



APPLICATION NOTE

Combined Form Factors

Using Multiple Introspect Products Simultaneously



© Introspect Technology, 2022 Published in Canada on September 30, 2022 MK-G013E-E-22273

INTROSPECT.CA



TABLE OF CONTENTS

Table of Contents

Introduction	3
Types of Combined Form Factors	.4
Channel Extension Multiple Hardware	
Creating Combined Form Factors	
Targeting Specific Hardware	. 8



Introduction

OVERVIEW

Each Introspect Technology product can be driven from the Introspect ESP Software by using the Form Factor associated with said product. If a user has many products, they can drive them via separate software instances. However, it is difficult to coordinate two separate software instances together for automation purposes.

Combined Form Factors allows users to drive multiple Introspect products from the same software, making it easy to coordinate two or more products together. They enable the creation of sophisticated test benches by coordinating different products to work together.

This feature is available on all platforms the Introspect ESP Software is supported on (Windows, Linux, macOs).



Types of Combined Form Factors

CHANNEL EXTENSION

In this mode, multiple Introspect products are used as one. It requires multiple identical boxes (same hardware & firmware). A single component, such as a TxChannelList, will be able to target multiple boxes at the same time.

Channel extension can only be used with identical SerDes products, and cannot be used with the following product lines:

- MIPI
- DisplayPort
- I3C
- SLVS

Note that channels are not aligned across boxes, although newer-generation SerDes products (SV5, SV7) contains features to guarantee rough Tx channel alignment.

The best use-case for this feature is when users need to do very similar things on identical hardware, such as running many BertScans at the same time, or setting up many patterns via a single TxChannel list. An example is highlighted in Figure 1.



When applicable, channel extension is an easy way to coordinate multiple Introspect hardware together. It is less complex than the more general method described below.

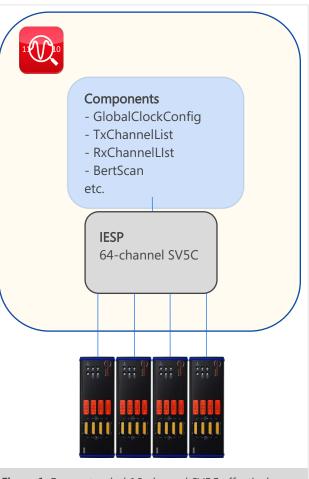


Figure 1: Four extended 16-channel SV5C effectively form one 64-channel SV5C

Introspect Technology

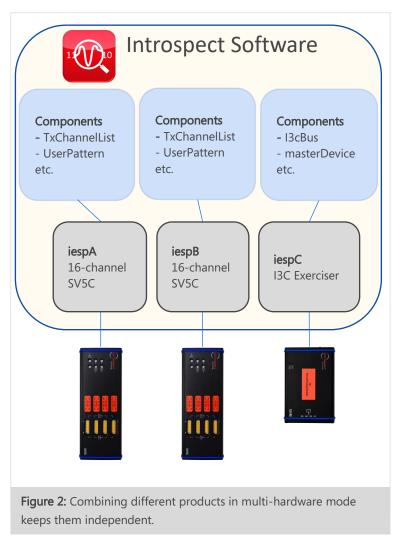


MULTIPLE HARDWARE

In this mode, multiple Introspect products are used independently. Each product has its own set of components, which cannot interact with each other, but can be used concurrently in the test procedure code. This is quite like having multiple test windows open, with each window connected to a different piece of hardware. Internally, different hardware are referred to as "iesp objects".

Unlike channel extension, any combination of Introspect products can be used together in multi-hardware mode. This flexibility enables the creation of tightly integrated test benches to validate multiple aspects of the same DUT(s). While the components for different hardware components may not be mixed, they may all be referred to from the same Python code in the test Procedure.

While channel-extended combined form factors are just used as normal, multiple hardware requires the user to select which hardware they want to configure the components on.



The example in Figure 2 depicts an example with three hardware, two of which are identical. The different hardware can be targeted from the dropdown above the component list, as seen in Figure 3. Each hardware has its own set of components. E.g.: you cannot set the RxChanneList from iespA into the BertScan of iespB. This is clearly highlighted in the GUI, as seen in Figure 4.

Note that certain utility components, such as functions or test cases, are not tied to a particular hardware, and therefore are always visible (see Figure 5).



TYPES OF COMBINED FORM FACTORS

Mintrospect ESP (v 22.4.b11) (GUI 22.4.b8) - Untitled (SV5C_DUAL_I3C) − □ ×						
File Edit IESP/SV5C_DUAL	I3C Wizards ControlPanels	Tools Results Help				
Params	Log *	Re + +				
Components (class: BertScan)						
iespA 🗸 🗸	timeUnits unitW	/idth				
bettScan1 iespA_globalClockConfig rxChannelList1 function1	rxChannelList channels expectedPatterms bertDvartionInBits 1e+00 startPhase rxChannelList RX Channel List component	6 v (B)				
Nod Nemove Lonny Test Procedure function1 1 iespA, iespB, iespC = getAllespInstances() 2 function1() 3 bertScanl.run() 4 bertScan2.run() 5 masterDevice.setup() 6						

Figure 1: (A) Use the dropdown to target different hardware. (B) Test procedure code can refer to any component.

🔯 Introspect ESP (v 22.4.b11) (GU	ll 22.4.b8) - Untitled (SV5C_DUA	AL_I3C) — □ ×				
File Edit IESP/SV5C_DUAL_I	3C Wizards ControlPanel	s Tools Results Help				
Params	Log *					
Components	Components bertScan2 properties (class: BertScan)					
iespB ~	timeUnits nxChannelList	unitWidth				
iespB_globalClockConfig function1	expected Patterns bert Duration In Bits	None Default Value New RxChannelList				
🚳 Introspect ESP (v 22.4.b11) (GUI 22.4.b8) - Untitled (SV5C_DUAL_I3C) — 🛛 🛛 🛛						
File Edit IESP/SV5C_DUAL_I3C Wizards ControlPanels Tools Results Help						
Params	Params Log * Re					
Components	bertScan1 properties (class: BertScan)					
iespA 🗸 🗸	timeUnits	unitWidth				
bertScan1 iespA_globalClockConfig rxChannelList1	expectedPatterns I bertDurationInBits	xChannelList 1 None Default Value				
function1	startPhase	New RxChannelList				

Figure 4: rxChanneList of iespA is invisible to the BertScan of IespB

Params	Log *				
Components	function 1 properties (class: Function)				
iacBus i3cDataCapture i3cProtocol masterDevice masterParams1	args code bert Scan 1.run() want All Vars Global False				
Add Remove Config					
args: 1 # Function code ca 2 print(bertScan1) 3 print(bertScan2) 4 print(masterDevice	n also refer to all components!				
Figure 5: Utility compone visible.	ents (such as functions) are always				



Creating Combined Form Factors

Creating a combined form factor is quite simple. Using your favourite text editor, create a JSON file in the folder:

Documents/Introspect/Config/CustomFormFactors/<formFactorName>.json

Windows users should find example files already present in this folder, whereas Unix users may need to create the folder.

Below are two example JSON files that combine two SV1C_8C12G together, the first one in channelextended mode, the second one in multi-hardware mode:

"n	ame": "SV1C 16C12G",
"i	espClassMap": {
	"SV1C 16C12G": "SV1C 8C12G
},	
"b	oxesMap": {
	"SV1C 16C12G": [
	"box1",
	"box2"
]
۱	1

SV1C_8C12G_DUAL.json:				
{				
"name": "SV1C_8C12G_DUAL",				
"iespClassMap": {				
"iespA": "SV1C 8C12G",				
"iespB": "SV1C [_] 8C12G"				
}				
}				
, 				

Note that the JSON format does not accept trailing commas in in lists/arrays.

Once those files are created, you can simply start the GUI and select your new form factor from the dropdown menu.

Console users can also directly initialize the form factor through the usual means, but must pass the desired lesp object when creating a component:

```
import dftm.svt as svt
iespA, iespB = svt.initFormFactor("SV1C_8C12G_DUAL")
context = svt.createComponentContext()
clockConfig_a = context.createComponent('SvtGlobalClockConfig', iespA)
clockConfig b = context.createComponent('SvtGlobalClockConfig', iespB)
```



Targeting Specific Hardware

When using multiple identical hardware boxes, it is critical to ensure the software connects to each box in the correct order. This setting is configurable from the

The Windows GUI has a utility help users modify this file. Connect and power-on all relevant introspect boxes, then access the connection editor from lesp -> ConnectionConfig menu:

These settings are saved globally on your computer under: Documents/Introspect/Config/connectionForFF/<formFactorName>.json This file is automatically created the first time a new form factor is initialized.

Console users can edit this file directly. Additionally, it is possible to the connection configuration at runtime with the svt.updateFtdiSerials() function. It is documented under the Test Procedure Functions section of the documentation.

🔍 In	trospect	t ESP (v 23.1.b0) (GUI 22.4	.b8) - Untitle	d (SV5C_DUAL_I	3C)			
File	Edit	IESP/SV5C_DUAL_I3C	Wizards	ControlPanels	Tools	Results		
		Connect		Log				
		Disconnect	f		iespA_glo	balClockC		
iespA		Run Test	F5	Rate				
liespA	_globalC	Run Test After Del	-,	dth				
		Status		enceClocks nabled				
		Reset						
		Options						
		ConnectionConfig	g					
ConnectionConfig	ConnectionConfig X							
Specify the FTDI serial numbers of the hardware box(es) to be used with this form factor. The FTDI serial numbers are written on the bottom of the boxes. Otherwise, an easy way to find out the FTDI serial numbers is to physically disconnect all boxes but one to limit the serial numbers that are available in the drop-down menus								
Hardware		Default FTDI Pattern	Matching	FtdiSerialNum	nber			
Hardware iespA (SV5C_16C12G)			Matching	PtdiSerialNun default	nber		~	
		Default FTDI Pattern			nber		~	
iespA (SV5C_16C12G)	(ER	Default FTDI Pattem ^FTDI:INSV5(CIM).*		default	nber			
iespA (SV5C_16C12G) iespB (SV5C_16C12G)	<er< td=""><td>Default FTDI Pattem ^FTDI:INSV5(CIM).* ^FTDI:INSV5(CIM).*</td><td></td><td>default default</td><td>nber</td><td></td><td>~</td><td></td></er<>	Default FTDI Pattem ^FTDI:INSV5(CIM).* ^FTDI:INSV5(CIM).*		default default	nber		~	



Revision Number	History	Date
1.0	Document Release	September 30, 2022

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.



© Introspect Technology, 2022 Published in Canada on September 30, 2022 MK-G013E-E-22273

INTROSPECT.CA