

# SDI-12 TLM

## Table of contents

1 About.....	3
1.1 Overview.....	3
1.1.1 Introduction.....	3
1.1.2 Concepts.....	3
1.1.3 Features.....	4
1.1.4 Requirements.....	4
1.2 Terms of Use.....	4
2 Installation and Usage.....	6
2.1 Installation.....	6
2.1.1 TLM Installation.....	6
2.1.2 Other Utilities.....	6
2.2 Usage.....	7
2.2.1 Ladder Instructions.....	7
2.2.1.1 TLI: SDI_CMD_M.....	7
2.2.1.2 TLI: SDI_CMD_MX.....	8
2.2.1.3 TLI: SDI_CMD_C.....	9
2.2.1.4 TLI: SDI_CMD_CX.....	10
2.2.1.5 TLI: SDI_CMD_RX.....	11
2.2.2 Error Codes.....	12
2.3 Usage.....	14
2.3.1 Ladder Instructions.....	14
2.3.2 SDICmdTool.....	14
2.3.3 simsensor.....	15

3 All..... 15

## 1 About

### 1.1 Overview

#### 1.1.1 Introduction

The SDI12 TLM is an add-on firmware option for SoftPLC version 4.6 and later which adds communications support for devices using the SDI-12 protocol. This capability is implemented as a TOPDOC Loadable Module (**TLM**), written in C++ and implements several ladder instructions (TLIs) which can be used for gathering data from one or more SDI-12 devices. This document describes the installation, usage, and functionality of the SDI12 TLM.

This TLM initiates SDI-12 measurement command/reply sequences by way of a ladder logic instruction, giving the user full programmatic control over the trigger conditions. Queuing is implemented, allowing multiple commands to be triggered within a very small window of time. The TLM will queue up the requests on a first in, first out basis.

#### 1.1.2 Concepts

The **SoftPLC runtime engine** 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 and not as an inherent part of the scan. TLMs are made known to SoftPLC in the MODULES.LST file which may be edited by TOPDOC NexGen by traversing to: PLC | Modules.

This SDI12 TLM is to be used as a DRIVER, and it provides TLIs that you call from your ladder program. A new request is sent on any false to true rung condition edge. All of the provided TLIs take a CONTROL datatable element as their first parameter and use this to keep track of edge triggering and sending progress. Each TLI used in your ladder program must have its own dedicated unique CONTROL element.

### 1.1.3 Features

Up to 20 requests may be queued at one time before the SDI12-related TLIs will report that the queue is full. As soon as the first request has been processed, this frees up another slot and the TLM is then ready for another rising edge trigger condition.

**Note:**

When designing your ladder program, keep in mind the communication speed limits of the SDI-12 protocol (which specifies 1200 baud). A full communications sequence could take several seconds (per sensor) depending on the number of data measurements.

Each of the SDI\_CMD TLIs performs a full communication sequence with the specified sensor(s), meaning: start measurement command, get reply, wait for data to be ready, send data command, get reply. This includes the case of multiple "send data" commands being necessary to retrieve the full data set from the sensor.

Additionally, the SDI12 TLM package includes administrative/debugging utility programs to provide useful functionality and configuration/testing of SDI-12 sensor devices and your cabling setup. These utilities are further described in the [usage](#) section of this manual.

### 1.1.4 Requirements

- A sensor device using the SDI-12 protocol attached to your SoftPLC.
- Version 4.6 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 2012 SoftPLC Corporation ALL RIGHTS RESERVED

First Printing: Feb, 2014

Latest Printing: Feb, 2014

SoftPLC Corporation 25603 Red Brangus

Drive Spicewood, Texas 78669

USA Telephone: 1-800-SoftPLC

Fax: 512/264-8399

URL: <http://softplc.com>

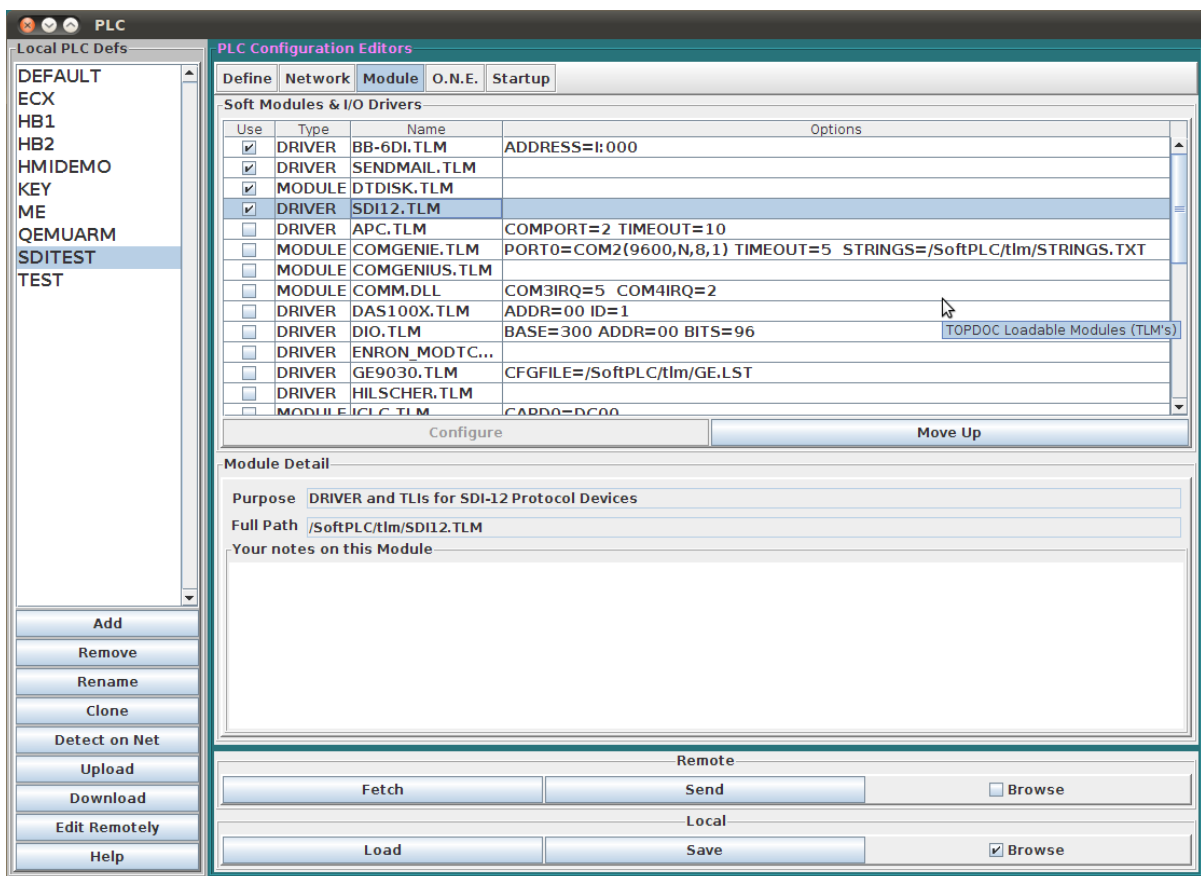
Email: [support@softplc.com](mailto:support@softplc.com)

## 2 Installation and Usage

### 2.1 Installation

#### 2.1.1 TLM Installation

The TLM is named `sdi12.tlm.so`, which you will need to install onto the SoftPLC from a packaged form using the `opkg` tool or directly copy the file onto the SoftPLC flash in the `/SoftPLC/tlm` directory. Then, enable it for use through NexGen's PLC | MODULES editor by selecting *Use* for SDI12.TLM.



#### 2.1.2 Other Utilities

The administrative utility **SDICmdTool** and sensor-simulator/observer **simsensor** are installed by the `opkg` tool to `/home/root/SDI-12-CmdTool` and `/home/root/SDI-12-Sim` respectively. They can be moved if desired, but it is not recommended to install these files directly since the package will download required dependencies to run the utilities.

## 2.2 Usage

### 2.2.1 Ladder Instructions

#### 2.2.1.1 TLI: SDI\_CMD\_M

The SDI\_CMD\_M instruction is used to perform a "Start Measurement" command and the necessary subsequent commands to retrieve the data measurements from the sensor.

There are five (5) parameters for this TLI:

Parameter Name	Description
<b>"Control:"</b>	Control element (R) address. The ".CTL" word bits are used to indicate activity status for the TLI. When the instruction is triggered, the enable (EN) and queue (EU) bits are set. On completion of sending, either the done (DN) or error (ER) bit will be set: the DN for a successful send, the ER for any error. The ER bit will also be set if the message queue is currently full.
<b>"Status:"</b>	Integer element (N) address. This is used to store the status code for the communication sequence.
<b>"Sensor Adr:"</b>	String element (ST) address. This should be a single-character string; it is used to specify the address of the sensor to which the command should be sent. Valid addresses: 0-9, A-Z, a-z
<b>"Results:"</b>	Float (F) block address. This is a block address (e.g. #F8:0) which will be used to store the data returned from the sensor. Data values are stored starting at the specified element, up to a max of 100 values (the file will automatically be resized such that it will support 100 elements if it does not already).
<b>"Use CRC:"</b>	Integer element (N) address, or integer value. A '1' or '0' value: 1 means yes, 0 means no. If yes, the sensor will return its data values with CRC characters added (this is handled and checked automatically by the driver).

Table 1: SDI\_CMD\_M TLI Parameters

### 2.2.1.2 TLI: SDI\_CMD\_MX

The SDI\_CMD\_MX instruction is used to perform a "Additional Measurement" command and the necessary subsequent commands to retrieve the data measurements from the sensor. There are six (6) parameters for this TLI:

Parameter Name	Description
<b>"Control:"</b>	Control element (R) address. The ".CTL" word bits are used to indicate activity status for the TLI. When the instruction is triggered, the enable (EN) and queue (EU) bits are set. On completion of sending, either the done (DN) or error (ER) bit will be set: the DN for a successful send, the ER for any error. The ER bit will also be set if the message queue is currently full.
<b>"Status:"</b>	Integer element (N) address. This is used to store the status code for the communication sequence.
<b>"Sensor Adr:"</b>	String element (ST) address. This should be a single-character string; it is used to specify the address of the sensor to which the command should be sent. Valid addresses: 0-9, A-Z, a-z
<b>"Results:"</b>	Float (F) block address. This is a block address (e.g. #F8:0) which will be used to store the data returned from the sensor. Data values are stored starting at the specified element, up to a max of 100 values (the file will automatically be resized such that it will support 100 elements if it does not already).
<b>"Use CRC:"</b>	Integer element (N) address, or integer value. A '1' or '0' value: 1 means yes, 0 means no. If yes, the sensor will return its data values with CRC characters added (this is handled and checked automatically by the driver).
<b>"Meas. Num:"</b>	String element (ST) address. This should be a single-character string; it is used to specify the 'additional' measurement number. Valid values: 1-9

Table 1: SDI\_CMD\_MX TLI Parameters



### 2.2.1.3 TLI: SDI\_CMD\_C

The SDI\_CMD\_C instruction is used to perform a "Start Concurrent Measurement" command and the necessary subsequent commands to retrieve the data measurements from the sensor(s).

There are five (5) parameters for this TLI:

Parameter Name	Description
<b>"Control:"</b>	Control element (R) address. The ".CTL" word bits are used to indicate activity status for the TLI. When the instruction is triggered, the enable (EN) and queue (EU) bits are set. On completion of sending, either the done (DN) or error (ER) bit will be set: the DN for a successful send, the ER for any error. The ER bit will also be set if the message queue is currently full.
<b>"Status:"</b>	Integer (N) block address. This is used to store the status code(s) for the communication sequence. Each element will hold a status corresponding to the sensor in the "Sensor Adr" parameter, by index in the string.
<b>"Sensor Adr:"</b>	String element (ST) address. This is used to specify the address of the sensor(s) to which the command should be sent. Valid addresses: 0-9, A-Z, a-z. Each address specified in the string must be unique (no repeated characters).
<b>"Results:"</b>	Float (F) block address. This is a block address (e.g. #F8:0) which will be used to store the data returned from the sensor. Data values are stored starting at the specified element, up to a max of 100 values PER SENSOR (the file will automatically be resized such that it will support 6200 elements (100 max values per sensor * 62 maximum sensors) if it does not already). Each sensor's values will be offset into the block by 100 * the index of the address in the string, starting at 0. It is strongly recommended you specify this block starting at element 0. Example: Sensor string of "04a" would have sensor 0's values starting at F8:0, sensor 4's values starting at F8:100, and sensor a's values starting at F8:200
<b>"Use CRC:"</b>	Integer element (N) address, or integer value. A '1' or '0' value: 1 means yes, 0 means no. If yes, the sensor will return its data values with CRC characters added

Parameter Name	Description
	(this is handled and checked automatically by the driver).

Table 1: SDI\_CMD\_C TLI Parameters

#### 2.2.1.4 TLI: SDI\_CMD\_CX

The SDI\_CMD\_CX instruction is used to perform a "Additional Concurrent Measurement" command and the necessary subsequent commands to retrieve the data measurements from the sensor(s).

There are six (6) parameters for this TLI:

Parameter Name	Description
<b>"Control:"</b>	Control element (R) address. The ".CTL" word bits are used to indicate activity status for the TLI. When the instruction is triggered, the enable (EN) and queue (EU) bits are set. On completion of sending, either the done (DN) or error (ER) bit will be set: the DN for a successful send, the ER for any error. The ER bit will also be set if the message queue is currently full.
<b>"Status:"</b>	Integer (N) block address. This is used to store the status code(s) for the communication sequence. Each element will hold a status corresponding to the sensor in the "Sensor Adr" parameter, by index in the string.
<b>"Sensor Adr:"</b>	String element (ST) address. This is used to specify the address of the sensor(s) to which the command should be sent. Valid addresses: 0-9, A-Z, a-z. Each address specified in the string must be unique (no repeated characters).
<b>"Results:"</b>	Float (F) block address. This is a block address (e.g. #F8:0) which will be used to store the data returned from the sensor. Data values are stored starting at the specified element, up to a max of 100 values PER SENSOR (the file will automatically be resized such that it will support 6200 elements (100 max values per sensor * 62 maximum sensors) if it does not already). Each sensor's values will be offset into the block by 100 * the index of the address in the string, starting at 0. It is strongly recommended you specify this block starting at element 0.

Parameter Name	Description
	Example: Sensor string of "04a" would have sensor 0's values starting at F8:0, sensor 4's values starting at F8:100, and sensor a's values starting at F8:200
"Use CRC:"	Integer element (N) address, or integer value. A '1' or '0' value: 1 means yes, 0 means no. If yes, the sensor will return its data values with CRC characters added (this is handled and checked automatically by the driver).
"Meas. Nums:"	String element (ST) address. This should be a single-character string; it is used to specify the 'additional' measurement number. Valid values: 1-9. Each character in the string will be matched, by index, to the corresponding sensor character in the "Sensor Adr" string.

Table 1: SDI\_CMD\_CX TLI Parameters

### 2.2.1.5 TLI: SDI\_CMD\_RX

The SDI\_CMD\_RX instruction is used to perform a "Continuous Measurement" command, to retrieve data from a sensor running in continuous mode (if supported). If the sensor does not support the requested 'R' measurement, the control element will have the ER bit set and the status will contain an error code indicating NO DATA.

There are six (6) parameters for this TLI:

Parameter Name	Description
"Control:"	Control element (R) address. The ".CTL" word bits are used to indicate activity status for the TLI. When the instruction is triggered, the enable (EN) and queue (EU) bits are set. On completion of sending, either the done (DN) or error (ER) bit will be set: the DN for a successful send, the ER for any error. The ER bit will also be set if the message queue is currently full.
"Status:"	Integer element (N) address. This is used to store the status code for the communication sequence.
"Sensor Adr:"	String element (ST) address. This should be a single-character string; it is used to specify the address of the sensor to which the command should be sent. Valid addresses: 0-9, A-Z, a-z

Parameter Name	Description
<b>"Results:"</b>	Float (F) block address. This is a block address (e.g. #F8:0) which will be used to store the data returned from the sensor. Data values are stored starting at the specified element, up to a max of 100 values (the file will automatically be resized such that it will support 100 elements if it does not already).
<b>"Use CRC:"</b>	Integer element (N) address, or integer value. A '1' or '0' value: 1 means yes, 0 means no. If yes, the sensor will return its data values with CRC characters added (this is handled and checked automatically by the driver).
<b>"Meas. Num:"</b>	String element (ST) address. This should be a single-character string; it is used to specify the measurement number. Valid values: 0-9

Table 1: SDI\_CMD\_RX TLI Parameters

### 2.2.2 Error Codes

Error Code	Name	Description
<b>0</b>	SDIE_SUCCESS	Successful command.
<b>101</b>	SDIE_INSUFFICIENT_API	(Not used for TLI status) Indicates that the SoftPLC version is out of date for the TLM and must be upgraded.
<b>201</b>	SDIE_OLD_REQUEST	The command was queued on a previous program run, ie: the runtime was transitioned to PROGRAM mode, then to RUN mode after this command was put in the queue.
<b>202</b>	SDIE_BAD_CMD_TYPE	The TLI's command type is not recognized by the TLM.
<b>203</b>	SDIE_NO_MEMORY	The TLM's FIFO queue for the TLI commands is full, so this command could not be added.

Error Code	Name	Description
<b>204</b>	SDIE_INVALID_ADDRS	1) The number of sensor addresses specified was invalid. 2) There was an invalid character specified as a sensor address. 3) There was a duplicate sensor address specified.
<b>205</b>	SDIE_INVALID_ADTL_NUM	1) The number of additional cmd numbers did not match the number of sensors. 2) An invalid character was specified as an additional cmd number.
<b>301</b>	SDIE_COMPORT_ERR	There was an error relating to the serial port used for SDI-12 communication.
<b>401</b>	SDIE_SENSOR_REPLY_NONE	The sensor did not reply within the timeout.
<b>402</b>	SDIE_SENSOR_REPLY_FAILED	The reply from the sensor was incomplete or mangled and could not be handled.
<b>403</b>	SDIE_SENSOR_REPLY_BADCRC	The CRC in the reply did not match the calculated CRC.
<b>404</b>	SDIE_SENSOR_REPLY_INVALID	The sensor's reply did not match the expected format.
<b>405</b>	SDIE_SENSOR_REPLY_NODATA	The sensor's data reply did not contain any values.
<b>406</b>	SDIE_SENSOR_SVCREQ_BAD	The sensor did not send a service request when expected, or the service request message was invalid. (Only relevant to 'M' or 'MX' command types)

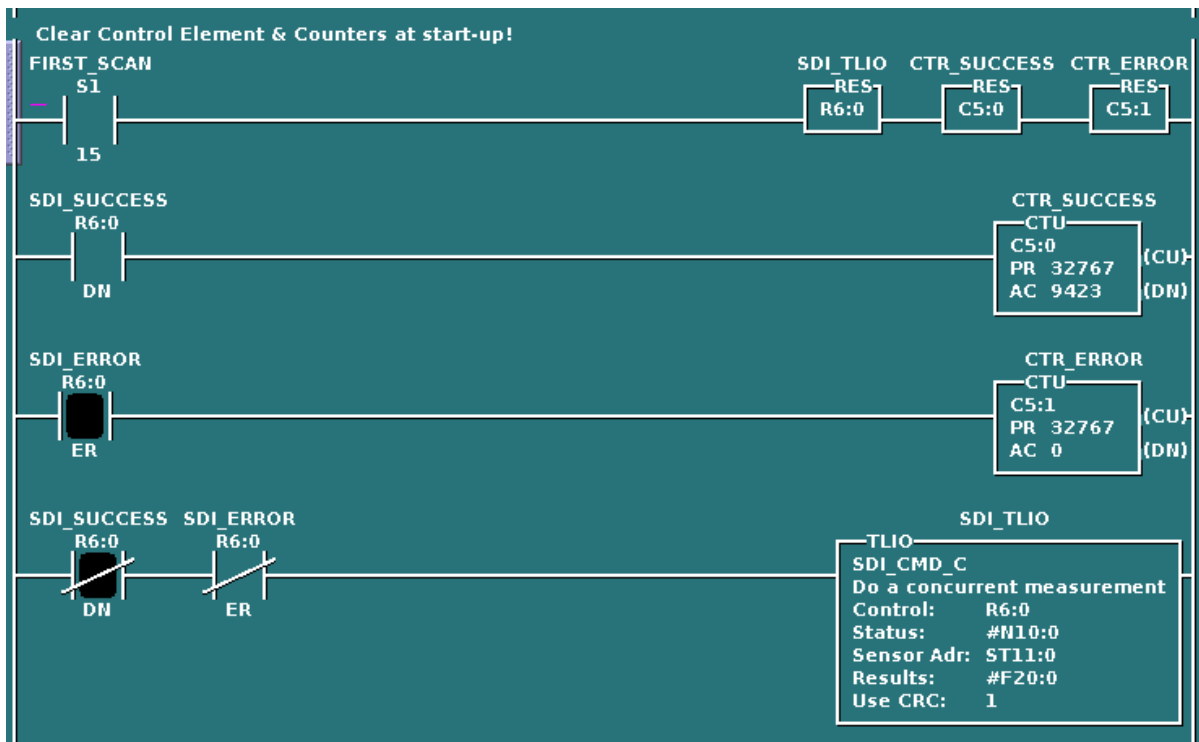
Table 1: SDI-12 TLI Error Codes

## 2.3 Usage

### 2.3.1 Ladder Instructions

Below is an example of a section of a ladder program using one of the SDI-12 TLIs. In this case, the TLI is queued as quickly as possible - the rung transitions to false on completion which resets the Control element, allowing a subsequent transition from false to true to queue the TLI command. The two rungs above that simply keep a count of successful or errored completions. The first rung is used to reset the Control and Counters on run mode transition from PROGRAM to RUN.

Other methods may be used to trigger the queueing of the TLI - the important detail is that the instruction queueing will only be triggered on the edge transition of the rung from false to true (and the instruction is not already queued or busy).



### 2.3.2 SDICmdTool

**SDICmdTool.py** is a command-line administrative/testing utility included with the SDI12 TLM. It can be used for initial configuration/validation of a sensor as well as manual testing of functionality. It also supports the "Transparent Mode" described by the SDI-12 protocol

specification, which allows manual entry of a command string (including extended/custom commands).

The utility is written in python and can be run after logging into the SoftPLC like so:

```
$ cd /home/root/SDI-12-CmdTool
$ python SDICmdTool.py
```

From there, the program has a help menu which lists the available commands and detailed text on each individual command.

### 2.3.3 simsensor

**simsensor** is a small utility program designed to (partially) simulate a SDI-12 sensor. It will respond only to measurement request commands (M, C, R) or data request commands (D) - for all commands the reply is a fixed value. When used as a simulator, this program is mainly useful for testing your use of TLIs in ladder (on another SoftPLC connected on the SDI-12 data line) against a known quantity.

Additionally, the program can be used as a SDI-12 network-wide listening utility. It prints out anything received to the console, regardless if the command or reply was directed at the simulator's sensor address.

Neither use of the program is exclusive of the other - you get both when you run the program, regardless. To start it:

```
$ cd /home/root/SDI-12-Sim
$ ./runsim address
```

Replace "address" with a valid SDI-12 sensor address that is NOT used by another sensor device/node in your system setup.

## 3 All