



TIMS-0102

USB to I2C/SPI INTERFACE MODULE

Operating Manual

<http://www.jovasolutions.com>

Model TIMS-0102

Part No. 910-0102

Published April 2006

Table of Contents

1.	<u>TIMS-0102 I2C AND SPI INTERFACE MODULE HARDWARE.....</u>	<u>4</u>
1.1	CONNECTOR DETAILS.....	5
1.1.1	I2C CONNECTOR.....	5
1.1.1.1	Mating Connectors	5
1.1.2	SPI CONNECTOR.....	6
1.1.2.1	Mating Connector.....	6
1.1.3	USB CONNECTOR.....	7
1.2	I2C SIGNALS.....	7
1.2.1	VOLTAGE LEVEL BUFFERING	7
1.2.2	PULL-UP RESISTORS.....	7
1.2.3	DRIVE CURRENT	7
1.2.4	CLOCK RATES.....	7
1.2.5	ADDRESSING.....	8
1.2.6	KNOWN LIMITATIONS.....	8
1.3	SPI SIGNALS.....	8
1.3.1	TRANSMIT (Tx) CLOCKING.....	8
1.3.2	RECEIVE (Rx) CLOCKING.....	8
1.3.3	DRIVE CURRENT	8
1.3.4	CLOCK RATES AND SPEED	8
1.3.5	CHIP-SELECT.....	10
1.3.6	MICROWIRE COMPATIBILITY.....	11
1.3.7	KNOWN LIMITATIONS.....	11
1.4	EXTERNAL +VREF	11
2.	<u>GENERAL SOFTWARE DESCRIPTION.....</u>	<u>12</u>
3.	<u>TIMS-0102 LABVIEW INSTRUMENT DRIVERS</u>	<u>14</u>
3.1	TIMS-0102 COMMUNICATIONS PORT.....	15
3.2	TIMS-0102 +5V POWER TO EXTERNAL CONNECTORS	16
3.3	TIMS-0102 I2C BASIC FUNCTIONS	16
3.3.1	I2C BUS CONFIGURATION FUNCTIONS.....	16
3.3.2	I2C SIMPLE I/O FUNCTIONS	17
3.3.3	I2C SIMPLE I/O EXAMPLES	17
3.4	TIMS-0102 I2C SCRIPTING FUNCTIONS	19
3.4.1	I2C SCRIPT BUILDING FUNCTIONS	19
3.4.2	I2C SCRIPTING EXECUTION FUNCTIONS	20
3.4.3	I2C SCRIPTING EXAMPLES	21
3.4.4	I2C SUBROUTINE-SPECIFIC ERRORS	22



	USB to I2C/SPI Interface Operating Manual
3.5	TIMS-0102 SPI FUNCTIONS..... 23
3.5.1	SPI DEVICE CONFIGURATION FUNCTIONS 23
3.5.2	SPI DEVICE SELECT..... 24
3.5.3	SPI I/O FUNCTIONS 25
3.5.4	SPI EXAMPLES..... 26
3.6	TIMS-0102 GENERAL FUNCTIONS 27
4.	<u>WARRANTY 28</u>
5.	<u>ABOUT JOVA SOLUTIONS, INC..... 28</u>
6.	<u>CONTACTING JOVA SOLUTIONS, INC. 28</u>
	<u>APPENDIX A 29</u>
	LABVIEW FUNCTION REFERENCE 29

INTRODUCTION

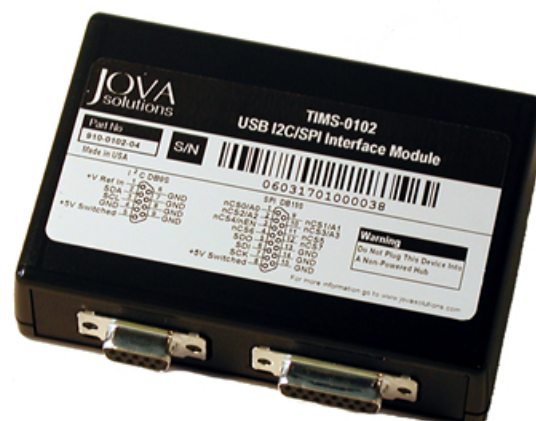
Jova Solutions' TIMS-0102 module offers a quick and easy method of communicating with I2C and SPI devices and circuits, and is intended for use in testing and prototyping devices and circuits that use the I2C or SPI protocol. TIMS-0102 modules are compact and completely self-contained, inexpensive, and connect to a host PC via USB.

The front panel contains both a DB-9S connector dedicated to I2C communications and a DB-15S connector dedicated to SPI or Microwire communications. The rear panel contains a single USB Type-B connector.

I2C buses are used extensively in a variety of microcontroller-based professional, consumer and telecommunications applications for control, diagnostics, and power-management. The I2C lines are buffered for mixed signal levels and the clock rate is programmable from 39.0625 kHz to 10 MHz. The TIMS-0102 operates as an I2C Master, only, with either 7-bit or 10-bit addressing.

The SPI interface is a serial interface in which a master device supplies clock pulses to exchange data serially with a slave over two data wires. In addition to the standard clock and data lines the TIMS-0102 comes with eight chip-select lines. These chip select lines can be optionally configured so that four lines are used as an address, with a separate chip-select line. The TIMS-0102 operates as an SPI Master only.

A complete LabVIEW driver library is provided for easy integration into higher level applications. A low-level Dynamic Link Library (DLL) is also provided for integration with other programming languages.



1. TIMS-0102 I2C AND SPI INTERFACE MODULE HARDWARE

The TIMS-0102 I2C and SPI interface module consists of electronic circuitry packaged in a 1.20"-high by 4.50"-wide by 3.25"-deep (including the connector) plastic enclosure. A single USB Type-B connector is located on the rear panel. The front panel contains both a DB-9S connector (dedicated for I2C communications), and a DB-15S connector (for SPI or Microwire communications).

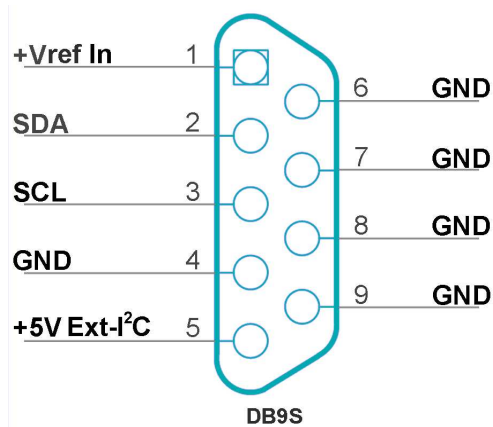
The functions of this module are:

- USB to I2C and USB to SPI communication between the host test controller and unit under test (UUT).
- I2C electrical buffer for mixed signal level use.
- Microwire compatible
- Programmable clock-rate.

1.1 CONNECTOR DETAILS

1.1.1 I2C Connector

The TIMS-0102 I2C connector DB-9S (Female) contains the I2C interface connection pins.



I2C DB-9S Pin Descriptions

Signal Name	Pin #	Description
+Vref In	1	User supplied I2C +V from target system to the P82B96 buffer
SDA	2	Serial Data Line – used to transfer data between transmitter and receiver
SCL	3	Serial Clock Line – used to synchronize communication
GND	4,6,7,8,9	Ground – connect to target system ground
+5V Ext – I2C	5	USB +5V through a MOSFET P-channel switch

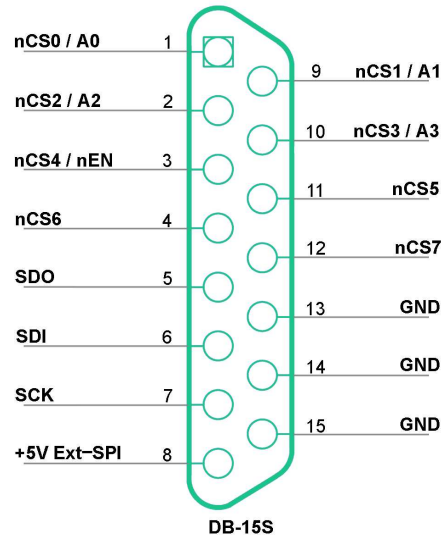
1.1.1.1 Mating Connectors

DB-9P (Male) Amp – Tyco Electronics Part No: 747904-2 or Digi-Key Part No: A2043-ND

USB to I2C/SPI Interface Operating Manual

1.1.2 SPI Connector

The TIMS-0102 SPI connector DB-15S (Female) contains the SPI or interface connection pins.



SPI and DB-15S Pin Descriptions

Signal Name	Pin #	Description
nCS0/A0	1	Not Chip Select 0 or LSB of 4-bit address
nCS2/A2	2	Not Chip Select 2 or bit 2 of 4-bit address
nCS4/nEN	3	Not Chip Select 4 or Not Enable
nCS6	4	Not Chip Select 6
SDO	5	Serial Data Output
SDI	6	Serial Data Input
SCK	7	Serial Clock (Master Mode Output / Slave Mode Input*)
+5V Ext-SPI	8	USB +5V through a MOSFET P-channel switch
nCS1/A1	9	Not Chip Select 1 or bit 1 of 4-bit address
nCS3/A3	10	Not Chip Select 3 or MSB of 4-bit address
nCS5	11	Not Chip Select 5
nCS7/nSS	12	Master mode Not Chip Select 7 output / Slave mode Not Slave Select input*
GND	13,14,15	Ground – connect to target system ground

* Slave mode will be supported in future firmware release.

1.1.2.1 Mating Connector: DB-15P (Male) Amp – Tyco Electronics Part No: 747908-2 or Digi-Key Part No: A2044-ND

1.1.3 USB Connector

The TIMS-0102 USB interface is a USB 1.1 and USB 2.0 Full Speed compliant interface, available through a USB Type-B connector on the rear panel. This device, when connected to user circuitry, could draw up to 500 mA. *Do not plug the TIMS-0102 into a non-powered hub.*

1.2 I2C SIGNALS

1.2.1 Voltage Level Buffering

The I2C SDA and SCL signals are buffered between the connected device and the TIMS-0102 module interface logic with a bi-directional bus buffer, Philips PN P82B96. This device is specifically designed to buffer I2C signals between different logic levels (2V – 15V).

NOTE: the user must supply the reference voltage to pin 1 of the I2C connector.

1.2.2 Pull-up Resistors

The TIMS-0102 does not provide internal pull-up resistors on the I2C SDA and SCL lines. It is imperative that the user provides pull-up resistors between the SDA and SCL signals and the +Vref voltage. The overall net current requirements based on all of the devices that will be connected to the SDA or SCL nodes, including the Philips P82B96 device, should be considered when selecting the pull-up resistor values.

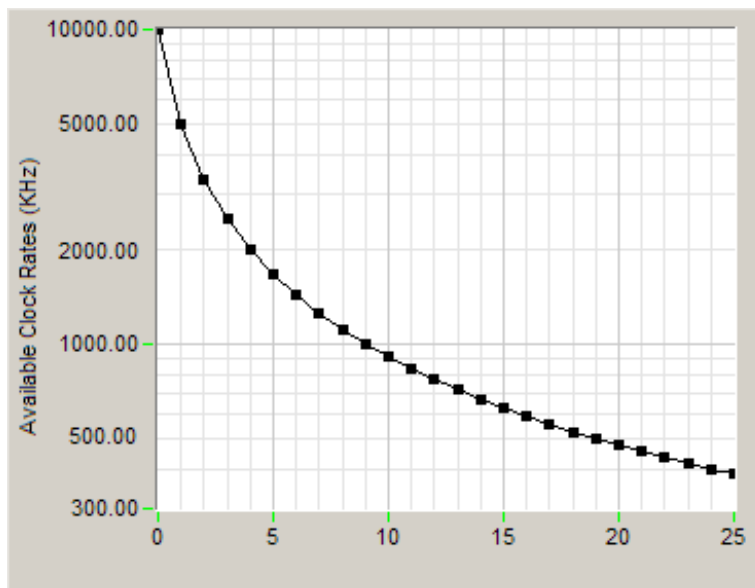
1.2.3 Drive Current

The Philips P82B96 buffer used in the TIMS-0102 has a 60 mA sink capability for driving low impedance buses.

1.2.4 Clock Rates

The TIMS-0102 supports the 100 KHz and 400 KHz I2C standard clock rates. 252 other clock speeds are available between 39.0625 KHz and 10 MHz, which can be used for non-standard communications, or for margin testing purposes.

NOTE: The I2C voltage level buffer, Philips P82B96 supports rates up to 400 KHz.



Available clock rates in the 400 KHz to 10 MHz range

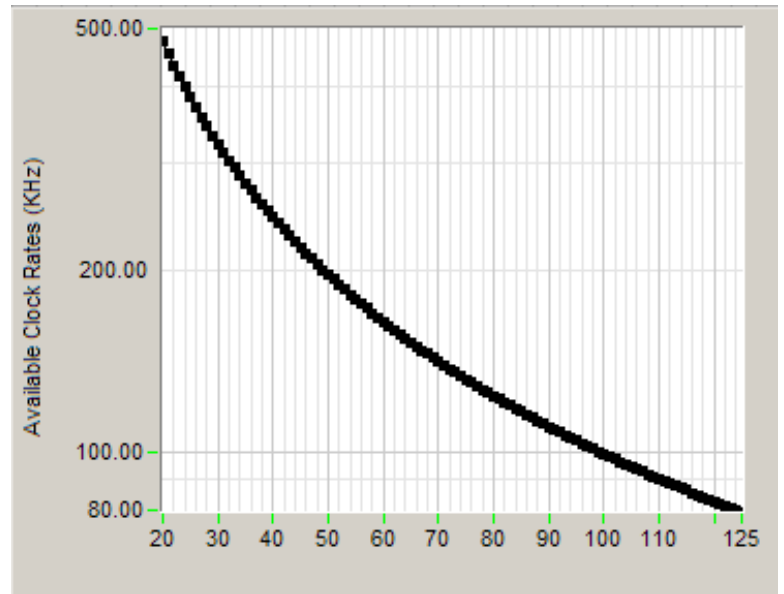
USB to I2C/SPI Interface Operating Manual

Overall command and data throughput to an I2C device from a host computer through the TIMS-0102 module can be affected by many variables, including setup time and the I2C protocol overhead (I2C requires that 9 bits be sent for every 8 bits of data).

The operating system also affect overall throughput. All devices on the USB bus share the overall bandwidth and the operating system must alternately service each device. Reducing the number of USB devices connected to the same root hub as the TIMS-0102 will reduce the possibility of performance degradation.

1.2.5 Addressing

The TIMS-0102 supports both 7 and 10-bit I2C device addressing.



Available clock rates in the 80 KHz to 500 KHz range

1.2.6 Known Limitations

The TIMS-0102 firmware version 1.0 supports only Master mode operation. Multi-master and slave mode operation will be supported in a future version of the firmware (field upgradeable).

Custom solutions are available by contacting Jova Solutions Sales at 1-866-755-1400.

1.3 SPI SIGNALS

The TIMS-0102 has configurable clock rates and supports several SPI modes through configurable Tx and Rx clocking parameters. The TIMS-0102 is Microwire compatible (see section 1.3.6)

1.3.1 Transmit (Tx) clocking

The Tx clock idle state is configurable to be high or low. Tx can also be configured for the falling or rising clock edge.

1.3.2 Receive (Rx) clocking

The Rx clocking can be configured so that it is sampled at the middle or end of the Tx output period.

1.3.3 Drive Current

The TIMS-0102 can drive all output signals with a maximum of 25 mA current source or sink.

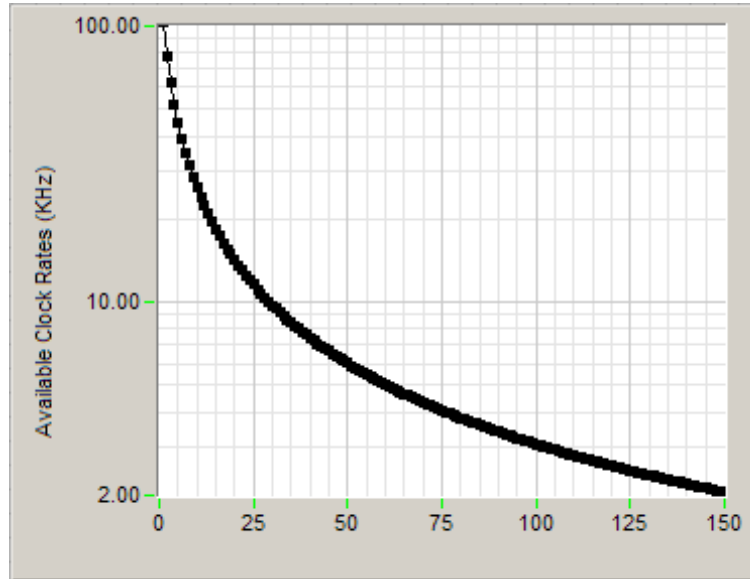
1.3.4 Clock Rates and Speed

The TIMS-0102 supports 766-clock rates between 1.44 KHz and 10 MHz. Three overlapping ranges, each with 255 supported clock rates, exist for SPI communications. The driver software simplifies

USB to I2C/SPI Interface Operating Manual

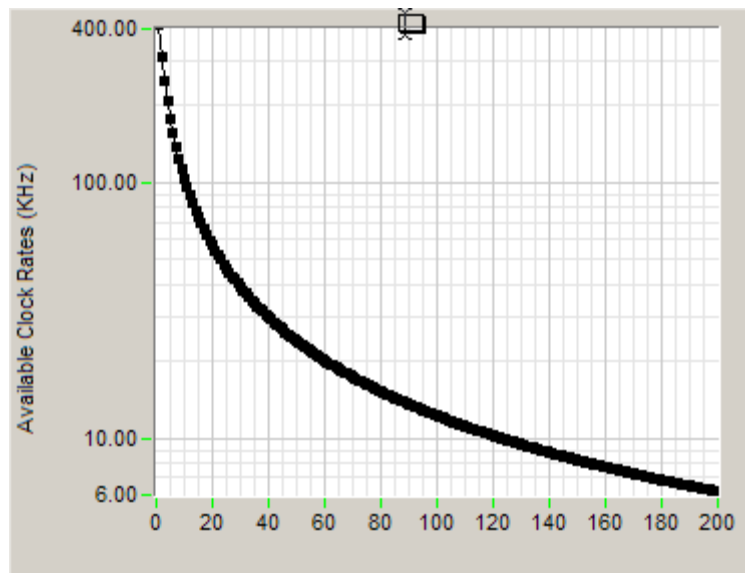
configuration by allowing “Target” clock rates to be specified. The actual clock rates are calculated and returned by the driver.

1.3.4.1 Range 1.



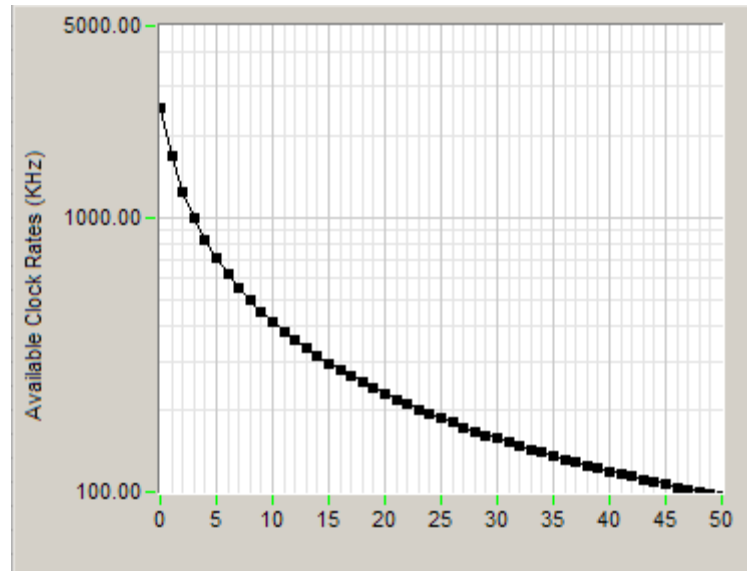
Available clock rates in the 2 KHz to 100 KHz range

1.3.4.2 Range 2.



Available clock rates in the 6 KHz to 400 KHz range

1.3.4.3 Range 3.



Available clock rates in the 100 KHz to 5 MHz range

Overall command and data throughput to an SPI device from a host computer through the TIMS-0102 module, can be affected by many variables including command setup time and USB bus overhead.

This table shows typical data byte rates for various clock speed settings. These are the real time interface data transfer rates between the TIMS-0102 and the SPI Device (USB overhead not measured).

Clock Speed	Data Byte Rate
10 MHz	333.33 Kbytes/sec
2.5 MHz	227.23 Kbytes/sec
1.0 MHz	111.11 Kbytes/sec

There is the additional overhead of sending commands and data over the USB bus between the Host computer and the TIMS-0102 module that will affect overall throughput from your application program. Individual commands will be executed at the hardware timed clock rates and actual data byte rates shown above but throughput for multiple commands will be much slower due to USB communications.

Reducing the number of USB devices connected to the same root hub as the TIMS-0102 will reduce the chances of performance degradation.

1.3.5 Chip-Select

Several modes of operation exist for performing Chip-select functionality.

Mode	Description
Mode 1	7 independent chip-select lines
Mode 2	Master mode operation. A 4-bit address and an nEN (Not Enable) are available for connection to the user's chip-select decode circuitry.
Mode 3	Slave mode where we receive an nSS (Not Slave Select) input.

1.3.6 Microwire Compatibility

The TIMS-0102 is compatible with the Microwire protocol (a subset of the SPI protocol), except that the TIMS-0102 uses dedicated Tx and Rx lines. The Tx line is a dedicated serial data-output line and the Rx is a dedicated serial data-input line.

1.3.7 Known Limitations

The TIMS-0102 firmware version 1.0 supports only Master-mode operation. Multi-master and slave mode operation will be supported in a future version of the firmware and will be field upgradeable.

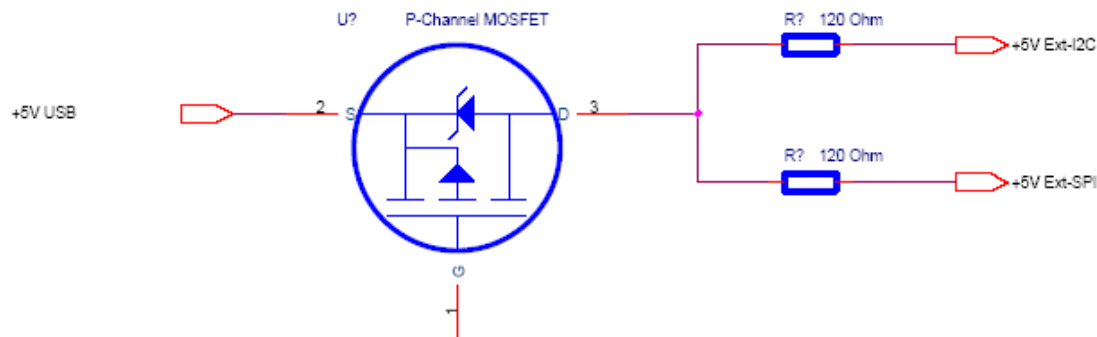
Devices connected to the TIMS-0102 SDI line must be tri-stated during I2C operation. SCK will be active during an I2C operation.

Our hardware is capable of supporting Slave-mode; the functionality will be added via a firmware field upgrade.

Custom solutions are available by contacting Jova Solutions Sales at 1-866-755-1400.

1.4 EXTERNAL +VREF

The +5V power is switched to the +5V connector pins on both the I2C and SPI connectors using the circuit diagrammed below.



NOTE: If a user is connecting both I2C and SPI connector signals to common user circuitry it is recommended that the user connect just one of the +5V outputs.

2. GENERAL SOFTWARE DESCRIPTION

The TIMS-0102 I2C and SPI Interface module is delivered with a full LabVIEW instrument driver is available for use in custom applications written using LabVIEW. A low level DLL interface is also available with a fully documented API for use in other applications.

This document is limited to the LabVIEW Driver library. Contact Jova Solutions for a copy of the TIMS Packet I/O Developers Manual for more information on the lower level interfaces available for integration with other programming languages.

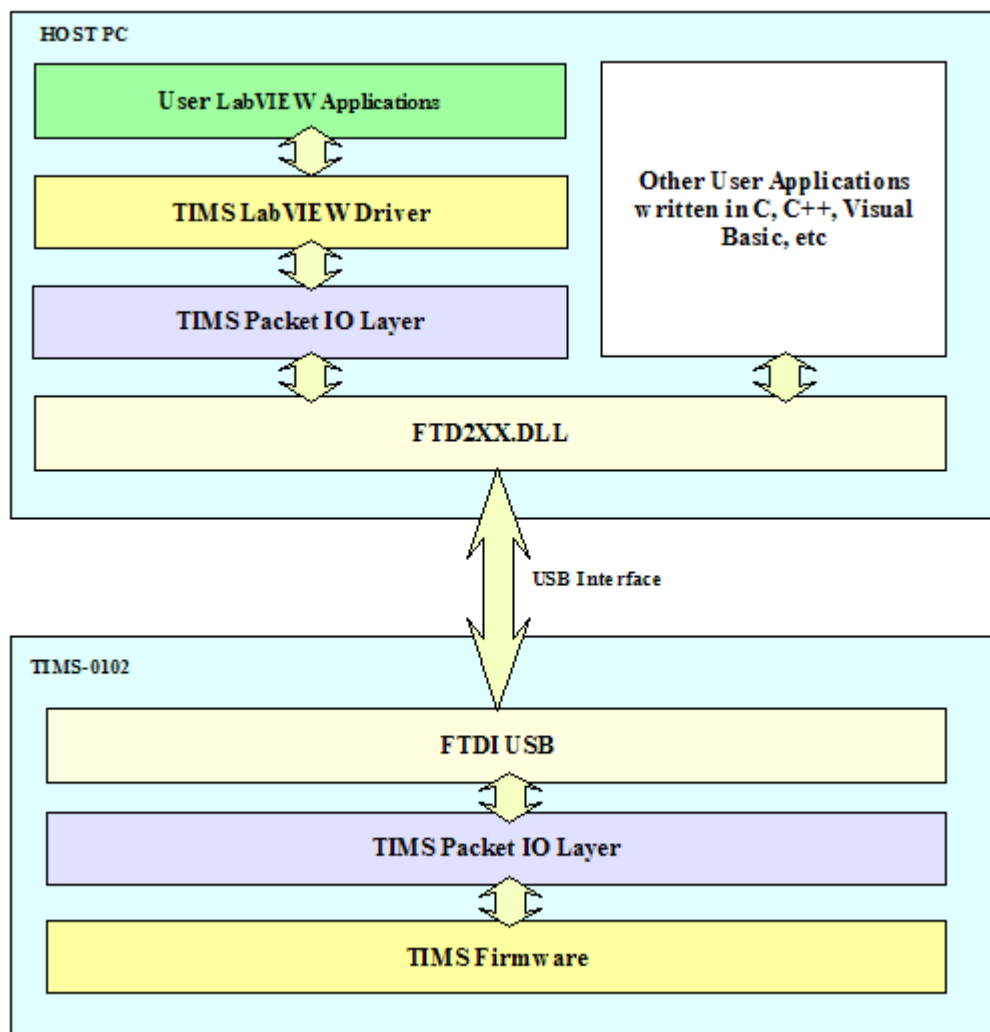


Figure 1 – Software Organization

USB to I2C/SPI Interface Operating Manual

The TIMS-0102 USB to I2C and SPI module utilized USB interface chips from FTDI. The FTD2XX.DLL is the lowest level software component on the Host PC and all communications to the TIMS-0102 goes through this DLL.

A complete LabVIEW Driver Library is provided for those who use LabVIEW, which eliminates the need to communicate directly with the DLL.

The complete low level packet communication protocol is also provided later in this section so that C, C++, Visual Basic, and other programmers can communicate with the TIMS-0102 directly through the FTD2XX.DLL.

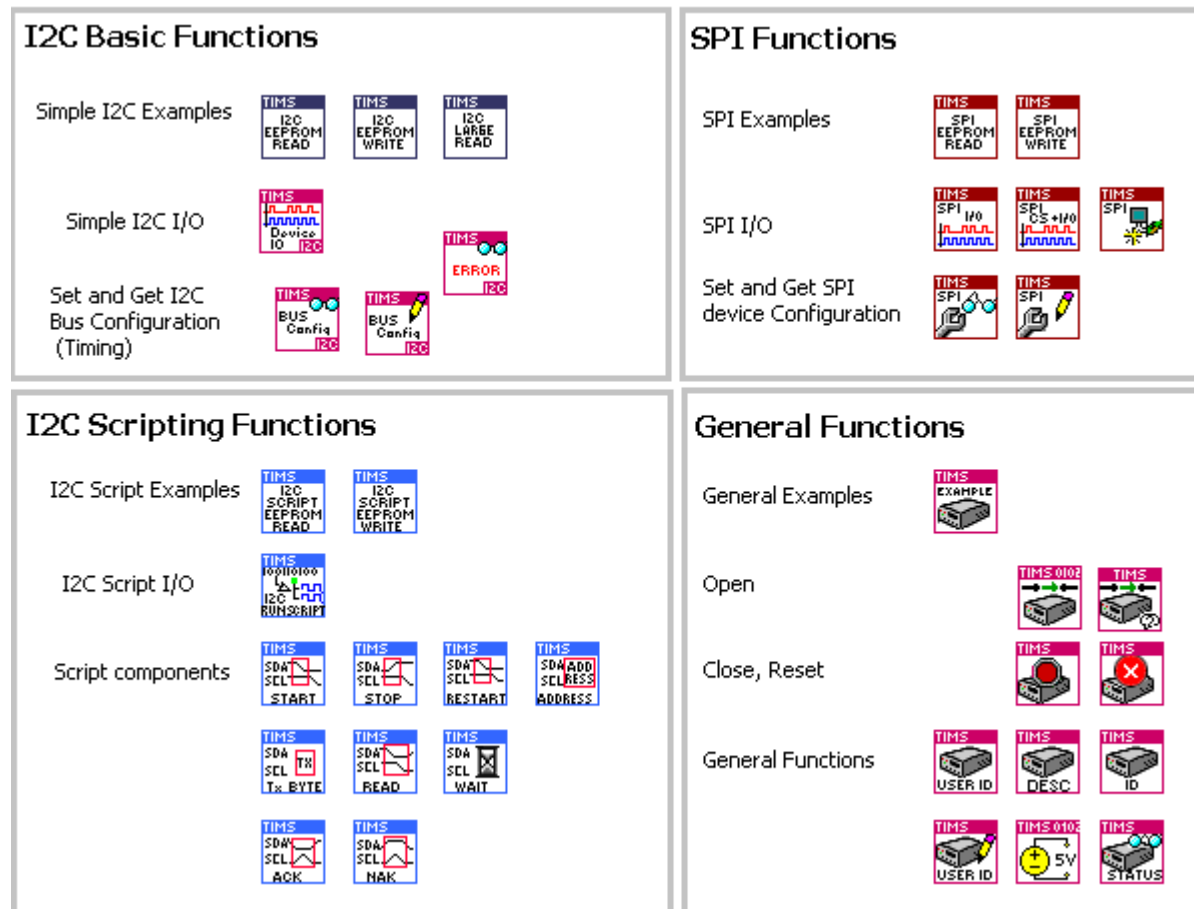
Each of these software layers is described in detail in the following sections.

3. TIMS-0102 LABVIEW INSTRUMENT DRIVERS

The LabVIEW drivers are installed, by default, into a TIMS-0102 directory. The driver library can be broken down into four functional groups:

- I2C Basic Functions
- I2C Scripting Functions
- SPI Functions
- General Functions

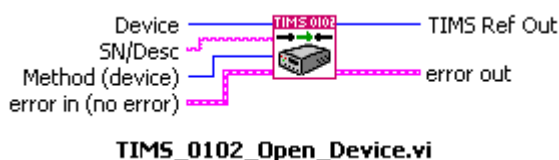
The TIMS0102_Tree.vi (installed to ...\\lib\\TIMS-0102\\TIMS0102_Tree.vi) diagram contains all the top-level functions needed for most applications as well as several examples.



3.1 TIMS-0102 COMMUNICATIONS PORT

A communications port must be opened through USB to the TIMS-0102 before any other driver function call. It is imperative that the port also be closed at the conclusion of a communications session. Leaving a port open will cause some function calls to error and, will also disallow any other communications ports to be opened.

The driver VI's TIMS_0102_Open_Device.VI and TIMS Port Close.vi are used for opening and closing communications ports to the TIMS-0102 module.

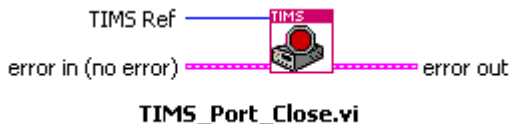


The TIMS_0102_Open_Device.VI creates a TIMS Reference, which is a reference that is used by all other VI's in the driver library to communicate to that specific device.

The TIMS Reference can be opened by Device number, TIMS Serial Number, or USB Description.

1. By device number: If you have only one TIMS unit connected to your system, this is the easiest method to use. Supply '0' to the Device number. If you have more than one device, you can access the devices by supplying their device number. NOTE: The windows OS is responsible for enumerating the devices, and therefore the assigned device numbers may change at any time. We do not recommend using the device number when you have more than one TIMS device attached to your system.
2. By TIMS Serial Number: You can open by TIMS Serial Number as well. This is the Serial Number printed on the label on the outside of your TIMS device.
3. By USB Description: If you have multiple models of TIMS units, you can access the unit by specifying the USB description. However, this method is not recommended if you have multiple units of the same model attached to the system. In this case the USB descriptions will be the same for similar models, and it is impossible to predict which specific device you will gain access to. The Device index input is an integer corresponding to the index of the device on the USB bus. The default value of 0 implies the use of the first TIMS-0102 device found.

Every application should start with this VI and conclude with a call to the TIMS Port Close.vi.



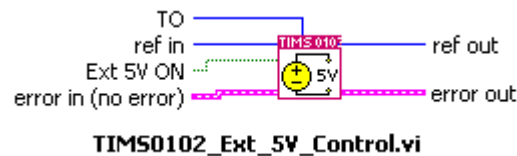
If the port has been opened, but has not been closed, another attempt to open the port will result in an error, "1, Invalid Handle". To recover from an orphaned reference, unplug and then reconnect the USB cable.

USB to I2C/SPI Interface Operating Manual

Once the TIMS port is open, communications with the TIMS module can take place. Configuration information can be read from or written to EEPROM memory in the TIMS module and I2C or SPI communications transactions can be performed to devices connected to the TIMS module.

3.2 TIMS-0102 +5V POWER TO EXTERNAL CONNECTORS

The +5V power pins on both the I2C and SPI connectors can be switched on or off using the TIMS0102_Ext_5V_Control.vi.



This single VI controls power to both the I2C and SPI connectors simultaneously.

3.3 TIMS-0102 I2C BASIC FUNCTIONS

The TIMS-0102 LabVIEW drivers provide several higher-level VI's to simplify I2C communications. These VI's work well, for the most common forms of I2C communications. I2C scripting is also available for the more complex or obscure forms of I2C (10-bit addressing, C BUS, etc). See Section 5.3, for details on I2C scripting functionality and usage.

3.3.1 I2C Bus Configuration Functions

The I2C bus communications configuration (clock and slew rates) is stored in EEPROM on the TIMS module. This bus-timing configuration is used for all I2C communications and is retained through power cycling. Two VI's are provided for reading or writing this configuration information.

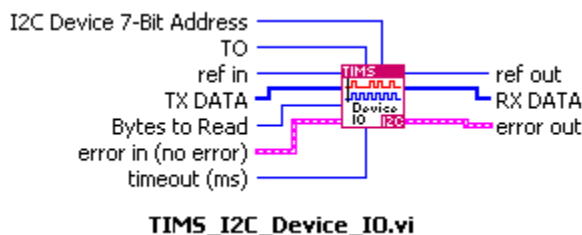
The I2C Bus Configuration is a cluster containing:

- **Target I2C Clock Rate (KHz) – DBL**
- The **Target I2C clock rate** can be programmed to be 100KHz, 400 KHz, 1MHz, and 252 other frequencies between 39.0625 KHz and 10 MHz.
- **Slew Rate – ENUM**
The **Slew Rate** can be set to **Disabled** for standard speed (100K or 1M) or **Enabled** for high speed (400K)

3.3.2 I2C Simple I/O Functions

In addition to the configuration VIs, simple I/O VI's are also provided.

The TIMS_I2C_Device_IO.vi performs a simple I2C I/O transaction to a 7-bit addressable device.



This VI can perform both a write and read operation. If the input TX DATA is empty the write operation is not performed. If the Bytes to Read input is set to 0, the read operation is not performed.

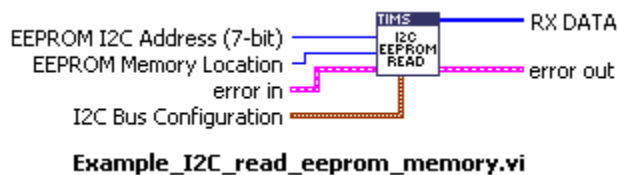
If the TX Data array input has data then this VI starts with a write operation. An I2C START bit is followed by an address byte (the R/W bit is automatically set or cleared by the firmware). The TX DATA bytes are then transmitted followed with an I2C STOP command. If the Bytes to Read input is set to a value other than zero a Read operation is performed. An I2C START command is sent followed by an address byte (with the R/W bit set to R) and then the requested number of bytes are received from the slave. The operation is completed with an I2C STOP command.

All read bytes are responded to with an ACK except for the last byte, which is responded to by a NAK.

There is no delay between the write operation and read operation. Some devices require processing time before responses can be generated (for example if you are writing to a memory location in the EEPROM you can not immediately read back the memory location without a small delay allowing the write cycle to complete). Use the I2C scripting function if you need more control over delay timing when executing commands.

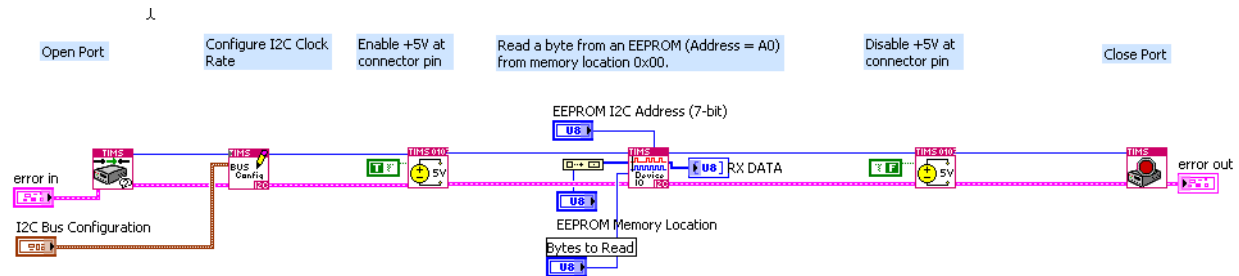
3.3.3 I2C Simple I/O Examples

Several examples that use the I2C Simple I/O VI's can be found in the TIMS-0102 examples directories. The first example, Example_I2C_read_eeprom_memory.vi is shown below:



USB to I2C/SPI Interface Operating Manual

This example VI shows the basics of I2C communications using the TIMS-0102 and the LabVIEW drivers.



The TIMS Port Open.vi and TIMS Port Close.vi are mandatory. Ports should not be left open indefinitely as some VIs return errors when connecting to open ports.

The TIMS Open Port.vi creates a reference that is used by all other VI's in the driver library. Every application should start with this VI and ensure that the TIMS Port Close.vi is also called once for every port open command.

If the port has been opened and not closed another attempt to open the port will result in an error 1, Invalid Handle.

Once the TIMS port is open, communications with the TIMS module can take place and normally the I2C bus configuration is set, as in the example, to specify the desired clock and slew rates. These settings are stored in the TIMS-0102 EEPROM and are used for all I2C communications until the configuration is set again.

The +5V power pins on both the I2C and SPI connectors can be switched on or off using the TIMS0102_Ext_5V_Control.vi

This example uses the TIMS_I2C_Device_IO.vi easy IO vi to send a command to and read a memory location from and EEPROM I2C device.

3.4 TIMS-0102 I2C SCRIPTING FUNCTIONS

In addition to the simple I2C I/O functions, the TIMS-0102 also allows for I2C scripting to handle the more complex or obscure communication protocol requirements. I2C scripting allows the user to build an I2C communication transaction out of the basic I2C protocol components. The script is sent to the TIMS module that then executes the script at hardware speeds. This gives you all the flexibility of using LabVIEW on a host computer to generate complex communications protocols yet ensures that the protocol is executed at the desired constant hardware generated clock rate.

This scripting mechanism allows the user to implement some other protocols not directly supported by the standard driver. C Bus and I2C 10-bit addressing are examples of functionality that can be built using the scripting functions.

3.4.1 I2C Script Building Functions

An I2C script is a string made up of a series of characters, each character representing an I2C communications protocol basic component. Script character codes are case sensitive. The script string is a simple string and standard LabVIEW string VI's which can also be used to build the script string. The VI's shown in the table below can also be used to assemble the script sting.

Script Character	I2C Communication Component	LabVIEW VI
S	START	Script input  Script output TIMS_I2C_Script_Start.vi
P	STOP	Script input  Script output TIMS_I2C_Script_Stop.vi
R	RESTART	Script input  Script output TIMS_I2C_Script_Restart.vi
K	ACK	Script input  Script output TIMS_I2C_Script_ACK.vi
N	NAK	Script input  Script output TIMS_I2C_Script_NAK.vi

USB to I2C/SPI Interface Operating Manual

?	Read Byte	Script input  Script output TIMS_I2C_Script_Read_Byte.vi
@	Wait 1-2 msec	Script input  Script output Number of waits (1-2msec pe...) TIMS_I2C_Script_Wait.vi
00-FF	Tx Hex Data	Script input  Script output Data Bytes TIMS_I2C_Script_Tx_Byte.vi

NOTE: Data sent using I2C scripting requires that the data be sent as 2 hex characters for every byte of data. All address bytes also need to be sent through the script as a 2-character hex block.

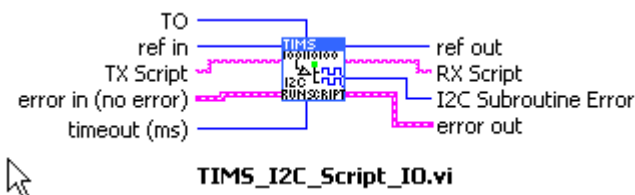
An additional VI is provided to add a 7-bit address (sent as a 2-character Hex string).



This VI generates the 2-character Hex block representing the address, with the least significant bit (lsb) set to reflect the Read/Write bit input.

3.4.2 I2C Scripting Execution Functions

A single VI, the TIMS-0102_I2C_Script_IO.vi, provides the script download and execution functionality.



This VI takes a Tx Script transmit string input, downloads the script to the TIMS-0102, executes the script, and returns a RX Script return sting containing the data and any errors encountered in the execution of the script. Even though Tx script strings may be up to 256 characters in length the response generated from the script execution cannot exceed 256 bytes or an error will be generated.

The Rx Script return string contains the original script with extra characters inserted the contain errors or response data bytes from the slave. If an error occurs during execution of the script an "!" exclamation point is placed in the string where the error occurred. An error will terminate the I/O operation with a forced STOP condition.

USB to I2C/SPI Interface Operating Manual

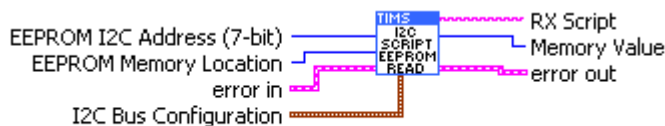
NOTE: Any queries made using the I2C scripting function may return more data bytes than are sent.

The scripting function will only return 256 bytes of data, any other bytes are discarded. If the returned data from a script exceeds 256 bytes, there will be no error code returned. However, any code should check for this condition and call the **TIMS_Status** function immediately and inspect [bit1] **Tx Overflow**. If this bit is true, the function attempted to return more than 256 bytes.

This issue is known, and will have an associated error code/warning generated in future versions of the TIMS driver.

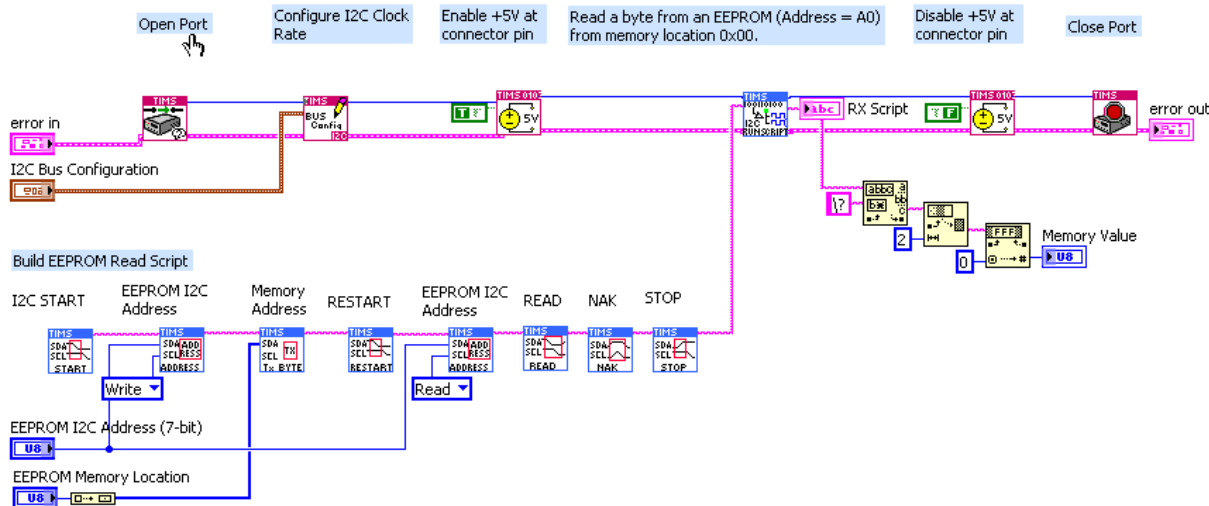
3.4.3 I2C Scripting Examples

Several examples that use the I2C Scripting VI's can be found in the TIMS-0102 examples directories. The first example, `Example_I2C_script_read_eeprom_memory.vi` is shown, below. This achieves the same overall functionality as the `Example_I2C_read_eeprom_memory.vi` that uses the simple I/O functions.



Example_I2C_script_read_eeprom_memory.vi

The example VI diagram shows the build up and execution of the script as well as the parsing of the Rx script string.



The lower portion of the diagram shows the building of the script. The final script sent to the script I/O is:

SA000RA1?NP

This TX script string translates to:

1. S Issue an I2C start bit

2. A0 Address the device A0 for writing
3. 00 Select EEPROM memory location 0 (0x00)
4. R Issue a RESTART bit
5. A1 Address the device A0 for reading
6. ? Read a byte
7. N Issue a NAK bit
8. P Issue a STOP bit

As you can see, with scripting, you have much more control over how the individual communication blocks are assembled. With this flexibility, comes the extra work of building up complete I/O transactions. The simple I/O example uses at least eight fewer function calls.

3.4.4 I2C Subroutine-Specific Errors

You may receive the **5341 I2C Function Error**. This is a general error code used to describe errors specific to the I2C bus, possibly due to a malfunctioning TIMS device, bad bus parameters, etc.

If you receive this error code, you should immediately call the **fnc_I2C_FUNCTION_ERROR** function. This function will then return the last I2C Function Error, so that you can get a better idea of what has occurred. Please see the **TIMS_Error_Codes** document for more detail about these codes.

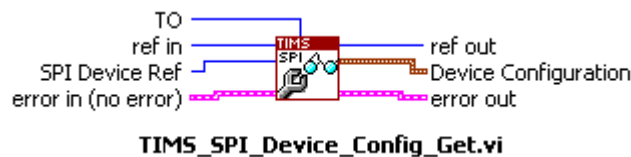
3.5 TIMS-0102 SPI FUNCTIONS

The TIMS-0102 LabVIEW drivers provide several VI's to simplify SPI communications.

3.5.1 SPI Device Configuration Functions

Unlike I2C, individual SPI devices operating on the same bus can all be operating at different clock rates and in different clock edge configurations. Because of this difference, the TIMS-0102 allows up to 16 different SPI device configurations to be stored in EEPROM (compared to just one setting used for all I2C devices). You only need to send an SPI device configuration once and it will be used for all subsequent operations directed at that particular SPI device. The TIMS-0102 will automatically switch to pre-configured settings stored for the particular SPI device addressed so that you don't have to keep sending clock configuration settings every time you address a different device.

The SPI device configurations are retained through power cycling. Two VI's are provided for reading or writing this configuration information.



The SPI Device Ref input must be an integer number between 1 and 16.

The SPI Device Configuration is a cluster containing:

- **Tx Clocking – ENUM**
 - Clock Idle Low, TX Falling Edge
 - Clock Idle High, TX Falling Edge
 - Clock Idle Low, TX Rising Edge
 - Clock Idle High, TX Rising Edge
- **Rx Clocking – ENUM**
 - Rx Sampled at Middle of TX Output Period
 - Rx Sampled at End of TX Output Period
- **Target Clock Rate – DBL Float**
 - Target clock rate must be in range from 39.0625 KHz to 10 MHz.

The second VI is for reading a SPI device configuration.

3.5.2 SPI Device Select

The TIMS-0102 comes with eight programmable device or chip select lines that are available at the output SPI DB-15F connector. These chip select lines can be independently selected or used to generate a combination of a 4-bit address and single chip select line.

The SPI Device Select is an ENUM (enumerated constant) with values of:

- ADR x0, nCS4
- ADR x1, nCS4
- ADR x2, nCS4
- ADR x3, nCS4
- ADR x4, nCS4
- ADR x5, nCS4
- ADR x6, nCS4
- ADR x7, nCS4
- ADR x8, nCS4
- ADR x9, nCS4
- ADR xA, nCS4
- ADR xB, nCS4
- ADR xC, nCS4
- ADR xD, nCS4
- ADR xE, nCS4
- ADR xF, nCS4
- nCS0
- nCS1
- nCS2
- nCS3
- nCS4
- nCS5
- nCS6
- nCS7
- No Select

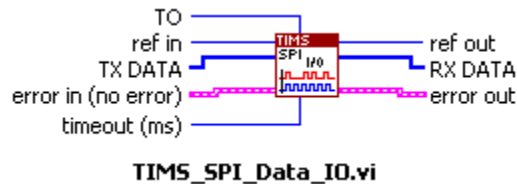
When ADRx0, nCS4 through ADRxF, nCS4 is selected the corresponding SPI Device configuration is recalled and used in the following transaction. The address lines are updated with the requested address and then the nCS4 (i.e., nEN Not Enable) is set low.

If nCS0 through nCS7 is selected then SPI Device Configurations 0 through 7 is recalled and used in the following transaction. The requested line, nCS0 – nCS7, is set low.

Selecting the No Select option sets all line high.

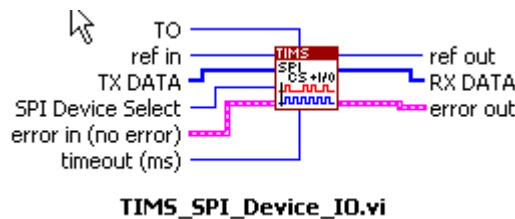
3.5.3 SPI I/O Functions

The TIMS-0102_SPI_Data_IO.vi performs a basic SPI communication transaction. This VI does not affect the device select lines; they must be separately enabled.



This VI is intended to be used when more than 256 Bytes need to be read or sent out between chip select enabled and chip select disabled, as in the following VI. Call this vi in a loop surrounded by calls to TIMS_SPI_Device_Enable.vi.

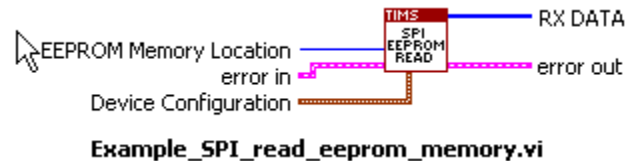
A second SPI VI, the TIMS0102_SPI_Device_IO.vi can also be used for performing a basic SPI communications transaction.



The TIMS0102_SPI_Device_IO.vi performs the chip select operation, a data I/O operations up to 255 bytes, then a chip deselect operation.

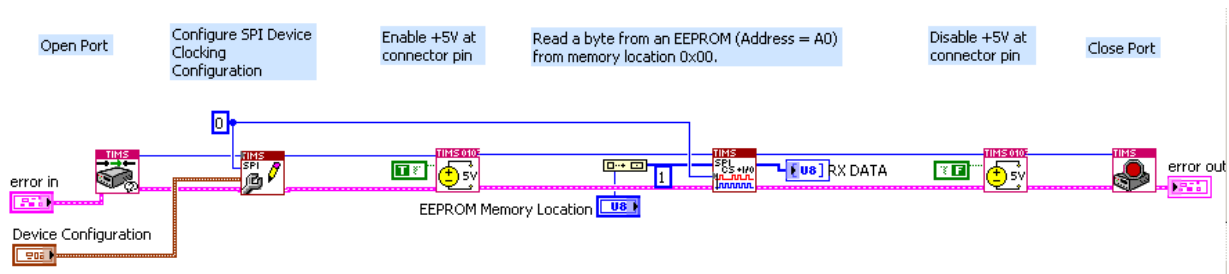
3.5.4 SPI Examples

Several examples that use the SPI I/O Function VI's can be found in the TIMS-0102 examples directories. The first example, Example_SPI_read_eeprom_memory.vi is shown below:



The diagram for this VI is shown below.

This example VI shows the basics of SPI communications using the TIMS-0102 and the LabVIEW drivers.



The TIMS Port Open.vi and TIMS Port Close.vi are mandatory and Ports should not be left open indefinitely as some VI's cannot execute on open ports.

The TIMS Open Port.vi creates reference that is used by all other VI's in the driver library. Every application should start with this VI and ensure that the TIMS Port Close.vi is also called once for every port open command.

If the port has been opened and not closed, another attempt to open the port will result in an error 1, Invalid Handle.

Once the TIMS port is open, communications with the TIMS module can take place and normally SPI Device configurations are set, as in the example, to specify the desired clock rates and edge settings. These settings are stored in the TIMS-0102 EEPROM and are used for all I2C communications until the configuration is set again.

The +5V power pins on both the I2C and SPI connectors can be switched on or off using the TIMS0102_Ext_5V_Control.vi

3.6 TIMS-0102 GENERAL FUNCTIONS

The VI's in the General Functions section are related to TIMS-0102 module functions not specifically I2C or SPI. These functions include opening and closing a communications port to the TIMS-0102 module, scanning the USB bus for TIMS modules, and reading configuration information from an a TIMS-0102 EEPROM.

Every application should start with TIMS_0102_Device_Open.vi and conclude with a call to the TIMS Port Close.vi.

If the port has been opened, but is not closed, another attempt to open the port will result in an error 1, Invalid Handle.

Once the TIMS port is open, communications with the TIMS module can take place. Configuration information can be read from or written to EEPROM memory in the TIMS module and I2C or SPI communications transactions can be performed to devices connected to the TIMS module.

4. WARRANTY

Jova Solutions, Inc. warrants its products for a period of one year from date of delivery against defects in material or workmanship. Returned product will either be repaired or replaced at the option of Jova Solutions, Inc.

Specifications are subject to change without notice.

5. ABOUT JOVA SOLUTIONS, INC.

Founded in 1998, Jova Solutions, Inc. provides software and services for industrial automation and interactive data networking. Jova Solutions, Inc. focuses on industry-specific solutions for Pharmaceuticals and Biotechnology, Electronics Manufacturing, Networking, and Telecommunications.

Jova Solutions, Inc. is a privately held company with headquarters in San Francisco, CA.

6. CONTACTING JOVA SOLUTIONS, INC.

To find out more about the TIMS-0102, please contact us at: 1-800-755-1400

For information on our products and services, please visit our website at: www.jovasolutions.com.

For technical support issues, please contact us at 1-415-348-1408

Appendix A

LabVIEW Function Reference

Example_I2C_Large_Block_Read_Write.vi

This example VI shows the basics of I2C communications using a TIMS I2C Device and the LabVIEW drivers.

The TIMS_Open_Dialog.vi and TIMS_Port_Close.vi are mandatory and Ports should not be left open indefinitely as some VI's can not execute on open ports.

The TIMS_Open_Dialog.vi creates a reference used by all other VI's in the driver library. Every application should start with this VI and ensure that the TIMS_Port_Close.vi is also called once for every port open command.

If the port has been opened and not closed another attempt to open the port will result in an error 1, Invalid Handle.

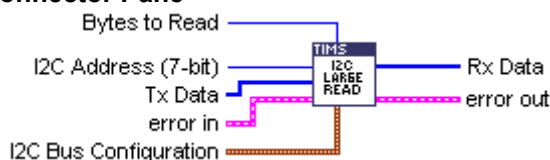
Once the TIMS port is open, communications with the TIMS module can take place and normally the I2C bus configuration is set, as in the example, to specify the desired clock and slew rates. These settings are stored in the TIMS I2C Device EEPROM and are used for all I2C communications until the configuration is set again.

The +5V power pins on both the I2C and SPI connectors can be switched on or off using the TIMS0102_Ext_5V_Control.vi

This example uses the TIMS_I2C_Device_IO.vi easy IO vi to send a command to and read or write a large amount of data to a memory location from an EEPROM I2C device. TIMS_I2C_Device_IO.vi only allows you to read a maximum of 255 bytes or write a maximum of 254 bytes of data at a time. This example calls TIMS_I2C_Device_IO.vi in a loop to write or read a large block of data. If Tx Data is empty, then the write operation is not done. If Bytes to Read is zero, then the read operation is not done.

=====
© Jova Solutions, Inc

Connector Pane



Controls and Indicators

 I2C Address (7-bit)

 Tx Data

 FROM

USB to I2C/SPI Interface Operating Manual



error in The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



I2C Bus Configuration



Slew Rate



Target I2C Clock Rate (Khz)



Timeout (ms)



Bytes to Read



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

USB to I2C/SPI Interface Operating Manual



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



Rx Data



FROM

Example_I2C_read_eeprom_memory.vi

This example VI shows the basics of I2C communications using a TIMS I2C Device and the LabVIEW drivers.

The TIMS_Open_Dialog.vi and TIMS_Port_Close.vi are mandatory and ports should not be left open indefinitely as some VI's can not execute on open ports.

The TIMS_Open_Dialog.vi creates a reference used by all other VI's in the driver library. Every application should start with this VI and ensure that the TIMS_Port_Close.vi is also called once for every port open command.

If the port has been opened and not closed another attempt to open the port will result in an error 1, Invalid Handle.

Once the TIMS port is open, communications with the TIMS module can take place and normally the I2C bus configuration is set, as in the example, to specify the desired clock and slew rates. These settings are stored in the TIMS I2C Device EEPROM and are used for all I2C communications until the configuration is set again.

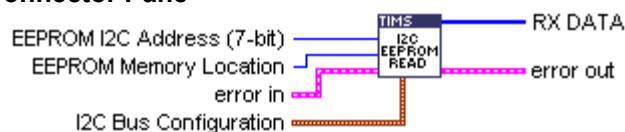
The +5V power pins on both the I2C and SPI connectors can be switched on or off using the TIMS0102_Ext_5V_Control.vi

This example uses the TIMS_I2C_Device_IO.vi easy IO vi to send a command to and read a memory location from and EEPROM I2C device.

=====

© Jova Solutions, Inc

Connector Pane



Controls and Indicators



EEPROM I2C Address (7-bit)



EEPROM Memory Location

USB to I2C/SPI Interface Operating Manual



error in The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**I2C Bus Configuration**

Slew Rate



Target I2C Clock Rate (Khz)



Timeout (ms)

**RX DATA**

FROM



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

Example_I2C_script_read_eeprom_memory.vi

This example VI shows the basics of I2C communications using the TIMS I2C Device and the LabVIEW I2C scripting drivers.

The TIMS_Open_Dialog.vi and TIMS_Port_Close.vi are mandatory and Ports should not be left open indefinitely as some VI's can not execute on open ports.

The TIMS_Open_Dialog.vi creates the reference (TIMS0102_Handle) used by all other VI's in the driver library. Every application should start with this VI and ensure that the TIMS_Port_Close.vi is also called once for every port open command.

If the port has been opened and not closed another attempt to open the port will result in an error 1, Invalid Handle.

Once the TIMS port is open, communications with the TIMS module can take place and normally the I2C bus configuration is set, as in the example, to specify the desired clock and slew rates. These settings are stored in the TIMS I2C Device EEPROM and are used for all I2C communications until the configuration is set again.

The +5V power pins on both the I2C and SPI connectors can be switched on or off using the TIMS0102_Ext_5V_Control.vi

This example uses the TIMS_I2C_Script_IO.vi to send a script to the TIMS I2C Device module and execute that script to read a memory location from an EEPROM I2C device.

An I2C script is a string made up of a series of characters, each character representing an I2C communications protocol basic component. Script character codes are case sensitive.

S - Start
P - Stop
R - Restart
K - ACK
N - NAK
? - Read Byte
@ - Wait 1-2 msec, minimum of 1 msec, not more than 2 msec
0-F - Tx Hex character, must be in pairs

An example script to write to an 27C01 EEPROM memory location and then read back the memory location value is as follows.

Tx Script: SA00000P@SA000RA1?NP
Rx Script: SA00000P@SA000RA1?00NP

This translates to:

1. Issue a START bit
2. Address the device (A0)

USB to I2C/SPI Interface Operating Manual

3. Select EEPROM memory location 0 (0x00)
4. Write a zero to the memory location (0x00)
5. Issue a STOP bit
6. Wait 1 msec
7. Issue a START bit
8. Readdress the device and select memory location 0.
9. Issue a RESTART bit
10. Readdress the device to read
11. Read a byte from location zero.
12. Issue a NAK indicating no more reading
13. Issue a STOP bit

The I2C scripting functionality allows users to build up a series of I2C communications elements, send them in mass to the TIMS I2C Device processor and then have the clocked out consistently at the configured clock rate.

This scripting mechanism allows the user to implement some other protocols not directly supported by the standard driver. C Bus and I2C 10-bit addressing are examples of functionality that can be built using the scripting function.

The Rx Script return string contains the original script with extra characters inserted the contain errors or response data bytes from the slave. If an error occurs a "!" exclamation point is placed in the string where the error occurred.

Another example that reads 16 bytes beginning at location 0x10:

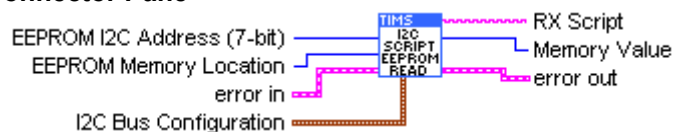
Tx Script: SA000RA1?K?K?K?K?K?K?K?K?K?K?K?K?K?K?K?NP

Rx Script: SA000RA1?00K?01K?02K?03K?04K?05K?06K?07K?08K?09K?0A?0B?0C?0D?0E?0FNP

=====

© Jova Solutions, Inc

Connector Pane



Controls and Indicators

**EEPROM I2C Address (7-bit)****EEPROM Memory Location****I2C Bus Configuration****Slew Rate****Target I2C Clock Rate (Khz)****Timeout (ms)**

error in The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

USB to I2C/SPI Interface Operating Manual

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



RX Script



Memory Value

Example_I2C_script_write_eeprom_memory.vi

This example VI shows the basics of I2C communications using the TIMS I2C Device and the LabVIEW I2C scripting drivers.

The TIMS_Open_Dialog.vi and TIMS_Port_Close.vi are mandatory and Ports should not be left open indefinitely as some VI's can not execute on open ports.

The TIMS_Open_Dialog.vi creates the reference (TIMS0102_Handle) used by all other VI's in the driver library. Every application should start with this VI and ensure that the TIMS_Port_Close.vi is also called once for every port open command.

If the port has been opened and not closed another attempt to open the port will result in an error 1, Invalid Handle.

Once the TIMS port is open, communications with the TIMS module can take place and normally the I2C bus configuration is set, as in the example, to specify the desired clock and slew rates. These settings are stored in the TIMS I2C Device EEPROM and are used for all I2C communications until the configuration is set again.

The +5V power pins on both the I2C and SPI connectors can be switched on or off using the TIMS0102_Ext_5V_Control.vi

This example uses the TIMS_I2C_Script_IO.vi to send a script to the TIMS I2C Device module and execute that script to write a value to a memory location in an EEPROM I2C device.

An I2C script is a string made up of a series of characters, each character representing an I2C communications protocol basic component. Script character codes are case sensitive.

S - Start

P - Stop

R - Restart

K - ACK

N - NAK

? - Read Byte

@ - Wait 1-2 msec, minimum of 1 msec, not more than 2 msec

0-F - Tx Hex character, must be in pairs

An example script to write to an 27C01 EEPROM memory location and then read back the memory location value is as follows.

Tx Script: SA00000P

Software Specification

USB to I2C/SPI Interface Operating Manual

Rx Script: SA00000

This translates to:

1. Issue a START bit
2. Address the device (A0)
3. Select EEPROM memory location 0 (0x00)
4. Write a zero to the memory location (0x00)
5. Issue a STOP bit

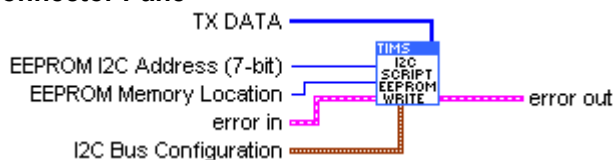
The I2C scripting functionality allows users to build up a series of I2C communications elements, send them in mass to the TIMS I2C Device processor and then have the clocked out consistently at the configured clock rate.

This scripting mechanism allows the user to implement some other protocols not directly supported by the standard driver. C Bus and I2C 10-bit addressing are examples of functionality that can be built using the scripting function.

The Rx Script return string contains the original script with extra characters inserted the contain errors or response data bytes from the slave. If an error occurs a "!" exclamation point is placed in the string where the error occurred.

=====
© Jova Solutions, Inc

Connector Pane



Controls and Indicators

- TX DATA**
- FROM**
- EEPROM I2C Address (7-bit)**
- EEPROM Memory Location**
- I2C Bus Configuration**
- Slew Rate**
- Target I2C Clock Rate (Khz)**
- Timeout (ms)**

error in The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

USB to I2C/SPI Interface Operating Manual

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

Example_I2C_write_eeprom_memory.vi

This example VI shows the basics of I2C communications using the TIMS I2C Device and the LabVIEW drivers.

The TIMS_Open_Dialog.vi and TIMS_Port_Close.vi are mandatory and Ports should not be left open indefinitely as some VI's can not execute on open ports.

USB to I2C/SPI Interface Operating Manual

The TIMS_Open_Dialog.vi creates a reference used by all other VI's in the driver library. Every application should start with this VI and ensure that the TIMS_Port_Close.vi is also called once for every port open command.

If the port has been opened and not closed another attempt to open the port will result in an error 1, Invalid Handle.

Once the TIMS port is open, communications with the TIMS module can take place and normally the I2C bus configuration is set, as in the example, to specify the desired clock and slew rates. These settings are stored in the TIMS I2C Device EEPROM and are used for all I2C communications until the configuration is set again.

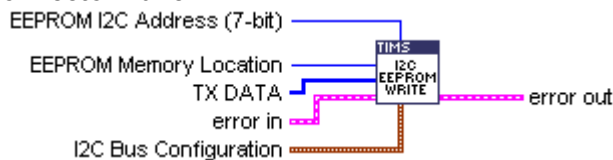
The +5V power pins on both the I2C and SPI connectors can be switched on or off using the TIMS0102_Ext_5V_Control.vi

This example uses the TIMS_I2C_Device_IO.vi easy IO vi to send a command to and write a value to a memory location in a EEPROM I2C device.

=====

© Jova Solutions, Inc

Connector Pane



Controls and Indicators

U8 TX DATA

U8 FROM

U8 EEPROM I2C Address (7-bit)

U8 EEPROM Memory Location

U05 I2C Bus Configuration

U16 Slew Rate

U5L Target I2C Clock Rate (Khz)

U16 Timeout (ms)

Err **error in** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

TF **status** The **status** boolean is either TRUE (X) for an error, or FALSE

USB to I2C/SPI Interface Operating Manual

(checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

read-EEPROM-25LC040_example.vi

=====

© Jova Solutions, Inc

Connector Pane



Controls and Indicators

TIMS0102_Control_Panel.vi

=====

© Jova Solutions, Inc

Connector Pane



Controls and Indicators

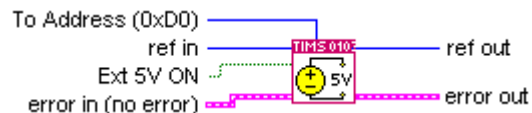
TIMS0102_Ext_5V_Control.vi

Controls power to the +5V pins of the I2C and SPI connectors on the TIMS-0102 module.

=====

© Jova Solutions, Inc

Connector Pane



Controls and Indicators



Ext 5V ON Selects the state of the 5V output line.



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

USB to I2C/SPI Interface Operating Manual

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



ref in TIMS Device Handle



To Address (0xD0) To Address specifies the TIMS processor address. The default value of 0xD0 refers to the main processor on any TIMS unit.



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



ref out TIMS Device Handle

TIMS_0102_Open_Device.vi

Opens a USB connection to a TIMS-0102 device. Returns a managed TIMS Reference for use in accessing this device.

The TIMS Reference can be opened by Device number, TIMS Serial Number, or USB Description.

1. By device number: If you have only one TIMS unit connected to your system, this is the easiest method to use. Supply '0' to the Device number. If you have more than one device, you can access the devices by supplying their device number. NOTE: The windows OS is responsible for enumerating the devices, and therefore the assigned device numbers may change at any time. We do not recommend using the device number when you have more than one TIMS device attached to your system.

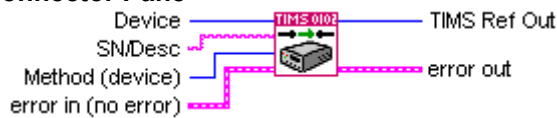
USB to I2C/SPI Interface Operating Manual

2. By TIMS Serial Number: You can open by TIMS Serial Number as well. This is the Serial Number printed on the label on the outside of your TIMS device.
3. By USB Description: If you have multiple models of TIMS units, you can access the unit by specifying the USB description. However, this method is not recommended if you have multiple units of the same model attached to the system. In this case the USB descriptions will be the same for similar models, and it is impossible to predict which specific device you will gain access to.

=====

© Jova Solutions, Inc

Connector Pane



Controls and Indicators



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



Device



SN/Desc



Method (device)



error out The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

USB to I2C/SPI Interface Operating Manual

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



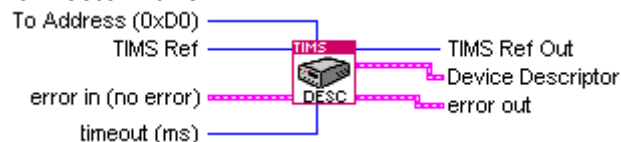
TIMS Ref Out The TIMS Reference used to communicate with a device.

TIMS_Device_Descriptor_Get.vi

Gets the Device Descriptor for the TIMS device. The device descriptor includes the model number, serial number, function, and user ID.

=====

© Jova Solutions, Inc

Connector Pane**Controls and Indicators**

error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

USB to I2C/SPI Interface Operating Manual

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



timeout (ms) timeout specifies the amount of time (in milliseconds) that the LabVIEW driver will wait after each byte is received from a TIMS unit. If the next byte is not received within the specified timeout, the IO function will return an error and any byte received up to that point.



TIMS Ref The TIMS Reference used to communicate with a device.



To Address (0xD0) To Address specifies the TIMS processor address. The default value of 0xD0 refers to the main processor on any TIMS unit.



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



Device Descriptor **Device Descriptor** contains general information about the TIMS device.



Model No.



Serial No.



Function



User ID

U32 **TIMS Ref Out** The TIMS Reference used to communicate with a device.

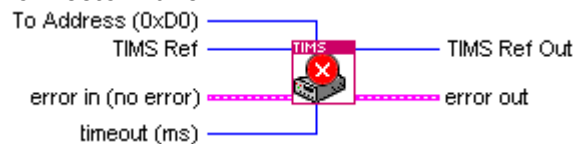
TIMS_Device_Reset.vi

Directs a specified TIMS device to perform a hardware reset.

=====

© Jova Solutions, Inc

Connector Pane



Controls and Indicators

EPH **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

TF **status** The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

I32 **code** The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

abc **source** The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

U32 **TIMS Ref** The TIMS Reference used to communicate with a device.

U8 **To Address (0xD0)** To Address specifies the TIMS processor address. The default value of 0xD0 refers to the main processor on any TIMS unit.

U32 **timeout (ms)** timeout specifies the amount of time (in milliseconds) that the LabVIEW driver will wait after each byte is received from a TIMS unit. If the next byte is not received within the specified timeout, the IO function will return an error and any byte received up to that point.

USB to I2C/SPI Interface Operating Manual



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



TIMS Ref Out The TIMS Reference used to communicate with a device.

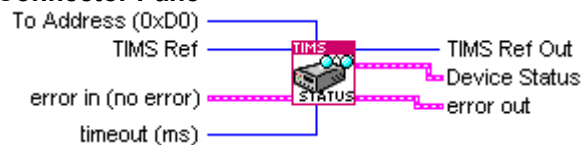
TIMS_Device_Status.vi

Returns the Status Information of a TIMS device.

General Status Byte-0 - u8
Error Status Byte-1 - u8
Reset Status Byte-2 - u8
PC Stack Status Byte-3 - u8
Expected SW Code Checksum - u32
Calculated SW Code Checksum - u32

=====
© Jova Solutions, Inc

Connector Pane



Controls and Indicators



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the Software Specification

error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



timeout (ms) timeout specifies the amount of time (in milliseconds) that the LabVIEW driver will wait after each byte is received from a TIMS unit. If the next byte is not received within the specified timeout, the IO function will return an error and any byte received up to that point.



TIMS Ref The TIMS Reference used to communicate with a device.



To Address (0xD0) To Address specifies the TIMS processor address. The default value of 0xD0 refers to the main processor on any TIMS unit.



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

 **Device Status** Contains the **Status Information**.

 General

-  **SW Valid**
-  **Unknown Code**
-  **TX Overflow**
-  **RX Timeout**
-  **(na4)**
-  **(na5)**
-  **FLASH Prgm**
-  **Test Mode**

 Error

-  **EE CRC**
-  **EE Write**
-  **EE Address**
-  **(na3)**
-  **FLASH Erase**
-  **FLASH Write**
-  **UART Framing**
-  **UART Overrun**

 Reset

-  **BOR**
-  **POR**
-  **PD**
-  **TO**
-  **RI**

-  (na5)
-  (na6)
-  RESET
-  **PC Stack**
-  SP0
-  SP1
-  SP2
-  SP3
-  SP4
-  (na5)
-  STKUNF
-  STKFUL
-  **Monitor FW Code Checksum**
-  **Expected SW Code Checksum**
-  **Calculated SW Code Checksum**
-  **TIMS Ref Out** The TIMS Reference used to communicate with a device.

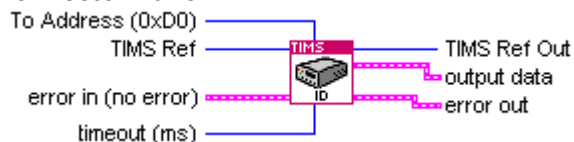
TIMS_DEVICE_Unit_ID_Get.vi

Gets the complete set of Device Unit Identification for the TIMS device.

=====

© Jova Solutions, Inc

Connector Pane



Controls and Indicators

-  **To Address (0xD0)** To Address specifies the TIMS processor address. The default value of 0xD0 refers to the main processor on any TIMS unit.

USB to I2C/SPI Interface Operating Manual



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



timeout (ms) timeout specifies the amount of time (in milliseconds) that the LabVIEW driver will wait after each byte is received from a TIMS unit. If the next byte is not received within the specified timeout, the IO function will return an error and any byte received up to that point.



TIMS Ref The TIMS Reference used to communicate with a device.



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information

about the error displayed.



TIMS Ref Out The TIMS Reference used to communicate with a device.



output data This cluster contains information about the TIMS device software, hardware and firmware.



User ID User supplied Name of the TIMS device.



Model ID Factory information about the TIMS device model.



Model No.



Description 1



Description 2



PCB Assy ID Factory information about the TIMS device PCB assembly



Part No.



Serial No.



Misc Info



Device ID



Device Function



Device Description



Application SW ID



Monitor SW ID



TOP Assy ID



Part No.



Serial No.



Misc Info

TIMS_DEVICE_User_ID_Get.vi

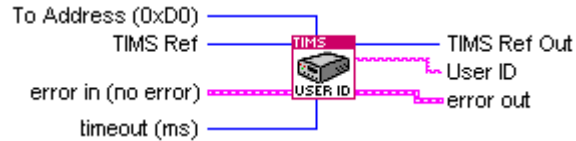
Gets the user specified Name for the TIMS device.

=====

© Jova Solutions, Inc

USB to I2C/SPI Interface Operating Manual

Connector Pane



Controls and Indicators

error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



timeout (ms) timeout specifies the amount of time (in milliseconds) that the LabVIEW driver will wait after each byte is received from a TIMS unit. If the next byte is not received within the specified timeout, the IO function will return an error and any byte received up to that point.



TIMS Ref The TIMS Reference used to communicate with a device.



To Address (0xD0) To Address specifies the TIMS processor address. The default value of 0xD0 refers to the main processor on any TIMS unit.



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information

USB to I2C/SPI Interface Operating Manual

about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



User ID The user specified name for the TIMS device.



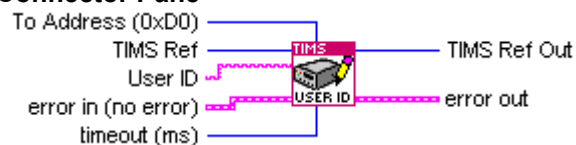
TIMS Ref Out The TIMS Reference used to communicate with a device.

TIMS_Device_User_ID_Set.vi

Sets the user specified **Name** for the TIMS device.

=====
© Jova Solutions, Inc

Connector Pane



Controls and Indicators



To Address (0xD0) To Address specifies the TIMS processor address. The default value of 0xD0 refers to the main processor on any TIMS unit.



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

USB to I2C/SPI Interface Operating Manual

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



timeout (ms) timeout specifies the amount of time (in milliseconds) that the LabVIEW driver will wait after each byte is received from a TIMS unit. If the next byte is not received within the specified timeout, the IO function will return an error and any byte received up to that point.



User ID The user-specified name for the TIMS device.



TIMS Ref The TIMS Reference used to communicate with a device.



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



TIMS Ref Out The TIMS Reference used to communicate with a device.

TIMS_Get_Ref_Info.vi

Returns the connection information associated with a TIMS device reference.

=====

© Jova Solutions, Inc

Connector Pane

Software Specification



Controls and Indicators

error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

TIMS Ref In The TIMS Reference used to communicate with a device.

error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

USB to I2C/SPI Interface Operating Manual

	device connection info
	Status
	Type
	Index
	Model No.
	Description
	User ID User supplied Name of the TIMS device.
	Serial No.
	reference
	USB Serial Number
	USB Product Description

TIMS_I2C_BUS_Config_Get.vi

Returns the I2C Bus Configuration from the TIMS I2C Device EEPROM.

The TIMS I2C Device stores I2C bus configuration information in EEPROM.

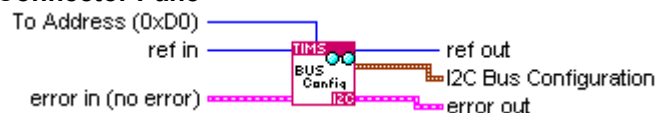
The I2C Bus Configuration includes:

Target Clock Rate: The I2C clock rate is programmable to 100KHz, 400KHz, 1MHz, and 252 other frequencies between 39.0625 KHz and 10MHz.

Slew Rate: The Slew Rate can be set to Disables for standard speed (100K or 1M) or Enabled for high speed (400K).

=====
© Jova Solutions, Inc

Connector Pane



Controls and Indicators

-  **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

USB to I2C/SPI Interface Operating Manual

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



ref in TIMS Device Handle



To Address (0xD0) To Address specifies the TIMS processor address. The default value of 0xD0 refers to the main processor on any TIMS unit.



I2C Bus Configuration



Slew Rate



Target I2C Clock Rate (Khz)



Timeout (ms)



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

USB to I2C/SPI Interface Operating Manual



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



ref out TIMS Device Handle

TIMS_I2C_BUS_Config_Set.vi

Writes the I2C Bus Configuration parameters to the TIMS I2C Device EEPROM.

The I2C bus communications configuration (clock and slew rates) is stored in EEPROM on the TIMS module. This bus-timing configuration is used for all I2C communications and is retained through power cycling.

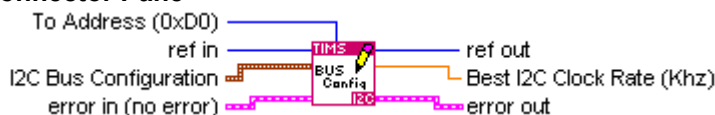
The I2C Bus Configuration includes:

Target Clock Rate: The I2C clock rate should be set to a value compatible with all devices on your I2C bus. The I2C clock rate is programmable to 100KHz, 400KHz, 1MHz, and 252 other frequencies between 39.0625 KHz and 10MHz.

Slew Rate: The Slew Rate can be set to Disabled for standard speed (100K or 1M) or Enabled for high speed (400K).

=====

© Jova Solutions, Inc

Connector Pane**Controls and Indicators**

error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information

about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



I2C Bus Configuration



Slew Rate



Target I2C Clock Rate (Khz)



Timeout (ms)



ref in TIMS Device Handle



To Address (0xD0) To Address specifies the TIMS processor address.
The default value of 0xD0 refers to the main processor on any TIMS unit.



Best I2C Clock Rate (Khz)



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



ref out TIMS Device Handle

TIMS_I2C_Device_IO.vi

Performs a simple I2C I/O transaction with RESTART to a 7-bit addressable device.

USB to I2C/SPI Interface Operating Manual

This VI can perform both a write and read operation. If the input TX DATA is empty the write operation is not performed. If the Bytes to Read input is set to 0 the read operation is not performed.

If the TX Data array input has data then this VI starts with a write operation. An I2C START bit is followed by an address byte (the R/W bit is automatically inserted). The TX DATA bytes are then transmitted followed with an I2C RESTART command if the Bytes to Read input is set to a value other than zero. A Read operation is then performed starting with an I2C START bit followed by an address byte (with the R/W bit set to R) and then the requested number of bytes are received from the slave. The operation is completed with an I2C STOP command.

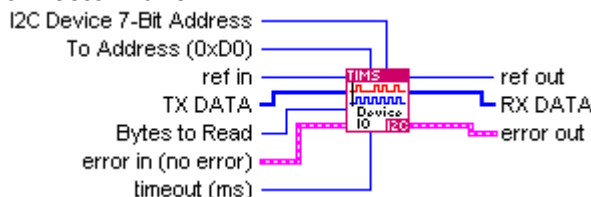
There is no delay between the write operation and read operation. Some devices require processing time before responses can be generated (for example if you are writing to a memory location in the EEPROM you can not immediately read back the memory location without a small delay allowing the write cycle to complete). Use the I2C scripting function if you need more control over delay timing when executing commands.

The maximum number of bytes you can write is 254. The maximum number of bytes you can read is 255.

=====

© Jova Solutions, Inc

Connector Pane



Controls and Indicators

U8 TX DATA Array of bytes to send to the slave I2C device.

U8 FROM

U32 **timeout (ms)** timeout specifies the amount of time (in milliseconds) that the LabVIEW driver will wait after each byte is received from a TIMS unit. If the next byte is not received within the specified timeout, the IO function will return an error and any byte received up to that point.

U8 **Bytes to Read** Number fo bytes to read from the slave device.

U8 **I2C Device 7-Bit Address** Address of the I2C slave device.

Err **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

USB to I2C/SPI Interface Operating Manual



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



ref in TIMS Device Handle



To Address (0xD0) To Address specifies the TIMS processor address. The default value of 0xD0 refers to the main processor on any TIMS unit.



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



RX DATA Array of U8 bytes read from the the I2C device.



FROM



ref out TIMS Device Handle

USB to I2C/SPI Interface Operating Manual

TIMS_I2C_Error_Get.vi

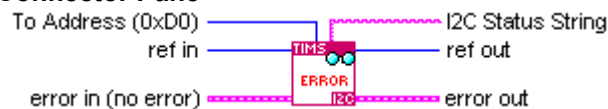
Returns a description of the last I2C error as a string.

This VI should be called after an I2C call has returned Error 5341 (0x14DD) in order to receive a better description of the exact I2C error.

=====

© Jova Solutions, Inc

Connector Pane



Controls and Indicators

error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

ref in TIMS Device Handle

To Address (0xD0) To Address specifies the TIMS processor address. The default value of 0xD0 refers to the main processor on any TIMS unit.

error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

status The **status** boolean is either TRUE (X) for an error, or FALSE

USB to I2C/SPI Interface Operating Manual

(checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

 **code** The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

 **source** The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

 **ref out** TIMS Device Handle

 **I2C Status String**

TIMS_I2C_Script_7bit_Address.vi

Adds an I2C ADDRESS BYTE script element to the script string.

=====

© Jova Solutions, Inc

Connector Pane**Controls and Indicators**

 **Script input**

 **Address (7-bit)**

 **Read/Write bit**

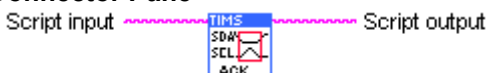
 **Script output**

TIMS_I2C_Script_ACK.vi

Adds an I2C ACK script element to the script string.

=====

© Jova Solutions, Inc

Connector Pane**Controls and Indicators**

Software Specification

**Script input****Script output****TIMS_I2C_Script_IO.vi**

Sends an I2C script to the TIMS I2C Device module then executes the script and returns any bytes read during the execution of the script.

An I2C script is a string made up of a series of characters, each character representing an I2C communications protocol basic component. Script character codes are case sensitive.

S - Start
P - Stop
R - Restart
K - ACK
N - NAK
? - Read Byte
@ - Wait 1-2 msec, minimum of 1 msec, not more than 2 msec
0-F - Tx Hex character, must be in pairs

An example script to write to an 27C01 EEPROM memory location and then read back the memory location value is as follows.

Tx Script: SA00000P@SA000RA1?NP
Rx Script: SA00000P@SA000RA1?00NP

This translates to:

1. Issue a START bit
2. Address the device (A0)
3. Select EEPROM memory location 0 (0x00)
4. Write a zero to the memory location (0x00)
5. Issue a STOP bit
6. Wait 1 msec
7. Issue a START bit
8. Readdress the device and select memory location 0.
9. Issue a RESTART bit
10. Readdress the device to read
11. Read a byte from location zero.
12. Issue a NAK indicating no more reading
13. Issue a STOP bit

The I2C scripting functionality allows users to build up a series of I2C communications elements, send them in mass to the TIMS I2C Device processor and then have the clocked out consistently at the configured clock rate.

This scripting mechanism allows the user to implement some other protocols not directly supported by the standard driver. C Bus and I2C 10-bit addressing are examples of functionality that can be built using the scripting function.

USB to I2C/SPI Interface Operating Manual

The Rx Script return string contains the original script with extra characters inserted the contain errors or response data bytes from the slave. If an error occurs a "!" exclamation point is placed in the string where the error occurred.

Another example that reads 16 bytes beginning at location 0x10.

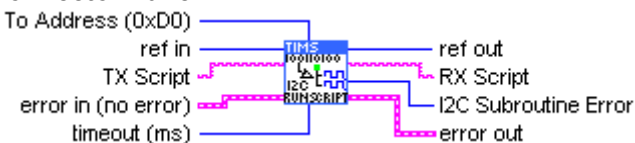
Tx Script: SA000RA1?K?K?K?K?K?K?K?K?K?K?K?K?K?K?K?NP

Rx Script: SA000RA1?00K?01K?02K?03K?04K?05K?06K?07K?08K?09K?0A?0B?0C?0D?0E?0FNP

=====

© Jova Solutions, Inc

Connector Pane



Controls and Indicators

U32 **timeout (ms)** timeout specifies the amount of time (in milliseconds) that the LabVIEW driver will wait after each byte is received from a TIMS unit. If the next byte is not received within the specified timeout, the IO function will return an error and any byte received up to that point.

Err **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

TF **status** The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

I32 **code** The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

abc **source** The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

abc **TX Script**

USB to I2C/SPI Interface Operating Manual

 **ref in** TIMS Device Handle **To Address (0xD0)** To Address specifies the TIMS processor address.
The default value of 0xD0 refers to the main processor on any TIMS unit. **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

 **status** The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

 **code** The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

 **source** The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

 **RX Script** **I2C Subroutine Error** **ref out** TIMS Device Handle**TIMS_I2C_Script_NAK.vi**

Adds an I2C NAK script element to the script string.

=====

© Jova Solutions, Inc

Connector Pane

Script input  Script output

Controls and Indicators **Script input** **Script output****TIMS_I2C_Script_Read_Byte.vi**

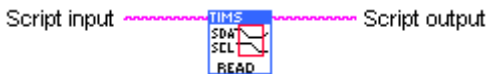
Software Specification

Adds a READ BYTE script element to the script string.

=====

© Jova Solutions, Inc

Connector Pane



Controls and Indicators

 Script input

 Script output

TIMS_I2C_Script_Restart.vi

Adds an I2C RESTART command script element to the script string.

=====

© Jova Solutions, Inc

Connector Pane



Controls and Indicators

 Script input

 Script output

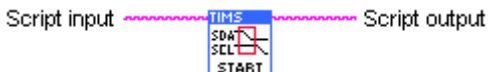
TIMS_I2C_Script_Start.vi

Adds an I2C START script element to the script string.

=====

© Jova Solutions, Inc

Connector Pane



Controls and Indicators

 Script input

 Script output

TIMS_I2C_Script_Stop.vi

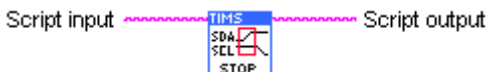
Adds an I2C STOP script element to the script string..

=====

© Jova Solutions, Inc

USB to I2C/SPI Interface Operating Manual

Connector Pane



Controls and Indicators

 **Script input**

 **Script output**

TIMS_I2C_Script_Tx_Byte.vi

Adds DATA BYTE script elements to the script string.

=====

© Jova Solutions, Inc

Connector Pane



Controls and Indicators

 **Script input**

 **Data Bytes**

 **data byte**

 **Script output**

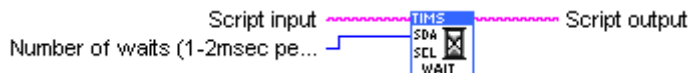
TIMS_I2C_Script_Wait.vi

Adds a WAIT script element to the script string.

=====

© Jova Solutions, Inc

Connector Pane



Controls and Indicators

 **Script input**

 **Number of waits (1-2msec per wait)**

 **Script output**

TIMS_List_Devices.vi

Lists all TIMS devices. opened and unopened, connected to this computer.

=====
© Jova Solutions, Inc

Connector Pane



Controls and Indicators

 **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

 **status** The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

 **code** The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

 **source** The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

 **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

 **status** The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

 **code** The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

 **source** The **source** string describes the origin of the error or warning.

USB to I2C/SPI Interface Operating Manual

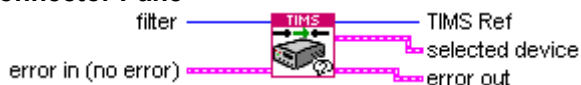
The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

**FOUND TIMS USB DEVICES****Cluster****Status****Type****Index****Model No.****Description****User ID** User supplied Name of the TIMS device.**Serial No.****reference****USB Serial Number****USB Product Description****TIMS_Open_Dialog.vi**

Presents a user dialog for selecting a TIMS device. Opens a USB connection to a TIMS device. Returns a managed TIMS Reference to be used to access this device.

=====

© Jova Solutions, Inc

Connector Pane**Controls and Indicators****filter**

error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

USB to I2C/SPI Interface Operating Manual



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



TIMS Ref The TIMS Reference used to communicate with a device.



error out The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



selected device



Status



Type



Index



Model No.

USB to I2C/SPI Interface Operating Manual

	Description
	User ID User supplied Name of the TIMS device.
	Serial No.
	reference
	USB Serial Number
	USB Product Description

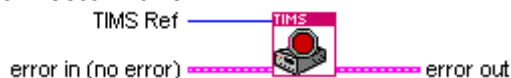
TIMS_Port_Close.vi

Closes an opened TIMS reference.

Note: This VI will operate even if there is an incoming error.

=====

© Jova Solutions, Inc

Connector Pane**Controls and Indicators**

error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

USB to I2C/SPI Interface Operating Manual

U32 **TIMS Ref** The TIMS Reference used to communicate with a device.

error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

TF **status** The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

I32 **code** The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

abc **source** The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

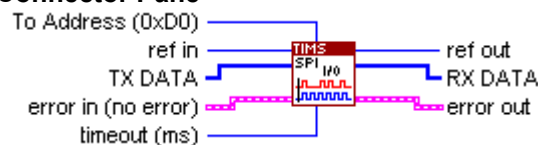
TIMS_SPI_Data_IO.vi

This vi performs a basic SPI communication transaction. This VI does not affect the device select lines; they must be separately enabled.

This VI is intended to be used when more than 255 bytes need to be read or sent out between chip select enabled and chip select disabled, as in the following VI. In this situation, call TIMS_SPI_Data_IO.vi in a loop between calls to TIMS_SPI_Device_Enable.vi.

=====

© Jova Solutions, Inc

Connector Pane**Controls and Indicators**

U8 **TX DATA**

U8 **FROM**

U32 **timeout (ms)** timeout specifies the amount of time (in milliseconds) that the LabVIEW driver will wait after each byte is received from a TIMS unit. If the next byte is not received within the specified timeout, the IO function will return an

error and any byte received up to that point.



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



ref in TIMS Device Handle



To Address (0xD0) To Address specifies the TIMS processor address. The default value of 0xD0 refers to the main processor on any TIMS unit.



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information

about the error displayed.

U8 RX DATA

U8 FROM

U32 ref out TIMS Device Handle

TIMS_SPI_Device_Config_Get.vi

Returns the SPI Device Configuration from the TIMS SPI Device EEPROM.

The TIMS SPI Device can store up to 16 different SPI device configurations in EEPROM. This allows you to pre-configure communications with multiple SPI devices, each using different clock rates and clock edge configurations.

The SPI Device Configuration includes:

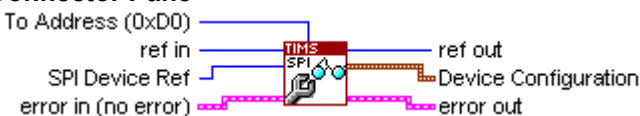
Tx Clocking: The clock idle state can be configured to be high or low and transmission can be configured to be on the falling or rising edge.

Rx Clocking: Rx Clocking can be configured so that when clocking in data from the slave, data is sampled at either the Middle or End of the TX output period.

Target Clock Rate: The TIMS SPI Device supports 255 clock speeds from 39.0625 KHz to 10 MHz, including standard clock rates of 100 KHz, 400 KHz, and 1 MHz.

=====
© Jova Solutions, Inc

Connector Pane



Controls and Indicators

U8 SPI Device Ref Specifies which of the 16 available device configurations to retrieve.

Err error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

TF status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information

USB to I2C/SPI Interface Operating Manual

about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



ref in TIMS Device Handle



To Address (0xD0) To Address specifies the TIMS processor address. The default value of 0xD0 refers to the main processor on any TIMS unit.



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



Device Configuration



TX Clocking



RX Clocking



Target SPI Clock Frequency (KHz)



ref out TIMS Device Handle

USB to I2C/SPI Interface Operating Manual

TIMS_SPI_Device_Config_Set.vi

Writes the SPI Device Configuration to the TIMS SPI Device EEPROM.

The TIMS SPI Device can store up to 16 different SPI device configurations in EEPROM. This allows you to pre-configure communications with multiple SPI devices, each using different clock rates and clock edge configurations. When issuing SPI I/O commands you can reference the preconfigured settings and quickly switch between communications with devices on the same bus but using different clock rates and edge processing configurations. This saves quite a bit of time required to constantly process the reconfigure commands.

The SPI Device Configuration includes:

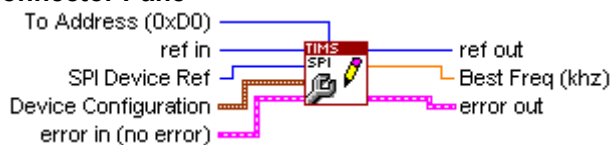
Tx Clocking: The clock idle state can be configured to be high or low and transmission can be configured to be on the falling or rising edge.

Rx Clocking: Rx Clocking can be configured so that when clocking in data from the slave, data is sampled at either the Middle or End of the TX output period.

Target Clock Rate: The TIMS SPI Device supports 255 clock speeds from 39.0625 KHz to 10 MHz, including standard clock rates of 100 KHz, 400 KHz, and 1 MHz.

=====

© Jova Solutions, Inc

Connector Pane**Controls and Indicators** **Device Configuration** **TX Clocking** **RX Clocking** **Target SPI Clock Frequency (KHz)** **SPI Device Ref** Specifies which of the 16 available device configurations to set. **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

 **status** The **status** boolean is either TRUE (X) for an error, or FALSE

USB to I2C/SPI Interface Operating Manual

(checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

 **code** The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

 **source** The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

 **ref in** TIMS Device Handle

 **To Address (0xD0)** To Address specifies the TIMS processor address. The default value of 0xD0 refers to the main processor on any TIMS unit.

 **error out** The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

 **status** The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

 **code** The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

 **source** The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

 **Best Freq (khz)**

 **ref out** TIMS Device Handle

TIMS_SPI_Device_Enable.vi

USB to I2C/SPI Interface Operating Manual

The TIMS SPI Device comes with eight programmable device or chip select lines that are available at the output SPI DB-15F connector. These chip select lines can be independently selected or used to generate a combination of a 4-bit address and single chip select line.

The following options are available:

ADR x0, nCS4
ADR x1, nCS4
ADR x2, nCS4
ADR x3, nCS4
ADR x4, nCS4
ADR x5, nCS4
ADR x6, nCS4
ADR x7, nCS4
ADR x8, nCS4
ADR x9, nCS4
ADR xA, nCS4
ADR xB, nCS4
ADR xC, nCS4
ADR xD, nCS4
ADR xE, nCS4
ADR xF, nCS4

nCS0

nCS1

nCS2

nCS3

nCS4

nCS5

nCS6

nCS7

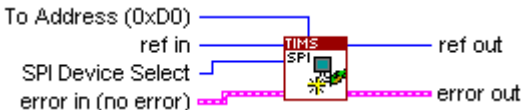
No Select

When ADRx0, nCS4 through ADRxF, nCS4 is selected the corresponding SPI Device configuration is recalled and used in the following transaction. The address lines are updated with the requested address and then the nCS4 (i.e., nEN Not Enable) is set low.

If nCS0 through nCS7 is selected then SPI Device Configurations 0 through 7 is recalled and used in the following transaction. The requested line, nCS0 - nCS7, is set low.

Selecting the No Select option sets all line high.

=====

Connector Pane**Controls and Indicators**

U8 **SPI Device Select** Specifies which SPI device to address.

When communicating with an SPI device, the TIMS-0102 will use the config settings

corresponding to the address you provide here.



error in (no error) The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



ref in TIMS Device Handle



To Address (0xD0) To Address specifies the TIMS processor address. The default value of 0xD0 refers to the main processor on any TIMS unit.



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information

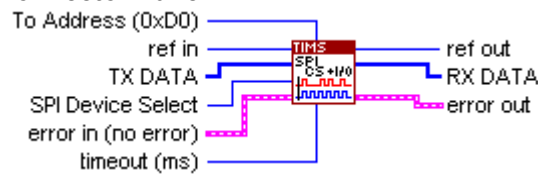
about the error displayed.

 **ref out** TIMS Device Handle

TIMS_SPI_Device_IO.vi

This vi performs the chip select operation, a data I/O operations up to 255 bytes, then a chip deselect operation.

Connector Pane



Controls and Indicators

 **TX DATA**

 **FROM**

 **timeout (ms)** timeout specifies the amount of time (in milliseconds) that the LabVIEW driver will wait after each byte is received from a TIMS unit. If the next byte is not received within the specified timeout, the IO function will return an error and any byte received up to that point.

 **error in (no error)** The **error in** cluster can accept error information wired from VIs previously called. Use this information to decide if any functionality should be bypassed in the event of errors from other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

 **status** The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

 **code** The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

 **source** The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.

USB to I2C/SPI Interface Operating Manual



SPI Device Select Specifies which SPI device to address.

When communicating with an SPI device, the TIMS-0102 will use the config settings corresponding to the address you provide here.



ref in TIMS Device Handle



To Address (0xD0) To Address specifies the TIMS processor address. The default value of 0xD0 refers to the main processor on any TIMS unit.



error out The **error out** cluster passes error or warning information out of a VI to be used by other VIs.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



status The **status** boolean is either TRUE (X) for an error, or FALSE (checkmark) for no error or a warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



code The **code** input identifies the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



source The **source** string describes the origin of the error or warning.

The pop-up option **Explain Error** (or Explain Warning) gives more information about the error displayed.



RX DATA



FROM



ref out TIMS Device Handle

write-EEPROM-25LC040_example.vi

Connector Pane



Controls and Indicators

none