



# Introspect ESP Software Breakthrough: Combined Form Factors

Presenter:

David Tétreault-La Roche


Software Developer at Introspect

September 8, 2022



# Agenda

1. Combine any Introspect product
2. History & Motivation
3. Technical insight
4. Going forward

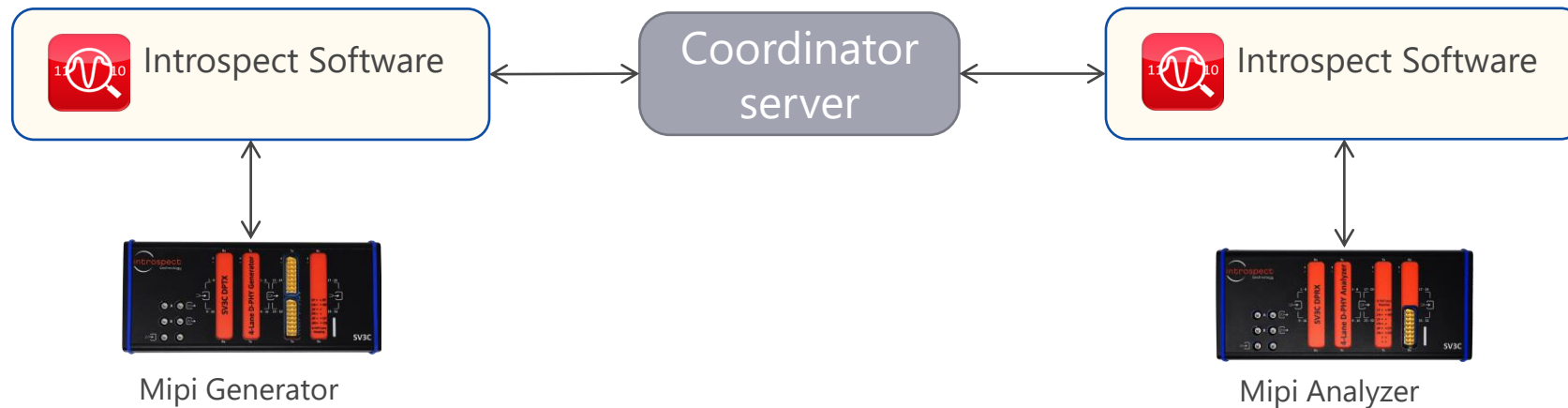


# Combining Introspect products

# Combining hardware: the old way

## MULTIPLE HARDWARE, MULTIPLE SOFTWARE

- Requires two tests folder to be saved
  - Tightly coupled code must be written in two places
- Need to use a coordinator server for full automation
- Flexible: combine any introspect product together



# Combining hardware: the old way

Introspect ESP (v 22.4.b9) (GUI 22.4.b8) - Untitled (SV3C\_4L6G\_MIPI\_DPH...

File Edit IESP/MIPI\_DPHY\_GENERATOR Wizards ControlPanels Tools Results Help

Params Log Results

Components

coordinator1	mipiGenerator1
csiImagePattern1	mipiProtocol
dphyParameters1	
mipiClockConfig1	

Add Remove Config

coordinator1 properties (class: Coordinator)

clientName	generator
clientSecret	
serverHostName	localhost
serverPort	12013

client Name  
Name used to identify this client to the server and to other clients. If this is None (empty), the component name will be used.

Test Procedure

```
1 iesp = getIespInstance()
2
3 coordinator1.setState("READY")
4 coordinator1.waitForClientState('analyzer', 'READY')
5
6 mipiGenerator1.setup()
7
8 coordinator1.setState("TRANSMITTING")
9 coordinator1.waitForClientState('analyzer', 'DONE')
10
11 iesp.disableMipiPatterns()
```

MIPI DPHY Csi2\_v1\_3 Run

CoordinatorServerWin

The "coordinator" server is used to help coordinate the execution of two or more processes (on the same or different computers). It is used with "coordinator" client components in the Test Procedure. Specify the IP address and port to be used by the server.

IP address: 127.0.0.1 port: 12013 Stop Server

Starting coordinator server at 127.0.0.1:12013  
1661547604.378520: Coordinator server running at 127.0.0.1:12013

Introspect ESP (v 22.4.b9) (GUI 22.4.b8) - Untitled (SV3C\_4L3G\_MIPI\_DPHY...

File Edit IESP/MIPI\_DPHY\_ANALYZER2 Wizards ControlPanels Tools Results Help

Params Log \* Results

Components

coordinator1	
csiDataCapture1	
laneList1	
mipiClockConfig1	
mipiProtocol	

Add Remove Config

coordinator1 properties (class: Coordinator)

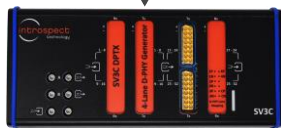
clientName	analyzer
clientSecret	
serverHostName	localhost
serverPort	12013

client Name  
Name used to identify this client to the server and to other clients. If this is None (empty), the component name will be used.

Test Procedure

```
1 coordinator1.setState("READY")
2 coordinator1.waitForClientState('generator', 'READY')
3 coordinator1.waitForClientState('generator', 'TRANSMITTING')
4
5 mipiClockConfig1.setup()
6 csiDataCapture1.run()
7
8 coordinator1.setState("DONE")
9
```

MIPI DPHY Csi2\_v1\_3 Run



Mipi Generator

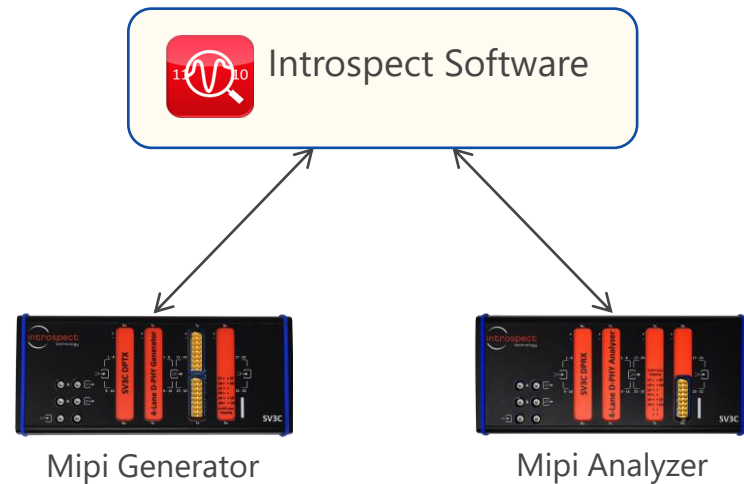


Mipi Analyzer

# Combining hardware: the new way

## MULTIPLE HARDWARE, ONE SOFTWARE

- One test folder for the whole setup – all the code is in one place
- No coordinator!
- Same flexibility: combine any Introspect product together



# Combining hardware: the new way

The screenshot shows the Introspect ESP (v 22.4.b9) GUI. The main window is titled "Introspect ESP (v 22.4.b9) - Untitled (MIPI\_DPHY\_DPHY\_BRIDGE)". It features a menu bar with "File", "Edit", "IESP/MIPI\_DPHY\_DPHY\_BRIDGE", "Wizards", "ControlPanels", "Tools", "Results", "Help", and "Debug". Below the menu bar are three tabs: "Params", "Log", and "Results".

The "Params" tab is active, showing a "Components" list on the left and a "csiDataCapture1 properties (class: CsiDataCapture)" table on the right. The "Components" list includes "iespRx", "csiDataCapture1", "iespRx\_mipiClockConfig1", "iespRx\_mipiProtocol", and "laneList1". The "csiDataCapture1 properties" table lists various parameters and their values:

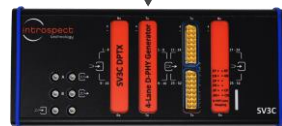
Property	Value
laneList	laneList 1
lanes	[1-4]
captureMode	burst
triggerCondition	frameStart
preTriggerDuration	1e+04
postTriggerType	numberOfFrameEnds
postTriggerDuration	1
ignoreHsData	False
csiRawFormatBayerCell	BGGR
csiEpd	False
triggerTimeout	20
captureTimeout	20

Below the table is a "laneList" section with the text "LaneList component". At the bottom of the "Params" tab are "Add", "Remove", and "Config" buttons.

The "Test Procedure" tab is also visible, containing a script:

```
1 iespTx, iespRx = getAllIespInstances()
2
3 mipiGenerator1.setup()
4
5 iespRx_mipiClockConfig1.setup()
6 csiDataCapture1.run()
7
8 iespTx.disableMipiPatterns()
```

At the bottom of the window, there is a "MIPI DPHY Csi2\_v1\_3" label and a "Run" button.



Mipi Generator



Mipi Analyzer

## Two tests & coordinator

```
iesp = getIespInstance()

coordinator1.setState("READY")
coordinator1.waitForClientState('analyzer', 'READY')

mipiGenerator1.setup()

coordinator1.setState("TRANSMITTING")
coordinator1.waitForClientState('analyzer', 'DONE')

iesp.disableMipiPatterns()
```

```
coordinator1.setState("READY")
coordinator1.waitForClientState('generator', 'READY')
coordinator1.waitForClientState('generator', 'TRANSMITTING')

mipiClockConfig1.setup()
csiDataCapture1.run()

coordinator1.setState("DONE")
```

## Combined form factor

```
iespTx, iespRx = getAllIespInstances()

mipiGenerator1.setup()

iespRx_mipiClockConfig1.setup()
csiDataCapture1.run()

iespTx.disableMipiPatterns()
```





# History & Motivation

# Why now?

## COORDINATOR: SIMPLE BUT UNSCALABLE

- Pro: Simple implementation: does not require any changes to architecture
- Con: Becomes very tedious to handle 3+ pieces of hardware (one test per hardware!).
  - Each test is effectively multithreaded (multiprocessed)
- Con: Writing multithreaded code is difficult, even for software engineers.
  - Many of Introspect's customers aren't software engineers themselves!

## COMBINED FORMFACTOR: MAJOR CHANGES REQUIRED

- Con: Years of "single hardware" assumptions must be undone
- Con: Much of the internal software architecture must be rethought
- Pro: much easier for the user

# 10 years ago...

## INTROSPECT'S PRODUCTS WERE ITERATIVE

SV1, SV2, and SV3 were interchangeable:

- All are SerDes products
- Similar feature set
- The later generation of products had better performance than the previous one

No real drive to combine products: they all fit in a similar niche.

*Need 32 channels? Don't use four SV1C, just use an SV3C instead!*

## SINGLE HARDWARE, SINGLE SOFTWARE

- Simplifies software engineering quite a bit
- Easier to prototype

# Today

## SERDES PLATFORMS

- SV1
- SV2
- SV2\_PAM4
- SV3
- SV5
- SV7 (upcoming!)

## PROTOCOL PRODUCTS

- MIPI
- Display port
- SLVS
- DDR
- I3C
- *and more...*

# Many products, many opportunities

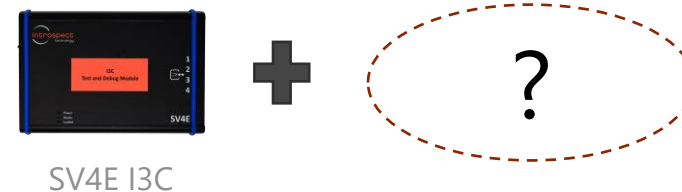
Combine MIPI TX & RX  
*Simplify your test bench*



64-channel SV5  
*Used in DDR test suite*



I3C + anything!



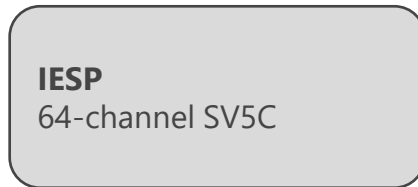


Technical insight

# Combined form factors: two types

## EXTENDED IESP

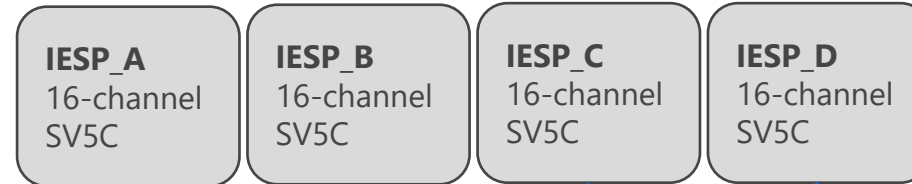
- Multiple boxes used as if one



4x 16-channel SV5C

## MULTIPLE IESP

- Similar to having multiple tests open, but a single unified Python test procedure
- Replaces coordinator server

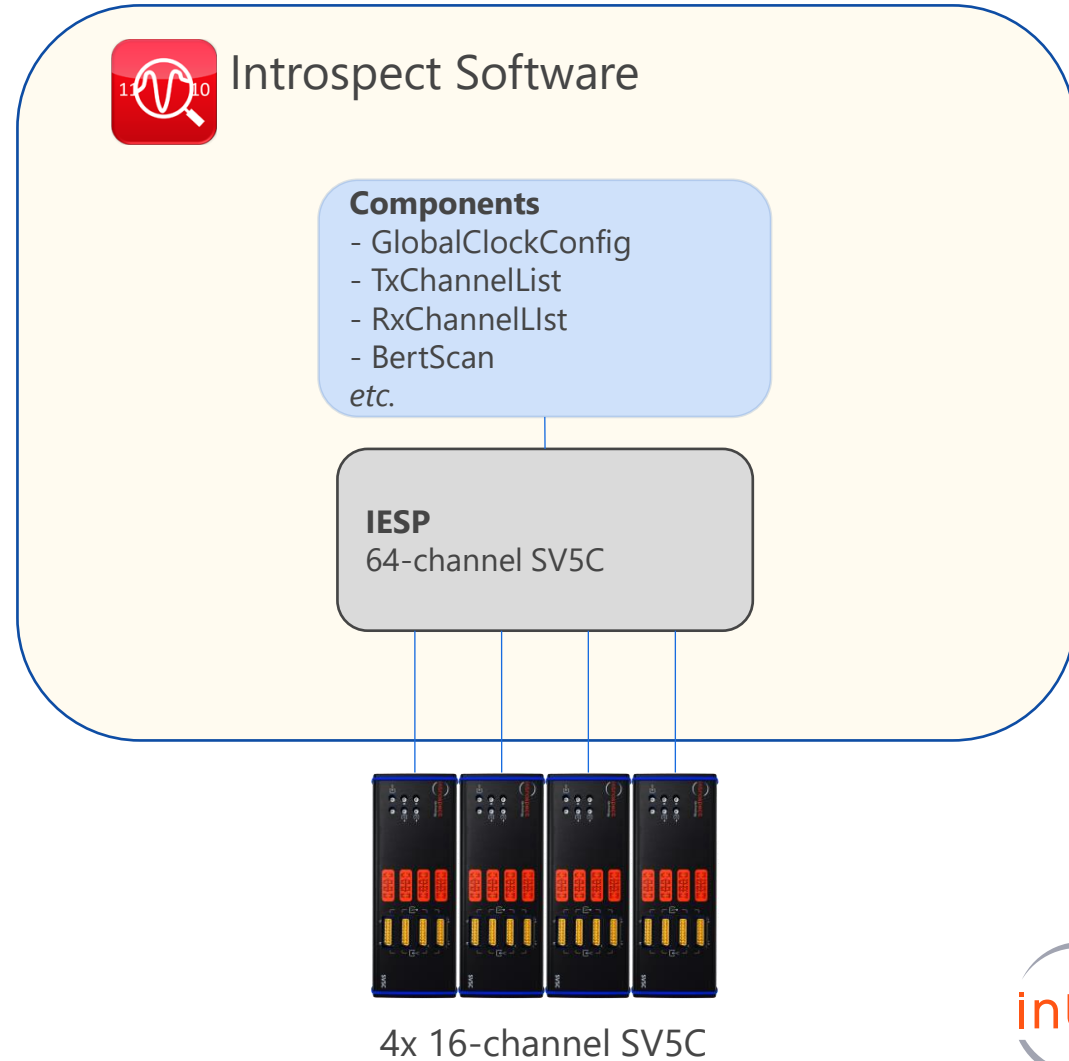


4x 16-channel SV5C

# New architecture: extended IESP

## ADD BOXES FOR MORE CHANNELS

- Same as regular usage, except more channels!
- Only possible for SERDES form factors
  - Protocol form factors (Mipi, Display port, I3C, etc.) *cannot* be extended.
- Note: alignment between channels not guaranteed
- No limit to number of hardware pieces used.

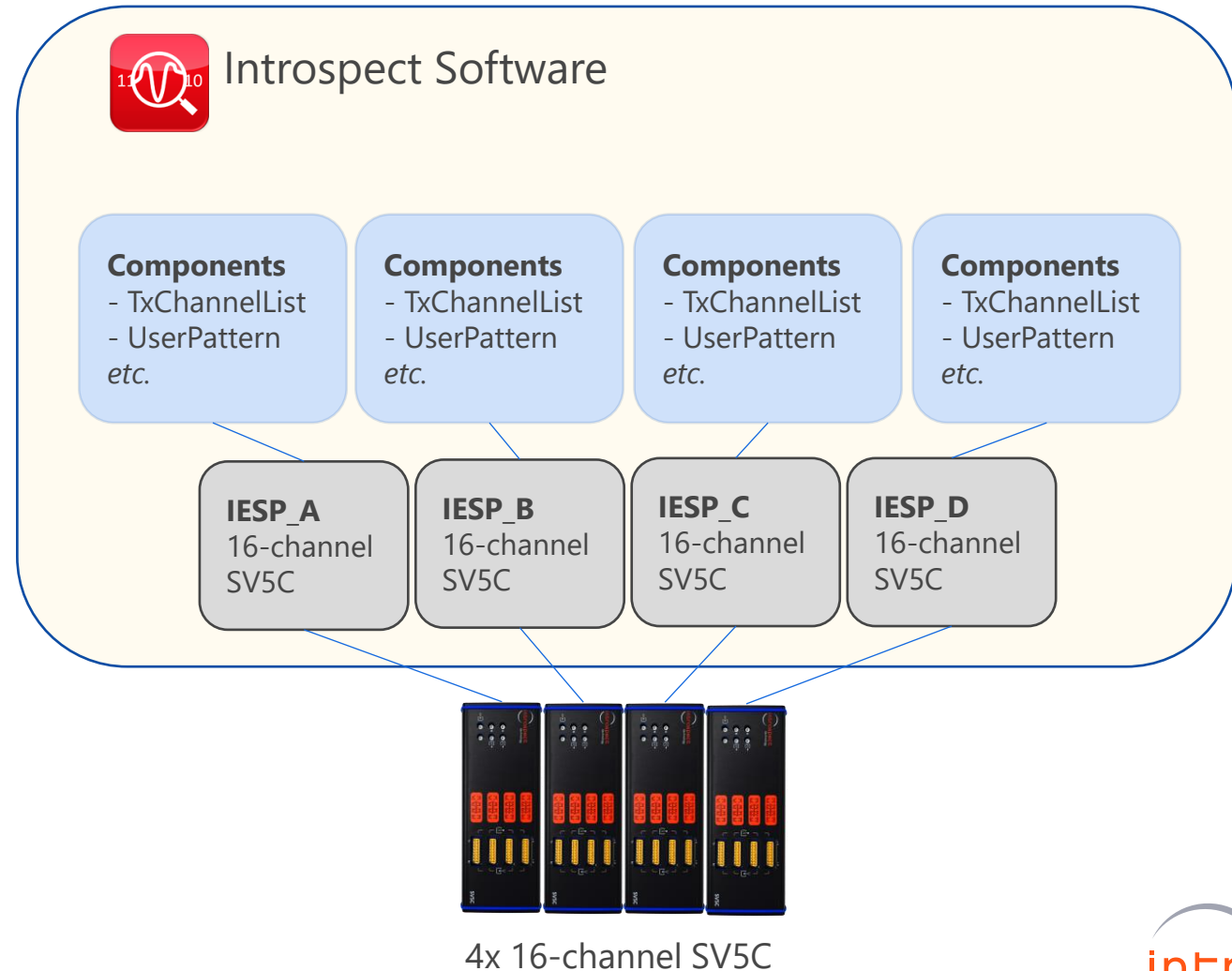




# New architecture: multiple IESP

## COMPONENTS FOR EACH HARDWARE

- One *set of components* per hardware
  - Components of one set can only be used together
  - Can't use a UserPattern for lespA with the TxChannelList for lespB
- The test procedure code can refer to both sets of components seamlessly
- No limit to number or type hardware pieces used.



# New architecture: multiple IESP

## COMPONENTS FOR EACH HARDWARE

- One *set of components* per hardware
  - Components of one set can only be used together
  - Can't use a UserPattern for IespA with the TxChannelList for IespB
- The test procedure code can refer to both sets of components seamlessly
- No limit to number of hardware pieces used.

The image displays two screenshots of the Introspect ESP GUI (v 22.4.b9) for a MIPI\_DPHY\_DPHY\_BRIDGE. The top screenshot shows the configuration for the 'iespTx' component. The 'Components' list includes: csImagePattern1, dphyParameters1, IespTx\_mipiClockConfig1, IespTx\_mipiProtocol, jitterInjection1, and mipiGenerator1. The 'csImagePattern1 properties' are shown, including timeUnits (nanosecond), imageFiles (IntrospectLogo.png), imageFormat (CSI\_RGB888), gaussianBlurRadius (0), lineTimeMode (lineTimeTotal), horizLineTime (30000.0), frameBlankingMode (frameRate), frameRate, and lineNumber. The 'Test Procedure' code is:

```
1 IespTx, IespRx = GetAllIespInstances()
2
3 mipiGenerator1.setup()
4
5 IespRx_mipiClockConfig1.setup()
6 csiDataCapture1.run()
7
8 IespTx.disableMipiPatterns()
```

The bottom screenshot shows the configuration for the 'iespRx' component. The 'Components' list includes: csiDataCapture1, IespRx\_mipiClockConfig1, IespRx\_mipiProtocol, and laneList1. The 'csiDataCapture1 properties' are shown, including captureMode (burst), triggerCondition (frameStart), preTriggerDuration (1e+04), postTriggerType (numberOfFrameEnds), postTriggerDuration (1), ignoreHsData (False), csiRawFormatBayerCell (BGGR), csiEpd (False), and triggerTimeout (20). The 'Test Procedure' code is:

```
1 IespTx, IespRx = GetAllIespInstances()
2
3 mipiGenerator1.setup()
4
5 IespRx_mipiClockConfig1.setup()
6 csiDataCapture1.run()
7
8 IespTx.disableMipiPatterns()
```

# How-to?

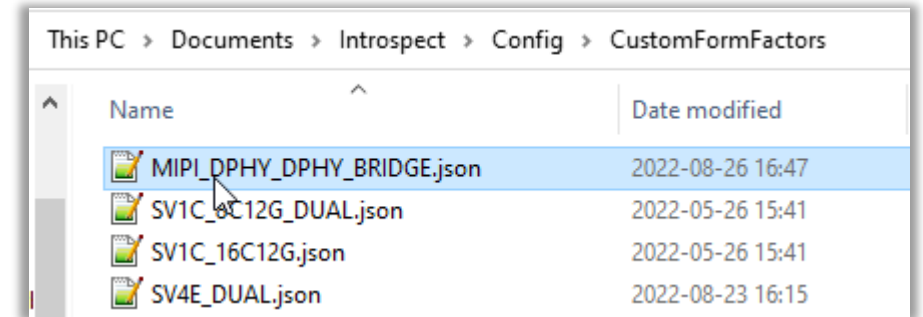
## MAKE YOUR OWN COMBINED FORM FACTOR

- Can only be done by creating a .json describing the necessary products
- Users need to know exactly what they want (can't explore options)
- **Very cumbersome workflow:** this will be heavily improved in the future

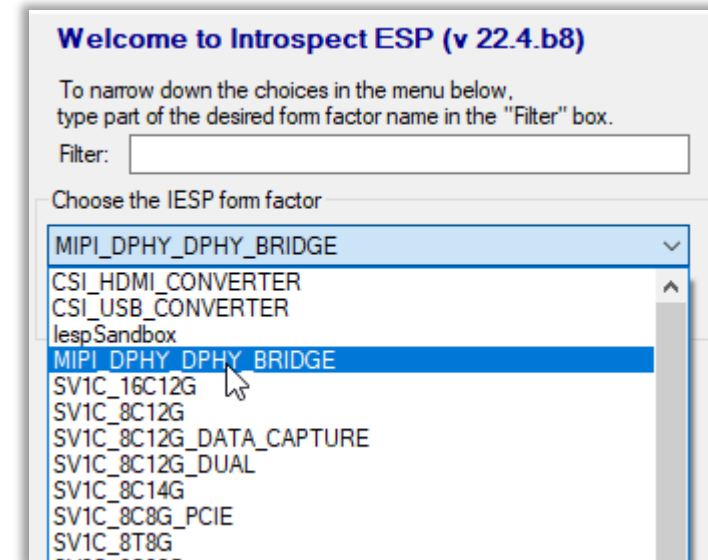
1 - Create combined form factor file

```
MIPI_DPHY_DPHY_BRIDGE.json x
1 {
2     "name": "MIPI_DPHY_DPHY_BRIDGE",
3     "iespClassMap": {
4         "iespTx": "SV3C_4L6G_MIPI_DPHY_GENERATOR",
5         "iespRx": "SV3C_4L3G_MIPI_DPHY_ANALYZER2"
6     }
7 }
```

2 - Place it in this folder



3 - Select it in the GUI





Going forward

# Whats next?

## WHEN WILL IT BE AVAILABLE?

- Its already in place! The most recent version(s) of the IntrospectESP software already support combined form factors.

## FUTURE IMPROVEMENTS

- Updates to how the combined form factors are created since current approach is less than ideal
- Add a mechanism to save the combined form factor as part of a standard test folder.



To Learn More Visit:

**INTROSPECT.CA**