

# **MODBUS SLAVE Source Code Library**

Test Procedure

Version 2.1

January 2003, Copyright Sunlux Technologies Ltd., All rights reserved.  
Sunlux Technologies Ltd., No. 497, 6<sup>th</sup> 'A' Main, H I G Colony, R M V II Stage, Bangalore – 94, India.  
Ph: ++91 80 3417072 Fax: ++91 80 3417072  
Email: [info@sunlux-india.com](mailto:info@sunlux-india.com) Web: [www.sunlux-india.com](http://www.sunlux-india.com)

## Table of Contents

<b>1.</b>	<b>Introduction.....</b>	<b>3</b>
<b>2.</b>	<b>Starting the Application.....</b>	<b>4</b>
<b>3.</b>	<b>Using the Modbus Serial communication mode .....</b>	<b>5</b>
<b>4.</b>	<b>Error Codes for serial communication .....</b>	<b>8</b>
<b>5.</b>	<b>Using the Modbus Ethernet communication mode .....</b>	<b>9</b>

## **1. Introduction**

The MODBUS Slave Source Code Library (SCL) Test Procedure is a document that guides the end user in testing and validating his implementation of the MODBUS Slave SCL on his native platform. The document is written with respect to a 'Test Application' that is supplied along with the SCL package. The Test Application is intended to be used by end users as a easy to use MODBUS Master to test the Modbus slave implementation. The Test Application supports the following MODBUS functions:

<b>Read Multiple Coils</b>	<b>: FC 01</b>
<b>Read Discrete Inputs</b>	<b>: FC 02</b>
<b>Read Holding Registers</b>	<b>: FC 03</b>
<b>Read Input Registers</b>	<b>: FC 04</b>
<b>Write Single Coil</b>	<b>: FC 05</b>
<b>Write Single Register</b>	<b>: FC 06</b>
<b>Write Multiple Coils</b>	<b>: FC 15</b>
<b>Write Multiple Registers</b>	<b>: FC 16</b>

### **Note:**

Here Coils mean Digital Input/Outputs, which can both be read or written.

Discrete Inputs are Digital inputs, which can only be read.

Holding Registers are 16-bit Analog Input/Outputs, which can both be read or written.

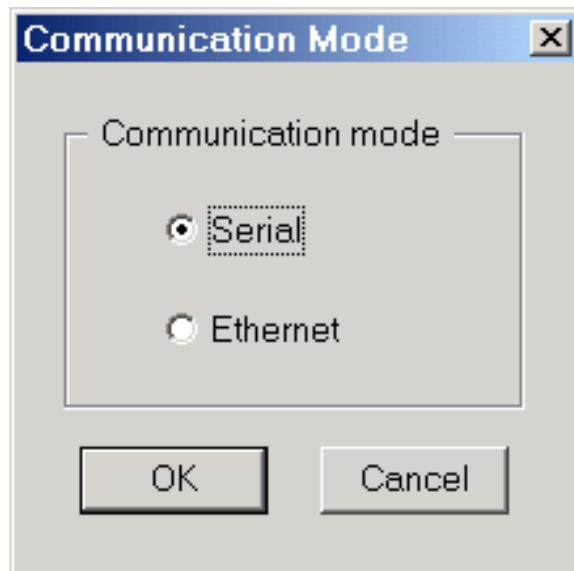
Input Registers are 16-bit Analog inputs, which can only be read.

### **How Does this Test application help the User ?**

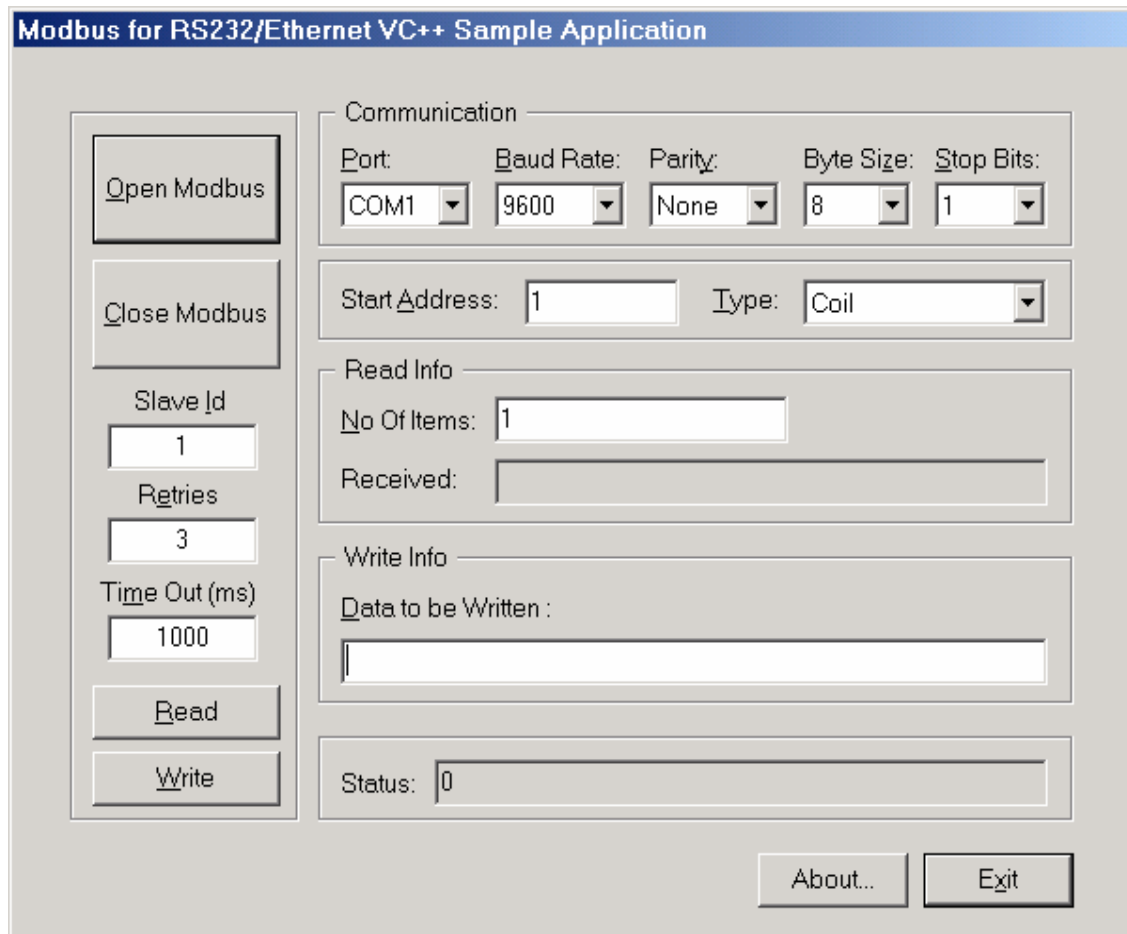
After porting the Modbus Slave SCL on the required target platform the user can test and validate it using this test application. This test application follows Modbus Master standards and is capable of sending Modbus requests to the Slave over serial and Ethernet mode. This document describes the steps to be followed to run the Modbus test application.

## 2. Starting the Application

1. Ensure that your slave program is running on the slave device.
2. Run the ModbusTest.exe on the master device.
3. The following window appears asking for the mode of communication.
4. Select the required mode and click OK.



### 3. Using the Modbus Serial communication mode



#### 1. Opening a Modbus Session –

- Select a communication port over which you want to initiate a Modbus session.
- Select the appropriate communication parameters – Baud Rate, Parity, Byte Size and Stop Bits.  
**Note:** The communication parameters for the Modbus Master and Modbus Slave must match for proper communication to take place.
- Click on the 'Open Modbus' button to open and initialize the communication path with the selected communication parameters. You will be prompted with the success or failure of the operation.

#### 2. Reading and Writing –

- Enter the Device Id of the slave from/to which you want to Read / Write

- Enter appropriate values for TimeOut (0 – 10 sec) and retries (1 – 10)
- Enter the Start Address from which you want to Read / Write.
- Select the type of data (coil, discrete input, holding register, input register) to be Read / Written
- Enter the number of items to be read in case of a Read operation
- Enter the data to be written (separated by commas or white spaces) in case of a Write operation. For digital variables enter a '1' for high and '0' for low. For analog variables, enter the analog value in decimal.
- Click on the 'Read' or 'Write' button to Read / Write the required data. You will be prompted with the success or failure of the Read / Write operation. In case of an error, check the returned error code in the status box and refer the table below for a description of the error.

### 3. Closing the Modbus Session –

- Click on the 'Close Modbus' button to end an open session of Modbus

### 4. Function codes called on selection of different options –

- **Read Multiple Coils (FC01)** is called when the following options are chosen in the respective boxes and "Read" button is clicked.
  - **Type** : Coils
  - **No Of Items** : Valid number of Items to read
  - **Start address** : Valid Starting address

-- On successful completion of the operation the read data will be displayed in the edit box "Received"
- **Read Discrete Inputs ( FC 02)**
  - **Type** : Discrete inputs
  - **No Of Items** : Valid number of Items
  - **Start address** : Valid Starting address

-- On successful completion of the operation the read data will be displayed in the edit box "Received"
- **Read Holding Registers ( FC 03)**
  - **Type** : Holding registers
  - **No Of Items** : Valid number of Items
  - **Start address** : Valid Starting address

-- On successful completion of the operation the read data will be displayed in the edit box "Received"

- **Read Input Registers (FC 04)**

- **Type** : Input registers
- **No Of Items** : Valid number of Items
- **Start address** : Valid Starting address

-- On successful completion of the operation the read data will be displayed in the edit box "Received"

- **Write Single Coil ( FC 05) - is called when the following options are chosen in the respective boxes and "Write" button is clicked.**

- § **Type** : Coils
- § **Start address** : Valid Starting address
- § **Data to be written** : Single value ( 0 or 1)

- **Write Single Register ( FC 06)**

- § **Type** : Holding Registers
- § **Start address** : Valid Starting address
- § **Data to be written** : Single valid value

- **Write Multiple Coils ( FC 15)**

- § **Type** : Coils
- § **Start address** : Valid Starting address
- § **Data to be written** : Multiple values  
(Comma/space separated)

- § **Write Multiple Registers ( FC 16)**

- § **Type** : Holding Registers
- § **Start address** : Valid Starting address
- § **Data to be written** : Multiple values  
(Comma/space separated)

§ **NOTE** : For write operations 'NoOfItems' need not be entered. The values to be written must be entered (comma/space separated) in the edit box "Data to be written" . The application will determine the number of items to write based on this entry.

#### 4. Error Codes for serial communication

During Modbus transaction on Serial communication mode the error codes are displayed in the status box. The following table shows the possible error codes and their description.

ErrorNo.	Description
10	No Memory. Memory allocation operation in the Application failed
11	Invalid request – user tried to write to either Digital Inputs or Input Registers, both of which are read only
12	Invalid MODBUS address
13	Number of items requested to read or write is exceeding the maximum limit
14	Time out. Master has sent the request and no reply has come
15	Checksum Error. Master has received the reply from the slave. But the integrity of the received data is incorrect
16	Write to communication path error. Master could not transmit the request to the specified Port/Network during read/write operation. In this case the application has to be closed and again opened
17	Read from communication path error. Master could not receive the reply from the specified Port/Network during read/write operation. In this case the application has to be closed and again opened.
18	Slave address mismatch. Master has received the reply. But the slave number field in the received frame doesn't matches with that sent by the Master.
19	The network status is marked as bad. In this case the application has to be closed and opened.
20	Invalid slave number entered by the user. Valid slave numbers are: For read functions 1 to 247. For write functions 0 to 247, where 0 is for broadcasting.



## 5. Using the Modbus Ethernet communication mode

The screenshot shows a software application window titled "Modbus for RS232/Ethernet VC++ Sample Application". The interface is divided into several sections:

- Communication:** IP Address: 0 . 0 . 0 . 0, Port: 502
- Start Address:** 1, Type: Coil (dropdown menu)
- Read Info:** No Of Items: 1, Received: (empty text box)
- Write Info:** Data to be Written: (empty text box)
- Status:** 0

On the left side, there are control buttons and input fields:

- Buttons: Open Modbus, Close Modbus, Read, Write
- Input fields: Slave Id (1), Retries (3), Time Out (ms) (1000)

At the bottom right, there are "About..." and "Exit" buttons.

### 1. Opening a Modbus Session –

- Select an IP address with which you want to initiate a Modbus session.
- Click on the 'Open Modbus' button to open and initialize the communication socket. You will be prompted with the success or failure of the operation.

### 2. Reading and Writing –

- Enter the Device Id of the slave to which you want to Read / Write
- Enter appropriate values for TimeOut (0 – 10 sec) and retries (1 – 10)
- Enter the Start Address from which you want to Read / Write.
- Select the type of data (coil, discrete input, holding register, input register) to be Read / Written
- Enter the number of items to be read in case of a Read operation

- **Enter the data to be written (separated by commas or white spaces) in case of a Write operation. For digital variables enter a '1' for high and '0' for low. For analog variables, enter the analog value in decimal.**
- **Click on the 'Read' or 'Write' button to Read / Write the required data. You will be prompted with the success or failure of the Read / Write operation.**

### **3. Closing the Modbus Session –**

- n **Click on the 'Close Modbus' button to end an open session of Modbus.**

#### **NOTE :**

##### § **When a command is executed successfully –**

**On master side** you will be prompted with a proper message and for read requests data will be available in the respective edit box.

**On the slave side** the requested function code will be printed if the MACRO **DEBUGENABLED** is defined in the SCL.

##### § **When a command execution fails –**

**On master side** appropriate error code will be generated. For example if the slave doesn't reply within the timeout period Error Number 14 will appear in the status box.

**On Slave side** the error message specifying the cause of error will be printed if the MACRO **DEBUGENABLED** is defined in the SCL.