

Modbus Slave TLM

Table of contents

1 Modbus Slave.....	2
1.1 Overview.....	2
1.1.1 Introduction.....	2
1.1.2 Features.....	2
1.1.3 Requirements.....	3
1.2 Terms of Use.....	4
1.3 Configuration Basics.....	5
1.3.1 Module Editor.....	5
1.3.2 Modicon to Datatable Mapping.....	6
1.3.3 Sample Configuration File.....	6
1.3.4 Final Steps.....	7
1.4 Debugging Tips.....	9
1.4.1 Enabling Debug Prints.....	9
2 All.....	10

1. Modbus Slave

1.1. Overview

1.1.1. Introduction

This document describes the installation, usage, and functionality of a **TLM** (TOPDOC Loadable Module) for [SoftPLC](#) version 3.x and later. This TLM implements the slave side of the Modbus Master/Slave Protocol using a serial line. See [here](#) and [here](#) for definitions of this protocol.

TLMs may be developed by any competent C/C++/Java programmer who has access to the SoftPLC C/C++/Java Programmer's Toolkit, a product readily available from SoftPLC Corporation. There are a number of Systems Integrators who are SoftPLC Partners who possess the requisite expertise. End users may also have this capability.

The TLM described by this document is called **MODBSLAV**. The Modbus Slave TLM is used to add Modbus slave capabilities to a SoftPLC runtime and to allow it to participate in master-slave communications between intelligent devices on a serial RS-232, RS-485, or RS-422 communications link.

SoftPLC offers other TLMs in support of Modbus TCP or Modbus UDP. This TLM only implements the serial line form of the protocol and only the slave side of it.

Media Type	Master	Slave
Serial Line	Third Party	*this TLM*
TCP and UDP	ModbusIPmaster	ModbusIPslave

Table 1: Four types of Modbus TLMs

1.1.2. Features

The Modbus Slave TLM operates as a SoftPLC driver and may be configured with TOPDOC NexGen. Only Modbus RTU protocol is supported, Modbus ASCII protocol is not. Modbus protocol was originally developed to talk to Modicon PLCs, and therefore devices that implement the slave side of the protocol must recognize references to Modicon memory locations. This TLM maps SoftPLC datatable files into virtual Modicon PLC memory locations. This mapping is controlled in the configuration file of the TLM.

Possible baud rates are 2400,4800,9600, 19200, 38400, 115200. Hardware handshaking is not supported, and this may make it difficult for slaves of this type to participate in a

multi-drop bus without special RS-232 to RS-485 converters which would do the hardware handshaking. Hardware handshaking is not required for a single slave bus.

The TLM supports the following Modbus Functions:

Function Number	Description
0x01	Read Coils
0x02	Read Input Discretes
0x03	Read Multiple Registers
0x04	Read Input Registers
0x05	Write Single Output
0x06	Write Single Register
0x07	Read Exception Status
0x08	Diagnostics
0x0F	Force Multiple Coils
0x10	Write Multiple Registers
0x16	Mask Write Registers

1.1.3. Requirements

- Version 3.x SoftPLC or later.

1.2. Terms of Use

Because of the variety of uses of the information described in this manual, the users of, and those responsible for applying this information must satisfy themselves as to the acceptability of each application and use of the information. In no event will SoftPLC Corporation be responsible or liable for its use, nor for any infringements of patents or other rights of third parties which may result from its use.

SOFTPLC CORPORATION MAKES NO REPRESENTATIONS OR WARRANTIES WITH RESPECT TO THE CONTENTS HEREOF AND SPECIFICALLY DISCLAIMS ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE.

SoftPLC Corporation reserves the right to change product specifications at any time without notice. No part of this document may be reproduced by any means, nor translated, nor transmitted to any magnetic medium without the written consent of SoftPLC Corporation.

SoftPLC, and TOPDOC are registered trademarks of SoftPLC Corporation.

© Copyright 2006 SoftPLC Corporation ALL RIGHTS RESERVED

First Printing: January, 2006

Last Printing: January, 2006

SoftPLC Corporation 25603 Red Brangus Drive

Spicewood, Texas 78669

USA Telephone: 1-800-SoftPLC

Fax: 512/264-8399

URL: <http://softplc.com>

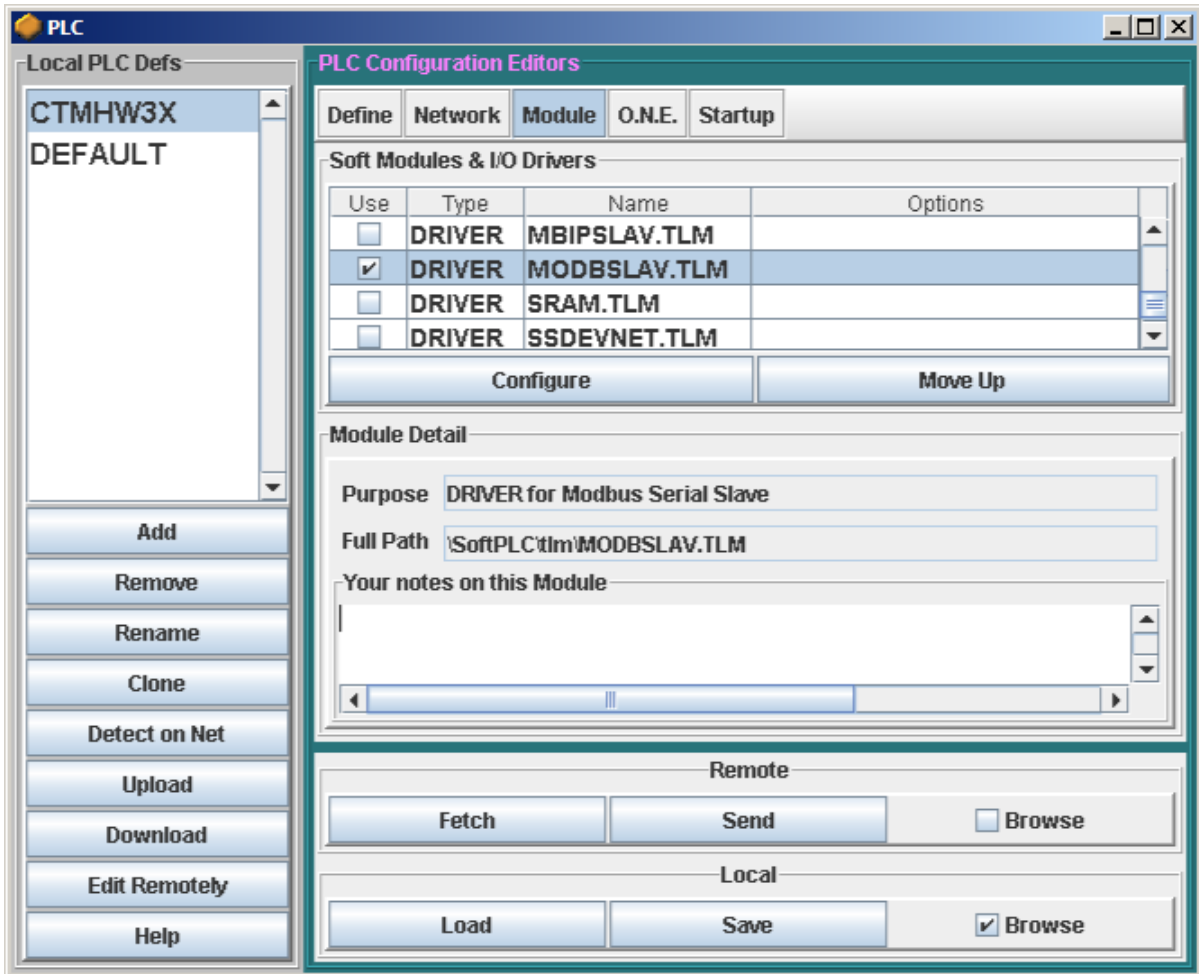
Email: support@softplc.com

1.3. Configuration Basics

Perform the steps in this section in sequence.

1.3.1. Module Editor

Use TOPDOC NexGen's PLC » Module Editor to load and configure the MODBSLAV.TLM for use with SoftPLC, as shown in the figure below. Clicking **Configure** opens the **configuration text editor**. Then press **Fetch** to transfer the default MODBSLAV.LST file from the processor to the editor. This is the configuration file where you will assign the serial ports and datatable files to be used by the MODBSLAV driver. A maximum of 8 serial ports may be defined.



module editor

1.3.2. Modicon to Datable Mapping

SoftPLC datatable files can hold up to 10,000 elements per file. Therefore any of the standard Modicon memory regions can have the following reference number ranges:

Modicon Memory Type	Supported Reference Number Range
Holding Registers	400001 to 410000
Input Registers	300001 to 310000
Input States	100001 to 110000
Output Coils	000001 to 010000

Note:

Usually the Modicon memory regions are assigned to an 'N' (Integer) file within SoftPLC. **The assignment must start at element number zero.** Notice the N104:0 used in the example below. Any datatable file references made in the MODBSLAV configuration file must exist in the associated SoftPLC's running APP file. This datatable memory must be created by using TOPDOC NexGen. This TLM does not automatically create this datatable memory for you.

1.3.3. Sample Configuration File

Below is a sample MODBSLAV.LST file

```

; This is the configuration file for the MODBUS Slave TLM.
; Anything after a semicolon is ignored during parsing.
; There are two sections: [DRIVER] and [PORTS]:

; [DRIVER] contains global options, such as DEBUG and IOCHECK
;
; Set DEBUG to > 0 if you want diagnostic output temporarily.
;  DEBUG=0 gives no diagnostic output.
;  DEBUG=1 gives a nice Query Response trace.
;  DEBUG=2 gives what 1 does and more.
;
; Set IOCHECK to YES if you want the turn around time to be reduced by
;  having SoftPLC service the modbus commands more frequently than once
;  per scan. This is effective only when you have more rungs than
;  STARTUP.LST's oneCheckInterval setting.

[DRIVER]
DEBUG=0
IOCHECK=NO

; [PORTS] should include one line for each serial port you want active.
; All the StartAddress's must be based at element zero of a file of one of
; the following types: INTEGER, BIT, INPUT, or OUTPUT.

[PORTS]
;COMPORT 0-31 where 0=COM1, 1=COM2, etc.
; | Baudrate: 9600 or 19200 or 38400
; | | Databits: 8
; | | | Parity: N or E or O
; | | | Stopbits: 1 or 2
; | | | Modbus slave address: 1 - 247
; | | | Holding Registers StartAddress:
; | | | | Output Coils StartAddress
; | | | | Input Registers StartAddress
; | | | | Input States
StartAddress
; | | | | | | | | |
; | 1, 9600, 8, N, 2, 2, N104:0, N100:0, N103:0, N101:0
; 2, 19200, 8, N, 1, 4, N100:0, B3:0, N7:0, I:0 spare, as needed

; Make sure you use TOPDOC to create the Datatable Files that you
; are referencing for each port.

;EOF

```

1.3.4. Final Steps

- After you have finished configuring the driver, press **Save** and then **Send**. This will save the MODBSLAV.LST file to your Windows workstation and then transfer the file to the SoftPLC's Flash disk.
- Now reboot the Tealware/SoftPLC Processor. SoftPLC is now ready to process Modbus

- serial data requests from the Master.
- The last step is to program the Modbus Master to send the read and write request(s) to the SoftPLC Modbus Slave.

1.4. Debugging Tips

This section gives tips on debugging problems on the Modbus network.

1.4.1. Enabling Debug Prints

In the configuration file there is the `DEBUG` setting. It may be set to 0, 1, or 2 to indicate that you want no, some, or most debugging respectively. Remember a `DEBUG` value of "0" means no debugging. On version 4.x SoftPLC, all process output from the SoftPLC runtime engine is normally directed to the syslog, because SoftPLC runs as a daemon normally. The syslog can be configured in a number of different ways, but the default uses a small RAM resident FIFO and eventually will run out of space and wrap back around on itself. Rather than reconfiguring the syslog, there is an easier way.

Following is a procedure to get the debugging output into a text file.

1. Log into SoftPLC using either a) PUTTY from Windows or b) using ssh from Linux or c) at the command prompt of the SoftPLC system.
2. Run this command:

```
# /etc/init.d/softplc.sh stop
```
3. Change into the /SoftPLC/run directory:

```
# cd /SoftPLC/run
```
4. You can run SoftPLC from the command prompt now and redirect its output to an arbitrary file (named `out.txt` here). We put that file into the RAM disk which is anchored in the `/tmp` directory.

```
# ./runsplc > /tmp/out.txt
```
5. Let this run for 5-60 seconds, then press `control-C`. Now you have the output captured in file `/tmp/out.txt`, each request-response transaction will be captured in that file.
6. You can look at the file using the program named "less".

```
# less /tmp/out.txt
```

You can look at this output with the Modbus Specification, and the manual for your Modbus master software in hand. Press `ESC` when done.
7. When done, remember to set debug back to "0", then you can start SoftPLC as a daemon either by a) power cycling the box or b) doing the following:

```
# /etc/init.d/softplc.sh start
```

2. All