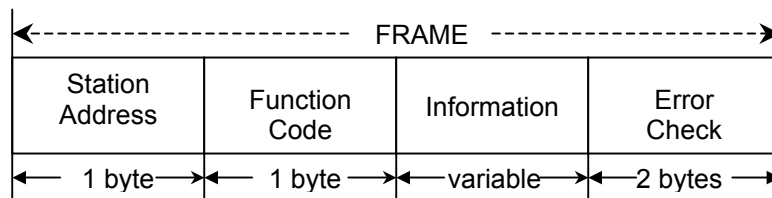
Upon receipt of a valid query, the converter will send the MODBUS RTU query to MODBUS RTU slave. After which, it waits for a response from the MODBUS RTU slave. The maximum amount of time, that is allowed between the time the last byte of the command is sent from the converter to the time the MODBUS RTU monitor responds, is 400 milliseconds That is, around 500 milliseconds is the maximum amount of time that a MODBUS RTU master can wait for the response from the converter after a query have been sent. If no response is received from the MODBUS RTU slaver, the converter will abort the communication process of the current query and monitors the MODBUS RTU port B and port D 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. This converter acts as a fault tolerant converter,since it monitors both the master ports continuously. If an error- free response is received from the MODBUS RTU slave, the converter will send it to MODBUS Master.

For all 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 CRC-16 code of MODBUS RTU response are used for error check. If a response from MODBUS RTU slave contains errors in its CRC-16 code, the converter will ignore this response and wait for next query from the MODBUS RTU master. The converter will not re-transmit the MODBUS RTU commands to the MODBUS RTU slave and only send response to the MODBUS RTU master if no error-free response is received from the MODBUS RTU slave. If a query with errors is received, the converter will ignore this query and wait for next query.

## 7 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:



## 7.1 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.

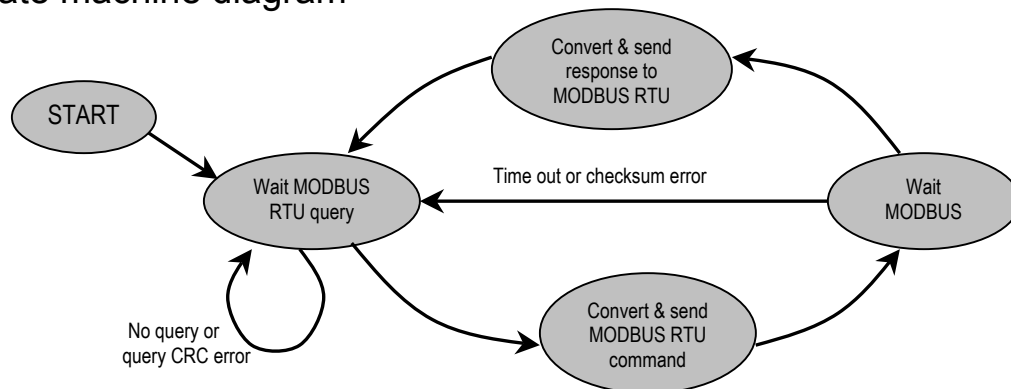
If YES, increment byte count.

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.

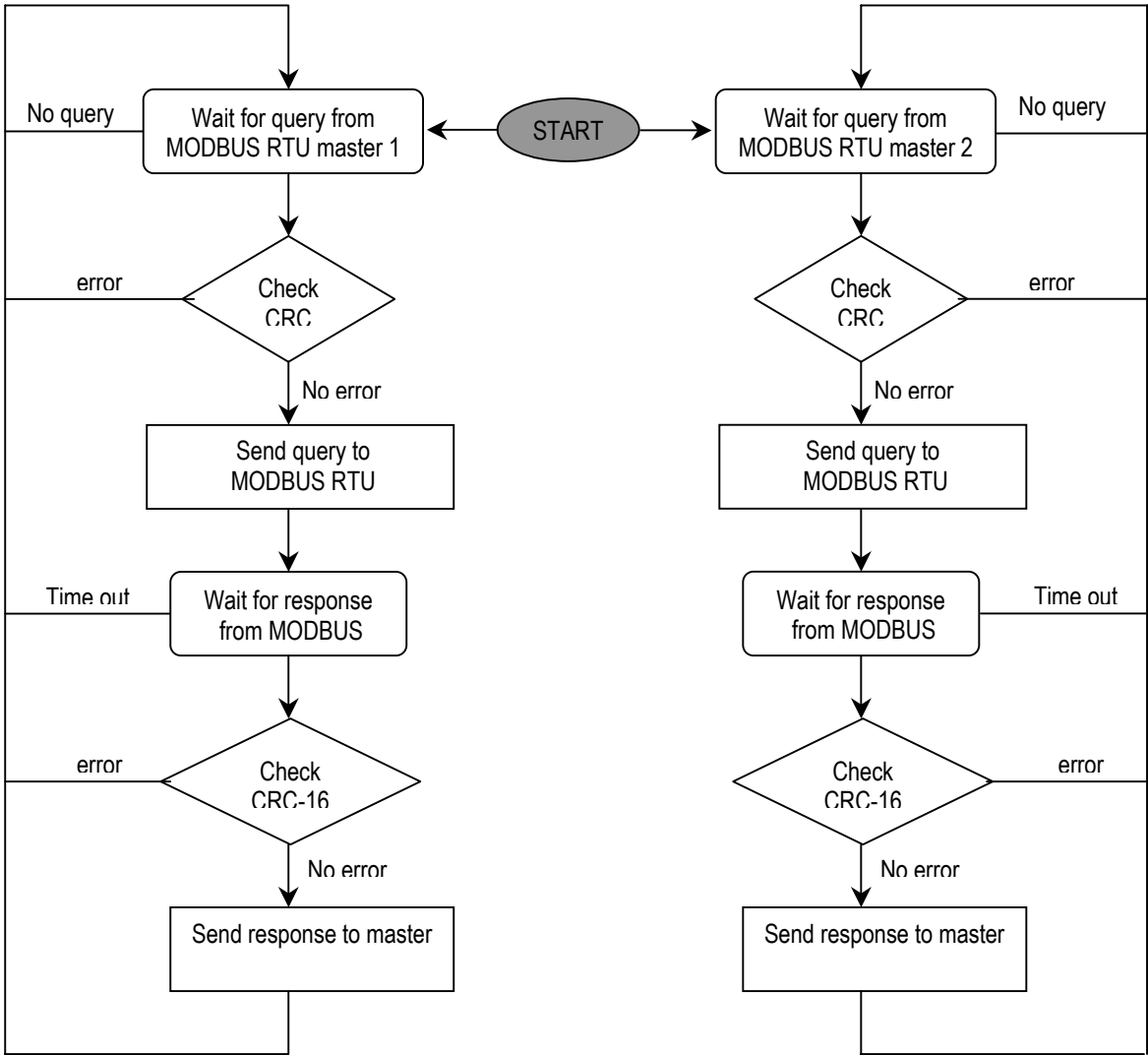
If YES, add current CRC to end of data message for transmission and exit.

## 8 State machine diagram



State Machine for Modbus-to-MODBUS RTU Protocol Converter

9 Flow Chart of the Protocol Converter



Flow chart of the Converter

## **Troubleshoot**

In normal operation both the 'Tx' LED and 'Rx' LED should be flashing normally.

<b><u>Problem</u></b>	<b><u>Implies</u></b>	<b><u>Possible cause</u></b>
Port 1 'Tx' LED flashing but 'Rx' LED not flashing	No response from Modbus Slave	Wrong comm. Setting or unmatch data query
Port 1 'Tx' LED not flashing but 'Rx' LED flashing	Wrong cable connection	Tx pin and Rx pin are wrongly terminated
Port 2 or 3 'Tx' LED not flashing but 'Rx' LED flashing. Port 1 'Tx' and 'Rx' LED not flashing	No response from Modbus Slave	Wrong comm. setting for port 1 or wrong query command

If the settings are correct and the cable connection is correct but no communication took place, do a power reset to the converter. The converter will start the communication process described above automatically.