FPD-Link™ III: Use of Analog Launch Pad (ALP) GUI to configure the EVM

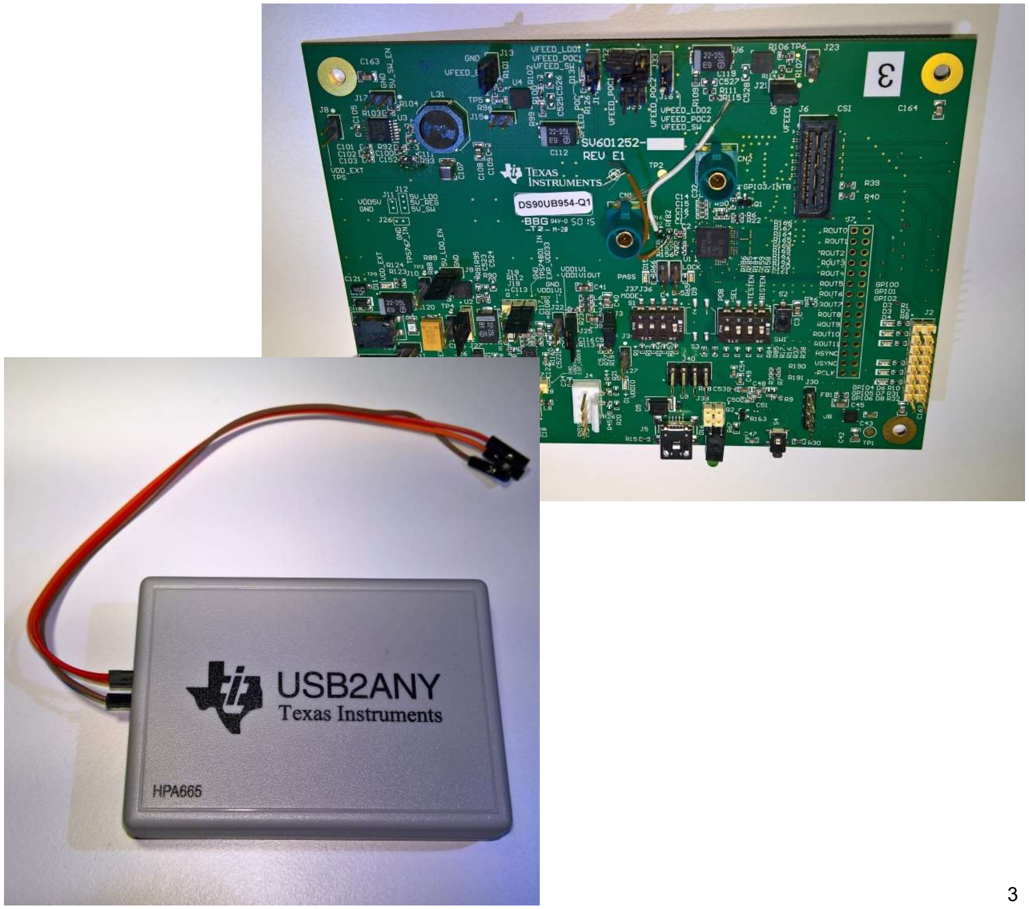
FPD-Link Product Line
Texas Instruments Inc.

What you will get from this module

- Learn how to configure and use USB2ANY
- Learn how to configure and use ALP
- Learn how to use the “Scripting Tab” in ALP
- An example of a successful communication with an IC via ALP.

Required components for this demo

- Hardware:
  - EVM
  - USB2ANY
  - Cable assemblies
- Software
  - ALP
  - USB2ANY SDK



What is USB2ANY

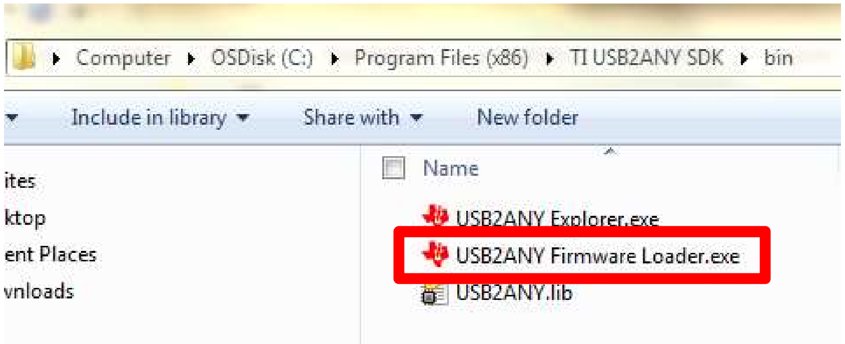
- The USB2ANY Controller Board is a small dongle that, via a USB connection to a computer, enables access to devices using the following interfaces:

○ I2C, SPI, ADC, DAC, PWM, UART, GPIO, Shared memory, FEC, Interrupts, EasyScale™

- The USB2ANY board can source regulated 3.3V and 5.0V DC power to the target device, up to a maximum of 300mA (combined 3.3V and 5.0V).

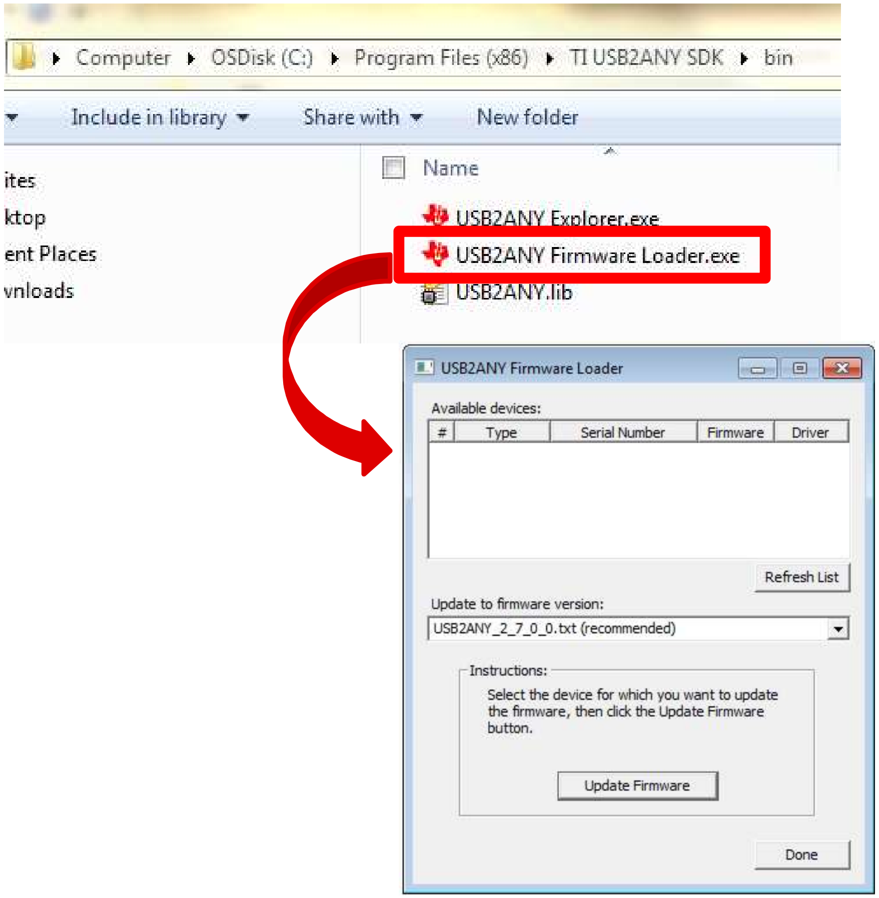
How to configure and use the USB2ANY

- Install the USB2ANY SDK (Contact your local FAE to get it)
- Connect the USB2ANY to the USB Port on your computer
- Go to the installation folder (C:\Program Files (x86)\TI USB2ANY SDK\bin)
- Run the "USB2ANY Firmware Loader.exe" and flash the newest firmware



How to configure and use the USB2ANY

- Install the USB2ANY SDK (Contact your local FAE to get it)
- Connect the USB2ANY to the USB Port on your computer
- Go to the installation folder (C:\Program Files (x86)\TI USB2ANY SDK\bin)
- Run the "USB2ANY Firmware Loader.exe" and flash the newest firmware



Update to firmware version: USB2ANY_2_7_0_0.txt (recommended)

Instructions: Select the device for which you want to update the firmware, then click the Update Firmware button.

How to configure and use the ALP GUI

- Download and Install the ALP Software
  – http://www.ti.com/tool/alp
- To start the MainGUI.exe, execute the "Analog Launch PAD" shortcut from the start menu.
- The default start menu location is under All Programs > Texas Instruments > Analog LaunchPAD vx.x.x > Analog LaunchPAD.

Analog LaunchPAD

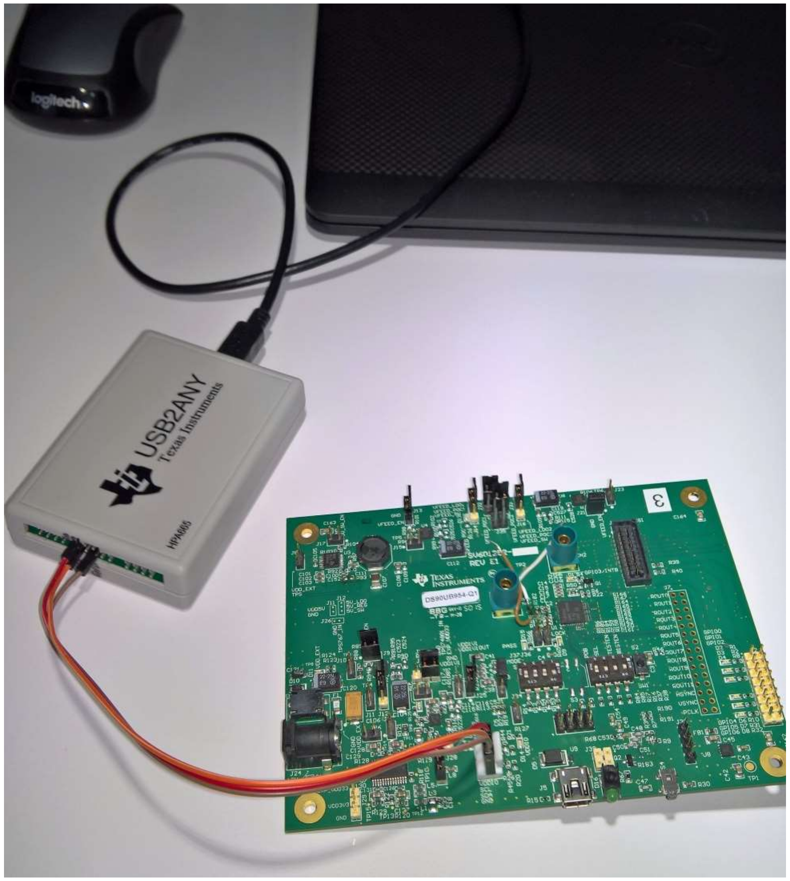
Texas Instruments

Copyright © Texas Instruments Incorporated. All rights reserved

How to use the ALP GUI: Connecting hardware

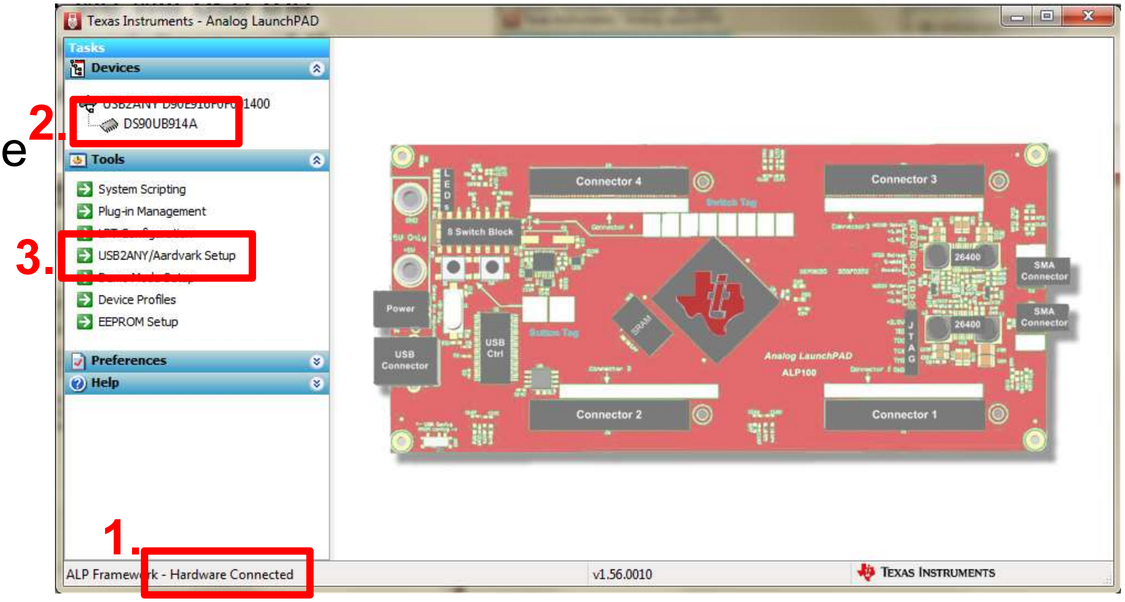
1. Connect the USB2ANY to the EVM through SCL, SDA and GND pins.
2. Power on the EVM.
3. Go to ALP GUI and follow the following steps.

- Note: always run ALP as administrator.



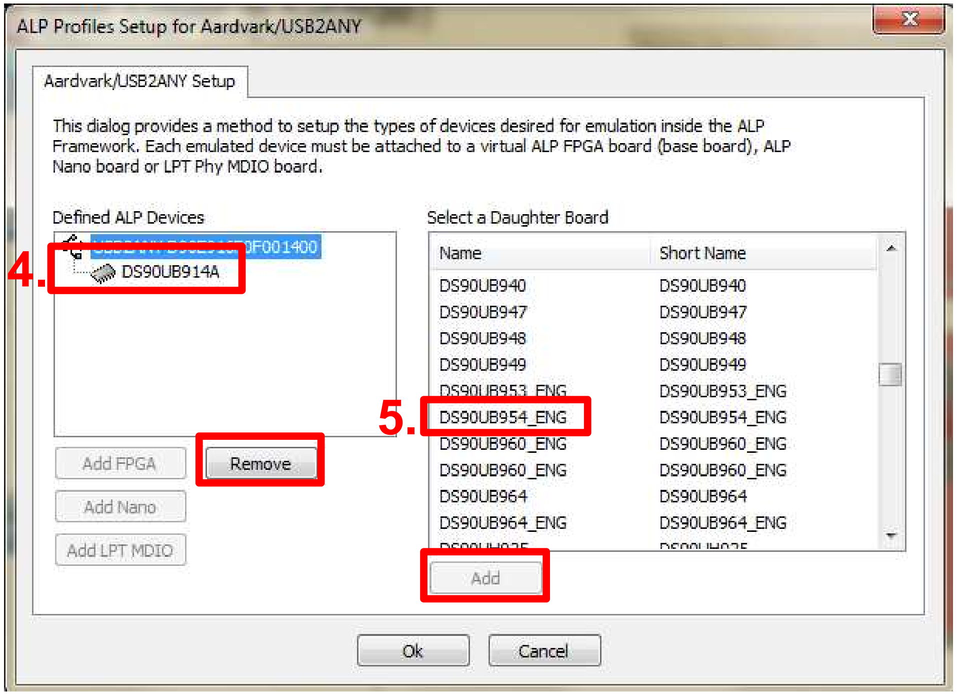
How to use the ALP GUI: Connecting hardware

1. Make sure the hardware is connected .
2. Make sure the correct device profile is loaded (As an example we will use the DS90UB954 deserializer)
3. To change the profile, go to USB2ANY/Aardvark Setup.



How to use the ALP GUI: Choose the right profile

4. Chose the existing device and click Remove.
5. Under “Select a Daughter Board" search for the required device profile, in this case DS90UB954, and click “Add” and then “Ok”.
6. If the required profile is not available, please contact your local FAE.



ALP Profiles Setup for Aardvark/USB2ANY

Aardvark/USB2ANY Setup

This dialog provides a method to setup the types of devices desired for emulation inside the ALP Framework. Each emulated device must be attached to a virtual ALP FPGA board (base board), ALP Nano board or LPT Phy MDIO board.

Defined ALP Devices

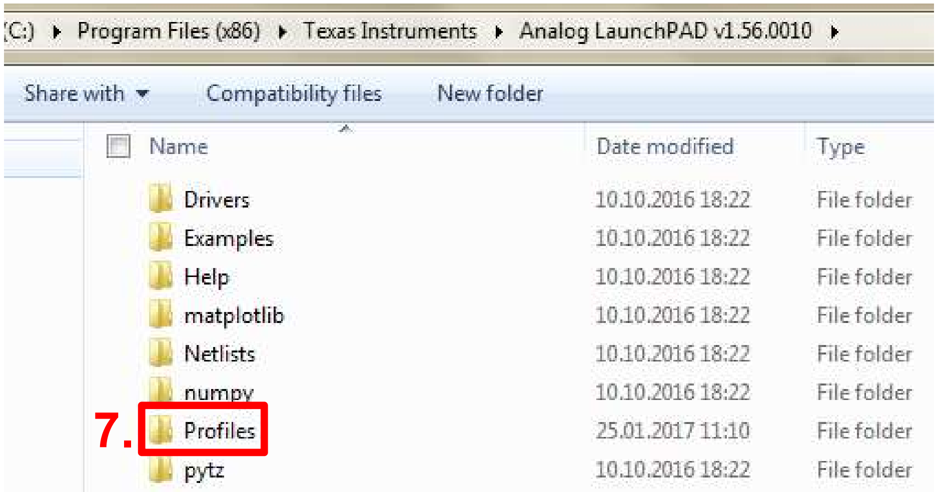
- [illegible]0F001400
  - DS90UB914A

Select a Daughter Board

Name	Short Name
DS90UB940	DS90UB940
DS90UB947	DS90UB947
DS90UB948	DS90UB948
DS90UB949	DS90UB949
DS90UB953_ENG	DS90UB953_ENG
DS90UB954_ENG	DS90UB954_ENG
DS90UB960_ENG	DS90UB960_ENG
DS90UB960_ENG	DS90UB960_ENG
DS90UB964	DS90UB964
DS90UB964_ENG	DS90UB964_ENG

How to use the ALP GUI: adding a profile

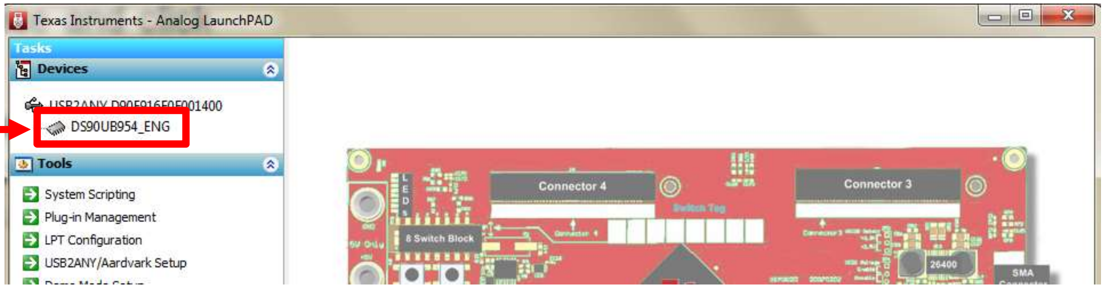
7. Once you have the required profile, unzip it and save it in the installation folder under C:\....\ALP-Installationfolder\Profiles.
8. Restart "Analog Launch PAD" and repeat the steps from 3 – 5.


(C:) ▸ Program Files (x86) ▸ Texas Instruments ▸ Analog LaunchPAD v1.56.0010 ▸
Share with ▾
Compatibility files
New folder
7.

Name	Date modified	Type
Drivers	10.10.2016 18:22	File folder
Examples	10.10.2016 18:22	File folder
Help	10.10.2016 18:22	File folder
matplotlib	10.10.2016 18:22	File folder
Netlists	10.10.2016 18:22	File folder
numpy	10.10.2016 18:22	File folder
Profiles	25.01.2017 11:10	File folder
pytz	10.10.2016 18:22	File folder

How to use the ALP GUI: ALP overview

- Now you should have the right device profile loaded. Once you click on it, a new window with tabs will appear.



Texas Instruments - Analog LaunchPAD

Tasks

Devices

USB2ANY D90E916F0F001400

DS90UB954_ENG

Tools

System Scripting

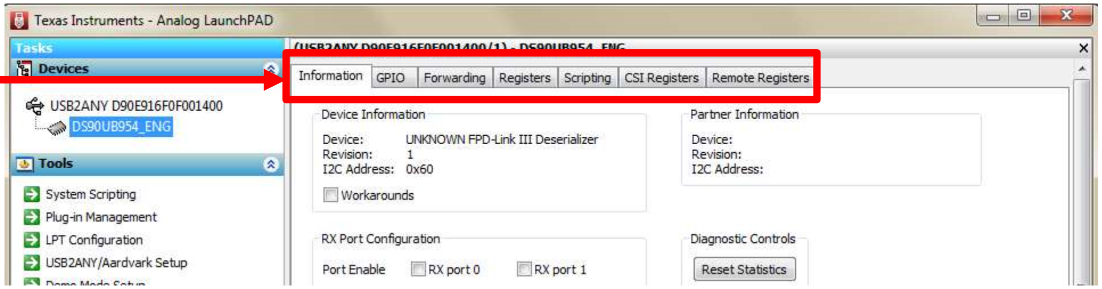
Plug-in Management

LPT Configuration

USB2ANY/Aardvark Setup

- The number of tabs will vary between different devices and profiles.

- For the DS90UB954 there are 7 tabs: Information, GPIO, Forwarding, Scripting, CSI Registers, Remote Registers.



Texas Instruments - Analog LaunchPAD

Tasks

Devices

USB2ANY D90E916F0F001400

DS90UB954_ENG

Tools

System Scripting

Plug-in Management

LPT Configuration

USB2ANY/Aardvark Setup

(USB2ANY D90E916F0F001400/(1) - DS90UB954_ENG

Information | GPIO | Forwarding | Registers | Scripting | CSI Registers | Remote Registers

Device Information

Device: UNKNOWN FPD-Link III Deserializer
Revision: 1
I2C Address: 0x60

□ Workarounds

Partner Information

Device:
Revision:
I2C Address:

RX Port Configuration

Port Enable □ RX port 0 □ RX port 1

Diagnostic Controls

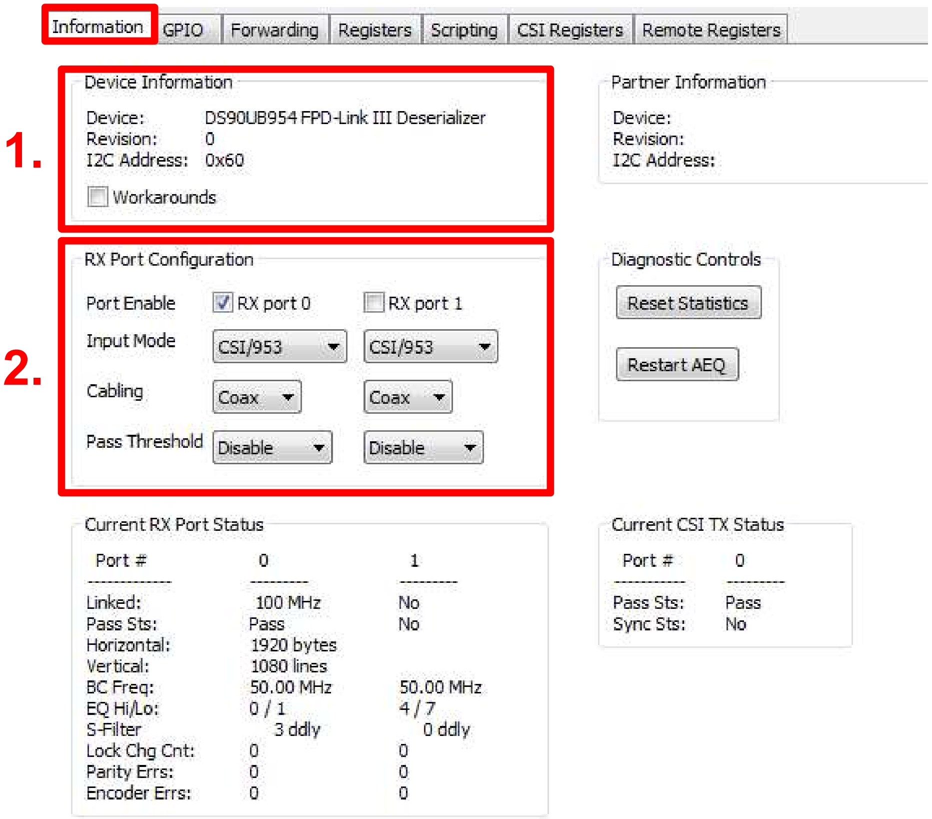
Reset Statistics

How to use the ALP GUI: Information Tab

- Under the "Information" tab multiple boxes are shown containing different information about the connected Device.

1. Device Information:
   - Device name
   - Revision
   - I^{2}C Address

2. RX Port Configuration:
   - Port Enable/Disable
   - Used Cable
   - Pass Threshold



Device Information

Device: DS90UB954 FPD-Link III Deserializer
Revision: 0
I2C Address: 0x60

□ Workarounds

Partner Information

Device:
Revision:
I2C Address:

RX Port Configuration

Port Enable	☑ RX port 0	□ RX port 1
Input Mode	CSI/953	CSI/953
Cabling	Coax	Coax
Pass Threshold	Disable	Disable

Diagnostic Controls

Current RX Port Status

```
 Port #             0              1
-------------   ---------      ---------
Linked:          100 MHz        No
Pass Sts:       Pass            No
Horizontal:     1920 bytes
Vertical:       1080 lines
BC Freq:        50.00 MHz      50.00 MHz
EQ Hi/Lo:       0 / 1          4 / 7
S-Filter            3 ddly         0 ddly
Lock Chg Cnt:   0              0
Parity Errs:    0              0
Encoder Errs:   0              0
```

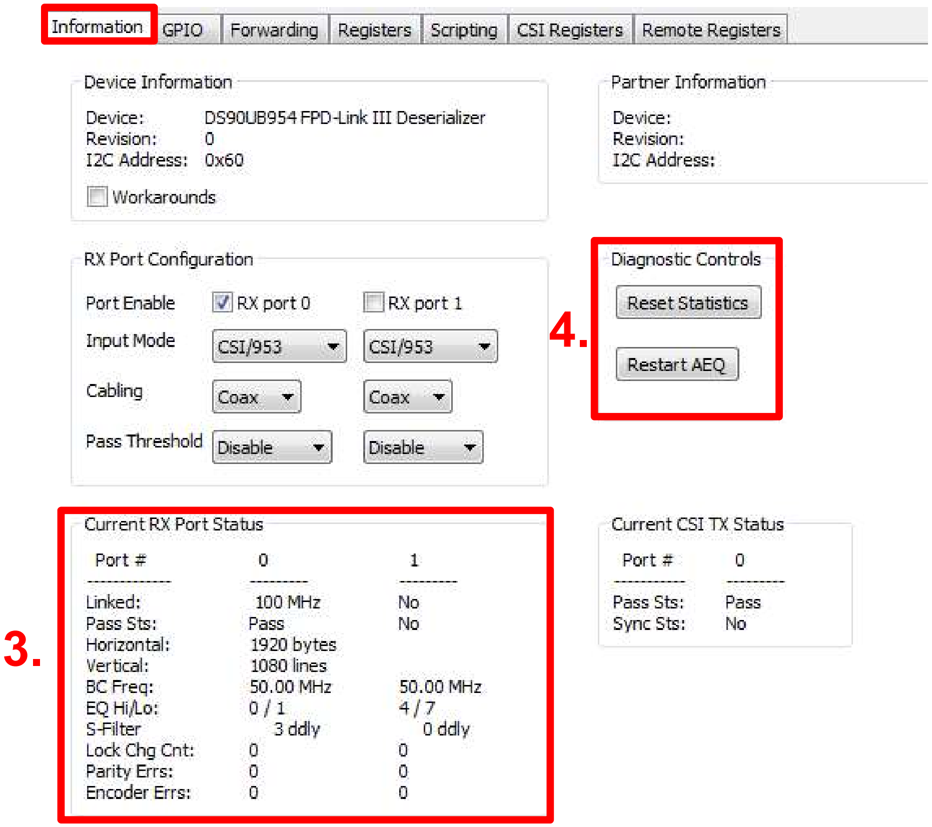
Current CSI TX Status

```
 Port #        0
----------   ---------
Pass Sts:    Pass
Sync Sts:    No
```

How to use the ALP GUI: Information Tab

3. RX Port Status:
   - Lock and Pass
   - Horizontal and Vertical resolution.
   - BC Frequency
   - AEQ & S-Filter
   - Errors

4. Diagnostic Controls



Device Information

Device: DS90UB954 FPD-Link III Deserializer
Revision: 0
I2C Address: 0x60

□ Workarounds

Partner Information

Device:
Revision:
I2C Address:

RX Port Configuration

Port Enable	☑ RX port 0	□ RX port 1
Input Mode	CSI/953	CSI/953
Cabling	Coax	Coax
Pass Threshold	Disable	Disable

Diagnostic Controls

Current RX Port Status

```
Port #           0            1
-------------    ---------    ---------
Linked:          100 MHz      No
Pass Sts:        Pass         No
Horizontal:      1920 bytes
Vertical:        1080 lines
BC Freq:         50.00 MHz    50.00 MHz
EQ Hi/Lo:        0 / 1        4 / 7
S-Filter            3 ddly       0 ddly
Lock Chg Cnt:    0            0
Parity Errs:     0            0
Encoder Errs:    0            0
```

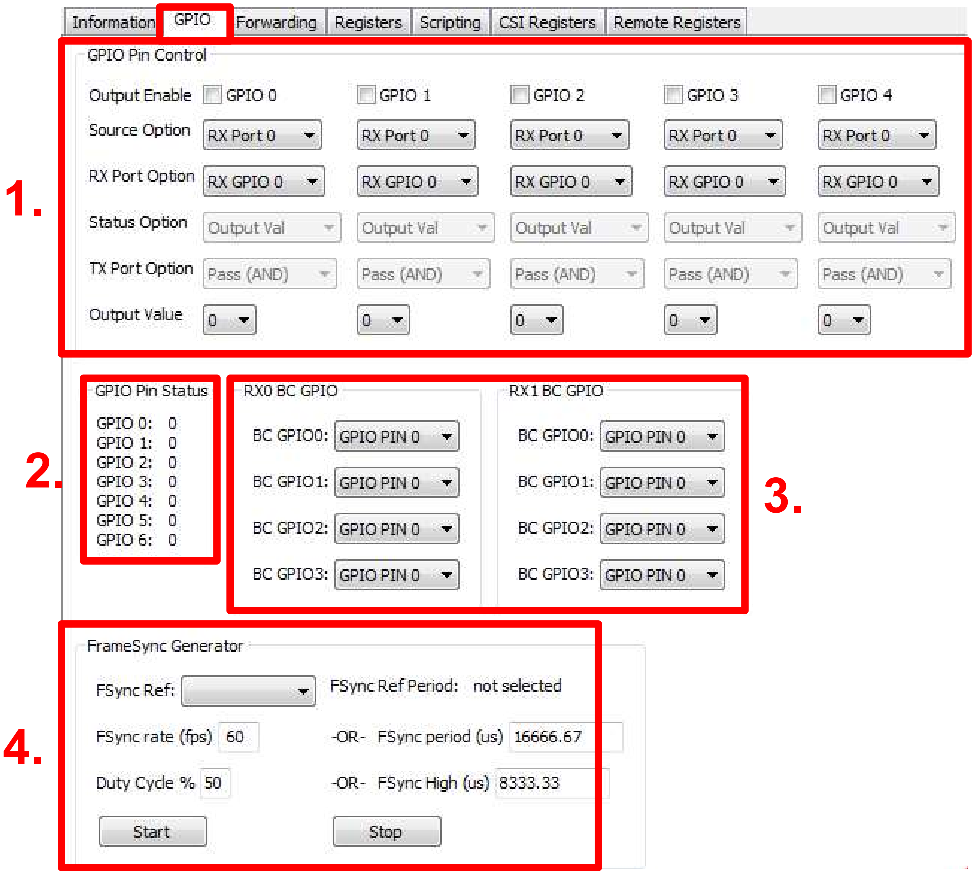
Current CSI TX Status

```
Port #        0
-----------   ---------
Pass Sts:     Pass
Sync Sts:     No
```

How to use the ALP GUI: GPIO Tab

- Under the "GPIO" tab, multiple boxes are shown containing different settings of the Device GPIOs:

1. GPIO Pin Control: Contains panels and drop downs to enable, disable and chose different functions of GPIOs without the need to write specific values into registers
2. GPIO Pin Status
3. BC GPIO
4. FrameSync Generator



GPIO Pin Control

Output Enable	□ GPIO 0	□ GPIO 1	□ GPIO 2	□ GPIO 3	□ GPIO 4
Source Option	RX Port 0	RX Port 0	RX Port 0	RX Port 0	RX Port 0
RX Port Option	RX GPIO 0	RX GPIO 0	RX GPIO 0	RX GPIO 0	RX GPIO 0
Status Option	Output Val	Output Val	Output Val	Output Val	Output Val
TX Port Option	Pass (AND)	Pass (AND)	Pass (AND)	Pass (AND)	Pass (AND)
Output Value	0	0	0	0	0

GPIO Pin Status

```
GPIO 0:  0
GPIO 1:  0
GPIO 2:  0
GPIO 3:  0
GPIO 4:  0
GPIO 5:  0
GPIO 6:  0
```

RX0 BC GPIO		RX1 BC GPIO	
BC GPIO0:	GPIO PIN 0	BC GPIO0:	GPIO PIN 0
BC GPIO1:	GPIO PIN 0	BC GPIO1:	GPIO PIN 0
BC GPIO2:	GPIO PIN 0	BC GPIO2:	GPIO PIN 0
BC GPIO3:	GPIO PIN 0	BC GPIO3:	GPIO PIN 0

FrameSync Generator

FSync Ref: FSync Ref Period: not selected

FSync rate (fps) 60 -OR- FSync period (us) 16666.67

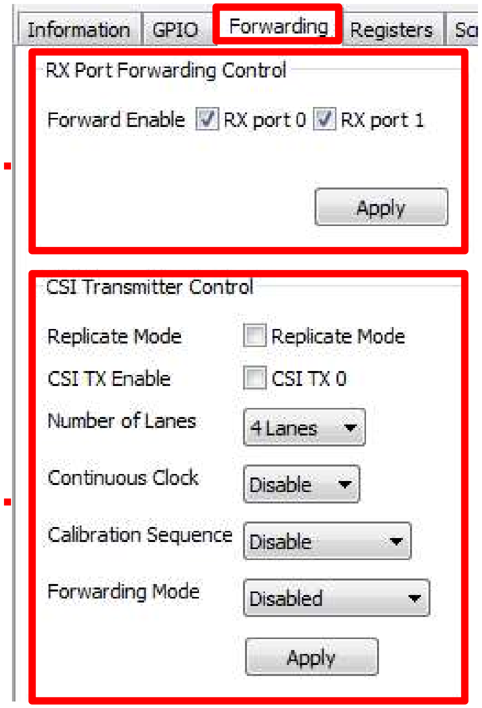
Duty Cycle % 50 -OR- FSync High (us) 8333.33

How to use the ALP GUI: Forwarding Tab

- Under the "Forwarding" tab, two boxes are shown containing Port forwarding and CSI Control.

1. RX Port Forwarding Control
   For activating the Port Forwarding from RX Port 0 and / or RX Port 1 without the need to write specific values into registers

2. CSI Transmitter Control
   For configuring the CSI Transmitter without the need to write specific values into registers



1. RX Port Forwarding Control

Forward Enable ☑ RX port 0 ☑ RX port 1

2. CSI Transmitter Control

Replicate Mode □ Replicate Mode

CSI TX Enable □ CSI TX 0

Number of Lanes: 4 Lanes

Continuous Clock: Disable

Calibration Sequence: Disable

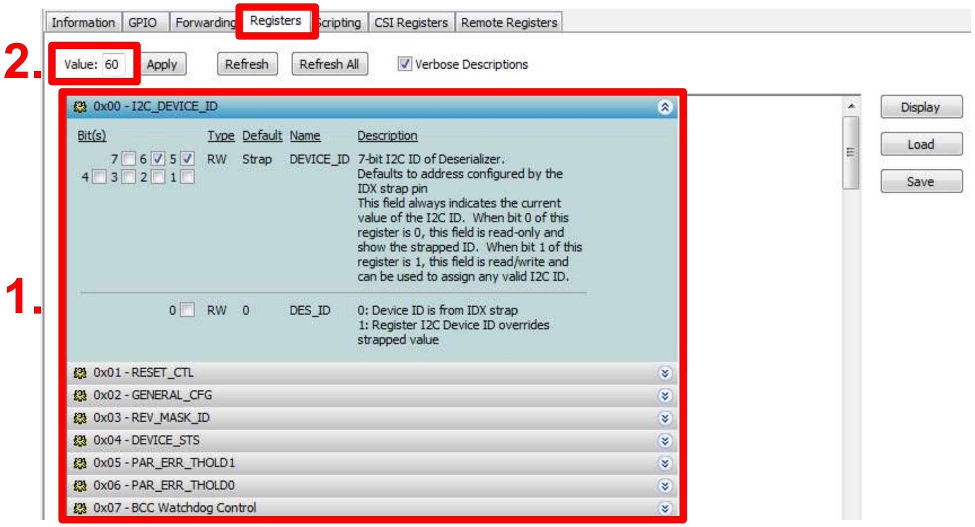
Forwarding Mode: Disabled

How to use the ALP GUI: Registers Tab

- Under the "Registers" tab, the device’s registers can be found and modified:

1. Click on the arrow on the right side of any register to expand and show the exact bits and their description from the datasheet. To collapse click on the arrow again.

2. Value:
   Once a register is selected in the above step, the value of this register can be read or changed here.



Value: 60

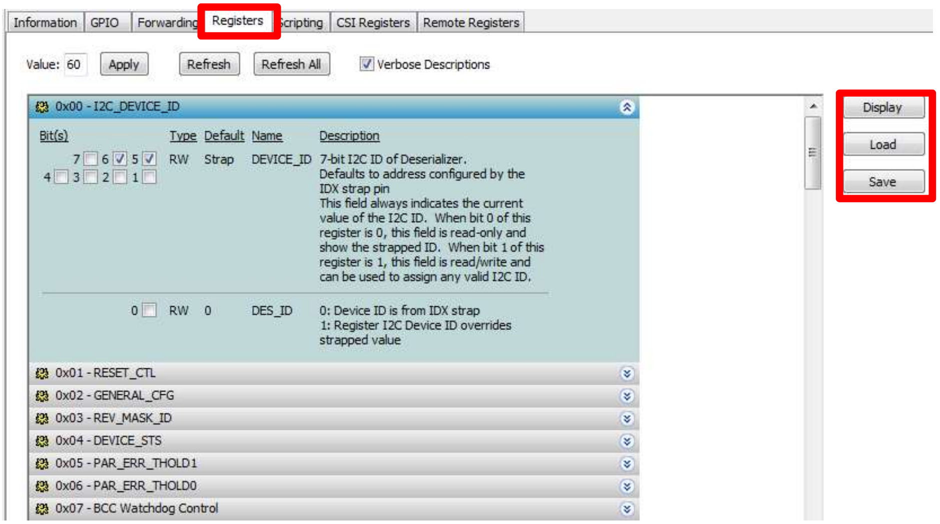
0x00 - I2C_DEVICE_ID

Bit(s)	Type	Default	Name	Description
7 □ 6 ☑ 5 ☑ 4 □ 3 □ 2 □ 1 □	RW	Strap	DEVICE_ID	7-bit I2C ID of Deserializer. Defaults to address configured by the IDX strap pin This field always indicates the current value of the I2C ID. When bit 0 of this register is 0, this field is read-only and show the strapped ID. When bit 1 of this register is 1, this field is read/write and can be used to assign any valid I2C ID.
0 □	RW	0	DES_ID	0: Device ID is from IDX strap 1: Register I2C Device ID overrides strapped value

0x01 - RESET_CTL
0x02 - GENERAL_CFG
0x03 - REV_MASK_ID
0x04 - DEVICE_STS
0x05 - PAR_ERR_THOLD1
0x06 - PAR_ERR_THOLD0
0x07 - BCC Watchdog Control

How to use the ALP GUI: Registers Tab

- Display - shows a dump of all registers.
- Load - loads a previously saved dump.
- Save - saves a register dump.



Value: 60

☑ Verbose Descriptions

0x00 - I2C_DEVICE_ID

Bit(s)	Type	Default	Name	Description
7 □ 6 ☑ 5 ☑ 4 □ 3 □ 2 □ 1 □	RW	Strap	DEVICE_ID	7-bit I2C ID of Deserializer. Defaults to address configured by the IDX strap pin This field always indicates the current value of the I2C ID. When bit 0 of this register is 0, this field is read-only and show the strapped ID. When bit 1 of this register is 1, this field is read/write and can be used to assign any valid I2C ID.
0 □	RW	0	DES_ID	0: Device ID is from IDX strap 1: Register I2C Device ID overrides strapped value

0x01 - RESET_CTL

0x02 - GENERAL_CFG

0x03 - REV_MASK_ID

0x04 - DEVICE_STS

0x05 - PAR_ERR_THOLD1

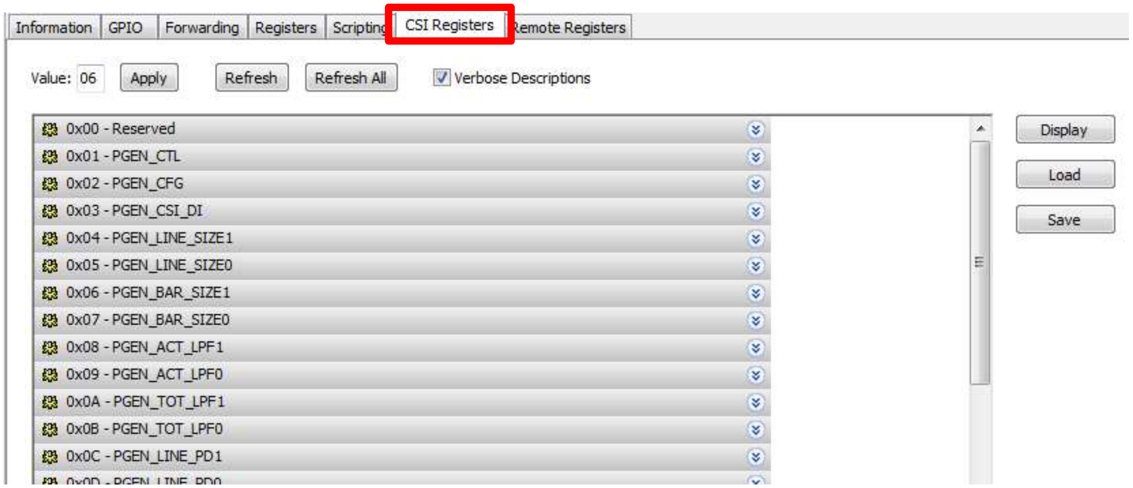
0x06 - PAR_ERR_THOLD0

0x07 - BCC Watchdog Control

How to use the ALP GUI: CSI-Registers Tab

- Under the "CSI Registers" tab are the specific CSI registers and pattern generator register to be found.

- Everything else is the same as in the "Registers" tab



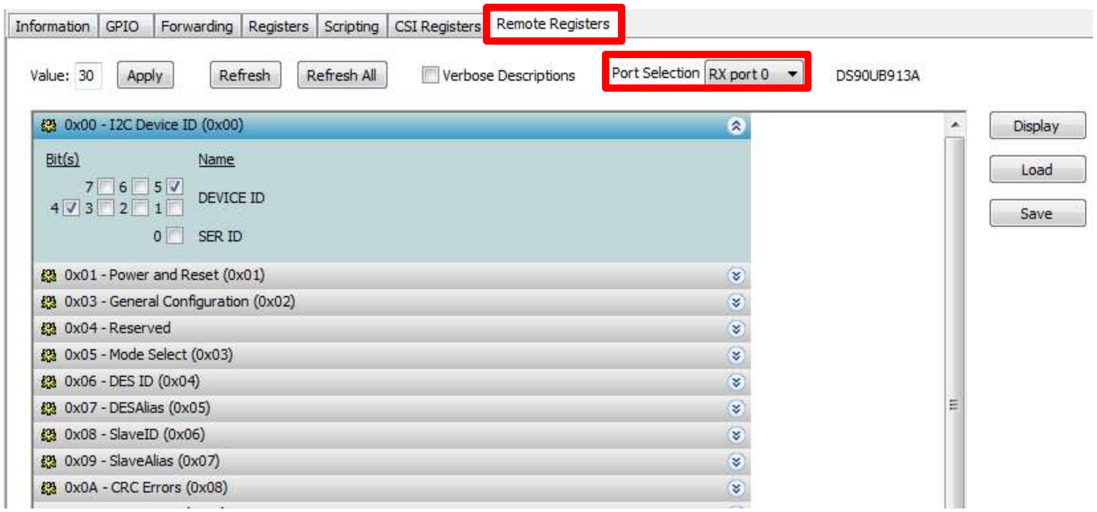
Value: 06

☑ Verbose Descriptions

0x00 - Reserved
0x01 - PGEN_CTL
0x02 - PGEN_CFG
0x03 - PGEN_CSI_DI
0x04 - PGEN_LINE_SIZE1
0x05 - PGEN_LINE_SIZE0
0x06 - PGEN_BAR_SIZE1
0x07 - PGEN_BAR_SIZE0
0x08 - PGEN_ACT_LPF1
0x09 - PGEN_ACT_LPF0
0x0A - PGEN_TOT_LPF1
0x0B - PGEN_TOT_LPF0
0x0C - PGEN_LINE_PD1
0x0D - PGEN_LINE_PD0

How to use the ALP GUI: Remote-Registers Tab

- Under the "Remote Registers" tab are the registers of the paired device (in this case DS90UB953) to be found.
- If two serializers are paired with this device, the required serializer's registers can be chosen by selecting the right RX port under "Port Selection".
- Everything else is the same as in the "Registers" tab



Value: 30

□ Verbose Descriptions

Port Selection: RX port 0

DS90UB913A

0x00 - I2C Device ID (0x00)

Bit(s)	Name
7 □ 6 □ 5 ☑ 4 ☑ 3 □ 2 □ 1 □	DEVICE ID
0 □	SER ID

0x01 - Power and Reset (0x01)

0x03 - General Configuration (0x02)

0x04 - Reserved

0x05 - Mode Select (0x03)

0x06 - DES ID (0x04)

0x07 - DESAlias (0x05)

0x08 - SlaveID (0x06)

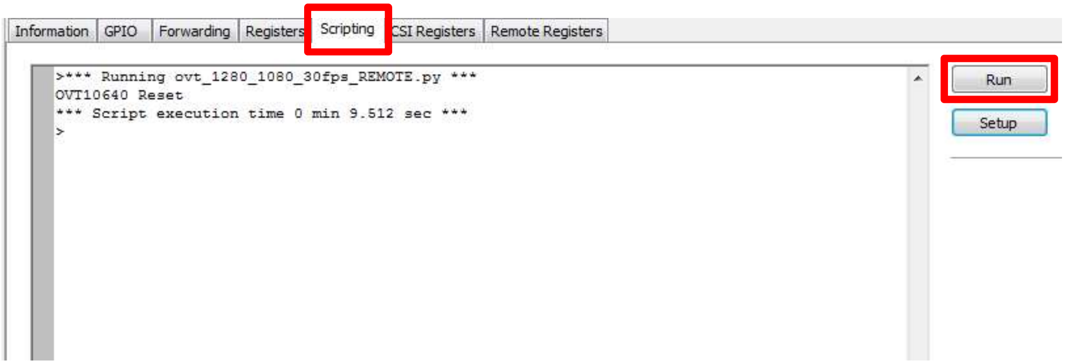
0x09 - SlaveAlias (0x07)

0x0A - CRC Errors (0x08)

Using the Scripting tab: How to run a script

- Under the "Scripting" tab is where code lines can be written and executed or complete scripts can be loaded and executed

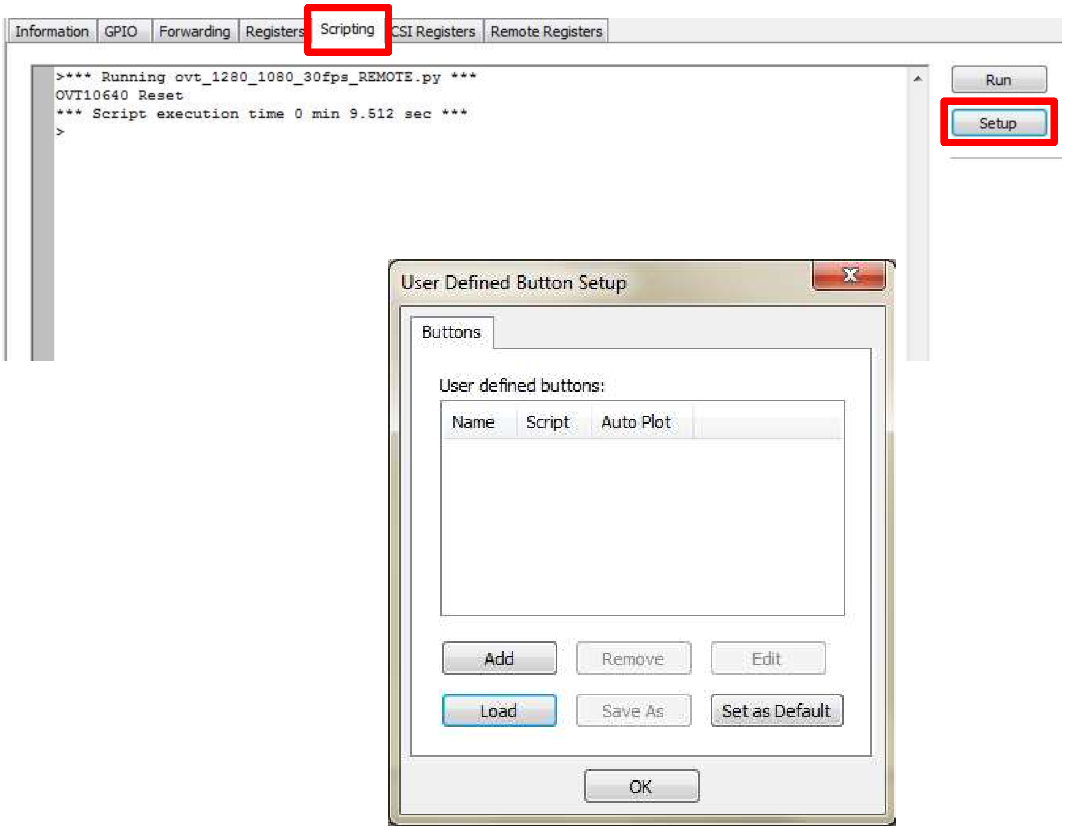
- Clicking the "Run" button will allow you to load and run a pre-written script, e.g. for initializing the deserializer, initializing an image sensor or to activate the pattern generator.



```
>*** Running ovt_1280_1080_30fps_REMOTE.py ***
OVT10640 Reset
*** Script execution time 0 min 9.512 sec ***
>
```

Using the Scripting tab: How to add a button

- By clicking the "Setup" button, a new window will appear with different options for adding/loading user defined buttons.



```
>*** Running ovt_1280_1080_30fps_REMOTE.py ***
OVT10640 Reset
*** Script execution time 0 min 9.512 sec ***
>
```

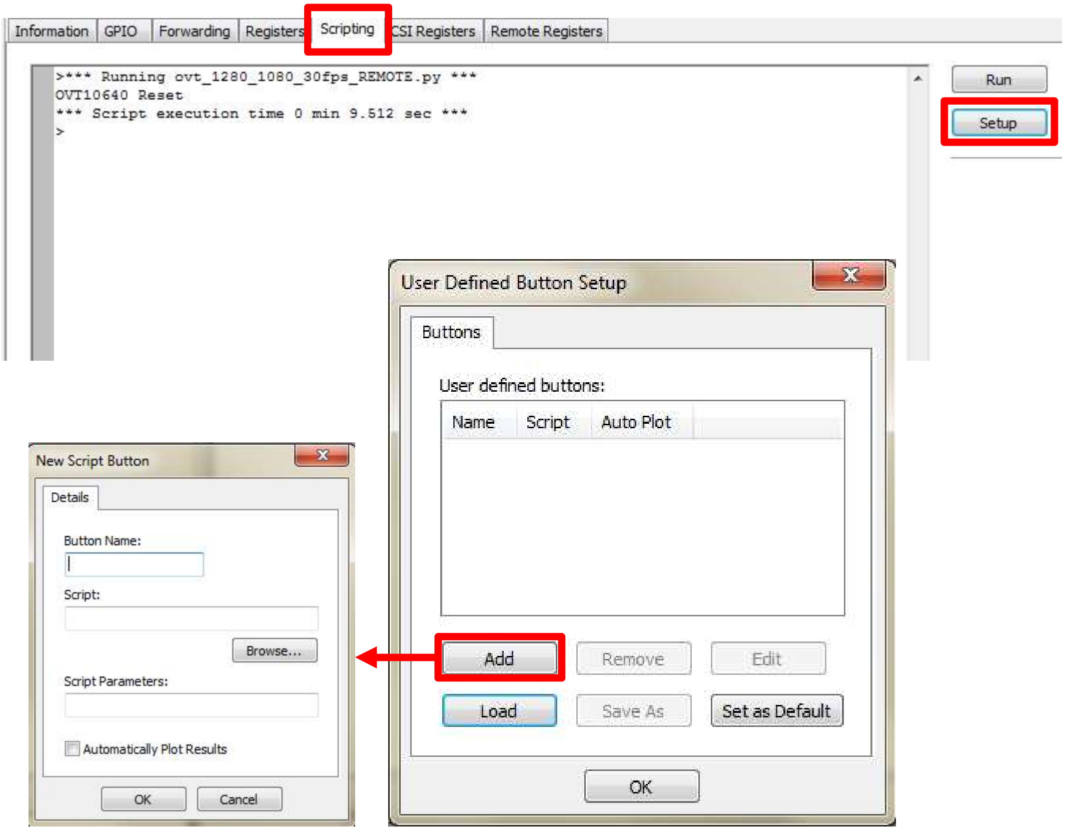
User defined buttons:

Name	Script	Auto Plot

Using the Scripting tab: How to add a button

- To add a new button to run a specific script, click on "Add", a new window will appear, type a name and then click on "Browse" to load the required script and press "OK".

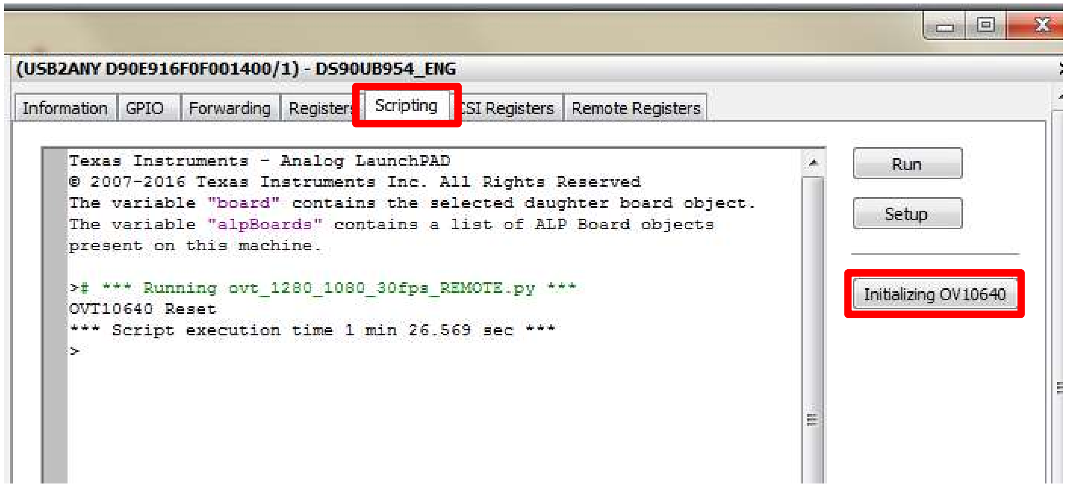
- You can "Save" the created button or make it as default and then press "OK".



```
>*** Running ovt_1280_1080_30fps_REMOTE.py ***
OVT10640 Reset
*** Script execution time 0 min 9.512 sec ***
>
```

Using the Scripting tab: How to add a button

- The newly created button will appear under the "Setup" button which is under the "Scripting" tab.



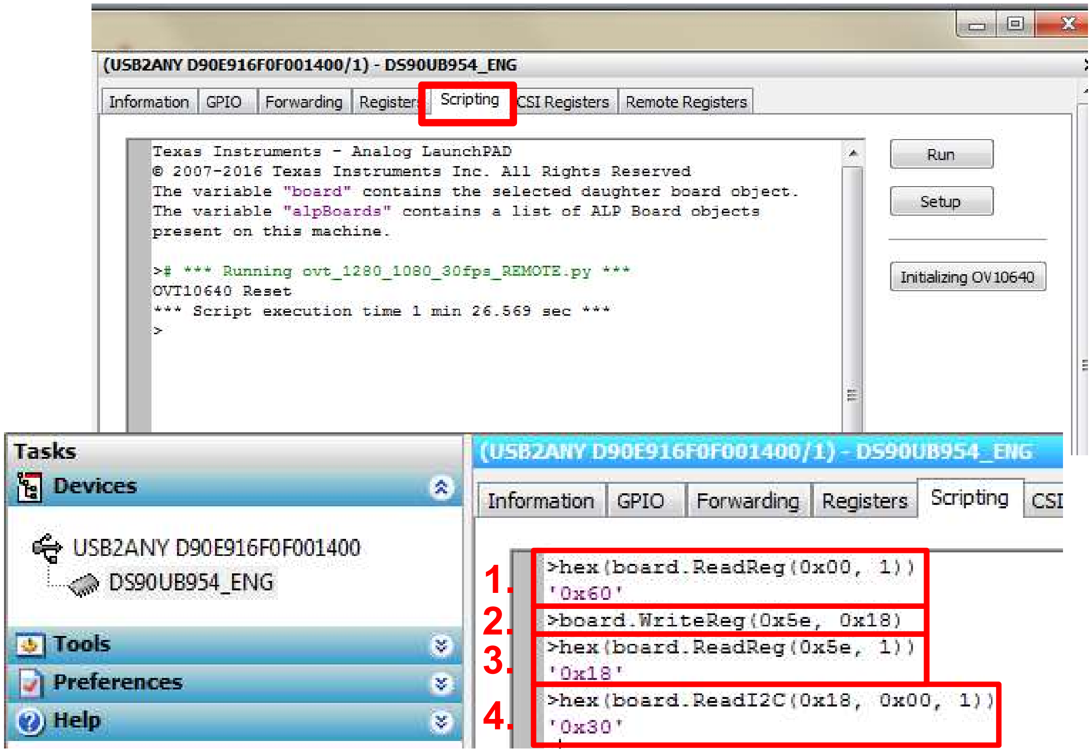
```
Texas Instruments - Analog LaunchPAD
© 2007-2016 Texas Instruments Inc. All Rights Reserved
The variable "board" contains the selected daughter board object.
The variable "alpBoards" contains a list of ALP Board objects
present on this machine.

># *** Running ovt_1280_1080_30fps_REMOTE.py ***
OVT10640 Reset
*** Script execution time 1 min 26.569 sec ***
>
```

Using the Scripting tab: How to write I^{2}C command

- Under the "Scripting" tab, commands can be written to read or write registers locally and remotely.

- Examples:

1. To read the value of a Register Locally
2. To write new value to any Register locally
3. To read back the written value
4. To read the value of a Register remotely



```
Texas Instruments - Analog LaunchPAD
© 2007-2016 Texas Instruments Inc. All Rights Reserved
The variable "board" contains the selected daughter board object.
The variable "alpBoards" contains a list of ALP Board objects
present on this machine.

># *** Running ovt_1280_1080_30fps_REMOTE.py ***
OVT10640 Reset
*** Script execution time 1 min 26.569 sec ***
>
```

```
1. >hex(board.ReadReg(0x00, 1))
   '0x60'
2. >board.WriteReg(0x5e, 0x18)
3. >hex(board.ReadReg(0x5e, 1))
   '0x18'
4. >hex(board.ReadI2C(0x18, 0x00, 1))
   '0x30'
```

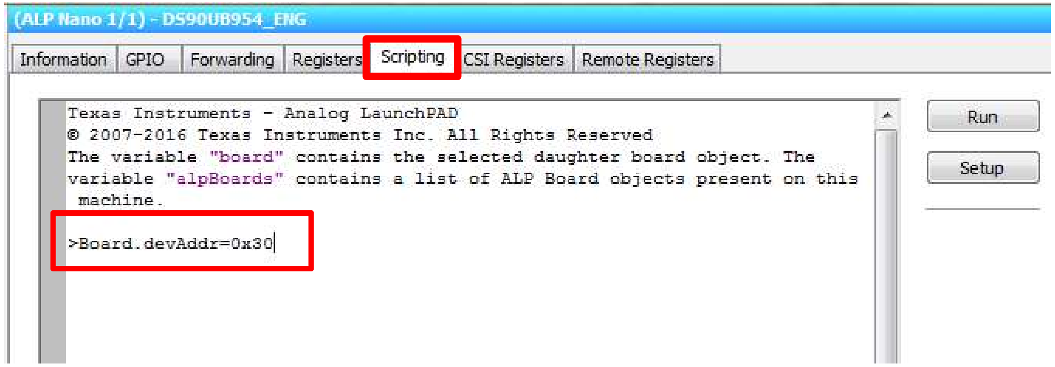
Using the Scripting tab: How to write I^{2}C command

- To change the device address to be contacted from ALP, go to scripting and write the following command:

- Board.devAddr=hex address (example 0x30)

- In this way the detected address is overwritten.

- Note: do not go to Information tab.

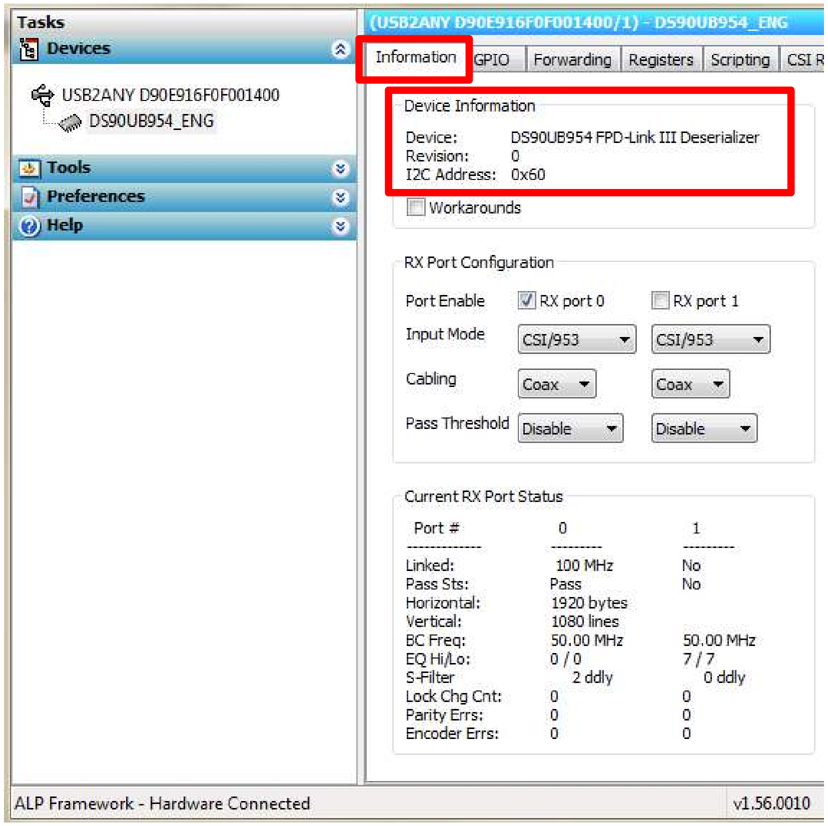


```
Texas Instruments - Analog LaunchPAD
© 2007-2016 Texas Instruments Inc. All Rights Reserved
The variable "board" contains the selected daughter board object. The
variable "alpBoards" contains a list of ALP Board objects present on this
 machine.

>Board.devAddr=0x30
```

Successful I²C communication to the EVM.

- To confirm there is successful I^{2}C communication to the connected device (DS90UB954), go to the "Information" tab. In the "Device Information" section, the connected Device name, Revision and Device I^{2}C Address information should appear there.

- After that, follow the following steps.



Device Information

Device: DS90UB954 FPD-Link III Deserializer
Revision: 0
I2C Address: 0x60

□ Workarounds

RX Port Configuration

Port Enable	☑ RX port 0	□ RX port 1
Input Mode	CSI/953	CSI/953
Cabling	Coax	Coax
Pass Threshold	Disable	Disable

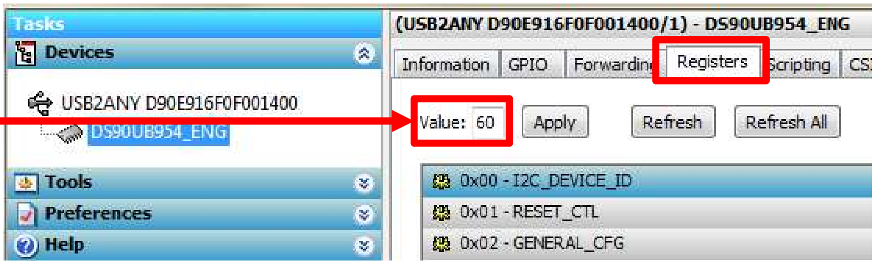
Current RX Port Status

```
 Port #            0              1
-------------   ---------      ---------
Linked:          100 MHz        No
Pass Sts:       Pass            No
Horizontal:     1920 bytes
Vertical:       1080 lines
BC Freq:        50.00 MHz       50.00 MHz
EQ Hi/Lo:       0 / 0           7 / 7
S-Filter             2 ddly          0 ddly
Lock Chg Cnt:   0               0
Parity Errs:    0               0
Encoder Errs:   0               0
```

ALP Framework - Hardware Connected   v1.56.0010

Successful I²C communication to the EVM.

1. Read device (DS90UB954) ID locally
   - Local doesn't require transactions across the bidirectional control channel (BCC)

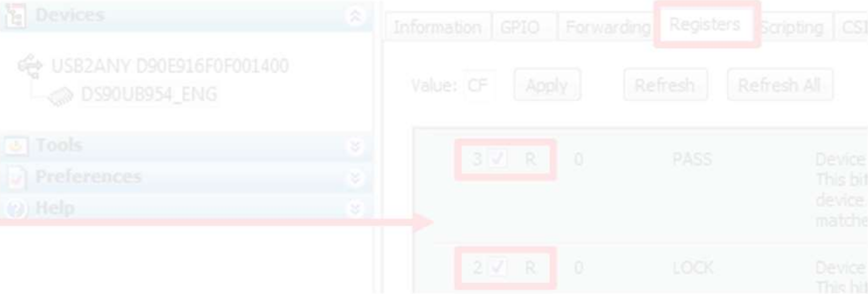


(USB2ANY D90E916F0F001400/1) - DS90UB954_ENG

Value: 60

- 0x00 - I2C_DEVICE_ID
- 0x01 - RESET_CTL
- 0x02 - GENERAL_CFG

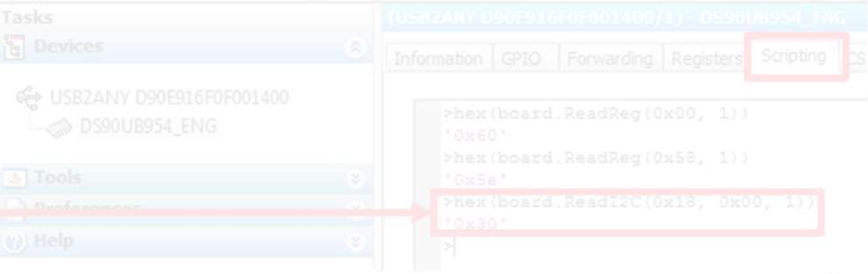
2. Verify SER and DES are Locked
   - On the (DS90UB954) , DEVICE_STS Register 0x04 bit [3:2] holds the Lock and Pass status of the device



Value: CF

3	✓	R	0	PASS	Device This bit device matche
2	✓	R	0	LOCK	Device This bit

3. Read device ID of SER using DES
   - Transaction over the BCC and verify it works

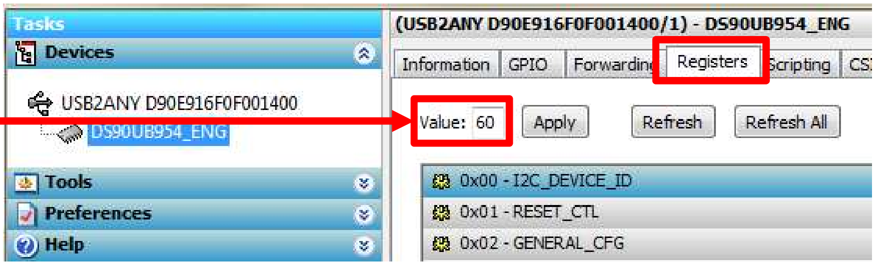


(USB2ANY D90E916F0F001400/1) - DS90UB954_ENG

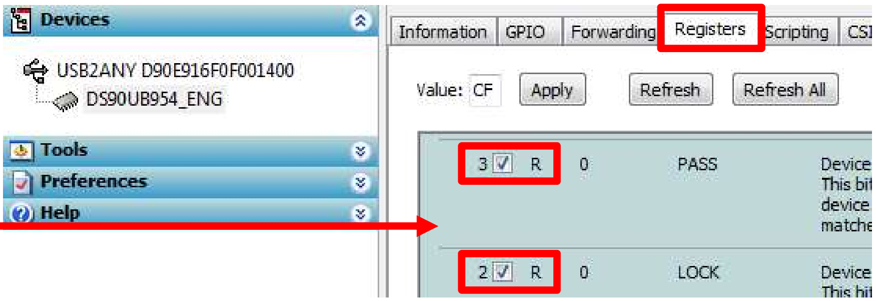
```
>hex(board.ReadReg(0x00, 1))
'0x60'
>hex(board.ReadReg(0x58, 1))
'0x5e'
>hex(board.ReadI2C(0x18, 0x00, 1))
'0x30'
>
```

Successful I²C communication to the EVM.

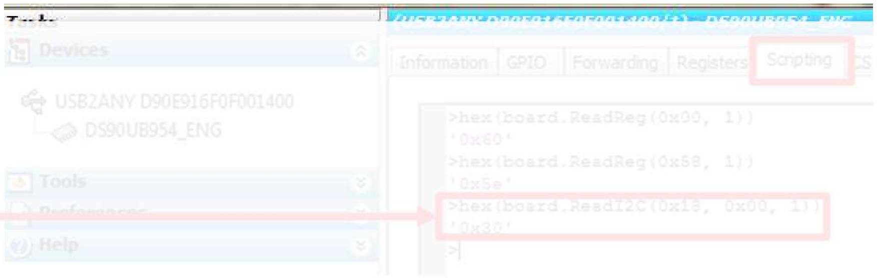
1. Read device (DS90UB954) ID locally
    - Local doesn't require transactions across the bidirectional control channel (BCC)



2. Verify SER and DES are Locked
    - On the (DS90UB954) , DEVICE_STS Register 0x04 bit [3:2] holds the Lock and Pass status of the device



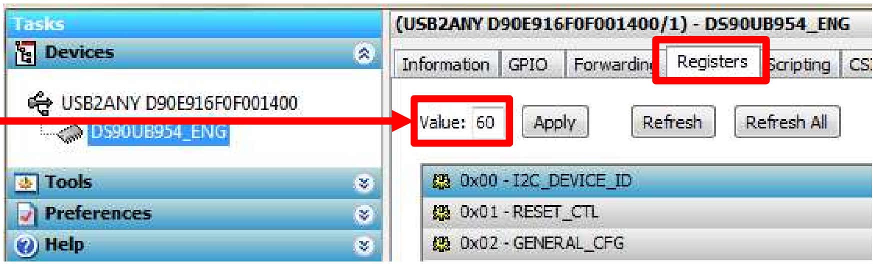
3. Read device ID of SER using DES
    - Transaction over the BCC and verify it works



```
>hex(board.ReadReg(0x00, 1))
'0x60'
>hex(board.ReadReg(0x58, 1))
'0x5e'
>hex(board.ReadI2C(0x18, 0x00, 1))
'0x30'
>
```

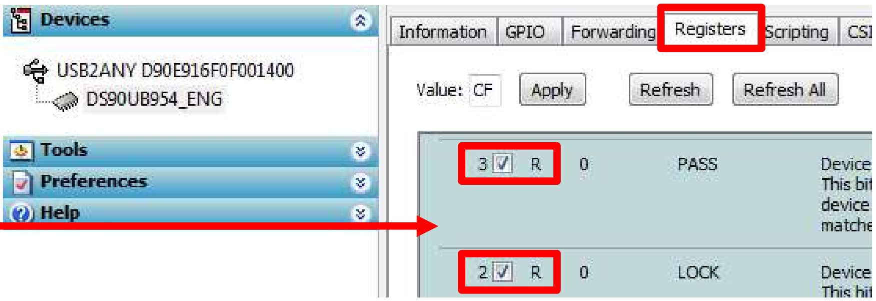
Successful I^{2}C communication to the EVM.

1. Read device (DS90UB954) ID locally
   - Local doesn't require transactions across the bidirectional control channel (BCC)



Value: 60

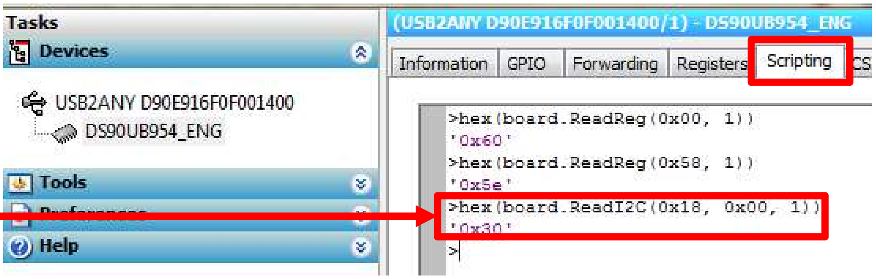
2. Verify SER and DES are Locked
   - On the (DS90UB954) , DEVICE_STS Register 0x04 bit [3:2] holds the Lock and Pass status of the device



Value: CF

3 ☑	R	0	PASS	Device This bi device matche
2 ☑	R	0	LOCK	Device This bi

3. Read device ID of SER using DES
   - Transaction over the BCC and verify it works



```
>hex(board.ReadReg(0x00, 1))
'0x60'
>hex(board.ReadReg(0x58, 1))
'0x5e'
>hex(board.ReadI2C(0x18, 0x00, 1))
'0x30'
>
```

Changing the voltage on USB2ANY to 1.8V

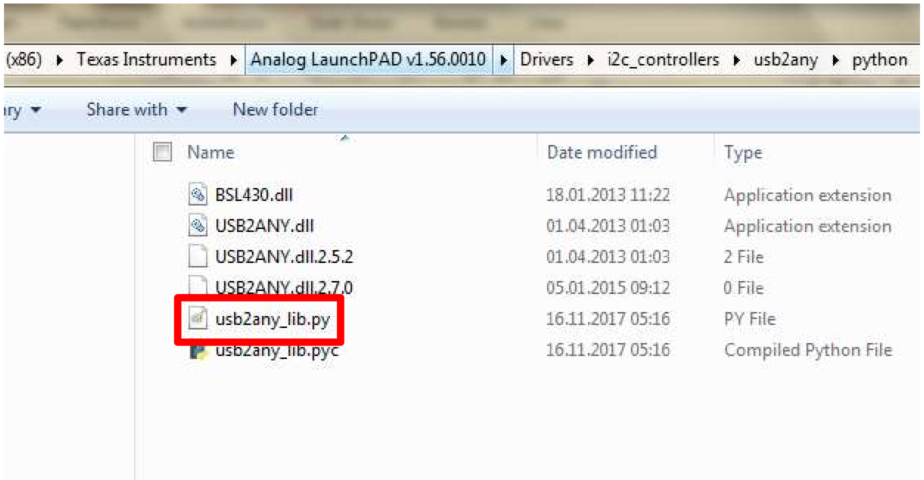
- USB2ANY works by default with 3.3V.

- To make it working with 1.8V instead, do the following steps:

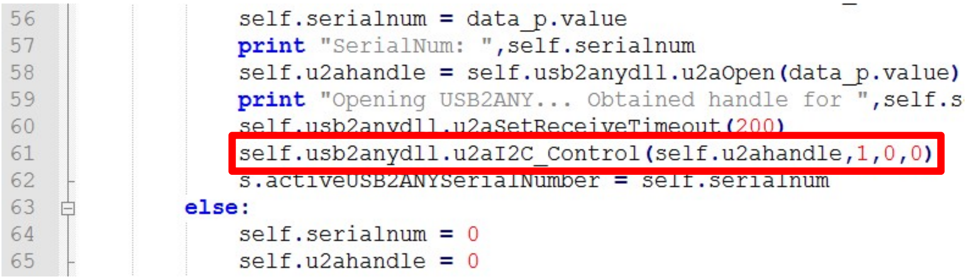
1. Go to:
   C:\Program Files (x86)\Texas Instruments\Analog LaunchPAD v1.54.0010\Drivers\i2c_controllers\usb2any\python\usb2any_lib.py

2. Change Line 61 to:

self.usb2anydll.u2aI2C_Control(self.u2ahandle,1,0,1)
self.usb2anydll.u2aPower_WriteControl(self.u2ahandle,1,0)


(x86) ▸ Texas Instruments ▸ Analog LaunchPAD v1.56.0010 ▸ Drivers ▸ i2c_controllers ▸ usb2any ▸ python
ry ▾
Share with ▾
New folder

Name	Date modified	Type
BSL430.dll	18.01.2013 11:22	Application extension
USB2ANY.dll	01.04.2013 01:03	Application extension
USB2ANY.dll.2.5.2	01.04.2013 01:03	2 File
USB2ANY.dll.2.7.0	05.01.2015 09:12	0 File
usb2any_lib.py	16.11.2017 05:16	PY File
usb2any_lib.pyc	16.11.2017 05:16	Compiled Python File



```
56          self.serialnum = data_p.value
57          print "SerialNum: ",self.serialnum
58          self.u2ahandle = self.usb2anydll.u2aOpen(data_p.value)
59          print "Opening USB2ANY... Obtained handle for ",self.s
60          self.usb2anydll.u2aSetReceiveTimeout(200)
61          self.usb2anydll.u2aI2C_Control(self.u2ahandle,1,0,0)
62          s.activeUSB2ANYSerialNumber = self.serialnum
63      else:
64          self.serialnum = 0
65          self.u2ahandle = 0
```

Additional resources

For more information about the Texas Instruments FPD-Link Portfolio, please visit
http://www.ti.com/lsds/ti/interface/fpd-linkiii-ser-des-overview.page

Contact the FPD-Link support team at
http://e2e.ti.com/support/interface/high_speed_interface/f/138

Texas Instruments

This material is provided strictly "as-is," for informational purposes only, and without any warranty.
Use of this material is subject to TI's **Terms of Use**, viewable at TI.com