MODBUS Slave Communication Driver

This document has the specific information related to the driver configuration. For a generic explanation on Devices, Channels, Nodes and Points configuration, please refer to reference guide.

Contents

Section 1 – Summary Information

Communication Driver Name: Modbus Slave

Implementation DLL: T.ProtocolDriver.ModbusSlave.dll

Protocol: MODBUS RTU, ASCII and TCP

Interface: TCP/IP and Serial

Description: Modbus Slave driver implements communication with master devices compatibles with Modbus Open Standard protocol. It operates as a Slave on TCP/IP or serial networks.

Devices supported: Any master Modbus device.

Protocol Options: Message Format (ASCII, RTU or RTU TCP), SlaveID

Multi-threading: One thread with a pooling cycle for each master connected.

Max number of nodes: One node for each Channel.

PC Hardware requirements: Standard PC Ethernet interface board, RS485 or RS232 port

Supported Operands:

Operand	Read	Write	Data Type	Address size
0 – Coils	✓	✓	Bit	1 bit
1 – Input Status	✓	-	Bit	1 bit
3 – Input Registers	✓	-	Word	2 bytes
4 – Holding Registers	✓	✓	Word	2 bytes

Table 1

Section 2 – Channel Configuration

Protocol Options

SlaveID: Defines the driver slave address in the Modbus Network.

Encoding: Determines how information will be packed into the message fields and decoded. The options are:

- **RTU**: Remote Terminal Unit mode, where each 8–bit byte in a message contains two 4–bit hexadecimal characters
- **ASCII**: The message is encoded in ASCII mode, where each 8–bit byte in a message is sent as two ASCII characters
- **RTU TCP**: The default transmission mode when the message is carried on a MODBUS TCP/IP network. It contains information to allow the recipient to recognize message boundaries even if the message has been split into multiple packets

Settings

Serial and MultiSerial channels:

- Default configuration for ASCII mode :
  DataBits: 7
  StopBits: 1 if parity is used, 2 if no parity
- Default configuration for RTU mode :
  DataBits: 8
  StopBits: 1 if parity is used, 2 if no parity

  Set the other fields according to your Serial or MultiSerial port configuration

TCP/IP channels:

- **Listening Port**: Defines the Tcp port where the driver will be listening for the connections, the default Tcp port for the Modbus Network is 502.

> ✍Note:
> You may need to configure your firewall to open the listening port

Section 3 – Node Configuration

Station Configuration

There is no station configuration for the Modbus Slave driver

Section 4 – Point Configuration

The syntax for the Modbus communication points is: <Operand><Address>

Where: <Operand> indicates the memory area, the valid values are:

0	for Coils
1	for Input Status
3	for Input Registers
4	for Holding Registers

For more information about the valid operands, see the Table 1:

<Address> indicates the data address in the memory area, from 1 to 65535

Ex: 400001 (Operand = Holding Register, Address = 1)

Section 5 – Troubleshoot

The status of the driver execution can be observed through the diagnostic tools, which are:

- Trace window
- Property Watch
- Module Information

The above tools indicate if the operations have succeeded or have failed where the status 0 (zero) means success. Negative values are internal error codes and positive values are protocol error codes.

Modbus protocol error codes:

Error	Name	Description
1	ILLEGAL FUNCTION	The function code received in the query is not allowable.
2	ILLEGAL DATA ADDRESS	The data address received in the query is not allowable.
3	ILLEGAL DATA VALUE	A value contained in the query data field is not allowable.
4	SLAVE DEVICE FAILURE	Error while attempting to perform the requested action.
5	ACKNOWLEDGE	Request accepted, but a long duration of time will be required.
6	SLAVE DEVICE BUSY	The slave is engaged in a long–duration program command.
7	NEGATIVE ACKNOWLEDGE	Cannot perform the program function received in the query.
8	MEMORY PARITY ERROR	Parity error in the extended memory.

Revision History

Revision	Description	Date
A	Initial Revision	May, 20th 2010