Open Architecture Control Software

SoftPLC DeviceNet I/O Driver for SST 5136-DNx Cards

Publication: SPZ-SSDN-MN1

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.

SoftPLC, and TOPDOC are registered trademarks of SoftPLC Corporation.

First Printing: July, 1998
Latest Printing: June, 2002

SoftPLC Corporation
25603 Red Brangus Dr.
Spicewood, Texas 78669 USA
Telephone: 512/264-8390 or 1-800-SoftPLC
Fax: 512/264-8399
WWW: http://www.softplc.com
Email: info@softplc.com

TABLE OF CONTENTS

1. INTRODUCTION 1

2. SOFTWARE INSTALLATION 1

3. HARDWARE INSTALLATION 2

4. DEVICE CONFIGURATION 2

5. DRIVER CONFIGURATION 4
Figure 1 - Loading the Driver 5
Figure 2 - Retrieving the Template Configuration File 5
5.1 [DRIVER] 7
5.2 [CARDS] 8
5.3 [DEVICES] 9
5.4 [INPUTS] and [OUTPUTS] 13

6. OPERATING MODES AND ERROR CODES 15

7. RESTARTING THE DRIVER 17

1. INTRODUCTION

SST is a Canadian company that makes a series of intelligent I/O scanner cards, which are sold through distribution. SoftPLC is a distributor of some of these cards. The SoftPLC driver for DeviceNet currently supports these cards from SST: 5136-DN, 5136-DNP, 5136-DN-104, 5136-DNP-104.

This DeviceNet I/O Driver for SoftPLC provides access to as many as 8 SST 5136-DN cards for ISA or PC/104 bus. Either the 8 bit cards or the 16 bit “PRO” cards may be used, in any combination. On each card there may be one DeviceNet with as many as 63 other devices.

The driver consists of these files: SSDEVNET.TLM, SSTDEVNET.LST, DNSCAN.SS1. and DNSCAN.SS2. SSDEVNET.TLM is the driver executable code, SSTDEVNET.LST is the configuration file, and DNSCAN.SS1 and DNSCAN.SS2 are card executable files.

SSDEVNET.TLM utilizes a common 16K memory window for all cards, no interrupts, and a block of 3 I/O ports unique for each card. Baudrates of 125K, 250K, and 500K are supported.

The SSDEVNET.TLM can co-exist with other SoftPLC I/O drivers concurrently. SoftPLC supports as many as 16 TLM’s any of which may be drivers.

The driver does not handle device configuration. You will need some other tool to perform that task. When SoftPLC uses the driver, the I/O devices are assumed to already be configured. The section on DEVICE CONFIGURATION discusses some of your options regarding device configuration tools.

2. SOFTWARE DRIVER INSTALLATION

All SoftPLC installations automatically install SSDEVNET.TLM, SSDEVNET.LST, DNSCAN.SS1, and DNSCAN.SS2. Separate installation is not needed for this driver.

3. DEVICE CONFIGURATION

You must use a third party DeviceNet configuration tool to establish node addresses and operating behavior for each device. There are several such software/hardware tools on the market: SST provides configuration software with the cards. DeviceNet Manager is available from Allen-Bradley, and there may be others. You might contact SST for yet other choices, or consult the ODVA website.

Since these tools generally run on a Windows system, you will either need an additional DeviceNet interface card in a desktop PC or you will have to move the card from your SoftPLC machine temporarily when configuring your devices. After configuration that card could be reinstalled in the SoftPLC machine. Since DeviceNet supports multiple masters, it is common to have a portable PC act as the DeviceNet configuration node, even while SoftPLC is attached. You might also consider having a spare SST DN card in the event of a failure, and to use that same spare for configuration.

4. HARDWARE INSTALLATION

Each SST DN card must be assigned a unique block of 8 I/O ports. This is done by setting the I/O port base starting address on each card from the following table. There is only one bank of switches on each card.

The memory address is a different setting from the I/O port address. Unlike the I/O port base address, which is set via hardware switches, the memory address is software selectable via the SSDEVNET.LST configuration file.

The extreme end of a DeviceNet needs a 120 ohm termination resistor. For more information on wiring devices, you should see the SST manual PC_HDW.PDF section 3.x. This manual is available from the SST website: http://www.sstech.on.ca electronically.

Also, your device vendor should have a **DeviceNet Installation and Planning Guide**.

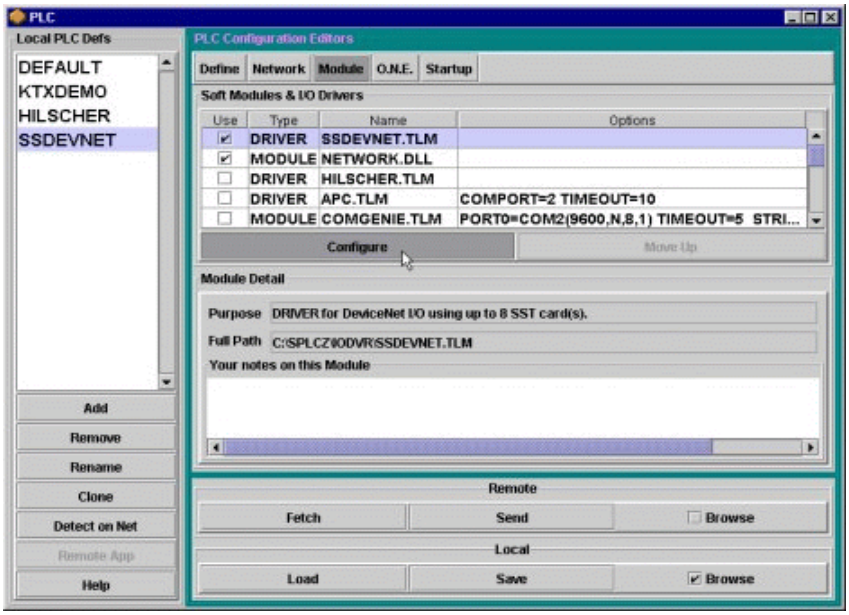
I/O Address	SW1 Setting						Comment
	1	2	3	4	5	6	
0x200	0	0	0	0	0	0	Game Port
0x208	1	0	0	0	0	0	
0x210	0	1	0	0	0	0	
0x218	1	1	0	0	0	0	
0x220	0	0	1	0	0	0	
0x228	1	0	1	0	0	0	
0x230	0	1	1	0	0	0	
0x238	1	1	1	0	0	0	
0x240	0	0	0	1	0	0	
0x248	1	0	0	1	0	0	
0x250	0	1	0	1	0	0	Default
0x258	1	1	0	1	0	0	recommended
0x260	0	0	1	1	0	0	recommended
0x268	1	0	1	1	0	0	recommended
0x270	0	1	1	1	0	0	
0x278	1	1	1	1	0	0	LPT2
0x280	0	0	0	0	1	0	
0x288	1	0	0	0	1	0	
0x290	0	1	0	0	1	0	
0x298	1	1	0	0	1	0	
0x2A0	0	0	1	0	1	0	
0x2A8	1	0	1	0	1	0	
0x2B0	0	1	1	0	1	0	
0x2B8	1	1	1	0	1	0	
0x2C0	0	0	0	1	1	0	
0x2C8	1	0	0	1	1	0	
0x2D0	0	1	0	1	1	0	
0x2D8	1	1	0	1	1	0	
0x2E0	0	0	1	1	1	0	
0x2E8	1	0	1	1	1	0	Com Port 4
0x2F0	0	1	1	1	1	0	
0x2F8	1	1	1	1	1	0	Com Port 2

5. DRIVER CONFIGURATION

Only after your devices are configured will you have sufficient information to configure the driver and run SoftPLC. So you must configure your devices first.

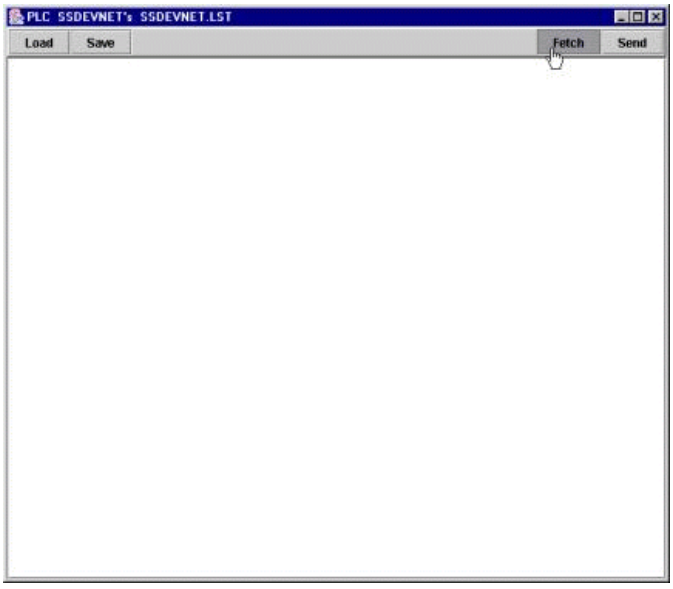
Use TOPDOC NexGen's Module Editor to load and configure the SSDEVNET.TLM for use with SoftPLC, as shown in Figure 1. Clicking Configure opens the driver configuration file, SSDEVNET.LST, for editing.

Figure 1 - Loading the Driver



When the blank text window has opened, click Fetch to retrieve a default SSDEVNET.LST file from the SoftPLC. Once you have made the edits, click Save to store the new file locally, and Send if you wish to store the edits as a default on the SoftPLC machine.

Figure 2 - Retrieving the Template File



After closing the text editor, click Save to store your TLM and DLL selections, and Send if you wish to store the selections on the SoftPLC.

Figure 3 shows the template SSDEVNET.LST file. This text file has 5 sections in it: [DRIVER], [CARDS], [DEVICES], [INPUTS], and [OUTPUTS].

Figure 3 - Sample SSDEVNET.LST File

```
;This is a SoftPLC configuration file for the SST DN 8 or 16 bit
;DeviceNet scanner card.  Anything to the right of a ; is a comment.
;Blank lines are ignored.

;The 5 sections [DRIVER], [CARDS], [DEVICES], [INPUTS], and [OUTPUTS]
;must always be present, and in this sequence.

;Up to 8 cards are supported.

;Documentation references:
;   SSDEVNET.PDF by SoftPLC Corp.
;   DNSCAN.PDF by SST


[DRIVER]
SAFEMODE=ZEROOUTPUTS
;INITEVENONFAILURE=NO      ;default is YES


[CARDS]
;Type
 ;      Port
 ;      ;       Memory(20 bit physical form)
 ;      ;       ;       BaudRate
 ;      ;       ;       ;       Watchdog(msecs)
 ;      ;       ;       ;       ;       ScanInterval(msecs)
 ;      ;       ;       ;       ;       ;       CardIndex (implied by row)
 SS1,   250,    d8000,  125,    200,    0       ;0
;SS2,   280,    d0000,  125,    500,    0       ;1


[DEVICES]
;CardIndex
 ;   MacID
 ;   ;   Vendor
 ;   ;   ;   DeviceType
 ;   ;   ;   ;   ProductCode
 ;   ;   ;   ;   ;   Flags(may use HEX with 0x prefix)
 ;   ;   ;   ;   ;   ;       Io1Interval
 ;   ;   ;   ;   ;   ;       ;   Output1Size
 ;   ;   ;   ;   ;   ;       ;   ;   Input1Size
 ;   ;   ;   ;   ;   ;       ;   ;   ;   Io2Interval
 ;   ;   ;   ;   ;   ;       ;   ;   ;   ;   Output2Size
 ;   ;   ;   ;   ;   ;       ;   ;   ;   ;   ;   Input2Size
 ;   ;   ;   ;   ;   ;       ;   ;   ;   ;   ;   ;   ExplicitSize(future)
 ;   ;   ;   ;   ;   ;       ;   ;   ;   ;   ;   ;   ;       comment
 0,  1,  0,  0,  0,  0x04,   0,  1,  1,  0,  0,  0,  0       ; prox switch
 0,  2,  0,  0,  0,  0x02,   0,  1,  1,  0,  0,  0,  0       ; D50
;0,  3,  0,  0,  0,  0x02,   0,  1,  1,  0,  0,  0,  0       ; D50


;These sections map the per device memory on the SST cards into
; the SoftPLC datatable.  The byteSwap column may be set to Y if
; you want each word in the block to have LSB and MSB exchanged.

[INPUTS]
;PLCmemAddr cardIndex   macID    byteSwap(Y/N)   comment
 I:2,       0,          1        N               ;prox switch
 I:1,       0,          2        N               ;D50

[OUTPUTS]
;PLCmemAddr cardIndex   macID    byteSwap(Y/N)   comment
 O:2,       0,          2        N               ;D50
```

5.1 [DRIVER]

This section contains parameter settings which affect the driver as a whole.

SAFEMODE=<safemode>

<safemode> can be either ZEROOUTPUTS or ZEROLENGTH. SAFEMODE specifies *what to do when entering PROGRAM and TEST modes of SoftPLC operation*.

The default is ZEROOUTPUTS. This means that all output devices will be sent zeros for all output data in these two modes PROGRAM and TEST. Presumably all digital outputs and all analog outputs will behave reasonably when receiving all zeros (probably turn off or go to the low end of a range).

An alternative, ZEROLENGTH, causes the driver to rely on the device to determine what is done in these two SoftPLC operating modes PROGRAM and TEST. ZEROLENGTH causes the driver to issue output messages of length zero to devices when in these two modes. A zero length output message has special significance to DeviceNet devices. Some output devices can be configured to do something other than turn “off” when receiving a zero length output message. This might be useful for analog outputs or digital outputs that you don’t want to be turned off when putting SoftPLC into PROGRAM or TEST modes. This setting places the burden on each device to “know what to do” when receiving zero length output messages. If an output device is not programmable with respect to “zero length message” behavior, then you should determine from the vendor what that behavior will be. Generally it may be: use last state, turn off, or go to a pre-determined value.

INITEVENONFAILURE=<yes or no>

<yes or no> can be either YES or NO. The default is YES, even if this option is not present.

The setting controls what to do if the driver fails to initialize. A setting of NO means the driver should prevent SoftPLC from starting up. A setting of YES means the driver should allow SoftPLC to startup, but the initial mode will be “FAULTED” and the I/O fault code will be in STATUS word 15, S:15. See the section on fault codes for the interpretation of this value.

5.2 [CARDS]

There may be up to 8 cards, with each card listed on its own row. In this section you list for each card:

Type

May be either SS1 (the 8 bit card) or SS2 (the 16 bit / PRO) card from SST.

Port

The I/O port may be any value in hex from the table given above with the switch settings. Each card must have a unique value.

Memory

The base physical memory address is given in 20 bit address form. *This is a 5 digit hex number.* For example D000:0000 would be given as D0000. D000:8000 would be given as D8000. All cards should be located in the same 16kbyte region, so provide the same value for all cards. The I/O ports are used to multiplex multiple cards into the same memory address space.

Baudrate

The baudrate for any card may be 125, 250, or 500. This number is assumed to mean "'k bits/sec".

WatchDog

This watchdog is a software mechanism implemented by the processor on an SST card. The SST watchdog feature keeps track of when the last time it has heard from the SoftPLC. If in RUN mode, and if the card has not heard from SoftPLC for this number of *milliseconds*, then all output devices are sent "zero length output messages". Output devices are generally designed to go to a safe position when receiving this message type. The recommended value is 500. The minimum value is 50 msecs, unless you are intending to turn off the watchdog altogether in which case you provide 0. Values between 0 and 50 are not supported.

ScanInterval

Specify the interval in msecs of the DeviceNet scan on the wire. This value specifies the minimum scan interval. If the actual minimum scan interval (due to network loading factors) is greater than the ScanInterval parameter, the ScanInterval parameter has no effect. The ScanInterval parameter can limit the rate of the I/O scan. It cannot speed up the I/O scan faster than network factors allow. A value of 0 means run as fast as you can and that is what is used most commonly.

5.3 [DEVICES]

There may be up to 8 x 63 devices, with each device listed on its own row. In this section you list for each device:

CardIndex

This is simply the 0 based index into the list of [CARDS]. The first card given in the [CARDS] section has a CardIndex of 0. The next has 1, etc.

MacID

Is a unique identifier 0-63 for each device.

Device Identification Parameters

This subset of device parameters: **VendorId**, **DeviceType**, and **ProductCode** are called device identification parameters.

If any of the identification parameters are non-zero, the scanner verifies the corresponding device identification attribute on startup. Any mismatch prevents connection to the device and reports the appropriate error at SoftPLC startup time. This will prevent SoftPLC from starting up, but will ensure that the devices on your network are what you expect.

If any of the identification parameters are zero, this indicates a “don’t care” for that parameter and causes the scanner to read that parameter from the device and to update the device list internally.

VendorId

All DeviceNet vendors have been assigned a vendor number. Or use 0 (zero) if you don’t want verification at startup.

DeviceType

The type of device assigned by the vendor, or 0.

ProductCode

The product code assigned by the vendor, or 0.

Flags

This is a bit mapped word given in hex format whose bits indicate what type of connection(s) to use to this device. Up to 2 I/O connections and one explicit messaging connection may be opened to each device.

Byte	Bit							
	7	6	5	4	3	2	1	0
0	Res.	AKS	CYC	COS	Res.	ST	P	EX
1	Reserved							G3

Group 2 only I/O Connection Flags:

AKS Acknowledge suppress
CYC Cyclic I/O Connection
COS Change-of-state I/O connection
ST Bit Strobed I/O Connection
P Polled I/O Connection

Explicit Connection Flags

EX Explicit Messaging Connection
G3 Group 3 Dynamic Explicit Messaging Connection only

The most common connection types are ST (bit strobed) and P (polled). Each device may have one or two I/O connection types. For example to set both the ST and P bits, you would OR together 0x04 and 0x02 yielding 0x06.

Explicit messaging is not supported at this time by this driver, even though the SST cards support it.

Here we show the relationships between Flags, Connection Types, and how the card uses the two I/O data areas for all card supported combinations of connection types. This table is taken from SST's DNSCAN.PDF, section 3.5.2.

Flags	Explicit	I/O 1	I/O 2
0x0001	G3 Dynamic	N/A	N/A
0x0002	No	G2 M/S Poll	N/A
0x0003	G2 M/S or G3 Dynamic	G2 M/S Poll	N/A
0x0004	No	G2 M/S Strobe	N/A
0x0005	G2 M/S or G3 Dynamic	G2 M/S Strobe	N/A
0x0006	No	G2 M/S Poll	G2 M/S Strobe
0x0007	G2 M/S or G3 Dynamic	G2 M/s Poll	G2 M/S Strobe
0x0010	No	G2 M/s COS	N/A
0x0011	G2 M/S or G3 Dynamic	G2 M/S COS	N/A
0x0012	No	G2 M/S Poll	G2 M/S COS
0x0013	G2 M/S or G3 Dynamic	G2 M/S Poll	G2 M/S COS
0x0014	No	G2 M/S Strobe	G2 M/S COS
0x0015	G2 M/S or G3 Dynamic	G2 M/S Strobe	G2 M/S COS
0x0020	No	G2 M/S Cyclic	N/A
0x0021	G2 M/S or G3 Dynamic	G2 M/S Cyclic	N/A
0x0022	No	G2 M/S Poll	G2 M/S Cyclic
0x0023	G2 M/S or G3 Dynamic	G2 M/S Poll	G2 M/S Cyclic
0x0024	No	G2 M/S Strobe	G2 M/S Cyclic
0x0025	G2 M/S or G3 Dynamic	G2 M/S Strobe	G2 M/S Cyclic
0x0050	No	G2 M/S COS / Ack	N/A
0x0051	G2 M/S or G3 Dynamic	G2 M/S COS / Ack	N/A
0x0052	No	G2 M/S Poll	G2 M/S COS / Ack
0x0053	G2 M/S or G3 Dynamic	G2 M/S Poll	G2 M/S COS / Ack
0x0054	No	G2 M/S Poll	G2 M/S COS / Ack
0x0055	G2 M/S or G3 Dynamic	G2 M/S Poll	G2 M/S COS / Ack
0x0060	No	G2 M/S Cyclic/Ack	N/A
0x0061	G2 M/S or G3 Dynamic	G2 M/S Cyclic/Ack	N/A
0x0063	G2 M/S or G3 Dynamic	G2 M/S Poll	G2 M/S Cyclic/Ack
0x0064	No	G2 M/S Strobe	G2 M/S Cyclic/Ack
0x0065	G2 M/S or G3 Dynamic	G2 M/S Strobe	G2 M/S Cyclic/Ack
0x0100	G3 Dynamic	N/A	N/A

Io1Interval

Represents the I/O interval in msecs between updates for this device on the first I/O connection. A common value is 0 (zero), meaning go as fast as you can. This field can be used in concert with the same field from other devices to implement a scan priority amongst all devices. Call SST for more information on this field.

Output1Size

Give the number of bytes of output assembly data for this first I/O connection.

Input1Size

Give the number of bytes of input assembly data for this first I/O connection.

Io2Interval

Represents the I/O interval in msecs between updates for this device on the second I/O connection. A common value is 0 (zero), meaning go as fast as you can. This field can be used in concert with the same field from other devices to implement a scan priority amongst all devices. Call SST for more information on this field.

Output2Size

Give the number of bytes of output assembly data for this second I/O connection.

Input2Size

Give the number of bytes of input assembly data for this second I/O connection.

ExplicitSize

(future. Set to zero now.)

5.4 [INPUTS] and [OUTPUTS]

SoftPLC's **Input Image** table consists of 8, 64, 128, or 512 16-bit words, depending on whether you have the 128(LT), 1K, 2K, or 8K kernel respectively. For example, 64 x 16 = 1024 bits, and 128 x 16 = 2048 bits. There is a like table for the **Output Image**. These I/O image tables support **I/O forcing**. *These are the only datatable sections which support I/O forcing.* I/O forcing is very useful for digital inputs or digital outputs. Forcing is not quite as useful for analog I/O.

The driver lets you put I/O data in any datatable region. However, I/O forcing is only supported if you use the Input Image table for inputs, and the Output Image table for outputs.

Consider the following sample [INPUTS] section

```
[INPUTS]
;PLCmemAddr cardIndex   macID     swapBytes(Y/N)
 I:2,       0,          1         N
 I:1,       0,          2         N
```

PLCmemAddr is the starting address of one or more words that will receive the input byte(s) for the device identified by cardIndex and macID. So macID indicates some input device, let's say a prox switch, and it is attached to card 0, and the configuration shows that all input bytes associated with that prox switch will be placed into word I2. The combination of cardIndex and macID point to a unique row in the [DEVICES] section.

If there are 2 I/O connections to the device, then the input data for I/O connection 1 will be first, followed immediately by the input data for I/O connection 2. I/O connection input data will always be word aligned. There is no way to pack connection 1 and connection 2 data into the same SoftPLC word.

If your device seems to be swapping the least significant byte for the most significant byte, you can counteract this by having the TLM swap each byte in the word. Enable this by placing a Y in the swapBytes column. A blank is the same as an N.

To determine the number of SoftPLC datatable words that will be consumed, take the parameters Input1Size and Input2Size which indicate a quantity of bytes. Round each of these counts up to the nearest power of 2. Then add them together. Then divide by 2 to give you the number of SoftPLC words:

numWords = (ROUND2(Input1Size) + ROUND2(Input2Size)) / 2;

For example, let's say the prox switch produces a single byte of input data on a single I/O connection. Rounding 1 byte up to nearest power of 2 yields 2. Since there's only one I/O connection to this device we can ignore Input2Size because rounding 0 to a nearest power of 2 yields 0. Dividing 2 by 2 yields 1 word. So our prox switch will consume one word of Input Image table, and likely the least significant bit of that word will contain the actual signal, although the device vendor documentation should always be consulted to verify the interpretation of input and output data.

The same formula that applies to inputs also applies to outputs. Again, output data for connection 1 is immediately followed by output data for connection 2 if it is used. The first word in this block is what is listed in the SSDEVNET.LST file.

Should you give an address such as N12:10 for either input or output data, then it is your obligation to use TOPDOC and create all referenced datatable words, including those that are implied by virtue of a block size exceeding 1 (one) word.

SSDEVNET.TLM will verify that your ladder program has all the necessary datatable words, and if not, it will prevent you from putting SoftPLC into RUN or TEST modes. However, SSDEVNET.TLM cannot create datatable words for you. That must be done by you using TOPDOC.

6. OPERATING MODES AND ERROR CODES

The driver's runtime behavior is determined by SoftPLC's mode of operation:

- When SoftPLC is in PROGRAM, REMOTE PROGRAM or FAULT mode, I/O is not active.
- When SoftPLC is in TEST mode, inputs are read from input devices and transferred to SoftPLC's datatable. Outputs are disabled, either set to “off” or to a position determined by each device. The SAFEMODE [DRIVER] parameter determines which of these two behaviors apply.
- When SoftPLC is in RUN or REMOTE RUN mode, inputs are read from input devices and transferred to SoftPLC's datatables. Outputs are written from SoftPLC's datatables to output devices.

Device Errors

In TEST or RUN modes, if any device should go inactive due to a wire or power problem, then that device number is recorded in the low byte of SoftPLC word S15, and the cardIndex will be recorded in the high byte. If you are using more than one card and could reasonably expect to get a cardIndex larger than zero, then be aware that the normal display of S15 in the status file is in decimal. To get a hex display you might want to MOV S:15 into a D datatable section word for display within the ladder.

If the device becomes available again, then the SST card attempts to reestablish communications with it every 10 seconds until it succeeds. Should communication be reestablished, this reporting mechanism will not clear S:15. You will need to do that with ladder after making a record of its non-zero value.

Card or Bus Errors

A Card or Bus error is more severe and means the entire DeviceNet comes down. If any one card goes into this mode, then SoftPLC will enter a FAULTED mode and the remaining cards, if any, will be taken offline. Status bit S11/6 will be set and word S:15 will be set to one of the following values:

S:15	Meaning
100	No card found at the given I/O port address.
101	Card firmware is wrong or not running.
102	Card is not timely responding to a command.
103	Card is indicating an undefined error.
104	Card firmware module id is a mismatch.
105	SSDEVNET.LST contains an illegal baudrate.
106	SSDEVNET.LST is not in same directory as SSDEVNET.TLM.
107	SSDEVNET.LST has a format error in it.
108	Card seems to have a memory conflict with other hardware.
110	Driver is out of memory.
111	Card decided to go offline due to excessive comm errors.
112	Driver cannot find DNSCAN.SS1 or DNSCAN.SS2.
113	No card found at I/O port indicated in SSDEVNET.LST.
114	RAM conflict at memory address given in SSDEVNET.LST.
115	Card memory is not storing.
116	Card firmware file DNSCAN.SS? has a checksum error.
117	Card flunked its own diagnostics routine. Probably a bad card.
118	Card did not acknowledge an interrupt properly.
119	One of the devices on the network did not come on line. Cabling?
120	Datatable memory indicated by SSDEVNET.LST is missing.
121	DNSCAN.SS? is truncated, shorter than expected.
122	Driver could not get input data locks on card.
123	Driver could not get output data locks on card.
124	Card issued a runtime exception and shut itself down.
125	Card is not updating the client status area.
126	Card is not updating the client status area.
127	Card’s CPU seems never to have started.
201-218	Subtract 200 from the value and look up the meaning in SST’s DNSCAN.PDF table 4.5.2. These are re-mapped SST command errors.

7. RESTARTING THE DRIVER

The driver goes through its entire INIT sequence, including loading SSDEVNET.LST every time it sees a REM/PROG to REM/PROG transition. You can initiate this sequence with TOPDOC using the F2 key while viewing ladder online. There should be no intervening mode between the REM/PROG to REM/PROG transition, and of course you will have to select REM/PROG twice in a row to generate this sequence.

Should you find a faulted SoftPLC, you can quickly clear the fault with TOPDOC by entering the online datatable editor, doing ALT_P to position on the STATUS file, then do an ALT_J to clear all faults. This will put you into REM/PROG mode, and then a subsequent single REM/PROG mode change will generate the REM/PROG to REM/PROG transition that will reinitialize the driver.

Note that since the driver reloads the SSDEVNET.LST file during initialization, this makes it possible to update this file using FTP and never have to shutdown a running SoftPLC session, you simply have to use TOPDOC to generate the “double clutch” REM/PROG to REM/PROG transition.

Most SoftPLC I/O drivers work this way too, so the same mode transition will reinitialize all I/O drivers in your system should you be using more than one concurrently.

Because it can take 12 seconds or more for the driver to go through its initialization sequence, TOPDOC will appear to be hung during this time since it is waiting for the driver to complete. It is common to see a TIMEOUT error during this time in TOPDOC. Since TOPDOC uses the ONE “longtimeout” rather than “normaltimeout” to do a mode transition, you can avoid this mode transition timeout by adjusting your longtimeout for the associated ONE channel upwards.