

## QUICK START GUIDE



I3C Test and Debug Module

# **E SERIES**





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

INTROSPECT.CA



# Table of Contents

Introduction	3
Overview Quick Start Documentation Quick Start Hardware Description	3 3 <b>3</b>
Requirements Hardware Description Introspect ESP Software Installation	3 4 5
System Requirements Introspect ESP Software Installation USB Driver Installation	5 5 8
SV4E-I3C Demonstration	0
Step-by-Step Guide: Dynamic Addresses and Direct Reads1	0
Appendix2	5
FTDI Driver Manual Installation	5



# 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 G Specify the hardware as "SV4E\_2L2G\_MIPI\_I3C\_EXERCISE 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.

ow of the GU	. introspect
	v bechnology
C_LALIACISLIA	Welcome to Introspect ESP
	To narrow down the choices in the menu below, type part of the desired form factor name in the "Filter" box.
	Fiter:
	Choose the IESP form factor
	SV4E_2L2G_MIPI_I3C_EXERCISER V
	SV3C-32C12G SV3C-32C14G
	SV3C-32C14G_MD SV3C-32T18BC12G
	SV3C_32T24RC12G
	SV3C_4L3G_MIPI_CPHY_GENERATOR
	SV3C_4L3G_MIPLDPHT_ANALTZERZ SV3C_4L3G_MIPLDPHY_GENERATOR_P
	SV3C_4L6G_MIPI_DPHY_GENERATOR SV3D_32C14G
	SV4_5507 SV4D_4L2G_MIPI_DPHY_GENERATOR
	SV4E 2L2G MIPI 13C EXERCISER SV4E 3L2G MIPI CPHY ANALYZER
aletta a "Navet"	Introspect ESP ×
ck the next	$\frown$
	introspect
	Walcome to Interest ESD
	To exercise the design in the many below
	type part of the desired form factor name in the "Filter" box.
	Fiter:
	State 2120 MIPLING EXERCISER
	Svec_2220_mint_JSC_Exencisen *
	ETC. ADDLE. Freiterices
	Next 1
	I DOWN
File Edit IESP/MIPI_I3C_EXERCISER	Wizards ControlPanels Tools Results Help
Components	i3cBus properties (class: Mpil3cBus)
3cBus 3cDataCapture	ports [1, 2, 3, 4]
masterDevice masterParams1	high Voltage 1200.0
slaveParams1	

matter Parma 1 alave Parama 1	
Add Remove Config	ports List of port numbers (e.g. [1,2,3]). This specifies the ports that are connected to this bus.
Test Procedure	
<pre>1 masterDevice.setup() 2 i3cDataCapture.start()</pre>	
0	Bun

×



#### 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.

File Edit IE	ESP/MIPI_I3C_EXERCISE	R Wizards Contro	trolPanels Tools Results Help
	Connect	Log	Results
	Disconnect		i3cBus properties (class: MpiI3cBus)
(3cBus (3cDatsCaptur masterParans) slaveParans1 slaveParans1	Run Test Run Test After Delay Status Reset Options	FS B y Vokage	1200 POLICIA 1200 P
Add Re	move Config	ports List of port numbers (e	in g. [1,2,3]. This specifies the ports that are connected to this bus.
1 masterD 2 13cData	evice.setup() Capture.start()		

Personality:	FWSV4E4I3C01C013	
Connected:	True	
Temperature:	38	
Status:	0000000	

#### 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 mipil3cDataCapture 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.

Kenter Strate St	MIPI_I3C_EXERCISER Status	×
Personality: Connected:	FWSV4E4I3C01C013 True	
Temperature: Status:	38	
Status.	000000	

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.



File Edit IESP/MIPI_I3C_EXERCISER	Wizards ControlPanels	Tools Results Help
Params	Log	Results
Components		masterDevice properties (class: Mipil3cDevice)
13-blus 13-blusCapture maater Params 1 slaveParams 1	etarupState masterModeParams slaveModeParams bus provid	master masterParams 1 i3aveParams 1 i3cBus QxCB3488F8519E
Add Remove Config	startupState Specifies the startup state of the d time. If set to offline, the device wi	evice. There can be at most one device defined as the master of the bus at a given not respond to any messages until it joins the bus as a slave using hot join.
Test Procedure 1 masterDevice.setup() 2 i3cDataCapture.start()		
0		Run
File Edit IESP/MIPI_I3C_EXERCISER	Wizards ControlPanels	Tools Results Help
Companyate	LOG	Dete Creative annual (classe Mini (2-Dete Creative)
i3cBus	13	22-Dura
ISODefaCapture masterDevice masterParams 1 slaveParams 1	bus want Analysis save Results	i3chus True True
Add Remove Config Test Procedure	bus A component of the class Svt Mipi	13cBus which specifies the bus to capture.
<pre>I masterDevice.setup() 2 i3cDataCapture.start()</pre>		
0		Run

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.

Params	Log	Results				
Components	masterPara	ms1 properties (class: Mipil3cMasterParameters)				
cBus	known/3cStaticAddm	Π				
DataCapture	openDrain13c SciFreq	40				
sterDevice	pushPull/3cSc/Freq	10.0				
veParame1	i2cSclFreq	0.4				
ver arams r	tHoldStartDuration	800.0				
	tSetup RepStartDuration	800.0				
	t Setup Stop Duration	800.0				
	openDrainI3cSdaSetup	0.9				
	pushPull/3cSdaSetup	0.9				
	i2cSdaSetup	0.9				
	sdaDriveStrength	pushPull				
	ibiResponse	ack And Proceed				
	masterRegResponse	ack And Proceed				
	hotJoin Response	ack And Proceed				
	mixedBus	False				
	wantDaaTableReset	True				
st Procedure						
st Procedure 1 masterDevice.setup() 2 i3cDataCapture.start	0					
at Procedure 1 masterDevice.setup() 2 13cDataCapture.start	O Pu	1				
at Procedure 1 masterDevice.setup() 2 13CDataCapture.start e Edit IESP/MIPLJ3C_EXERC	() SER Wizards ControlPanels Tools	n Results Help				
d Procedure I masterDevice.setup() 2 13cDataCapture.start E Edit IESP/MIPLJSC_EXERC Params	() Fu ISER Wizards ControlPanels Tools Log	n Results Help Results				
nt Procedure I masterEvrice.setup() 2 13cDataCapture.start e Edit IESP/MIPLJ3C_EXERC Params Components	() ISER Wizards ControlPanels Tool: Log alavePara	n Results Help Results ms1 properties (class: MpII3c:SlaveParameters)				
e Edit IESP/MIPLJ3C_EXERC Params Components Bus	() SSER Wizards ControlPanels Tools Log avePara avePara	n Results Help Penuts ns 1 properties (class: Mipil3:SlaveParameters) 13:Sc30-Only				
at Procedure at Procedure 1 asterEPerice.setup() 2 13cDataCapture.start 2 13cDataCapture.start Captor Start Components But DerScapture	() SER Wizards ControlPanels Tools Log advePare adveType adveType	n Results Help Results ms1 properties (dass: Mpil3s:SlaveParameters) 13s:SdOnly dynamic.Addr				
e Edit IESP/MIPLISC_EXERC Params Components DasCapture Base Components DasCapture terParams	() ISER Wizards ControlPanels Tools Log slaveType addType dymanicAdr	n Results Help Results ns 1 properties (class: MpI/3cSlaveParameters) (3cSdrOnly dynamic/ddr None				
at Procedure at Procedure at Procedure at ansate::Pervice.setup() 2 13cDataCapture.start 2 13cDataCapture.start Captor at a start DataCapture terDevice terDevice terDevice terDevice	() ISER Wizards ControlPanels Tools Log alaveType addrType addrType dynamcAddr busCharReg	n Results Help Results ms1 properties (dass: Mpil3cSlaveParameters) i3cSdOnly dynamicAdar None QuO6				
e Edit IESP/MIPUJ3C_EXERC Parama Components Dus DaGachure terDevice terDevice terDevice terDevice	() ISER Wizards ControlPanels Tools Log slaveType addType dymmicAddr busCherReg devCharReg	Results Help Results Help ms1 properties (class: MpI3:S3aveParameters) (3:S34Only dynamic/dafr None 0x06 0x00				
at Procedure at Procedure 1 master:Device.setup() 2 1 3cDataCapture.start 2 1 3cDataCapture.start Capacity of the setup of the set	() SER Wizards ControlPanels Tools Log slaveType addType dynamicAddr bucDnaffeg devChaffeg maxWiLian	n Results Help Results ms1 propeties (dass: Mipl3c:SlaveParameters) (3:S40/oly dynamicAddr None Ox06 0x00 8				
at Procedure at Procedure 1 asterEDVICe.setup() 2 1 3cDataCapture.start 2 1 3cDataCapture.start 2 1 3cDataCapture.start Params Components Dus Dus Dus Dus Dus Dus Dus Du	() ISER Wizards ControlPanels Tools Log addType dynamicAdr budDarReg devDarReg maxWiten maxRelen	Results Help Results ns1 properties (class: MpII3cStaveParameters) i3cScHOrdy dynamicAddr None 0x06 0x00 8 16				
et Procedure 1 masterDevice.setup() 2 13CDataCapture.start 2 13CDataCapture.start Edit IESP/MIPLJ3C_EXERC Params DataCapture MatCap	() SER Wizards ControlPanels Tools Log davePara ddavePara ddavePara ddavePara ddaveReg devCharReg devCharReg maxRdLen tikRdLen tikRdLen	n Results Help Results mail properties (class: Mipil2:SlaveParameters) (2:Sc80-Only dynamicAddr None Dof5 bx00 8 15 1				
et Procedure I masteztevice.setup() 2 13cDataCapture.start 2 13cDataCapture	() ISER Wizards ControlPanels Tool: Log adarType dynamicAddr budCharReg maxWiLen maxReLen titReLen titReLen	Results Help Results Help 13:540-0y 0006 0x00 8 16 1 0x00				
Test Procedure Test Procedure 1 masterEDevice.setup() 2 13cDataCapture.start ) ) ile Edit IESP/MIPIJSC_EXERC Params Components Edit -DataCopture anterDavice anterDavice anterDavice	() ISER Wizards ControlPanels Tools Log davePara daveType ddvCharReg maxWilen maxRellen biRdLen vendorStatus revisterDefs	Results Help Results Help i32:54Orby dynamicAddr None 0x06 0x00 8 16 1 1 0x00				

Run

slaveType

Add Remove Config

1 masterDevice.setup()
2 i3cDataCapture.start()

Test Procedure

0



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.

File Edit IESP/MIPI_I3C_EXERCISER	Wizards ControlPanels	Tools Results Help	
Params	Log	Results	
Components		i3cBus properties (class: Mipi13c8	lus)
100000 1320bat/200 masterParans 1 slaveParans 1	ports high Voltage	[1, 2, 3, 4]	
Add Remove Config	ports List of port numbers (e.g. [1.2,3]	). This specifies the ports that are connect	ed to this bus.
Test Procedure 1 masterDevice.setup() 2 i3cDataCapture.start()			
0		Run	

 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_EXERC	ISER Wizards ControlPanels Tools Results Help	
Params	Log Results	
Components	i3cBus properties (class: Mipil3cBus)	
iGoBus i3cDataGaptire masterDevice masterParams1 slaveParams1	pots [1, 2, 3, 4] highVoltage 1200.0	
Add Remove Config	ports List of port numbers (e.g. [1,2,3]). This specifies the ports that are connected to this bus.	
Test Procedure 1 masterDevice.setup() 2 i3cDataCapture.start	0	
	Run	
Hou Component     Green devices     Green devices     StateParameters     StateParameters     uity	Represents a MIPI ISC 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 a trached to provId - unique provisional identifier Methods: setup() - setup this device as specified by its params tearDown() - undo the effect of a setup update() - get port number of device getCurrentState() - get the current device state getDart() - get info about slaves (from a master device) assignDynamicAddress() - reset the dynamic address of all slave devices resetSlaveDynamicAddress() - get info about slaves of all slave device getDynamicAddress() - get the dynamic address of all slave device getDynamicAddress() - get his device offline (from a slave device) requestBit() - do a NR (from a slave device) sendMastershipRequest() - do an NR (from a slave device) gedDifine() - take this device offline (from a master device) sendMastershipRequest() - do a nr Slave device) deDrowdcastWrite() - do a private write doDrivateWrite() - do a private w	~
	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.

File Edit IESP/MIPI_I3C_EXER	CISER Wizards ControlPanels	Tools Results Help	
Params	Log	Results	
Sams *** Logging to file: C: *** Starting Test 'Untitled 2020-04-07_1011_20 Components used by Test SlaveBevice, slaveParam i3CDtaCapture: Startin Resetting all dynamic a Assigning dynamic addre Reading addresses of in slaveAddr: 0x00 Test finished Test took 0.7 seconds	USers/AppData/Local/Temp Procedure: [130Bus, 130 3] g Data Capture on bus '13 diresses on bus sees dividual devices on bus	nesus p\tmpvl3kyspl\Logs\log_2 DataCapture, masterDevic 3cBus' 	020-04-07_1011.txt  e, masterParams1,

- 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.



244	PHY States, 2	1 I3C State	es, 1 Message		Go To:	Timestamp		~		FSM		
PHY	I3C States	Message	s									
ID	Time (us)	BitsCD	Duration (us)	I3C State							^	
0	0.000	10	0.390	0								Start PHV star
1	0.390	00	0.020	1								Start Hiri Sta
2	0.410	01	0.030									
3	0.440	11	0.050									
4	0.490	01	0.050									
5	0.540	11	0.050									
6	0.590	01	0.050									
7	0.640	11	0.050									
8	0.690	01	0.050									
9	0.740	11	0.050									
10	0.790	01	0.050									End PHV stat
11	0.840	11	0.050									End PTT stat
12	0.890	01	0.050									
13	0.940	11	0.050								~	
PHY	States 0-10										-	<b>+</b> _
							,				-	
					<u> </u>	<u> </u>	<u> </u>		). г	<b>_</b> .	1 I	Resulting waveform
0	0.1		0.2	0.3	0.4 relative tim	0.5 e (us)	0.6	5	0.7	0.8		11

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.

301 PHY States, 26 I3C States, 2 Messages Go To: Timestamp V FSM								
PHY I3C States Messages								
ID	Time (us)	Description	Param	PHY States	Duration (us)	Message		^
0	0.000	DELIM_BUS_IDLE	0×00	<u>0-0 (1)</u>	0.405			
1	0.405	SDR_BCAST_I3C_WR	0xFC	<u>1-17 (17)</u>	1.000	Q		
2	1.405	SDR_BCAST_I3C_ACK	0×00	<u>18-19 (2)</u>	0.125			
3	1.530	SDR_BCAST_CCC	0x06	20-36 (17)	0.420			
4	1.950	SDR_BCAST_CCC_TBIT	0x01	37-38 (2)	0.055			
5	2.005	DELIM_BUS_IDLE	0×00	<u>39-42 (4)</u>	19473.580			
6	19475.585	SDR_BCAST_I3C_WR	0xFC	43-59 (17)	1.005	1		
7	19476.590	SDR_BCAST_I3C_ACK	0×00	<u>60-61 (2)</u>	0.125			
8	19476.715	SDR_BCAST_CCC	0x07	<u>62-77 (16)</u>	0.420			
9	19477.135	SDR_BCAST_CCC_TBIT	0×00	78-79 (2)	0.050			
10	19477.185	DAA_SR	0×00	80-83 (4)	0.830			
11	19478.015	DAA_I3C_BCAST_RD	0xFD	84-99 (16)	0.425			
12	19478.440	DAA_I3C_BCAST_ACK	0×00	<u>100-102 (3)</u>	0.125			~
PHY States 1-17								
	ا ي ا		لبر			<u> </u>	ليبر لِب	
	relative time (us)							



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.

301 P	HY States, 26	I3C States, 2 M	essages		FSM
HY	I3C States	Messages			
ID	Time (us)	Description	I3C States	Duration (us)	
)	0.405	SDR_BCAST	<u>1-4 (4)</u>	1.600	
	19475.585	DAA	<u>6-25 (20)</u>		

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.

Params       Log       Results         Components       slaveDevice properties (class: MipII3cDevice)         I3cBus       slaveDevice properties (class: MipII3cDevice)         I3cBus       slaveDevice properties (class: MipII3cDevice)         masterParama1       slaveDevice         slaveDevice       masterParama1         slaveDevice       slaveDevice         provid       0x0123456789AB         Device       Config         Test Procedure       slaveDevice.assignDynamicAddresses()         12       13 #Get the slave's dynamic addresses         14 print("Reading addresses of individual devices on bus")       slaveDevice.getDynamicAddress()         16 print("SlaveAddr: OxfO2X " % (mySlave))       *         17       slaveAddres = [mySlave]         18 #This demonstrates a direct read CCC       9 print("Reading provisional IDs of slave devices on the bus")         20 slaveAddr = masterDevice.doDirectReads('GETPID', slaveAddrs, 10)       22 for slave	File Edit IESP/MIPI_I3C_EXERCISER	Wizards ControlPanels	Tools Results Help				
Components       slaveDevice properties (class: Mipil&CDevice)         I3CBus       (slobataCapture materDevice)         materDevice       materModeParams       materParams1         slaveParams1       slaveParams1       slaveParams1         12       slaveParams1       slaveParams1       slaveParams1         12       slaveParams1       slavePa	Params	Log	Results				
ICeDus       statupState       slave         IceDus (CoDarGapture masterParams1 slaveHodeParams       slaveFarams1         slaveParams1       slaveHodeParams       slaveParams1         slaveParams1       slaveParams1       slaveParams1         slaveParams1       slaveParams1       slaveParams1         bus       i3cBus       slaveParams1         bus       i3cBus       slaveParams1         bus       i3cBus       provid         Add       Remove       Config         Test Procedure       specifies the statup state of the device. There can be at most one device defined as the master of the bus at a given time. If set to offine, the device will not respond to any messages until it joins the bus as a slave using hot join.         Test Procedure       10       print("Assigning dynamic addresses ()         12       13       #Get the slave's dynamic addresses ()       *         12       13 #Get the slave dynamic addresses ()       *         15       mySlave = slaveDevice.getDynamicAddresse ()       *         16       print("Reading addresses of individual devices on bus")       *         17       if finis demonstrates a direct read CCC       if print ("Reading provisional IDs of slave devices on the bus")         20       slaveAddrs = [mySlave]       if usveAddrs:       isaveAd	Components	slaveDevice properties (class: Mipil3cDevice)					
Add       Remove       Config       time. If set to offine, the device will not respond to any messages until tions the bus as a slave using hot join.         Test Procedure <ul> <li>In master Device . assign Dynamic Addresses ()</li> <li>g for the slave's dynamic addresses ()</li> <li>f for the slave's dynamic addresses ()</li> <li>f for the slave's dynamic addresses ()</li> <li>f for the slave slave Device .get Dynamic Addresse ()</li> <li>f print ("Reading addresses of individual devices on bus")</li> <li>f mySlave = slaveDevice.get DynamicAddress ()</li> <li>f print ("Reading provisional IDs of slave devices on the bus")</li> <li>g slaveAddrs = [mySlave]</li> <li>f resultsBySlaveAddr = masterDevice.doDirectReads ('GETPID', slaveAddrs, 10)</li> <li>f print ("slaveAddr: 0x%02X bytesFromSlave: %s" % (slaveAddr, bytesFromSlave))</li> <li>yrint ("slaveAddr: 0x%02X bytesFromSlave: %s" % (slaveAddr, bytesFromSlave))</li> </ul>	i3cBus i3cDataCapture masterDevice masterParams1 slaveDevice slaveParice slaveParams1	startup State     slave       master/ModeParams     master/Params 1       slaveModeParams     slaveParams 1       bus     i3ceBus       provid     0x0123456789AB   startup State       Soecifies the startup state of the device. There can be at most one device defined as the master of the bus at a civen					
<pre>10 print("Assigning dynamic addresses") 11 masterDevice.assignDynamicAddresses() 12 13 fdet the slave's dynamic address 14 print("Reading addresses of individual devices on bus") 15 mySlave = slaveDevice.getDynamicAddress() 16 print("BlaveAddr: 0x%02X "% (mySlave)) 17 18 fThis demonstrates a direct read CCC 19 print("Reading provisional IDs of slave devices on the bus") 20 slaveAddrs = [mySlave] 21 resultsBySlaveAddr = masterDevice.doDirectReads('GETFID', slaveAddrs, 10) 22 for slaveAddr in slaveAddrs: 23 bytesFromSlave = resultsBySlaveAddr[slaveAddr] 24 print("slaveAddr: 0x%02X bytesFromSlave: %s" % (slaveAddr, bytesFromSlave)) </pre>	Add Remove Config	time. If set to offline, the device v	vill not respond to any messages until it joins	the bus as a slave using hot join.			

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.



File	Edit IESP/MIPI_I3C_EXE	CISER Wizards ControlPanels	Tools Results Help	
	Params	Log	Results	
*** L( *** Start: 2020-(	ogging to file: C: ing Test 'Untitled 04-07_1013	\Users\AppData\Local\Tem 1	p\tmpv13kysp1\Logs\log_2	020-04-07_1013.txt
Compoi slavel i3cData Reset: Readin slavei Test : Test :	nents used by Test Device, slaveParar taCapture: Startin ting all dynamic addre ng addresses of in Addr: 0x08 ng provisional IDs Addr: 0x08 bytesF: finished took 0.7 seconds	: Procedure: [13cBus, 13c usl] ug Data Capture on bus 'i uddresses on bus sses udividual devices on bus. of slave devices on the comSlave: [ 1 35 69 10	DataCapture, masterDevic 3cBus'  bus 3 137 171]	re, masterParams1,
0		[	Run	

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.



498 P	498 PHY States, 46 I3C States, 3 Messages Go To: Timestamp V FSM						
PHY	PHY I3C States Messages						
ID	Time (us)	Description	Param	PHY States	Duration (us)	Message	^
9	19500.500	SDR_BCAST_CCC_TBIT	0×00	<u>78-79 (2)</u>	0.050		
10	19500.550	DAA_SR	0×00	80-83 (4)	0.830		
11	19501.380	DAA_I3C_BCAST_RD	0xFD	<u>84-99 (16)</u>	0.425		
12	19501.805	DAA_I3C_BCAST_ACK	0x00	100-102 (3)	0.125		
13	19501.930	DAA_SLV_INFO_B0	0x01	103-119 (17)	1.000		
14	19502.930	DAA_SLV_INFO_B1	0x23	120-139 (20)	1.000		
15	19503.930	DAA_SLV_INFO_B2	0x45	140-161 (22)	1.000		
16	19504.930	DAA_SLV_INFO_B3	0x67	162-181 (20)	1.000		
17	19505.930	DAA_SLV_INFO_B4	0x89	182-201 (20)	1.000		
18	19506.930	DAA_SLV_INFO_B5	0xAB	202-223 (22)	1.005		
19	19507.935	DAA_SLV_INFO_B6	0x00	224-240 (17)	1.000		
20	19508.935	DAA_SLV_INFO_B7	0x00	241-256 (16)	1.000		
21 19509.935 DAA_DADDR 0x10 257-273 (17) 1.000							
PHY States 84-256							
	relative time (us)						

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.

