Lime Microsystems Limited

Surrey Technology Centre
Occam Road
The Surrey Research Park
Guildford, Surrey GU2 7YG
United Kingdom

Tel: +44 (0) 1483 685 063
e-mail: enquiries@limemicro.com

LimeSDR-QPCIe v1.2 Quick Start Manual

- Hardware and software description-

Version: 0.02 Last modified: 11/06/2018

REVISION HISTORY

The following table shows the revision history of this document:

Date	Version	Description of Revisions
4/10/2017	0.00	Initial version
03/11/2017	0.01	Board overview, LEDs, PMOD#A, PMOD#B, LMS7002#1, LMS7002#2, DAC#1, DAC#2, ADC tables filled, board component description figures added, clock distribution, power distribution and temperature control sections filled
08/12/2017	0.02	New sections: SDRAM, GNSS, RTC, FPGA configuration, Temperature sensor, User I/O, RF Loopback Control, Main clock sources, Clock buffer source selection, VCTCXO clock tunning, Windows PCIe driver installation procedure, Linux PCIe drivers, Obtaining FPGA programming files.

Table of Contents

1. Introduction

LimeSDR-QPCIe is low-cost software defined radio board based on Lime LMS7002M Field Programmable Radio Frequency (FPRF) transceiver and Altera Cyclone V PFGA, through which apps can be programmed to support any type of wireless standard, e.g. UMTS, LTE, LoRa, GPS, WiFi, Zigbee, RFID, Digital Broadcastimng, Radar and many more.

2. LimeSDR-QPCIe Board Key Features

The LimeSDR-QPCIe development board provides a hardware platform for developing and prototyping high-performance and logic-intensive digital and RF designs using Altera’s Cyclone V FPGA and Lime Microsystems transceiver.

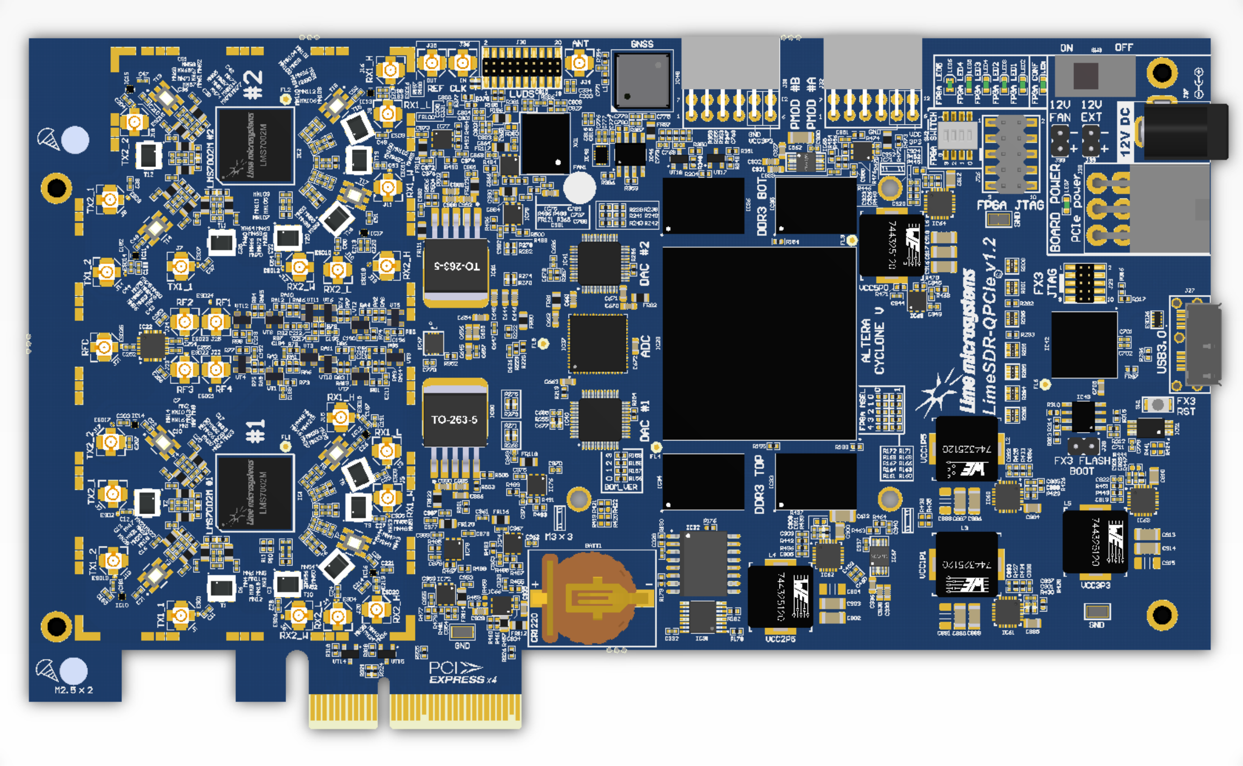


Figure 1 LimeSDR-QPCIe v1.2 board

For more information on the following topics, refer to the respective documents:

- Cyclone V device family, refer to Cyclone V Device support resources [link]
- LMS7002M transceiver resources [link]

LimeSDR-QPCIe v1.2 board features:

- **USB Interface**
  - Cypress FX3 Super Speed USB 3rd generation controller
- **FPGA Features**
  - Cyclone V, 5CGXFC7D7F31C8N device in 896-pin FBGA package
  - 150’000 logic elements
  - 6860 Kbits embedded memory
  - 312 embedded 18x18 multipliers
  - 7 PLLs
  - 9 Transceivers (2.5Gbps)
  - PCIe Hard IP Blocks
  - 2 Hard Memory Controllers
- **FPGA Configuration**
  - JTAG mode configuration
  - Active serial mode configuration
  - Possibility to update FPGA gateware by using FX3 (USB)
  - Possibility to update FPGA gateware by using PCIe interface.
- **RF**
  - 2x LMS7002M, FPRF transceivers
  - Onboard RSSI measurement circuits
  - Onboard loopback control switches
- **DACs and ADCs**
  - 2x DAC5672A, dual, 14-bit, Digital-To-Analog converters
  - 1x ADS424, Dual-Channel, 14-bit, Analog-To-Digital converter
- **Memory Devices**
  - 4 x 2Gbit DDR3 SDRAM (128M x 16)
  - 4Mbit flash for FX3 firmware
  - 128Mbit flash for FPGA gateware
  - 2 x 128Kbit and 2 x 512Kbit EEPROMs for LMS MCU firmware, LMS MCU data
  - 1 x 128K EEPROM for FX3 or FPGA data
- **Connections**
  - microUSB3.0 (type B) connector
  - PCIe x4 edge connector (Gen1)
  - Coaxial RF (U.FL) connectors
  - 2x PMOD header (0.1” pitch)
  - FPGA (0.1” pitch) and FX3 (0.05” pitch) JTAG connectors
  - 12V DC power jack and pinheader
  - LVDS connector (0.05” pitch)
  - Fan connector (12V/5V)
  - PCIe 6-pin power connector
  - Holder for coin cell CR1220 battery
- **Clock System**
  - 30.72MHz VCTCXO (precision: ±1 ppm initial, ±4 ppm stable).
  - Possibility to lock VCTCXO to external clock using ADF4002 or tune VCTCXO by onboard DAC (AD5662)
  - Programmable clock generator for the FPGA reference clock input or LMS PLLs

- VCTCXO clock output for external device synchronization.
- 1x 100 MHz, 4 x 125MHz crystal oscillators for FPGA

- **Miscellaneous devices**
  - LM75 Digital temperature sensor with 2-Wire Interface.
  - DS3231 real-time clock.
  - M0578-A3 GPS/GNSS module receiver

- **Board Size** 190mm x 106.7mm (7.48” x 4.20”)

2.1 LimeSDR-QPCIe Board Overview

LimeSDR-QPCIe board version 1.2 picture with highlighted major connectors presented in *Figure* 2. There are three connector types – data and debugging (PCIe, USB3.0, PMOD, LVDS and JTAG), power (DC jack and external supply pinheaders) and high frequency (RF and reference clock).

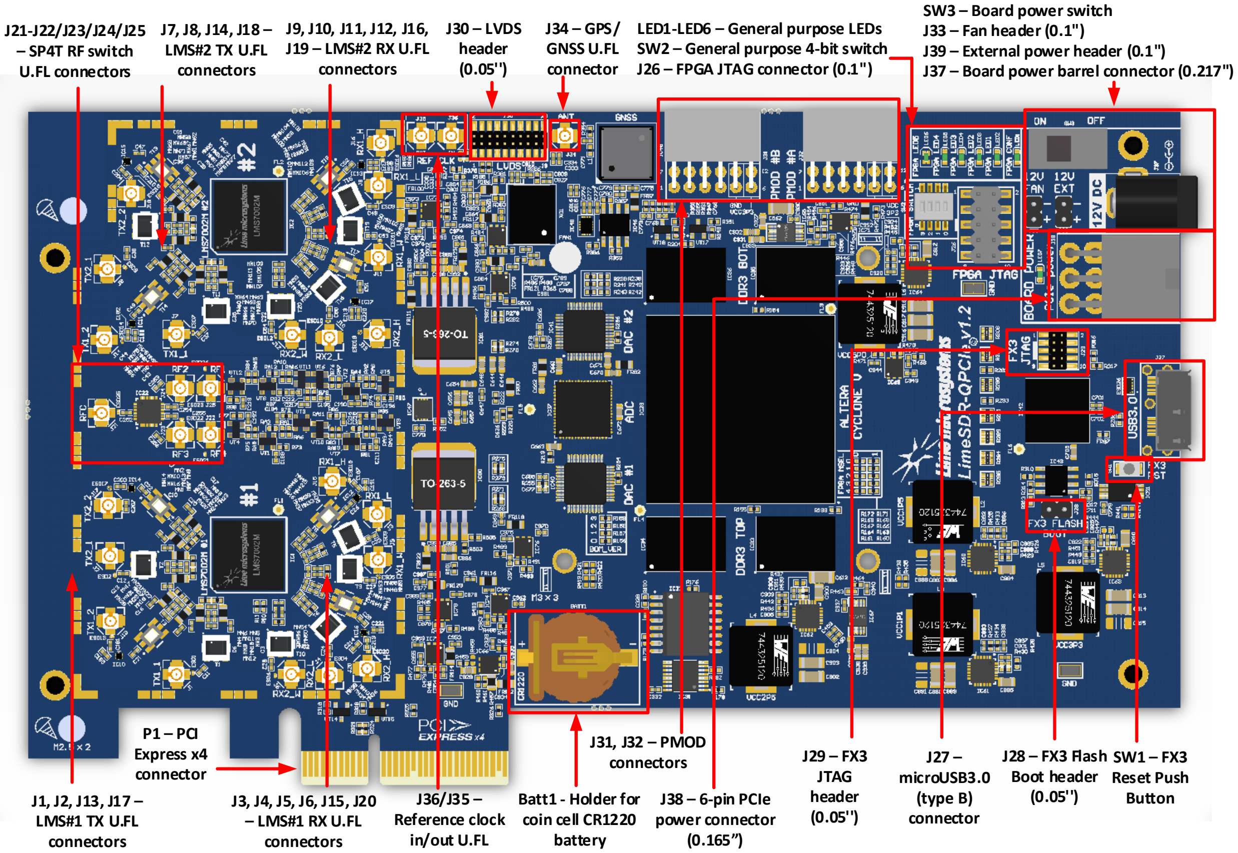

J21-J22/J23/J24/J25 – SP4T RF switch U.FL connectors
J7, J8, J14, J18 – LMS#2 TX U.FL connectors
J9, J10, J11, J12, J16, J19 – LMS#2 RX U.FL connectors
J30 – LVDS header (0.05")
J34 – GPS/ GNSS U.FL connector
LED1-LED6 – General purpose LEDs
SW2 – General purpose 4-bit switch
J26 – FPGA JTAG connector (0.1")
SW3 – Board power switch
J33 – Fan header (0.1")
J39 – External power header (0.1")
J37 – Board power barrel connector (0.217")
P1 – PCI Express x4 connector
J31, J32 – PMOD connectors
J29 – FX3 JTAG header (0.05")
J27 – microUSB3.0 (type B) connector
J28 – FX3 Flash Boot header (0.05")
SW1 – FX3 Reset Push Button
J1, J2, J13, J17 – LMS#1 TX U.FL connectors
J3, J4, J5, J6, J15, J20 – LMS#1 RX U.FL connectors
J36/J35 – Reference clock in/out U.FL
Batt1 - Holder for coin cell CR1220 battery
J38 – 6-pin PCIe power connector (0.165")

Figure 2 LimeSDR-QPCIe v1.2 Board Connectors

Board components description listed in the Table 1.

Table 1. Board components

Featured Devices		
Board Reference	**Type**	**Description**
IC1, IC2	FPRF	Field programmable RF transceivers LMS7002M
IC8	FPGA	Altera Cyclone V GX, 5CGXFC7D7F31C8N, 896-BGA
IC13	USB3.0 microcontroller	Cypress FX3 Supper Speed USB 3rd generation controller CYUSB3013
Miscellaneous devices on board		
IC7, IC8	IC	8-bit shift registers 74HC595BQ,115
IC9, IC49	IC	Bidirectional voltage shifters SN74AVC4T774RSVR

IC10, IC11, IC12, IC13, IC14, IC15, IC16, IC17	IC	100MHz – 3 GHz SPDT RF switches SKY13323-378LF
IC18, IC20, IC23, IC25, IC26, IC28	IC	12-bit ADCs MAX11108AVB+T
IC19, IC24, IC27	IC	1MHz–10GHz dual log detector/controller ADL5519
IC22	IC	SP4T RF switch PE42442A-Z
IC31	IC	4 parallel 2:1 switches TS3A5018PWR
IC37	IC	14-bit 2-channel ADC ADS4246IRGCT
IC38	IC	Dual differential amplifier ADA4930
IC39, IC56	IC	Differential line drivers SN65LVDS1DBVR
IC40, IC41	IC	Dual differential DACs DAC5672AIPFB
IC44, IC45	IC	Bidirectional 8-channel voltage translators FXLA108BQX
IC47	IC	Temperature sensor LM75
IC48	IC	GPS receiver module M10578-A3
IC50	IC	Real time clock (RTC) DS3231S#
BATT1	Holder	Holder for coin cell CR1220 battery
ESD26	TVS	USB3.0 ESD protection TVS diode
ESD1, ESD2, ESD3, ESD4, ESD5, ESD6, ESD7, ESD8, ESD9, ESD10, ESD11, ESD12, ESD13, ESD14, ESD15, ESD16, ESD17, ESD18, ESD19, ESD20, ESD21, ESD22, ESD23, ESD24, ESD25	TVS	RF connector ESD protection TVS diodes
Configuration, Status and Setup Components		
R56, R57, R58, R59	0 Ohm resistor	Board BOM version BOM_VER[3:0]. Default BOM_VER=0 (all resistors populated).
R160, R161, R163, R164, R166, R167, R168, R169, R171, R172	0 Ohm resistor	FPGA (IC8) MSEL[3:0]. Default mode: Active Serial Standard configuration.
[R268, R271, R275, R279], [R269, R273, R277, R281]	0 Ohm resistor	DAC#1 differential channels TX1_BB_I/Q connection selection to either LMS7002M #1 or LMS7002M #2. Default populated group is [R268, R271, R275, R279]. Resistor groups are defined in [] brackets.
[R270, R274, R278, R282], [R272, R276, R280, R283]	0 Ohm resistor	DAC#2 differential channels TX2_BB_I/Q connection selection to either LMS7002M #1 or LMS7002M #2. Default populated group is [R270, R274, R278, R282]. Resistor groups are defined in [] brackets.

R364, R379, R384 and their respective power connecting resistors R365, R380, R382	0 Ohm resistor	Clock buffer (IC52) CLKin0 (pin 13) clock source selection. R364 and R365 are populated by default.
R368, R372, R374	0 Ohm resistor	Clock buffer (IC52) CLKin1 (pin 28) clock source selection. R374 is populated by default.
R375	0 Ohm resistor	Clock buffer (IC52) source (CLKin0 or CLKin1) selection. If unpopulated, clock source is CLKin0 (default). If populated, clock source is CLKin1.
R302, R305, R307	10 kOhm resistor	USB3.0 microcontroller (IC13) boot configuration (PMODE0[2:0]) resistors. Default mode: SPI boot, On Failure - USB Boot
R294, R296, R298	10 kOhm resistor	USB3.0 microcontroller (IC13) crystal/clock frequency selection (FSLC[2:0]) resistors. Default mode: 19.2MHz crystal
J28, R313	Pin header, 0 Ohm resistor	USB3.0 microcontroller (IC13) boot source (Flash memory or USB), 0.1” pitch jumper or 0402 0R resistor. In normal operation jumper or resistor must be placed.
J29	JTAG chain pin header	USB3.0 microcontroller (IC13) debugging pin header, 0.05” pitch
SW1	Push-button	USB3.0 microcontroller reset button
J26	JTAG chain pin header	FPGA programming pin header for Altera USB-Blaster download cable, 0.1” pitch
LED1	Green status LED	FPGA configuration done LED
LED2-LED5	Green status LEDs	User defined general purpose green LEDs
LED6	Red-green status LED	User defined general purpose dual colour LED
General User Input/Output		
J31, J32	Connector 0.1”	PMOD connectors
SW2	Switch	4-bit FPGA switch
J33	Pin header	Board cooling fan pin header, 0.1”
Memory Devices		
IC3, IC5, IC51	EEPROM	128Kbit (16K x 8) EEPROM, LMS7002 MCU firmware and general purpose memory
IC4, IC6	EEPROM	512Kbit (64K x 8) EEPROM, connected to main I2C bus
IC30	Flash memory	128Mbit (16M x 8) Flash for FPGA configuration (unpopulated)

IC32	Flash memory	128Mbit (16M x 8) Flash for FPGA configuration
IC33, IC34, IC35, IC36	DDR3 memory	2Gbit (128M x 16) DDR3 SDRAM
Communication Ports		
J27	USB3.0 connector	microUSB3.0 (type B) connector
P1	PCIe connector	PCI Express (Gen1) x4 connector
Clock Circuitry		
XO1	VCOCXO	10MHz voltage- and oven-controlled crystal oscillator
XO2, XO3	VCTCXO	30.72MHz voltage-controlled crystal oscillator
XO4	VCTCXO	40MHz voltage-controlled crystal oscillator
IC57	IC	Programmable clock generator for the FPGA reference clock input and RF boards
IC53	IC	ADF4002 phase detector
IC54	IC	16-bit DAC for VCTCXO/VCOCXO frequency tuning
IC52	IC	Clock buffer
IC55	IC	Clock buffer
J36	U.FL connector	Reference clock input
J35	U.FL connector	Reference clock output
XO5	Crystal oscillator	100MHz single-ended FPGA clock
XO6	Crystal oscillator	125MHz single-ended FPGA clock
XO7	Crystal oscillator	125MHz differential FPGA-DDR clock
XO8	Crystal oscillator	125MHz differential FPGA-DDR clock
XO9	Crystal oscillator	125MHz differential FPGA clock for PCIe REFCLK1
IC56	IC	Single-ended to differential clock converter. Clock source is IC57 pin 9. Connected to FPGA PCIe REFCLK2 and LVDS connector J30.
Power Supply		
J37	DC input jack	External 12V DC power supply
J38	Header	6-pin PCIe power connector, 0.165” pitch
J39	Pin header	External 12V DC power supply and main internal power rail

LimeSDR-QPCIe board version 1.2 picture with highlighted top components are presented in *Figure 3*.

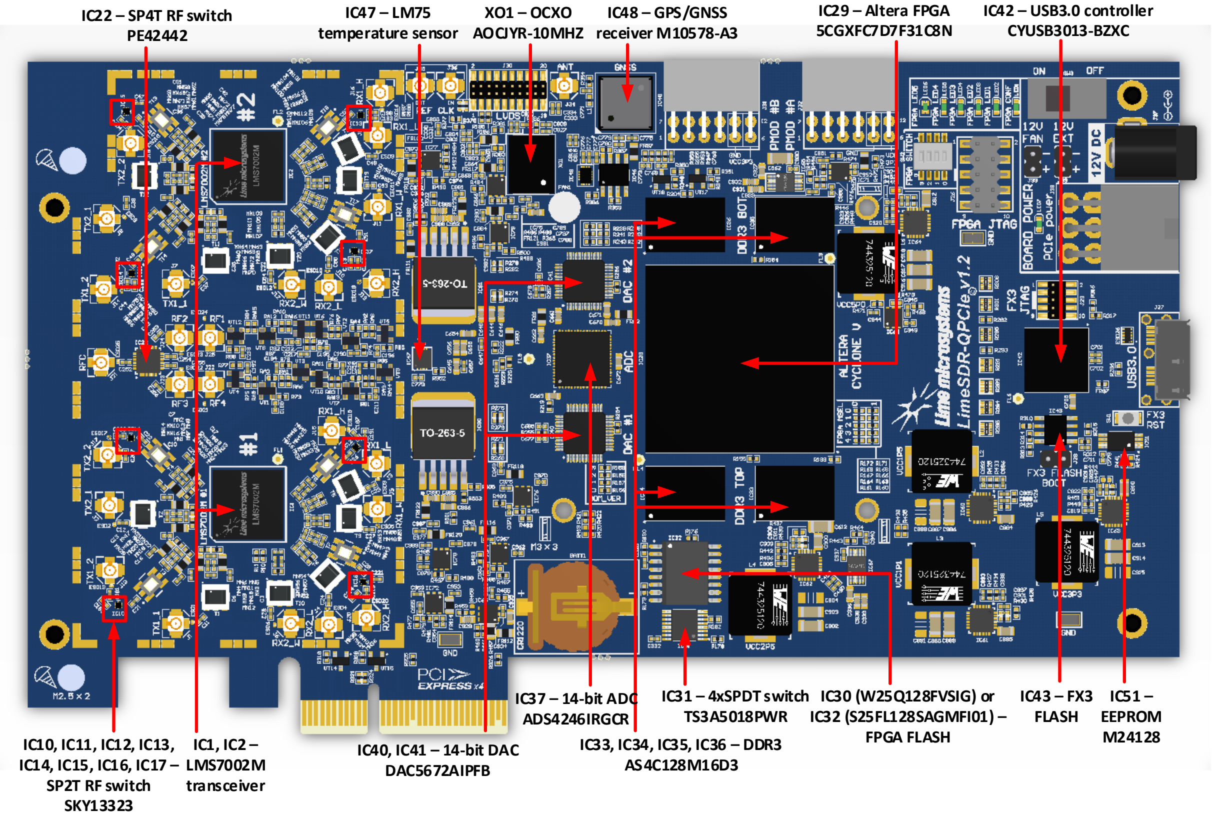

IC22 – SP4T RF switch PE42442
IC47 – LM75 temperature sensor
XO1 – OCXO AOCJYR-10MHZ
IC48 – GPS/GNSS receiver M10578-A3
IC29 – Altera FPGA 5CGXFC7D7F31C8N
IC42 – USB3.0 controller CYUSB3013-BZXC
IC37 – 14-bit ADC ADS4246IRGCR
IC31 – 4xSPDT switch TS3A5018PWR
IC30 (W25Q128FVSIG) or IC32 (S25FL128SAGMFI01) – FPGA FLASH
IC43 – FX3 FLASH
IC51 – EEPROM M24128
IC10, IC11, IC12, IC13, IC14, IC15, IC16, IC17 – SP2T RF switch SKY13323
IC1, IC2 – LMS7002M transceiver
IC40, IC41 – 14-bit DAC DAC5672AIPFB
IC33, IC34, IC35, IC36 – DDR3 AS4C128M16D3

Figure 3 LimeSDR-QPCIe v1.2 Top Components

LimeSDR-QPCIe board version 1.2 picture with highlighted bottom components is presented in *Figure 4*.

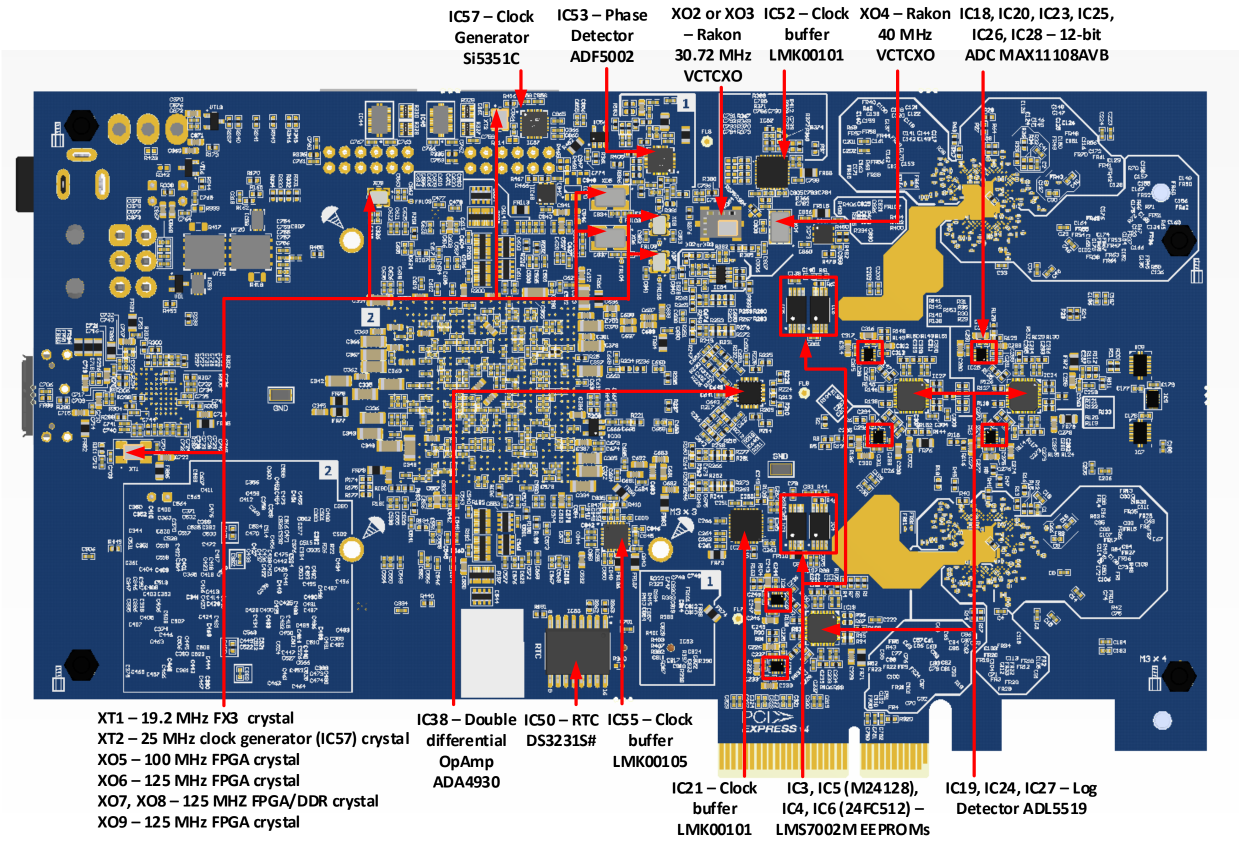

IC57 – Clock Generator Si5351C
IC53 – Phase Detector ADF5002
XO2 or XO3 – Rakon 30.72 MHz VCTCXO
IC52 – Clock buffer LMK00101
XO4 – Rakon 40 MHz VCTCXO
IC18, IC20, IC23, IC25, IC26, IC28 – 12-bit ADC MAX11108AVB
XT1 – 19.2 MHz FX3 crystal
XT2 – 25 MHz clock generator (IC57) crystal
XO5 – 100 MHz FPGA crystal
XO6 – 125 MHz FPGA crystal
XO7, XO8 – 125 MHZ FPGA/DDR crystal
XO9 – 125 MHz FPGA crystal
IC38 – Double differential OpAmp ADA4930
IC50 – RTC DS3231S#
IC55 – Clock buffer LMK00105
IC21 – Clock buffer LMK00101
IC3, IC5 (M24128), IC4, IC6 (24FC512) – LMS7002M EEPROMs
IC19, IC24, IC27 – Log Detector ADL5519

Figure 4 LimeSDR-QPCIe v1.2 Bottom Components

2.2 LimeSDR-QPCIe Board Architecture

The heart of the LimeSDR-QPCIe board is Altera Cyclone V GX FPGA. Its main function is to transfer digital data between the PC through an edge PCIE and a USB3.0 connector. The block diagram for LimeSDR-QPCIe board is presented in the *Figure 5*.

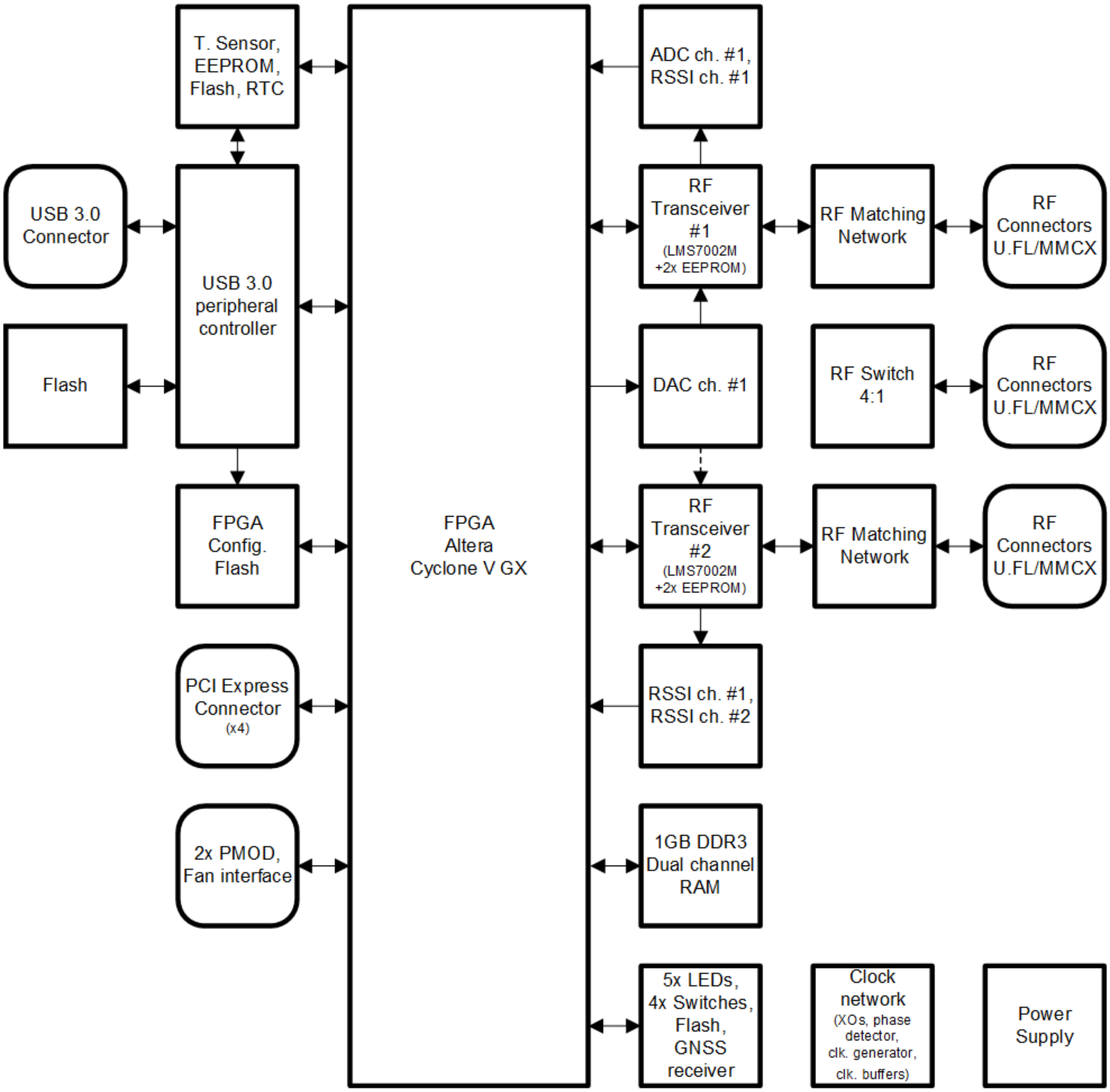

T. Sensor, EEPROM, Flash, RTC
USB 3.0 Connector
USB 3.0 peripheral controller
Flash
FPGA Config. Flash
PCI Express Connector (x4)
2x PMOD, Fan interface
FPGA Altera Cyclone V GX
ADC ch. #1, RSSI ch. #1
RF Transceiver #1 (LMS7002M +2x EEPROM)
DAC ch. #1
RF Transceiver #2 (LMS7002M +2x EEPROM)
RSSI ch. #1, RSSI ch. #2
1GB DDR3 Dual channel RAM
5x LEDs, 4x Switches, Flash, GNSS receiver
RF Matching Network
RF Switch 4:1
RF Matching Network
Clock network (XOs, phase detector, clk. generator, clk. buffers)
RF Connectors U.FL/MMCX
RF Connectors U.FL/MMCX
RF Connectors U.FL/MMCX
Power Supply

Figure 5 LimeSDR-QPCIe v1.2 Diagram

2.2.1 FPGA configuration

FPGA is set to use x1 Active Serial (AS) configuration scheme. In this scheme if valid configuration file exists in FLASH memory (IC30 or IC32) it is automatically loaded after power is applied to the board. In Table 2 it is listed resistor setup for AS configuration mode select.

Table 2 FPGA configuration setup

Schematic signal name	Logic level	0R Resistor setup		Comment
MSEL0	H	R160 (NF)	R161 (Fit)	
MSEL1	H	R163 (NF)	R164 (Fit)	
MSEL2	L	R166 (Fit)	R167 (NF)	
MSEL3	L	R168 (Fit)	R169 (NF)	
MSEL4	H	R171 (NF)	R172 (Fit)	

There are two options which allows to change configuration file in FLASH memory:

- **USB 3.0 controller –** CYUSB3013 (IC42) has access to configuration memory. With valid firmware and software, gateware for FPGA can be uploaded into FLASH memory (IC30 or IC32) by using USB3.0 cable. IC42 can initiate FPGA reconfiguration. For signal interconnect details see chapter **2.2.2.3 USB 3.0 Controller**.
- **JTAG Header –** 10pin connector (J26) provides access to FPGA JTAG chain. By using external download cable such as USB-Blaster and Quartus II Programmer software FLASH memory (IC30 or IC32) can be reprogrammed. JTAG connections are listed in Table 3.

Table 3 JTAG header (J26)

Connector pin	Schematic signal name	FPGA pin (IC29)	Comment
1	FPGA_JTAG_TCK	AC7	R170 Pull-Down resistor
2	VCC2P5	-	
3	FPGA_JTAG_TDO	W9	
4	VCC2P5	-	
5	FPGA_JTAG_TMS	V7	R162 Pull-Up resistor
6	-	-	
7	NC	-	
8	-	-	
9	FPGA_JTAG_TDI	U7	R165 Pull-Up resistor
10	GND	-	

2.2.2 Main components

This chapter describes main components mounted on LimeSDR-QPCIe v1.2 board.

2.2.2.1 LMS7002M RF transceiver

There are two LMS7002M field programmable RF transceiver ICs (LMS7002M#1 - IC1 and LMS7002M#1 - IC2), interface signals can be acknowledged by corresponding names LMSx_*, where x can be 1 or 2. For example LMS1_* signals belongs to IC1 and LMS2_* belongs to IC2.

In the following manner interface and control signals are described below:

- **Digital Interface Signals:** LMS7002 is using data bus LMSx_DIQ1_D[11:0] and LMSx_DIQ2_D[11:0], LMSx_ENABLE_IQSEL1 and LMSx_ENABLE_IQSEL2, LMSx_FCLK1 and LMSx_FCLK2, LMSx_MCLK1 and LMSx_MCLK2 signals to transfer data to/from FPGA. Indexes 1 and 2 indicate transceiver digital data PORT-1 or PORT-2. Any of these ports can be used to transmit or receive data. By default, PORT-1 is selected as receive port and PORT-2 is selected as transmit port. The FCLK# is input clock and MCLK# is output clock for LMS7002M transceiver. TXNRX signals sets ports directions. For LMS7002M interface timing details refer to LMS7002M transceiver datasheet page 12-13. [link].

- **LMS Control Signals:** these signals are used for optional functionality:
  - LMSx_RXEN, LMSx_TXEN – receiver and transmitter enable/disable signals.
  - LMS_RESET – LMS7002M reset signal.

- **SPI Interface:** LMS7002M transceiver is configured via 4-wire SPI interface; FPGA_SPI0_SCLK, FPGA_SPI0_MOSI, FPGA_SPI0_MISO_LMSx, FPGA_SPI0_LMSx_SS. The SPI interface controlled from FPGA.

- **LMS I2C Interface:** LMS EEPROM are connected to this interface. The signals LMSx_I2C_SCL, LMSx_I2C_DATA is not connected to FPGA

The Table 4 and Table 5below lists RF transceiver respectively LMS7002#1 and LMS7002#2 pins, schematic signal names, FPGA interconnections and I/O standard.

Table 4 RF transceiver (LMS7002M#1) digital interface pins

Chip pin (IC1)	Chip reference (IC1)	Schematic signal name	FPGA pin	FPGA I/O standard	Comments
AM24	xoscin_rx	LMS1_RxPLL_CLK	NC	3.3V	Connected to 30.72 MHz clock
P34	MCLK2	LMS1_MCLK2	U21	2.5V/3.3V	
R29	FCLK2	LMS1_FCLK2	Y22	2.5V/3.3V	
U31	TXNRX2	LMS1_TXNRX2	U26	2.5V/3.3V	
V34	RXEN	LMS1_RXEN	Y26	2.5V/3.3V	

Chip pin (IC1)	Chip reference (IC1)	Schematic signal name	FPGA pin	FPGA I/O standard	Comments
R33	ENABLE_IQSEL2	LMS1_ENABLE_IQSEL2	AA26	2.5V/3.3V	
H30	DIQ2_D0	LMS1_DIQ2_D0	AC27	2.5V/3.3V	
J31	DIQ2_D1	LMS1_DIQ2_D1	AB27	2.5V/3.3V	
K30	DIQ2_D2	LMS1_DIQ2_D2	Y21	2.5V/3.3V	
K32	DIQ2_D3	LMS1_DIQ2_D3	AA29	2.5V/3.3V	
L31	DIQ2_D4	LMS1_DIQ2_D4	Y28	2.5V/3.3V	
K34	DIQ2_D5	LMS1_DIQ2_D5	AC26	2.5V/3.3V	
M30	DIQ2_D6	LMS1_DIQ2_D6	W27	2.5V/3.3V	
M32	DIQ2_D7	LMS1_DIQ2_D7	AA25	2.5V/3.3V	
N31	DIQ2_D8	LMS1_DIQ2_D8	V26	2.5V/3.3V	
N33	DIQ2_D9	LMS1_DIQ2_D9	AH29	2.5V/3.3V	
P30	DIQ2_D10	LMS1_DIQ2_D10	V27	2.5V/3.3V	
P32	DIQ2_D11	LMS1_DIQ2_D11	W28	2.5V/3.3V	
E5	xoscin_tx	LMS1_TxPLL_CLK	NC	3.3V	Connected to 30.72 MHz clock
AB34	MCLK1	LMS1_MCLK1	U22	2.5V/3.3V	
AA33	FCLK1	LMS1_FCLK1	Y30	2.5V/3.3V	
V32	TXNRX1	LMS1_TXNRX1	U27	2.5V/3.3V	
U29	TXEN	LMS1_TXEN	V21	2.5V/3.3V	
Y32	ENABLE_IQSEL1	LMS1_ENABLE_IQSEL1	U28	2.5V/3.3V	
AG31	DIQ1_D0	LMS1_DIQ1_D0	T28	2.5V/3.3V	
AF30	DIQ1_D1	LMS1_DIQ1_D1	Y23	2.5V/3.3V	
AF34	DIQ1_D2	LMS1_DIQ1_D2	AB28	2.5V/3.3V	
AE31	DIQ1_D3	LMS1_DIQ1_D3	T29	2.5V/3.3V	
AD30	DIQ1_D4	LMS1_DIQ1_D4	AA23	2.5V/3.3V	
AC29	DIQ1_D5	LMS1_DIQ1_D5	V22	2.5V/3.3V	
AE33	DIQ1_D6	LMS1_DIQ1_D6	V24	2.5V/3.3V	
AD32	DIQ1_D7	LMS1_DIQ1_D7	Y27	2.5V/3.3V	
AC31	DIQ1_D8	LMS1_DIQ1_D8	AC24	2.5V/3.3V	
AC33	DIQ1_D9	LMS1_DIQ1_D9	V25	2.5V/3.3V	
AB30	DIQ1_D10	LMS1_DIQ1_D10	W22	2.5V/3.3V	
AB32	DIQ1_D11	LMS1_DIQ1_D11	AA24	2.5V/3.3V	
U33	CORE_LDO_EN	LMS1_CORE_LDO_EN	Y25	2.5V/3.3V	
E27	RESET	LMS1_RESET	L21	2.5V/3.3V	
D28	SEN	FPGA_SPI0_LMS1_SS	V29	2.5V/3.3V	SPI interface

Chip pin (IC1)	Chip reference (IC1)	Schematic signal name	FPGA pin	FPGA I/O standard	Comments
C29	SCLK	FPGA_SPI0_SCLK	T25	2.5V/3.3V	SPI interface
F30	SDIO	FPGA_SPI0_MOSI	R26	2.5V/3.3V	SPI interface
F28	SDO	FPGA_SPI0_MISO_LMS1	R30	2.5V/3.3V	SPI interface
D26	SDA	LMS1_I2C_SDA	-	2.5V	Connected to EEPROM
C27	SCL	LMS1_I2C_SCL	-	2.5V	Connected to EEPROM

Table 5 RF transceiver (LMS7002M#2) digital interface pins

Chip pin (IC2)	Chip reference (IC2)	Schematic signal name	FPGA pin	FPGA I/O standard	Comments
AM24	xoscin_rx	LMS2_RxPLL_CLK	NC	3.3V	Connected to 30.72 MHz clock
P34	MCLK2	LMS2_MCLK2	U23	2.5V/3.3V	
R29	FCLK2	LMS2_FCLK2	AC29	2.5V/3.3V	
U31	TXNRX2	LMS2_TXNRX2	AC30	2.5V/3.3V	
V34	RXEN	LMS2_RXEN	AE25	2.5V/3.3V	
R33	ENABLE_IQSEL2	LMS2_ENABLE_IQSEL2	AF25	2.5V/3.3V	
H30	DIQ2_D0	LMS2_DIQ2_D0	AA28	2.5V/3.3V	
J31	DIQ2_D1	LMS2_DIQ2_D1	AJ30	2.5V/3.3V	
K30	DIQ2_D2	LMS2_DIQ2_D2	AB29	2.5V/3.3V	
K32	DIQ2_D3	LMS2_DIQ2_D3	AD24	2.5V/3.3V	
L31	DIQ2_D4	LMS2_DIQ2_D4	AG28	2.5V/3.3V	
K34	DIQ2_D5	LMS2_DIQ2_D5	AG27	2.5V/3.3V	
M30	DIQ2_D6	LMS2_DIQ2_D6	AB26	2.5V/3.3V	
M32	DIQ2_D7	LMS2_DIQ2_D7	AF24	2.5V/3.3V	
N31	DIQ2_D8	LMS2_DIQ2_D8	AH30	2.5V/3.3V	
N33	DIQ2_D9	LMS2_DIQ2_D9	AE23	2.5V/3.3V	
P30	DIQ2_D10	LMS2_DIQ2_D10	AG29	2.5V/3.3V	
P32	DIQ2_D11	LMS2_DIQ2_D11	AE26	2.5V/3.3V	
E5	xoscin_tx	LMS2_TxPLL_CLK	NC	3.3V	Connected to 30.72 MHz clock
AB34	MCLK1	LMS2_MCLK1	T24	2.5V/3.3V	

Chip pin (IC2)	Chip reference (IC2)	Schematic signal name	FPGA pin	FPGA I/O standard	Comments
AA33	FCLK1	LMS2_FCLK1	W30	2.5V/3.3V	
V32	TXNRX1	LMS2_TXNRX1	AF28	2.5V/3.3V	
U29	TXEN	LMS2_TXEN	AD27	2.5V/3.3V	
Y32	ENABLE_IQSEL1	LMS2_ENABLE_IQSEL1	AF29	2.5V/3.3V	
AG31	DIQ1_D0	LMS2_DIQ1_D0	AD25	2.5V/3.3V	
AF30	DIQ1_D1	LMS2_DIQ1_D1	AD29	2.5V/3.3V	
AF34	DIQ1_D2	LMS2_DIQ1_D2	AH27	2.5V/3.3V	
AE31	DIQ1_D3	LMS2_DIQ1_D3	AE30	2.5V/3.3V	
AD30	DIQ1_D4	LMS2_DIQ1_D4	AE28	2.5V/3.3V	
AC29	DIQ1_D5	LMS2_DIQ1_D5	AD30	2.5V/3.3V	
AE33	DIQ1_D6	LMS2_DIQ1_D6	AJ28	2.5V/3.3V	
AD32	DIQ1_D7	LMS2_DIQ1_D7	AF26	2.5V/3.3V	
AC31	DIQ1_D8	LMS2_DIQ1_D8	AE27	2.5V/3.3V	
AC33	DIQ1_D9	LMS2_DIQ1_D9	AJ29	2.5V/3.3V	
AB30	DIQ1_D10	LMS2_DIQ1_D10	AD28	2.5V/3.3V	
AB32	DIQ1_D11	LMS2_DIQ1_D11	AF30	2.5V/3.3V	
U33	CORE_LDO_EN	LMS2_CORE_LDO_EN	AD23	2.5V/3.3V	
E27	RESET	LMS2_RESET	AA30	2.5V/3.3V	
D28	SEN	FPGA_SPI0_LMS2_SS	U29	2.5V/3.3V	SPI interface
C29	SCLK	FPGA_SPI0_SCLK	T25	2.5V/3.3V	SPI interface
F30	SDIO	FPGA_SPI0_MOSI	R26	2.5V/3.3V	SPI interface
F28	SDO	FPGA_SPI0_MISO_LMS2	V30	2.5V/3.3V	SPI interface
D26	SDA	LMS2_I2C_SDA	-	2.5V	Connected to EEPROM
C27	SCL	LMS2_I2C_SCL	-	2.5V	Connected to EEPROM

2.2.2.2 SDRAM

LimeSDR-QPCIe board has four 2Gb DDR3 SDRAM memory ICs (AS4C128M16D3B-12BCN [link]) which are connected to Cyclone V GX FPGA. The memory can be used for data manipulation at high data rates between transceiver and FPGA. There are two independent DDR3 SDRAM interfaces:

- **DDR3 TOP** – this is 32bit data interface which consist of two x16 memory devices (IC33 AND IC34) with a single address and command bus. Interface is connected to FPGA Bank 7A and 8A and uses hard memory controller. **Error! Reference source not found.** lists DDR3 TOP interface pins.
- **DDR3 BOT** – this is 32bit data interface which consist of two x16 memory devices (IC35 AND IC36) with a single address and command bus. Interface is connected to FPGA Bank 3B and 4A and uses hard memory controller. lists DDR3 BOT interface pins.

Following Table 6 lists signal and pin information for DDR3 TOP interface and Table 7 for the DDR3 BOT interface.

Table 6 DDR3 TOP interface pins

RAM reference	RAM pin	Schematic signal name	FPGA pin (IC29)	FPGA I/O standard	Comments
Address bus (IC33 and IC34 shared signals)					
A0	N3	DDR3_TOP_A0	B11	SSTL-15 Class I	Active termination
A1	P7	DDR3_TOP_A1	A11	SSTL-15 Class I	Active termination
A2	P3	DDR3_TOP_A2	F9	SSTL-15 Class I	Active termination
A3	N2	DDR3_TOP_A3	E10	SSTL-15 Class I	Active termination
A4	P8	DDR3_TOP_A4	F16	SSTL-15 Class I	Active termination
A5	P2	DDR3_TOP_A5	E16	SSTL-15 Class I	Active termination
A6	R8	DDR3_TOP_A6	D9	SSTL-15 Class I	Active termination
A7	R2	DDR3_TOP_A7	C10	SSTL-15 Class I	Active termination
A8	T8	DDR3_TOP_A8	E12	SSTL-15 Class I	Active termination
A9	R3	DDR3_TOP_A9	D13	SSTL-15 Class I	Active termination
A10/AP	L7	DDR3_TOP_A10	B7	SSTL-15 Class I	Active termination
A11	R7	DDR3_TOP_A11	A8	SSTL-15 Class I	Active termination

A12/BC#	N7	DDR3_TOP_A12	B6	SSTL-15 Class I	Active termination
A13	T3	DDR3_TOP_A13	A6	SSTL-15 Class I	Active termination
Bank address bus (IC33 and IC34 shared signals)					
BA0	M2	DDR3_TOP_BA0	A10	SSTL-15 Class I	Active termination
BA1	N8	DDR3_TOP_BA1	F15	SSTL-15 Class I	Active termination
BA2	M3	DDR3_TOP_BA2	E15	SSTL-15 Class I	Active termination
Data bus [0:15] (IC33)					
DQ0	E3	DDR3_TOP_DQ0	C15	SSTL-15 Class I	
DQ1	F7	DDR3_TOP_DQ1	C16	SSTL-15 Class I	
DQ2	F2	DDR3_TOP_DQ2	C11	SSTL-15 Class I	
DQ3	F8	DDR3_TOP_DQ3	A13	SSTL-15 Class I	
DQ4	H3	DDR3_TOP_DQ4	D17	SSTL-15 Class I	
DQ5	H8	DDR3_TOP_DQ5	E17	SSTL-15 Class I	
DQ6	G2	DDR3_TOP_DQ6	D12	SSTL-15 Class I	
DQ7	H7	DDR3_TOP_DQ7	A14	SSTL-15 Class I	
DQ8	D7	DDR3_TOP_DQ8	B17	SSTL-15 Class I	
DQ9	C3	DDR3_TOP_DQ9	C17	SSTL-15 Class I	
DQ10	C8	DDR3_TOP_DQ10	A16	SSTL-15 Class I	
DQ11	C2	DDR3_TOP_DQ11	C14	SSTL-15 Class I	
DQ12	A7	DDR3_TOP_DQ12	F18	SSTL-15 Class I	
DQ13	A2	DDR3_TOP_DQ13	G18	SSTL-15 Class I	
DQ14	B8	DDR3_TOP_DQ14	B18	SSTL-15 Class I	
DQ15	A3	DDR3_TOP_DQ15	A19	SSTL-15 Class I	
Data bus [16:31] (IC34)					
DQ0	E3	DDR3_TOP_DQ16	D18	SSTL-15 Class I	
DQ1	F7	DDR3_TOP_DQ17	D19	SSTL-15 Class I	
DQ2	F2	DDR3_TOP_DQ18	A21	SSTL-15 Class I	
DQ3	F8	DDR3_TOP_DQ19	B21	SSTL-15 Class I	
DQ4	H3	DDR3_TOP_DQ20	E18	SSTL-15 Class I	
DQ5	H8	DDR3_TOP_DQ21	F19	SSTL-15 Class I	
DQ6	G2	DDR3_TOP_DQ22	B23	SSTL-15 Class I	
DQ7	H7	DDR3_TOP_DQ23	B24	SSTL-15 Class I	
DQ8	D7	DDR3_TOP_DQ24	C19	SSTL-15 Class I	
DQ9	C3	DDR3_TOP_DQ25	D20	SSTL-15 Class I	
DQ10	C8	DDR3_TOP_DQ26	A25	SSTL-15 Class I	
DQ11	C2	DDR3_TOP_DQ27	D22	SSTL-15 Class I	
DQ12	A7	DDR3_TOP_DQ28	C20	SSTL-15 Class I	

DQ13	A2	DDR3_TOP_DQ29	C21	SSTL-15 Class I	
DQ14	B8	DDR3_TOP_DQ30	D23	SSTL-15 Class I	
DQ15	A3	DDR3_TOP_DQ31	C25	SSTL-15 Class I	
Data mask[0:1] (IC33)					
LDM	E7	DDR3_TOP_DM0	B14	SSTL-15 Class I	
UDM	D3	DDR3_TOP_DM1	B19	SSTL-15 Class I	
Data mask[2:3] (IC34)					
LDM	E7	DDR3_TOP_DM2	C24	SSTL-15 Class I	
UDM	D3	DDR3_TOP_DM3	D25	SSTL-15 Class I	
Data strobe[0:1] (IC33)					
LDQS	G3	DDR3_TOP_DQS0_P	K17	Differential 1.5-V SSTL Class I	
LDQS#	F3	DDR3_TOP_DQS0_N	J17	Differential 1.5-V SSTL Class I	
UDQS	C7	DDR3_TOP_DQS1_P	K16	Differential 1.5-V SSTL Class I	
UDQS#	B7	DDR3_TOP_DQS1_N	L16	Differential 1.5-V SSTL Class I	
Data strobe[2:3] (IC34)					
LDQS	G3	DDR3_TOP_DQS2_P	L18	Differential 1.5-V SSTL Class I	
LDQS#	F3	DDR3_TOP_DQS2_N	K18	Differential 1.5-V SSTL Class I	
UDQS	C7	DDR3_TOP_DQS3_P	K20	Differential 1.5-V SSTL Class I	
UDQS#	B7	DDR3_TOP_DQS3_N	J19	Differential 1.5-V SSTL Class I	
Memory clock (IC33 and IC34 shared signals)					
CK#	K7	DDR3_TOP_CK_N	M8	Differential 1.5-V SSTL Class I	
CK	J7	DDR3_TOP_CK_P	M9	Differential 1.5-V SSTL Class I	
Control signals(IC33 and IC34 shared signals)					
CKE	K9	DDR3_TOP_CKE	A18	SSTL-15 Class I	
WE#	L3	DDR3_TOP_WEn	C7	SSTL-15 Class I	Active termination
CAS#	K3	DDR3_TOP_CASn	C9	SSTL-15 Class I	Active termination
RAS#	J3	DDR3_TOP_RASn	B8	SSTL-15 Class I	Active termination
CS#	L2	DDR3_TOP_CSn	J15	SSTL-15 Class I	Active termination
ODT	K1	DDR3_TOP_ODT	B13	SSTL-15 Class I	Active termination

RESET#	T2	DDR3_TOP_RESETn	B22	1.5V	Active termination
VREF (IC33 and IC34 shared signals)					
VREFDQ	H1	VREF_DDR3_TOP	-		
VREFCA	M8	VREF_DDR3_TOP	-		
Memory ZQ impedance calibration (IC33)					
ZQ	L8	DDR3_TOP_RZQ0	-		
Memory ZQ impedance calibration (IC34)					
ZQ	L8	DDR3_TOP_RZQ1	-		
FPGA OCT calibration pin					
-	-	OCT_RZQIN1	B12	SSTL-15	

Table 7 DDR3 BOT interface pins

RAM reference	RAM pin	Schematic signal name	FPGA pin (IC29)	FPGA I/O standard	Comments
Address bus (IC35 and IC36 shared signals)					
A0	N3	DDR3_BOT_A0	AJ12	SSTL-15 Class I	Active termination
A1	P7	DDR3_BOT_A1	AK12	SSTL-15 Class I	Active termination
A2	P3	DDR3_BOT_A2	AH11	SSTL-15 Class I	Active termination
A3	N2	DDR3_BOT_A3	AH12	SSTL-15 Class I	Active termination
A4	P8	DDR3_BOT_A4	AG13	SSTL-15 Class I	Active termination
A5	P2	DDR3_BOT_A5	AG14	SSTL-15 Class I	Active termination
A6	R8	DDR3_BOT_A6	AK10	SSTL-15 Class I	Active termination
A7	R2	DDR3_BOT_A7	AK11	SSTL-15 Class I	Active termination
A8	T8	DDR3_BOT_A8	AF11	SSTL-15 Class I	Active termination
A9	R3	DDR3_BOT_A9	AG11	SSTL-15 Class I	Active termination
A10/AP	L7	DDR3_BOT_A10	AJ8	SSTL-15 Class I	Active termination
A11	R7	DDR3_BOT_A11	AK8	SSTL-15 Class I	Active termination
A12/BC#	N7	DDR3_BOT_A12	AJ7	SSTL-15 Class I	Active termination
A13	T3	DDR3_BOT_A13	AK7	SSTL-15 Class I	Active termination

Bank address bus (IC35 and IC36 shared signals)					
BA0	M2	DDR3_BOT_BA0	AH9	SSTL-15 Class I	Active termination
BA1	N8	DDR3_BOT_BA1	AH10	SSTL-15 Class I	Active termination
BA2	M3	DDR3_BOT_BA2	AJ10	SSTL-15 Class I	Active termination
Data bus [0:15] (IC35)					
DQ0	E3	DDR3_BOT_DQ0	AF15	SSTL-15 Class I	
DQ1	F7	DDR3_BOT_DQ1	AE16	SSTL-15 Class I	
DQ2	F2	DDR3_BOT_DQ2	AJ14	SSTL-15 Class I	
DQ3	F8	DDR3_BOT_DQ3	AH15	SSTL-15 Class I	
DQ4	H3	DDR3_BOT_DQ4	AE17	SSTL-15 Class I	
DQ5	H8	DDR3_BOT_DQ5	AD17	SSTL-15 Class I	
DQ6	G2	DDR3_BOT_DQ6	AJ15	SSTL-15 Class I	
DQ7	H7	DDR3_BOT_DQ7	AF14	SSTL-15 Class I	
DQ8	D7	DDR3_BOT_DQ8	AK17	SSTL-15 Class I	
DQ9	C3	DDR3_BOT_DQ9	AK16	SSTL-15 Class I	
DQ10	C8	DDR3_BOT_DQ10	AG17	SSTL-15 Class I	
DQ11	C2	DDR3_BOT_DQ11	AJ18	SSTL-15 Class I	
DQ12	A7	DDR3_BOT_DQ12	AG16	SSTL-15 Class I	
DQ13	A2	DDR3_BOT_DQ13	AF16	SSTL-15 Class I	
DQ14	B8	DDR3_BOT_DQ14	AJ19	SSTL-15 Class I	
DQ15	A3	DDR3_BOT_DQ15	AH20	SSTL-15 Class I	
Data bus [16:31] (IC36)					
DQ0	E3	DDR3_BOT_DQ16	AE18	SSTL-15 Class I	
DQ1	F7	DDR3_BOT_DQ17	AD18	SSTL-15 Class I	
DQ2	F2	DDR3_BOT_DQ18	AJ20	SSTL-15 Class I	
DQ3	F8	DDR3_BOT_DQ19	AK22	SSTL-15 Class I	
DQ4	H3	DDR3_BOT_DQ20	AF19	SSTL-15 Class I	
DQ5	H8	DDR3_BOT_DQ21	AF18	SSTL-15 Class I	
DQ6	G2	DDR3_BOT_DQ22	AH21	SSTL-15 Class I	
DQ7	H7	DDR3_BOT_DQ23	AK23	SSTL-15 Class I	
DQ8	D7	DDR3_BOT_DQ24	AG19	SSTL-15 Class I	
DQ9	C3	DDR3_BOT_DQ25	AG18	SSTL-15 Class I	
DQ10	C8	DDR3_BOT_DQ26	AH24	SSTL-15 Class I	
DQ11	C2	DDR3_BOT_DQ27	AK25	SSTL-15 Class I	
DQ12	A7	DDR3_BOT_DQ28	AE20	SSTL-15 Class I	
DQ13	A2	DDR3_BOT_DQ29	AD19	SSTL-15 Class I	
DQ14	B8	DDR3_BOT_DQ30	AG24	SSTL-15 Class I	
DQ15	A3	DDR3_BOT_DQ31	AK26	SSTL-15 Class I	
Data mask[0:1] (IC35)					

LDM	E7	DDR3_BOT_DM0	AE15	SSTL-15 Class I	
UDM	D3	DDR3_BOT_DM1	AH19	SSTL-15 Class I	
Data mask[2:3] (IC36)					
LDM	E7	DDR3_BOT_DM2	AJ23	SSTL-15 Class I	
UDM	D3	DDR3_BOT_DM3	AJ27	SSTL-15 Class I	
Data strobe[0:1] (IC35)					
LDQS	G3	DDR3_BOT_DQS0_P	Y16	Differential 1.5-V SSTL Class I	
LDQS#	F3	DDR3_BOT_DQS0_N	AA16	Differential 1.5-V SSTL Class I	
UDQS	C7	DDR3_BOT_DQS1_P	Y17	Differential 1.5-V SSTL Class I	
UDQS#	B7	DDR3_BOT_DQS1_N	Y18	Differential 1.5-V SSTL Class I	
Data strobe[2:3] (IC36)					
LDQS	G3	DDR3_BOT_DQS2_P	Y20	Differential 1.5-V SSTL Class I	
LDQS#	F3	DDR3_BOT_DQS2_N	AA20	Differential 1.5-V SSTL Class I	
UDQS	C7	DDR3_BOT_DQS3_P	AB19	Differential 1.5-V SSTL Class I	
UDQS#	B7	DDR3_BOT_DQS3_N	AC19	Differential 1.5-V SSTL Class I	
Memory clock (IC35 and IC36 shared signals)					
CK#	K7	DDR3_BOT_CK_N	AA14	Differential 1.5-V SSTL Class I	
CK	J7	DDR3_BOT_CK_P	Y13	Differential 1.5-V SSTL Class I	
Control signals(IC35 and IC36 shared signals)					
sCKE	K9	DDR3_BOT_CKE	AK18	SSTL-15 Class I	
WE#	L3	DDR3_BOT_WEn	AK5	SSTL-15 Class I	Active termination
CAS#	K3	DDR3_BOT_CASn	AF9	SSTL-15 Class I	Active termination
RAS#	J3	DDR3_BOT_RASn	AG9	SSTL-15 Class I	Active termination
CS#	L2	DDR3_BOT_CSn	Y12	SSTL-15 Class I	Active termination
ODT	K1	DDR3_BOT_ODT	AH14	SSTL-15 Class I	Active termination
RESET#	T2	DDR3_BOT_RESETn	AK21	1.5V	Active termination
VREF (IC35 and IC36 shared signals)					

VREFDQ	H1	VREF_DDR3_BOT	-		
VREFCA	M8	VREF_DDR3_BOT	-		
Memory ZQ impedance calibration (IC35)					
ZQ	L8	DDR3_BOT_RZQ0	-		
Memory ZQ impedance calibration (IC36)					
ZQ	L8	DDR3_BOT_RZQ1	-		
FPGA OCT calibration pin					
-	-	OCT_RZQIN0	AK13	SSTL-15	

2.2.2.3 USB 3.0 Controller

Software can control LimeSDR-QPCIe board via the USB3 microcontroller (CYUSB3013 (FX3) [link]). The data transfer to/from the board, SPI communication, FPGA configuration is done via the USB3 controller. The controller signals description showed below:

- FX3_DQ[15:0] - FX3 16-bit GPIF data interface is connected FPGA.
- FX3_CTL[12:0] - FX3 GPIF interface control signals.
- FX3_PCLK - GPIF interface clock, connected to FPGA.
- FX3_SPI - interface is used to program FX3 firmware flash or FPGA configuration flash memory.
- FX3 I2C - bus is connected to the main I2C bus.
- PMODE[2:0] – boot options, by default boot from SPI and USB boot is enabled. If J28 jumper is present or R313 is soldered FX3 will boot from IC43 flash memory if correct firmware exists.
- SW1 – resets FX3
- J29 – FX3 JTAG programming/debugging pin header.

In the Table 8 are listed USB3.0 controller (FX3) pins, schematic signal name, FPGA interconnections and I/O standard.

Table 8. USB3 controller (FX3) pins

Chip pin (IC42)	Chip reference (IC42)	Schematic signal name	FPGA pin	I/O standard	Comment
F10	GPIO0	FX3_DQ0	U12	1.8V	
F9	GPIO1	FX3_DQ1	U11	1.8V	
F7	GPIO2	FX3_DQ2	U8	1.8V	
G10	GPIO3	FX3_DQ3	U9	1.8V	
G9	GPIO4	FX3_DQ4	T11	1.8V	
F8	GPIO5	FX3_DQ5	R10	1.8V	
H10	GPIO6	FX3_DQ6	T10	1.8V	
H9	GPIO7	FX3_DQ7	T9	1.8V	
J10	GPIO8	FX3_DQ8	V11	1.8V	
J9	GPIO9	FX3_DQ9	V9	1.8V	

Chip pin (IC42)	Chip reference (IC42)	Schematic signal name	FPGA pin	I/O standard	Comment
K11	GPIO10	FX3_DQ10	V10	1.8V	
L10	GPIO11	FX3_DQ11	W10	1.8V	
K10	GPIO12	FX3_DQ12	Y10	1.8V	
K9	GPIO13	FX3_DQ13	Y11	1.8V	
J8	GPIO14	FX3_DQ14	AA11	1.8V	
G8	GPIO15	FX3_DQ15	AA8	1.8V	
J6	GPIO16	FX3_PCLK	AB16	1.5V	
K8	GPIO17	FX3_CTL0	AA9	1.8V	
K7	GPIO18	FX3_CTL1	AB8	1.8V	
J7	GPIO19	FX3_CTL2	AC9	1.8V	
H7	GPIO20	FX3_CTL3	AD9	1.8V	
G7	GPIO21	FX3_CTL4	AF8	1.8V	
G6	GPIO22	FX3_CTL5	AF7	1.8V	
K6	GPIO23	FX3_CTL6	AG7	1.8V	
H8	GPIO24	FX3_CTL7	AF6	1.8V	
G5	GPIO25	FX3_CTL8	AG6	1.8V	
H6	GPIO26	FX3_CTL9	AH7	1.8V	
K5	GPIO27	FX3_CTL10	AH6	1.8V	
J5	GPIO28	FX3_CTL11	AH4	1.8V	
H5	GPIO29	FX3_CTL12	AH5	1.8V	
G4	GPIO30	FX3_PMODE0	-	1.8V	
H4	GPIO31	FX3_PMODE1	-	1.8V	
L4	GPIO32	FX3_PMODE2	-	1.8V	
K2	GPIO33	-	-	-	
J4	GPIO34	-	-	-	
K1	GPIO35	-	-	-	
J2	GPIO36	-	-	-	
J3	GPIO37	-	-	-	
J1	GPIO38	-	-	-	
H2	GPIO39	-	-	-	
H3	GPIO40	-	-	-	
F4	GPIO41	FPGA_CONF_DONE	L8	-	Connected to FPGA configuration status CONF_DONE. Additionally, connected to LED1.

Chip pin (IC42)	Chip reference (IC42)	Schematic signal name	FPGA pin	I/O standard	Comment
G2	GPIO42	FPGA_NSTATUS	K7	-	Connected to FPGA nSTATUS pin.
G3	GPIO43	FPGA_NCONFIG	C5	-	A high-to-low logic initiates a FPGA reconfiguration.
F3	GPIO44	FX3_AS_SW	-	-	Logic level L connects FX3_SPI interface to FPGA conf. Flash (IC30, IC32)
F2	GPIO45	FX3_SPI_AS_SS	-	-	FPGA conf. Flash slave select
F5	GPIO46	FX3_SPI_FPGA_SS	-	-	
E1	GPIO47	FX3_FPGA_GPIO0	AA10	1.8V	
E5	GPIO48	FX3_FPGA_GPIO1	AB9	1.8V	
E4	GPIO49	FX3_FPGA_GPIO2	AG8	1.8V	
D1	GPIO50	FX3_FPGA_GPIO3	AK3	1.5V	
D2	GPIO51	FX3_FPGA_GPIO4	AJ4	1.5V	
D3	GPIO52	FX3_FPGA_GPIO5	AJ3	1.5V	
D4	GPIO53	FX3_SPI_SCLK	P20	2.5V/3.3V	
C1	GPIO54	FX3_SPI_FLASH_SS	-	-	Connected to FX3 memory through header J28
C2	GPIO55	FX3_SPI_MISO	M21	2.5V/3.3V	
D5	GPIO56	FX3_SPI_MOSI	N20	2.5V/3.3V	
C4	GPIO57	-	-	-	
A3	SSRXM	FX3_USB_SSRX_P	-	-	
A4	SSRXP	FX3_USB_SSRX_N	-	-	
A6	SSTXM	FX3_USB_SSTX_IC_P	-	-	
A5	SSTXP	FX3_USB_SSTX_IC_N	-	-	
B3	R_usb3	-	-	-	USB3 precision resistor
C9	OTG_ID	FX3_USB_OTG_ID	-	-	
A9	DP	FX3_USB_D_P	-	-	
A10	DM	FX3_USB_D_N	-	-	

Chip pin (IC42)	Chip reference (IC42)	Schematic signal name	FPGA pin	I/O standard	Comment
C8	R_usb2	-	-	-	USB2 precision resistor
E11	VBUS	FX3_VBUS	-	-	
B2	FSLC[0]	-	-	-	10k pulldown for 19.2MHz crystal selection
C6	XTALIN	CYUSB_XTAL_P	-	-	
C7	XTALOUT	CYUSB_XTAL_N	-	-	
B4	FSLC[1]	-	-	-	10k pulldown for 19.2MHz crystal selection
E6	FSLC[2]	-	-	-	
D7	CLKIN	-	-	-	
D6	CLKIN_32	-	-	-	
D9	I2C_SCL	FX3_I2C_SCL/ I2C_SCL	AG23	1.5V	Voltage level reduced to connect to FPGA. Net labels on FPGA side: I2C_SCL_LS, I2C_SDA_LS
D10	I2C_SDA	FX3_I2C_SDA/ I2C_SDA	AH22	1.5V	
E7	TDI	FX3_JTAG_TDI	-	-	10-pin JTAG connector J29
C10	TDO	FX3_JTAG_TDO	-	-	
B11	TRST#	FX3_JTAG_TRST	-	-	
E8	TMS	FX3_JTAG_TMS	-	-	
F6	TCK	FX3_JTAG_TCK	-	-	
D11	O[60]	-	-	-	

2.2.2.4 ADC

There is one Dual-Channel 14-Bit, analog-to-digital converter (ADS4246 – IC37) mounted on board. ADC analog input is connected to RX BB outputs of LMS7002M#1 IC. Digital output pins are connected to FPGA.

The Table 9 lists 14-bit analog to digital converter ADC (IC37) pins, schematic signal name, FPGA interconnections and I/O standard.

Table 9 14-bit ADC (IC37) digital interface pins

Chip pin (IC37)	Chip reference (IC37)	Schematic signal name	FPGA pin	I/O standard	Comment
41	DA0_P/DA1	ADC_DA0_P	L10	1.5V	
40	DA0_M/DA0	ADC_DA0_N	L9	1.5V	

43	DA2_P/DA3	ADC_DA1_P	P10	1.5V	
42	DA2_M/DA2	ADC_DA1_N	N11	1.5V	
45	DA4_P/DA5	ADC_DA2_P	N10	1.5V	
44	DA4_M/DA4	ADC_DA2_N	N9	1.5V	
47	DA6_P/DA7	ADC_DA3_P	R12	1.5V	
46	DA6_M/DA6	ADC_DA3_N	R11	1.5V	
51	DA8_P/DA13	ADC_DA4_P	P12	1.5V	
50	DA8_M/DA12	ADC_DA4_N	N12	1.5V	
53	DA10_P/DA9	ADC_DA5_P	M12	1.5V	
52	DA10_M/DA8	ADC_DA5_N	M11	1.5V	
55	DA12_P/DA11	ADC_DA6_P	L11	1.5V	
54	DA12_M/DA10	ADC_DA6_N	K11	1.5V	
61	DB0_P/DB1	ADC_DB0_P	K12	1.5V	
60	DB0_M/DB0	ADC_DB0_N	J12	1.5V	
63	DB2_P/DB3	ADC_DB1_P	E22	1.5V	
62	DB2_M/DB2	ADC_DB1_N	E21	1.5V	
3	DB4_P/DB5	ADC_DB2_P	E10	1.5V	
2	DB4_M/DB4	ADC_DB2_N	D10	1.5V	
5	DB6_P/DB7	ADC_DB3_P	G14	1.5V	
4	DB6_M/DB6	ADC_DB3_N	F14	1.5V	
7	DB8_P/DB13	ADC_DB4_P	H12	1.5V	
6	DB8_M/DB12	ADC_DB4_N	G12	1.5V	
9	DB10_P/DB9	ADC_DB5_P	J14	1.5V	
8	DB10_M/DB8	ADC_DB5_N	H14	1.5V	
11	DB12_P/DB11	ADC_DB6_P	K13	1.5V	
10	DB12_M/DB10	ADC_DB6_N	J13	1.5V	
35	CTRL1	ADC_CTRL1	-	-	
36	CTRL2	ADC_CTRL2	-	-	
34	CTRL3	ADC_CTRL3	-	-	
29	INP_A	ADC_INA_P	-	-	
30	INM_A	ADC_INA_N	-	-	
23	VCM	ADC_VCM	-	-	
57	CLKOUTP/ CLKOUT	ADC_CLKOUT_P	L14	1.5V	
56	CLKOUTM/ UNUSED	ADC_CLKOUT_N	L13	1.5V	
19	INP_B	ADC_INB_P	-	-	
20	INM_B	ADC_INB_N	-	-	
25	CLKP	ADC_CLK_P	-	-	
26	CLKM	ADC_CLK_N	-	-	
13	SCLK	FPGA_SPI0_SCLK	T25	2.5V/3.3V	
14	SDATA	FPGA_SPI0_MOSI	R26	2.5V/3.3V	

64	SDOUT	FPGA_SPI0_MISO_ADC	L20	1.5V	
15	SEN	FPGA_SPI0_ADC_SS	E26	1.5V	
12	RESET	FPGA_ADC_RESET	D6	1.5V	

2.2.2.5 DACs

LimeSDR-QPCIe board has two 14-Bit Dual Transmit Digital-To-Analog Converters. By default, analog output pins are connected to TX BB input pads of LMS70002M#1 IC. By changing on-board resistors it can be connected to LMS70002M#2 instead. To connect DACs to LMS70002M#2 TX BB input pads R268, R271, R275, R279, R270, R274, R278, R282 resistors has to be removed and R269, R273, R277, R281, R272, R276, R280, R283 resistors has to be fitted.

The tables below list 14-bit digital to analog converter DAC#1 (IC40 - Table 10) and DAC#2 (IC41 - Table 11) pins, schematic signal name, FPGA interconnections and I/O standard.

Table 10 14-bit DAC#1 (IC40) digital interface pins

Chip pin (IC40)	Chip reference (IC40)	Schematic signal name	FPGA pin	I/O standard	Comment
14	DA0	DAC1_DA0	E27	2.5V/3.3V	
13	DA1	DAC1_DA1	F25	2.5V/3.3V	
12	DA2	DAC1_DA2	D28	2.5V/3.3V	
11	DA3	DAC1_DA3	E28	2.5V/3.3V	
10	DA4	DAC1_DA4	F30	2.5V/3.3V	
9	DA5	DAC1_DA5	E30	2.5V/3.3V	
8	DA6	DAC1_DA6	D27	2.5V/3.3V	
7	DA7	DAC1_DA7	C29	2.5V/3.3V	
6	DA8	DAC1_DA8	C30	2.5V/3.3V	
5	DA9	DAC1_DA9	D29	2.5V/3.3V	
4	DA10	DAC1_DA10	D30	2.5V/3.3V	
3	DA11	DAC1_DA11	B29	2.5V/3.3V	
2	DA12	DAC1_DA12	A29	2.5V/3.3V	
1	DA13	DAC1_DA13	B28	2.5V/3.3V	
17, 18	WRTA/WRTIQ, CLKA/CLKIQ	DAC1_CLK	-	2.5V/3.3V	Clocked from buffer IC55 pin 3. Clock value passed from FPGA to buffer from FPGA pin M23, DAC_CLK_WRT
36	DB0	DAC1_DB0	F28	2.5V/3.3V	
35	DB1	DAC1_DB1	F30	2.5V/3.3V	

Chip pin (IC40)	Chip reference (IC40)	Schematic signal name	FPGA pin	I/O standard	Comment
34	DB2	DAC1_DB2	J28	2.5V/3.3V	
33	DB3	DAC1_DB3	F29	2.5V/3.3V	
32	DB4	DAC1_DB4	K30	2.5V/3.3V	
31	DB5	DAC1_DB5	K28	2.5V/3.3V	
30	DB6	DAC1_DB6	G29	2.5V/3.3V	
29	DB7	DAC1_DB7	J29	2.5V/3.3V	
28	DB8	DAC1_DB8	J30	2.5V/3.3V	
27	DB9	DAC1_DB9	H27	2.5V/3.3V	
26	DB10	DAC1_DB10	H29	2.5V/3.3V	
25	DB11	DAC1_DB11	H30	2.5V/3.3V	
24	DB12	DAC1_DB12	H26	2.5V/3.3V	
23	DB13	DAC1_DB13	H25	2.5V/3.3V	
19, 20	WRTB/RESETIQ, CLKB/SELECTIQ	DAC1_CLK	-	2.5V/3.3V	Clocked from buffer IC55 pin 3. Clock value passed from FPGA to buffer from FPGA pin M23, DAC_CLK_WRT
48	MODE	DAC1_MODE	J23	2.5V/3.3V	
37	SLEEP	DAC1_SLEEP	J25	2.5V/3.3V	
42	GSET	-	-	3.3V	Hardwired to VCC3P3D_DAC1
43	EXTIO	-	-	-	Connected to 100nF capacitor

Table 11 14-bit DAC#2 (IC41) digital interface pins

Chip pin (IC41)	Chip reference (IC41)	Schematic signal name	FPGA pin	I/O standard	Comment
14	DA0	DAC2_DA0	R27	2.5V/3.3V	
13	DA1	DAC2_DA1	K26	2.5V/3.3V	
12	DA2	DAC2_DA2	N27	2.5V/3.3V	
11	DA3	DAC2_DA3	P30	2.5V/3.3V	
10	DA4	DAC2_DA4	N29	2.5V/3.3V	
9	DA5	DAC2_DA5	M27	2.5V/3.3V	
8	DA6	DAC2_DA6	M28	2.5V/3.3V	
7	DA7	DAC2_DA7	L26	2.5V/3.3V	
6	DA8	DAC2_DA8	L28	2.5V/3.3V	
5	DA9	DAC2_DA9	L29	2.5V/3.3V	

Chip pin (IC41)	Chip reference (IC41)	Schematic signal name	FPGA pin	I/O standard	Comment
4	DA10	DAC2_DA10	L25	2.5V/3.3V	
3	DA11	DAC2_DA11	L30	2.5V/3.3V	
2	DA12	DAC2_DA12	P29	2.5V/3.3V	
1	DA13	DAC2_DA13	N26	2.5V/3.3V	
17, 18	WRTA/WRTIQ, CLKA/CLKIQ	DAC2_CLK	-	2.5V/3.3V	Clocked from buffer IC55 pin 5. Clock value passed from FPGA to buffer from FPGA pin M23, DAC_CLK_WRT
36	DB0	DAC2_DB0	P26	2.5V/3.3V	
35	DB1	DAC2_DB1	N25	2.5V/3.3V	
34	DB2	DAC2_DB2	P25	2.5V/3.3V	
33	DB3	DAC2_DB3	R28	2.5V/3.3V	
32	DB4	DAC2_DB4	R25	2.5V/3.3V	
31	DB5	DAC2_DB5	K27	2.5V/3.3V	
30	DB6	DAC2_DB6	N24	2.5V/3.3V	
29	DB7	DAC2_DB7	M23	2.5V/3.3V	
28	DB8	DAC2_DB8	M22	2.5V/3.3V	
27	DB9	DAC2_DB9	N22	2.5V/3.3V	
26	DB10	DAC2_DB10	R20	2.5V/3.3V	
25	DB11	DAC2_DB11	T21	2.5V/3.3V	
24	DB12	DAC2_DB12	R21	2.5V/3.3V	
23	DB13	DAC2_DB13	R22	2.5V/3.3V	
19, 20	WRTB/RESETIQ, CLKB/SELECTIQ	DAC2_CLK	-	2.5V/3.3V	Clocked from buffer IC55 pin 5. Clock value passed from FPGA to buffer from FPGA pin M23, DAC_CLK_WRT
48	MODE	DAC2_MODE	L23	2.5V/3.3V	
37	SLEEP	DAC2_SLEEP	L24	2.5V/3.3V	
42	GSET	-	-	3.3V	Hardwired to VCC3P3D_DAC2
43	EXTIO	-	-	-	Connected to 100nF capacitor

2.2.2.6 GNSS receiver

LimeSDR-QPCIe board has GNSS receiver module GPS/GNSS M0578 (IC48). External active antenna for this module can be connected to J34 connector. Module is connected to FPGA (IC29), pin connections can be found on Table 12.

Table 12 GNSS receiver pin connections

GNSS pin (IC48)	GNSS chip reference (IC48)	Schematic signal name	FPGA pin (IC29)	I/O standard	Comment
9	UART RX	GNSS_RX	K25	3.3V(2.5V)	
10	UART TX	GNSS_TX	K22	3.3V(2.5V)	
6	1PPS	GNSS_TPULSE	T23	3.3V(2.5V)	
13	FIX	GNSS_FIX	R23	3.3V(2.5V)	

2.2.2.7 RTC

For applications which requires accurate time LimeSDR-QPCIe has mounted Real-Time-Clock DS3231 (IC50). Pin connections can be found in Table 13. For I2C interface see chapter **2.2.4.2 I2C interfaces.**

Table 13 RTC pin connections

RTC pin (IC50)	RTC chip reference (IC50)	Schematic signal name	FPGA pin (IC29)	I/O standard	Comment
16	SCL	I2C_SCL	AG23	1.5V	Connected through level convertor to I2C_SCL_LS.
15	SDA	I2C_SDA	AH22	1.5V	Connected through level convertor to I2C_SDA_LS.
3	INT/SQW	RTC_INT_SQW	K15	1.5V	10k, Pull-UP, resistor.
1	32kHz	RTC_32KHZ	H17	1.5V	10k, Pull-UP, resistor.

2.2.2.8 Temperature sensor

Board temperature can be measured with LM75 (IC47) temperature sensor using its I2C interface. Pin connections can be found in Table 13. For I2C interface of this sensor see chapter **2.2.4.2 I2C interfaces.**

LM75 pin (IC47)	LM75 chip reference (IC47)	Schematic signal name	FPGA pin (IC29)	I/O standard	Comment
2	SCL	I2C_SCL	AG23	1.5V	Connected through level convertor to I2C_SCL_LS.
1	SDA	I2C_SDA	AH22	1.5V	Connected through level convertor to I2C_SDA_LS.
3	OS	LM75_OS	A28	1.5V	10k, Pull-UP, resistor.

2.2.3 Connectors

This chapter describes connectors that exists on LimeSDR-QPCIe v1.2 board.

2.2.3.1 PCI Express connector

For data transfer LimeSDR – QPCIe board has PCI express connector with four lanes. PCI express interface is implemented in FPGA. Pin connection and corresponding signal names are listed in Table 14.

Table 14 PCIe connector pins

Connector pin	Schematic signal name	FPGA pin	I/O standard	Comment
B5	PCIE_SMCLK	W12	1.5V	Connected trough voltage level shifter to 1.5V
B6	PCIE_SMDAT	V12	1.5V	Connected through voltage level shifter to 1.5V
B11	PCIE_WAKEn	G26	3.3V	Signal is disconnected with not fitted R100 resistor
B14	PCIE_HSO0_P	AG2	1.5-V PCML	
B15	PCIE_HSO0_N	AG1	1.5-V PCML	
B19	PCIE_HSO1_P	AE2	1.5-V PCML	
B20	PCIE_HSO1_N	AE1	1.5-V PCML	
B23	PCIE_HSO2_P	AC2	1.5-V PCML	
B24	PCIE_HSO2_N	AC1	1.5-V PCML	
B27	PCIE_HSO3_P	AA2	1.5-V PCML	
B28	PCIE_HSO3_N	AA1	1.5-V PCML	
A11	PCIE_PERSTn	W24	3.3V	
A13	PCIE_REFCLK_P	W8	HCSL	
A14	PCIE_REFCLK_N	W7	HCSL	

A16	PCIE_HSI0_P	AF4	1.5-V PCML	AC coupled
A17	PCIE_HSI0_N	AF3	1.5-V PCML	AC coupled
A21	PCIE_HSI1_P	AD4	1.5-V PCML	AC coupled
A22	PCIE_HSI1_N	AD3	1.5-V PCML	AC coupled
A25	PCIE_HSI2_P	AB4	1.5-V PCML	AC coupled
A26	PCIE_HSI2_N	AB3	1.5-V PCML	AC coupled
A29	PCIE_HSI3_P	Y4	1.5-V PCML	AC coupled
A30	PCIE_HSI3_N	Y3	1.5-V PCML	AC coupled

2.2.3.2 LVDS connector

J30 is dedicated connector for FPGA transceiver applications.

Connector pin	Schematic signal name	FPGA pin	I/O standard	Comment
1	GND	-		
3	EXT_GXB_RX6_P	R2	Depends from design	
5	EXT_GXB_RX6_N	R1		
7	GND	-		
9	EXT_GXB_TX6_P	P4	Depends from design	AC Coupled
11	EXT_GXB_TX6_N	P3		AC Coupled
13	GND	-		
15	EXT_GXB_CLK_P	P8	Depends from design	
17	EXT_GXB_CLK_N	N7		
19	GND	-		
2	GND	-		
4	EXT_GXB_RX7_P	N2	Depends from design	
6	EXT_GXB_RX7_N	N1		
8	GND	-		
10	EXT_GXB_TX7_P	M4	Depends from design	AC Coupled
12	EXT_GXB_TX7_N	M3		AC Coupled
14	GND	-		
16	VCC5P0(VCC3P3)	-		
18	VCC5P0(VCC3P3)	-		
20	GND	-		

2.2.3.3 PMOD connectors

Two 10 pin 0.1” right angle PMOD connectors (J31, J32) are connected to the FPGA. A complete pinout description, dedicated FPGA pins and their I/O standard is presented in the tables below.

Table 15. FPGA PMOD#A connector pins

Connector pin	Schematic net name	FPGA pin	I/O standard	Comment
1	PMOD_A_PIN1_LS	AD12	1.5V	Level shifter (1.5V to 3.3V) between FPGA and connector
2	PMOD_A_PIN2_LS	AE10	1.5V	
3	PMOD_A_PIN3_LS	AF10	1.5V	
4	PMOD_A_PIN4_LS	AD13	1.5V	
7	PMOD_A_PIN7_LS	AE12	1.5V	
8	PMOD_A_PIN8_LS	AE13	1.5V	
9	PMOD_A_PIN9_LS	AJ2	1.5V	
10	PMOD_A_PIN10_LS	AJ1	1.5V	
5, 11	VCC3P3	-	1.5V	
6, 12	GND	-	GND	

Table 16. FPGA PMOD#B connector pins

Connector pin	Schematic net name	FPGA pin	I/O standard	Comment
1	PMOD_B_PIN1_LS	AF20	1.5V	Level shifter (1.5V to 3.3V) between FPGA and connector
2	PMOD_B_PIN2_LS	AG21	1.5V	
3	PMOD_B_PIN3_LS	AF21	1.5V	
4	PMOD_B_PIN4_LS	AD20	1.5V	
7	PMOD_B_PIN7_LS	AK27	1.5V	
8	PMOD_B_PIN8_LS	AE22	1.5V	
9	PMOD_B_PIN9_LS	AC21	1.5V	
10	PMOD_B_PIN10_LS	AH26	1.5V	
5, 11	VCC3P3	-	1.5V	
6, 12	GND	-	GND	

2.2.4 Low Speed Interfaces

This chapter describes low speed interfaces on LimeSDR-QPCIe v1.2 board which are dedicated to communicate between various board components.

LimeSDR-QPCIe board low speed interface is divided into FPGA-RF and FPGA-FX3 groups and are presented in Figure 6 and Figure 7. The latter block diagrams depict the main ICs, corresponding IC pin numbers, data buses and serial protocol addresses.

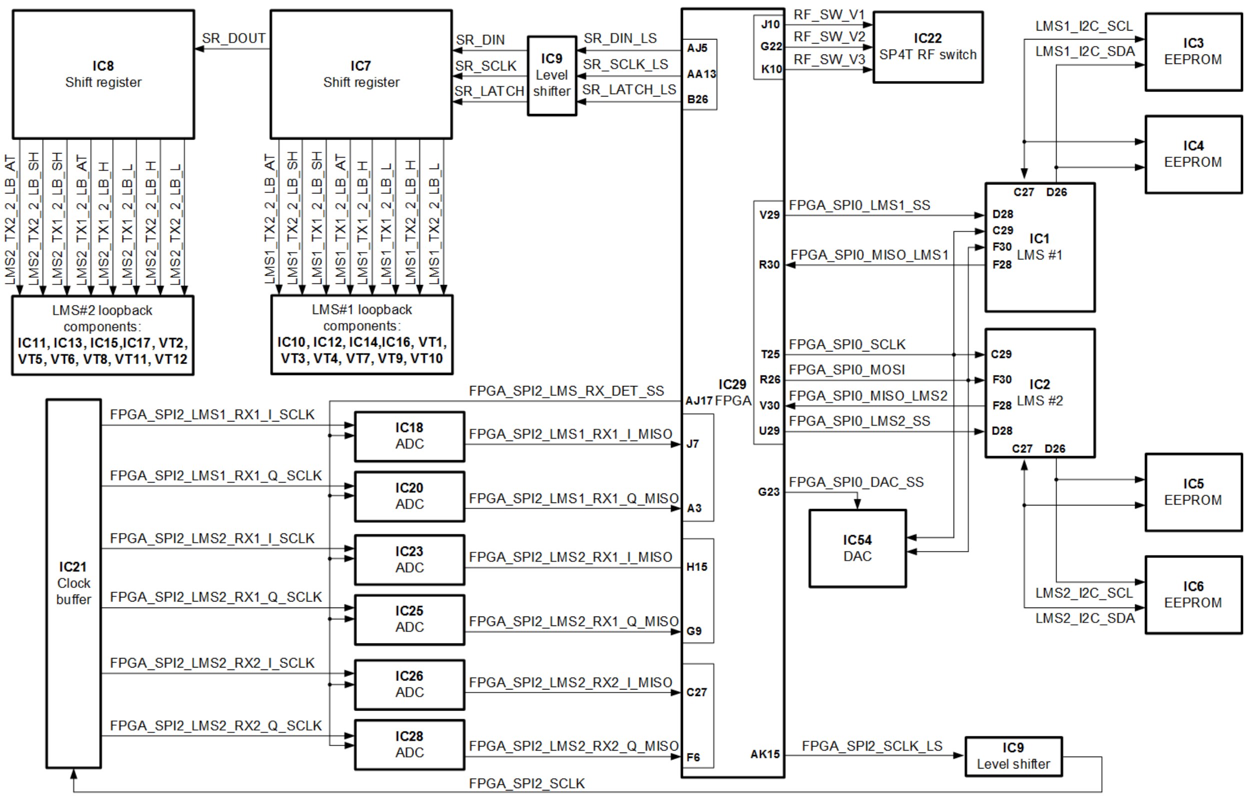


Figure 6 LimeSDR-QPCIe v1.2 FPGA-RF circuit low speed interface block diagram

pLimeSDR-QPCIe board peripherals can be controlled via USB interface. All commands that comes from USB are firstly processed by FX3 controller. I2C and multiple SPI interfaces provide connection to various on-board ICs, such as temperature sensor, port expander, clock generator, memory and real-time clock (RTC).

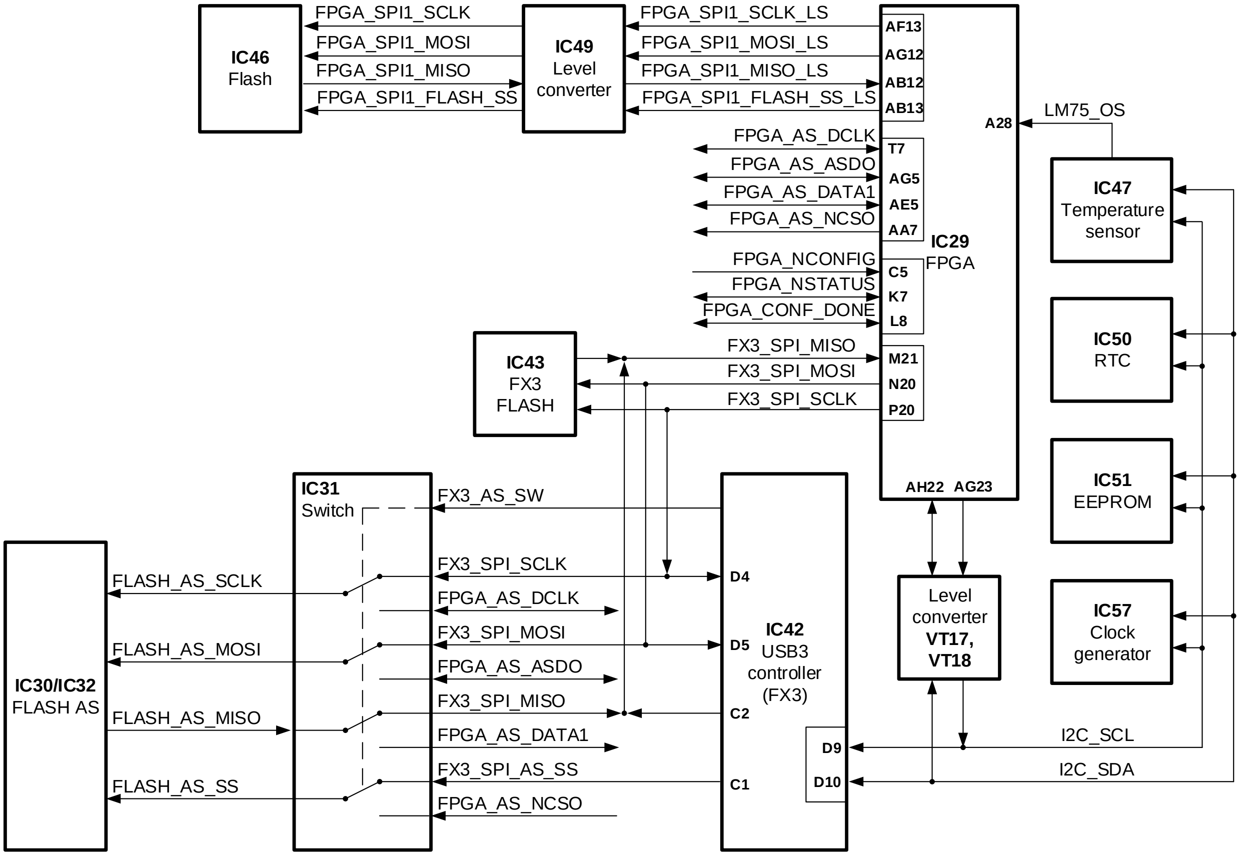


Figure 7 LimeSDR-QPCIe v1.2 FPGA-FX3 low speed interface block diagram

2.2.4.1 SPI interfaces

There are several SPI interfaces with their slave devices:

- **FX3_SPI** - Master of this bus is CYUSB3013 (IC42) and this bus has these slave devices:
  - Flash memory M25P40 (IC43) dedicated for FX3 firmware;
  - Flash memory W25Q128 (IC30) or S25FL128 (IC32) dedicated for FPGA configuration file**.** Using switch (IC31) flash memory is switched from FPGA to FX3_SPI BUS. Then flash content is updated and flash memory is switched back to FPGA. This is done when it is needed to update FPGA gateware in flash memory.
  - FPGA (IC29) - If SPI slave is implemented in FPGA logic it can be accessed through FX3_SPI interface.

Table 17 FX3 SPI interface

FX3 pin (IC42)	FX3 chip reference (IC42)	Schematic signal name	FPGA pin (IC29)	I/O standard	Comment
D4	SPI_SCK	FX3_SPI_SCLK	P20	3.3V	
D5	SPI_MOSI	FX3_SPI_MOSI	N20	3.3V	
C2	SPI_MISO	FX3_SPI_MISO	M21	3.3V	

F5	GPIO[46]	FX3_SPI_FPGA_SS	AG22	3.3V	
F2	GPIO[45]	FX3_SPI_AS_SS	-	3.3V	
C1	SPI_SSN	FX3_SPI_FLASH_SS	-	3.3V	

- **FPGA_SPI0 –** master of this interface is FPGA (IC29), slave devices of this interface:
  - RFIC LMS7002M (IC1)
  - RFIC LMS7002M (IC2)
  - 14-bit ADC ADS4246 (IC37)
  - XO VC DAC AD5662 (IC54)
  - Phase detector ADF4002 (IC53)

Schematic signal name	FPGA pin (IC29)	I/O standard	Comment
FPGA_SPI0_SCLK	T25	2.5V(3.3V)	
FPGA_SPI0_MOSI	R26	2.5V(3.3V)	
FPGA_SPI0_MISO_LMS1	R30	2.5V(3.3V)	
FPGA_SPI0_MISO_LMS2	V30	2.5V(3.3V)	
FPGA_SPI0_MISO_ADC	L20	1.8V	
FPGA_SPI0_LMS1_SS	V29	2.5V(3.3V)	
FPGA_SPI0_LMS2_SS	U29	2.5V(3.3V)	
FPGA_SPI0_ADC_SS	E26	1.5V	Configure FPGA output as Open-Drain
FPGA_SPI0_DAC_SS	G23	1.5V	Configure FPGA output as Open-Drain
FPGA_SPI0_ADF_SS	E25	1.5V	Configure FPGA output as Open-Drain

- **FPGA_SPI1**: master of this interface is FPGA (IC29), slave devices of this interface:
  - Flash memory M25P40 (IC46)

Table 18 FPGA_SPI1 interface

Schematic signal name	FPGA pin (IC29)	I/O standard	Comment
FPGA_SPI1_SCLK_LS	AF13	1.5V	Connected through voltage level shifter. Net label on memory side FPGA_SPI1_SCLK
FPGA_SPI1_MOSI_LS	AG12	1.5V	Connected through voltage level shifter. Net label on memory side FPGA_SPI1_MOSI
FPGA_SPI1_MISO_LS	AB12	1.5V	Connected through voltage level shifter. Net label on memory side FPGA_SPI1_MISO

FPGA_SPI1_FLASH_SS_LS	AB13	1.5V	Connected through voltage level shifter. Net label on memory side FPGA_SPI1_FLASH_SS

- **FPGA_SPI2:** master of this interface is FPGA (IC29), slave devices of this interface:
  - 6x ADC (IC18, IC20, IC23, IC25, IC26, IC28) dedicated for onboard RSSI use.

Table 19 FPGA_SPI2 interface

Schematic signal name	FPGA pin (IC29)	I/O standard	Comment
FPGA_SPI2_SCLK_LS	AK15	1.5V	Connected through voltage level shifter and clock buffer to all six ADC. Net label after level shifter FPGA_SPI2_SCLK. Fanouts through clock buffer to nets: FPGA_SPI2_LMS1_RX1_I_SCLK FPGA_SPI2_LMS1_RX1_Q_SCLK FPGA_SPI2_LMS2_RX2_I_SCLK FPGA_SPI2_LMS2_RX1_Q_SCLK FPGA_SPI2_LMS2_RX1_I_SCLK FPGA_SPI2_LMS2_RX2_Q_SCLK
FPGA_SPI2_LMS1_RX1_I_MISO	J7	1.5V	
FPGA_SPI2_LMS1_RX1_Q_MISO	A3	1.5V	
FPGA_SPI2_LMS2_RX1_I_MISO	H15	1.5V	
FPGA_SPI2_LMS2_RX1_Q_MISO	G9	1.5V	
FPGA_SPI2_LMS2_RX2_I_MISO	C27	1.5V	
FPGA_SPI2_LMS2_RX2_Q_MISO	F6	1.5V	
FPGA_SPI2_LMS_RX_DET_SS	AJ17	1.5V	

2.2.4.2 I2C interfaces

Board has three independent I2C interfaces: I2C, LMS1_I2C and LMS2_I2C.

I2C – master of this interface can be either FX3 (IC42) or either FPGA (IC29). Master selection is done through R287 and R288 resistors. By default, master is FX3 (resistors fitted). To select FPGA as master, remove R287 and R288 resistors. This interface has several slave devices which are listed in Table 20.

- RTC DS3231 (IC50);
- EEPROM M24128 (IC51);
- Temperature sensor LM75 (IC47), EEPROM and clock generator. Information for slave devices are provided in Table 20, signal connectivity information is in Table 21.

Table 20. I2C interface devices

I2C slave device	Slave device	I2C address	I/O standard	Comment
IC47	Temperature sensor	1 0 0 1 0 0 0 RW	3.3V	LM75
IC51	EEPROM	1 0 1 0 0 0 0 RW	3.3V	M24128
IC57	Clock generator	1 1 0 0 0 0 0 RW	3.3V	Si5351C
IC50	Real-time clock (RTC)	1 1 0 1 0 0 0 RW	3.3V	DS3231

Table 21 I2C interface pins

Schematic signal name	FPGA pin (IC29)	FX3 pin (IC42)	I/O standard	Comment
I2C_SDA_LS	AH22	-	1.5V	Connected through level converter to I2C_SDA. Remove R287 and R288 to use FPGA as I2C master.
I2C_SCL_LS	AG23	-	1.5V	Connected through level converter to I2C_SCL. Remove R287 and R288 to use FPGA as I2C master.
FX3_I2C_SDA	-	D9	3.3V	Connected through R287 to I2C_SDA.
FX3_I2C_SCL	-	D10	3.3V	Connected through R288 to I2C_SCL.

- **LMS1_I2C:** this interface has two EEPROMs. This interface is only accessible from LMS7002M (IC1). In Table 22 are listed all LMS1_I2C slave devices and their information. In Table 23 listed pin connections.

Table 22. LMS1_I2C interface devices

I2C slave device	Slave device	I2C address	I/O standard	Comment

IC3	EEPROM for LMS7 MCU firmware	1 0 1 0 A2 A1 A0 RW	2.5V(3.3V)	M24128
IC4	EEPROM	1 0 1 0 A2 A1 A0 RW	2.5V(3.3V)	24FC512

Table 23 LMS1_I2C pins

Schematic signal name	RFIC pin (IC1)	I/O standard	Comment
LMS1_I2C_SDA	D26	2.5V(3.3V)	Accessible only from IC1
LMS1_I2C_SCL	C27	2.5V(3.3V)	Accessible only from IC1

- **LMS2_I2C:** this interface has two EEPROMs. This interface is only accessible from LMS7002M (IC2). In Table 24 are listed all LMS1_I2C slave devices and their information. In Table 25 listed pin connections.

Table 24. LMS2_I2C interface devices

I2C slave device	Slave device	I2C address	I/O standard	Comment
IC5	EEPROM for LMS7 MCU firmware	1 0 1 0 A2 A1 A0 RW	2.5V(3.3V)	M24128
IC6	EEPROM	1 0 1 0 A2 A1 A0 RW	2.5V(3.3V)	24FC512

Table 25 LMS2_I2C pins

Schematic signal name	RFIC pin (IC2)	I/O standard	Comment
LMS2_I2C_SDA	D26	2.5V(3.3V)	Accessible only from IC2
LMS2_I2C_SCL	C27	2.5V(3.3V)	Accessible only from IC2

2.2.4.3 UART

UART interface is dedicated for communication between FPGA and GNSS receiver GPS/GNSS M0578 (IC48).

Table 26 GNSS UART interface pins

Schematic signal name	FPGA pin (IC29)	I/O standard	Comment
GNSS_RX	K25	3.3V(2.5V)	
GNSS_TX	K22	3.3V(2.5V)	

2.2.5 User I/O

This chapter describes available inputs and outputs of LimeSDR-QPCIe v1.2 board which can be used for user applications.

2.2.5.1 FPGA switch

4 poles slide switch SW2 is connected to FPGA and can be used to implement additional functionality which requires input control. Each switch line has external pull up resistors. When switch is in position "On", it pulls down the line to logic '0' level.

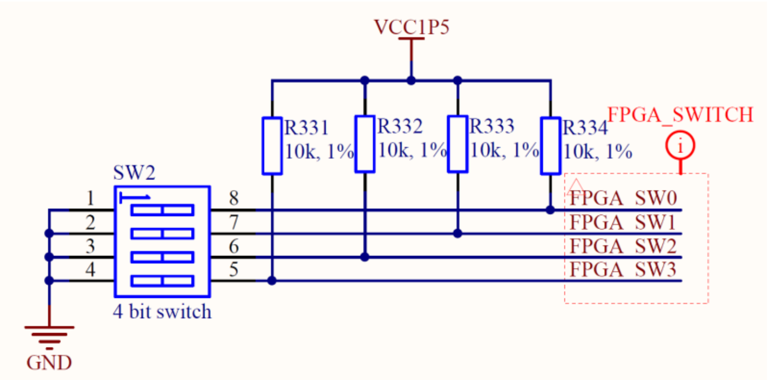


Figure 8 LimeSDR-QPCIe v1.2 4 poles slide switch

In **Error! Reference source not found.** are listed each switch line and correspond FPGA pins.

Table 27. FPGA Switch connections

Switch pole	Schematic signal name	FPGA pin (IC29)	I/O standard
1	FPGA_SW0	H7	1.5V
2	FPGA_SW1	A2	1.5V
3	FPGA_SW2	E6	1.5V
4	FPGA_SW3	F8	1.5V

2.2.5.2 Indication LEDs

LimeSDR-QPCIe board comes with four single colour (green) general purpose LEDs, one general purpose dual colour (red-green) LED, one green LED informing about successful FPGA configuration and one for power indication.

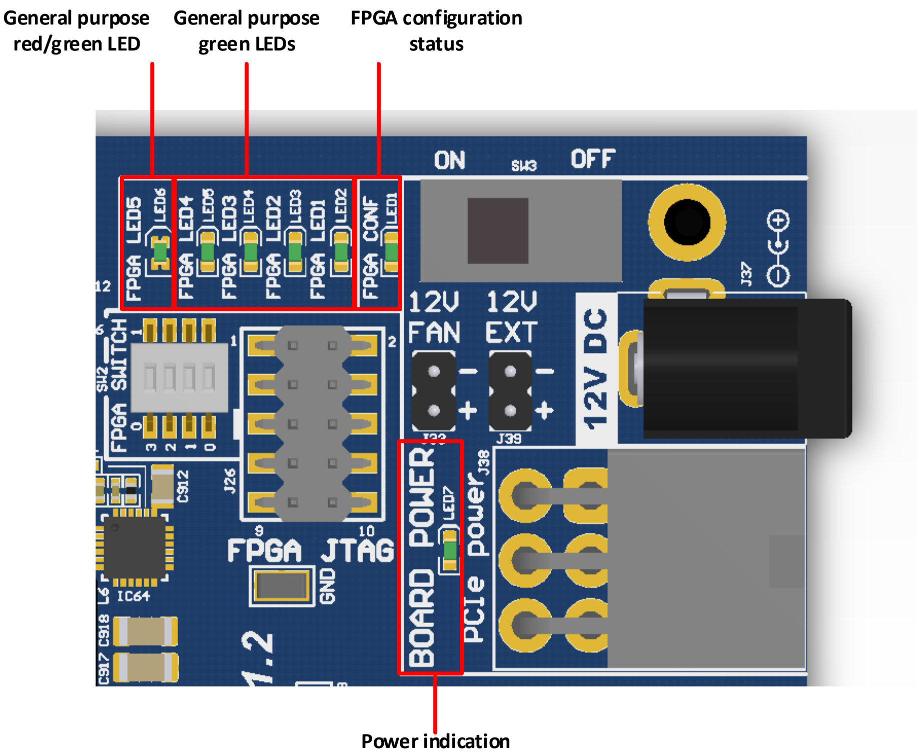


Figure 9 LimeSDR-QPCIe v1.2 indication LEDs

All LEDs are connected to FPGA and their function can be changed except for LED1. Default LEDs functions and other information are listed in the table below.

Table 28 Default functions of LEDs

Board Reference	Schematic name	Board label	Type	Description
LED1	FPGA_CONF_DONE	FPGA CONF	FPGA status	LED lights up after successful FPGA configuration
LED2, LED3, LED4, LED5	FPGA_LED1 FPGA_LED2 FPGA_LED3 FPGA_LED4	FPGA LED1 FPGA LED2 FPGA LED3 FPGA LED4	General purpose	General purpose reconfigurable single colour LED.
LED6	FPGA_LED5_G, FPGA_LED5_R	FPGA LED5	General purpose	General purpose reconfigurable dual colour (red-green) LED.
LED7	VCC3P3, GND	BOARD POWER	Power indication	LED lights up when VCC3P3 power rail is active.

2.2.6 RF Loopback Control

There is RF loopback circuit for RF transceivers which can be controlled from FPGA through shift registers 74HC595 (IC7 and IC8). Pin connection can be found in Table 29.

Table 29 Pin connection of shift registers

Schematic signal name	FPGA pin (IC29)	I/O standard	Comment
SR_SCLK_LS	AA13	1.5V	Connected through level converter to SR_SCLK.
SR_DIN_LS	AJ5	1.5V	Connected through level converter to SR_DIN.
SR_LATCH_LS	B26	1.5V	Connected through level converter to SR_LATCH.

Table 30 LMS#1 loopback control truth table

IC7 pin reference								Result
Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	
x	x	x	x	L	H	x	x	LMS#1 TX1_2 → U.FL (J13)
x	x	x	x	H	L	x	x	LMS#1 TX1_2 → LMS#1 RX1_H
x	x	H	x	x	x	x	x	LMS#1 TX1_2 shunt to ground on
x	x	L	x	x	x	x	x	LMS#1 TX1_2 shunt to ground off
x	x	x	H	x	x	x	x	LMS#1 TX1_2 → RX1_H attenuation -40 dB
x	x	x	L	x	x	x	x	LMS#1 TX1_2 → RX1_H attenuation -25 dB
x	x	x	x	x	x	L	H	LMS#1 TX2_2 → U.FL (J17)
x	x	x	x	x	x	H	L	LMS#1 TX2_2 → LMS#1 RX2_H
H	x	x	x	x	x	x	x	LMS#1 TX2_2 shunt to ground on
L	x	x	x	x	x	x	x	LMS#1 TX2_2 shunt to ground off
x	H	x	x	x	x	x	x	LMS#1 TX2_2 → RX2_H attenuation -40 dB
x	L	x	x	x	x	x	x	LMS#1 TX2_2 → RX2_H attenuation -25 dB

Note 1: Schematic signals corresponds to IC7 pins as below:
Q0 – LMS1_TX2_2_LB_SH Q1 – LMS1_TX2_2_LB_AT Q2 – LMS1_TX1_2_LB_SH Q3 – LMS1_TX1_2_LB_AT
Q4 – LMS1_TX1_2_LB_H Q5 – LMS1_TX1_2_LB_L Q6 – LMS1_TX2_2_LB_H Q7 – LMS1_TX2_2_LB_L

Table 31 LMS#2 sloopback control truth table

IC8 pin reference								Result
Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	
x	x	H	L	x	x	x	x	LMS#2 TX1_2 → U.FL (J14)
x	x	L	H	x	x	x	x	LMS#2 TX1_2 → LMS#2 RX1_H
x	x	x	x	x	H	x	x	LMS#2 TX1_2 shunt to ground on
x	x	x	x	x	L	x	x	LMS#2 TX1_2 shunt to ground off
x	x	x	x	x	x	H	x	LMS#2 TX1_2 → RX1_H attenuation -40 dB
x	x	x	x	x	x	L	x	LMS#2 TX1_2 → RX1_H attenuation -25 dB
H	L	x	x	x	x	x	x	LMS#2 TX2_2 → U.FL (J18)

IC8 pin reference								Result
Q0	Q1	Q2	Q3	Q4	Q5	Q6	Q7	
L	H	x	x	x	x	x	x	LMS#2 TX2_2 → LMS#2 RX2_H
x	x	x	x	x	x	x	H	LMS#2 TX2_2 shunt to ground on
x	x	x	x	x	x	x	L	LMS#2 TX2_2 shunt to ground off
x	x	x	x	H	x	x	x	LMS#2 TX2_2 → RX2_H attenuation -40 dB
x	x	x	x	L	x	x	x	LMS#2 TX2_2 → RX2_H attenuation -25 dB

Note 1: Q0 – LMS2_TX2_2_LB_L Q1 – LMS2_TX2_2_LB_H Q2 – LMS2_TX1_2_LB_L Q3 – LMS2_TX1_2_LB_H
Q4 – LMS2_TX2_2_LB_AT Q5 – LMS2_TX1_2_LB_SH Q6 – LMS2_TX1_2_LB_AT Q7 – LMS2_TX2_2_LB_SH

The table below describes RF transceiver LMS7002#1 and LMS7002#2 external loopback configuration, pins, schematic signal name, FPGA interconnections and I/O standard.

Table 32 LMS7002 (IC1, IC2) external loopback configuration

Loopback for	RF switch	Schematic control signal name	Shift register pin	FPGA-shift register pins	Comment
LMS7002 #1 (IC1)	IC10, IC12	LMS1_TX1_2_LB_L LMS1_TX1_2_LB_H	IC7.Q5 IC7.Q4	AJ5, AA13, B26 (through level shifter IC9)	When LMS1_TX1_2_LB_L is high and LMS1_TX1_2_LB_H is low then LMS#1 TX1_2 (LMS1_LOOPBACK_TX12 net) is fed to U.FL connector J13. When LMS1_TX1_2_LB_L is low and LMS1_TX1_2_LB_H is high then LMS#1 TX1_2 (LMS1_LOOPBACK_TX12 net) is fed to LMS#1 RX1_H (LMS1_LOOPBACK_RX1H net)
	IC14, IC15	LMS1_TX2_2_LB_L LMS1_TX2_2_LB_H	IC7.Q7 IC7.Q6		When LMS1_TX2_2_LB_L is high and LMS1_TX2_2_LB_H is low then LMS#1 TX2_2 (LMS1_LOOPBACK_TX22 net) is fed to U.FL connector J17. When LMS1_TX2_2_LB_L is low and LMS1_TX2_2_LB_H is high then LMS#1 TX2_2 (LMS1_LOOPBACK_TX22 net) is fed to LMS#1 RX2_H (LMS1_LOOPBACK_RX2H net)

	VT3, VT9	LMS1_TX1_2_LB_SH	IC7.Q2		Loopback shunt to ground for high isolation. When high – shunt active, when low – shunt off.
	VT1	LMS1_TX1_2_LB_AT	IC7.Q0		When LMS1_TX1_2_LB_AT is low, TX1_2 to RX1_H loopback attenuation is set to -40 dB. When LMS1_TX1_2_LB_AT is high, loopback attenuation is set to -25 dB.
	VT7	LMS1_TX2_2_LB_AT	IC7.Q1		When LMS1_TX2_2_LB_AT is low, TX2_2 to RX2_H loopback attenuation is set to -40 dB. When LMS1_TX2_2_LB_AT is high, loopback attenuation is set to -25 dB.
LMS7002 #2 (IC2)	IC11, IC13	LMS2_TX1_2_LB_L LMS2_TX1_2_LB_H	IC8.Q2 IC8.Q3		When LMS2_TX1_2_LB_L is high and LMS2_TX2_2_LB_H is low then LMS#2 TX1_2 (LMS2_LOOPBACK_TX12 net) is fed to U.FL connector J14. When LMS2_TX1_2_LB_L is low and LMS2_TX1_2_LB_H is high then LMS#2 TX1_2 (LMS2_LOOPBACK_TX12 net) is fed to LMS#2 RX1_H (LMS2_LOOPBACK_RX1H net)
	IC15, IC17	LMS2_TX2_2_LB_L LMS2_TX2_2_LB_H	IC8.Q0 IC8.Q1		When LMS2_TX2_2_LB_L is high and LMS2_TX2_2_LB_H is low then LMS#2 TX2_2 (LMS2_LOOPBACK_TX22 net) is fed to U.FL connector J18. When LMS2_TX2_2_LB_L is low and LMS2_TX2_2_LB_H is high then LMS#2 TX2_2 (LMS2_LOOPBACK_TX22 net) is fed to LMS#2 RX2_H (LMS2_LOOPBACK_RX2H net)

	VT11, VT12	LMS2_TX2_2_LB_SH	IC8.Q7		Loopback shunt to ground for high isolation. When high – shunt active, when low – shunt off.
	VT2	LMS2_TX1_2_LB_AT	IC8.Q6		When LMS2_TX1_2_LB_AT is low, TX1_2 to RX1_H loopback attenuation is set to -40 dB. When LMS2_TX1_2_LB_AT is high, loopback attenuation is set to -25 dB.
	VT8	LMS2_TX2_2_LB_AT	IC8.Q4		When LMS2_TX2_2_LB_AT is low, TX2_2 to RX2_H loopback attenuation is set to -40 dB. When LMS2_TX2_2_LB_AT is high, loopback attenuation is set to -25 dB.

2.2.7 Board Temperature Control

LimeSDR-QPCIe has integrated temperature sensor (IC47) which controls FAN to keep board in operating temperature range. FAN has dedicated holes for mounting over the main digital ICs (FPGA, DDRs, DACs and ADC) and must be connected to J33 (0.1” pitch) connector.

Fan will be turned on when board heats up to 55°C and will be turned off when the temperature reduces to 45°C.

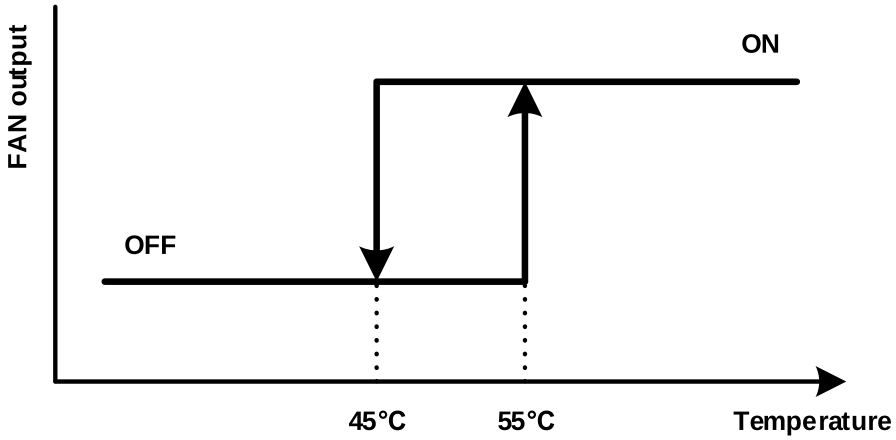
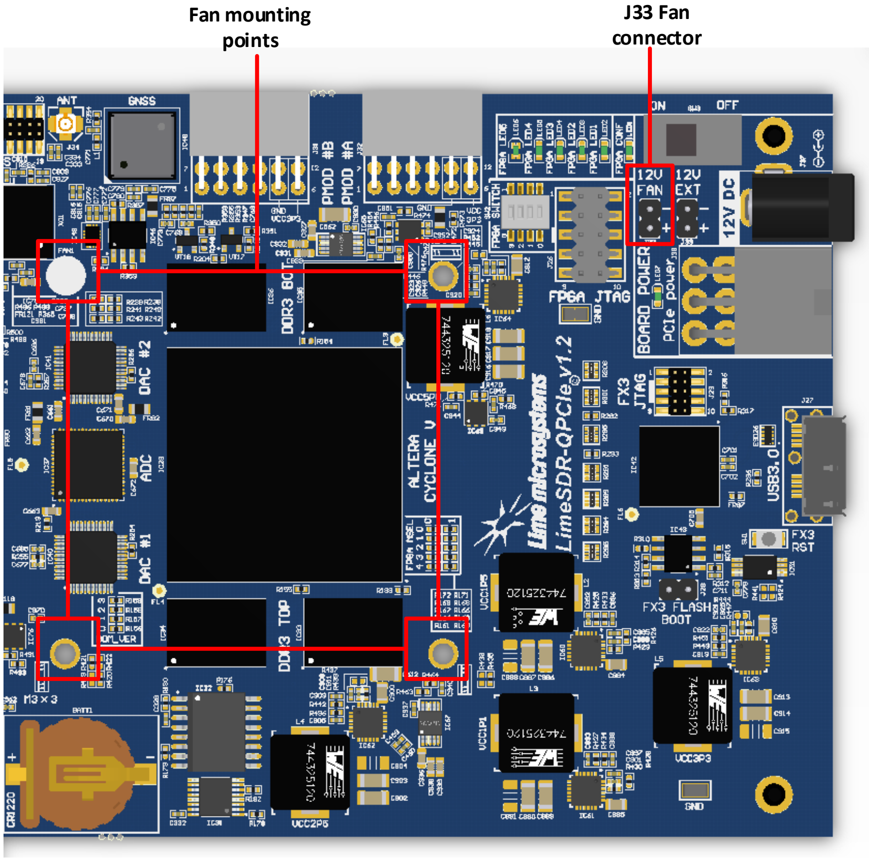


Figure 10 FAN control temperature hysteresis

Measured temperature value can read by using LimeSuiteGUI as described in chapter “3.13 Reading Board Temperature”.

LimeSDR-QPCIe board comes with a dedicated 60mm DC FAN mounting space. Three M3 exposed copper holes (connected to board GND plane) and a space for a standoff are provided and are shown in *Figure* 11. The hole centre to hole centre distance is 50mm.The FAN is controlled via J33 (0.1” pitch) connector which can provide either 12V (default) or a 5V supply rail. The gate of the FAN driving MOSFET is connected to IC29 (FPGA) pin J22 (2.5V/3.3V bank).


Fan mounting points

J33 Fan connector

Figure 11 LimeSDR-QPCIe v1.2 Dedicated FAN mounting space

2.2.8 Clock Distribution

LimeSDR-QPCIe board clock distribution block diagram is presented in Figure 12.

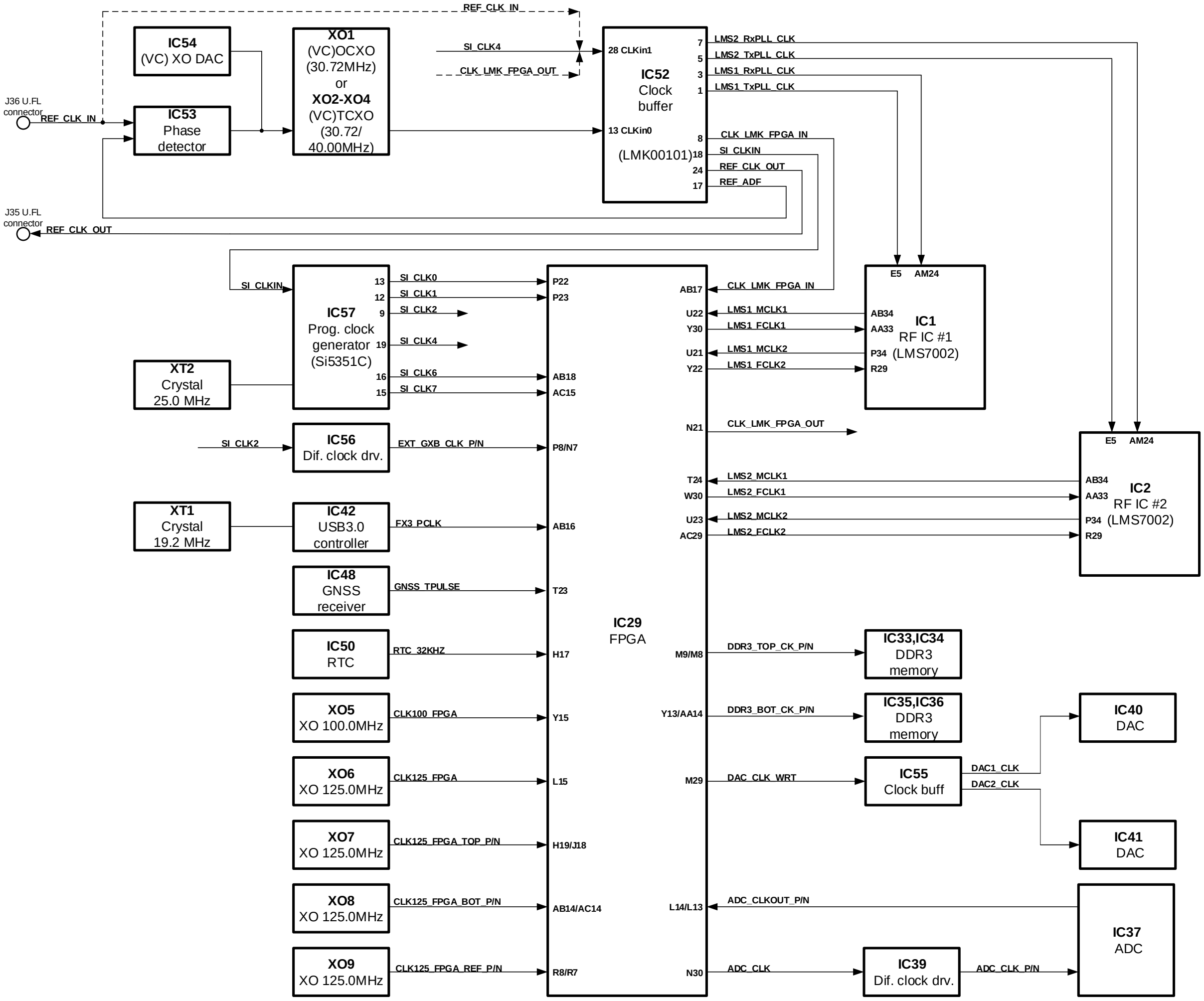


Figure 12 LimeSDR-QPCIe v1.2 board clock distribution block diagram

2.2.8.1 Main clock sources

There are various crystal oscillators with various frequencies mounted on LimeSDR-QPCIe board. The programmable clock generator IC37 (Si5351C [link]) can generate any reference clock frequency, starting from 8 kHz – 160 MHz, for FPGA and LMS PLLs. A real-time clock (RTC) chip (IC50) is also included on the LimeSDR-QPCIe board. The output of IC50 is connected to the FPGA pin H17. Main clock sources and destinations are listed in Table 33.

Table 33. LimeSDR-QPCIe clock sources and destinations

Src.	Dest.	Schematic net name	Freq.	I/O standard	FPGA pin	Description
	FPGA	SI_CLK0	8kHz-160MHz	2.5/3.3V	P22	

Src.	Dest.	Schematic net name	Freq.	I/O standard	FPGA pin	Description
Clk. Gen. (IC57)		SI_CLK1	8kHz-160MHz	2.5/3.3V	P23	
		EXT_GXB_CLK_P	8kHz-160MHz	LVDS	P8	Single-ended converted to differential through IC56
		EXT_GXB_CLK_N	8kHz-160MHz		N7	
	Clk. Buff. (IC52)	SI_CLK4	8kHz-160MHz	-	-	Can be selected as IC52 CLKin1 clock source
	FPGA	SI_CLK6	8kHz-160MHz	1.5V	AB18	
		SI_CLK7	8kHz-160MHz	1.5V	AC15	
FX3 (IC42)	FPGA	FX3_PCLK	66MHz, 100MHz	1.5V	AB16	
GNSS (IC48)	FPGA	GNSS_TPULSE	*	3.3V/2.5V	T23	
RTC (IC50)	FPGA	RTC_32KHZ	32kHz	1.5V	H17	
XO5	FPGA	CLK100_FPGA	100MHz	1.5V	Y15	
XO6	FPGA	CLK125_FPGA	125MHz	1.5V	L15	
XO7	FPGA	CLK125_FPGA_TOP_P	125MHz	LVDS	H19	
		CLK125_FPGA_TOP_N			J18	
XO8	FPGA	CLK125_FPGA_BOT_P	125MHz	LVDS	AB14	
		CLK125_FPGA_BOT_N			AC14	
XO9	FPGA	CLK125_FPGA_REF_P	125MHz	LVDS	R8	
		CLK125_FPGA_REF_N			R7	
FPGA	DAC (IC40, IC41)	DAC_CLK_WRT	*	2.5/3.3V	M29	Goes through clock buffer (IC21)
Clk. Buff. (IC52)	FPGA	CLK_LMK_FPGA_IN	*	1.5V	AB17	
RFIC (IC1)	FPGA	LMS1_MCLK1	*	2.5/3.3V	U22	
		LMS1_MCLK2	*	2.5/3.3V	U21	
FPGA	RFIC (IC1)	LMS1_FCLK1	*	2.5/3.3V	Y30	
		LMS1_FCLK2	*	2.5/3.3V	Y22	
FPGA	Clk. Buff. (IC52)	LMK_CLK_FPGA_OUT	*	2.5/3.3V	N21	
	FPGA	LMS2_MCLK1	*	2.5/3.3V	T24	

Src.	Dest.	Schematic net name	Freq.	I/O standard	FPGA pin	Description
RFIC (IC2)		LMS2_MCLK2	*	2.5/3.3V	U23	
FPGA	RFIC (IC2)	LMS2_FCLK1	*	2.5/3.3V	W30	
		LMS2_FCLK2	*	2.5/3.3V	AC29	
FPGA	DDR3 (IC33, IC34)	DDR3_TOP_CK_P	*	Differential 1.5-V SSTL Class I	M9	
		DDR3_TOP_CK_N	*		M8	
FPGA	DDR3 (IC35, IC36)	DDR3_BOT_CK_P	*	Differential 1.5-V SSTL Class I	Y13	
		DDR3_BOT_CK_N	*		AA14	
FPGA	DAC (IC40, IC41)	DAC_CLK_WRT	*	3.3V	M29	Feeds both DAC through clock buffer (IC55)
ADC (IC37)	FPGA	ADC_CLKOUT_P	*	LVDS	L14	
		ADC_CLKOUT_N	*		L13	
FPGA	ADC (IC37)	ADC_CLK	*	3.3/2.5V	N30	Converted from single eneded to LVDS through IC39

* - depends from design

2.2.8.2 Clock buffer source selection

Clock buffer (IC52) presented in Figure 12 provides clock signals for following components:

- LMS7002 transceivers (IC1, IC2);
- FPGA (IC29) pin AB17;
- Phase detector (IC53);
- Clock generator (IC57);
- U.FL connector J35 (REF CLK OUT label on board). To use this output 0R resistor R378 has to be fitted.

For Clock buffer (IC52) there are two possible clock sources (CLKin0 and CLKin1). Source is selected via R375 resistor.

CLKin0 – to select this input as a source for Clock buffer (IC52) R375 resistor has to be removed (removed by default). As a source for this input one of the following high-precision crystal oscillators can be selected:

1. XO1 – 30.72 MHz VCOCXO (precision: ±20 ppb stable)

2. XO2 or XO3 – 30.72 MHz VCTCXO (precision: ±1 ppm initial, ±4 ppm stable);
3. XO4 – 40 MHz VCTCXO (precision: ±1 ppm initial).

IC52 buffer CLKin0 clock source is selected by one of the 0402 size 0R resistor combinations, required modifications can be found in Table 34.

Table 34 Crystal oscillator selection for clock buffer (IC52)

Source for clock buffer (IC52) CLKin0 input	0R Fit	0R NF	Comment
XO1	R364, R365	R379, R380, R382, R384	Default selection
XO2, X03	R379, R380	R364, R365, R382, R384	
XO4	R382, R384	R364, R365, R379, R380	

CLKin1 – to select this input as a source for Clock buffer (IC52) R375 resistor has to be fitted (removed by default). As a source for this input one of the following sources can be selected:

1. Clock generator (IC57)
2. U.FL Connector (J36)
3. FPGA (IC29) output pin N21

Table 35 Source for clock buffer (IC52) CLKin1 input selection

Source for clock buffer (IC52) CLKin1 input	Schematic net name	0R Fit	0R NF	Comment
Clock generator (IC57) output CLK4	SI_CLK4	R374	R368, R372	Default selection
J36 U.FL Connector	REF_CLK_IN	R372	R368, R374, R389	
FPGA (IC29) output pin N21	CLK_LMK_FPGA_OUT	R368	R372, R374	

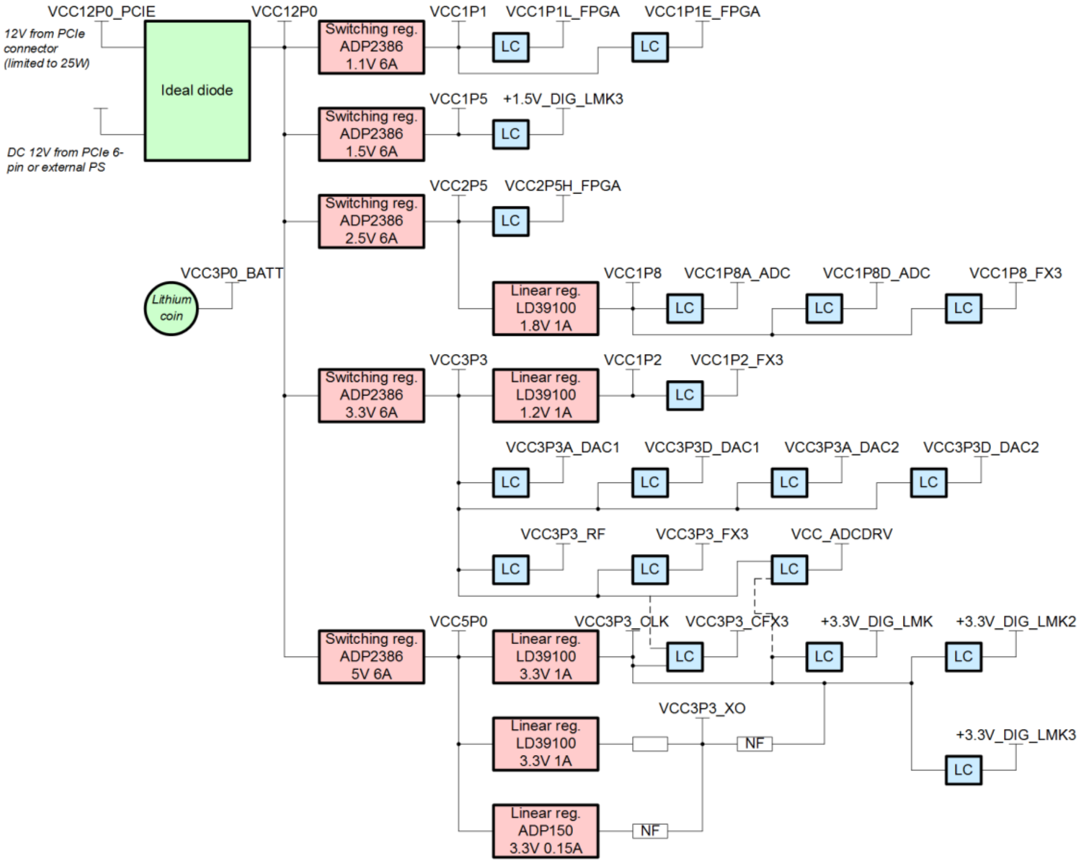
2.2.8.3 VCTCXO clock tuning

VCTCXO can be tuned by on-board phase detector (IC53, ADF4002 [link]) or by 16-bit DAC (IC54). The on-board phase detector is used to synchronize on-board VCTCXO with external equipment (via J36 U.FL connector) to calibrate frequency error. At the same time only ADF or DAC can control VCTCXO. Both ADF and DAC are connected to FPGA_SPI0 interface. For details see chapter **2.2.4.1 SPI interfaces**. With valid configuration selection between ADF and DAC is done automatically. When board is powered, by default VCTCXO is controlled by DAC.

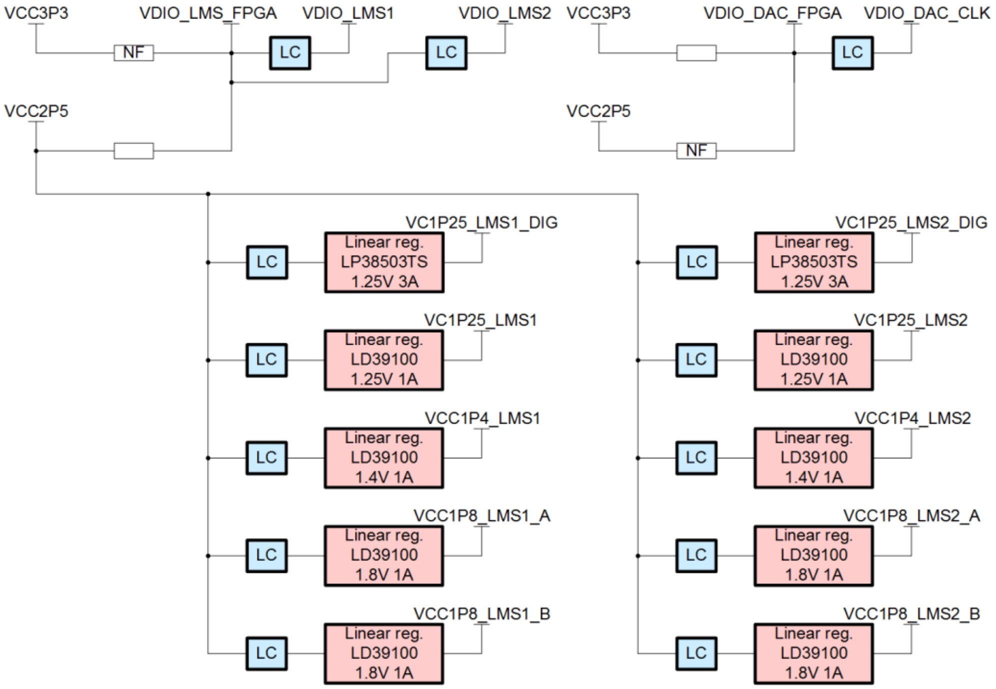
2.2.9 Power Distribution

LimeSDR-QPCIe board can be powered from several sources. The first power supply source option is 12V DC through a 2.5mm centre positive barrel connector. The second one is through a standard 6-pin PCIe power connector J38 (0.165" pitch). The last supply source option is the board edge PCIe connector.

LimeSDR-QPCIe board has complex power delivery network consisting of many different power rails with different voltages, filters, power sequences. LimeSDR-QPCIe board power distribution block diagram is presented in *Figure* 13 in two parts.


VCC12P0_PCIE
12V from PCIe connector (limited to 25W)
Ideal diode
DC 12V from PCIe 6-pin or external PS
VCC12P0
Switching reg. ADP2386 1.1V 6A
VCC1P1
VCC1P1L_FPGA
VCC1P1E_FPGA
LC
Switching reg. ADP2386 1.5V 6A
VCC1P5
+1.5V_DIG_LMK3
Switching reg. ADP2386 2.5V 6A
VCC2P5
VCC2P5H_FPGA
VCC3P0_BATT
Lithium coin
Linear reg. LD39100 1.8V 1A
VCC1P8
VCC1P8A_ADC
VCC1P8D_ADC
VCC1P8_FX3
Switching reg. ADP2386 3.3V 6A
VCC3P3
Linear reg. LD39100 1.2V 1A
VCC1P2
VCC1P2_FX3
VCC3P3A_DAC1
VCC3P3D_DAC1
VCC3P3A_DAC2
VCC3P3D_DAC2
VCC3P3_RF
VCC3P3_FX3
VCC_ADCDRV
Switching reg. ADP2386 5V 6A
VCC5P0
Linear reg. LD39100 3.3V 1A
VCC3P3_CLK
VCC3P3_CFX3
+3.3V_DIG_LMK
+3.3V_DIG_LMK2
VCC3P3_XO
NF
+3.3V_DIG_LMK3
Linear reg. LD39100 3.3V 1A
Linear reg. ADP150 3.3V 0.15A
NF

a



b

Figure 13 LimeSDR-QPCIe v1.2 board power distribution block diagrams

Power network power circuit ICs are presented in Figure 14 and Figure 15.

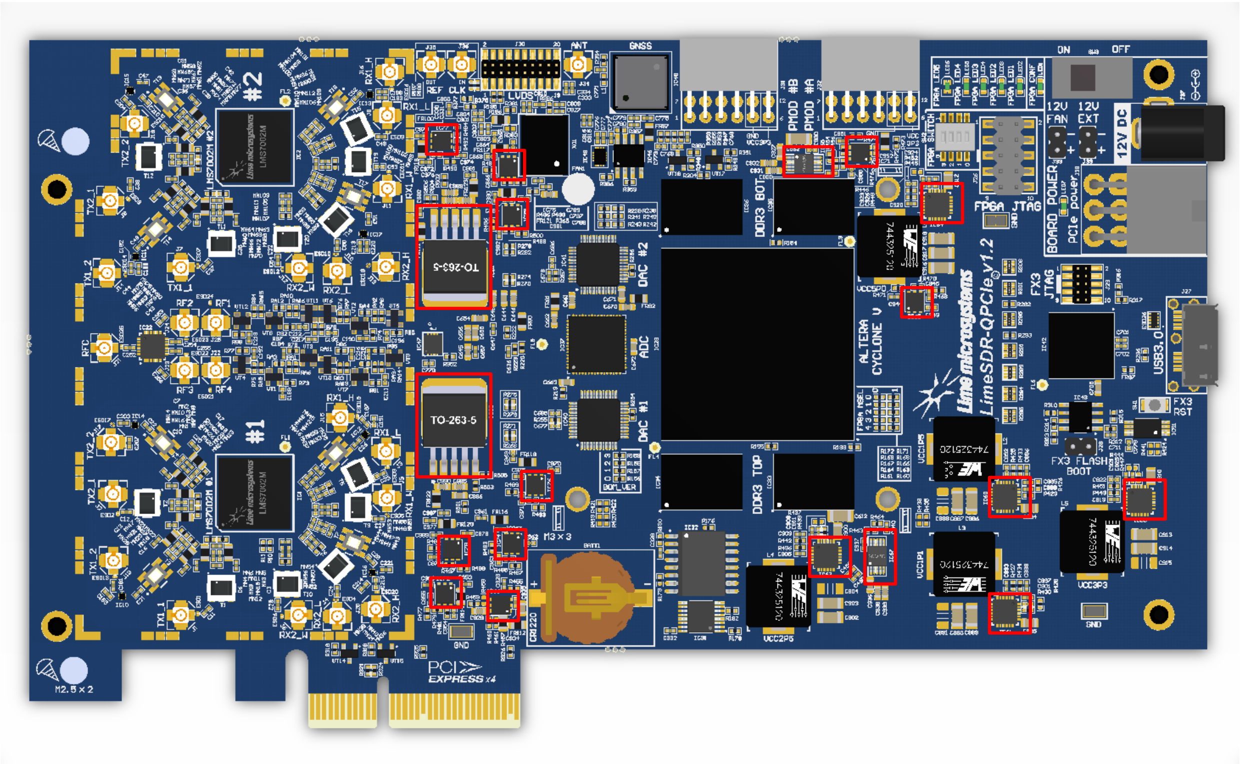


Figure 14 LimeSDR-QPCIe v1.2 board power ICs on TOP side

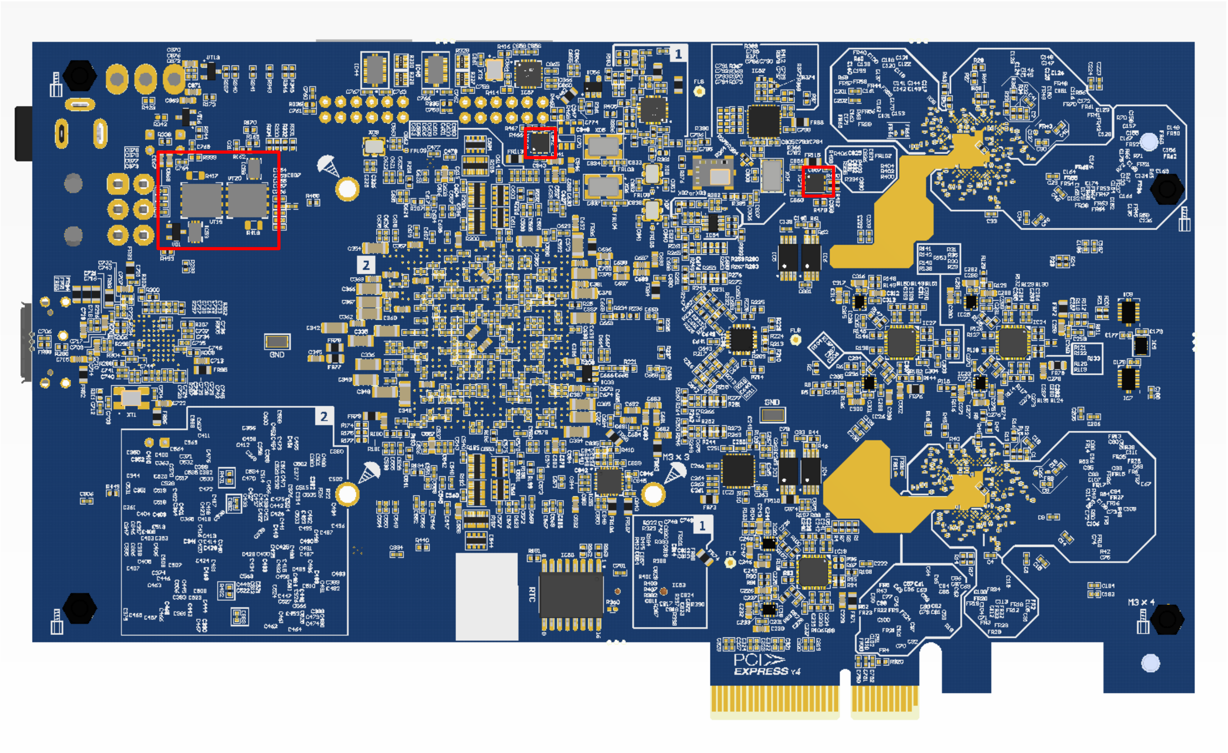


Figure 15 LimeSDR-QPCIe v1.2 board power ICs on BOTTOM side

3. Getting Started with LimeSDR-USB

3.1 Launching LimeSuiteGUI and Connecting to the LimeSDR-USB Board

First of all, connect LimeSDR-USB board to PC USB3.0 socket. Please go the section *4 "Drivers Installation"* to see how to install OS drivers so your PC can see LimeSDR-USB board. In the provided USB Flash there is a folder GUI in which you will find *LimeSuiteGUI.exe* file. Open it.

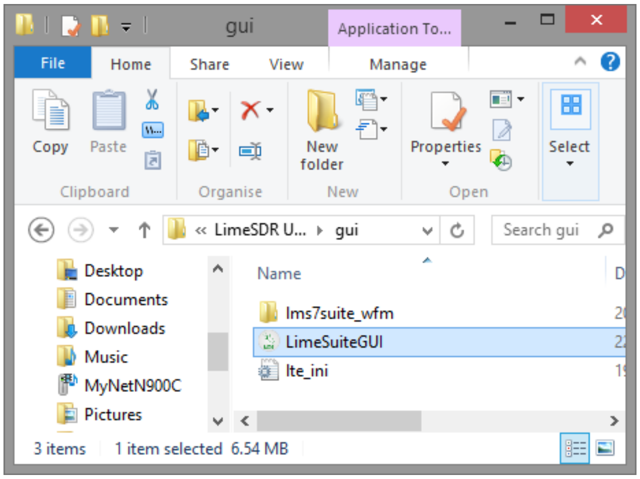


Figure 16 Opening LimeSuiteGUI.exe file

To launch LimeSuiteGUI application go to menu and select: Options -> Connection Settings as shown in *Figure 17*.

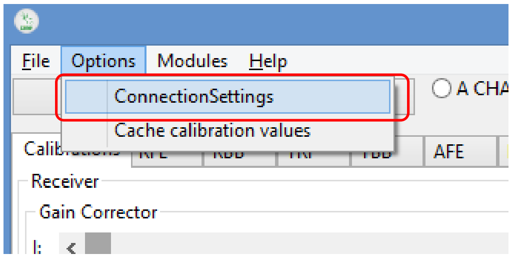


Figure 17 Opening connection settings

Select **LimeSDR-USB** board as shown in Figure 18 and press Connect.

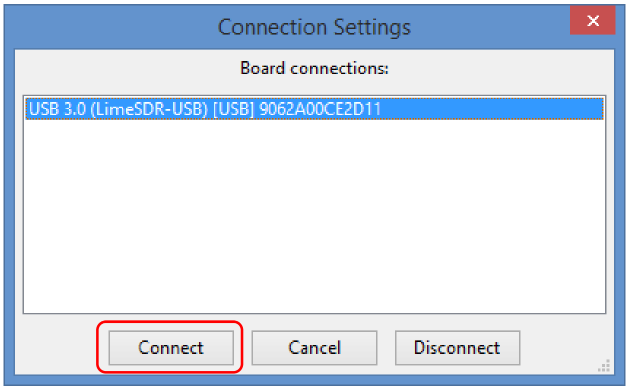


Figure 18 LimeSuiteGUI select ports

3.2 Loading and Saving Register Settings

In order to load settings, click button Open as shown in Figure 19.

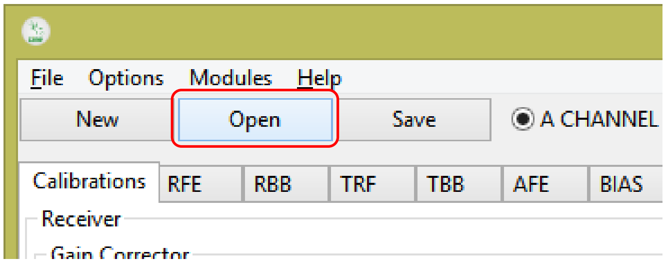


Figure 19 Opening LimeSuiteGUI settings file

Select .ini setting file and click Open as shown in Figure 20.

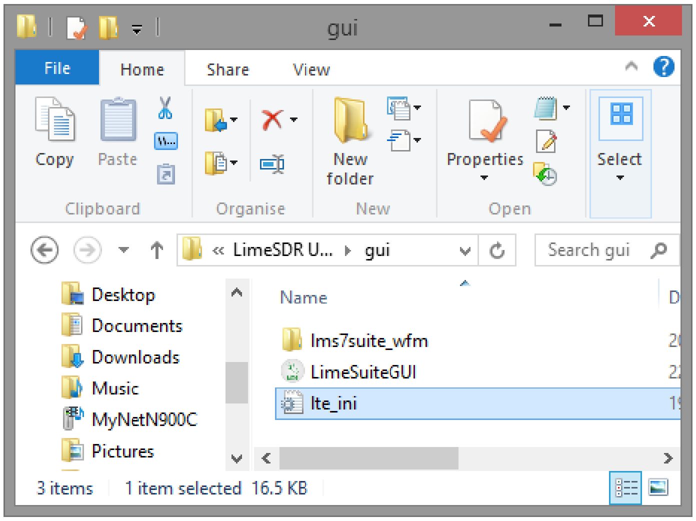


Figure 20 Selecting LimeSuiteGUI settings file

Then click GUI --> Chip button as shown in Figure 21.

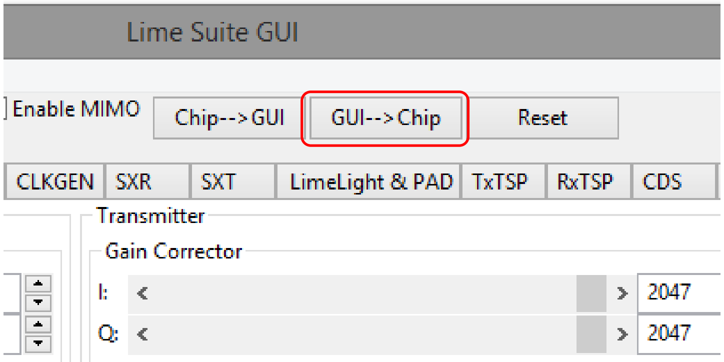


Figure 21 Sending LMS7002M settings from GUI to LimeSDR-USB board

If you want load all LMS7002M settings from LimeSDR-USB to GUI, then click button Chip-->GUI as shown in Figure 22.

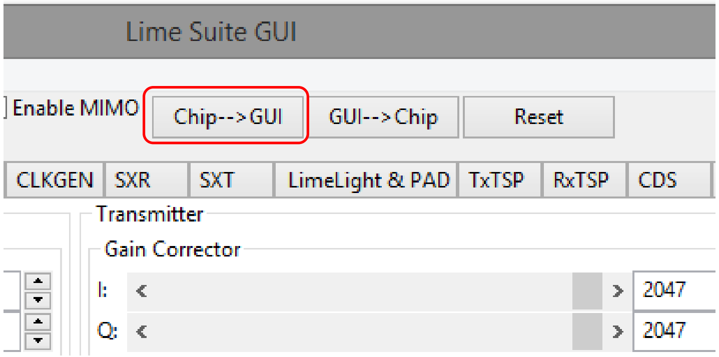
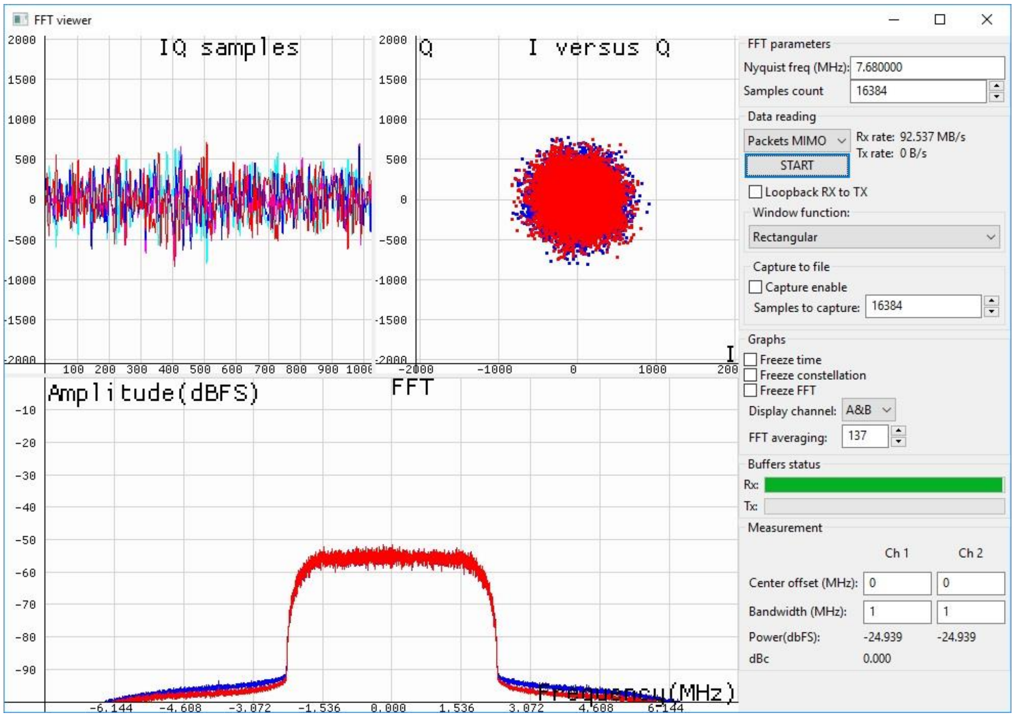


Figure 22 Sending LMS7002M settings from LimeSDR-USB to GUI

3.3 Quick Test

If there is a need to check if the board is fully working you can run very simple and quick board test. All instruction on how to do it you can find online here. One you see the graph as it show in Figure 23 of W-CDMA signal on FFT Viewer, you know that the board is working.



FFT parameters
Nyquist freq (MHz): 7.680000
Samples count: 16384

Data reading
Packets MIMO
Rx rate: 92.537 MB/s
Tx rate: 0 B/s
□ Loopback RX to TX
Window function: Rectangular

Capture to file
□ Capture enable
Samples to capture: 16384

Graphs
□ Freeze time
□ Freeze constellation
□ Freeze FFT
Display channel: A&B
FFT averaging: 137

Buffers status
Rx: 🟩
Tx:

Measurement

	Ch 1	Ch 2
Center offset (MHz):	0	0
Bandwidth (MHz):	1	1
Power(dbFS):	-24.939	-24.939
dBc	0.000	

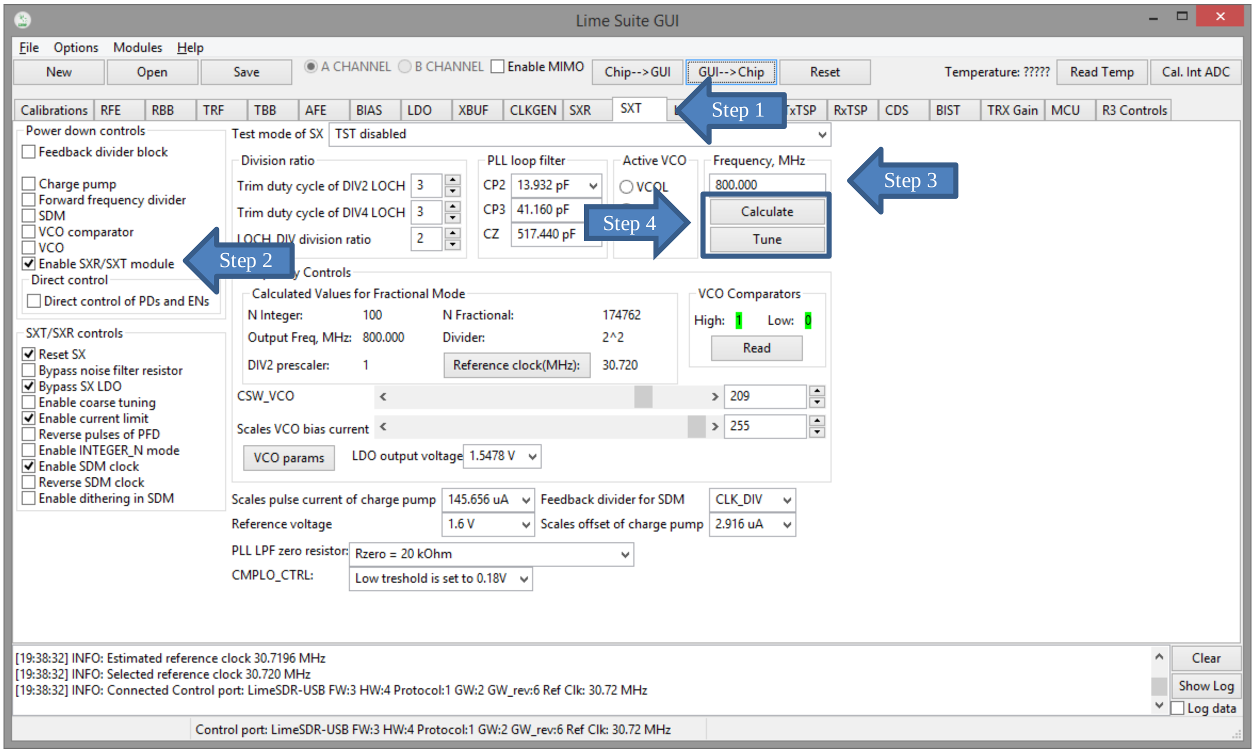
Figure 23 Performing Quick Test

3.4 Changing TX / RX Frequency

After power up in order to configure LMS7002M Tx or Rx LO to 2140 MHz, do the following:

1. Select the **SXR** tab for Receiver or **SXT** tab for Transmitter
2. Enable SXR/SXT module
3. Type the wanted frequency in **Frequency, GHz** box. In this case, 800 MHz
4. Press **Calculate** followed by **Tune**

See *Figure 24* below to check related controls in the LimeSuiteGUI.


Lime Suite GUI
File Options Modules Help
New Open Save
● A CHANNEL ○ B CHANNEL □ Enable MIMO
Chip-->GUI GUI-->Chip Reset
Temperature: ????? Read Temp Cal. Int ADC
Calibrations RFE RBB TRF TBB AFE BIAS LDO XBUF CLKGEN SXR SXT TxTSP RxTSP CDS BIST TRX Gain MCU R3 Controls
Step 1
Step 2
Step 3
Step 4
Power down controls
□ Feedback divider block
□ Charge pump
□ Forward frequency divider
□ SDM
□ VCO comparator
□ VCO
☑ Enable SXR/SXT module
Direct control
□ Direct control of PDs and ENs
SXT/SXR controls
☑ Reset SX
□ Bypass noise filter resistor
☑ Bypass SX LDO
□ Enable coarse tuning
☑ Enable current limit
□ Reverse pulses of PFD
□ Enable INTEGER_N mode
☑ Enable SDM clock
□ Reverse SDM clock
□ Enable dithering in SDM
Test mode of SX TST disabled
Division ratio
Trim duty cycle of DIV2 LOCH 3
Trim duty cycle of DIV4 LOCH 3
LOCH_DIV division ratio 2
PLL loop filter
CP2 13.932 pF
CP3 41.160 pF
CZ 517.440 pF
Active VCO
○ VCOL
Frequency, MHz
800.000
Calculate
Tune
Controls
Calculated Values for Fractional Mode
N Integer: 100
N Fractional: 174762
Output Freq, MHz: 800.000
Divider: 2^2
DIV2 prescaler: 1
Reference clock(MHz): 30.720
VCO Comparators
High: 1 Low: 0
Read
CSW_VCO 209
Scales VCO bias current 255
VCO params
LDO output voltage 1.5478 V
Scales pulse current of charge pump 145.656 uA
Feedback divider for SDM CLK_DIV
Reference voltage 1.6 V
Scales offset of charge pump 2.916 uA
PLL LPF zero resistor: Rzero = 20 kOhm
CMPLO_CTRL: Low treshold is set to 0.18V
Clear
Show Log
□ Log data

```
[19:38:32] INFO: Estimated reference clock 30.7196 MHz
[19:38:32] INFO: Selected reference clock 30.720 MHz
[19:38:32] INFO: Connected Control port: LimeSDR-USB FW:3 HW:4 Protocol:1 GW:2 GW_rev:6 Ref Clk: 30.72 MHz
```

Control port: LimeSDR-USB FW:3 HW:4 Protocol:1 GW:2 GW_rev:6 Ref Clk: 30.72 MHz

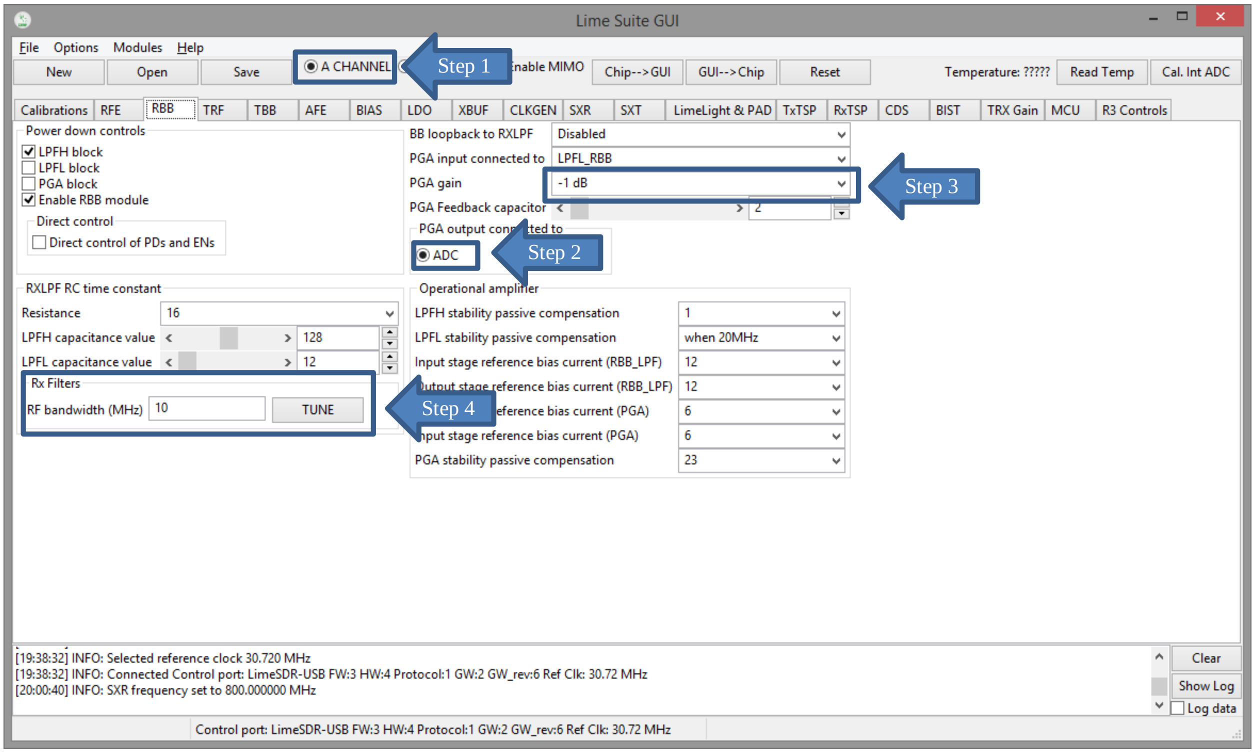
Figure 24 SXR / SXT register setup procedure

3.5 Changing Receiver Gain

Select the RBB tab to configure the PGA gain and baseband filter bandwidths. Follow the configuration steps below:

1. Select the **A CHANNEL** to control channel A
2. Select PGA output to **output pads**. This selection enables receiver analog outputs
3. Set **PGA gain** to -1 dB
4. Configure filter bandwidth. Type desired bandwidth and click Tune

See *Figure 25* below to check related controls in the LimeSuiteGUI.



● A CHANNEL

Temperature: ?????

Power down controls

☑ LPFH block
☐ LPFL block
☐ PGA block
☑ Enable RBB module

Direct control

☐ Direct control of PDs and ENs

BB loopback to RXLPF: Disabled
PGA input connected to: LPFL_RBB
PGA gain: -1 dB
PGA Feedback capacitor: 2

PGA output connected to

● ADC

RXLPF RC time constant

Resistance: 16
LPFH capacitance value: 128
LPFL capacitance value: 12

Rx Filters

RF bandwidth (MHz): 10

Operational amplifier

LPFH stability passive compensation: 1
LPFL stability passive compensation: when 20MHz
Input stage reference bias current (RBB_LPF): 12
Output stage reference bias current (RBB_LPF): 12
Output stage reference bias current (PGA): 6
Input stage reference bias current (PGA): 6
PGA stability passive compensation: 23

```
[19:38:32] INFO: Selected reference clock 30.720 MHz
[19:38:32] INFO: Connected Control port: LimeSDR-USB FW:3 HW:4 Protocol:1 GW:2 GW_rev:6 Ref Clk: 30.72 MHz
[20:00:40] INFO: SXR frequency set to 800.000000 MHz
```

Control port: LimeSDR-USB FW:3 HW:4 Protocol:1 GW:2 GW_rev:6 Ref Clk: 30.72 MHz

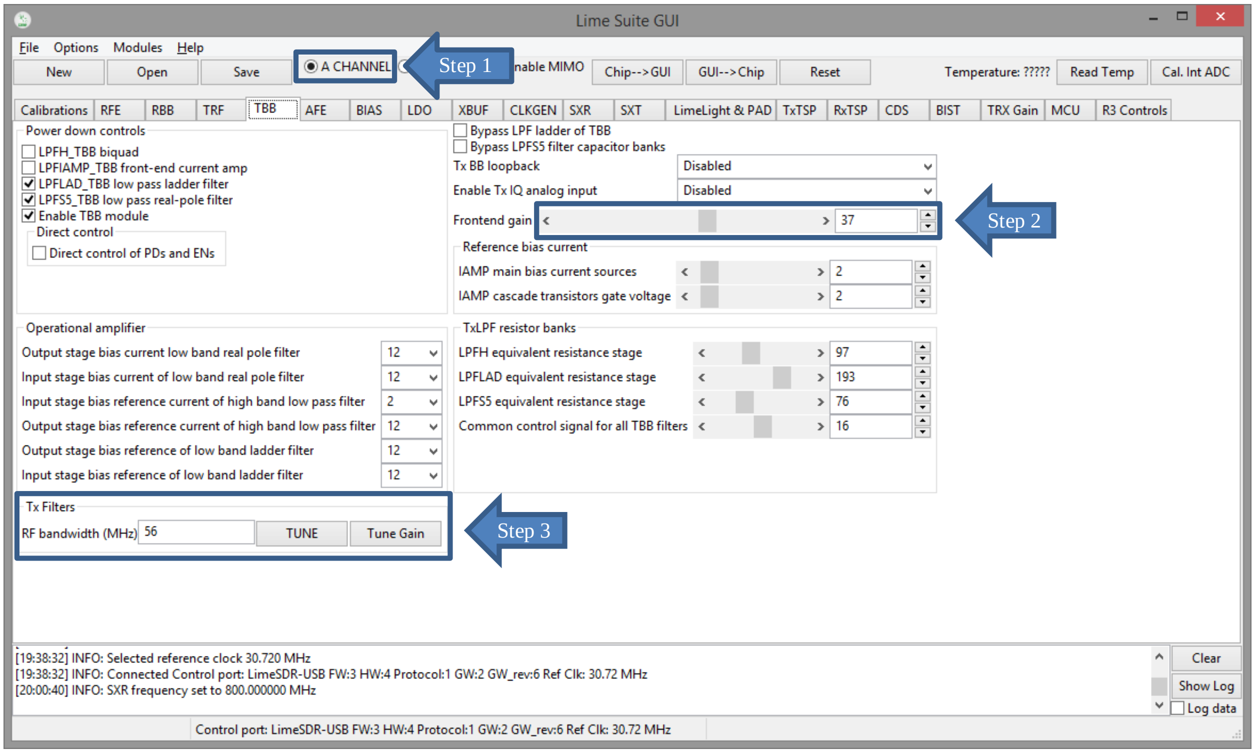
Figure 25 RBB register setup procedure

3.6 Changing Transmitter Output Signal Gain

In the **TBB** tab the baseband gain and filter bandwidth are controlled. Follow the instructions below to set up TBB:

1. Select the **A CHANNEL** to control channel A
2. Set **Frontend gain** to your wanted
3. Configure the base band filter settings. Type desired bandwidth and click Tune and Tune gain

See *Figure 26* below to check related controls in the LimeSuiteGUI.


Lime Suite GUI
File Options Modules Help
New Open Save
● A CHANNEL
Enable MIMO
Chip-->GUI GUI-->Chip Reset
Temperature: ?????
Read Temp Cal. Int ADC
Calibrations RFE RBB TRF TBB AFE BIAS LDO XBUF CLKGEN SXR SXT LimeLight & PAD TxTSP RxTSP CDS BIST TRX Gain MCU R3 Controls
Step 1
Step 2
Step 3
TUNE Tune Gain
Clear Show Log
Log data

Power down controls
- ☐ LPFH_TBB biquad
- ☐ LPFIAMP_TBB front-end current amp
- ☑ LPFLAD_TBB low pass ladder filter
- ☑ LPFS5_TBB low pass real-pole filter
- ☑ Enable TBB module

Direct control
- ☐ Direct control of PDs and ENs

- ☐ Bypass LPF ladder of TBB
- ☐ Bypass LPFS5 filter capacitor banks

Tx BB loopback: Disabled
Enable Tx IQ analog input: Disabled
Frontend gain: 37

Reference bias current
- IAMP main bias current sources: 2
- IAMP cascade transistors gate voltage: 2

Operational amplifier
- Output stage bias current low band real pole filter: 12
- Input stage bias current of low band real pole filter: 12
- Input stage bias reference current of high band low pass filter: 2
- Output stage bias reference current of high band low pass filter: 12
- Output stage bias reference of low band ladder filter: 12
- Input stage bias reference of low band ladder filter: 12

TxLPF resistor banks
- LPFH equivalent resistance stage: 97
- LPFLAD equivalent resistance stage: 193
- LPFS5 equivalent resistance stage: 76
- Common control signal for all TBB filters: 16

Tx Filters
RF bandwidth (MHz): 56

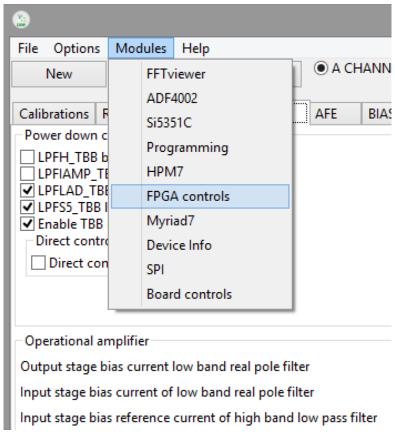
```
[19:38:32] INFO: Selected reference clock 30.720 MHz
[19:38:32] INFO: Connected Control port: LimeSDR-USB FW:3 HW:4 Protocol:1 GW:2 GW_rev:6 Ref Clk: 30.72 MHz
[20:00:40] INFO: SXR frequency set to 800.000000 MHz
```

Control port: LimeSDR-USB FW:3 HW:4 Protocol:1 GW:2 GW_rev:6 Ref Clk: 30.72 MHz

Figure 26 TBB register setup procedure

3.7 Load Waveform for Tx Path

The programed FPGA is acting as waveform player for LMS7002M transceiver. In order to load the waveform, select **Modules** from top menu, then **FPGA Controls** from the drop down menu as shown in *Figure* 27.



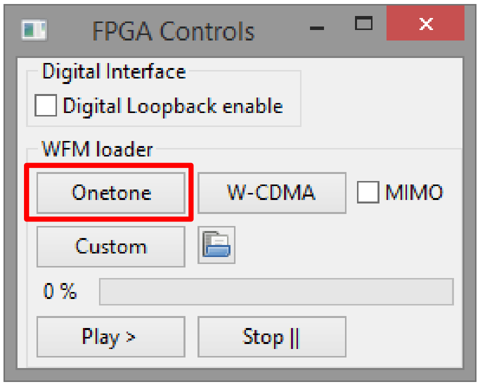
Power down c
- ☐ LPFH_TBB b
- ☐ LPFIAMP_TE
- ☑ LPFLAD_TBE
- ☑ LPFS5_TBB l
- ☑ Enable TBB

Direct contro
- ☐ Direct con

Operational amplifier
- Output stage bias current low band real pole filter
- Input stage bias current of low band real pole filter
- Input stage bias reference current of high band low pass filter

Figure 27 Selecting FPGA Control window

New window will appear in the bottom of the GUI, offering you to load supplied waveforms or custom waveforms. Please select to load CW waveform by clicking on **Onetone** button, as shown in *Figure* 28.



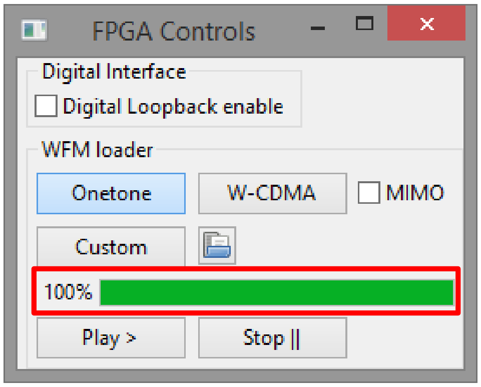
□ Digital Loopback enable

□ MIMO

0 %

Figure 28 Waveform selection

The file loading process to the FPGA is shown by indication bar, see *Figure* 29.



□ Digital Loopback enable

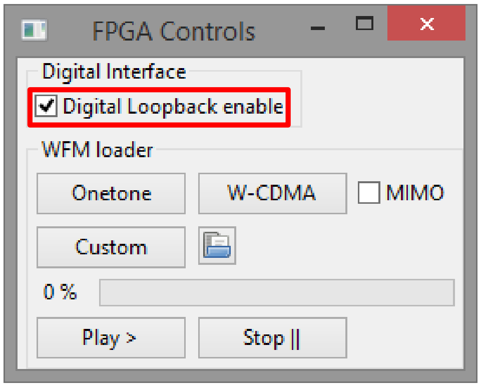
□ MIMO

100%

Figure 29 Loaded waveform indication

3.8 Digital Loopback Enable

There is also implemented option to receive data from LMS7002M receiver and stream back on to LMS7002M transmitter. In order to enable this option, click on the ‘Digital Loopback enable’ check box in the ‘FPGA Control’ Module Figure 30.



☑ Digital Loopback enable

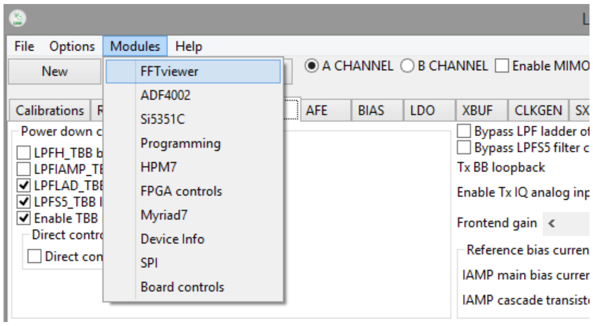
☐ MIMO

0 %

Figure 30 Select Digital Loopback enable

3.9 Run FTT Viewer

FFTviewer module is a part of LimeSuiteGUI software. To run FFTviewer, go to top menu, select **Modules** and choose **FFTviewer** as shown in Figure 31.

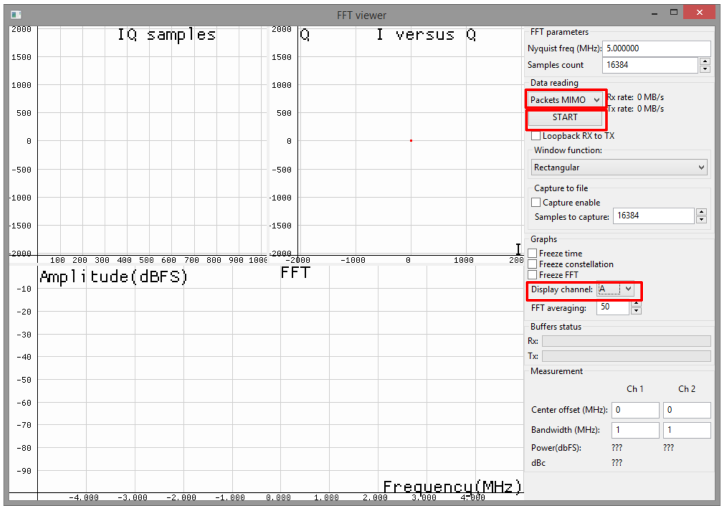


● A CHANNEL ○ B CHANNEL ☐ Enable MIMO

☐ LPFH_TBB b
☐ LPFIAMP_TI
☑ LPFLAD_TBI
☑ LPFS5_TBB I
☑ Enable TBB
☐ Direct con
☐ Bypass LPF ladder of
☐ Bypass LPFS5 filter c

Figure 31 LimeSuiteGUI module menu to select FFTviewer

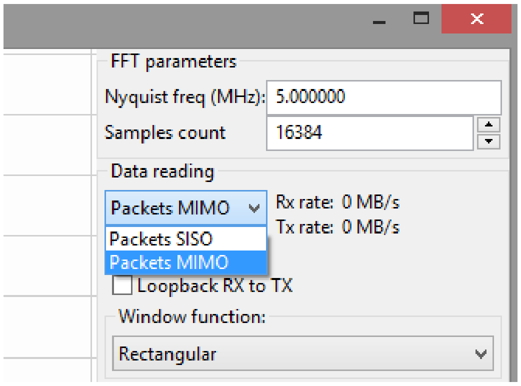
FFTviewer control window will appear. Before start capturing data, set the **Data reading** type to “Packets MIMO”, Display channel and press **Start**, as shown in *Figure* 32 and Figure 33.



Nyquist freq (MHz): 5.000000
Samples count 16384
Packets MIMO
Rx rate: 0 MB/s
Tx rate: 0 MB/s
☐ Loopback RX to TX
Window function: Rectangular
☐ Capture enable
Samples to capture: 16384
☐ Freeze time
☐ Freeze constellation
☐ Freeze FFT
Display channel: A
FFT averaging: 50

	Ch 1	Ch 2
Center offset (MHz):	0	0
Bandwidth (MHz):	1	1
Power(dbFS):	???	???
dBc	???	

Figure 32 FFTviewer Controls



FFT parameters

Nyquist freq (MHz): 5.000000

Samples count 16384

Data reading

Packets MIMO

Packets SISO
Packets MIMO

Rx rate: 0 MB/s
Tx rate: 0 MB/s

☐ Loopback RX to TX

Window function:

Rectangular

Figure 33 Setting data type to Packets MIMO

At this point, the FFTviewer start capturing data. Connect the generator to selected LimeSDR receiver path. In the *Figure 34* showed the FFTviewer data capture with 1 MHz CW signal offset from LO.

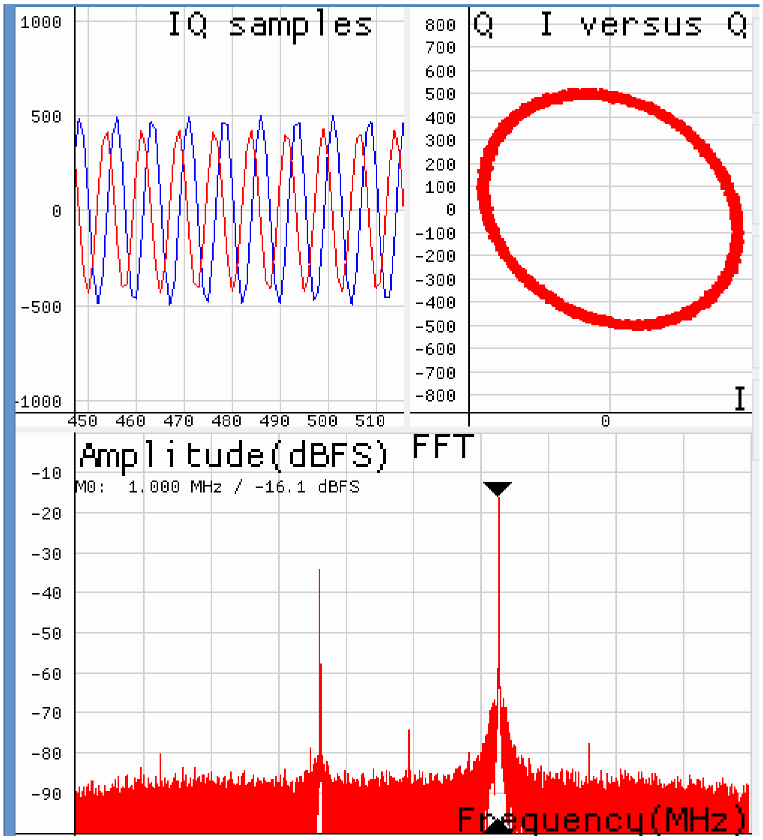
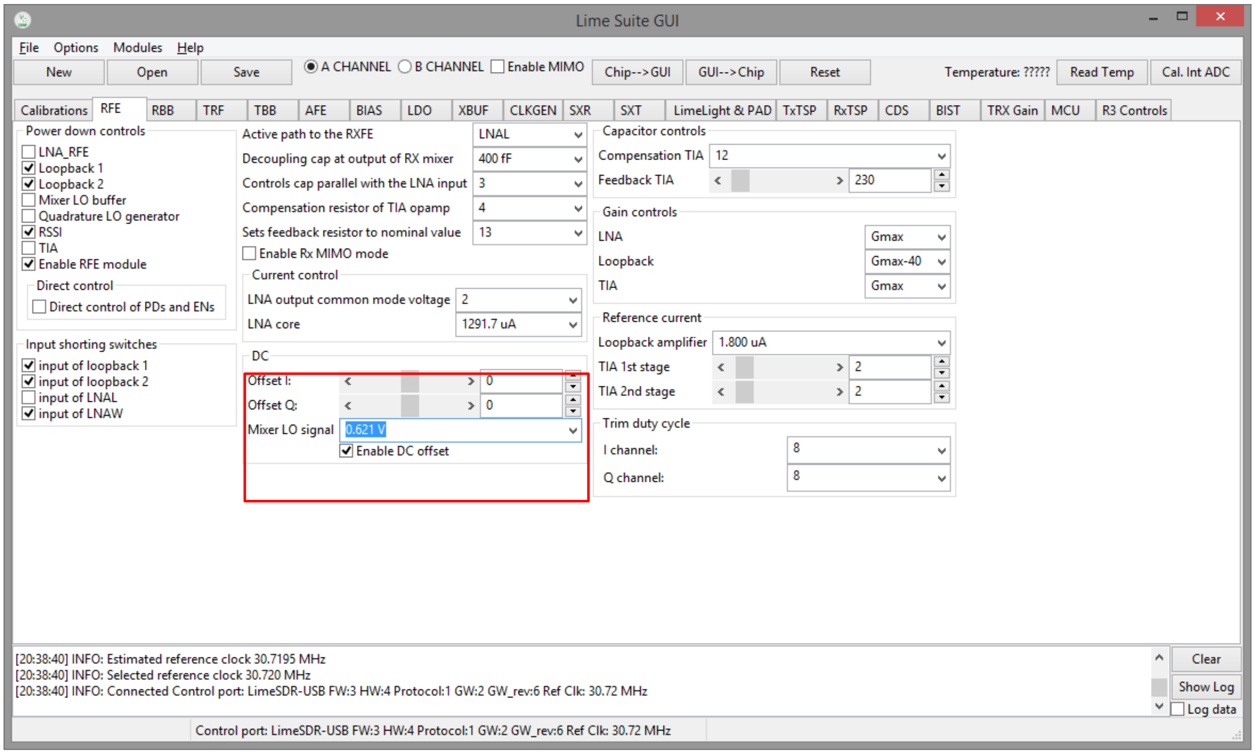


Figure 34 FFTviewer window in operation

3.10 Manual Calibration of RX Path

Rx DC offset and Rx Unwanted SSB calibration routines have to be executed to calibrate receiver path. The Rx DC offset calibration split in two parts; Analog DC Offset calibration and digital DC offset removal procedure.

To execute Analog DC Offset calibration, select the **RFE** tab in the main GUI window. Make sure that you have selected channel A. In the **DC** box, change **Mixer LO signal** to **0.621 V** and look for the best Offset I/Q values to reach minimum level of DC Offset. See *Figure 35* below.



● A CHANNEL ○ B CHANNEL □ Enable MIMO

Temperature: ?????

Power down controls
□ LNA_RFE
☑ Loopback 1
☑ Loopback 2
□ Mixer LO buffer
□ Quadrature LO generator
☑ RSSI
□ TIA
☑ Enable RFE module

Direct control
□ Direct control of PDs and ENs

Input shorting switches
☑ input of loopback 1
☑ input of loopback 2
□ input of LNAL
☑ input of LNAW

Active path to the RXFE: LNAL
Decoupling cap at output of RX mixer: 400 fF
Controls cap parallel with the LNA input: 3
Compensation resistor of TIA opamp: 4
Sets feedback resistor to nominal value: 13
□ Enable Rx MIMO mode

Current control
LNA output common mode voltage: 2
LNA core: 1291.7 uA

DC
Offset I: 0
Offset Q: 0
Mixer LO signal: 0.621 V
☑ Enable DC offset

Capacitor controls
Compensation TIA: 12
Feedback TIA: 230

Gain controls
LNA: Gmax
Loopback: Gmax-40
TIA: Gmax

Reference current
Loopback amplifier: 1.800 uA
TIA 1st stage: 2
TIA 2nd stage: 2

Trim duty cycle
I channel: 8
Q channel: 8

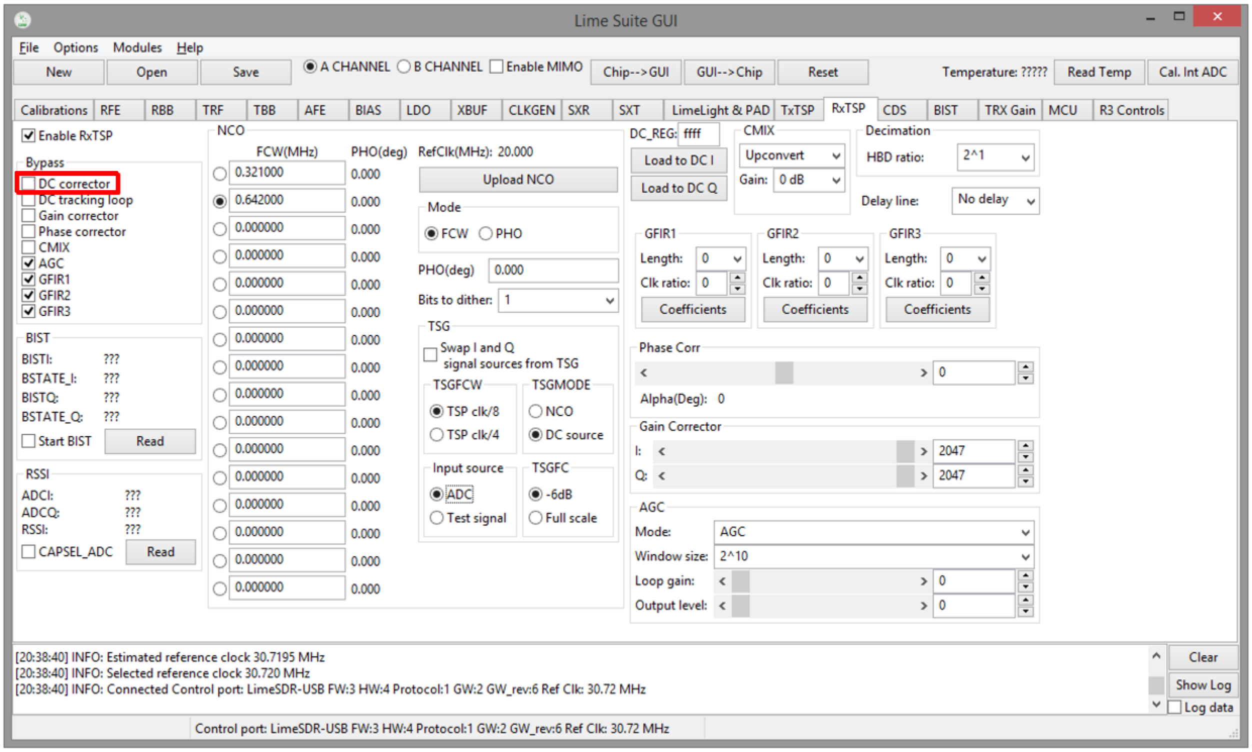
```
[20:38:40] INFO: Estimated reference clock 30.7195 MHz
[20:38:40] INFO: Selected reference clock 30.720 MHz
[20:38:40] INFO: Connected Control port: LimeSDR-USB FW:3 HW:4 Protocol:1 GW:2 GW_rev:6 Ref Clk: 30.72 MHz
```

□ Log data

Control port: LimeSDR-USB FW:3 HW:4 Protocol:1 GW:2 GW_rev:6 Ref Clk: 30.72 MHz

Figure 35 RX DC offset manual calibration

For residual DC offset calibration you need to enable the **DC corrector** in **RxTSP** tab. See Figure 36. It should be enabled (check box un-checked) by default.


Lime Suite GUI
File Options Modules Help
New Open Save
● A CHANNEL ○ B CHANNEL □ Enable MIMO
Chip-->GUI GUI-->Chip Reset
Read Temp Cal. Int ADC
Calibrations RFE RBB TRF TBB AFE BIAS LDO XBUF CLKGEN SXR SXT LimeLight & PAD TxTSP RxTSP CDS BIST TRX Gain MCU R3 Controls
Upload NCO
Load to DC I
Load to DC Q
Coefficients
Read
Clear
Show Log

Temperature: ?????

☑ Enable RxTSP

Bypass
- □ DC corrector
- □ DC tracking loop
- □ Gain corrector
- □ Phase corrector
- □ CMIX
- ☑ AGC
- ☑ GFIR1
- ☑ GFIR2
- ☑ GFIR3

BIST
- BISTI: ???
- BSTATE_I: ???
- BISTQ: ???
- BSTATE_Q: ???
- □ Start BIST

RSSI
- ADCI: ???
- ADCQ: ???
- RSSI: ???
- □ CAPSEL_ADC

NCO

	FCW(MHz)	PHO(deg)
○	0.321000	0.000
●	0.642000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000

RefClk(MHz): 20.000

Mode: ● FCW ○ PHO

PHO(deg): 0.000

Bits to dither: 1

TSG
- □ Swap I and Q signal sources from TSG
- TSGFCW: ● TSP clk/8 ○ TSP clk/4
- TSGMODE: ○ NCO ● DC source
- Input source: ● ADC ○ Test signal
- TSGFC: ● -6dB ○ Full scale

DC_REG: ffff

CMIX: Upconvert; Gain: 0 dB

Decimation: HBD ratio: 2^1; Delay line: No delay

GFIR1: Length: 0; Clk ratio: 0

GFIR2: Length: 0; Clk ratio: 0

GFIR3: Length: 0; Clk ratio: 0

Phase Corr: 0; Alpha(Deg): 0

Gain Corrector: I: 2047; Q: 2047

AGC: Mode: AGC; Window size: 2^10; Loop gain: 0; Output level: 0

```
[20:38:40] INFO: Estimated reference clock 30.7195 MHz
[20:38:40] INFO: Selected reference clock 30.720 MHz
[20:38:40] INFO: Connected Control port: LimeSDR-USB FW:3 HW:4 Protocol:1 GW:2 GW_rev:6 Ref Clk: 30.72 MHz
```

□ Log data

Control port: LimeSDR-USB FW:3 HW:4 Protocol:1 GW:2 GW_rev:6 Ref Clk: 30.72 MHz

Figure 36 Enable DC corrector in RxTSP

The unwanted SSB can be seen on FFTviewer window by applying signal to one of the transceiver inputs. See *Figure* 37.

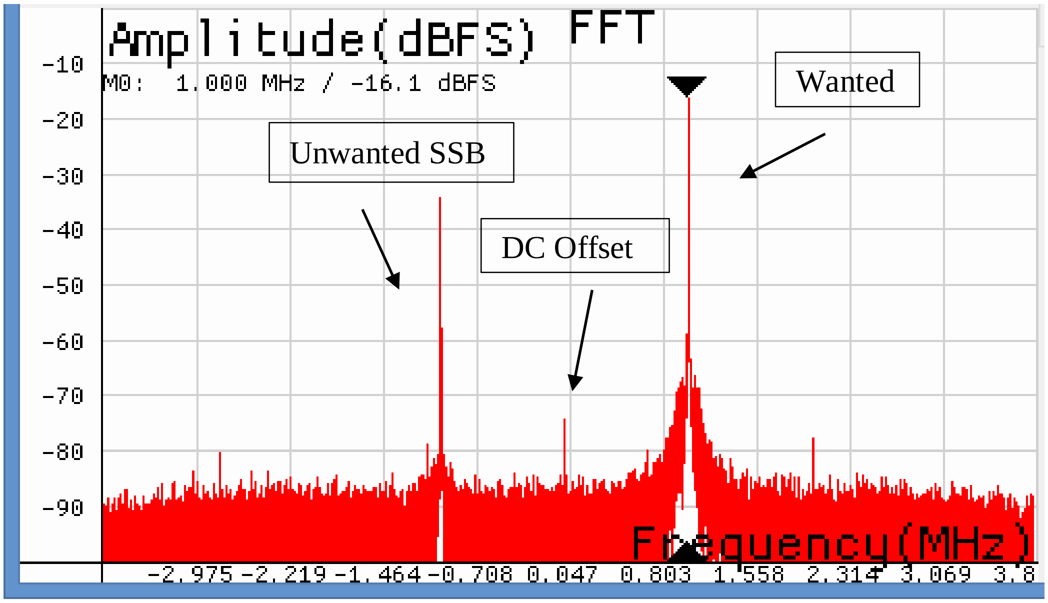
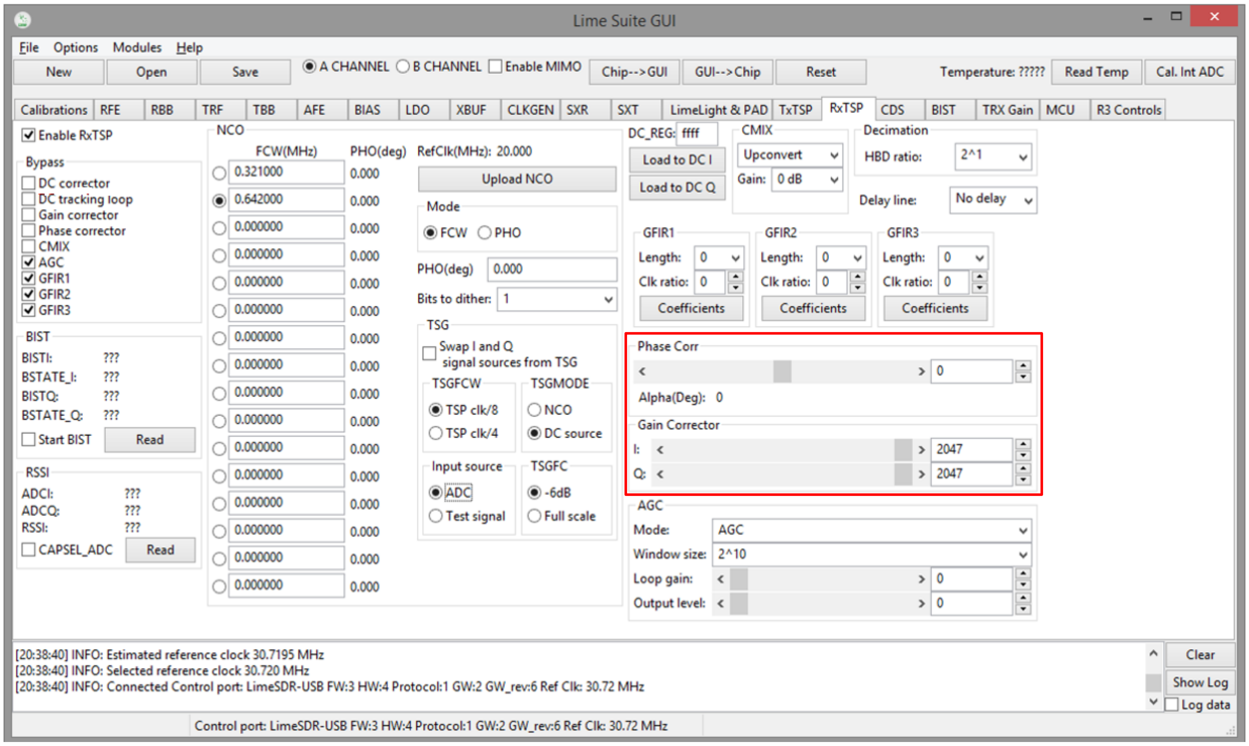


Figure 37 Receiver spectrum with un-calibrated IQ imbalance

To calibrate RX IQ imbalance, go to **RxTSP** tab on *LimeSuiteGUI* GUI. On **IQ Correction** box adjust **Gain ch**. **I** or G**ain ch. Q** followed by **Phase correction** to reduce the Unwanted SSB. See Figure 38.



```
[20:38:40] INFO: Estimated reference clock 30.7195 MHz
[20:38:40] INFO: Selected reference clock 30.720 MHz
[20:38:40] INFO: Connected Control port: LimeSDR-USB FW:3 HW:4 Protocol:1 GW:2 GW_rev:6 Ref Clk: 30.72 MHz
```

Control port: LimeSDR-USB FW:3 HW:4 Protocol:1 GW:2 GW_rev:6 Ref Clk: 30.72 MHz

Figure 38 Rx IQ correction block

Calibrated receiver spectrum should look like in the *Figure* 39.

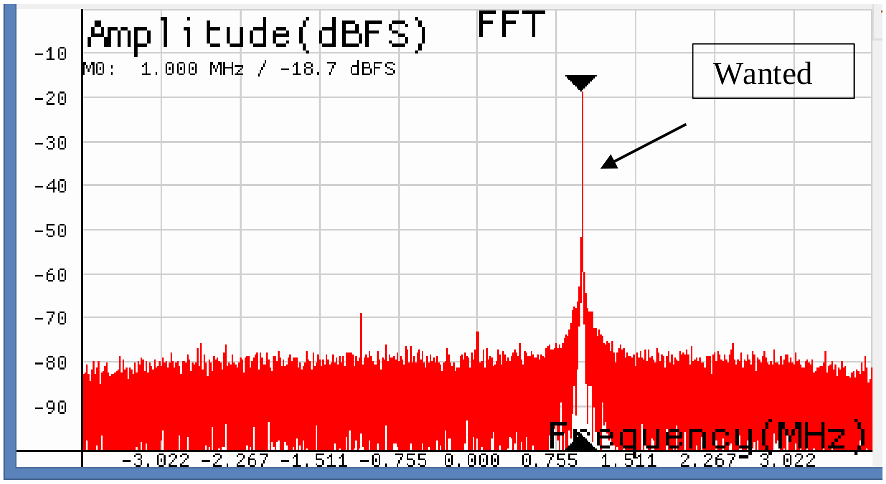
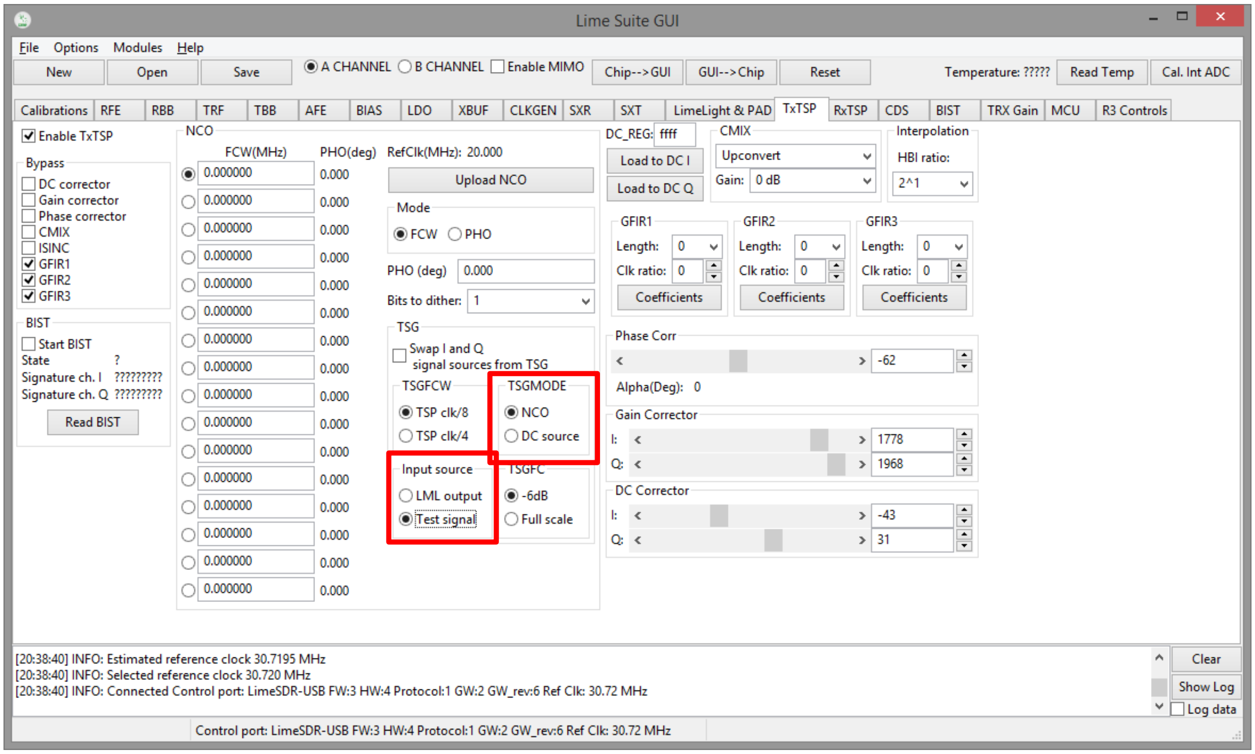


Figure 39 Calibrated Rx Spectrum

3.11 Manual Calibration of TX Path

The LO leakage and IQ imbalance have to be calibrated for the LMS7002M transceiver in order to get optimum performance for Tx EVM measurement. The IQ imbalance calibration is done by generating CW and adjusting IQ phase/gain error for IQ mismatch. Th LO leakage calibration is doem by adjust DC offset registers. The internal test NCO can be enabled for this purpose. To do this, select **TxTSP** tab in *LimeSuiteGUI* and select the **Test Signal** as input for Tx path and **NCO** as **TSGMODE**, as showed in figure below.


Lime Suite GUI
File Options Modules Help
New Open Save
Chip-->GUI GUI-->Chip Reset
Temperature: ????? Read Temp Cal. Int ADC
Calibrations RFE RBB TRF TBB AFE BIAS LDO XBUF CLKGEN SXR SXT LimeLight & PAD TxTSP RxTSP CDS BIST TRX Gain MCU R3 Controls
Upload NCO
Load to DC I
Load to DC Q
Coefficients
Read BIST
Clear
Show Log

● A CHANNEL ○ B CHANNEL □ Enable MIMO

☑ Enable TxTSP

Bypass: □ DC corrector □ Gain corrector □ Phase corrector □ CMIX □ ISINC ☑ GFIR1 ☑ GFIR2 ☑ GFIR3

BIST: □ Start BIST; State ?; Signature ch. I ?????????; Signature ch. Q ?????????

NCO

	FCW(MHz)	PHO(deg)
●	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000

RefClk(MHz): 20.000

Mode: ● FCW ○ PHO

PHO (deg): 0.000

Bits to dither: 1

TSG: □ Swap I and Q signal sources from TSG

TSGFCW: ● TSP clk/8 ○ TSP clk/4

TSGMODE: ● NCO ○ DC source

Input source: ○ LML output ● Test signal

TSGFC: ● -6dB ○ Full scale

DC_REG: ffff

CMIX: Upconvert; Gain: 0 dB

Interpolation: HBI ratio: 2^1

GFIR1: Length: 0; Clk ratio: 0
GFIR2: Length: 0; Clk ratio: 0
GFIR3: Length: 0; Clk ratio: 0

Phase Corr: -62; Alpha(Deg): 0

Gain Corrector: I: 1778; Q: 1968

DC Corrector: I: -43; Q: 31

```
[20:38:40] INFO: Estimated reference clock 30.7195 MHz
[20:38:40] INFO: Selected reference clock 30.720 MHz
[20:38:40] INFO: Connected Control port: LimeSDR-USB FW:3 HW:4 Protocol:1 GW:2 GW_rev:6 Ref Clk: 30.72 MHz
```

□ Log data

Control port: LimeSDR-USB FW:3 HW:4 Protocol:1 GW:2 GW_rev:6 Ref Clk: 30.72 MHz

Figure 40 Enable the test NCO

NOTE: Before configuring **TxTSP** tab, select the **A/RXT** channel in top right of the GUI.

On the transmitter output you should see the wanted CW with 3.8MHz offset from LO, unwanted SSB on the other side of spectrum and LO leakage. See *Figure* 41.

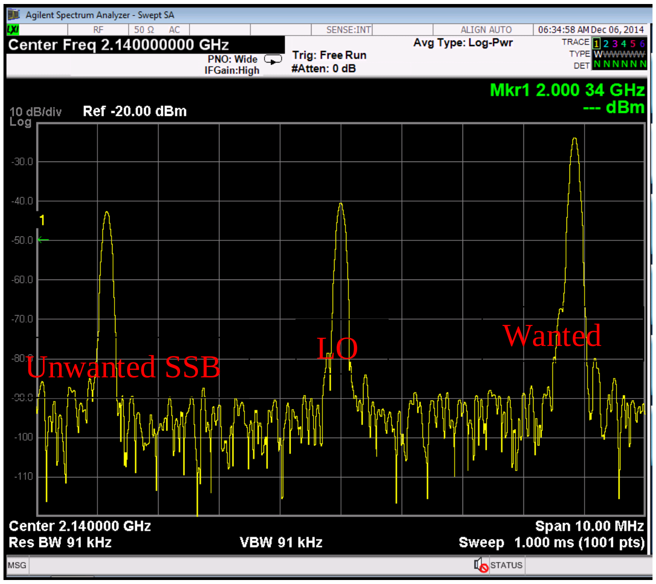
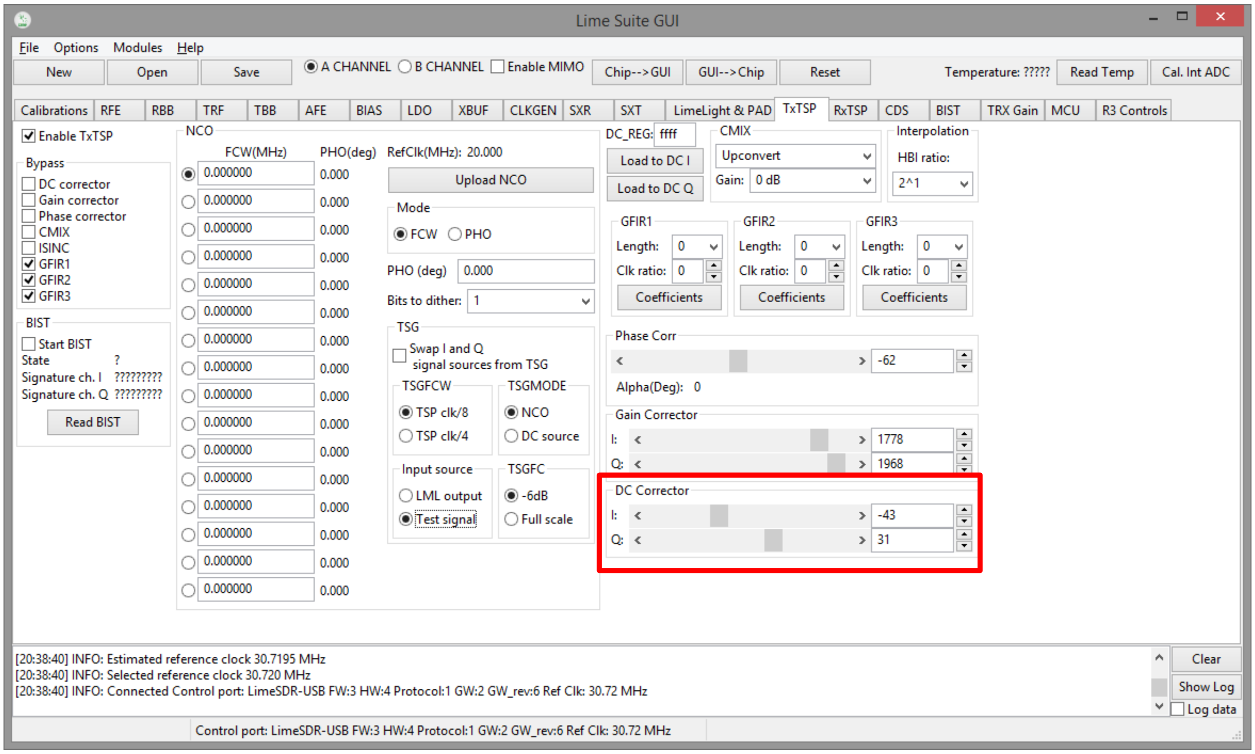


Figure 41 Not calibrated Tx Output

To do the LO leakage calibration, select **TxTSP** tab in the LimeSuiteGUI GUI and adjust the **DC Corrector** settings (see Figure 42) for channel I and Q separately to get minimum LO leakage.


Lime Suite GUI
File Options Modules Help
New Open Save
● A CHANNEL ○ B CHANNEL □ Enable MIMO
Chip-->GUI GUI-->Chip Reset
Temperature: ?????
Read Temp Cal. Int ADC
Calibrations RFE RBB TRF TBB AFE BIAS LDO XBUF CLKGEN SXR SXT LimeLight & PAD TxTSP RxTSP CDS BIST TRX Gain MCU R3 Controls
Upload NCO
Load to DC I
Load to DC Q
Coefficients
Coefficients
Coefficients
Read BIST
Clear
Show Log
□ Log data

☑ Enable TxTSP

Bypass
□ DC corrector
□ Gain corrector
□ Phase corrector
□ CMIX
□ ISINC
☑ GFIR1
☑ GFIR2
☑ GFIR3

BIST
□ Start BIST
State ?
Signature ch. I ?????????
Signature ch. Q ?????????

NCO

	FCW(MHz)	PHO(deg)
●	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000
○	0.000000	0.000

RefClk(MHz): 20.000

Mode
● FCW ○ PHO

PHO (deg) 0.000
Bits to dither: 1

TSG
□ Swap I and Q signal sources from TSG

TSGFCW
● TSP clk/8
○ TSP clk/4

TSGMODE
● NCO
○ DC source

Input source
○ LML output
● Test signal

TSGFC
● -6dB
○ Full scale

DC_REG: ffff

CMIX
Upconvert
Gain: 0 dB

Interpolation
HBI ratio: 2^1

GFIR1
Length: 0
Clk ratio: 0

GFIR2
Length: 0
Clk ratio: 0

GFIR3
Length: 0
Clk ratio: 0

Phase Corr
-62
Alpha(Deg): 0

Gain Corrector
I: 1778
Q: 1968

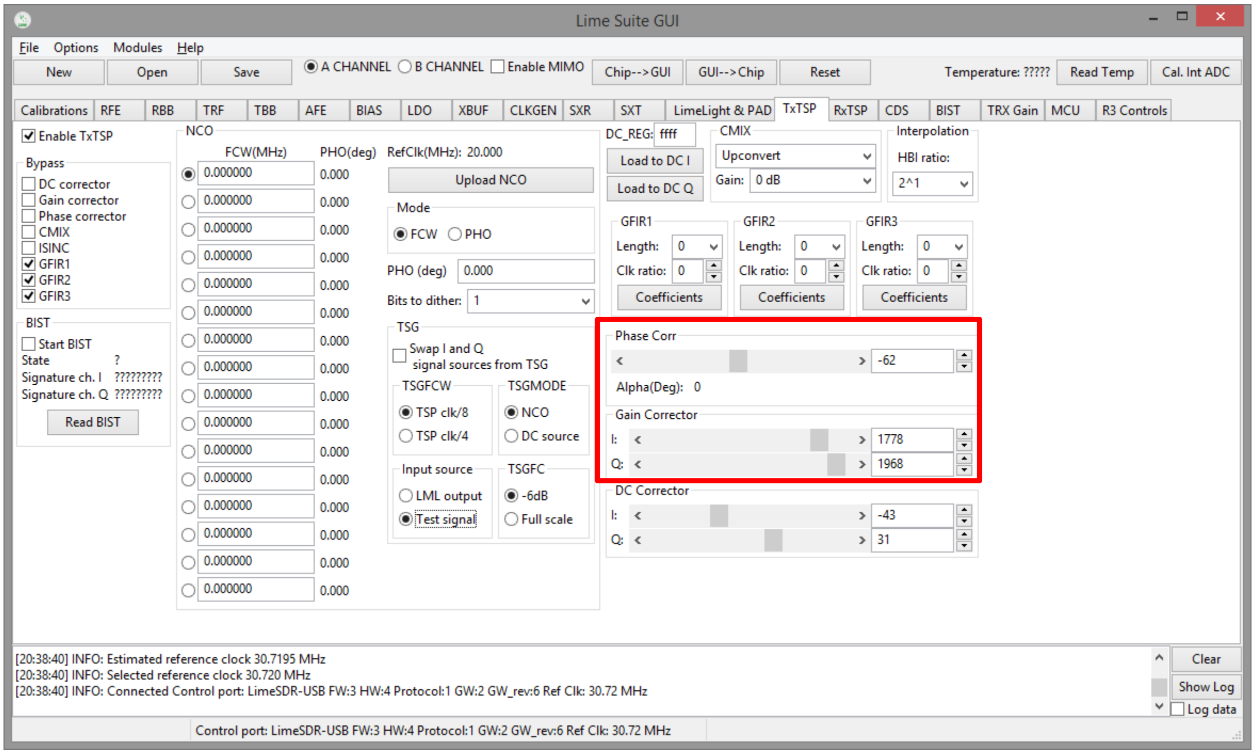
DC Corrector
I: -43
Q: 31

```
[20:38:40] INFO: Estimated reference clock 30.7195 MHz
[20:38:40] INFO: Selected reference clock 30.720 MHz
[20:38:40] INFO: Connected Control port: LimeSDR-USB FW:3 HW:4 Protocol:1 GW:2 GW_rev:6 Ref Clk: 30.72 MHz
```

Control port: LimeSDR-USB FW:3 HW:4 Protocol:1 GW:2 GW_rev:6 Ref Clk: 30.72 MHz

Figure 42 DC offset block control

To calibrate Unwanted SSB, use the **IQ Corrector** controls in the **TxTSP** tab. Change **I ch. gain** or **Q ch. gain** followed by **Phase correction** to reduce the Unwanted SSB as shown in Figure 43.


Lime Suite GUI
File Options Modules Help
New Open Save
● A CHANNEL ○ B CHANNEL □ Enable MIMO
Chip-->GUI GUI-->Chip Reset
Temperature: ????? Read Temp Cal. Int ADC
Calibrations RFE RBB TRF TBB AFE BIAS LDO XBUF CLKGEN SXR SXT LimeLight & PAD TxTSP RxTSP CDS BIST TRX Gain MCU R3 Controls
Upload NCO
Load to DC I
Load to DC Q
Coefficients
Read BIST
Clear
Show Log
□ Log data

☑ Enable TxTSP

Bypass: □ DC corrector, □ Gain corrector, □ Phase corrector, □ CMIX, □ ISINC, ☑ GFIR1, ☑ GFIR2, ☑ GFIR3

BIST: □ Start BIST, State ?, Signature ch. I ?????????, Signature ch. Q ?????????

NCO: FCW(MHz) 0.000000 (16 rows, first selected), PHO(deg) 0.000; RefClk(MHz): 20.000

Mode: ● FCW ○ PHO; PHO (deg) 0.000; Bits to dither: 1

TSG: □ Swap I and Q signal sources from TSG; TSGFCW: ● TSP clk/8 ○ TSP clk/4; TSGMODE: ● NCO ○ DC source; Input source: ○ LML output ● Test signal; TSGFC: ● -6dB ○ Full scale

DC_REG: ffff; CMIX: Upconvert, Gain: 0 dB; Interpolation: HBI ratio: 2^1

GFIR1: Length: 0, Clk ratio: 0; GFIR2: Length: 0, Clk ratio: 0; GFIR3: Length: 0, Clk ratio: 0

Phase Corr: -62; Alpha(Deg): 0

Gain Corrector: I: 1778; Q: 1968

DC Corrector: I: -43; Q: 31

```
[20:38:40] INFO: Estimated reference clock 30.7195 MHz
[20:38:40] INFO: Selected reference clock 30.720 MHz
[20:38:40] INFO: Connected Control port: LimeSDR-USB FW:3 HW:4 Protocol:1 GW:2 GW_rev:6 Ref Clk: 30.72 MHz
```

Control port: LimeSDR-USB FW:3 HW:4 Protocol:1 GW:2 GW_rev:6 Ref Clk: 30.72 MHz

Figure 43 IQ Corrector block control

Calibrated Transceiver TX output should look like in the *Figure* 44.

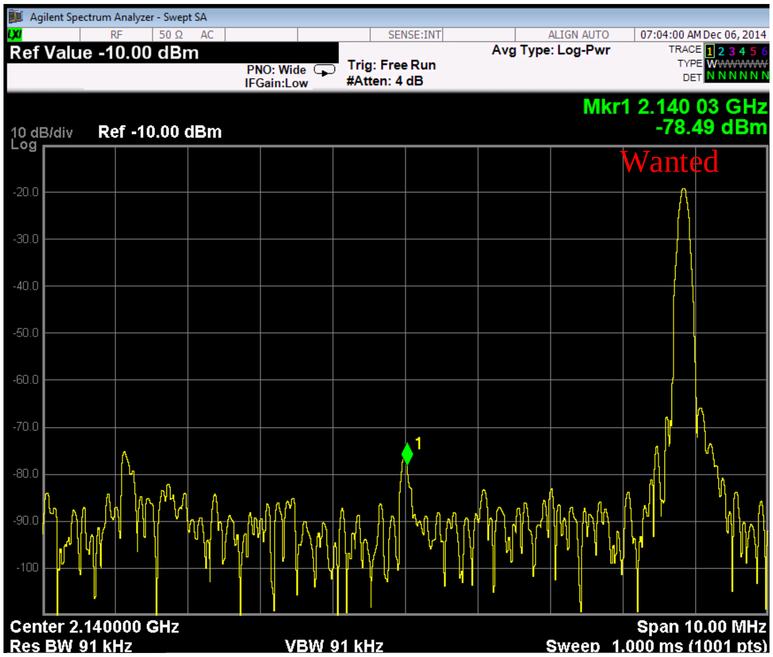


Figure 44 Calibrated Tx output

Once TX is calibrated the settings can be saved and can be recalled after chip power cycle. After calibration is complete and configure Tx path to accept data from Stream board; go to **TxTSP** and select **LML output** under Tx **Input Source** has to be selected to in TxTSP tab. See *Figure* 40.

NOTE: The Tx IQ and LO leakage calibration procedure can be done using auto calibration routines. The routines are accessed from **Calibration** tab in the GUI.

3.12 Clock Configuration

Onboard clock sources can be configured by LimeSuiteGUI. More information about clock distribution is detailed in chapter “2.2.8 Clock Distribution”.

3.12.1 VCTCXO Tuning

VCTCXO can be tuned by onboard phase detector (IC23, ADF4002) or by DAC (IC22). The onboard phase detector is used to synchronize onboard VCTCXO with external equipment (via J19 U.FL connector) to calibrate frequency error. At the same time only phase detector or DAC can control VCTCXO. DAC and phase detector is controlled by FX3 (USB) and selection between them is done automatically. When board is powered, by default VCTCXO is controlled by DAC.

FPGA2 LED indicates DAC or phase detector controls VCTCXO and phase detector lock state.

3.12.2 Tuning VCTCXO Using Frequency Synthesizer (ADF4002)

VCTCXO can be tuned by onboard phase detector (IC23, ADF4002) or by DAC (IC22). If phase detector is configured from *LimeSuiteGUI* software, then DAC is disabled automatically and VCTCXO tuning voltage is supplied from phase detector. When phase detector controls VCTCXO, FPGA2 LED indicates its lock state: red – not locked, green – locked.

The phase detector is used to synchronize onboard VCTCXO with external equipment (via J19 U.FL connector) to calibrate frequency error and can be configured using *LimeSuiteGUI* software. Because VCTCXO also can be tuned by DAC, in this case DAC is disabled. Go to **Modules** form top menu and select **ADF4002** form the drop down menu, as shown in *Figure* 45.

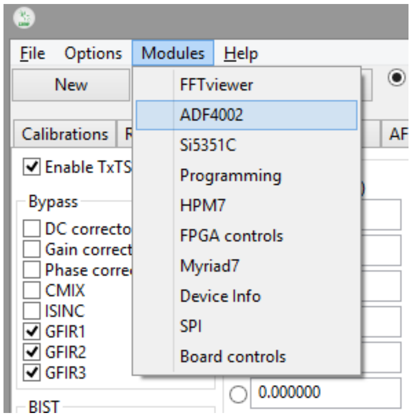
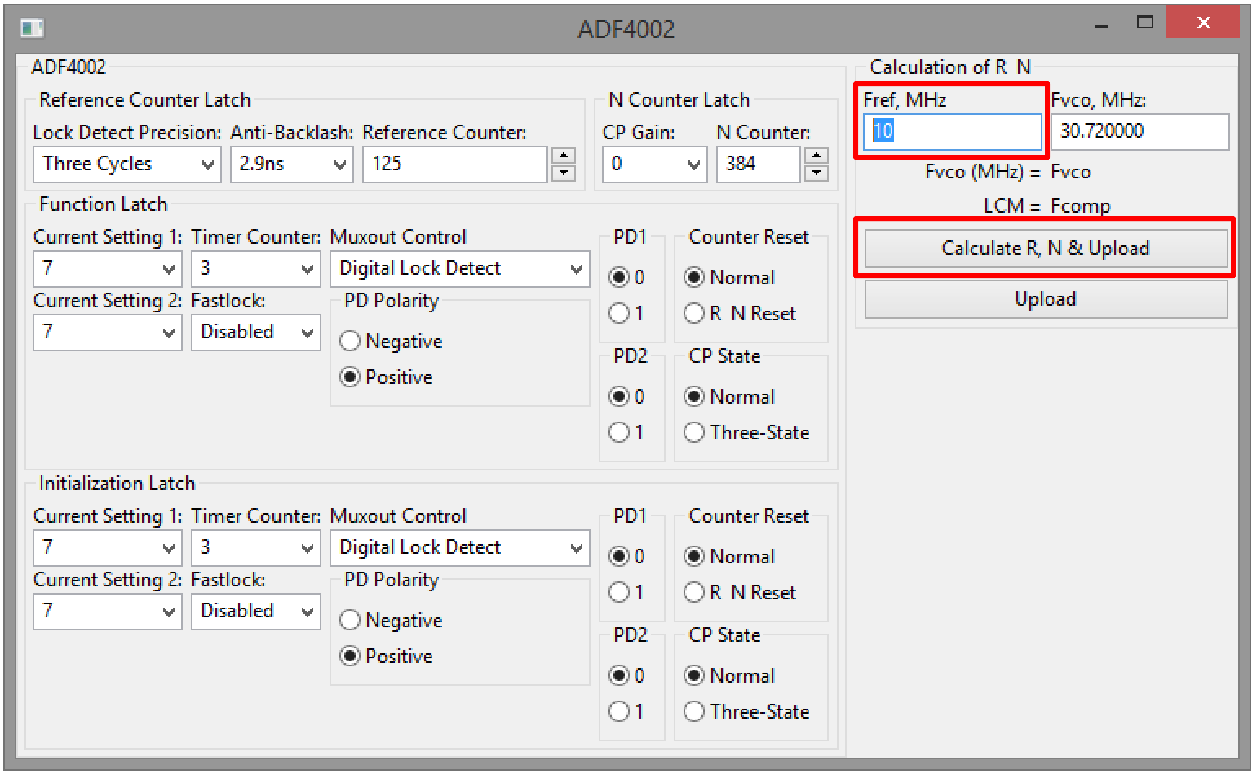


Figure 45 LimeSuiteGUI module menu to select ADF4002 configuration tool

New control window should appear, as shown in the *Figure* 46.



ADF4002

Reference Counter Latch
Lock Detect Precision: Three Cycles
Anti-Backlash: 2.9ns
Reference Counter: 125

N Counter Latch
CP Gain: 0
N Counter: 384

Function Latch
Current Setting 1: 7
Timer Counter: 3
Muxout Control: Digital Lock Detect
Current Setting 2: 7
Fastlock: Disabled
PD Polarity: ○ Negative ● Positive
PD1: ● 0 ○ 1
Counter Reset: ● Normal ○ R N Reset
PD2: ● 0 ○ 1
CP State: ● Normal ○ Three-State

Initialization Latch
Current Setting 1: 7
Timer Counter: 3
Muxout Control: Digital Lock Detect
Current Setting 2: 7
Fastlock: Disabled
PD Polarity: ○ Negative ● Positive
PD1: ● 0 ○ 1
Counter Reset: ● Normal ○ R N Reset
PD2: ● 0 ○ 1
CP State: ● Normal ○ Three-State

Calculation of R N
Fref, MHz: 10
Fvco, MHz: 30.720000
Fvco (MHz) = Fvco
LCM = Fcomp

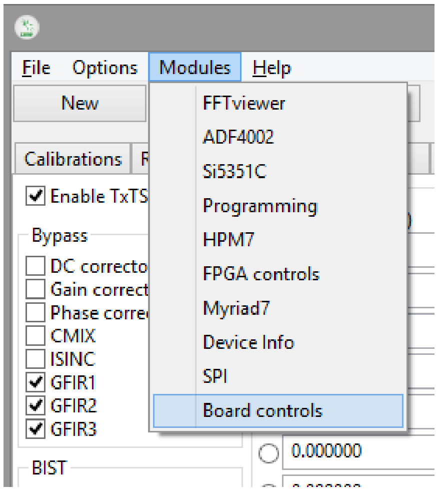
Figure 46 ADF4002 configuration window

Field **Fref** value indicates the frequency to that VCTCXO will be synchronized and must be fed to REF_CLK (J14) connector. Usually this value is 10 MHz. When all parameters are entered in this window, press button **Calculate R, N & Upload** and frequency synthesizer will be configured.

3.12.3 Tuning VCTCXO using DAC

VCTCXO can be tuned by onboard frequency synthesizer (IC16, ADF4002) or by DAC (IC15). If DAC is configured from *LimeSuiteGUI* software, then frequency synthesizer is shut down and VCTCXO tuning voltage is supplied from DAC. When DAC controls VCTCXO, FPGA2 LED is off.

DAC can be configured using *LimeSuiteGUI* software. Go to **Modules** form top menu and select **Board controls** form the drop down menu, as shown in *Figure* 47.



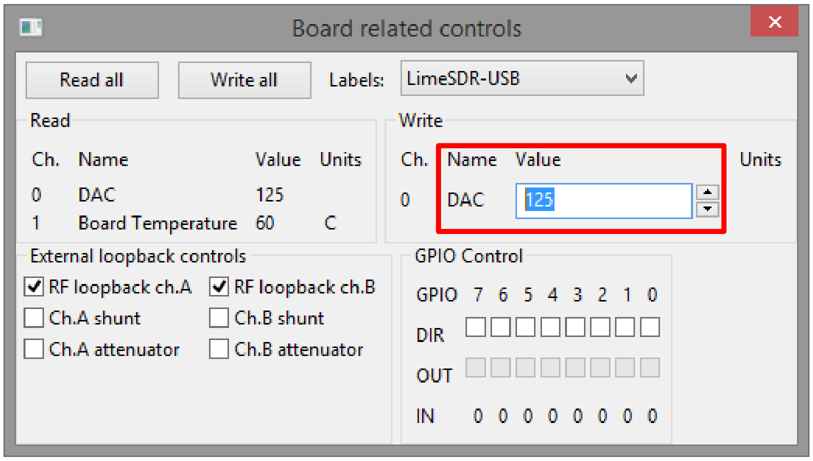
☑ Enable TxTS

Bypass

☐ DC correcto
☐ Gain correct
☐ Phase corre
☐ CMIX
☐ ISINC
☑ GFIR1
☑ GFIR2
☑ GFIR3

Figure 47 LimeSuiteGUI module menu to select DAC configuration tool

New window will appear, as shown in the *Figure* 48.



Labels: LimeSDR-USB

Read

Ch.	Name	Value	Units
0	DAC	125	
1	Board Temperature	60	C

Write

Ch.	Name	Value	Units
0	DAC	125	

External loopback controls

☑ RF loopback ch.A ☑ RF loopback ch.B
☐ Ch.A shunt ☐ Ch.B shunt
☐ Ch.A attenuator ☐ Ch.B attenuator

IN 0 0 0 0 0 0 0 0

Figure 48 Board related controls

Current DAC value can be read by pressing button **Read all**. DAC value will be displayed in group box **Read** with channel 0. In this case DAC value is 125.

Enter new DAC value in group box Write channel 0. After this press **Write all** and DAC value will be updated.

3.12.4 VCTCXO Calibration Procedure

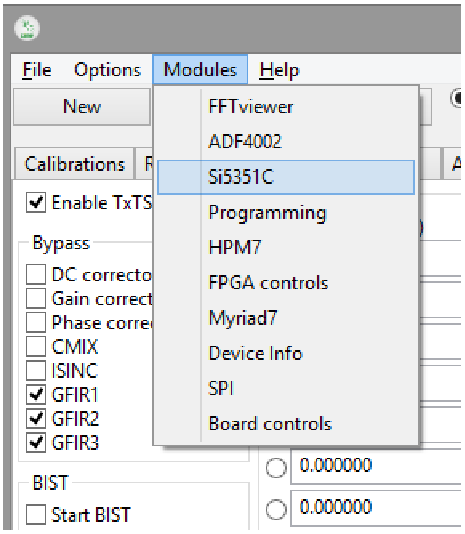
Board has VCTCXO DAC factory calibration value that is stored in non-volatile memory. This value is loaded to DAC output after each board power up or reset. This value can be changed manually or by automatic calibration procedure. To perform automatic calibration procedure, connect external reference clock to connector J19, start it from *LimeSuiteGUI* software and enter Fref frequency (default Fref value 10MHz).

Calibration procedure steps:

1. Lock phase detector (ADF4002) to external Fref clock as described in section **Error! Reference source not found. Error! Reference source not found.**. If phase detector cannot lock to reference clock, calibration procedure cannot continue and will be aborted.
2. Measure TCVCXO frequency and store for future comparison.
3. Start changing TCXO DAC value and detect when VCTCXO frequency is as close as possible to the value measured in step 2.
4. Store new VCTCXO value in non-volatile memory.

3.12.5 Programmable Clock Generator (Si5351C) Configuration

Programmable clock generator has eight channels and each can be configured individually using "*LimeSuiteGUI"*software. Go to **Modules** form top menu and select **Si5351C** form the drop down menu, as shown in *Figure 49.*



☑ Enable TxTS

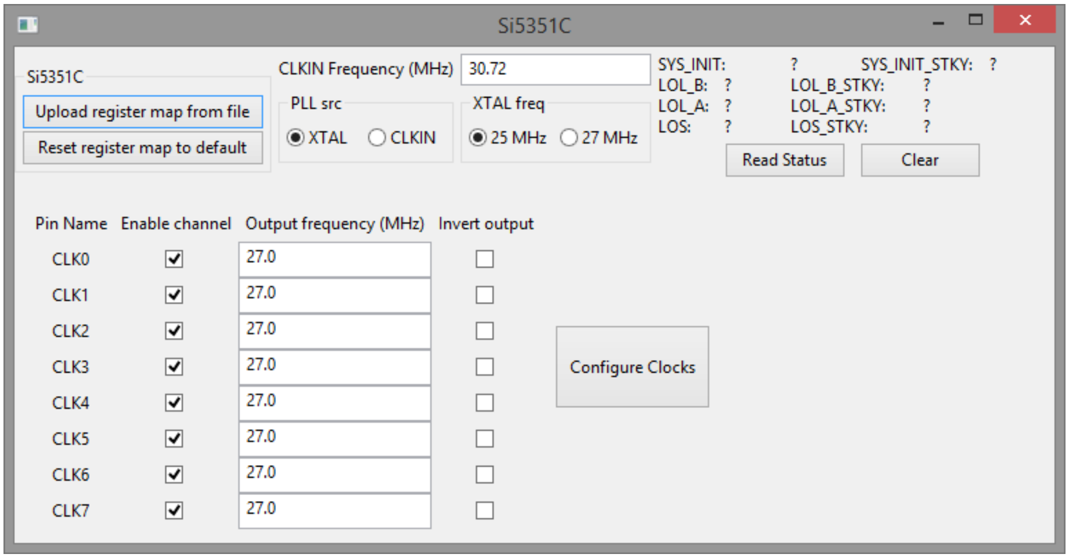
Bypass
☐ DC correcto
☐ Gain correct
☐ Phase corre
☐ CMIX
☐ ISINC
☑ GFIR1
☑ GFIR2
☑ GFIR3

BIST
☐ Start BIST

○ 0.000000
○ 0.000000

Figure 49 LimeSuiteGUI module menu to select Si5351C configuration tool

New window will appear, as shown in the Figure 50.



CLKIN Frequency (MHz) 30.72

PLL src: ● XTAL ○ CLKIN

XTAL freq: ● 25 MHz ○ 27 MHz

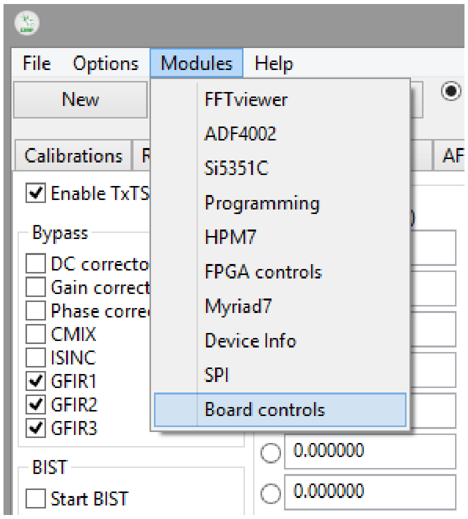
SYS_INIT: ? SYS_INIT_STKY: ?
LOL_B: ? LOL_B_STKY: ?
LOL_A: ? LOL_A_STKY: ?
LOS: ? LOS_STKY: ?

Pin Name	Enable channel	Output frequency (MHz)	Invert output
CLK0	☑	27.0	□
CLK1	☑	27.0	□
CLK2	☑	27.0	□
CLK3	☑	27.0	□
CLK4	☑	27.0	□
CLK5	☑	27.0	□
CLK6	☑	27.0	□
CLK7	☑	27.0	□

Figure 50 Si5351C configuration window

3.13 Reading Board Temperature

LimeSDR-USB has integrated temperature sensor. The sensor measured temperature may be displayed in software. Go to **Modules** form top menu and select **Board controls** form the drop down menu, as shown in *Figure 51*.



☑ Enable TxTS

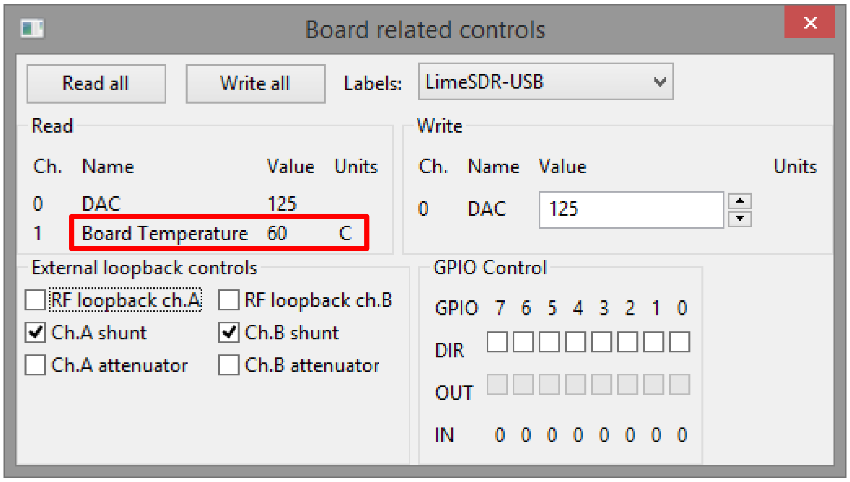
□ DC correcto
□ Gain correct
□ Phase corre
□ CMIX
□ ISINC
☑ GFIR1
☑ GFIR2
☑ GFIR3

□ Start BIST

○ 0.000000
○ 0.000000

Figure 51 LimeSuiteGUI module menu to select temperature tool

New window appears, as shown in the *Figure 52*.



Labels: LimeSDR-USB

Read

Ch.	Name	Value	Units
0	DAC	125	
1	Board Temperature	60	C

Write

Ch.	Name	Value	Units
0	DAC	125	

External loopback controls

☐ RF loopback ch.A ☐ RF loopback ch.B
☑ Ch.A shunt ☑ Ch.B shunt
☐ Ch.A attenuator ☐ Ch.B attenuator

GPIO Control

GPIO	7	6	5	4	3	2	1	0
DIR	☐	☐	☐	☐	☐	☐	☐	☐
OUT	☐	☐	☐	☐	☐	☐	☐	☐
IN	0	0	0	0	0	0	0	0

Figure 52 Board related controls, temperature section

Current temperature sensor value can be read by pressing button **Read all**. Temperature value will be displayed in group box **Read** with channel 1. In this case temperature is 60 °C.

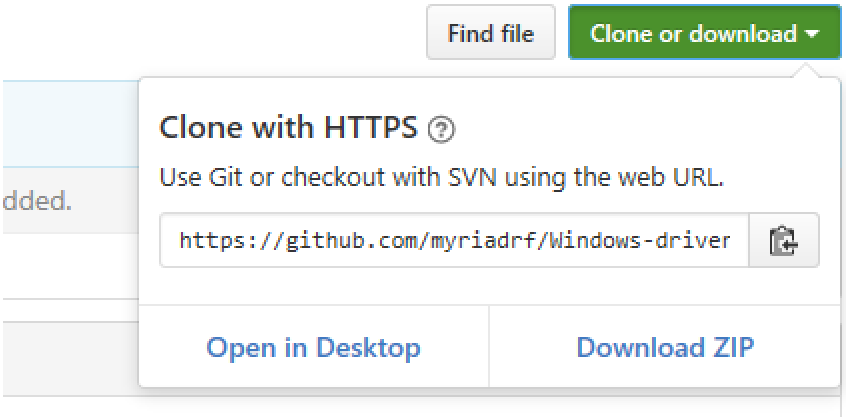
4. Drivers Installation

The communication between LimeSDR-QPCIe board and PC can done via the USB3 interface. Initially, LimeSDR-QPCIe board comes with preprogramed FX3 firmware and ready to use. If FX3 firmware needs to be updated, follow chapter 5.1.

This chapter guides through the USB3 driver installation for the LimeSDR- QPCIe board under Windows and Linux operating systems.

4.1 Windows USB Driver Installation Procedure

Download the latest drivers [here], select *Clone or Download and then Download ZIP* as shown in Figure 53. The name of the archive will be *Windows-drivers-master.zip* by default. Extract the archive.



Clone with HTTPS

Use Git or checkout with SVN using the web URL.

```
https://github.com/myriadrf/Windows-driver
```

Figure 53 Driver download from GiHub

First time LimeSDR- QPCIe board is connected to the PC, follow the installation procedure below.

1. Press **Start Menu** and right click on **Computer**, select **Properties** and **Device Manager**.

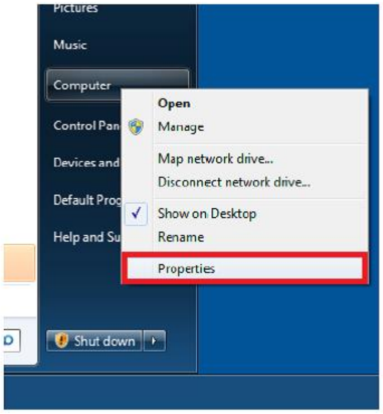


Figure 54 Open computer properties

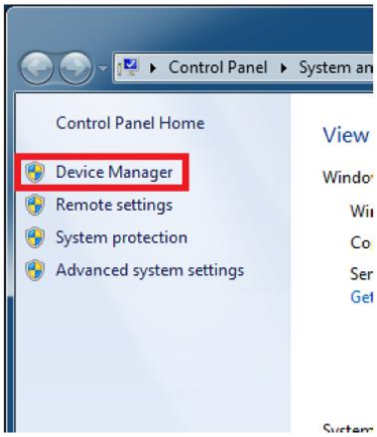


Figure 55 Open device manager

2. When LimeSDR- QPCIe board is plugged in, in **Device Manager** it appears as **LimeSDR-QPCIe** under **Other devices**. Right click on the **LimeSDR- QPCIe** and select **Update Driver Software**.

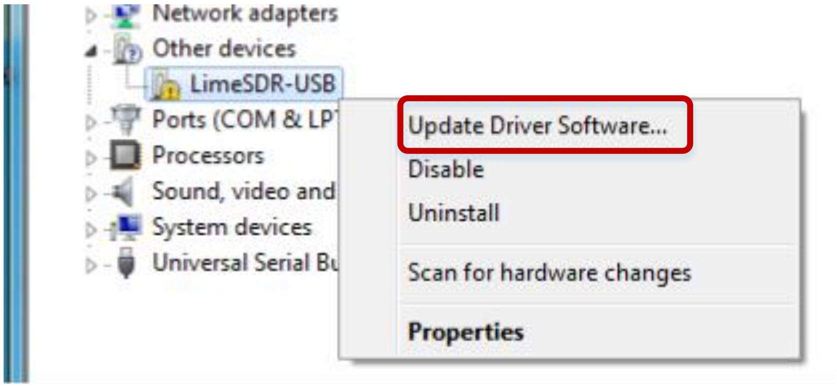


Figure 56 Update driver software

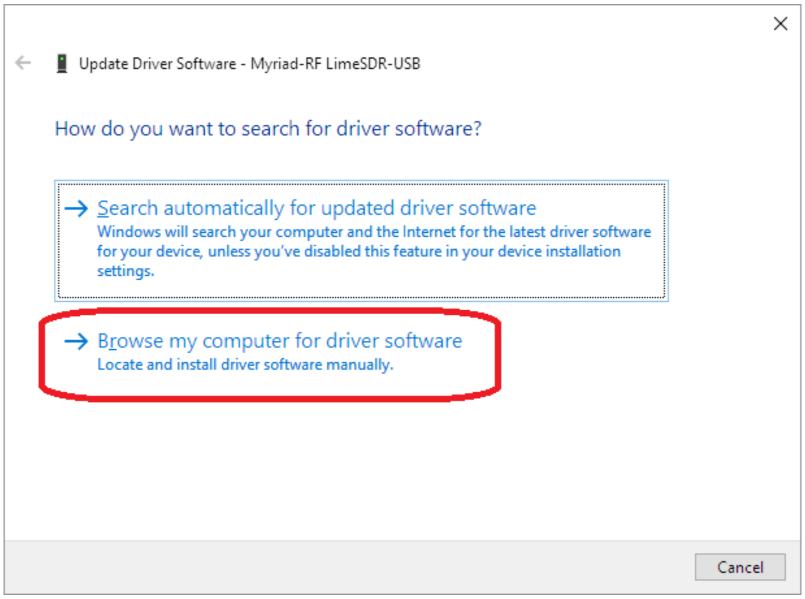
3. Select driver installation manually and choose driver from downloaded package (Windows-drivers-master\WinDriver_LimeSDR- QPCIe).

Choose the driver which is suitable for the operating system running:

- Windows XP (wxp)
- Windows Vista (vista)
- Windows 7 (win7)
- Windows 8 (win8)
- Windows 8.1 (win81)

OS version:

- x86 (32bit-i386)
- x64 (64bit-amd64)



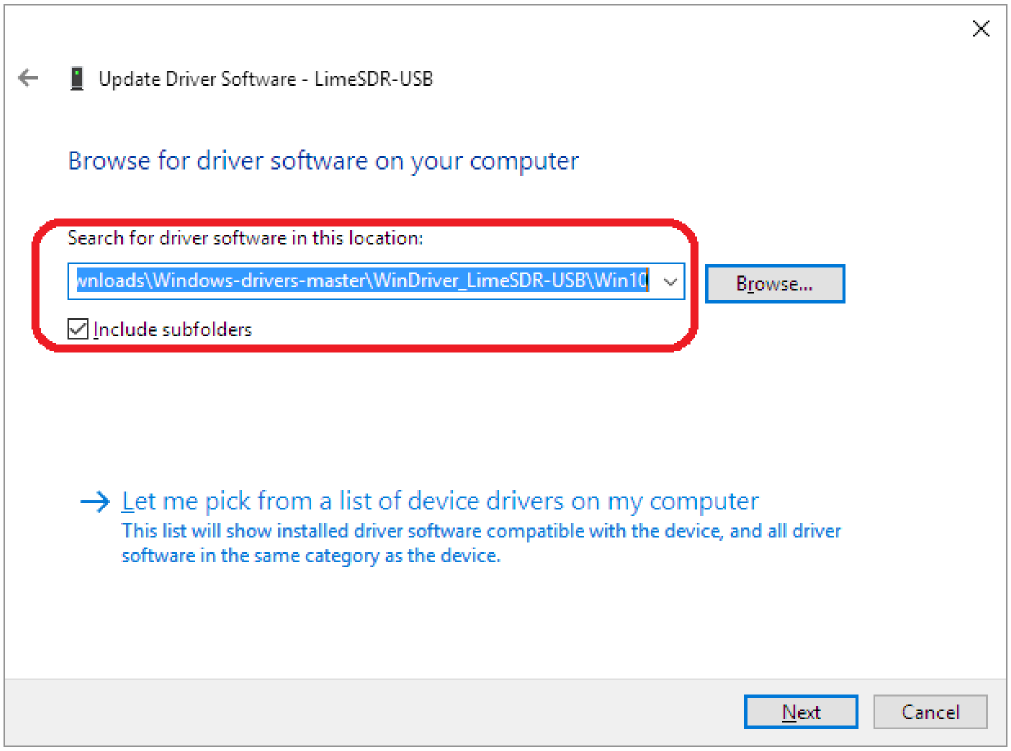
Update Driver Software - Myriad-RF LimeSDR-USB

How do you want to search for driver software?

→ Search automatically for updated driver software
Windows will search your computer and the Internet for the latest driver software for your device, unless you've disabled this feature in your device installation settings.

→ Browse my computer for driver software
Locate and install driver software manually.

Figure 57 Browse for driver software



Update Driver Software - LimeSDR-USB

Browse for driver software on your computer

Search for driver software in this location:

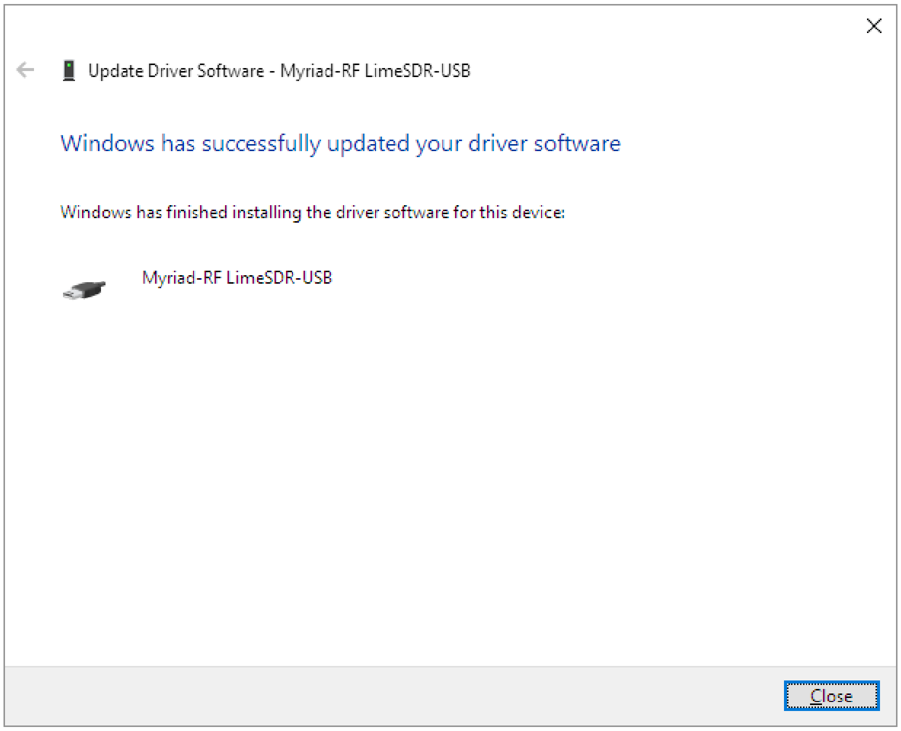
wnloads\Windows-drivers-master\WinDriver_LimeSDR-USB\Win10

☑ Include subfolders

Browse...
→ Let me pick from a list of device drivers on my computer
This list will show installed driver software compatible with the device, and all driver software in the same category as the device.
Next
Cancel

Figure 58 Select driver location

4. After successful installation *"Myriad-RF LimeSDR-* QPCIe*"* will appear under USB controller devices.



Update Driver Software - Myriad-RF LimeSDR-USB

Windows has successfully updated your driver software

Windows has finished installing the driver software for this device:

Myriad-RF LimeSDR-USB

Figure 59 Successful LimeSDR- QPCIe installation

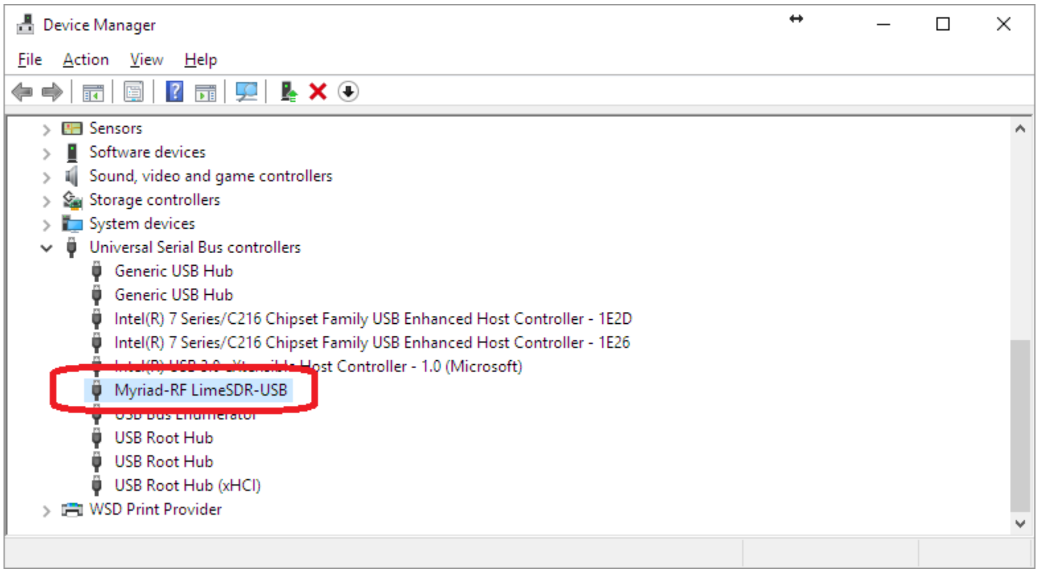

Device Manager
File
Action
View
Help
Sensors
Software devices
Sound, video and game controllers
Storage controllers
System devices
Universal Serial Bus controllers
Generic USB Hub
Generic USB Hub
Intel(R) 7 Series/C216 Chipset Family USB Enhanced Host Controller - 1E2D
Intel(R) 7 Series/C216 Chipset Family USB Enhanced Host Controller - 1E26
Intel(R) USB 3.0 eXtensible Host Controller - 1.0 (Microsoft)
Myriad-RF LimeSDR-USB
USB Bus Enumerator
USB Root Hub
USB Root Hub
USB Root Hub (xHCI)
WSD Print Provider

Figure 60 Device manager window after installation

4.2 Linux USB Drivers

No need to install USB3 drivers for Linux operating system, while it comes with libusb library.

4.3 Windows PCIe driver installation procedure

Download the latest drivers [here], select xillybus-windriver-1.2.0.0.zip package and unzip. First time LimeSDR-QPCIe board is connected to the PC, follow the installation procedure below.

5. Press **Start Menu** and right click on **Computer**, select **Properties** and **Device Manager**.

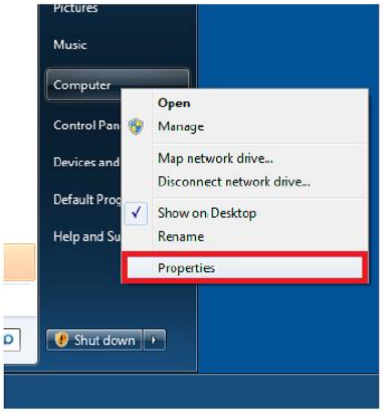


Figure 61 Open computer properties

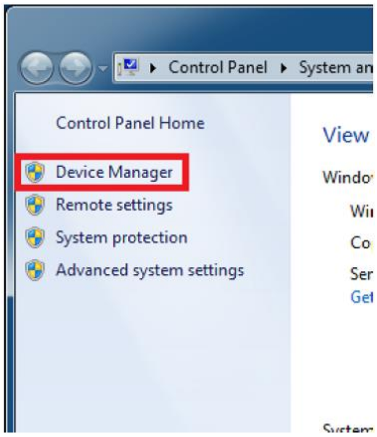


Figure 62 Open device manager

6. When LimeSDR-QPCIe board is plugged in, in **Device Manager** it appears as **PCI Device** under **Other devices**. Right click on the **PCI device** and select **Update Driver Software** Figure 63.

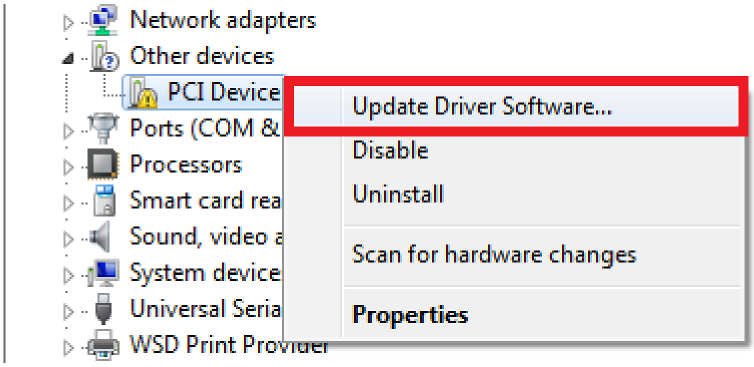
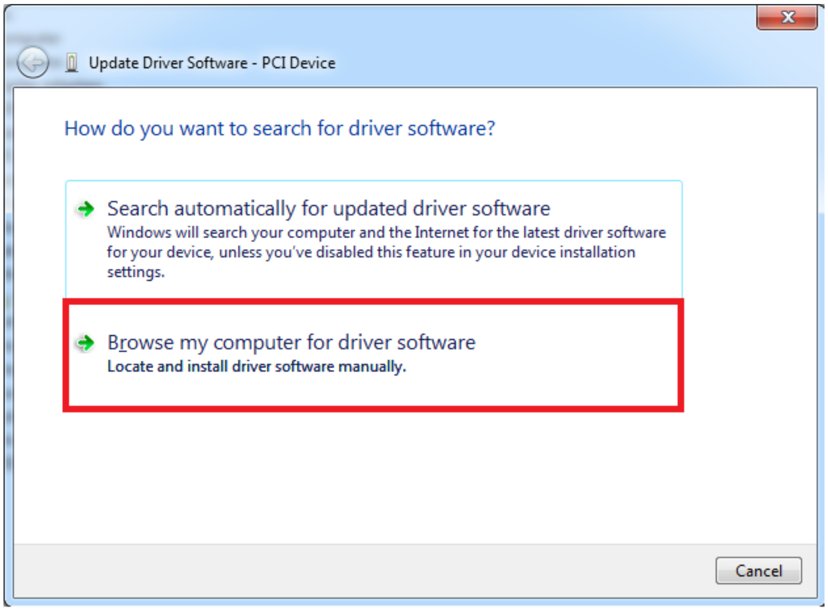


Figure 63 Update driver software

7. Select Browse my computer for driver software (Figure 64) and in browse window (Figure 65) choose driver from downloaded package (extracted files from xillybus-windriver-1.2.0.0.zip).



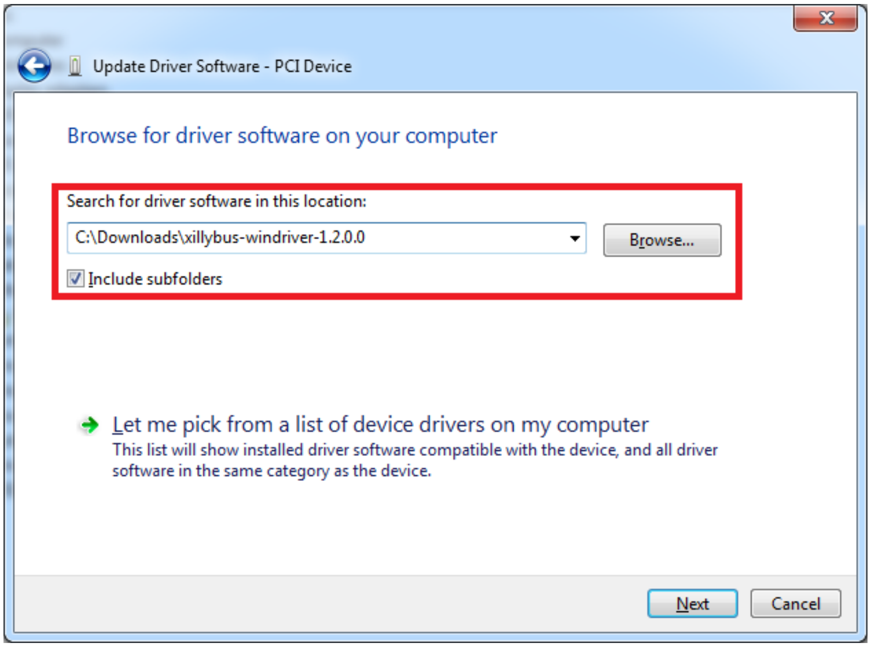
Update Driver Software - PCI Device

How do you want to search for driver software?

Search automatically for updated driver software
Windows will search your computer and the Internet for the latest driver software for your device, unless you've disabled this feature in your device installation settings.

Browse my computer for driver software
Locate and install driver software manually.

Figure 64 Browse for driver software



Update Driver Software - PCI Device

Browse for driver software on your computer

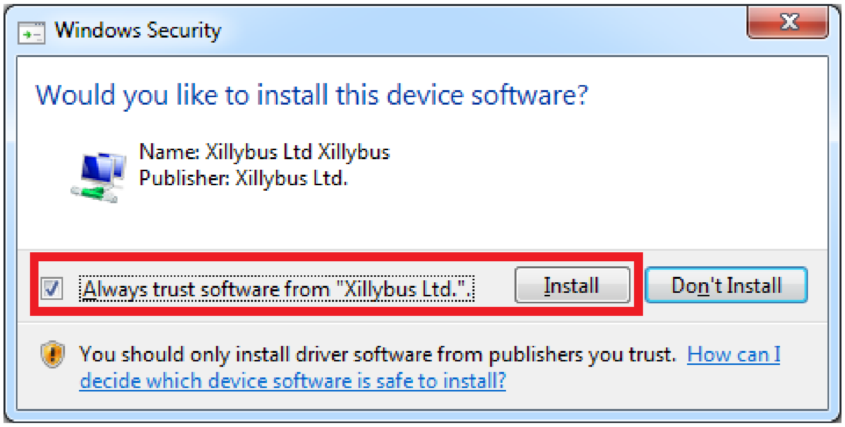
Search for driver software in this location:
C:\Downloads\xillybus-windriver-1.2.0.0

☑ Include subfolders

Let me pick from a list of device drivers on my computer
This list will show installed driver software compatible with the device, and all driver software in the same category as the device.

Figure 65 Select driver location

8. After selecting driver files and clicking Next button Windows security warning might appear, check Always trust software from “Xillybus Ltd” and click Install.



Windows Security

Would you like to install this device software?

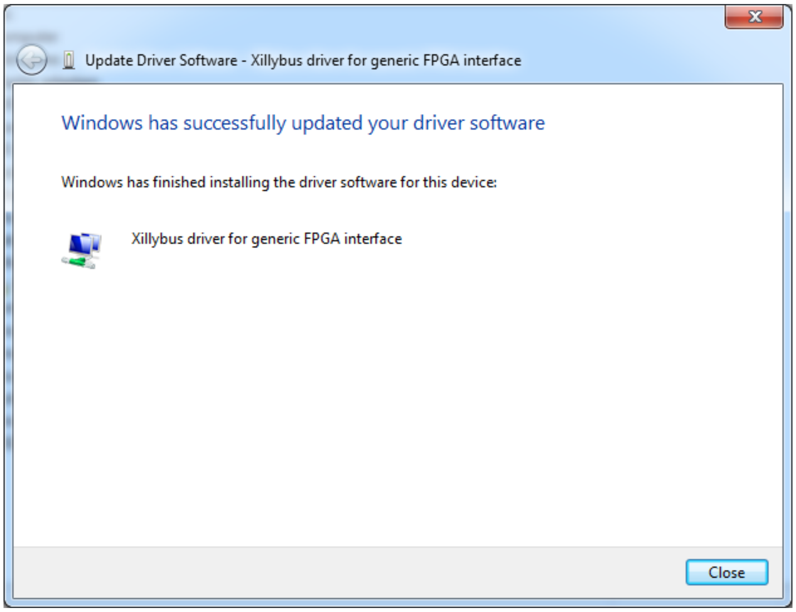
Name: Xillybus Ltd Xillybus
Publisher: Xillybus Ltd.

☑ Always trust software from "Xillybus Ltd.".

You should only install driver software from publishers you trust. How can I decide which device software is safe to install?

Figure 66 Windows security warning

9. After successful installation (Figure 67) *"Xillybus driver for generic FPGA interface"* will appear under Xillybus device (Figure 68).



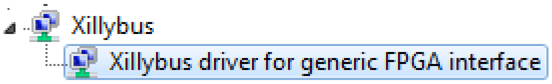
Update Driver Software - Xillybus driver for generic FPGA interface

Windows has successfully updated your driver software

Windows has finished installing the driver software for this device:

Xillybus driver for generic FPGA interface

Figure 67 Successful LimeSDR-PCIe installation



Xillybus
Xillybus driver for generic FPGA interface

Figure 68 Device manager window after installation

4.4 Linux PCIe drivers

No need to install PCIe drivers for Linux operating system.

5. LimeSDR-QPCIe Board Programming

This section describes how to program USB3 Microcontroler (FX3) and FPGA. FX3 can be programmed with Cypress tools (see **5.2 Updating USB3 Microcontroller Firmware in Bootloader Mode**) or LimeSuiteGUI (see **5.1 Updating FX3 Firmware Using LimeSuiteGUI**). This section describes also how to obtain FPGA programming file and program FPGA using LimeSuiteGUI (see **5.4.1 Uploading FPGA Gateware to FLASH Memory using LimeSuiteGUI**) or JTAG cable (see **5.4.2 Uploading FPGA gateware to FLASH memory using JTAG Cable**).

5.1 Updating FX3 Firmware Using LimeSuiteGUI

The firmware of FX3 MCU contains a functionality which enables to program FLASH memory FX3 MCU boots up from. In this case FX3 USB controller firmware can be updated using "*LimeSuiteGUI"* software, when FX3 MCU can boot from FLASH memory.

To call FPGA programing function, launch LimeSuiteGUI and connect to the board (see "3.1 Launching LimeSuiteGUI and Connecting to the LimeSDR-USB Board" section for more information). Then go to **Modules** from main menu and select **Programing** form the drop down menu, as shown in Figure 69.

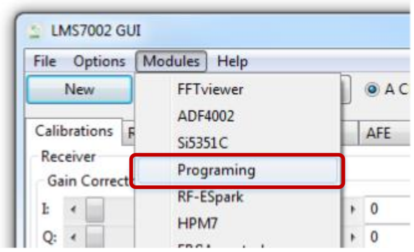
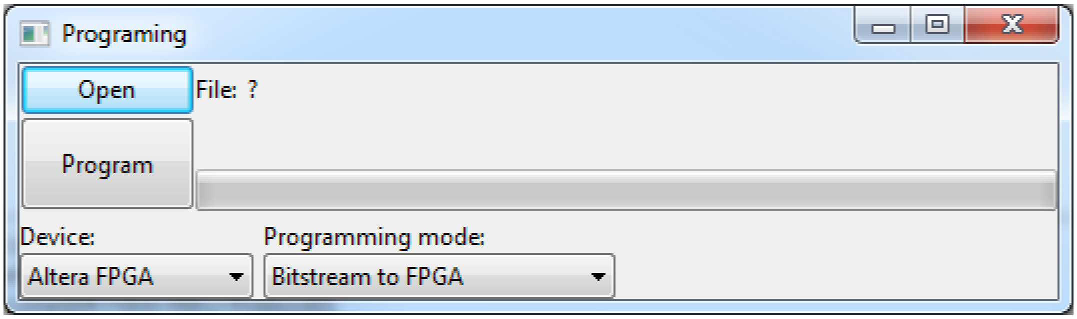


Figure 69 LimeSuiteGUI module menu to select FX3 programing tool

New window appears, as shown in the Figure 70.



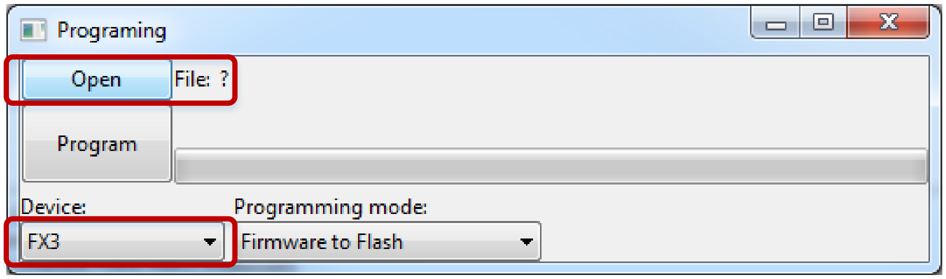
File: ?

Device: Altera FPGA

Programming mode: Bitstream to FPGA

Figure 70 Programing tool interface

Change device to "FX3" and press "Open" and select firmware image file.



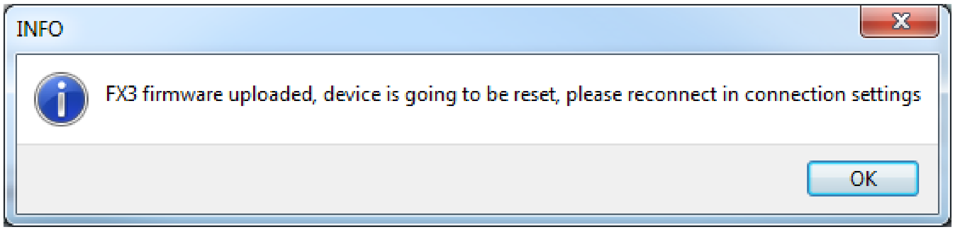
Device: FX3

Programming mode: Firmware to Flash

Figure 71 FX3 programing options

Initiate FLASH memory programing by clicking **Program**.

The new message will come up when programing is finished, as shown in Figure 72.



FX3 firmware uploaded, device is going to be reset, please reconnect in connection settings

Figure 72 Successfully FX3 programing message

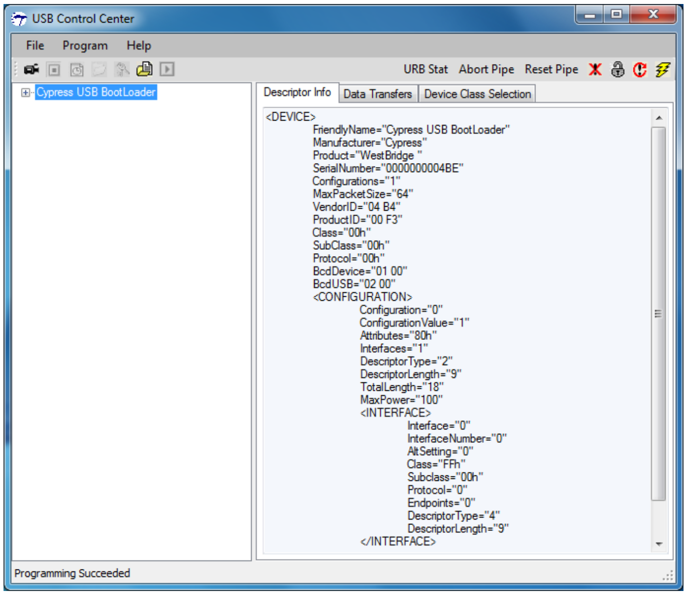
After successful firmware update, connect to the LimeSDR-USB board again as described in section 3.1.

5.2 Updating USB3 Microcontroller Firmware in Bootloader Mode

Cypress FX3 USB microcontroller has an integrated boot loader, which starts automatically after power-up or reset and when no valid firmware is present in the FLASH memory.

For USB microcontroller firmware upgrade, please use the *"CyControl.exe"* application from *cy_ssusbsuite_v1.3.3.zip* package which may be downloaded [here].

If FLASH memory is empty or connector J17 (on LimeSDR-USB board) is open, USB3 microcontroller boots-up into bootloader mode. Cypress drivers from *cy_ssusbsuite_v1.3.3.zip* package must be installed first. Run the *"USB Control Center"* application and select **Cypress USB BootLoader** line as shown in *Figure 73.*



```
<DEVICE>
        FriendlyName="Cypress USB BootLoader"
        Manufacturer="Cypress"
        Product="WestBridge "
        SerialNumber="0000000004BE"
        Configurations="1"
        MaxPacketSize="64"
        VendorID="04 B4"
        ProductID="00 F3"
        Class="00h"
        SubClass="00h"
        Protocol="00h"
        BcdDevice="01 00"
        BcdUSB="02 00"
        <CONFIGURATION>
                Configuration="0"
                ConfigurationValue="1"
                Attributes="80h"
                Interfaces="1"
                DescriptorType="2"
                DescriptorLength="9"
                TotalLength="18"
                MaxPower="100"
                <INTERFACE>
                        Interface="0"
                        InterfaceNumber="0"
                        AltSetting="0"
                        Class="FFh"
                        Subclass="00h"
                        Protocol="0"
                        Endpoints="0"
                        DescriptorType="4"
                        DescriptorLength="9"
                </INTERFACE>
```

Programming Succeeded

Figure 73 Default FX3 firmware, supplied by FX3 internal logic

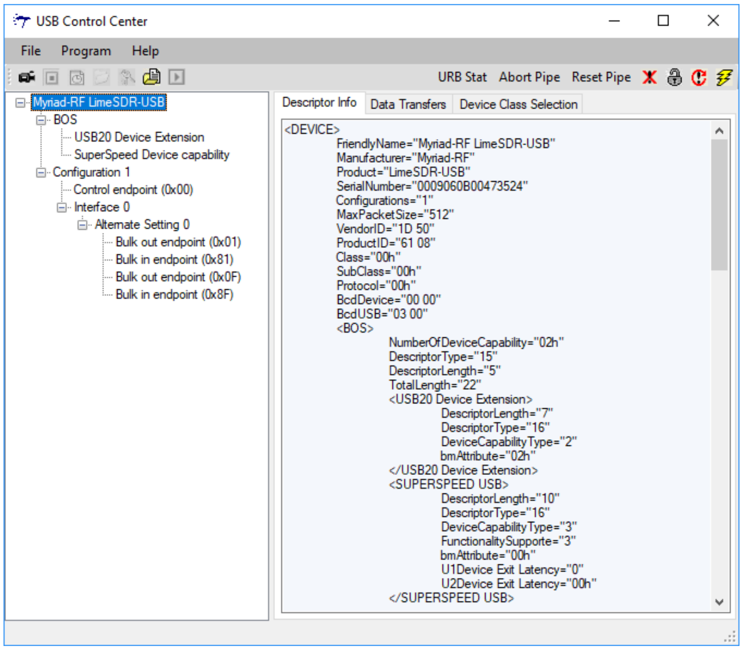
After entering into boot loader mode, there are two ways of uploading the firmware to USB3 microcontroller:

- Program external SPI FLASH memory connected to USB3 controller. Follow procedure described in chapter *"5.2.1 Uploading Firmware to SPI FLASH"*. The USB3 microcontroller will boot from FLASH memory after every power-on.
- Program internal RAM memory. Follow procedure described in chapter *"5.2.2 Uploading Firmware to the FX3 RAM"*. The memory will be cleared after first power cycle hence this step should be used for test purposes only.

5.2.1 Uploading Firmware to SPI FLASH Memory

Short the jumper J17 and connect LimeSDR-USB board to the PC. Start *"CyControl.exe"* application and select **Cypress USB BootLoader** as shown in *Figure 73*. Choose menu command **Program → FX3 → SPI FLASH**. In the status bar you will see **Waiting for Cypress Boot Programmer device to enumerate....** and after some time window will appear. Select firmware image file (file extension is "*.img") and press **Open**. Status bar of the **USB Control Center** application will indicate **Programming of SPI FLASH in Progress….** This message will change to the **Programming succeeded** after FLASH programming is done.

If you expand **Cypress USB StreamerExample** line in **USB Control Center** application now, you will see different USB configuration as shown in *Figure 74*.



```
<DEVICE>
        FriendlyName="Myriad-RF LimeSDR-USB"
        Manufacturer="Myriad-RF"
        Product="LimeSDR-USB"
        SerialNumber="0009060B00473524"
        Configurations="1"
        MaxPacketSize="512"
        VendorID="1D 50"
        ProductID="61 08"
        Class="00h"
        SubClass="00h"
        Protocol="00h"
        BcdDevice="00 00"
        BcdUSB="03 00"
        <BOS>
                NumberOfDeviceCapability="02h"
                DescriptorType="15"
                DescriptorLength="5"
                TotalLength="22"
                <USB20 Device Extension>
                        DescriptorLength="7"
                        DescriptorType="16"
                        DeviceCapabilityType="2"
                        bmAttribute="02h"
                </USB20 Device Extension>
                <SUPERSPEED USB>
                        DescriptorLength="10"
                        DescriptorType="16"
                        DeviceCapabilityType="3"
                        FunctionalitySupporte="3"
                        bmAttribute="00h"
                        U1Device Exit Latency="0"
                        U2Device Exit Latency="00h"
                </SUPERSPEED USB>
```

Figure 74 FX3 after custom firmware is downloaded

<u>NOTE</u>: USB3 microcontroller will boot firmware uploaded to FLASH each time after power-on if jumper J17 is shorted.

5.2.2 Uploading Firmware to the FX3 RAM

Start *"CyControl.exe*" application and select **Cypress USB BootLoader** as shown in *Figure 73*. Choose menu command **Program → FX3 → RAM**. In the new pop-up window, select firmware image file (file extension is "*.img") and press **Open**. Status bar of the **USB Control Center** application will indicate **Programming RAM**. This message will change to the **Programming succeeded** after programming is done.

Note please that this may be used for test purposes only, while firmware will disappear from the RAM after LimeSDR-USB board power cycle.

5.3 Obtaining FPGA programming files

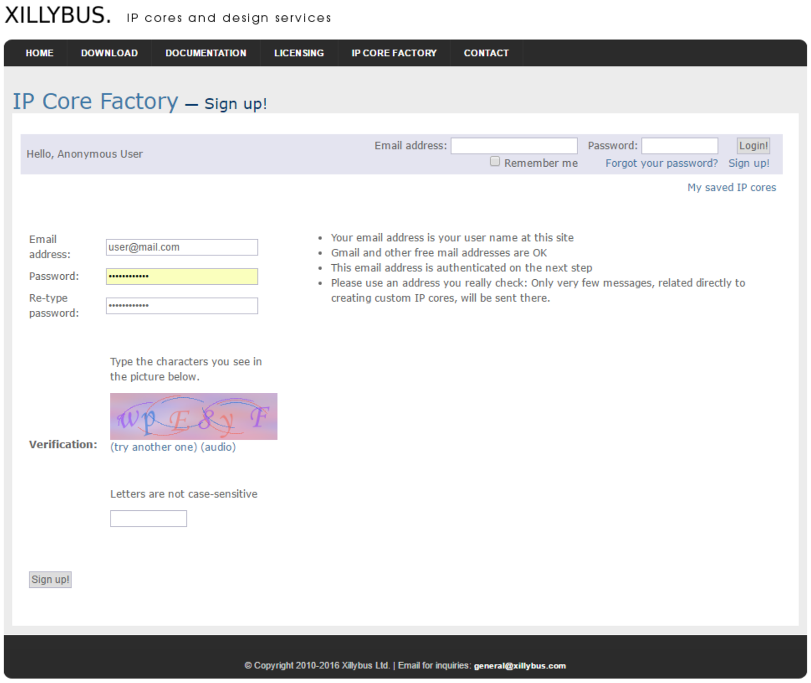
FPGA gateware programming file can be obtained by compiling provided LimeSDR-QPCIE_lms7_trx project with Intel Quartus Prime software. Software version used with this guide: Quartus prime 15.1.2 Build 193 02/01/2016 SJ Lite Edition. Quartus Prime Lite Edition software can be downloaded from [here].

5.3.1 PCIe core generation

PCIe Xillybus core has to be generated and downloaded in order to compile LimeSDR-QPCIE_lms7_trx FPGA project. This chapter describes steps and parameters required to generate Xillybus PCIe core.

5.3.1.1 Signing UP

Xillybus requires to fill up free registration form in order to download generated core. Go to [link], fill required fields (Figure 75) and confirm registration via received eMail.



IP Core Factory — Sign up!

Email address: user@mail.com

Password: ••••••••••••

Re-type password: ••••••••••••

- Your email address is your user name at this site
- Gmail and other free mail addresses are OK
- This email address is authenticated on the next step
- Please use an address you really check: Only very few messages, related directly to creating custom IP cores, will be sent there.

Type the characters you see in the picture below.

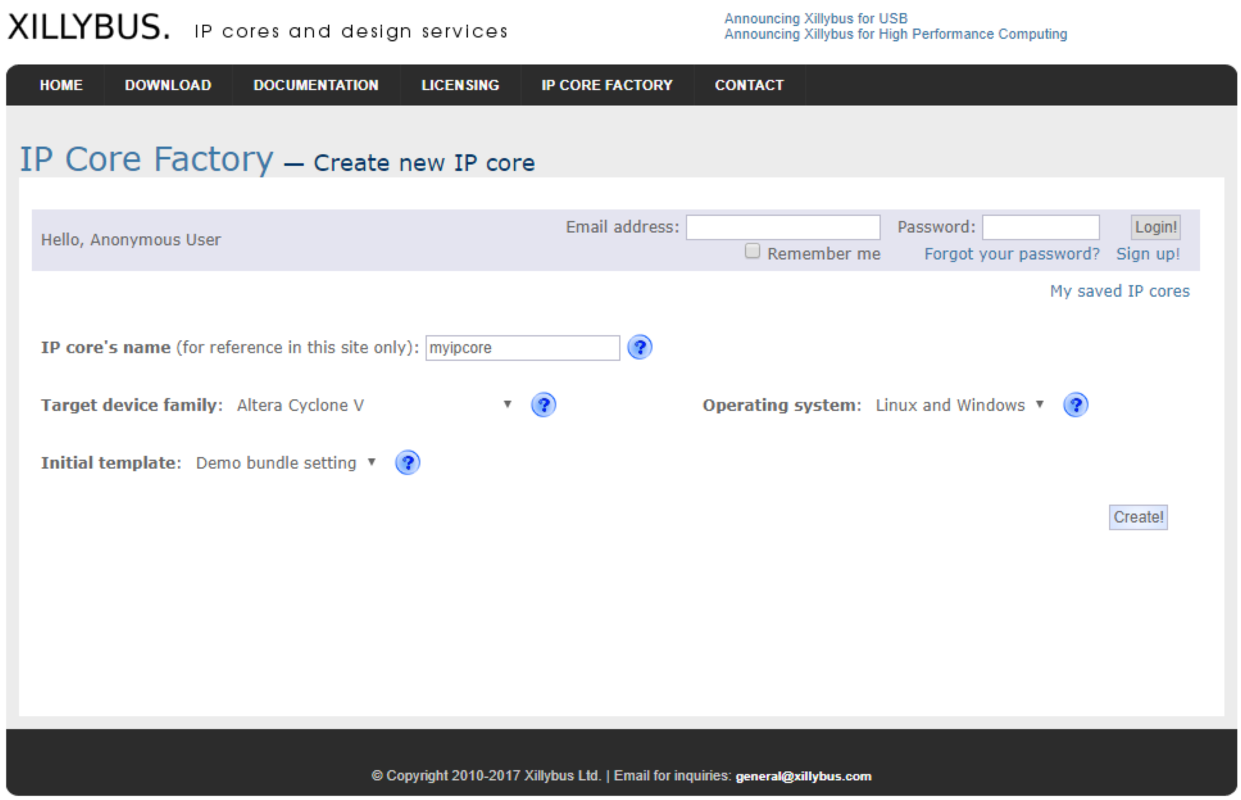
Verification:

Letters are not case-sensitive

Figure 75 Registration form

5.3.1.2 Creating new IP core

After successful registration, go to IP core Factory page [link] fill parameters as shown in Figure 76 and click *Create!*.


XILLYBUS. IP cores and design services
Announcing Xillybus for USB
Announcing Xillybus for High Performance Computing
HOME
DOWNLOAD
DOCUMENTATION
LICENSING
IP CORE FACTORY
CONTACT
IP Core Factory — Create new IP core
Hello, Anonymous User
Email address:
Password:
Login!
□ Remember me
Forgot your password?
Sign up!
My saved IP cores
?
Create!
© Copyright 2010-2017 Xillybus Ltd. | Email for inquiries: general@xillybus.com

IP core's name (for reference in this site only): myipcore

Target device family: Altera Cyclone V

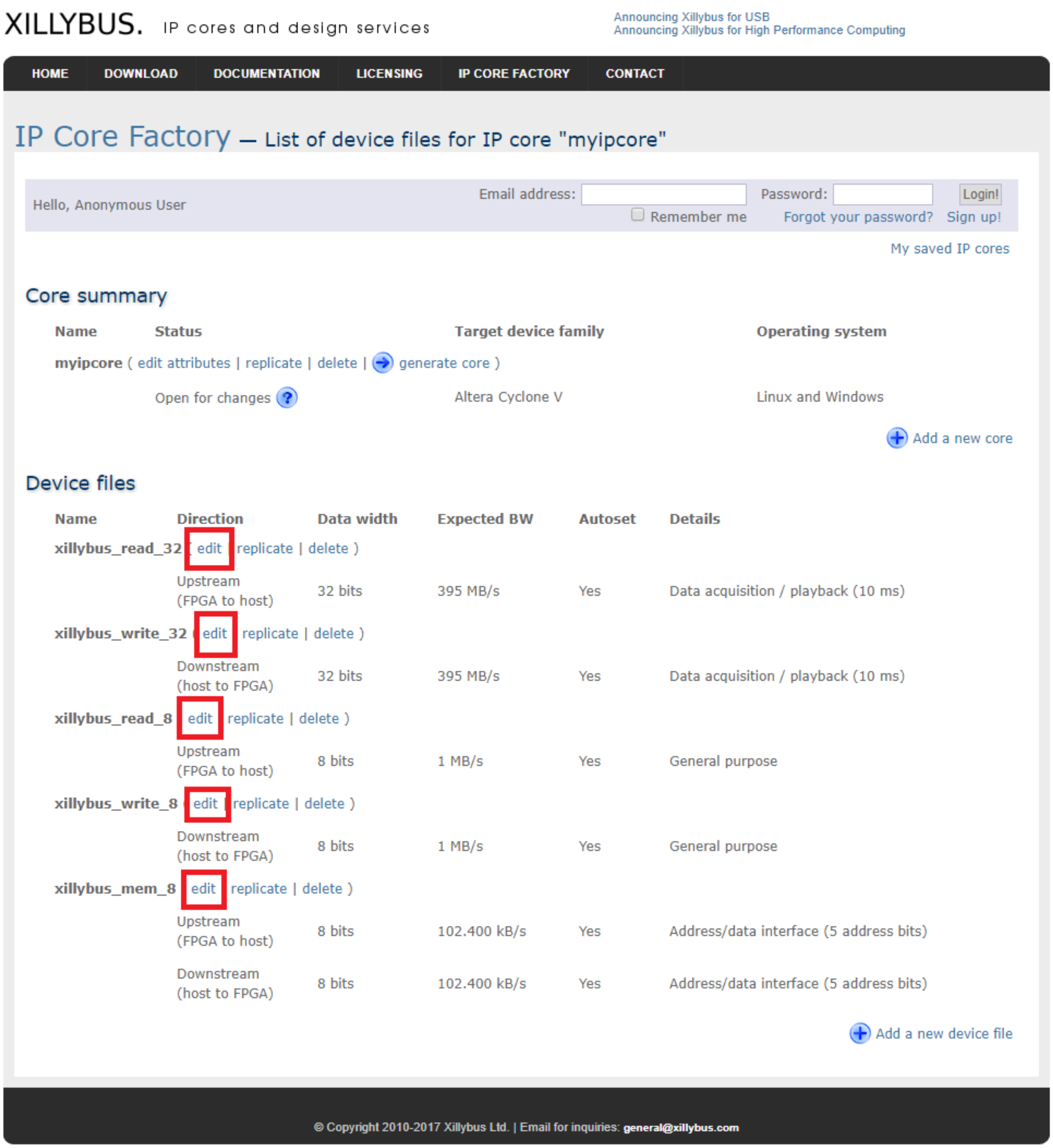
Operating system: Linux and Windows

Initial template: Demo bundle setting

Figure 76 Create new IP core dialog

5.3.1.3 Setting core parameters

After new core creation in next dialog click *Edit* to change settings for each device files (Figure 77).



IP Core Factory — List of device files for IP core "myipcore"

Hello, Anonymous User

Email address: ______ Password: ______

□ Remember me

Core summary

Name	Status	Target device family	Operating system
myipcore (edit attributes | replicate | delete | generate core)	Open for changes	Altera Cyclone V	Linux and Windows

Device files

Name	Direction	Data width	Expected BW	Autoset	Details
xillybus_read_32 (edit | replicate | delete)	Upstream (FPGA to host)	32 bits	395 MB/s	Yes	Data acquisition / playback (10 ms)
xillybus_write_32 (edit | replicate | delete)	Downstream (host to FPGA)	32 bits	395 MB/s	Yes	Data acquisition / playback (10 ms)
xillybus_read_8 (edit | replicate | delete)	Upstream (FPGA to host)	8 bits	1 MB/s	Yes	General purpose
xillybus_write_8 (edit | replicate | delete)	Downstream (host to FPGA)	8 bits	1 MB/s	Yes	General purpose
xillybus_mem_8 (edit | replicate | delete)	Upstream (FPGA to host)	8 bits	102.400 kB/s	Yes	Address/data interface (5 address bits)
	Downstream (host to FPGA)	8 bits	102.400 kB/s	Yes	Address/data interface (5 address bits)

© Copyright 2010-2017 Xillybus Ltd. | Email for inquiries: general@xillybus.com

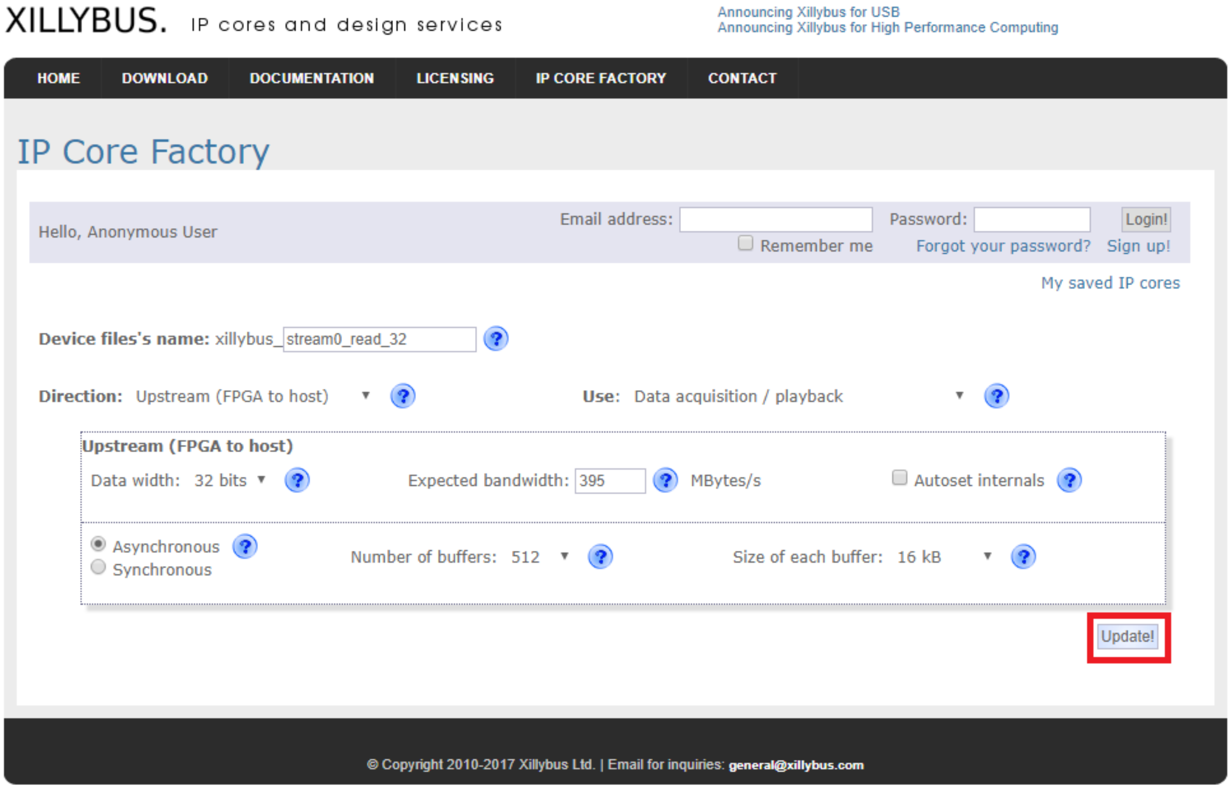
Figure 77 File editing

In *Edit* dialog (Figure 78) fill following parameters for corresponding file and click *Update!*. To enter all parameters *Autoset internals* has to be unchecked:

For xillybus_read_32:

Device file's name	- stream0_read_32
Direction	- Upstream (FPGA to host)
Use	- Data acquisition / playback
Data width	- 32 bits

Expected bandwidth	- 395
Autoset internals	- unchecked
Asynchronous/synchronous	- Asynchronous
Number of buffers	- 512
Size of each buffer	- 16 kB


XILLYBUS. IP cores and design services
Announcing Xillybus for USB
Announcing Xillybus for High Performance Computing
HOME
DOWNLOAD
DOCUMENTATION
LICENSING
IP CORE FACTORY
CONTACT
Login!
Forgot your password?
Sign up!
My saved IP cores
Update!
© Copyright 2010-2017 Xillybus Ltd. | Email for inquiries: general@xillybus.com

IP Core Factory

Hello, Anonymous User

Email address: Password: □ Remember me

Device files's name: xillybus_stream0_read_32

Direction: Upstream (FPGA to host)

Use: Data acquisition / playback

Upstream (FPGA to host)

Data width: 32 bits

Expected bandwidth: 395 MBytes/s

□ Autoset internals

● Asynchronous
○ Synchronous

Number of buffers: 512

Size of each buffer: 16 kB

Figure 78 xillybus_read_32 file editing

Edit rest of the files with following parameters:

For xillybus_write_32:

Device file's name	- stream0_write_32
Direction	- Downstream (host to FPGA)
Use	- Data acquisition / playback
Data width	- 32 bits
Expected bandwidth	- 395
Autoset internals	- unchecked
Asynchronous/synchronous	- Asynchronous
Number of buffers	- 512
Size of each buffer	- 16 kB
DMA acceleration	- 8 segments x 512 bytes

For xillybus_read_8:

Device file's name	- control0_read_32
Direction	- Upstream (FPGA to host)

- Use - General purpose
- Data width - 32 bits
- Expected bandwidth - 1
- Autoset internals - checked

For xillybus_write_8:

- Device file's name - control0_write_32
- Direction - Downstream (host to FPGA)
- Use - General purpose
- Data width - 32 bits
- Expected bandwidth - 1
- Autoset internals - checked

For xillybus_mem_8:

- Device file's name - mem_8
- Direction - Bidirectional
- Use - Address/data interface (5 address bits)

  Upstream (FPGA to host)
  - Data width - 8 bits
  - Expected bandwidth - 0.1
  - Autoset internals - checked

  Downstream (host to FPGA)
  - Data width - 8 bits
  - Expected bandwidth - 0.1
  - Autoset internals - checked

Next step is to add more device files, click *Add new device file* Figure 79 and add following device files with following parameters:

xillybus_stream1_read_32:

- Device file's name - stream1_read_32
- Direction - Upstream (FPGA to host)
- Use - Data acquisition / playback
- Data width - 32 bits
- Expected bandwidth - 395
- Autoset internals - unchecked
- Asynchronous/synchronous - Asynchronous
- Number of buffers - 512
- Size of each buffer - 16 kB

xillybus_ stream1_write_32**:**

- Device file's name - stream1_write_32
- Direction - Downstream (host to FPGA)
- Use - Data acquisition / playback
- Data width - 32 bits

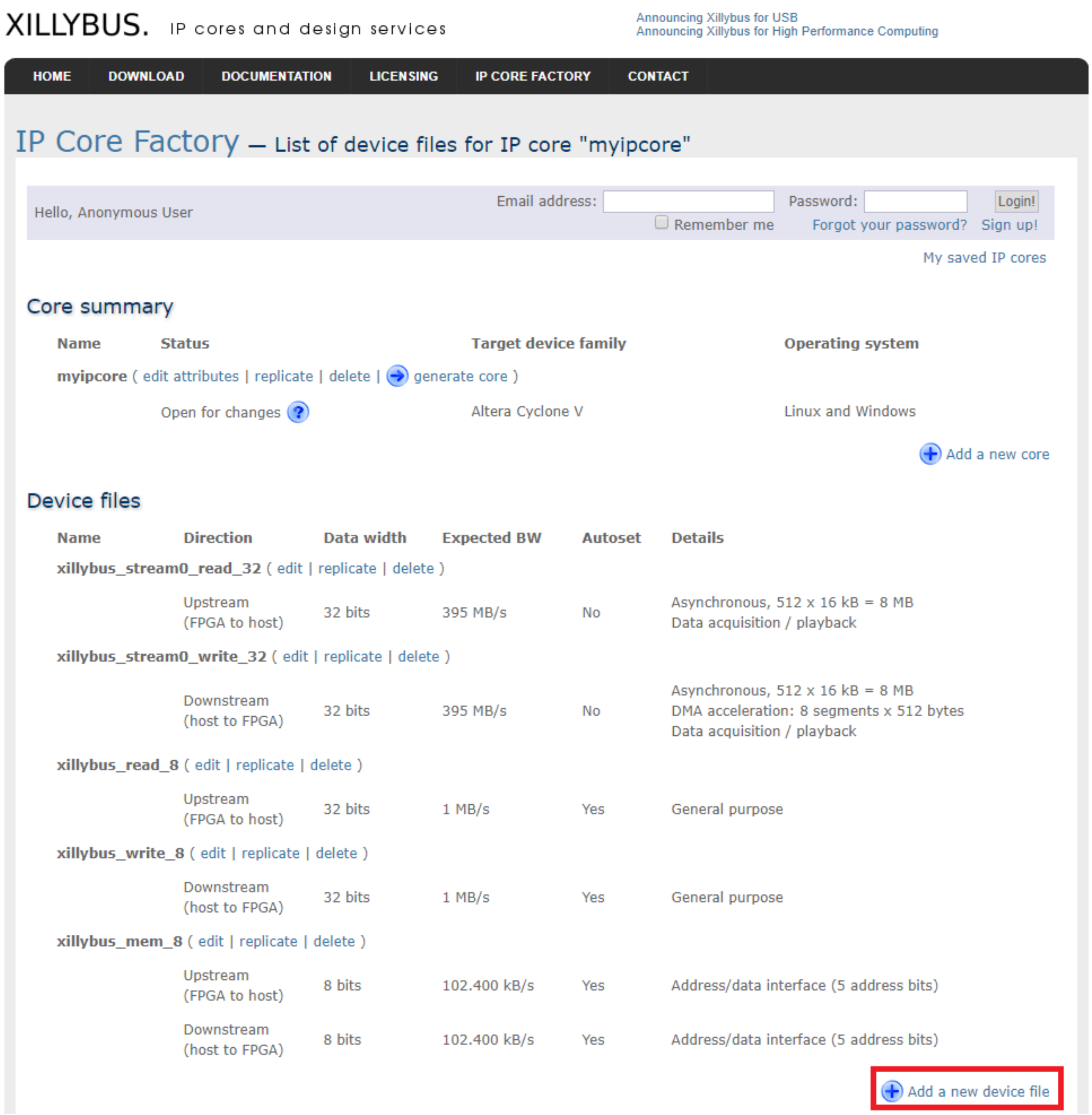
Expected bandwidth	- 395
Autoset internals	- unchecked
Asynchronous/synchronous	- Asynchronous
Number of buffers	- 512
Size of each buffer	- 16 kB
DMA acceleration	- 8 segments x 512 bytes

xillybus_stream2_read_32:

Device file's name	- stream2_read_32
Direction	- Upstream (FPGA to host)
Use	- Data acquisition / playback
Data width	- 32 bits
Expected bandwidth	- 395
Autoset internals	- unchecked
Asynchronous/synchronous	- Asynchronous
Number of buffers	- 512
Size of each buffer	- 16 kB

xillybus_stream2_write_32:

Device file's name	- stream2_write_32
Direction	- Downstream (host to FPGA)
Use	- Data acquisition / playback
Data width	- 32 bits
Expected bandwidth	- 395
Autoset internals	- unchecked
Asynchronous/synchronous	- Asynchronous
Number of buffers	- 512
Size of each buffer	- 16 kB
DMA acceleration	- 8 segments x 512 bytes


XILLYBUS. IP cores and design services
Announcing Xillybus for USB
Announcing Xillybus for High Performance Computing
HOME
DOWNLOAD
DOCUMENTATION
LICENSING
IP CORE FACTORY
CONTACT

IP Core Factory — List of device files for IP core "myipcore"

Hello, Anonymous User

Email address:
Password:
Login!

□ Remember me

Forgot your password? Sign up!

My saved IP cores

Core summary

Name	Status	Target device family	Operating system
myipcore (edit attributes | replicate | delete | generate core)			
	Open for changes ?	Altera Cyclone V	Linux and Windows

Add a new core

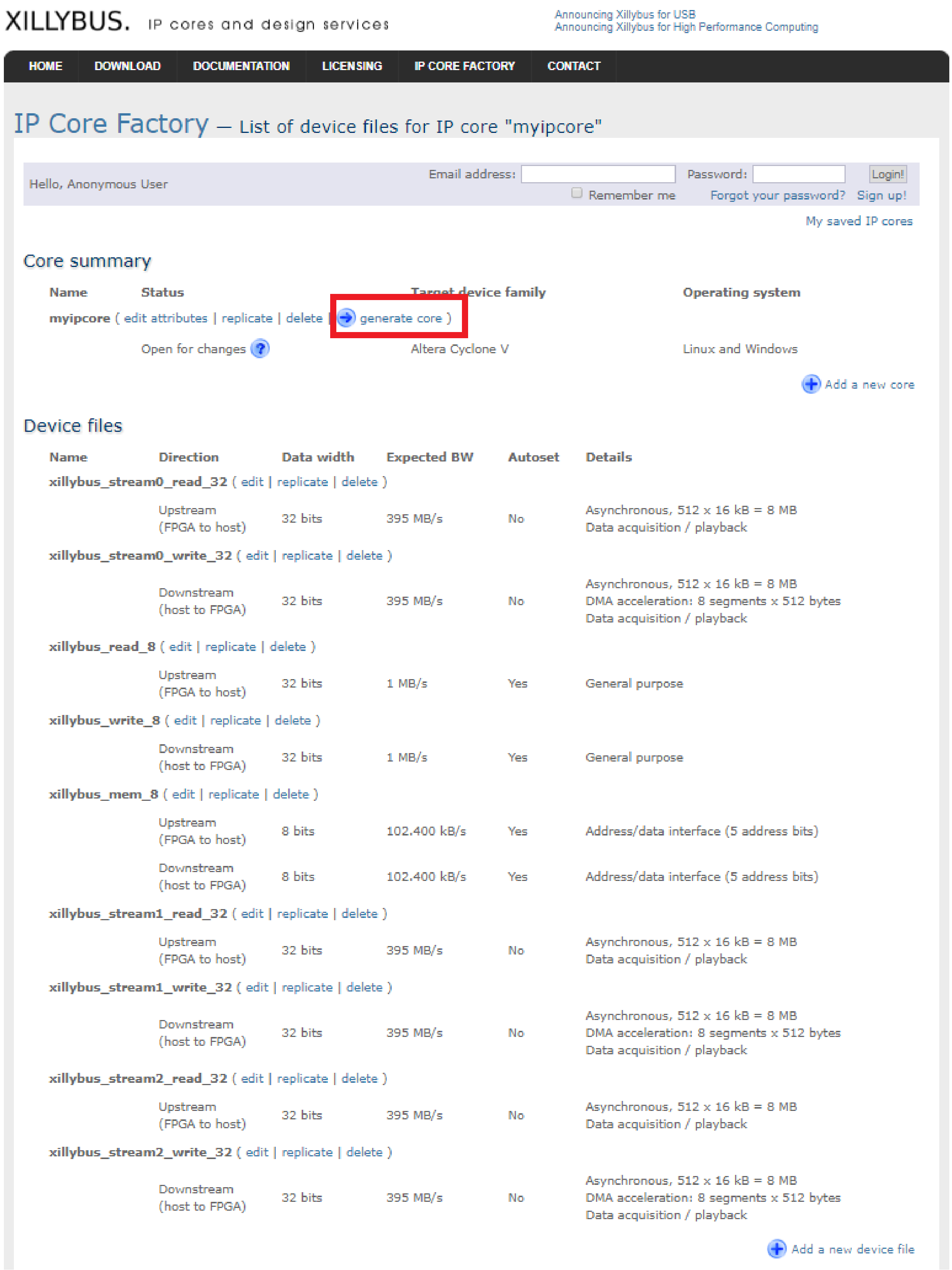
Device files

Name	Direction	Data width	Expected BW	Autoset	Details
xillybus_stream0_read_32 (edit | replicate | delete)					
	Upstream (FPGA to host)	32 bits	395 MB/s	No	Asynchronous, 512 x 16 kB = 8 MB Data acquisition / playback
xillybus_stream0_write_32 (edit | replicate | delete)					
	Downstream (host to FPGA)	32 bits	395 MB/s	No	Asynchronous, 512 x 16 kB = 8 MB DMA acceleration: 8 segments x 512 bytes Data acquisition / playback
xillybus_read_8 (edit | replicate | delete)					
	Upstream (FPGA to host)	32 bits	1 MB/s	Yes	General purpose
xillybus_write_8 (edit | replicate | delete)					
	Downstream (host to FPGA)	32 bits	1 MB/s	Yes	General purpose
xillybus_mem_8 (edit | replicate | delete)					
	Upstream (FPGA to host)	8 bits	102.400 kB/s	Yes	Address/data interface (5 address bits)
	Downstream (host to FPGA)	8 bits	102.400 kB/s	Yes	Address/data interface (5 address bits)

Add a new device file

Figure 79 Add new device file

After updating all files click *generate core* (Figure 80). Check core status and download it when available (Figure 81).



XILLYBUS. IP cores and design services

Announcing Xillybus for USB
Announcing Xillybus for High Performance Computing

IP Core Factory – List of device files for IP core "myipcore"

Hello, Anonymous User

Email address: Password: Login!

□ Remember me Forgot your password? Sign up!

My saved IP cores

Core summary

Name	Status	Target device family	Operating system
myipcore (edit attributes | replicate | delete | generate core)			
	Open for changes	Altera Cyclone V	Linux and Windows

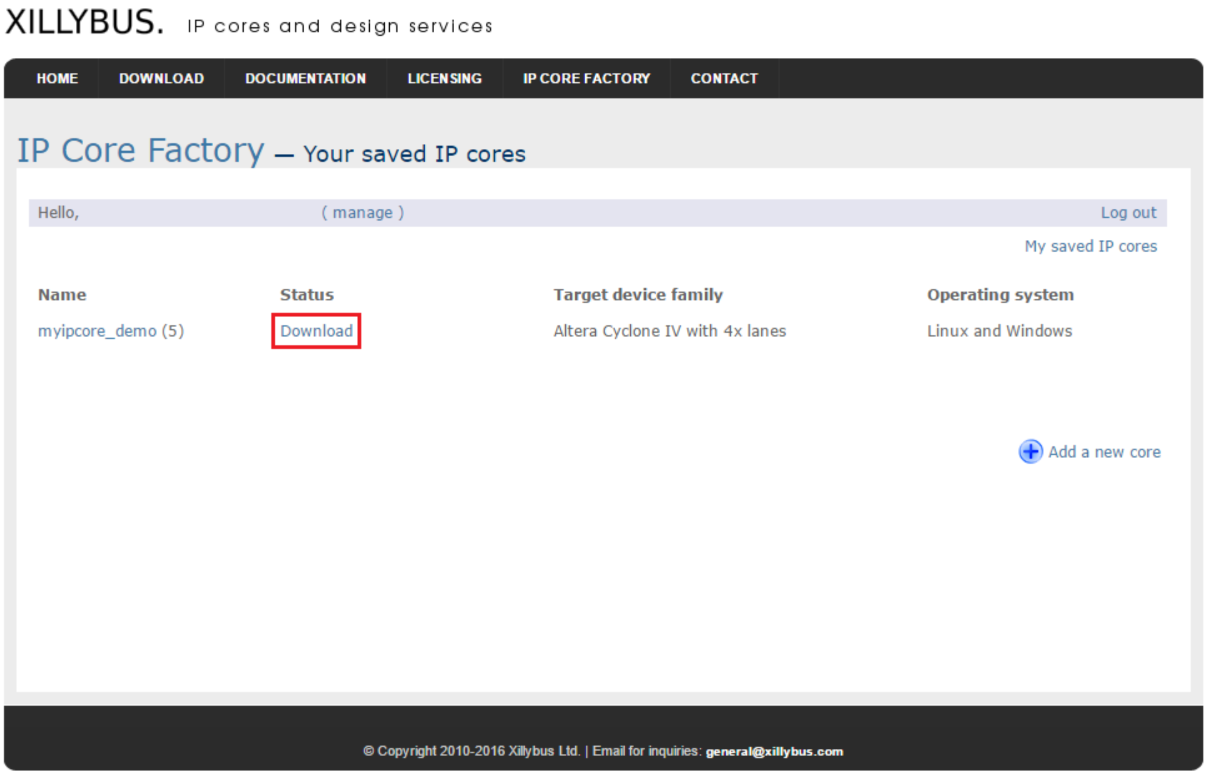
Add a new core

Device files

Name	Direction	Data width	Expected BW	Autoset	Details
xillybus_stream0_read_32 (edit | replicate | delete)					
	Upstream (FPGA to host)	32 bits	395 MB/s	No	Asynchronous, 512 x 16 kB = 8 MB Data acquisition / playback
xillybus_stream0_write_32 (edit | replicate | delete)					
	Downstream (host to FPGA)	32 bits	395 MB/s	No	Asynchronous, 512 x 16 kB = 8 MB DMA acceleration: 8 segments x 512 bytes Data acquisition / playback
xillybus_read_8 (edit | replicate | delete)					
	Upstream (FPGA to host)	32 bits	1 MB/s	Yes	General purpose
xillybus_write_8 (edit | replicate | delete)					
	Downstream (host to FPGA)	32 bits	1 MB/s	Yes	General purpose
xillybus_mem_8 (edit | replicate | delete)					
	Upstream (FPGA to host)	8 bits	102.400 kB/s	Yes	Address/data interface (5 address bits)
	Downstream (host to FPGA)	8 bits	102.400 kB/s	Yes	Address/data interface (5 address bits)
xillybus_stream1_read_32 (edit | replicate | delete)					
	Upstream (FPGA to host)	32 bits	395 MB/s	No	Asynchronous, 512 x 16 kB = 8 MB Data acquisition / playback
xillybus_stream1_write_32 (edit | replicate | delete)					
	Downstream (host to FPGA)	32 bits	395 MB/s	No	Asynchronous, 512 x 16 kB = 8 MB DMA acceleration: 8 segments x 512 bytes Data acquisition / playback
xillybus_stream2_read_32 (edit | replicate | delete)					
	Upstream (FPGA to host)	32 bits	395 MB/s	No	Asynchronous, 512 x 16 kB = 8 MB Data acquisition / playback
xillybus_stream2_write_32 (edit | replicate | delete)					
	Downstream (host to FPGA)	32 bits	395 MB/s	No	Asynchronous, 512 x 16 kB = 8 MB DMA acceleration: 8 segments x 512 bytes Data acquisition / playback

Add a new device file

Figure 80 Core generation



IP Core Factory — Your saved IP cores

Name	Status	Target device family	Operating system
myipcore_demo (5)	Download	Altera Cyclone IV with 4x lanes	Linux and Windows

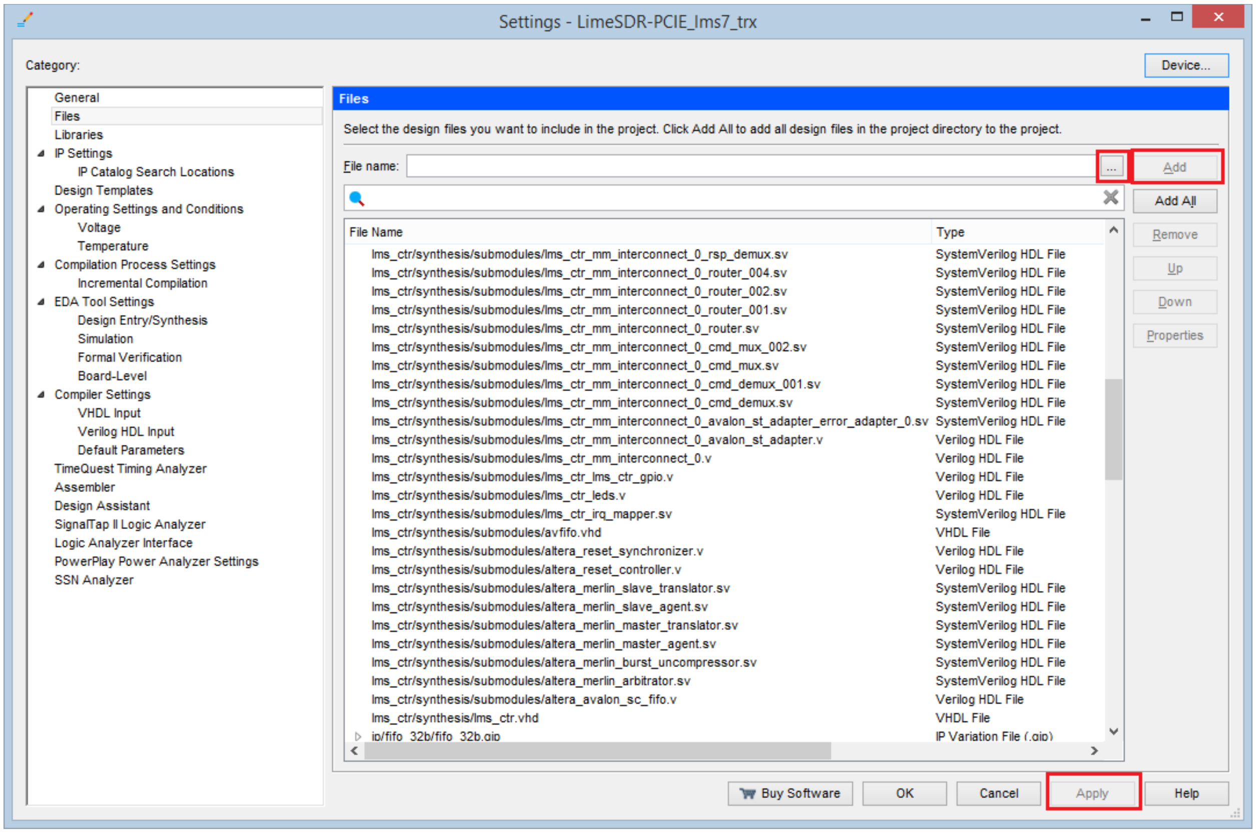
© Copyright 2010-2016 Xillybus Ltd. | Email for inquiries: **general@xillybus.com**

Figure 81 Download status

5.3.2 Adding PCIe core to project

This chapter describes steps to include Xillybus core to Quartus project:

- Extract downloaded .zip file "*corebundle-myipcore_demo.zip*" (*myipcore_demo* – name that was entered during core generation).
- Place file *xillybus.v* to Quartus project directory limesdr-qpcie_xillybus_core/
- Place file *xillybus_core.qxp* to Quartus project directory limesdr-qpcie_xillybus_core/
- Open Quartus *LimeSDR-QPCIE_lms7_trx* project and select *Project→ Add/Remove Files in Project.*. and add files *xillybus.v* and *xillybus_core.qxp* to Quartus project (Figure 82).
- Recompile project *Processing → Start Compilation*.



Settings - LimeSDR-PCIE_lms7_trx

Category:

- General
- Files
- Libraries
- IP Settings
  - IP Catalog Search Locations
- Design Templates
- Operating Settings and Conditions
  - Voltage
  - Temperature
- Compilation Process Settings
  - Incremental Compilation
- EDA Tool Settings
  - Design Entry/Synthesis
  - Simulation
  - Formal Verification
  - Board-Level
- Compiler Settings
  - VHDL Input
  - Verilog HDL Input
  - Default Parameters
- TimeQuest Timing Analyzer
- Assembler
- Design Assistant
- SignalTap II Logic Analyzer
- Logic Analyzer Interface
- PowerPlay Power Analyzer Settings
- SSN Analyzer

Files

Select the design files you want to include in the project. Click Add All to add all design files in the project directory to the project.

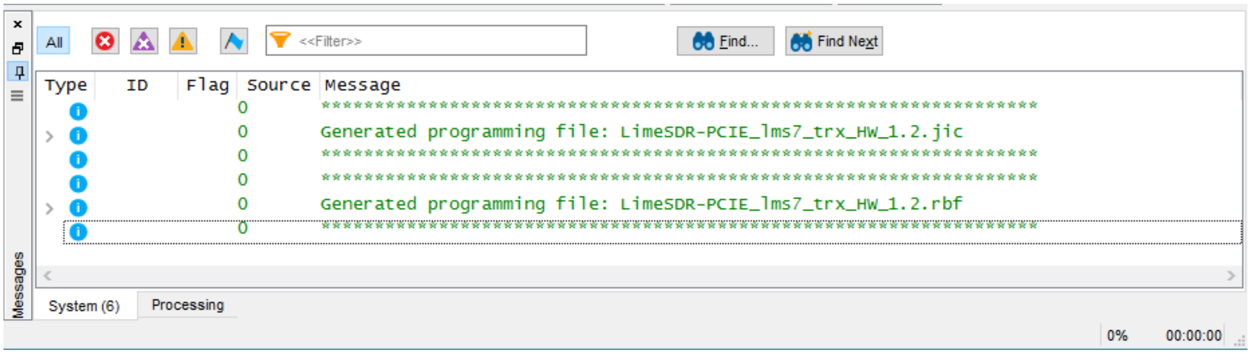
File name:

File Name	Type
lms_ctr/synthesis/submodules/lms_ctr_mm_interconnect_0_rsp_demux.sv	SystemVerilog HDL File
lms_ctr/synthesis/submodules/lms_ctr_mm_interconnect_0_router_004.sv	SystemVerilog HDL File
lms_ctr/synthesis/submodules/lms_ctr_mm_interconnect_0_router_002.sv	SystemVerilog HDL File
lms_ctr/synthesis/submodules/lms_ctr_mm_interconnect_0_router_001.sv	SystemVerilog HDL File
lms_ctr/synthesis/submodules/lms_ctr_mm_interconnect_0_router.sv	SystemVerilog HDL File
lms_ctr/synthesis/submodules/lms_ctr_mm_interconnect_0_cmd_mux_002.sv	SystemVerilog HDL File
lms_ctr/synthesis/submodules/lms_ctr_mm_interconnect_0_cmd_mux.sv	SystemVerilog HDL File
lms_ctr/synthesis/submodules/lms_ctr_mm_interconnect_0_cmd_demux_001.sv	SystemVerilog HDL File
lms_ctr/synthesis/submodules/lms_ctr_mm_interconnect_0_cmd_demux.sv	SystemVerilog HDL File
lms_ctr/synthesis/submodules/lms_ctr_mm_interconnect_0_avalon_st_adapter_error_adapter_0.sv	SystemVerilog HDL File
lms_ctr/synthesis/submodules/lms_ctr_mm_interconnect_0_avalon_st_adapter.v	Verilog HDL File
lms_ctr/synthesis/submodules/lms_ctr_mm_interconnect_0.v	Verilog HDL File
lms_ctr/synthesis/submodules/lms_ctr_lms_ctr_gpio.v	Verilog HDL File
lms_ctr/synthesis/submodules/lms_ctr_leds.v	Verilog HDL File
lms_ctr/synthesis/submodules/lms_ctr_irq_mapper.sv	SystemVerilog HDL File
lms_ctr/synthesis/submodules/avfifo.vhd	VHDL File
lms_ctr/synthesis/submodules/altera_reset_synchronizer.v	Verilog HDL File
lms_ctr/synthesis/submodules/altera_reset_controller.v	Verilog HDL File
lms_ctr/synthesis/submodules/altera_merlin_slave_translator.sv	SystemVerilog HDL File
lms_ctr/synthesis/submodules/altera_merlin_slave_agent.sv	SystemVerilog HDL File
lms_ctr/synthesis/submodules/altera_merlin_master_translator.sv	SystemVerilog HDL File
lms_ctr/synthesis/submodules/altera_merlin_master_agent.sv	SystemVerilog HDL File
lms_ctr/synthesis/submodules/altera_merlin_burst_uncompressor.sv	SystemVerilog HDL File
lms_ctr/synthesis/submodules/altera_merlin_arbitrator.sv	SystemVerilog HDL File
lms_ctr/synthesis/submodules/altera_avalon_sc_fifo.v	Verilog HDL File
lms_ctr/synthesis/lms_ctr.vhd	VHDL File
ip/fifo_32b/fifo_32b.qip	IP Variation File (.qip)

Figure 82 Adding files to Quartus project

5.3.3 Programming files

After performing full project compilation in Quartus prime software *Processing → Start Compilation* in Messages window (see Figure 83) should appear messages stating that programming files are created:



```
Type  ID  Flag  Source  Message
                0       ********************************************************************
                0       Generated programming file: LimeSDR-PCIE_lms7_trx_HW_1.2.jic
                0       ********************************************************************
                0       ********************************************************************
                0       Generated programming file: LimeSDR-PCIE_lms7_trx_HW_1.2.rbf
                0       ********************************************************************
```

Figure 83 Project compilation message window

Programming files can be found in folder *output_files* from project directory:

***.jic** - JTAG Indirect Configuration File can be used to program FPGA gateware to FLASH memory (if valid file is loaded FPGA boots from FLASH when board power is applied automatically).

***.sof** - SRAM Object File can be used to program FPGA (has to be programmed every time after board power is applied)
***.rbf** - Raw Binary File can be used to program FPGA gateware into FLASH memory through LimeSuiteGUI (valid gateware has to be already running)

5.4 Uploading FPGA Gateware to FLASH Memory

There are two ways of uploading FPGA gateware to onboard FLASH memory:

- Using LimeSuiteGUI (requires FX3 Firmware to be already uploaded)
- Using JTAG programming cable

5.4.1 Uploading FPGA Gateware to FLASH Memory using LimeSuiteGUI

This section describes how to load custom gateware to LimeSDR-QPCIe board FPGA Flash memory. This step requires that FX3 Firmware has to be already uploaded.

The Altera Cyclone V FPGA which sits on the LimeSDR-QPCIe board can be programmed using "*LimeSuiteGUI"*software. To call FPGA programing function, go to **Modules** from main menu and select **Programing** form the drop down menu, as shown in *Figure* 84.

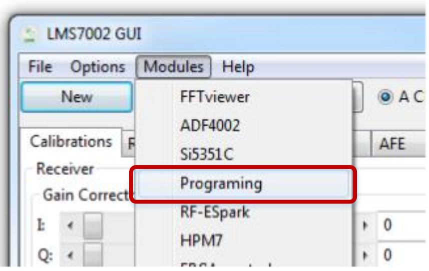
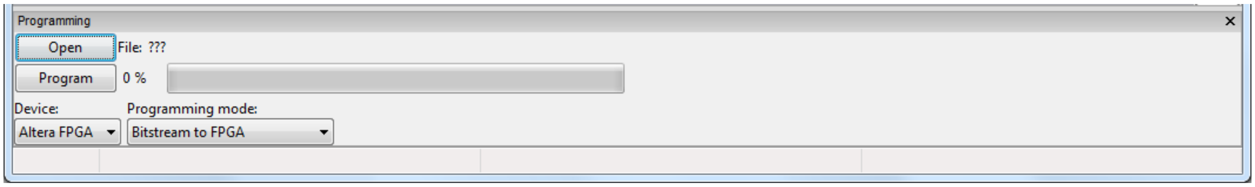


Figure 84 LimeSuiteGUI module menu to select FPGA programing tool

New window appears, as shown in the *Figure* 85.



File: ???
0 %
Device: Altera FPGA
Programming mode: Bitstream to FPGA

Figure 85 FPGA programing tool interface

Software loads raw binary files (*.rpd) [link] to FPGA and it offers couple options to do that, see *Figure* 86.

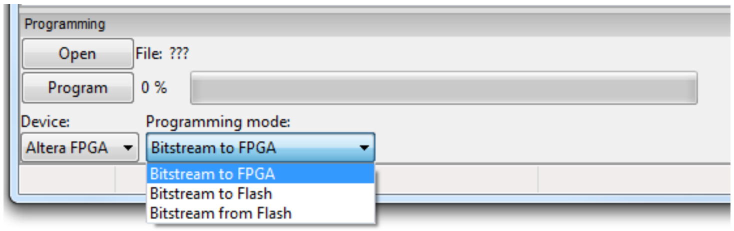
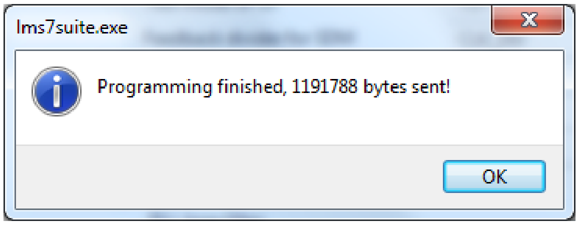


Figure 86 FPGA programing options

Select **Bitstream to FLASH programming** mode. This function loads selected *.rpd file from PC to external FPGA FLASH memory. Select required bitstream file by clicking **Open** and initiate FLASH memory programing by clicking on **Program**.

The new message will come up when the programing is finished, as shown in *Figure 87*.



Programming finished, 1191788 bytes sent!

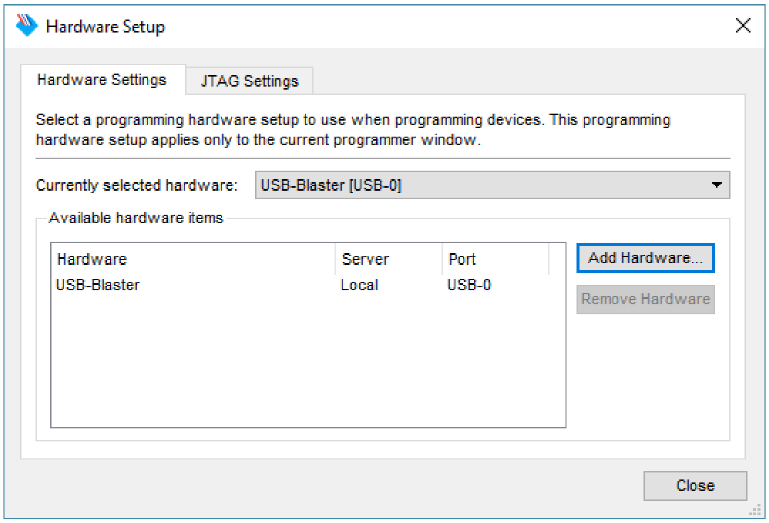
Figure 87 Successfully FPGA programing message (bytes shown may differ)

After writing new bitstream to Flash memory, it can be loaded to FPGA by changing **Programing mode** to **Bitstream from Flash** and pressing **Program**. New bitstream will be loaded to FPGA. Each time board is powered up, FPGA bitstream is loaded from FLASH automatically.

5.4.2 Uploading FPGA gateware to FLASH memory using JTAG Cable

For the first time use board can be programmed using JTAG header J26. This procedure requires two computers (LimeSDR-QPCIe board inserted into PCIe slot on computer #1 and Quartus Prime software running on computer #2).

- Insert LimeSDR-QPCIe board into computer #1. Make sure that computer is turned off while inserting board.
- Connect one end of download cable (e.g Altera USB Blaster) to LimeSDR-QPCIe board J26 connector and other end to USB port on the computer #2 running Quartus Prime software.
- Turn on computer #1 and interrupt the boot sequence to bring up the BIOS System Setup interface.
- Run Quartus Prime software in computer #2 and select Tools → Programmer
- Click Hardware Setup.. button and select your download cable, click Close (see Figure 88).



Select a programming hardware setup to use when programming devices. This programming hardware setup applies only to the current programmer window.

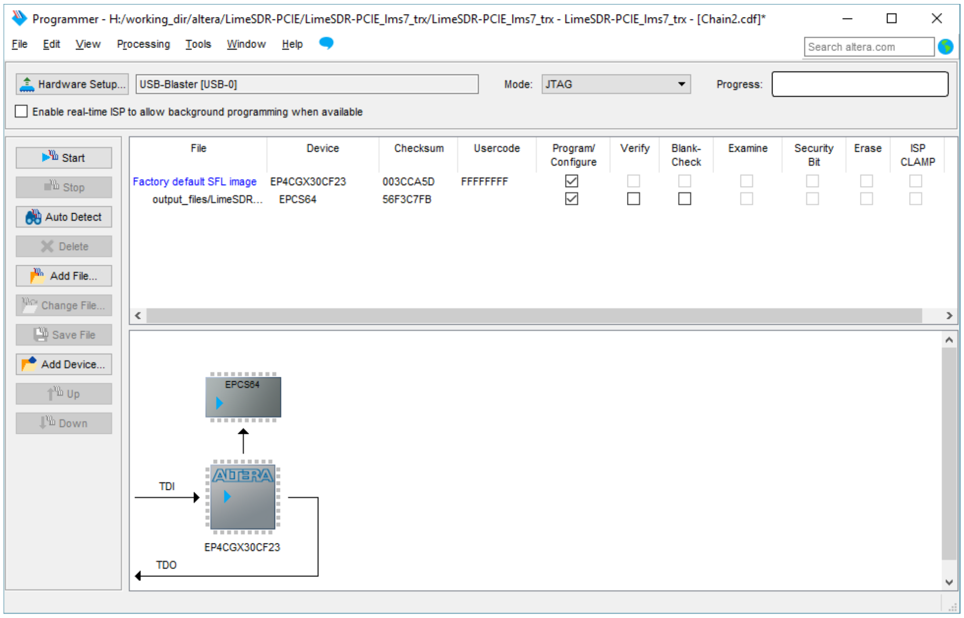
Currently selected hardware: USB-Blaster [USB-0]

Available hardware items

Hardware	Server	Port
USB-Blaster	Local	USB-0

Figure 88 Selecting programming hardware

- Click Add File.. and select *.jic file (see options below):
    a. Pre compiled bitstream can be found in gateware/LimeSDR-QPCIE_lms7_trx_bs
    b. If you have followed project compilation instructions and generated your own bitstream then your file is located in project directory /output_files.
- Apply settings as in Figure 89 and click Start.



Hardware Setup... USB-Blaster [USB-0]   Mode: JTAG   Progress:

□ Enable real-time ISP to allow background programming when available

File	Device	Checksum	Usercode	Program/ Configure	Verify	Blank-Check	Examine	Security Bit	Erase	ISP CLAMP
Factory default SFL image	EP4CGX30CF23	003CCA5D	FFFFFFFF	☑	□	□	□	□	□	□
output_files/LimeSDR...	EPCS64	56F3C7FB		☑	□	□	□	□	□	□

Figure 89 Adding programming file

- After successful programming turn off computer #1.
- FPGA boots from programmed FLASH memory automatically when computer #1 is turned on.