QUICK START GUIDE

SV4E-I3C

I3C Test and Debug Module

E SERIES

Published in Canada on April 9, 2020
EN-G035E-E-20100

INTROSPECT.CA

Table of Contents

Introduction

OVERVIEW

The SV4E-I3C is an all-inclusive solution for I3C-based device interface development, test, and programming. Containing three instruments in one, this tool can act as a protocol exerciser for testing and debugging I3C slave or master devices. It can also act as a complete protocol analyzer with fine-resolution timing analysis and a full suite of conformance test capability. Finally, it contains a deep vector memory, which allows it to be used as a general purpose I3C device programmer.

All three categories of instrumentation features are accessible simultaneously and in real-time using the award winning Introspect ESP Software.

QUICK START DOCUMENTATION

This Quick Start Manual will provide the information required for a user to get up and running with the SV4E-I3C Test and Debug Module. Basic hardware and software installation instructions are included followed by a step-by-step procedure to start sending and manipulating MIPI I3C commands using the Introspect ESP Software.

Quick Start Hardware Description

REQUIREMENTS

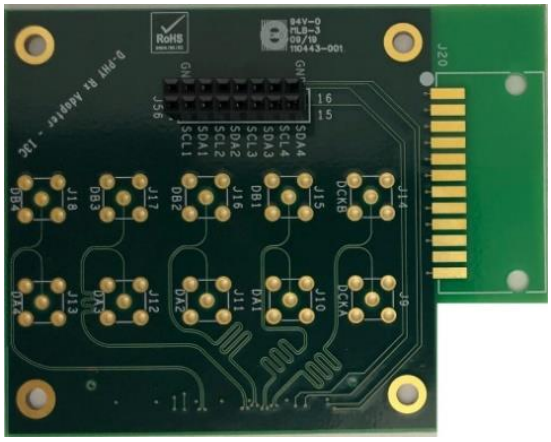
The full list of hardware required for this Quick Start Manual is provided below:

- 1 x SV4E-I3C Test and Debug Module
- 1x 12V, 5A volt DC power supply (Manufacturer: CUI, Part # ETSA120500U)
- 2 x Male to male jumper wires
- 1 x I3C Adapter for SV4E
- 1 x Personal Computer connected to the SV4E through a USB mini cable
- Optional: 1 x 1 GHz oscilloscope or higher for signal visualization

HARDWARE DESCRIPTION

The following figure shows a diagram of the physical ports of the SV4E-I3C module.

The I3C Adapter for SV4E breakout board shown below is used to connect the different internal I3C devices together or to connect internal I3C devices to external devices. For example, if you want to use two internal devices, a master and a slave, connect SCL1 to SCL2 and SDA1 to SDA2, as will be shown in the upcoming test example.



Introspect ESP Software Installation

SYSTEM REQUIREMENTS

The Introspect ESP Software provides an easy-to-use environment for device characterization and test plan development. To run the software, the following components are required:

- A PC installed with Windows XP, Vista, or Windows 7, 8 or 10
- The Introspect ESP Software install executable
- USB device drivers (refer to the driver installation instructions)

INTROSPECT ESP SOFTWARE INSTALLATION

1. PREPARE FOR INSTALLATION

Quit any Introspect ESP Software sessions before starting the installation.

2. INSTALL SOFTWARE

a) From the directory containing the installation files, double-click on the icon for "IntrospectESP_Installer.exe" and follow the instructions on the screen.
b) The installer will install a local Python environment, and it has no dependence on any Python installation that already exists on your PC.
c) When prompted, specify the location where you want to install the Introspect ESP Software application. (Note that this must be a new location, not a location of a previous installation). The default location is the "Introspect" folder under the Windows "Program Files" folder. The application will be installed into a sub-folder with a name that includes the version number. The application will also create a folder called “Introspect” under the “My Documents” folder of your account. This folder is where Test Procedures are typically stored.

3. INSTALL LICENSE FILE

a) Towards the end of the installation, you will be asked to provide either an activation key or a license file for the software.

b) If you have a valid activation key, simply select the "Use Activation Key" option and continue with the installation. You will be asked to enter the Activation Key code later when you start the GUI for the first time.

c) If you were provided with a license file instead, or if you have valid license files from a previous installation, select the "Use Existing License" option, and the installer will help you copy the license file into the new installation folder.

d) If you do not have any of the above, select the "Get a New License" option, and the installer will provide you with information that needs to be sent to Introspect Technology customer support. Before continuing, you will need to send this information to license_support@introspect.ca to request a license. Then, upon receipt of the valid license files, place them into the following directory:

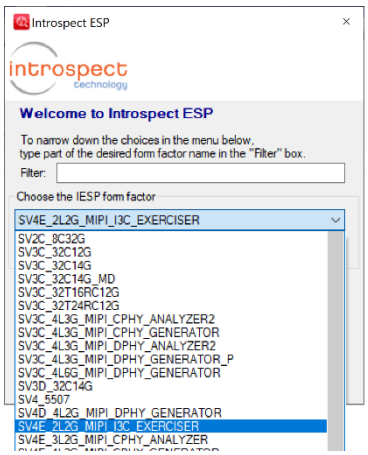
```
C:\[Your Introspect Installation Folder]\Licenses
```

NOTE

The installer creates a folder called "Introspect" under the "My Documents" folder of your Windows account. This folder is where Test Procedures are saved by default.

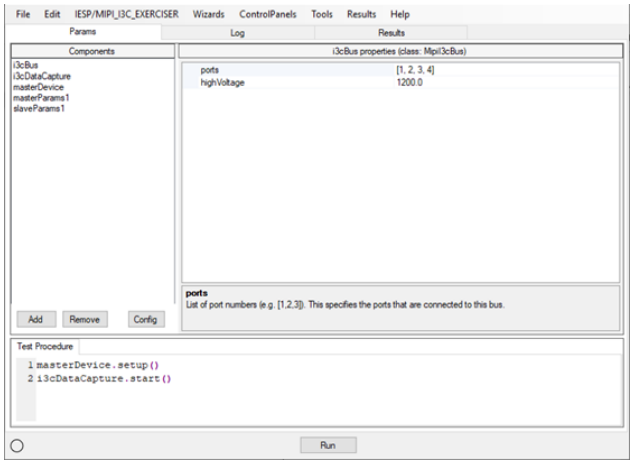
4. RUN THE INTROSPECT ESP SOFTWARE

a) Double-click on the "IntrospectESP" shortcut on your Desktop and you should see the first "welcome" window of the GUI. Specify the hardware as “SV4E_2L2G_MIPI_I3C_EXERCISER” and Press “Next” to continue.



b) Select the option “Create a new Test” and click the “Next” button.

c) With a valid license in the “Licenses” directory, the following GUI screen should come up, which indicates that the Introspect ESP Software has been successfully installed.



5. FURTHER DOCUMENTATION

The "[IntrospectESP_install_dir]\Doc" folder contains the following information on the software:

- "IntrospectESP_UserManual.pdf" is the user manual for the Introspect ESP Software and is recommended reading for all users.
- "svt.html" and "iesp.html" provide documentation on the Python component classes and lower-level functions specific to the selected form factor. Both files can be found in "<iESP_inst_dir>\Doc\FormFactors\SV4E_2L2G_MIPI_I3C_EXERCISER". These are intended for intermediate and advanced users.

NOTE

Both the user manual and the above html files are also conveniently available from the "Help" drop down menu located on the top right corner of the main Introspect ESP Software window.

"Application Notes" can also be found in the "[IntrospectESP_install_dir]\Doc" sub-folder and have more advanced features, often in the form of tutorials.

USB Driver Installation

The following procedure will allow for automated driver installation.

1. HARDWARE SETUP

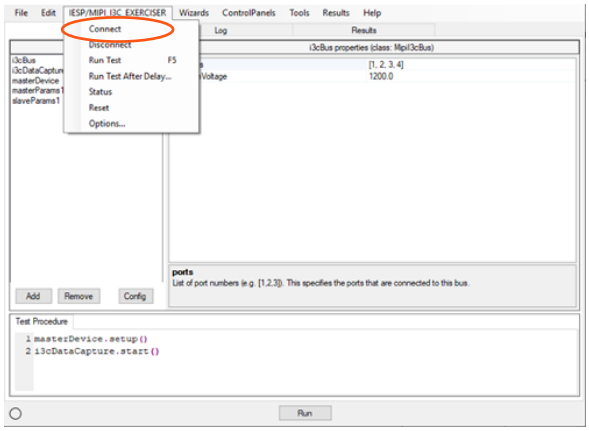
For this procedure, connect the SV4E-I3C to the PC via the USB cable as shown in the figure below, and power on the module. To allow for driver installation, the PC should be connected to the internet as well.

2. WAIT FOR THE PC TO DETECT THE NEW HARDWARE

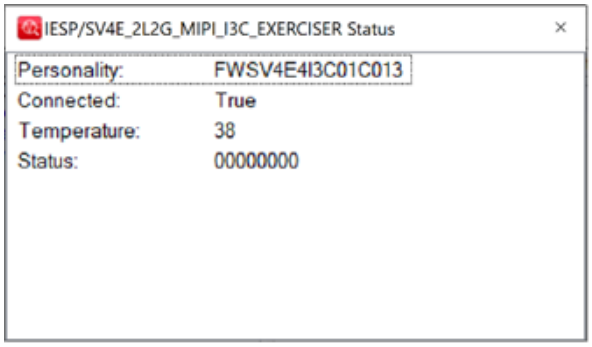
The PC should display the message “New drivers successfully installed” once the installation process is complete. If this does not occur, see the troubleshooting notes at the end of this section.

3. VERIFY DRIVER INSTALLATION

a) If it is not yet open, launch the Introspect ESP Software and select the “SV4E_2L2G_MIPI_I3C_EXERCISER” form factor. From the main GUI window, click the “IESP/MIPI_I3C_EXERCISER” drop down menu and click “Connect”, as shown here. Establishing the connection should take a couple of seconds.



b) To verify the connection, click the “IESP/MIPI_I3C_EXERCISER” drop down menu and select “Status”. A dialog window should confirm that the SV4E-I3C module is connected. Note that the firmware version may differ from what is shown here.



4. TROUBLESHOOTING

If the connection cannot be established, or if the drivers cannot be found or automatically installed, please refer to the “FTDI Driver Manual Installation” Appendix at the end of this document to install the required drivers.

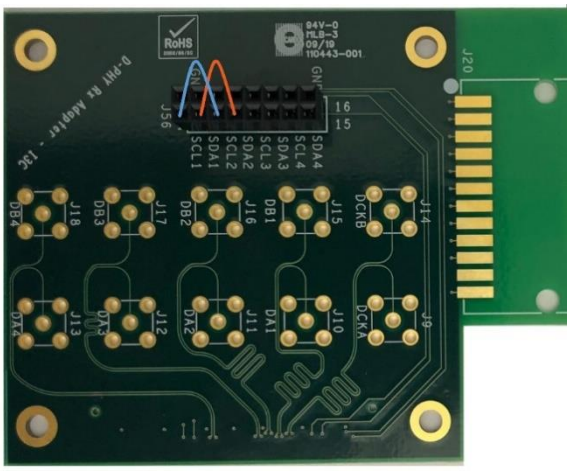
SV4E-I3C Demonstration

STEP-BY-STEP GUIDE: DYNAMIC ADDRESSES AND DIRECT READS

The following step by step guide will allow the user to assign a dynamic address to an I3C slave device and send it a direct read command from a master device inside the SV4E-I3C. It will also demonstrate how to visualize the response from the slave, using the mipiI3cDataCapture functionality of the SV4E-I3C module. The following procedure is intended to provide an overview of how to use the Introspect ESP Software GUI and highlight several of the GUI's features.

1. CONNECT THE HARDWARE COMPONENTS

For this procedure, please attach the I3C Adapter for SV4E breakout board to the SV4E-I3C module, and connect SCL1 and SCL2 together, as well as SDA1 and SDA2, as shown in the following figure.

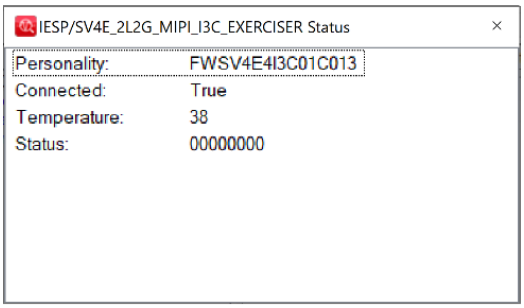


2. FAMILIARIZE YOURSELF WITH THE INTROSPECT ESP SOFTWARE GUI

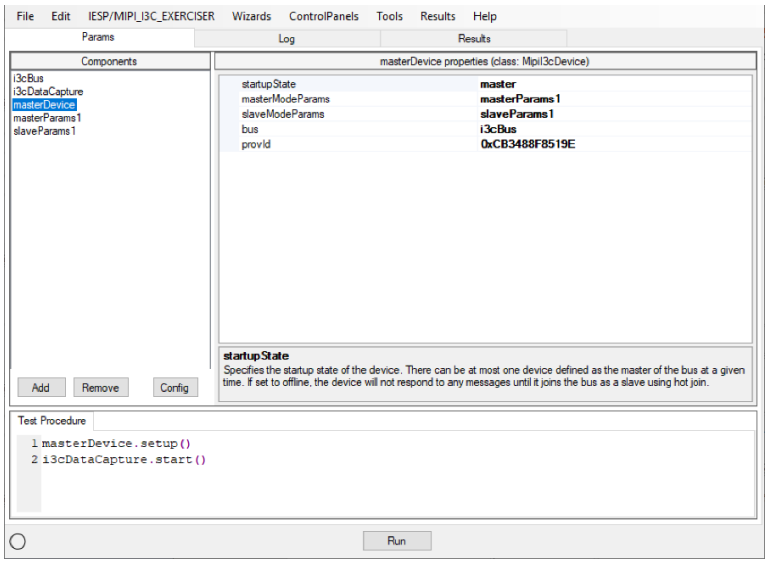
a) Double click on "IntrospectESP_GUI.exe" located in the Introspect ESP GUI installation folder and select form factor "SV4E_2L2G_MIPI_I3C_EXERCISER". At this point, connect the SV4E-I3C Test and Debug Module to your PC via the USB cable if this is not already done.

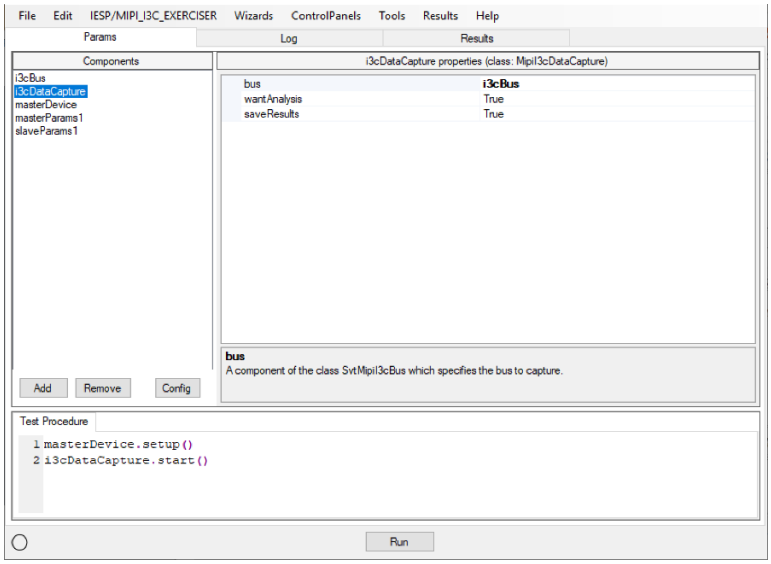
b) Select the "IESP/MIPI_I3C_EXERCISER" pull-down menu and click on the "Connect" option if you have not done so already in the previous USB driver installation procedure. Establishing connection should require a couple of seconds.

c) To verify the connection between the PC and the SV4E-I3C module, select the "IESP/MIPI_I3C_EXERCISER" pull-down menu and click on the "Status" option. A dialog window should confirm that the SV4E-I3C module is connected, as below, and will list the firmware version. Note that the firmware version may differ from what is shown below.



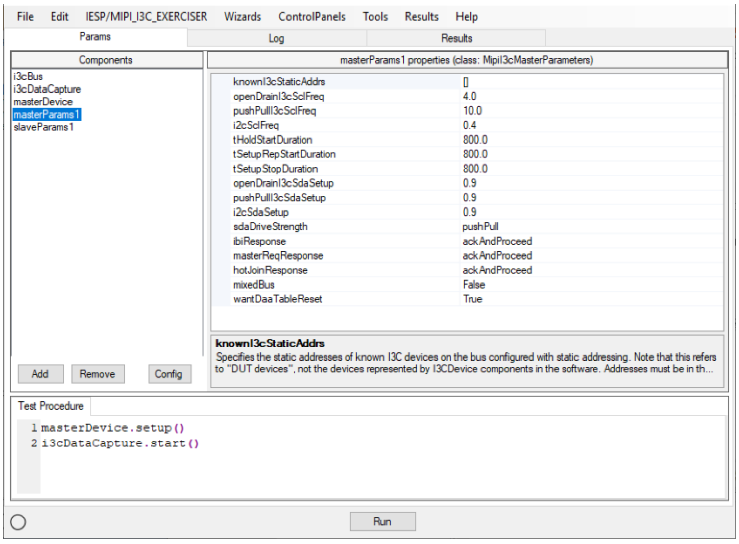
d) By default, when started in the SV4E_2L2G_MIPI_I3C_EXERCISER form factor, the GUI contains two commands in the Test Procedure window and five pre-populated components in the "Components" window pane of the Params tab, as shown in the screen shots on the following page. When executed, the masterDevice.setup() method configures a master device onto the I3C bus. Additionally, the i3cDataCapture.start() will start a bus capture.

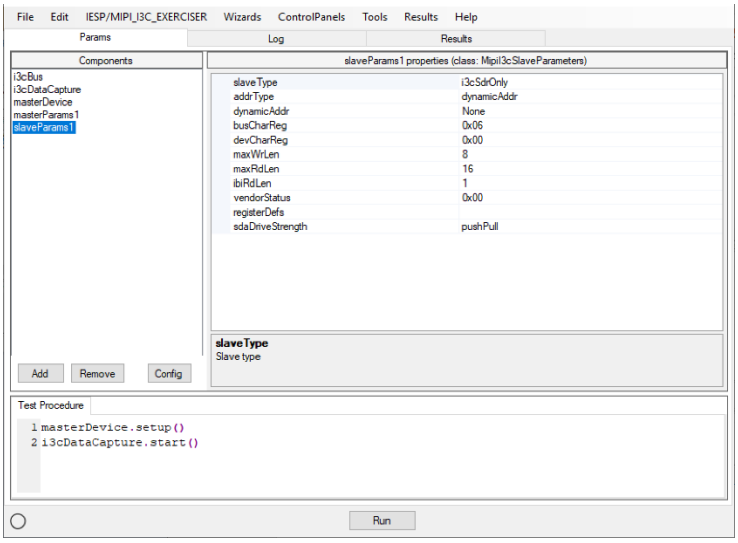




e) By default, the "Components" window pane on the left side of the GUI contains a single master device, masterDevice. Note that each device can act as either a slave, a master, or be offline on start-up, and this is specified by the startupState parameter. Also note that there can only be a single active master at a time on the SV4E-I3C module.

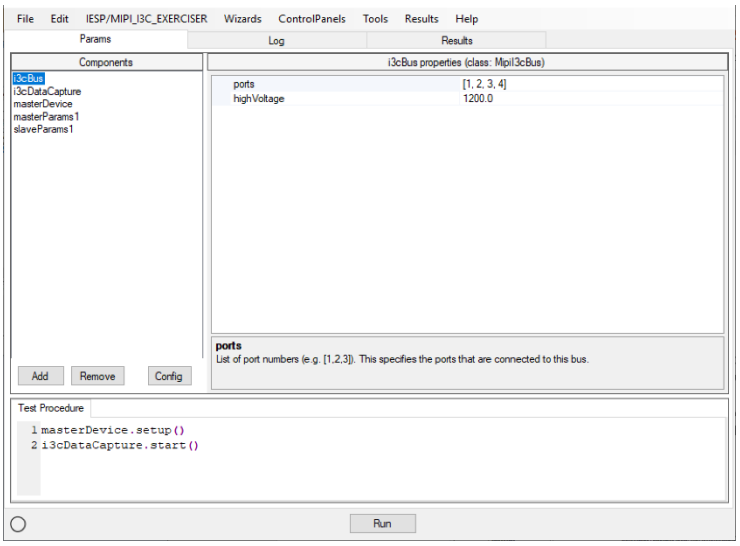
f) The masterParams1 and slaveParams1 components are used to specify the behaviour of the i3cDevice component class when it is in either mode. Note that every I3C device instance can have both master and slave parameters, regardless of its startupState, as the SV4E-I3C module supports mastership requests. Also, if a device parameter is changed during the test procedure, the update() method must be called on the i3cDevice for the new parameter to take effect.



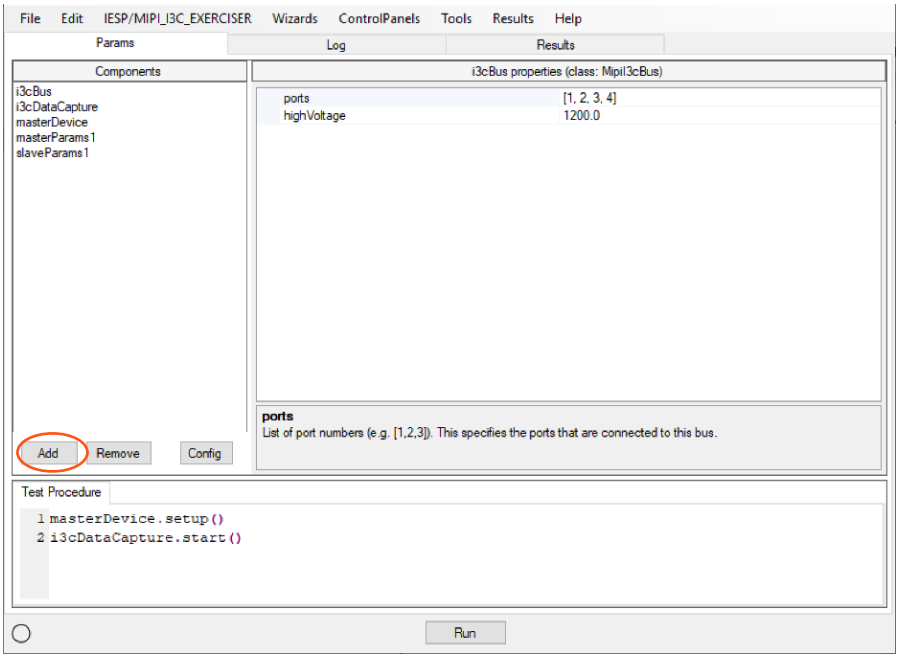


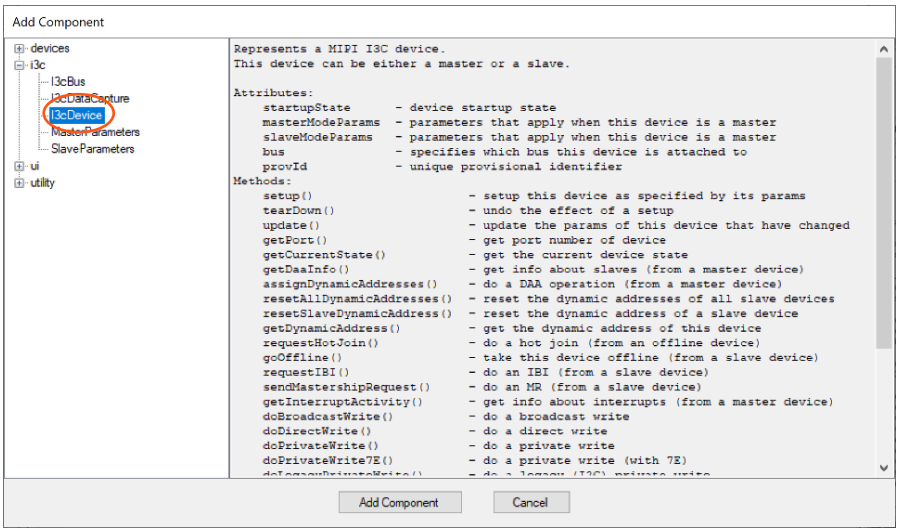
g) Observe also the i3cBus component. This component specifies the list of ports which may be connected to the named the bus. An assignment of a port on a bus occurs when a device is set up within a test (when either a "masterDevice.setup()" or "slaveDevice.setup()" is executed, as later in this document).

The i3cBus component also specifies the highVoltage level for SCL and SDA. This voltage is a global setting affects all ports on the i3cBus component.



h) In the example test to be created below, a master device will be sending commands to a slave device. To add a slave i3cDevice for this test, click the "Add" button in the "Components" window pane on the left side of the Introspect ESP Software window. In the resulting window, select i3cDevice from the list, then click "Add Component". These operations are as shown in the figures on the next page.


File
Edit
IESP/MIPI_I3C_EXERCISER
Wizards
ControlPanels
Tools
Results
Help
Params
Log
Results
Components
i3cBus
i3cDataCapture
masterDevice
masterParams1
slaveParams1
i3cBus properties (class: MipiI3cBus)
ports
[1, 2, 3, 4]
highVoltage
1200.0
ports
List of port numbers (e.g. [1,2,3]). This specifies the ports that are connected to this bus.
Add
Remove
Config
Test Procedure
1 masterDevice.setup()
2 i3cDataCapture.start()
Run

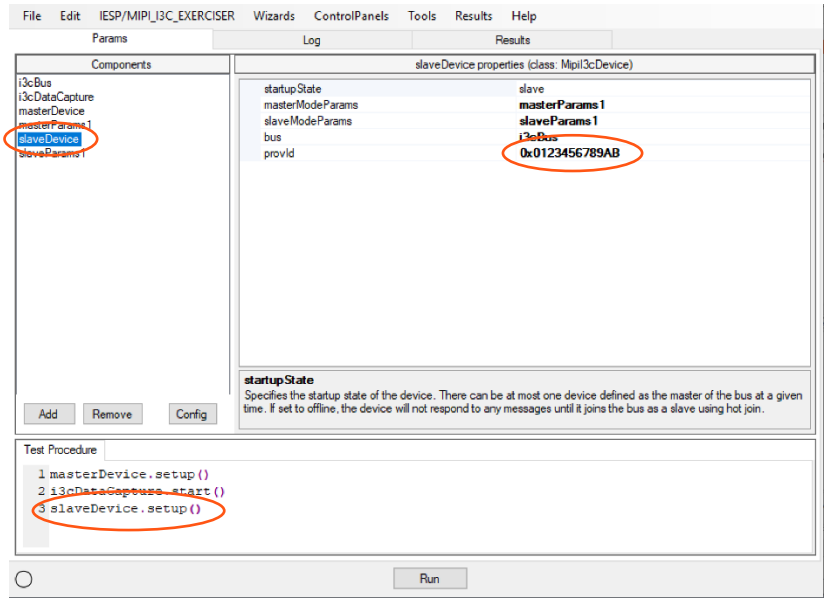

Add Component
devices
i3c
I3cBus
I3cDataCapture
I3cDevice
MasterParameters
SlaveParameters
ui
utility
Represents a MIPI I3C device.
This device can be either a master or a slave.
Attributes:
startupState - device startup state
masterModeParams - parameters that apply when this device is a master
slaveModeParams - parameters that apply when this device is a master
bus - specifies which bus this device is attached to
provId - unique provisional identifier
Methods:
setup() - setup this device as specified by its params
tearDown() - undo the effect of a setup
update() - update the params of this device that have changed
getPort() - get port number of device
getCurrentState() - get the current device state
getDaaInfo() - get info about slaves (from a master device)
assignDynamicAddresses() - do a DAA operation (from a master device)
resetAllDynamicAddresses() - reset the dynamic addresses of all slave devices
resetSlaveDynamicAddress() - reset the dynamic address of a slave device
getDynamicAddress() - get the dynamic address of this device
requestHotJoin() - do a hot join (from an offline device)
goOffline() - take this device offline (from a slave device)
requestIBI() - do an IBI (from a slave device)
sendMastershipRequest() - do an MR (from a slave device)
getInterruptActivity() - get info about interrupts (from a master device)
doBroadcastWrite() - do a broadcast write
doDirectWrite() - do a direct write
doPrivateWrite() - do a private write
doPrivateWrite7E() - do a private write (with 7E)
Add Component
Cancel

i) The new i3cDevice will be added to the list of components. To use this device as a slave device on start-up, select the new component from the "Component" window pane and ensure the startupState value is set to "slave". The slave device will be configured using the values defined in "slaveParams". Note that the i3cDevice which was just added will be configured onto the bus by the automatic addition of i3cDevice1.setup() statement in the test procedure.

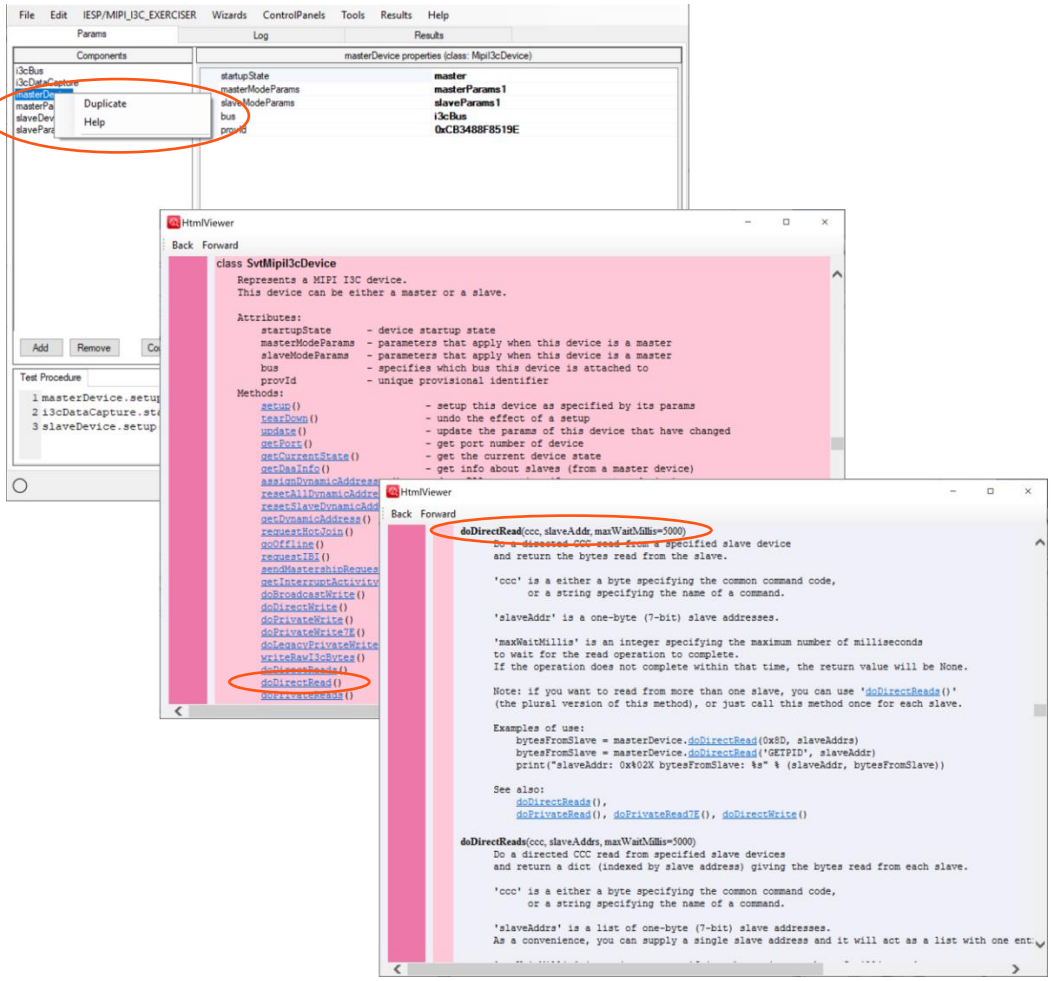
A 6-byte provisional ID is set in the i3CDevice component as shown in the newly added component. The provisional ID may be edited, for example to 0x0123456789AB as below.

It is useful to rename the newly added i3cDevice to something more descriptive. In this example, it will be renamed it to "slaveDevice" for convenience, since it will be used only as a slave. Notice the last line of the test procedure will automatically change to slaveDevice.setup() to track the renaming of this component.

The combined operations of adding, editing, and renaming of the i3cDevice1 component above result in the new "slaveDevice" component as shown in the figure below.



j) The best way to find further information on components and formats is to "**right click"** on the component and open the pop-up help file. The three figures on the following page show how to obtain help for a "doDirectRead" from the I3C Master Device.



The example which follows implements a direct read, as documented above.

3. EDITING THE TEST PROCEDURE

Now that all the necessary components have been added to the test procedure, you are ready to send your first I3C commands using the SV4E-I3C module and control the masterDevice and slave. The example below will demonstrate how to set a dynamic address to a slave device, read back the address, and then perform a direct read of the slave's provisional ID.

a) First, navigate to the bottom of the "Test Procedure" pane and notice that there are already three commands in the editor (as shown in the previous two figures). The first one, masterDevice.setup(), is used to initialize the master device already on the bus by default. The second one, i3cDataCapture1.start(), is used to start the capture of the I3C bus waveform and its analysis. Finally, the third one, slaveDevice.setup(), is used to initialize the slave device we added in the previous section.

b) The code listed below will allow you to assign a dynamic address to the connected slave device and print out the resulting address. Simply add the following lines to the "Test Procedure" editor pane situated at the bottom of the main GUI window:

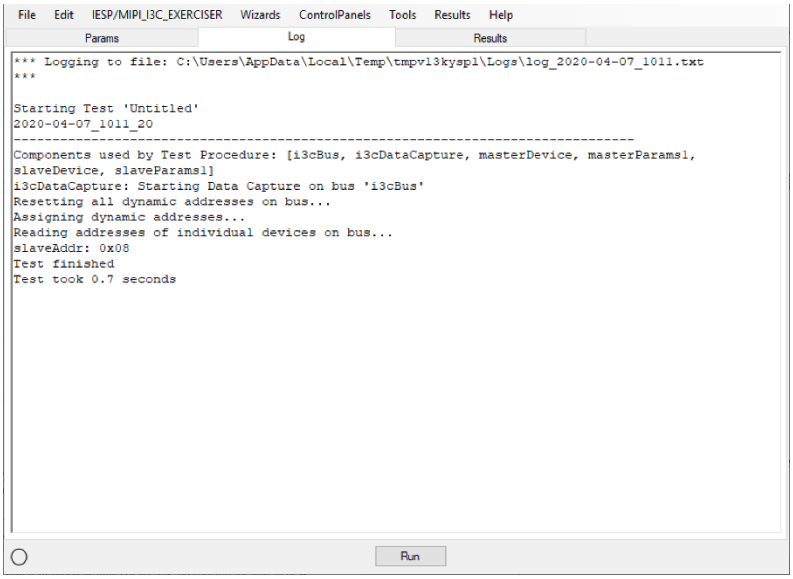
```
#Reset all Dynamic Addresses
print("Resetting all dynamic addresses on bus...")
masterDevice.resetAllDynamicAddresses()

#Do dynamic address assignment
print("Assigning dynamic addresses...")
masterDevice.assignDynamicAddresses()

#Get the slave's newly assigned dynamic address through software
print("Reading addresses of individual devices on bus...")
mySlave = slaveDevice.getDynamicAddress()
print("slaveAddr: 0x%02X " % (mySlave))
```

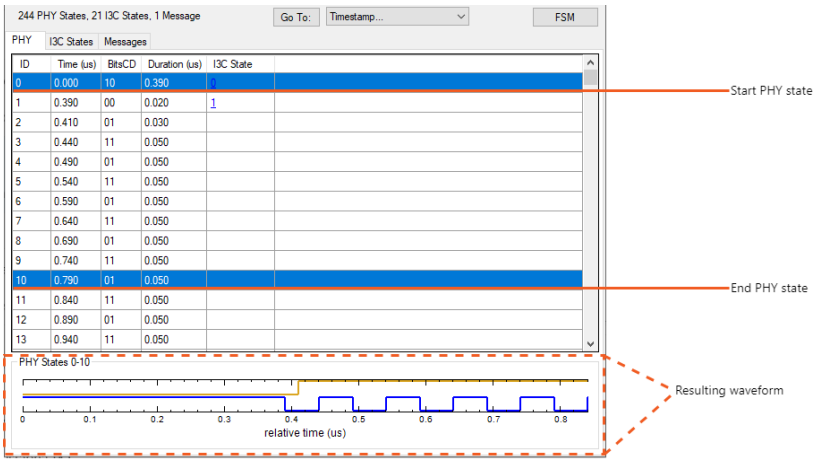
For more details on how to use these and other available methods, please consult the "Help" pull-down menu entry from the main Introspect ESP GUI window, or "**right click"** on the component in question in the component window to open the pop-up help file, as described previously.

If you click the "Run" button at the bottom of the GUI window, the Introspect ESP Software will execute the script. Switch to the "Log" tab, where you will see the messages from your test procedure, as shown in the image below. As seen below, the slave device was assigned address 0x08.



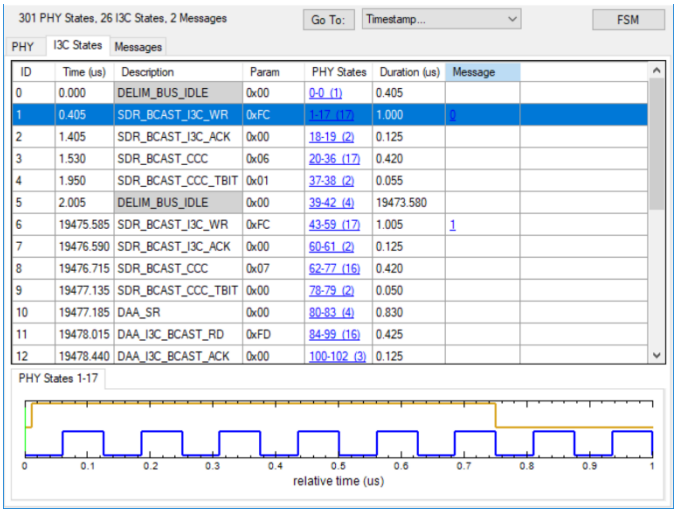
c) Since you have previously enabled the data capture component, you can visualize the I3C commands sent to the slave device to assign a dynamic address to it. To do so, navigate to the "Results" tab at the top of the GUI window, select the most recent run result and double click the "i3cDataCapture1" data capture icon. The data capture viewer is divided in three tabs: "PHY", "I3C States" and "Messages" and will allow you to see and visualize all of the bus activity.

d) The **"PHY"** tab shows the different physical states of the I3C bus during the run. A new entry will be added to this tab every time the I3C bus changes state. Note that you can visualize the waveform over several PHY states by selecting a start PHY state, pressing and holding the CTRL key on your keyboard, and selecting a second end PHY state, as shown in the image on the following page.

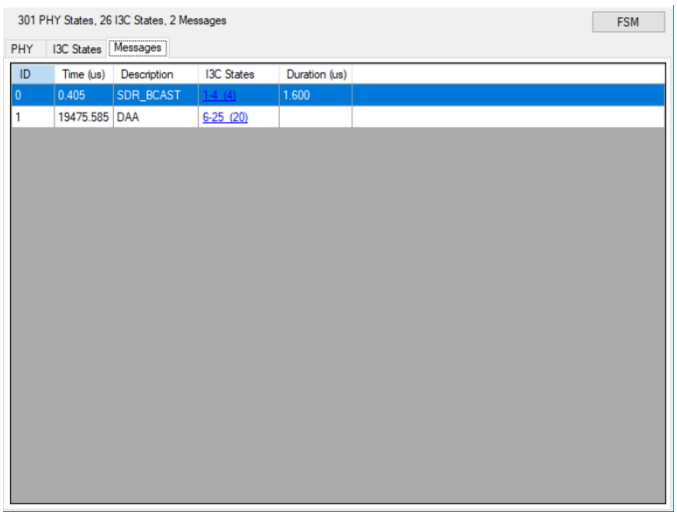


Note that the top waveform is for the SDA line and the bottom is for the SCL line.

e) The **"I3C States"** tab allows you to visualize the different I3C commands that were transmitted on the bus during the run. Just like in the previous tab, you can visualize the waveform of one or multiple I3C commands by selecting it from the list.



f) The third and final tab, **"Messages"**, allows the user to quickly identify the function of a group of I3C commands. In the current example, only two messages are presented. The SDR_BCAST is an automated message that the master sends to say SDR mode is being used. Then we see the DAA event we executed, listed as DAA for Dynamic Address Assignment.



g) Now that you have been able to assign an address to the slave device, you can send I3C commands to it over the SCL/SDA wires. For example, to read the provisional ID of the slave device (previously programmed through the slaveDevice.setup() operation), you will need to send a direct read command to the slave device with the direct command code "GETPID".

To do so, navigate back to the "Params" tab at the top of the GUI window and add the following code to your Test Procedure:

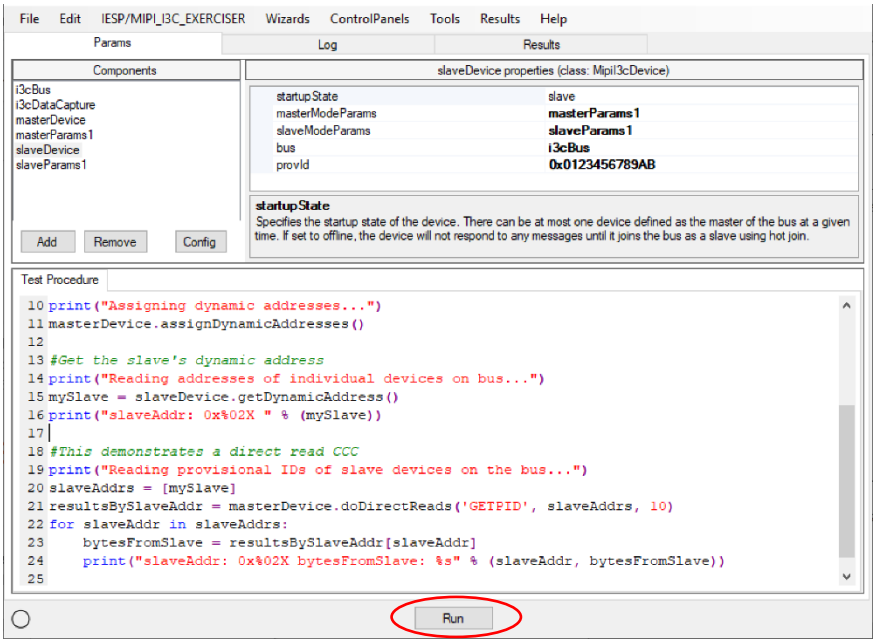
```
#This demonstrates a direct read CCC
print("Reading provisional IDs of slave devices on the bus...")
slaveAddrs = [mySlave]
resultsBySlaveAddr = masterDevice.doDirectReads('GETPID', slaveAddrs, 10)
for slaveAddr in slaveAddrs:
    bytesFromSlave = resultsBySlaveAddr[slaveAddr]
    print("slaveAddr: 0x%02X bytesFromSlave: %s" % (slaveAddr, bytesFromSlave))
```

For more details on the different types of commands supported by the Introspect ESP Software and how to use them, please refer to the "Help" manual, available by right clicking a component and selecting the "Help" option, as shown previously.

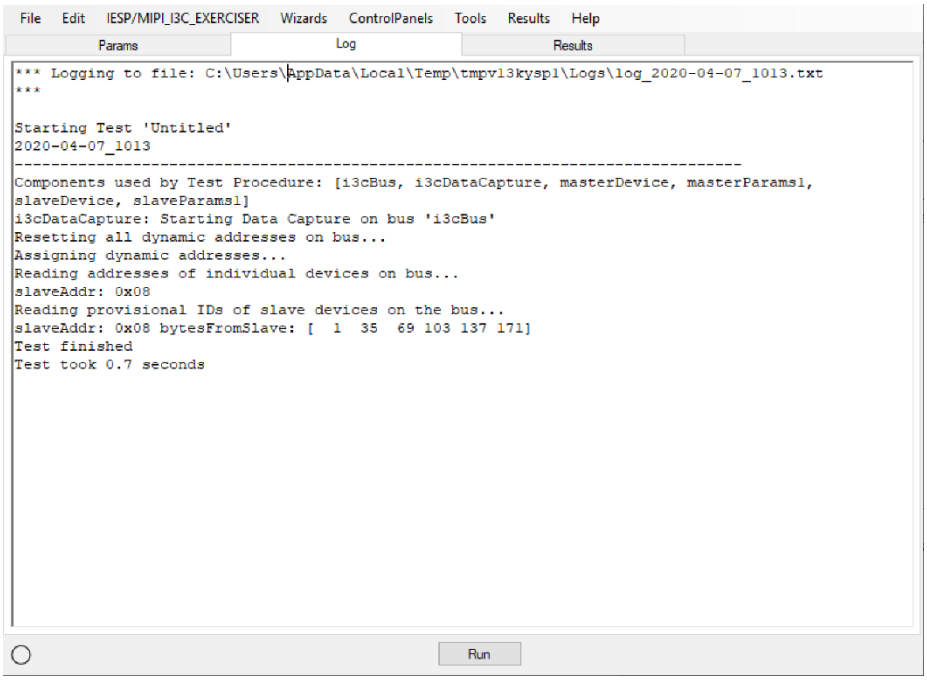
Your final test procedure is now ready to be run.

4. EXECUTE THE FINAL TEST

a) You have run some partial tests in the previous section. Here, it is useful to execute the final test, and we do so by clicking Run again at the bottom of the Introspect ESP Software. Alternatively, use the shortcut key F5 to rapidly run a Test Procedure.

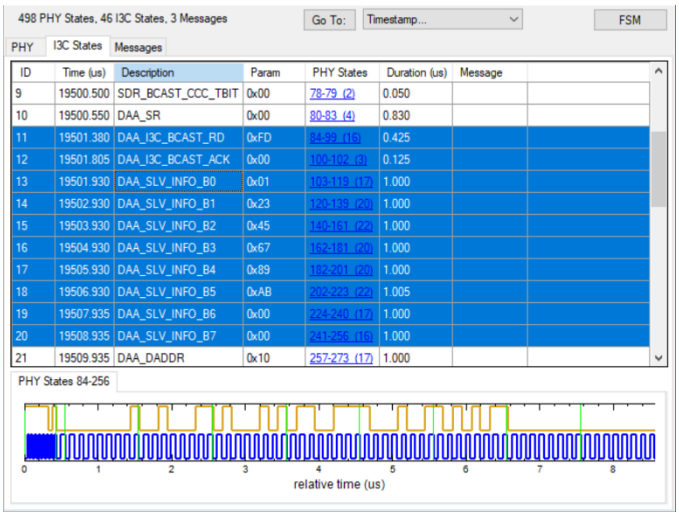


The Introspect ESP Software will execute the script. Switch to the "Log" tab, where you will see the resulting messages from your test procedure, as shown in the image on the next page. Observe that the provisional ID read back is six bytes log (each byte in the log is shown in decimal), matching the six hex bytes set previously in the slave parameters.



b) Once the test has finished, switch to the "Results" tab where you can visualize the waveform of the I3C communications that have occurred during the test run.

Double click the "i3cDataCapture1" file in the "Results" window. There, navigate to the "I3C States" tab and scroll down until you see a "SDR_DIR_RD_DATA" state. As show below, the "Param" column of the I3C State indicates the value that was transmitted during the direct read command. Observe the expected PID value that was set in the slave parameters.



This concludes the Quick Start SV4E-I3C demonstration. For further information, note that the user manual is available from the “Help -> User Manual” pull-down menu entry from the main Introspect ESP window.

Appendix

FTDI DRIVER MANUAL INSTALLATION

The Introspect ESP Software communicates with the SPI Controller via an FTDI device (connected via USB). If you don't already have required FTDI drivers installed on your Windows computer, or if the automated driver detection presented earlier in this document was unsuccessful, you will need to download them from the FTDI web site. To do this, follow the instructions found at

http://www.ftdichip.com/Documents/InstallGuides.htm

The latest drivers can be found at

http://www.ftdichip.com/Drivers/D2XX.htm

Note that the driver version used in our product development is 2.12.

You may wish to use the "usbview" utility program linked to on the following FTDI page:

http://www.ftdichip.com/Resources/Utilities.htm

This program will allow you to check that your computer can "see" the FTDI device over USB.

Revision Number	History	Date
1.0	Document Release	June 7, 2019
1.1	Modifications reflecting the user interface in version 3.6.74	August 30, 2019
1.2	Modifications reflecting the user interface in version 3.6.90	April 9, 2020

The information in this document is subject to change without notice and should not be construed as a commitment by Introspect Technology. While reasonable precautions have been taken, Introspect Technology assumes no responsibility for any errors that may appear in this document.

Published in Canada on April 9, 2020
EN-G035E-E-20100

INTROSPECT.CA