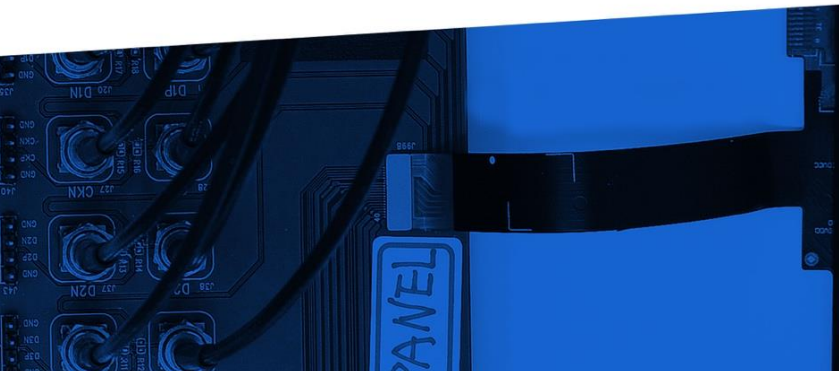
APPLICATION NOTE

Combined Form Factors

Using Multiple Introspect Products Simultaneously



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

INTROSPECT.CA

Table of Contents

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.

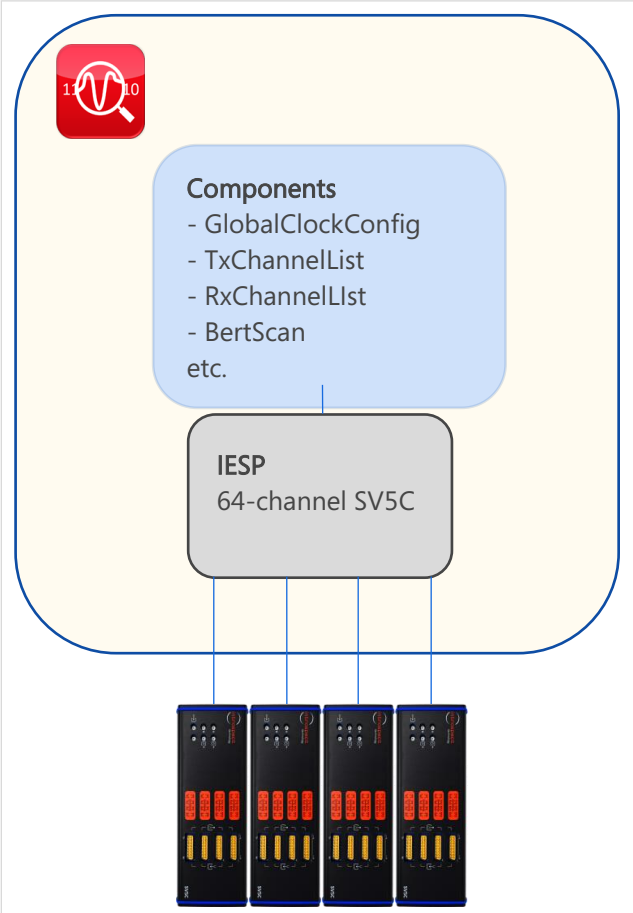


Figure 1: Four extended 16-channel SV5C effectively form one 64-channel SV5C

Once setup, the software is used in a similar fashion to the single-hardware care: the extra hardware is available as extra channels. In this example, since SV5C_16C12G has 16 channels, IespA is responsible for channels 1-16, IespB for channels 17-32, etc.

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.

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.

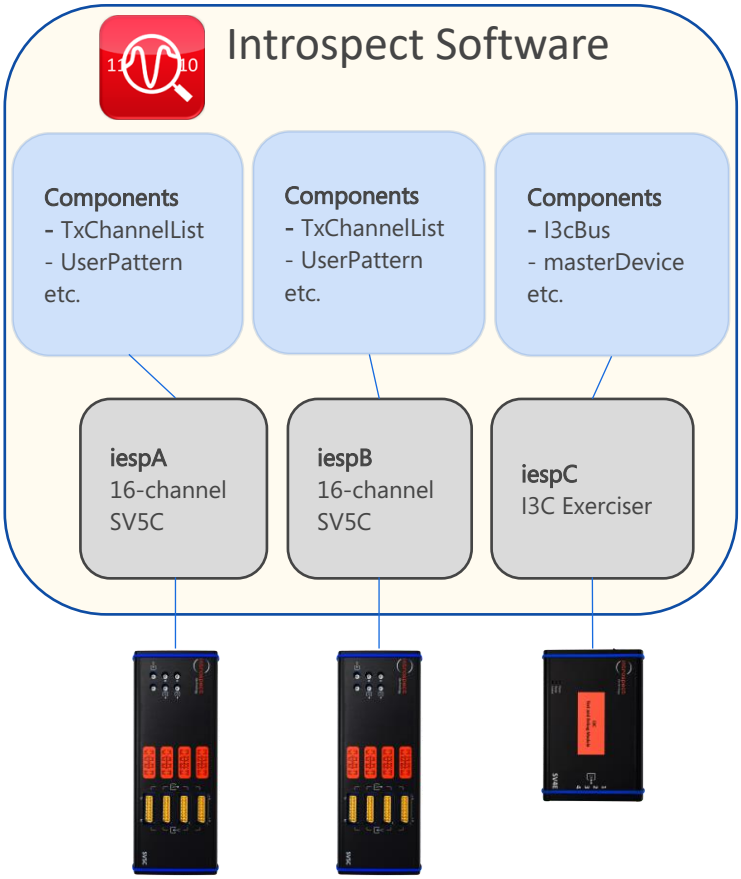


Figure 2: Combining different products in multi-hardware mode keeps them independent.

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).

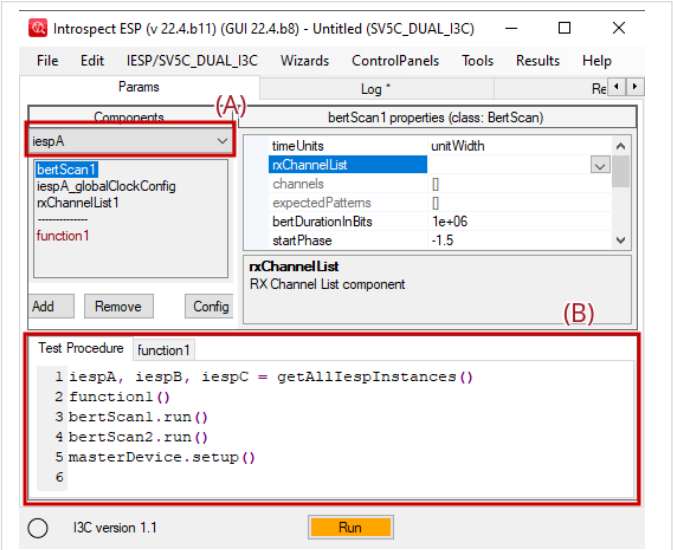


Figure 1: (A) Use the dropdown to target different hardware. (B) Test procedure code can refer to any component.

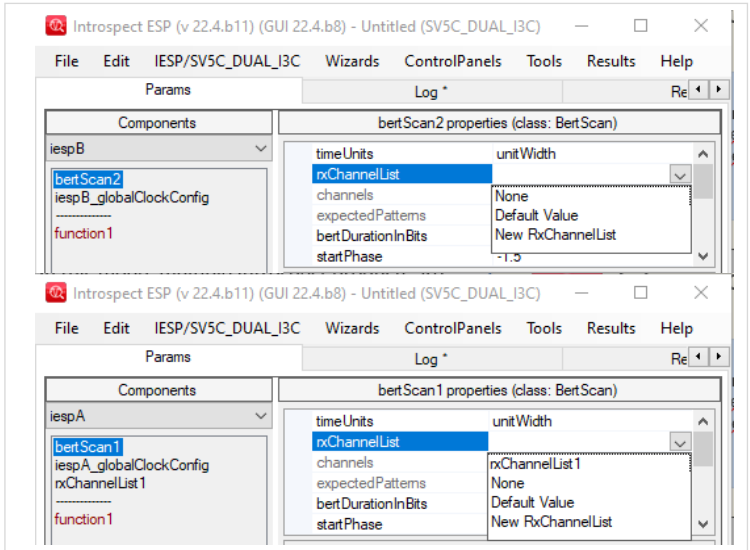


Figure 4: rxChanneList of iespA is invisible to the BertScan of IespB

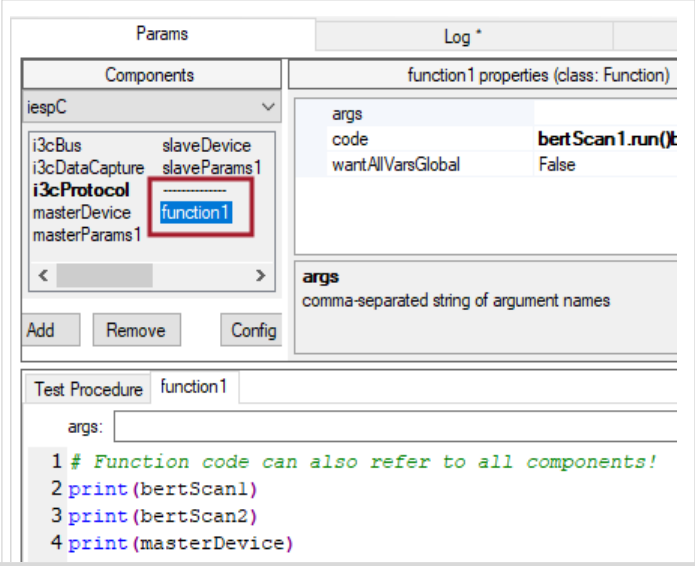


Figure 5: Utility components (such as functions) are always visible.

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 channel-extended mode, the second one in multi-hardware mode:

SV1C_16C12G.json:

```
{
    "name": "SV1C_16C12G",
    "iespClassMap": {
        "SV1C_16C12G": "SV1C_8C12G"
    },
    "boxesMap": {
        "SV1C_16C12G": [
            "box1",
            "box2"
        ]
    }
}
```

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 Iesp 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 Iesp -> 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.

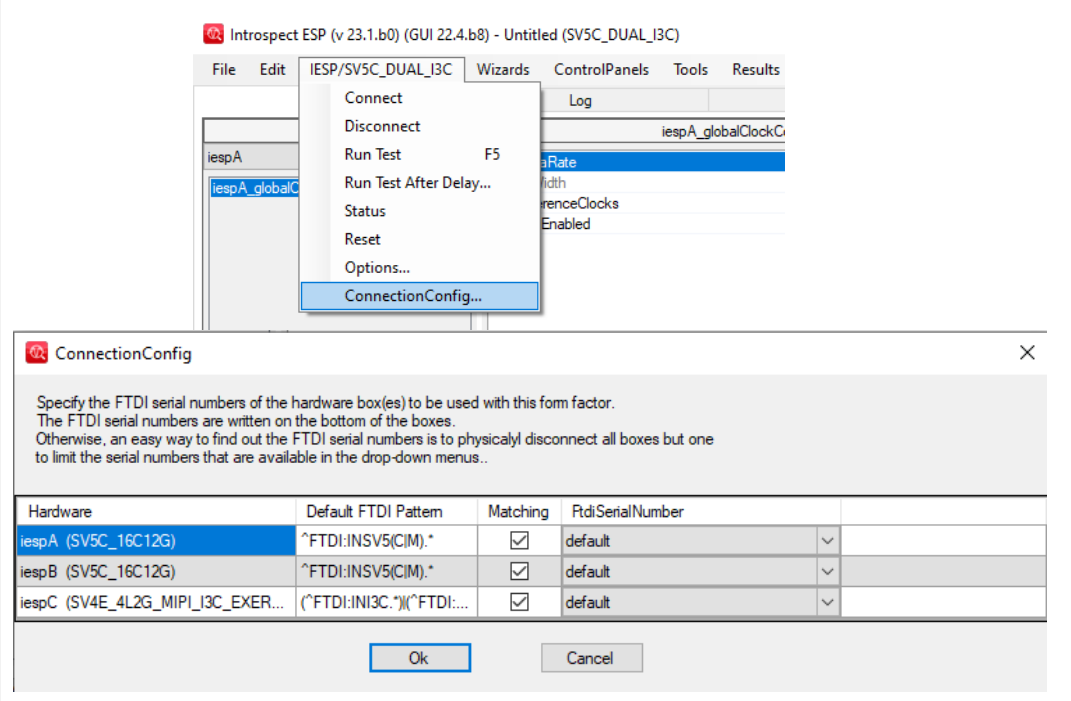


Figure 3: Setting up the connection configuration from the GUI

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.

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

INTROSPECT.CA