Converting an X310 into an NI-USRP Rio

Contents

This Application Note is deprecated. Use it at your own risk!

AN-503

Date	Author	Details
2016-05-03	Tim Fountain	Initial creation

This Application Note explains how to use an Ettus Research-branded USRP with LabVIEW, and in effect, convert it into an NI-USRP RIO.

NOTE: While this process is technically possible, NI/Ettus does *not* officially support it and does *not* guarantee providing any technical support for it. The user performs this process *at their own risk*. The process is documented here as a convenience to users.

This document outlines the steps necessary to modify an Ettus X310 + associated daughterboards (CBX, WBX, SBX or UBX) into the identical NI model (NI USRP-294x or NI USRP-295x). Note that you must have identical daughterboards in each X310 slot for LabVIEW to function. Identical daughterboards would be 2xCBX or 2xUBX for instance.

This document was created with version 15.0 of the NI USRP driver. In NI USRP 15.5 and later there will be a single LabVIEW vi that will conduct all 3 steps automatically.

There are 2 steps that need to be done to make an X310 into a USRP RIO.

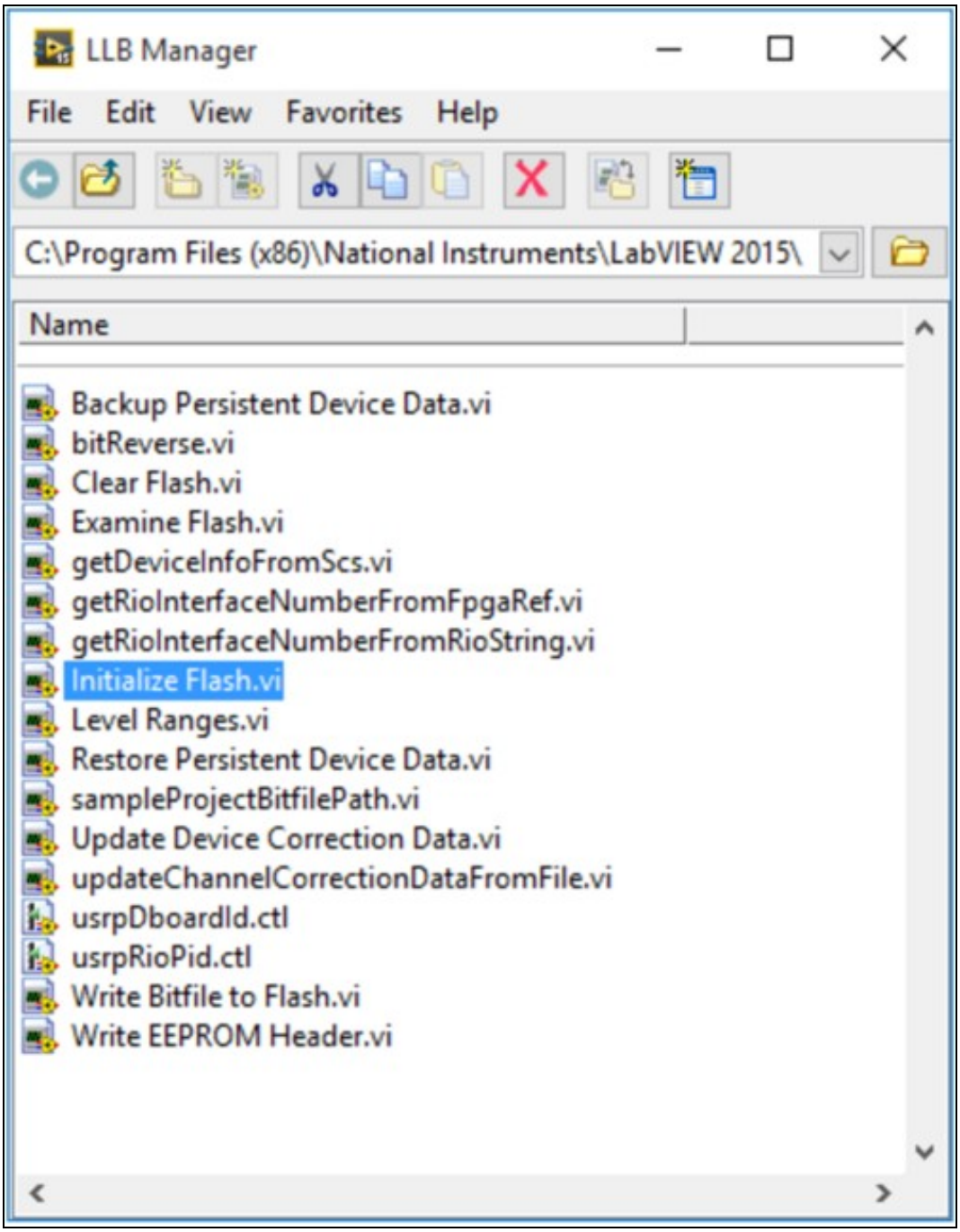
The daughterboard IDs need to be burned into the EEPROM.

The default location for the utilities is

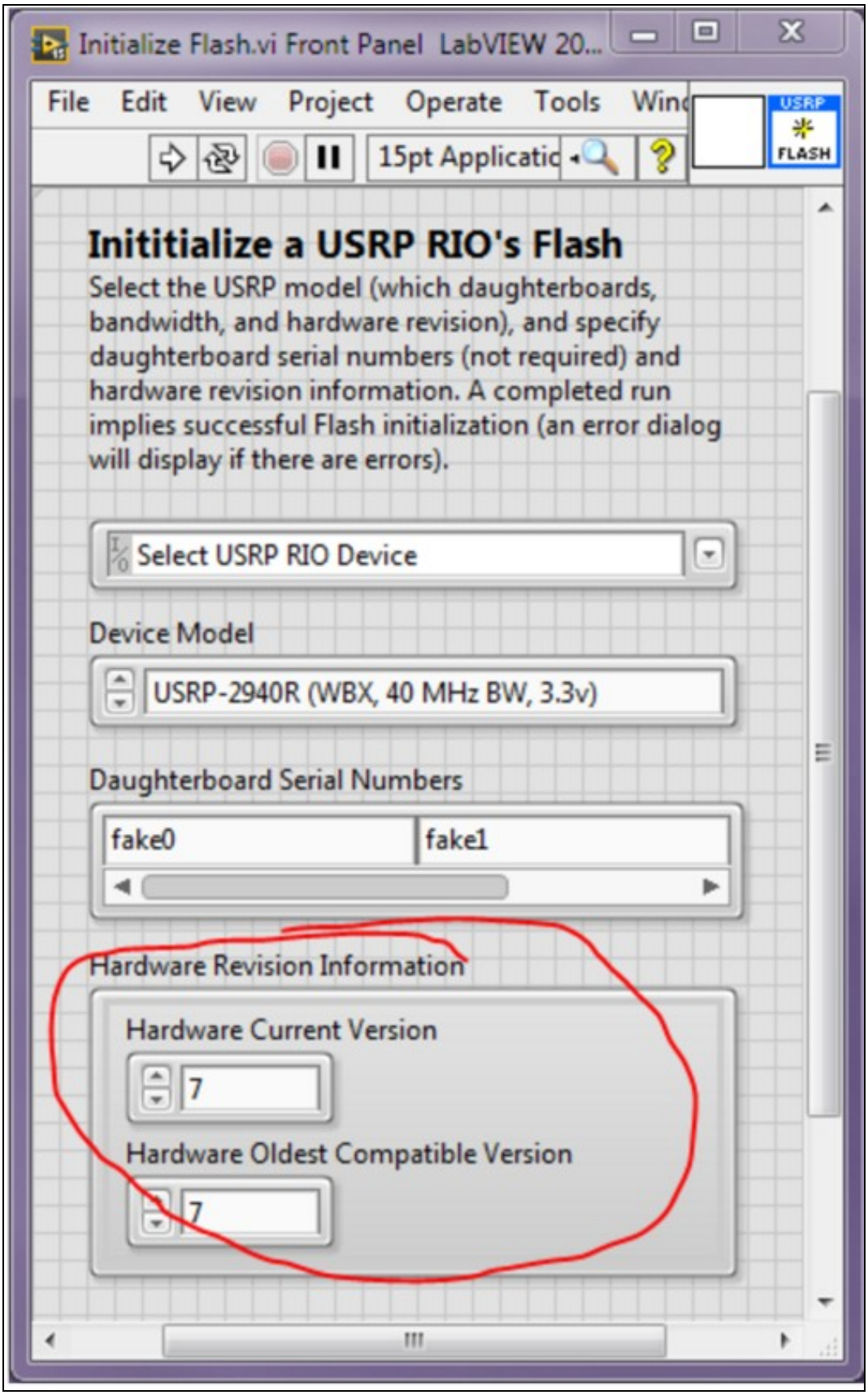
```
C:\Program Files (x86)\National Instruments\LabVIEW 2015\vi.lib\LabVIEW Targets\FPGA\USRP\niusrprio_tools.llb
```

- Note - edit path accordingly if you have a different version of LabVIEW and/or you have installed the x64 version

Use the initialize Flash.vi to load the correct daughterboard ID?s and serial numbers. The vi is auto populated with the supported daughterboard ID?s, the complete list is included in appendix 1 for reference. The serial numbers are not critical bit can be matched to the physical daughterboard serial numbers which are found on a printed label on each daughterboard.

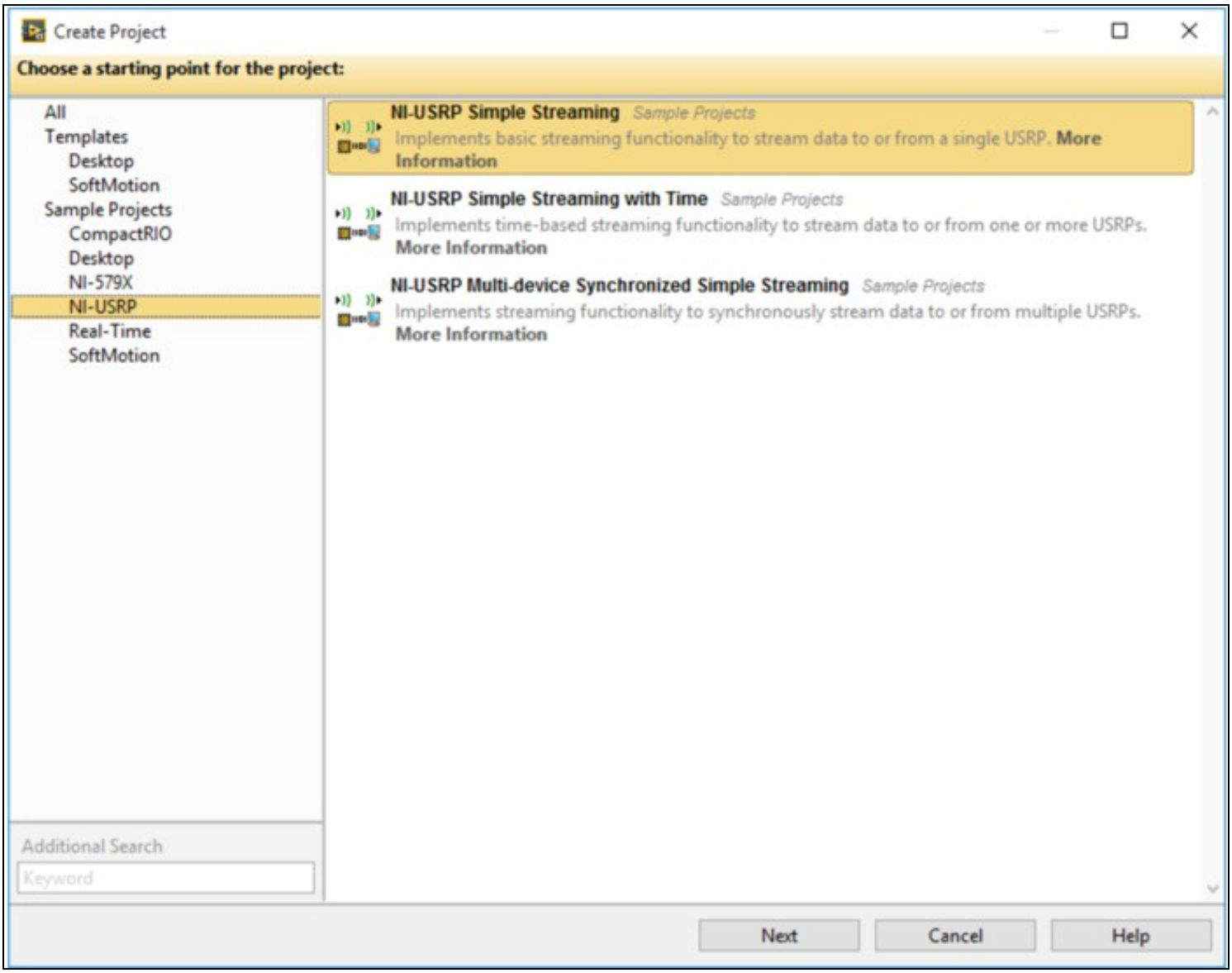


The letter revision on the X310 motherboard should be to set the HW current version (rev A = 1, B =2, etc). Set the oldest compatible version to the current HW version too. For revisions 6 and below (A-F), the 3.3v device model must be set. For revisions 7 and above (G and H), use the 1.6v device model must be set.

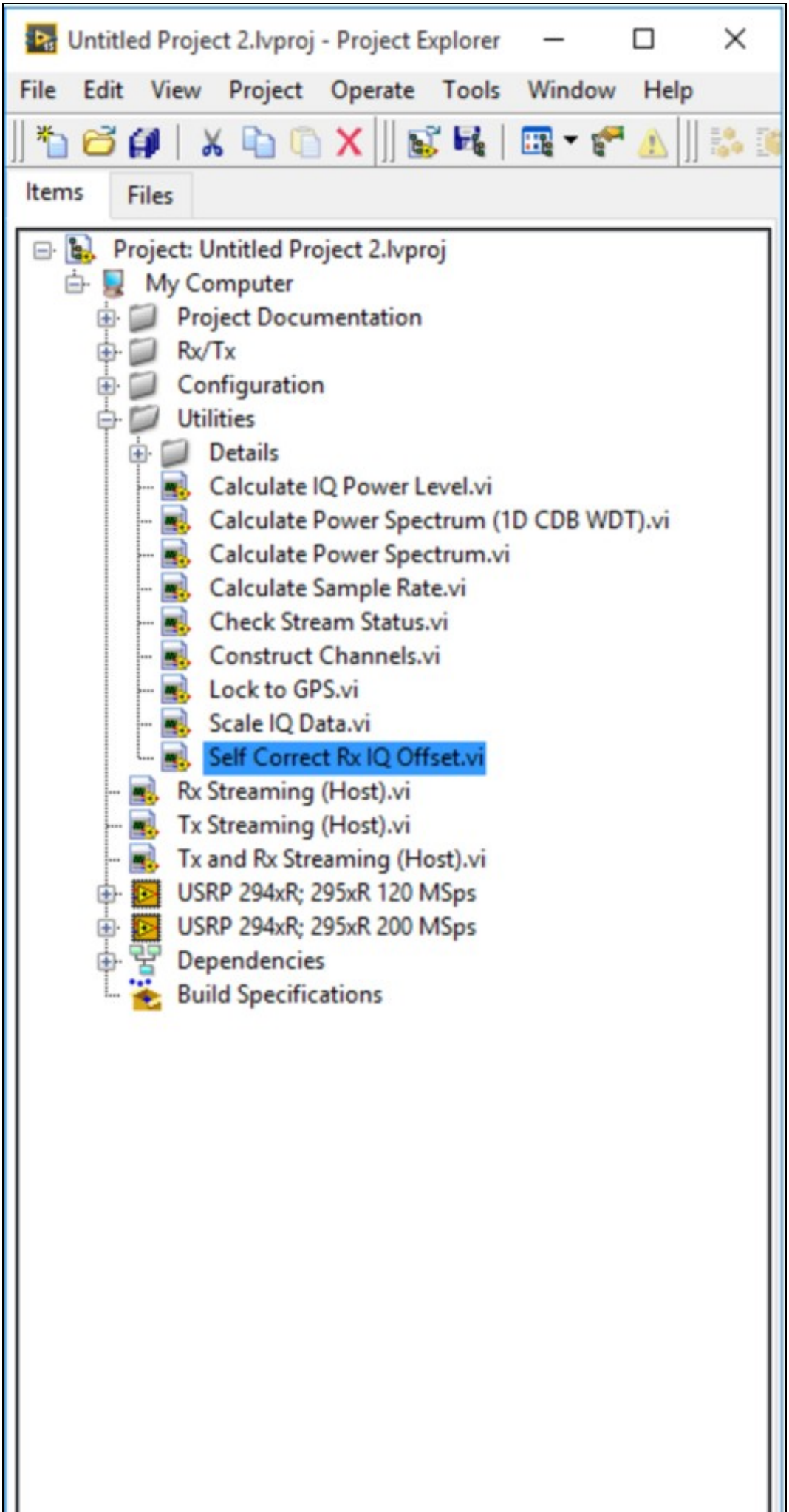


IQ imbalance corrections need to be loaded into the EEPROM.

The easiest way to load the IQ imbalance corrections into the daughterboard EEPROM is to create a sample USRP-RIO project. From the default LabVIEW windows, click create new project and select NI-USRP Simple Streaming:



Once the project has been created, navigate to the utilities in the project window and open the Self Correct RX IQ Offset.vi:


Untitled Project 2.lvproj - Project Explorer
File Edit View Project Operate Tools Window Help
Items Files
Project: Untitled Project 2.lvproj
My Computer
Project Documentation
Rx/Tx
Configuration
Utilities
Details
Calculate IQ Power Level.vi
Calculate Power Spectrum (1D CDB WDT).vi
Calculate Power Spectrum.vi
Calculate Sample Rate.vi
Check Stream Status.vi
Construct Channels.vi
Lock to GPS.vi
Scale IQ Data.vi
Self Correct Rx IQ Offset.vi
Rx Streaming (Host).vi
Tx Streaming (Host).vi
Tx and Rx Streaming (Host).vi
USRP 294xR; 295xR 120 MSps
USRP 294xR; 295xR 200 MSps
Dependencies
Build Specifications

This .vi will automatically calculate the optimal IQ imbalance correction factors and load them into the factory default location on the daughterboard EEPROM. Note there is no method at this time to load DC offset corrections without sending the unit back to the factory.

NAME	ID
B200	0x7737
B210	0x7738
B200mini	0x7739
B205	0x773a

Name (as reported by UHD)	RX ID	TX ID	Notes
Unknown	0xfff1	0xfff0	
Basic	0x0001	0x0000	
LF	0x000f	0x000e	

Name (as reported by UHD)	RX ID	TX ID	Notes
WBX LO	0x0051	0x0050	(Not registered in UHD)
WBX	0x0053	0x0052	
WBX + Simple GDB	0x0053	0x004f	
WBX v3	0x0057	0x0056	
WBX v3 + Simple GDB	0x0057	0x004f	
WBX v4	0x0063	0x0062	
WBX v4 + Simple GDB	0x0063	0x004f	
WBX-120	0x0081	0x0080	
WBX-120 + Simple GDB	0x0081	0x004f	

Name (as reported by UHD)	RX ID	TX ID	Notes
SBX	0x0054	0x0055	v3
SBX v4	0x0065	0x0064	
SBX v5	0x0069	0x0068	
SBX-120	0x0083	0x0082	

Name (as reported by UHD)	RX ID	TX ID	Notes
CBX	0x0067	0x0066	v3
CBX-120	0x0085	0x0084	

Name (as reported by UHD)	RX ID	TX ID	Notes
UBX v0.3	0x0074	0x0073	Prototype
UBX v0.4	0x0076	0x0075	Prototype
UBX-40 v1	0x0078	0x0077	
UBX-160 v1	0x007A	0x0079	

Name (as reported by UHD)	RX ID	TX ID	Notes
TwinRX v1.0	0x0091	0xffff	(0x90 Reserved)

Name	ID	Notes
DBS Rx	0x0002	
TV Rx	0x0003	
Flex 400 Rx	0x0004	
Flex 900 Rx	0x0005	
Flex 1200 Rx	0x0006	
Flex 2400 Rx	0x0007	
Flex 400 Tx	0x0008	
Flex 900 Tx	0x0009	
Flex 1200 Tx	0x000a	
Flex 2400 Tx	0x000b	
TV Rx Rev 2	0x000c	
DBS Rx ClkMod	0x000d	
DBSRX2	0x012	
Flex 400 Rx MIMO A	0x0014	
Flex 900 Rx MIMO A	0x0015	
Flex 1200 Rx MIMO A	0x0016	
Flex 2400 Rx MIMO A	0x0017	
Flex 400 Tx MIMO A	0x0018	
Flex 900 Tx MIMO A	0x0019	
Flex 1200 Tx MIMO A	0x001a	
Flex 2400 Tx MIMO A	0x001b	

Flex 400 Rx MIMO B	0x0024
Flex 900 Rx MIMO B	0x0025
Flex 1200 Rx MIMO B	0x0026
Flex 2400 Rx MIMO B	0x0027
Flex 400 Tx MIMO B	0x0028
Flex 900 Tx MIMO B	0x0029
Flex 1200 Tx MIMO B	0x002a
Flex 2400 Tx MIMO B	0x002b
Flex 2200 Rx MIMO B	0x002c
Flex 2200 Tx MIMO B	0x002d
Flex 1800 Rx	0x0030
Flex 1800 Tx	0x0031
Flex 1800 Rx MIMO A	0x0032
Flex 1800 Tx MIMO A	0x0033
Flex 1800 Rx MIMO B	0x0034
Flex 1800 Tx MIMO B	0x0035
TV Rx Rev 3	0x0040
DTT754	0x0041
DTT768	0x0042
TV Rx MIMO	0x0043
TV Rx Rev 2 MIMO	0x0044
TV Rx Rev 3 MIMO	0x0045
TVRX2	0x0046
WCDMA US	0x004d
WCDMA EU	0x004e
XCVR2450 Tx - No Div	0x0059
XCVR2450 Tx	0x0060
XCVR2450 Rx	0x0061
Bitshark Rx	0x0070
B150 v1 TX	0x0071
B150 v1 RX	0x0072

Name	ID	Notes
E300 AD9364 RevB	0x0100	
E310 AD9361 RevB	0x0110	
E300 AD9364 RevC	0x0101	
E310 AD9361 RevC	0x0111	
E330	0x0120	

Name	ID	Notes
Ettus Research	0x2500	
National Instruments	0x3923	
Cypress Semiconductor	0x04b4	

Name	ID	Notes
FX2	0x8613	
FX3	0x00f3	
FX3 (Re-enumerated)	0x00f0	
B100	0x0002	
B200	0x0020	
B210	0x0020	
B200 (NI)	0x7813	
B210 (NI)	0x7814	
B200mini	0x0021	
B205	0x0022	