

# ModbusIP Slave TLM

## Table of contents

1 About.....	2
1.1 Modbus IP Slave TLM.....	2
1.1.1 Introduction.....	2
1.2 Terms of Use.....	2
1.2.1 .....	2
1.3 Usage.....	3
1.3.1 Concepts.....	3
1.3.2 Features.....	3
1.3.2.1 Typical System Architecture.....	4
1.3.2.2 Supported Modbus Commands.....	5
1.3.3 Requirements.....	5
1.3.4 Installation.....	5
1.3.5 Configuration.....	6
1.3.5.1 Sample Configuration File.....	6
2 All.....	8

## 1. About

### 1.1. Modbus IP Slave TLM

#### 1.1.1. Introduction

This document describes the installation, usage, and functionality of a TOPDOC Loadable Module (TLM) for [SoftPLC](#) version 4.x and later. The TLM implements the slave (or server) side of the [Modbus TCP](#) protocol. As an extension of the standard, it also implements the same protocol on UDP/IP.

It may be used to talk to an MMI Graphical User Interface station or it may be used to talk between other controllers on an ethernet.

### 1.2. Terms of Use

#### 1.2.1.

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 2005 SoftPLC Corporation ALL RIGHTS RESERVED

First Printing: March, 2005

Latest Printing: June, 2005

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](mailto:support@softplc.com)

## 1.3. Usage

### 1.3.1. Concepts

The SoftPLC runtime software supports TLMs, which are shared library extensions to SoftPLC. A TLM may be loaded either as a **DRIVER** or as a **MODULE**. The difference between a DRIVER and a MODULE is that a DRIVER is called once per SoftPLC scan, and optionally an additional number of times per scan. A MODULE is only called when the control program decides to call it. TLMs are made known to SoftPLC in the MODULES.LST file which may be edited by TOPDOC NexGen by traversing to: PLC | Modules.

### 1.3.2. Features

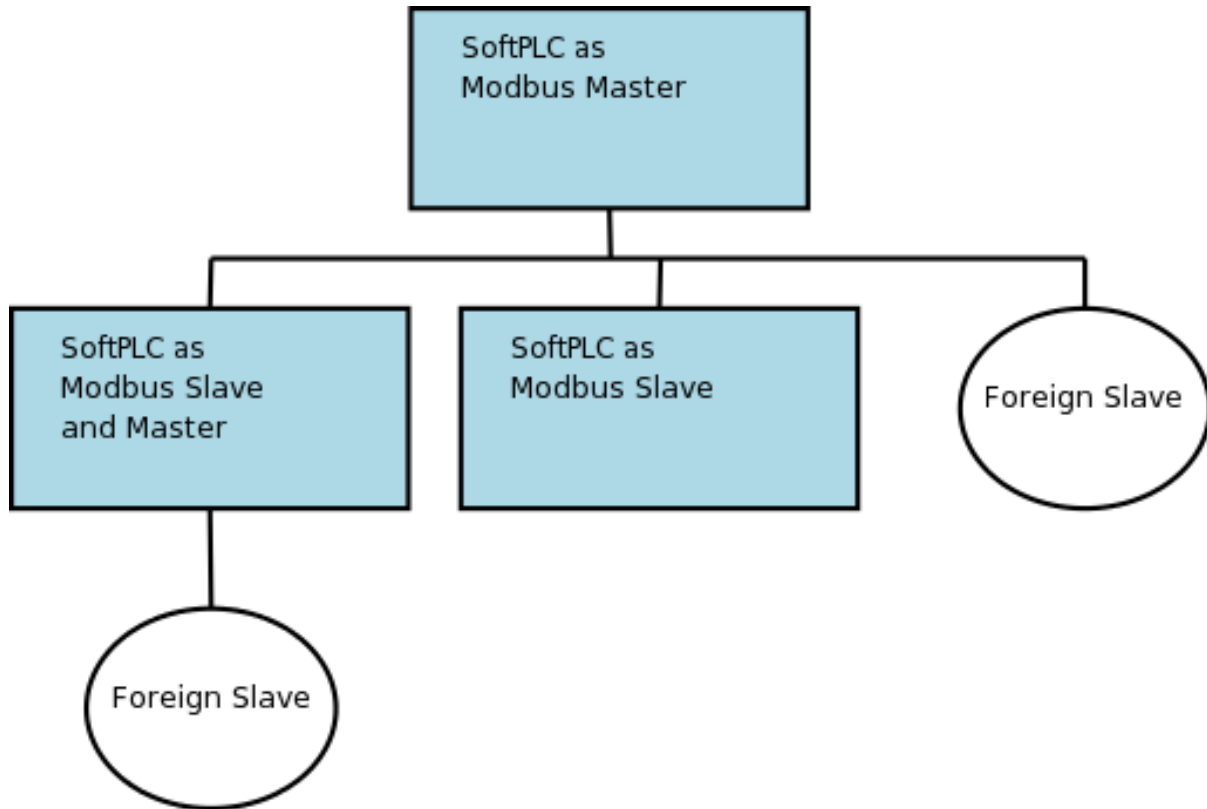
In order to use the modbus IP slave TLM you need a working ethernet connection. This TLM listens on the standard TCP and UDP port 502 and acts as a "slave" or a "server". On the other end of any Modbus conversation is a "master" or a "client". Because the slave implemented by this TLM supports both TCP and UDP carriers for the Modbus protocol, the master may choose which it will use. More than one master may talk to this slave at once, and each master may chose independently to do this on either TCP or UDP, or both.

When TCP is used by the master, there is a TCP connection established. When UDP is used by the master, the request response sequence takes place in a "connection-less" fashion. Because UDP is not a guaranteed delivery service, any Modbus master using UDP instead of TCP to carry the modbus requests and responses should implement timeout and retry logic. On most hardware platforms, SoftPLC will respond to a request within a millisecond or two, and knowledge of this can be used to pick reasonable timeouts for the required UDP logic.

This slave TLM supports 32 simultaneous Modbus/TCP connections and an unlimited number of Modbus/UDP sessions. SoftPLC 4.x's integrated firewall may be used to restrict which network nodes can communicate to this TLM.

SoftPLC also provides a Modbus IP Master TLM, which is documented [here](#). A single SoftPLC machine can be both a master and a slave. This capability gives the systems designer the power and flexibility to develop very powerful, fast and flexible distributed control systems. Obviously a SoftPLC Modbus master can talk to a SoftPLC Modbus slave as well as third party slaves.

### 1.3.2.1. Typical System Architecture



distributed control

### 1.3.2.2. Supported Modbus Commands

Modbus Function	Name	Supported
1	Read Coils	Yes
2	Read Input Discretes	Yes
3	Read Multiple Registers	Yes
4	Read Input Registers	Yes
5	Write Coil	Yes
6	Write Single Register	Yes
7	Read Exception Status	No
15	Force Multiple Coils	Yes
16	Write Multiple Registers	Yes
20	Read General References	No
21	Write General Registers	No
22	Mask Write Register	Yes
23	Read Write Registers	Yes
24	Read FIFO Queue	No

### 1.3.3. Requirements

- A working ethernet or PPP link.
- Version 4.x SoftPLC or later.

### 1.3.4. Installation

The TLM is named mbipslav.tlm.so and is found as part of the standard SoftPLC 4.x installation in the /SoftPLC/tlm directory. To use it you merely have to enable it in NexGen's PLC | MODULES editor. Then you must edit the text file MBIPSLAV.LST which is the TLM's configuration file.

It is easy to edit the configuration file from the PLC | MODULES editor as well. Simply click on the "Configure" button after selecting and enabling the Modbus IP Slave TLM there.

### 1.3.5. Configuration

Modbus commands were originally designed for a Modicon PLC. Therefore they assume 4 different types of memory regions:

1. Input States (boolean inputs)
2. Input Registers (16 bit words)
3. Output Coils (boolean outputs)
4. Output Registers (16 bit words)

In order to make use of the Modbus commands, we must map the 4 required memory regions to datatable files within SoftPLC.

Modbus TCP protocol includes the original "slave id" field which was part of the modbus on serial line protocol. In the case of communications on ethernet, either via TCP or UDP, this field is no longer used to qualify the actual network node that will respond to a request. The reason for this is because the **IP Address** in the ethernet frame serves this purpose. Therefore the "slave id" field becomes available for another use. This TLM can be configured to use the slave id field as a "selector" in the mapping of the four Modbus memory regions to SoftPLC datatable files.

All the configuration is done in the single text file MPIPSTAV.LST. Below is a sample. Any part of a line of text to the right of a semicolon ; is considered a comment and is ignored.

#### 1.3.5.1. Sample Configuration File

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

; [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=2
IOCHECK=NO
```

```

; [SLAVES] should include one line for each group of files you want to
; expose. Each SLAVE is a fictitious internal Modicon device that is
; addressed by including the corresponding "Slave ID" in the modbus
; request packet.
;
; Each SLAVE line can provide access to up to 4 datatable files, one
; for each of these purposes:
;
; 1) Holding Registers,
; 2) Output Coils,
; 3) Input Registers,
; 4) Input States.
; Any or all of the 4 files can be the same as each other, for each
; SLAVE. The given files must start at element 0 and be of type
; INTEGER, BIT, INPUT, OUTPUT, or STATUS.
;
; A SLAVE with a SlaveID of -1 is special. It will be used when the
; incoming request packets do not match any other SlaveID. In many cases,
; you will only need a single SLAVE row and will want to use -1 as the
; SlaveID.

[SLAVES]
;SlaveID 0-255
;| Holding Registers StartAddress:
;| Output Coils StartAddress
;| Input Registers StartAddress
;| Input States StartAddress
;|
-1, N12:0, N12:0, N12:0, N12:0

;0, N100:0, N101:0, N102:0, N103:0
;2, N100:0, B3:0, N7:0, I:0 edit/copy, uncomment as needed

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

; Make sure you install this TLM as a DRIVER and not a MODULE.

; FIREWALL:
; If you want to restrict which network nodes can access this
; SoftPLC's MODBUS IP SLAVE functionality, then you should setup
; the internal Gatecraft Firewall with support from SoftPLC Corp.

;EOF

```

**Note:**

You may supply a single SLAVE row that has a -1 as the SlaveID. A SlaveID of -1 is special. This row, if supplied, is the default when the incoming SlaveID does not match any other row.

After using NexGen to edit the configuration file, **Send** it down to the SoftPLC. The next step is to cycle power on the SoftPLC for the changes to take place. As an alternative to cycling power, you may enter "Remote Program" mode using NexGen, then select "Remote

Program" a second time. This psuedo transition from Remote Program to Remote Program is a signal to the TLM that it should reload its configuration file. This way you can reconfigure without cycling power, although it does require you enter "Remote Program" mode (twice!).

## 2. All