

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-USB v1.4 Quick Start Manual

- Hardware and software description -

REVISION HISTORY

The following table shows the revision history of this document:

Date	Version	Description of Revisions
22/08/2017	1.0	Initial version

Table of Contents

REVISION HISTORY	2
1. INTRODUCTION.....	5
2. LIMESDR-USB BOARD KEY FEATURES	6
2.1 LimeSDR-USB board overview	8
2.2 LimeSDR-USB board architecture.....	12
2.2.1 LMS7002M based connectivity.....	12
2.2.2 SDRAM	13
2.2.3 USB 3.0 controller.....	13
2.2.4 Indication LEDs	14
2.2.5 Low speed interfaces.....	15
2.2.6 Board temperature control	21
2.2.7 Clock Distribution	22
2.2.8 Power Distribution.....	23
3. GETTING STARTED WITH LIMESDR-USB	26
3.1 Launching LimeSuiteGUI and connecting to the board.....	26
3.2 Loading and saving settings	27
3.3 Quick Test.....	28
3.4 Changing TX / RX Frequency	29
3.5 Changing Receiver Gain	30
3.6 Changing Transmitter output signal gain	31
3.7 Load waveform for Tx Path	32
3.8 Digital Loopback Enable	33
3.9 Run FTT viewer.....	34
3.10 Calibrate Rx path	36
3.11 Calibrate TX Path.....	39
3.12 Clock configuration	43
3.12.1 VCTCXO tuning.....	43
3.12.2 Tuning VCTCXO using frequency synthesizer (ADF4002).....	43
3.12.3 Tuning VCTCXO using DAC	44
3.12.4 VCTCXO calibration	45
3.12.5 Programmable clock generator (Si5351C) configuration.....	46

3.13	Reading board temperature.....	47
4.	DRIVERS INSTALLATION.....	49
4.1	Windows USB driver installation procedure	49
4.2	Linux USB drivers.....	52
5.	BOARD PROGRAMMING	53
5.1	Connecting to the board	53
5.2	Updating FX3 firmware (normal mode)	53
5.3	Uploading FPGA bitstream to Flash	54
5.4	Flashing USB3 microcontroller firmware in bootloader mode	56
5.5	Uploading firmware to SPI FLASH	57
5.6	Uploading firmware to the FX3 RAM	58

1. Introduction

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

2. LimeSDR-USB Board Key Features

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

The LimeSDR-USB has two different USB connector versions. LimeSDR 1v0 is micro USB type B connector (socket) based, showed in *Figure 1*. LimeSDR 1v0s is a USB type A connector (plug) based, showed in *Figure 2*.

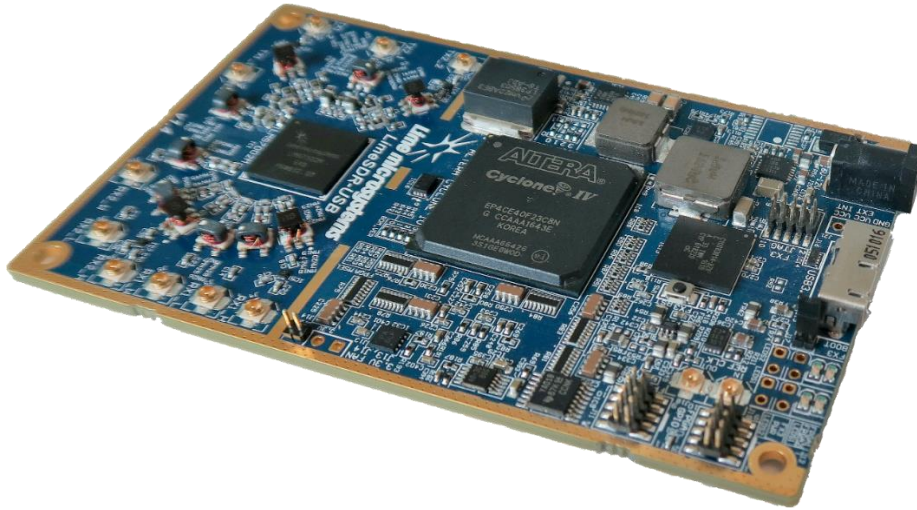


Figure 1 LimeSDR 1v0 – micro USB type B connector board



Figure 2 LimeSDR – USB type A connector board

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

- Cyclone IV device family, refer to Cyclone IV Device Handbook [\[link\]](#)
- LMS7002M transceiver resources [\[link\]](#)

LimeSDR-USB board features:

- **USB Interface**
 - Cypress FX3 Super Speed USB 3rd generation controller
- **FPGA Features**
 - Cyclone IV EP4CE40F23C8N device in 484-pin FPGA
 - 39'600 logic elements
 - 1134 Kbits embedded memory
 - 116 embedded 18x18 multipliers
 - 4 PLLs
- **FPGA Configuration**
 - JTAG mode configuration
 - Active serial mode configuration
 - Possibility to update FPGA gateware by using FX3 (USB)
- **Memory Devices**
 - 2x 1Gbit (64M x 16) DDR2 SDRAM
 - 4Mbit flash for FX3 firmware
 - 16Mbit flash for FPGA gateware
 - 2 x 128K (16K x 8) EEPROMs for LMS MCU firmware, LMS MCU data
 - 1 x 64K (8K x 8) EEPROM for FX3 data
- **Connections**
 - microUSB3.0 (type B) connector or USB3.0 (type A) plug
 - Coaxial RF (U.FL) connectors
 - FPGA GPIO headers (0.05" pitch)
 - FPGA and FX3 JTAG connectors (0.05" pitch)
 - 6..12V DC power jack and pinheader
 - Fan connector (3.3V)
- **Clock System**
 - 30.72MHz VCTCXO (precision: ± 1 ppm initial, ± 4 ppm stable).
 - Possibility to lock VCTCXO to external clock or tune VCTCXO by onboard DAC
 - Programmable clock generator for the FPGA reference clock input or LMS PLLs
- **Board Size** 60mm x 100mm (2.36" x 3.94")

2.1 LimeSDR-USB board overview

LimeSDR-USB board version 1.4 picture with highlighted major connections presented in *Figure 3*. There are three connector types – data and debugging (USB3.0, FPGA GPIO and JTAG), power (DC jack and external supply pinheader) and high frequency (RF and reference clock).

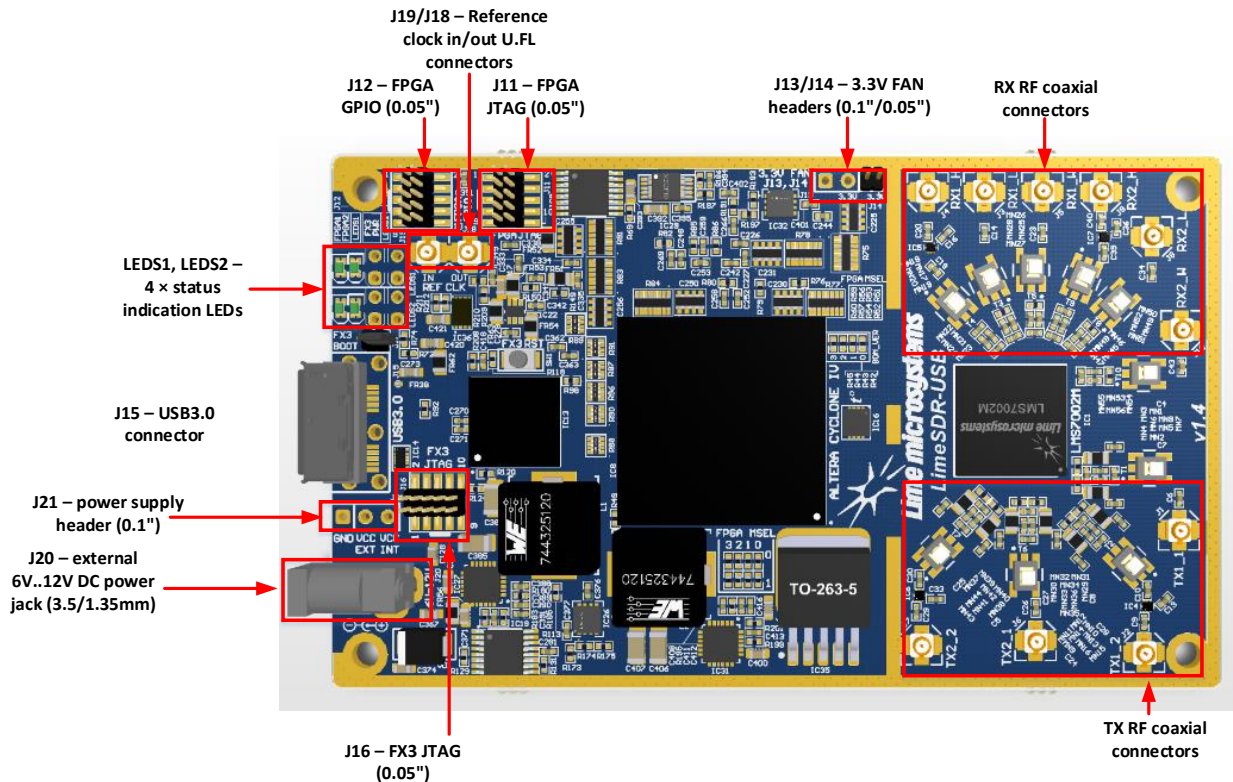


Figure 3 LimeSDR-USB Development Board Connectors

Board components description listed in the Table 1 and Table 2.

Table 1. Board components

Featured Devices		
Board Reference	Type	Description
IC1	FPRF	Field programmable RF transceiver LMS7002M
IC8	FPGA	Altera Cyclone IV, EP4CE40F23C8N, 484-BGA
IC13	USB3.0 microcontroller	Cypress FX3 Supper Speed USB 3 rd generation controller CYUSB3014
Miscellaneous devices onboard		
IC9	IC	Quad analogue switch TS3A5018PW
IC6	IC	Temperature sensor LM75
IC19	IC	SPI to I2C bridge SC18IS602BIPW (not mounted)
IC17	IC	I2C Port Expander with 4 Push-Pull Outputs and 4 Inputs MAX7322ATE+

Configuration, Status and Setup Elements		
Board Reference	Type	Description
R51, R52, R53, R54, R56, R57, R59, R60	0 Ohm resistor	FPGA (IC31) MSEL[3:0]. Default mode: Active Serial Standard configuration
R115, R116, R117	10 kOhm resistor	USB3.0 microcontroller (IC13) boot configuration (PMODE0[2:0]) resistors. Default mode: SPI boot, On Failure - USB Boot
R125, R127, R128	10 kOhm resistor	USB3.0 microcontroller (IC13) crystal/clock frequency selection (FSLC[2:0]) resistors. Default mode: 19.2MHz crystal
J16	JTAG chain pin header	USB3.0 (IC6) microcontroller's debugging pin header, 0.05" pitch
J17, R122	Pin header	USB3.0 microcontroller boot source (Flash memory or USB), 0.05" pitch jumper or 0402 0R resistor. In normal operation jumper or resistor must be placed.
SW1	Push-button	USB3.0 microcontroller reset button
J11	JTAG chain pin header	FPGA programming pin header for Altera USB-Blaster download cable, 0.05" pitch
LEDS1	Red-green status LEDs	User defined FPGA indication LED1 (near board edge if SMD; on the bottom if through-hole), User defined FPGA indication LED2 (farther board edge if SMD; on the top if through-hole)
LEDS2	Red-green status LEDs	FX3 (USB) status indication LED (near board edge if SMD; on the bottom if through-hole), board power indication LED (farther board edge if SMD; on the top if through-hole)
General User Input/Output		
Board Reference	Type	Description
J12	Pin header	8 FPGA GPIOs, 0.05" pitch
J13, J14	Pin header	3.3V fan connection pin headers, 0.1" and 0.05" pitch respectively

Table 2 Board components

Memory Devices		
Board Reference	Type	Description
IC11, IC12	DDR2 memory	1Gbit (64M x 16) DDR2 SDRAM with a 16-bit data bus
IC2, IC3	EEPROM	128K (16K x 8) EEPROM, LMS7002 MCU firmware, LMS7002M data
IC18	EEPROM	64K (8K x 8) EEPROM, connected to main I2C bus
IC10	Flash memory	16Mbit Flash for FPGA configuration
IC15	Flash memory	4Mbit Flash for FX3 firmware
Communication Ports		

Board Reference	Type	Description
J15	USB3.0 connector	microUSB3.0 (type B) connector or USB3.0 (type A) plug
Clock Circuitry		
Board Reference	Type	Description
XO1, XO2	VCTCXO	30.72MHz voltage-controlled crystal oscillator
IC24	IC	Programmable clock generator for the FPGA reference clock input and RF boards
IC23	IC	ADF4002 phase detector
IC22	IC	DAC for TCXO (XT4) frequency tuning
J19	U.FL connector	Reference clock input
J18	U.FL connector	Reference clock output
Power Supply		
Board Reference	Type	Description
J20	DC input jack	External 6V..12V DC power supply
J21	Pin header	External 6V..12V DC power supply and main internal power rail

LimeSDR-USB board version 1.4 picture with highlighted components on top presented in Figure 4.

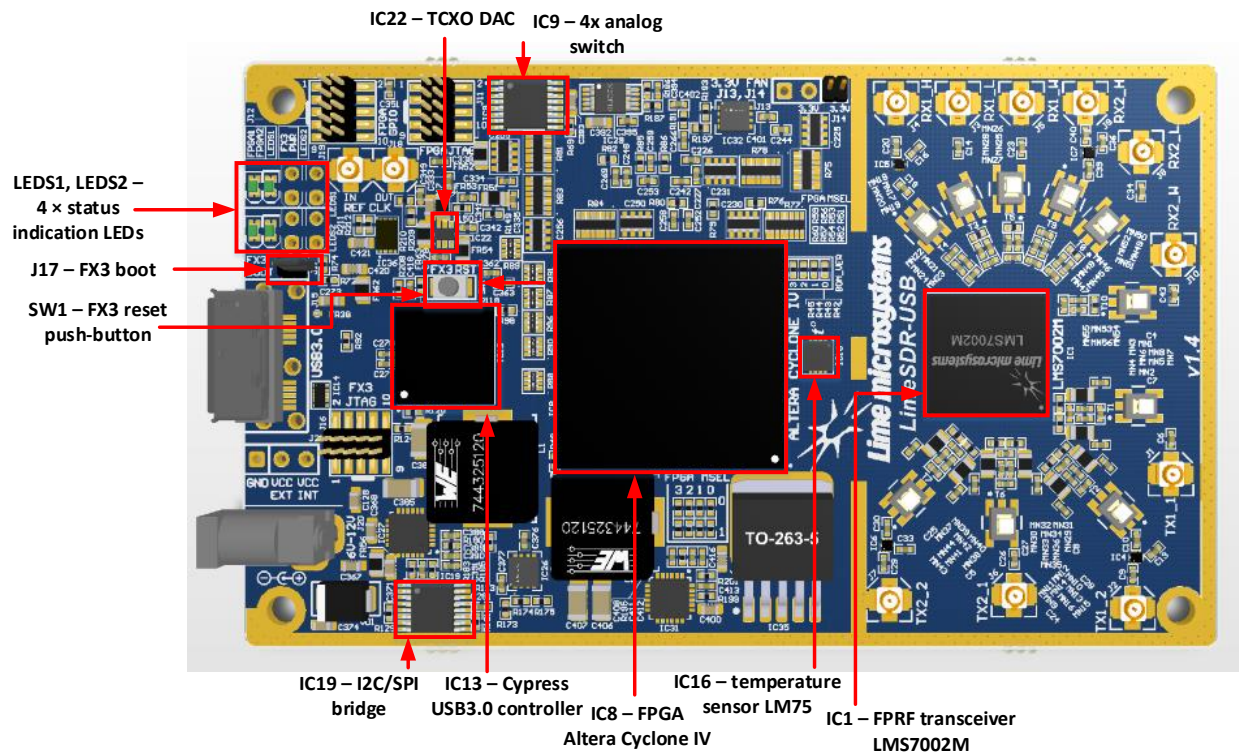


Figure 4 LimeSDR Development Board Top Components

LimeSDR-USB board version 1.4 picture with highlighted components on bottom presented in *Figure 5*

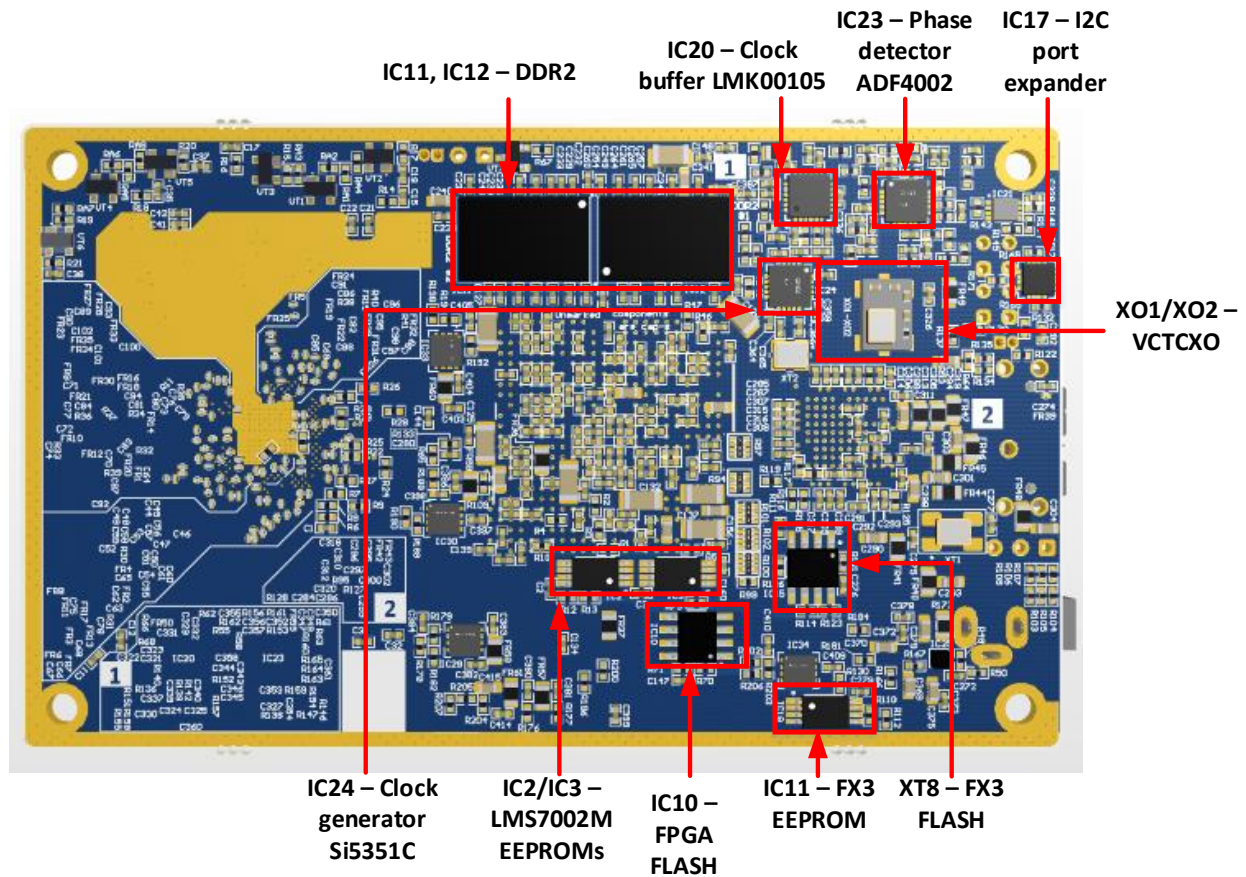


Figure 5 LimeSDR-USB Development Board Bottom Components

2.2 LimeSDR-USB board architecture

The heart of the LimeSDR-USB board is Altera Cyclone IV FPGA. It's main function is to transfer digital data between the PC through a USB3.0 connector. The block diagram for LimeSDR-USB board is presented in the *Figure 6*.

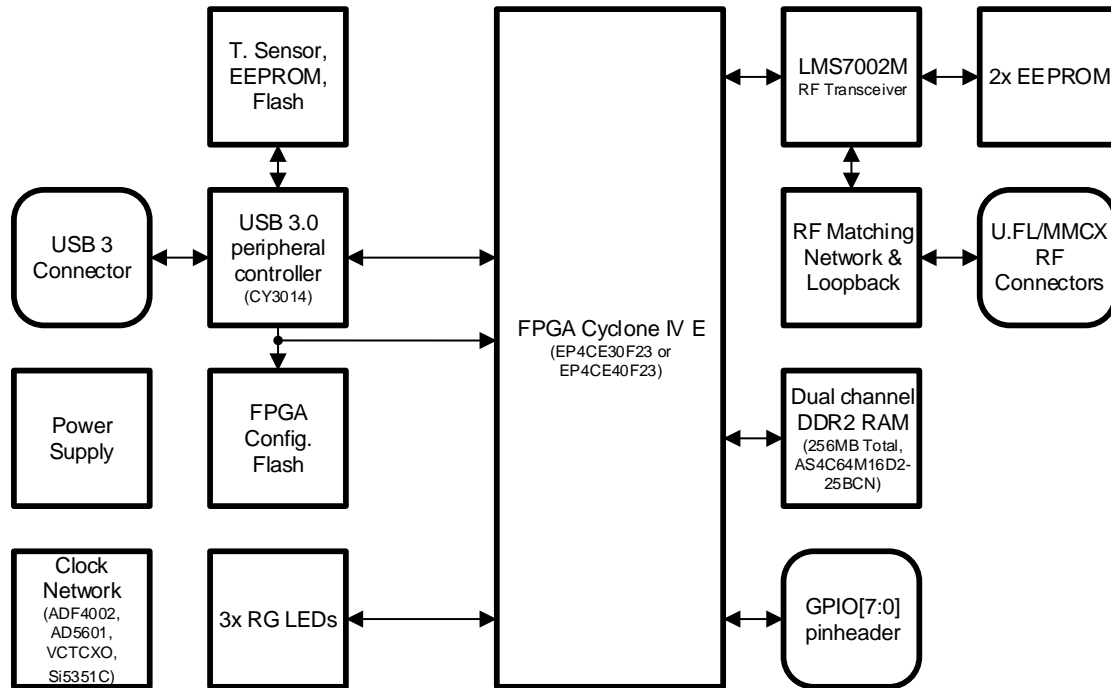


Figure 6 LimeSDR-USB Development Board Block Diagram

2.2.1 LMS7002M based connectivity

The interface and control signals are described below:

- Digital Interface Signals:** LMS7002 is using data bus LMS_DIQ1_D[11:0] and LMS_DIQ2_D[11:0], LMS_ENABLE_IQSEL1 and LMS_ENABLE_IQSEL2, LMS_FCLK1 and LMS_FCLK2, LMS_MCLK1 and LMS_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 transmit port and PORT-2 is selected as receiver 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:
 - LMS_RXEN, LMS_TXEN – receiver and transmitter enable/disable signals connected to FPGA Bank 8 (VDIO_LMS_FPGA; 2.5V).
 - LMS_RESET – LMS7002M reset connected to FPGA Bank 7 (VDIO_LMS_FPGA; 2.5V).

- **SPI Interface:** LMS7002M transceiver is configured via 4-wire SPI interface; FPGA_SPIO_SCLK, FPGA_SPIO_MOSI, FPGA_SPIO_MISO, FPGA_SPIO_LMS_SS. The SPI interface controlled from FPGA Bank 8 (VDIO_LMS_FPGA; 2.5V).
- **Main I2C Interface:** used to control external clock synthesizer, port expander, temperature sensor, EEPROM, I2C-SPI bridge on LimeSDR-USB board. The signals FX3_I2C_SCL, FX3_I2C_SDA connected to FX3. Also these I2C lines are connected via 0R resistors to FPGA Bank 8 (VCC3P3; 3.3V) lines FPGA_I2C_SCL, FPGA_I2C_SDA.
- **LMS I2C Interface:** can be used for LMS EEPROM content modifying or for debug purposes. The signals LMS_I2C_SCL, LMS_I2C_DATA connected to FPGA Bank 8 (VDIO_LMS_FPGA; 2.5V).

2.2.2 SDRAM

LimeSDR-USB board has two 128MB (16bit bus) DDR2 SDRAM ICs (AS4C64M16D2-25BCN [[link](#)]) connected to double data rate pins on Cyclone IV 1.8V Bank 2, Bank 3 and Bank 4. RAM chips are connected to separate memory controllers so RAM chips works in dual channel mode. The memory can be used for data manipulation at high data rates between transceiver and FPGA.

2.2.3 USB 3.0 controller

Software controls LimeSDR-USB board via the USB3 microcontroller (CYUSB3014 (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[31:0] – FX3 32-bit GPIF data interface is connected to Cyclone IV 1.8V Bank 5.
- FX3_CTL[8:0], [12:11] – FX3 GPIF interface control signals.
- FX3_PCLK – GPIF interface clock. Clock from FX3 is fed 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 J17 jumper is present or R122 is soldered FX3 will boot from IC8 flash memory if correct firmware exists.
- SW1 – resets FX3
- J16 – FX3 JTAG programming/debugging pin header.

2.2.4 Indication LEDs

LimeSDR-USB board comes with four dual colour (red and green (RG)) indication LEDs. These LEDs can be SMD or 3mm TH with dedicated right angle plastic holders. By default, dual colour SMD LEDs are populated and through hole LEDs are unpopulated. If required, the user can fit dual colour TH LEDs.

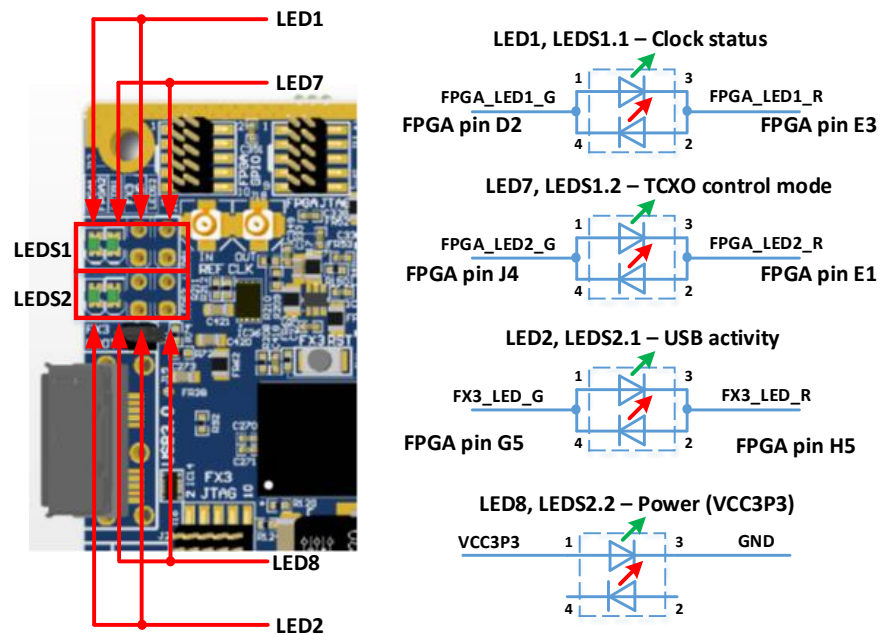


Figure 7 LimeSDR-USB indication LEDs

Each LED has its own function. Most of LEDs are connected to FPGA and their function can be changed. Default LEDs functions and other information are listed in the table below.

Table 3 Default LEDs functions

Board Reference	Schematic name	Board label	Type	Description
LED8, LEDS2.2	GND, VCC3P3	PWR	Power status	LED is hardwired to VCC3P3 power rail. Steady green light indicates presence of 3.3v.
LED2, LEDS2.1	FX3_LED_R, FX3_LED_G	FX3	USB activity	USB3.0 (FX3) controller, slave FIFO (GPIF) interface module and NIOS CPU activity indication: Green – idle, Red – busy.
LED1, LEDS1.1	FPGA_LED1_G, FPGA_LED1_R	FPGA1	Clock status	Blinking indicates presence of TCXO clock.

				Colour indicates status of FPGA PLLs that are used for LMS digital interface clocking: Green – both PLLs are locked; Red/Green – at least one PLL is not locked.
LED7, LEDS1.2	FPGA_LED2_G, FPGA_LED2_R	FPGA2	TCXO control mode	No light – TCXO is controlled from DAC Red – TCXO is controlled from phase detector and is not locked to external reference clock Green – TCXO is controlled from phase detector and is locked to external reference clock

2.2.5 Low speed interfaces

LimeSDR-USB board low speed interfaces are divided into FX3 and FPGA groups that are presented in Figure 3, Figure 8 and Figure 9. The latter block diagrams depict the main ICs, corresponding IC pin numbers, data buses and serial protocol addresses.

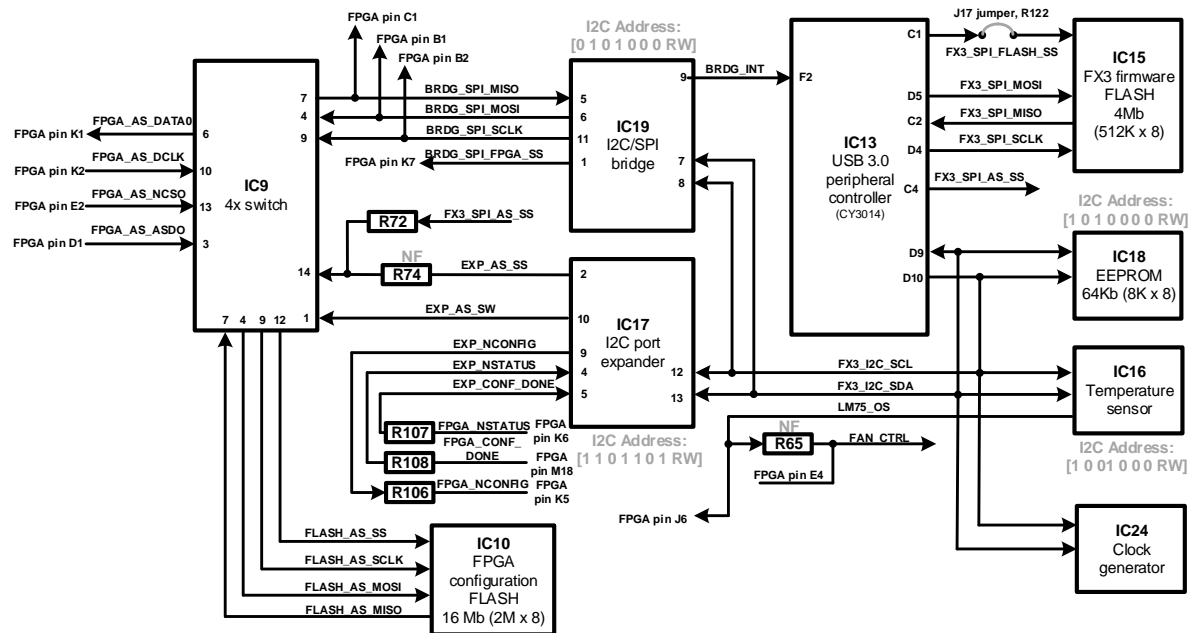


Figure 8 FX3 low speed interfaces block diagram

LimeSDR-USB board peripherals are controlled via USB interface. All commands that comes from USB are firstly processed by FX3 controller. If I2C peripheral (temperature sensor, port expander, clock generator) on FX3_I2C bus must be controlled this can be done directly. FX3_I2C is also connected to FPGA.

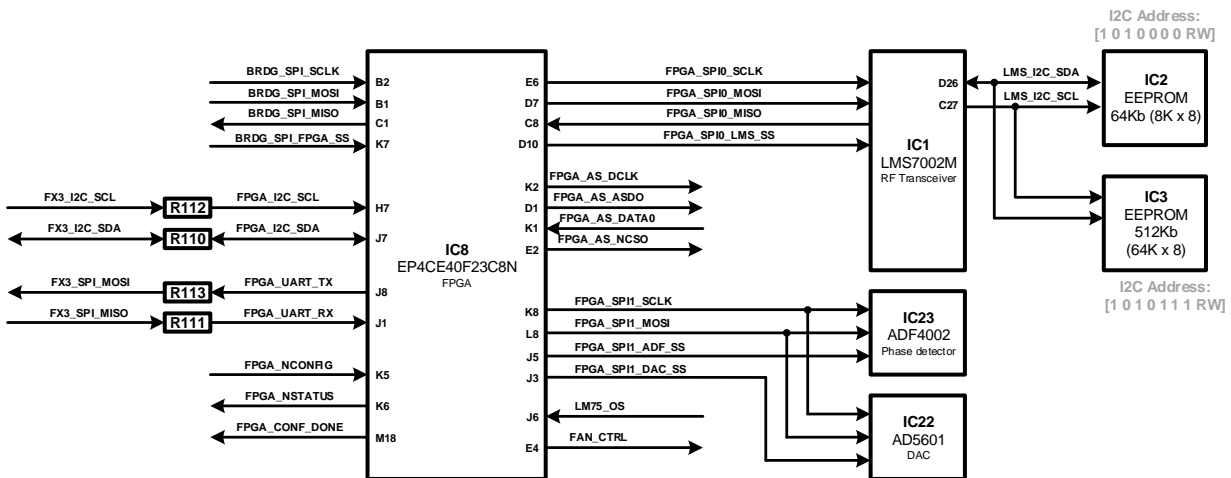


Figure 9 FPGA low speed interfaces block diagram

There are several SPI interfaces with their slave devices:

- FX3_SPI:** FX3 hardware SPI cannot be used if 32-bit GPIF interface is configured. If any slave device on this bus must be accessed then FX3 switches to 16-bit GPIF. This bus has these slave devices:
 - FX3 firmware Flash
 - FPGA configuration flash: using switch (IC9) flash memory is switched from FPGA to FX3_SPI BUS. Then flash content is updated and flash memory is switch back to FPGA. This is done when needs to update FPGA gateway in flash memory.
- FPGA_SPI0, FPGA_SPI1:** these SPI interfaces are connected to FPGA and slave devices can be accessed by transferring data to internal FPGA NIOS CPU. This bus has these slave devices:
 - RFIC** (FPGA_SPI0)
 - Phase detector** (FPGA_SPI1)
 - DAC** (FPGA_SPI1)
- Internal FPGA SPI module:** FPGA has its own SPI module and can be controlled as regular SPI device. By using FPGA SPI it is possible to control FPGA modes etc.
- BRDG_SPI:** this is alternative SPI interface for FX3 and currently is not used.

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

Table 4. USB3 controller (FX3) pins

Chip pin (IC6)	Chip reference (IC6)	Schematic signal name	FPGA pin	I/O standard	Comment
F10	GPIO0	FX3_DQ0	M19	1.8V	
F9	GPIO1	FX3_DQ1	AA21	1.8V	
F7	GPIO2	FX3_DQ2	Y22	1.8V	
G10	GPIO3	FX3_DQ3	Y21	1.8V	
G9	GPIO4	FX3_DQ4	W22	1.8V	
F8	GPIO5	FX3_DQ5	W21	1.8V	
H10	GPIO6	FX3_DQ6	W20	1.8V	
H9	GPIO7	FX3_DQ7	V22	1.8V	
J10	GPIO8	FX3_DQ8	V21	1.8V	
J9	GPIO9	FX3_DQ9	U22	1.8V	
K11	GPIO10	FX3_DQ10	U21	1.8V	
L10	GPIO11	FX3_DQ11	U20	1.8V	
K10	GPIO12	FX3_DQ12	U19	1.8V	
K9	GPIO13	FX3_DQ13	M22	1.8V	
J8	GPIO14	FX3_DQ14	M21	1.8V	
G8	GPIO15	FX3_DQ15	R22	1.8V	
J6	GPIO16	FX3_PCLK	T21	1.8V	
K8	GPIO17	FX3_CTL0	L6	1.8V	
K7	GPIO18	FX3_CTL1	L7	1.8V	
J7	GPIO19	FX3_CTL2	M1	1.8V	
H7	GPIO20	FX3_CTL3	M2	1.8V	
G7	GPIO21	FX3_CTL4	M3	1.8V	
G6	GPIO22	FX3_CTL5	M4	1.8V	
K6	GPIO23	FX3_CTL6	M6	1.8V	
H8	GPIO24	FX3_CTL7	M7	1.8V	
G5	GPIO25	FX3_CTL8	M8	1.8V	
H6	GPIO26	FX3_DCTL9	P9	1.8V	
K5	GPIO27	FX3_DCTL10	N7	1.8V	
J5	GPIO28	FX3_CTL11	N5	1.8V	
H5	GPIO29	FX3_CTL12	N6	1.8V	
G4	GPIO30	FX3_PMODE0	-	1.8V	
H4	GPIO31	FX3_PMODE1	-	1.8V	
L4	GPIO32	FX3_PMODE2	-	1.8V	
K2	GPIO33	FX3_DQ16	R21	1.8V	
J4	GPIO34	FX3_DQ17	R20	1.8V	
K1	GPIO35	FX3_DQ18	R19	1.8V	
J2	GPIO36	FX3_DQ19	R18	1.8V	
J3	GPIO37	FX3_DQ20	P22	1.8V	
J1	GPIO38	FX3_DQ21	P21	1.8V	

H2	GPIO39	FX3_DQ22	M20	1.8V	
H3	GPIO40	FX3_DQ23	P16	1.8V	
F4	GPIO41	FX3_DQ24	P15	1.8V	
G2	GPIO42	FX3_DQ25	N22	1.8V	
G3	GPIO43	FX3_DQ26	N21	1.8V	
F3	GPIO44	FX3_DQ27	N20	1.8V	
F2	GPIO45	BRDG_INT_LS		1.8V	Connected to I2C-SPI bridge
F5	GPIO46	FX3_DQ28	N19	1.8V	
E1	GPIO47	FX3_DQ29	N18	1.8V	
E5	GPIO48	FX3_DQ30	N17	1.8V	
E4	GPIO49	FX3_DQ31	N16	1.8V	
D1	GPIO50	-	-	1.8V	
D2	GPIO51	-	-	1.8V	
D3	GPIO52	-	-	1.8V	
D4	GPIO53	FX3_SPI_SCLK	-	3.3V	
C1	GPIO54	FX3_SPI_FLASH_SS	-	3.3V	
C2	GPIO55	FX3_SPI_MISO	-	3.3V	
D5	GPIO56	FX3_SPI_MOSI	-	3.3V	
C4	GPIO57	-	-	3.3V	
D9	I2C_GPIO58	FX3_I2C_SCL/ FPGA_I2C_SCL	H7	3.3V	Connected to I2C-SPI bridge
D10	I2C_GPIO59	FX3_I2C_SDA/ FPGA_I2C_SDA	J7	3.3V	Connected to I2C-SPI bridge

In the table below are listed RF transceiver (LMS7002) pins, schematic signal name, FPGA interconnections and I/O standard.

Table 5. RF transceiver (LMS7002) digital interface pins

Chip pin (IC1)	Chip reference (IC1)	Schematic signal name	FPGA pin	FPGA I/O standard	FPGA I/O standard
AM24	xoscin_rx	RxPLL_CLK	NC	3.3V	Connected to 30.72 MHz clock
P34	MCLK2	LMS_MCLK2	B11	2.5V	
R29	FCLK2	LMS_FCLK2	E5	2.5V	
U31	TXNRX2	LMS_TXNRX2	B8	2.5V	
V34	RXEN	LMS_RXEN	C3	2.5V	
R33	ENABLE_IQSE L2	LMS_ENABLE_IQSEL2	C7	2.5V	
H30	DIQ2_D0	LMS_DIQ2_D0	B7	2.5V	

J31	DIQ2_D1	LMS_DIQ2_D1	B6	2.5V	
K30	DIQ2_D2	LMS_DIQ2_D2	B4	2.5V	
K32	DIQ2_D3	LMS_DIQ2_D3	B3	2.5V	
L31	DIQ2_D4	LMS_DIQ2_D4	A10	2.5V	
K34	DIQ2_D5	LMS_DIQ2_D5	A9	2.5V	
M30	DIQ2_D6	LMS_DIQ2_D6	A8	2.5V	
M32	DIQ2_D7	LMS_DIQ2_D7	A7	2.5V	
N31	DIQ2_D8	LMS_DIQ2_D8	A6	2.5V	
N33	DIQ2_D9	LMS_DIQ2_D9	A5	2.5V	
P30	DIQ2_D10	LMS_DIQ2_D10	A4	2.5V	
P32	DIQ2_D11	LMS_DIQ2_D11	A11	2.5V	
E5	xoscin_tx	TxPLL_CLK	NC	3.3V	Connected to 30.72 MHz clock
AB34	MCLK1	LMS_MCLK1	G21	2.5V	
AA33	FCLK1	LMS_FCLK1	B20	2.5V	
V32	TXNRX1	LMS_TXNRX1	B9	2.5V	
U29	TXEN	LMS_TXEN	B10	2.5V	
1Y32	ENABLE_IQSE L1	LMS_ENABLE_IQS EL1	C4	2.5V	
AG31	DIQ1_D0	LMS_DIQ1_D0	B17	2.5V	
AF30	DIQ1_D1	LMS_DIQ1_D1	B17	2.5V	
AF34	DIQ1_D2	LMS_DIQ1_D2	B15	2.5V	
AE31	DIQ1_D3	LMS_DIQ1_D3	B14	2.5V	
AD30	DIQ1_D4	LMS_DIQ1_D4	B13	2.5V	
AC29	DIQ1_D5	LMS_DIQ1_D5	C13	2.5V	
AE33	DIQ1_D6	LMS_DIQ1_D6	A18	2.5V	
AD32	DIQ1_D7	LMS_DIQ1_D7	A17	2.5V	
AC31	DIQ1_D8	LMS_DIQ1_D8	A16	2.5V	
AC33	DIQ1_D9	LMS_DIQ1_D9	A15	2.5V	
AB30	DIQ1_D10	LMS_DIQ1_D10	A14	2.5V	
AB32	DIQ1_D11	LMS_DIQ1_D11	A13	2.5V	
U33	CORE_LDO_EN	LMS_CORE_LDO_EN	B18	2.5V	
E27	RESET	LMS_RESET	C6	2.5V	
D28	SEN	FPGA_SPI0_LMS_SS	D10	2.5V	SPI interface
C29	SCLK	FPGA_SPI0_SCLK	E6	2.5V	SPI interface
F30	SDIO	FPGA_SPI0_MOSI	D7	2.5V	SPI interface
F28	SDO	FPGA_SPI0_MISO	C8	2.5V	SPI interface
D26	SDA	LMS_I2C_SDA	C21	2.5V	Connected to EEPROM

C27	SCL	LMS_I2C_SCL	C17	2.5V	Connected to EEPROM
-----	-----	-------------	-----	------	---------------------

In the tables below are listed RF transceiver (LMS7002) pins, schematic signal name, FPGA interconnections and I/O standard.

Table 6. FPGA_SPI0 interface pins

Schematic signal name	FPGA pin	I/O standard	Comment
FPGA_SPI0_SCLK	E6	2.5V	Serial Clock (FPGA output)
FPGA_SPI0_MOSI	D7	2.5V	Data (FPGA output)
FPGA_SPI0_MISO	C8	2.5V	Data (FPGA input)
FPGA_SPI0_LMS_SS	D10	2.5V	IC1 (LMS7002) SPI slave select

Table 7. FPGA_SPI1 interface pins

Schematic signal name	FPGA pin	I/O standard	Comment
FPGA_SPI1_SCLK	K8	3.3V	Serial Clock (FPGA output)
FPGA_SPI1_MOSI	L8	3.3V	Data (FPGA output)
FPGA_SPI1_DAC_SS	J3	3.3V	IC15 SPI slave select (FPGA output)
FPGA_SPI1_ADF_SS	K7	3.3V	IC16 SPI slave select (FPGA output)

In the tables below are listed FX3_I2C interface slave devices and their other information.

Table 8. FX3_I2C interface pins

I2C slave device	Slave device	I2C address	I/O standard	Comment
IC9	Temperature sensor	1 0 0 1 0 0 0 RW	3.3V	LM75
IC12	I2C SPI Bridge	0 1 0 1 0 0 0 RW	3.3V	SC18IS602B
IC10	I2C Port Expander	1 1 0 1 1 0 1 RW	3.3V	MAX7322
IC11	EEPROM	1 0 1 0 0 0 0 RW	3.3V	24FC64F
IC17	Clock generator	1 1 0 0 0 0 0 RW	3.3V	Si5351C

8 GPIOs from FPGA are connected to 10 pin 0.05" header. Additional 2 pins are dedicated for power. In the tables below are listed FPGA_GPIO information.

Table 9. FPGA_GPIO connector pins

Connector pin	Schematic signal name	FPGA pin	I/O standard	Comment
1	GND			

2				Selectable power net. Default 3.3V
3	FPGA_GPIO0	H8	3.3V	
4	FPGA_GPIO1	H6	3.3V	
5	FPGA_GPIO2	H2	3.3V	
6	FPGA_GPIO3	H1	3.3V	
7	FPGA_GPIO4	G4	3.3V	
8	FPGA_GPIO5	G3	3.3V	
9	FPGA_GPIO6	F2	3.3V	
10	FPGA_GPIO7	F1	3.3V	

2.2.6 Board temperature control

LimeSDR-USB has integrated temperature sensor which controls FAN to keep board in operating temperature range. FAN must be connected to J19 (0.1" pitch) connector or to J20 (0.05" pitch) connector. FAN control voltage is 3.3V

Fan will be turned on if board will heat up to 55°C and FAN will be turned off if board will cool down to 45°C.

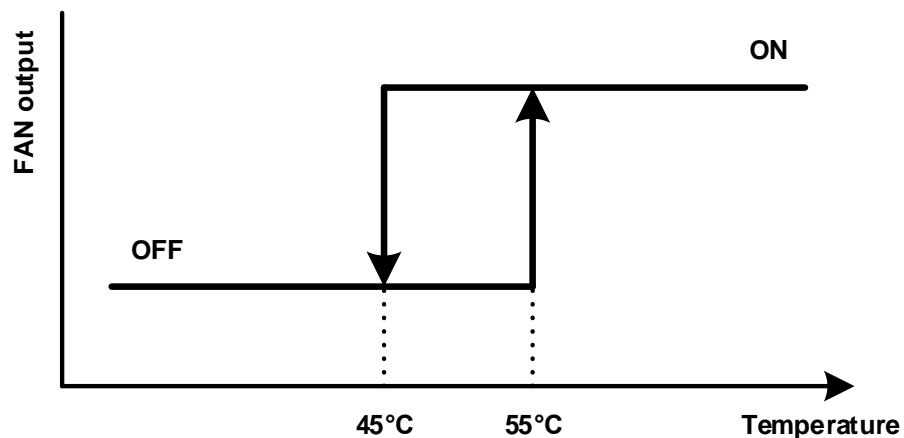


Figure 10 FAN control temperature hysteresis

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

2.2.7 Clock Distribution

LimeSDR-USB board clock distribution block diagram is presented in Figure 11.

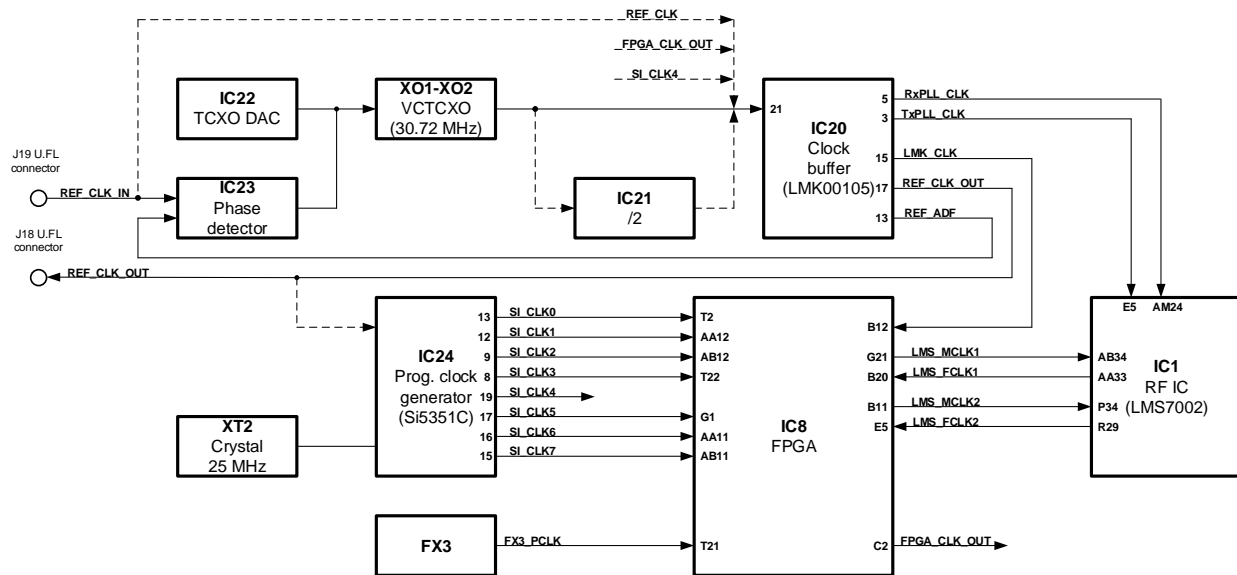


Figure 11 LimeSDR-USB board clock distribution block diagram

LimeSDR-USB board has onboard 30.72 MHz VCTCXO (precision: ± 1 ppm initial, ± 4 ppm stable) that is reference clock for LMS_PLLs. See block diagram of the clock distribution system in *Figure 11*.

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

J19 connector can also be used to supply external reference clock (fitting R147, removing R137, C354).

The programmable clock generator (Si5351C [\[link\]](#)) can generate any reference clock frequency, starting from 8 kHz – 160 MHz, for FPGA and LMS PLLs.

How to configure clocks using LimeSuiteGUI read in chapter 3.12

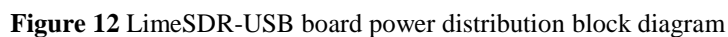
Table 10. LimeSDR-USB clock pins

Source	Schematic signal name	I/O standard	FPGA pin	Description
Programmable clock generator (IC24)	SI_CLK0	1.8V	T2	
	SI_CLK1	1.8V	AA12	
	SI_CLK2	1.8V	AB12	
	SI_CLK3	1.8V	T22	
	SI_CLK5	3.3V	G1	
	SI_CLK6	1.8V	AA11	
	SI_CLK7	1.8V	AB11	
Clock buffer (IC13)	LMK_CLK	3.3V	B12	30.72 MHz
RF transceiver (IC1)	LMS_MCLK1	3.3V	G21	
	LMS_FCLK1	3.3V	B20	
	LMS_MCLK2	3.3V	B11	
	LMS_FCLK2	3.3V	E5	
USB3.0 controller	FX3_PCLK	1.8V	T21	
	FPGA_CLK_OUT	3.3V	C2	

2.2.8 Power Distribution

LimeSDR-USB board can be powered from USB port. In applications where USB power is insufficient board can be powered from external 6..12V power supply. External power supply can be fed to J20 barrel power connector by using power plug (1.35mm ID, 3.5mm OD) or pin header J21 (GND and VCC_EXT). External power supply connections have automatic source selection between USB and external source with polarity protection.

LimeSDR-USB board has complex power delivery network consisting of many different power rails with different voltages, filters, power sequences. LimeSDR-USB board power distribution block diagram is presented in *Figure 12*.



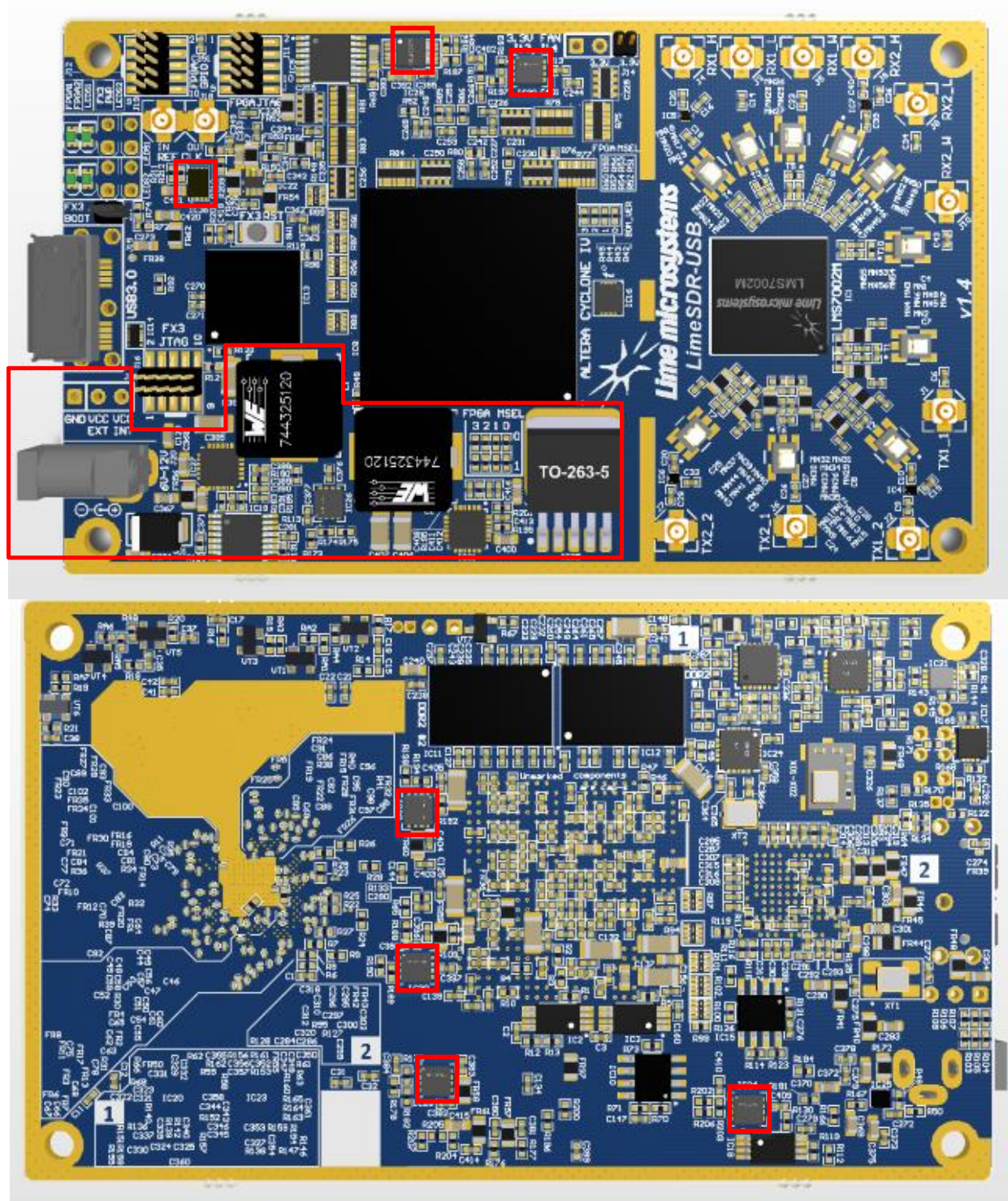


Figure 13 LimeSDR-USB board power ICs on top and bottom

3. Getting Started with LimeSDR-USB

3.1 Launching LimeSuiteGUI and connecting to the board

First of all, connect LimeSDR 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 board. In the provided USB Flash there is a folder GUI in which you will find *LimeSuiteGUI.exe* file. Open it.

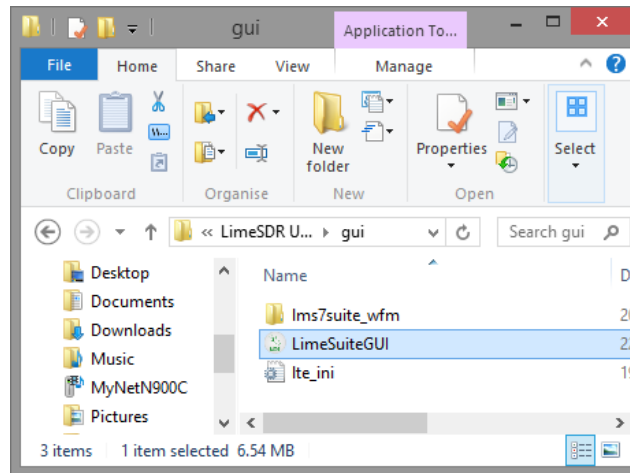


Figure 14 Opening LimeSuiteGUI.exe file

To launch LimeSuiteGUI application go to menu and select: Options -> Connection Settings as shown in *Figure 15*.

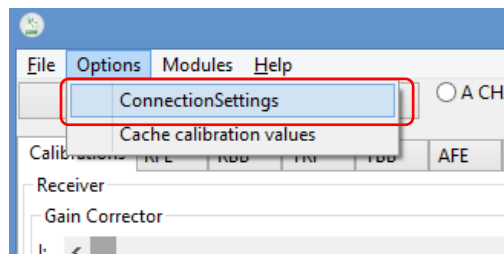


Figure 15 Opening connection settings

Select both ports as **USB3.0 (LimeSDR-USB)**, and press Connect.

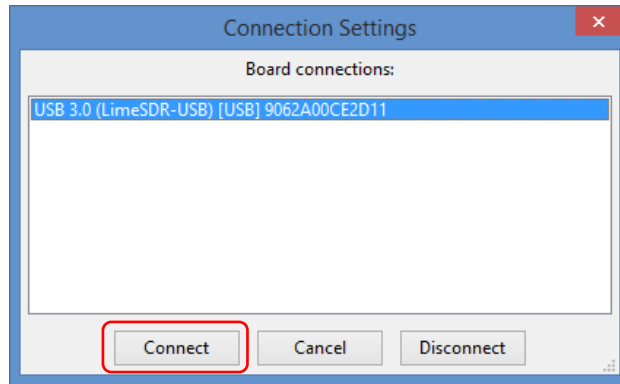


Figure 16 LimeSuiteGUI select ports

3.2 Loading and saving settings

In order to load settings, click button Open.

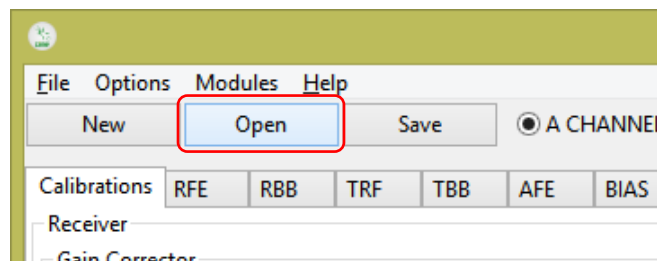


Figure 17 opening LimeSuiteGUI settings file

Select .ini setting file and click Open.

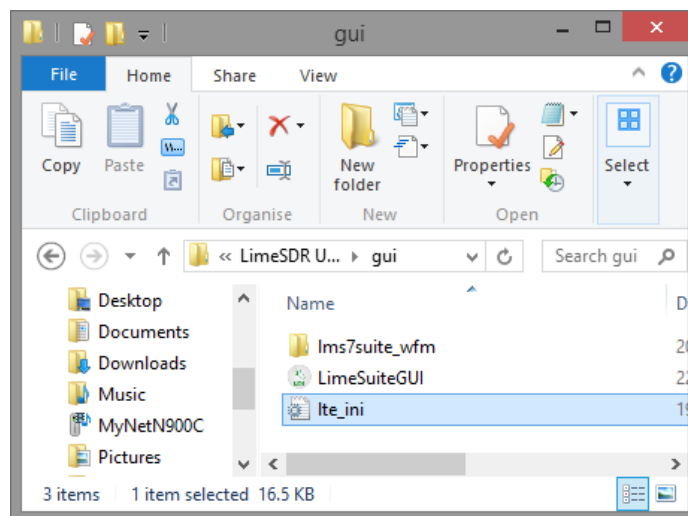


Figure 18 selecting LimeSuiteGUI settings file

and click GUI -- Chip button.

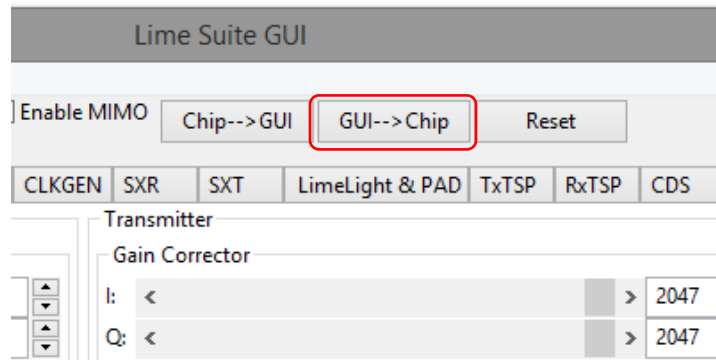


Figure 19 sending LMS7002M settings from GUI to LimeSDR board

If you want load all LMS7002M settings from LimeSDR to GUI, then click button Chip--GUI

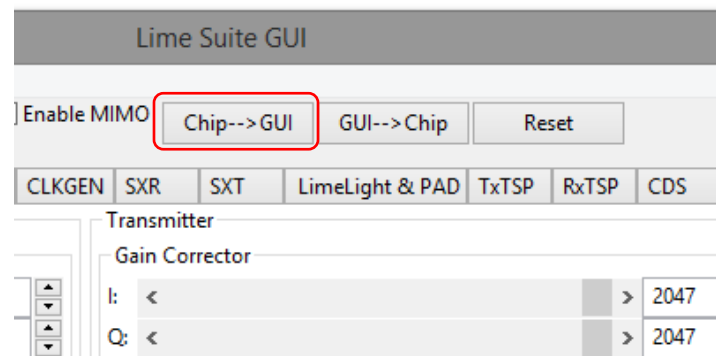


Figure 20 sending LMS7002M settings from LimeSDR to GUI

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](#). Once you see the graph as it shows in Figure 21 of W-CDMA signal on FFT Viewer, you know that the board is working.

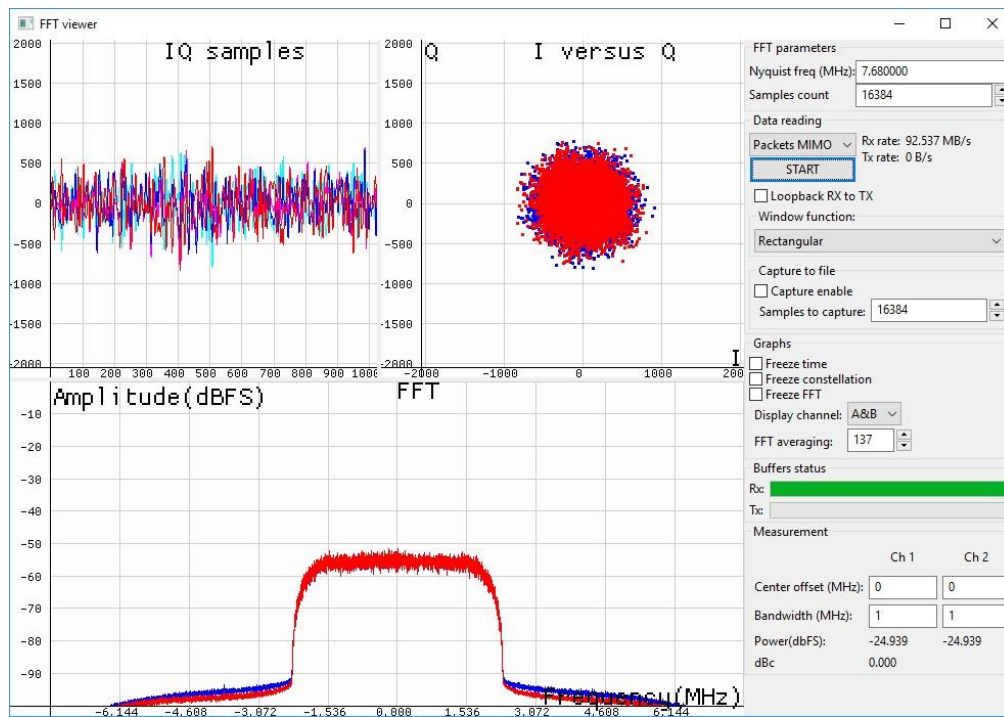


Figure 21 Performing Quick Test

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 22 below to check selections.

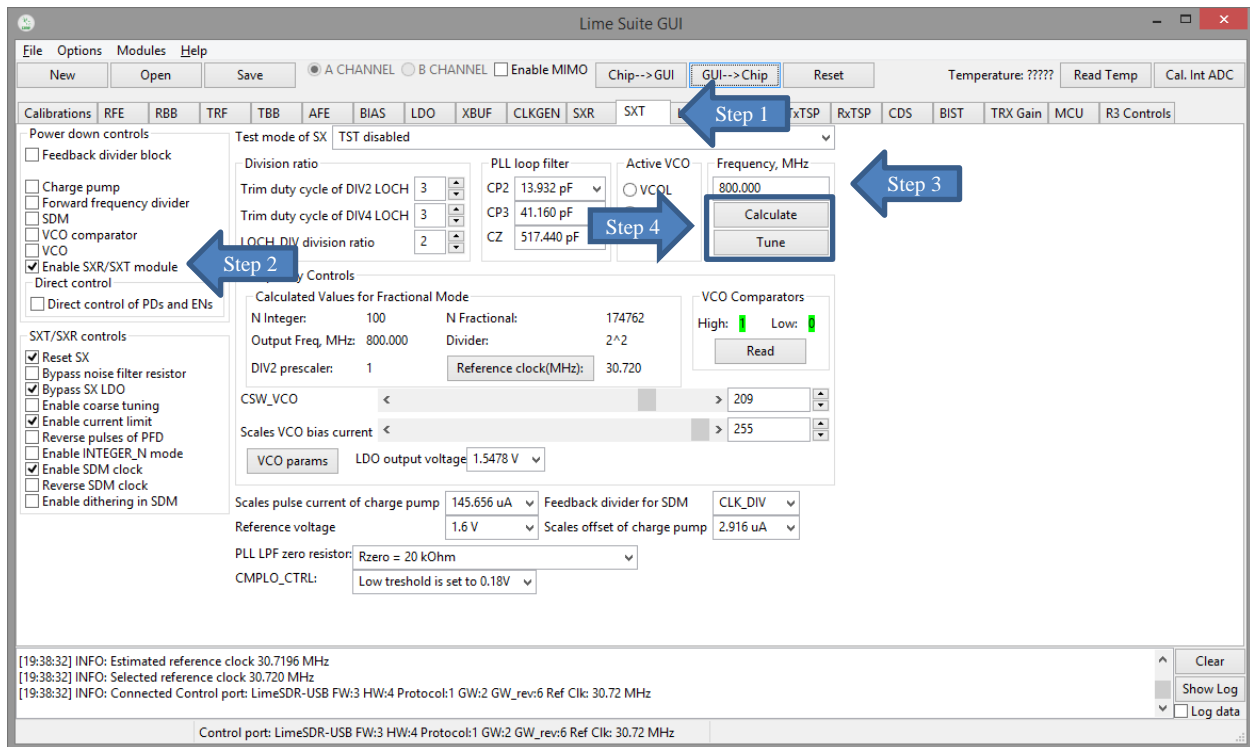


Figure 22 SXR / SXT register setup procedure

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 23* below to check selections.

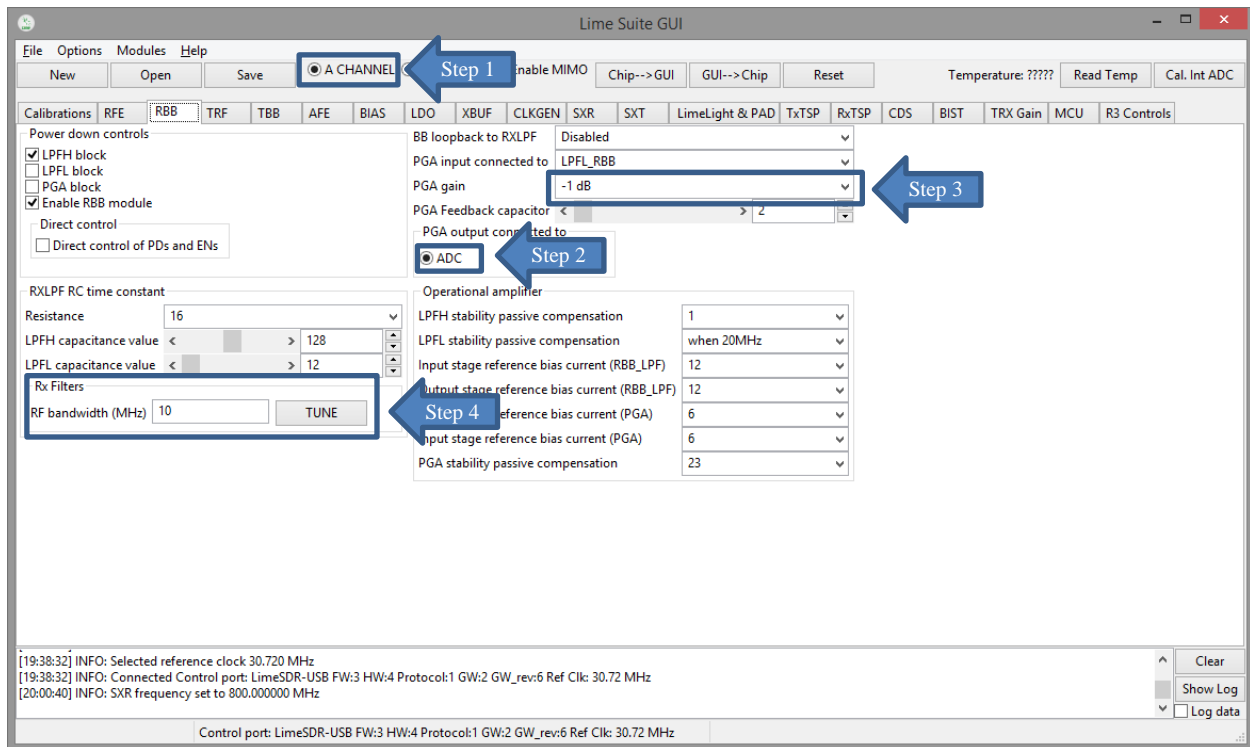


Figure 23 SXR register setup procedure

3.6 Changing Transmitter output signal gain

In the **TBB** tab the baseband gain and filter bandwidth are controlled. Follow the instructions below 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 24* below to check selections.

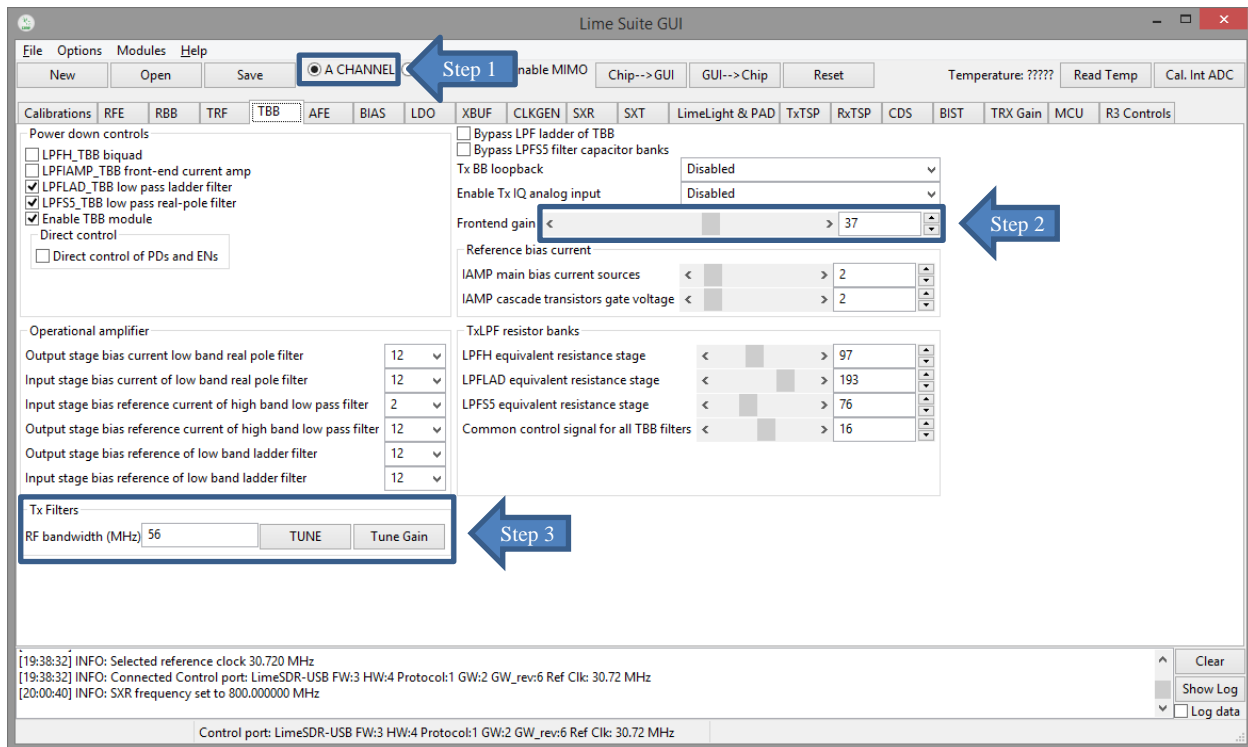


Figure 24 TBB register setup procedure

FFTviewer module is a part of LimeSuiteGUI software. To run FFTviewer, go to top menu, select **Modules** and choose **FFTviewer**. See Figure 29.

3.7 Load waveform for Tx Path

The programmed 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. See Figure 25.

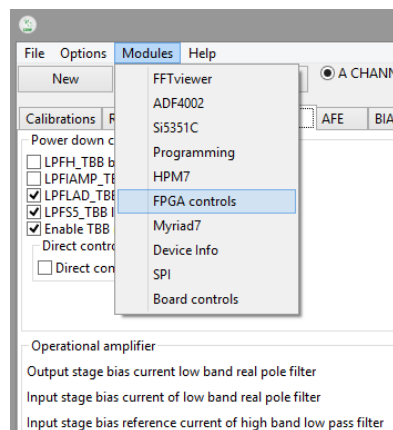


Figure 25 Select FPGA Control window

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. See *Figure 26*.

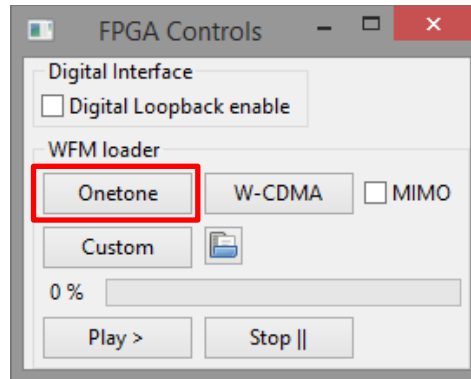


Figure 26 Waveform selection

The file loading process to the FPGA is shown by indication bar, see *Figure 27*.

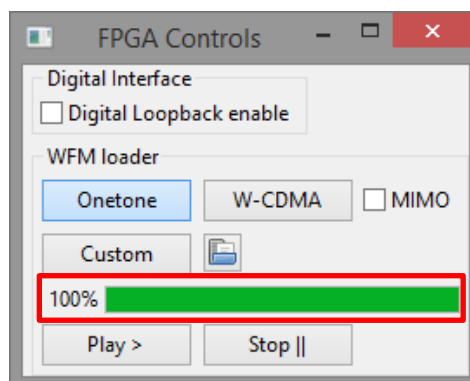


Figure 27 Loaded waveform indication

3.8 Digital Loopback Enable

On this FPGA 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 28*.

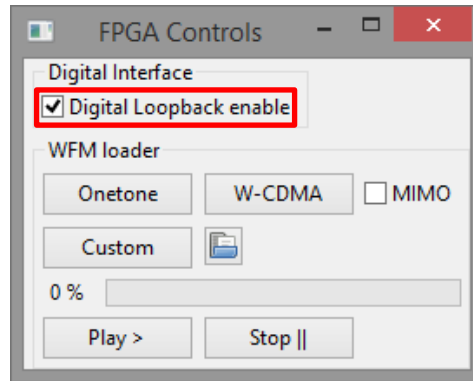


Figure 28 Select Digital Loopback enable

3.9 Run FFT viewer

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

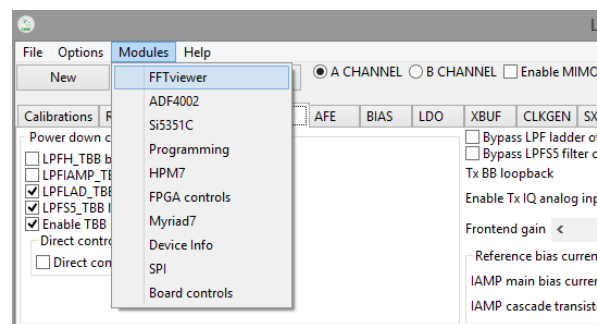


Figure 29 LimeSuiteGUI module menu to select FFTviewer

FFTviewer control window will appear. Before start capturing data, set the **Data reading** type to “Packets MIMO”, Display channel and press **Start**. See Figure 30.

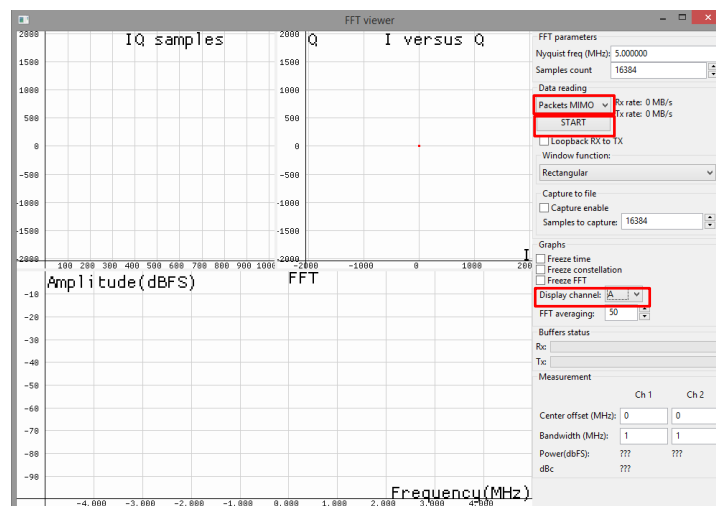
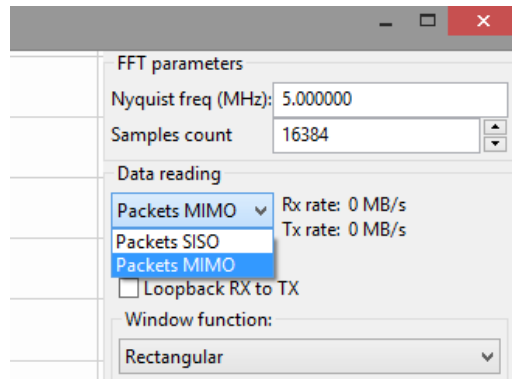
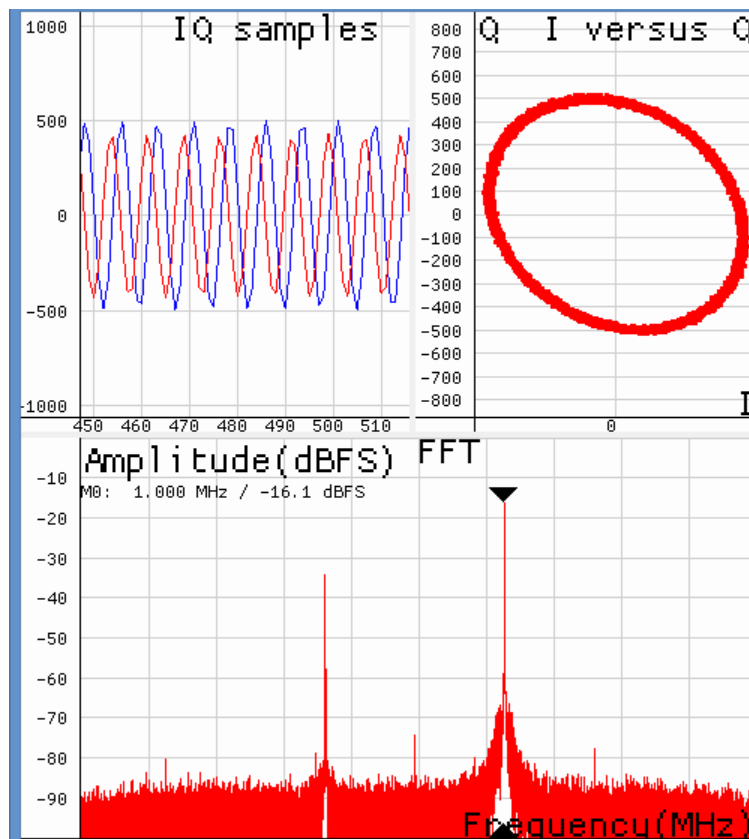


Figure 30 FFTviewer Controls**Figure 31** Setting data type to Packets MIMO

At this point, the FFTviewer start capturing data. Connect the generator to selected LimeSDR receiver path. In the *Figure 32* showed the FFTviewer data capture with 1 MHz CW signal offset from LO.

**Figure 32** FFTviewer window in operation

3.10 Calibrate 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 33* below.

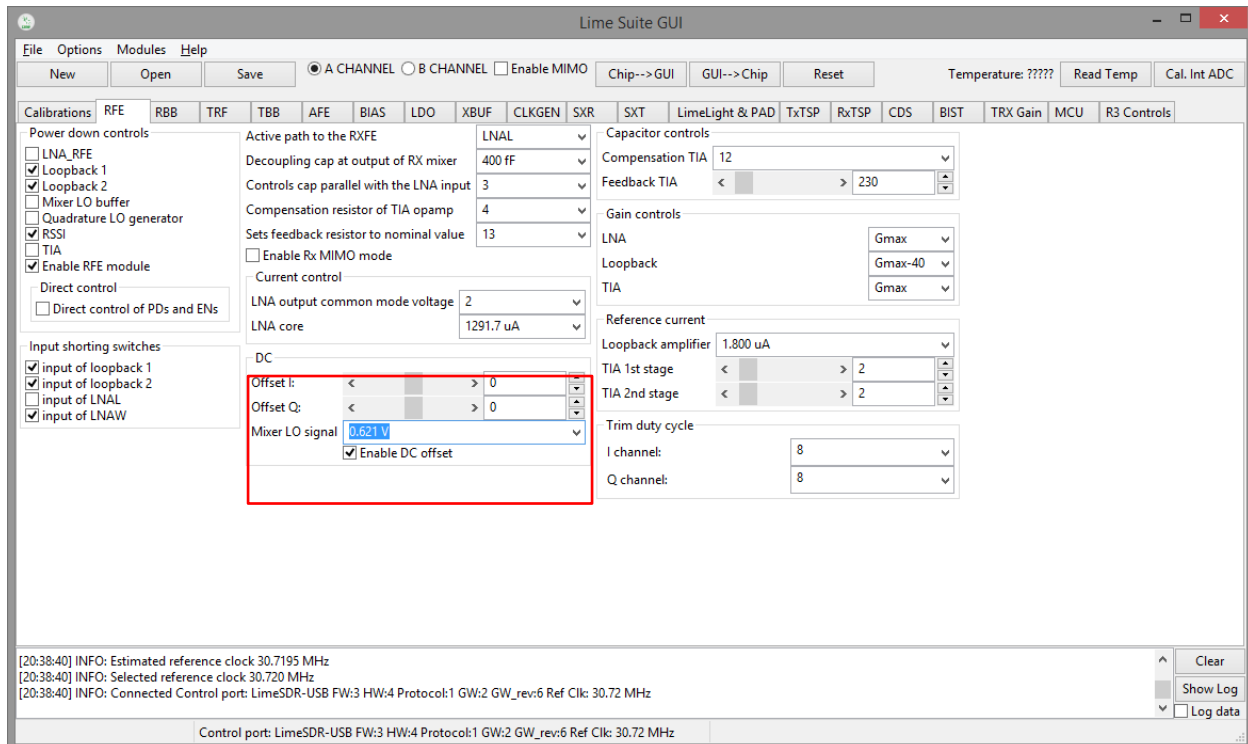


Figure 33 Calibrate RX DC offset

For residual DC offset calibration you need to enable the **DC corrector** in **RxTSP** tab. See *Figure 34*. It should be enabled by default.

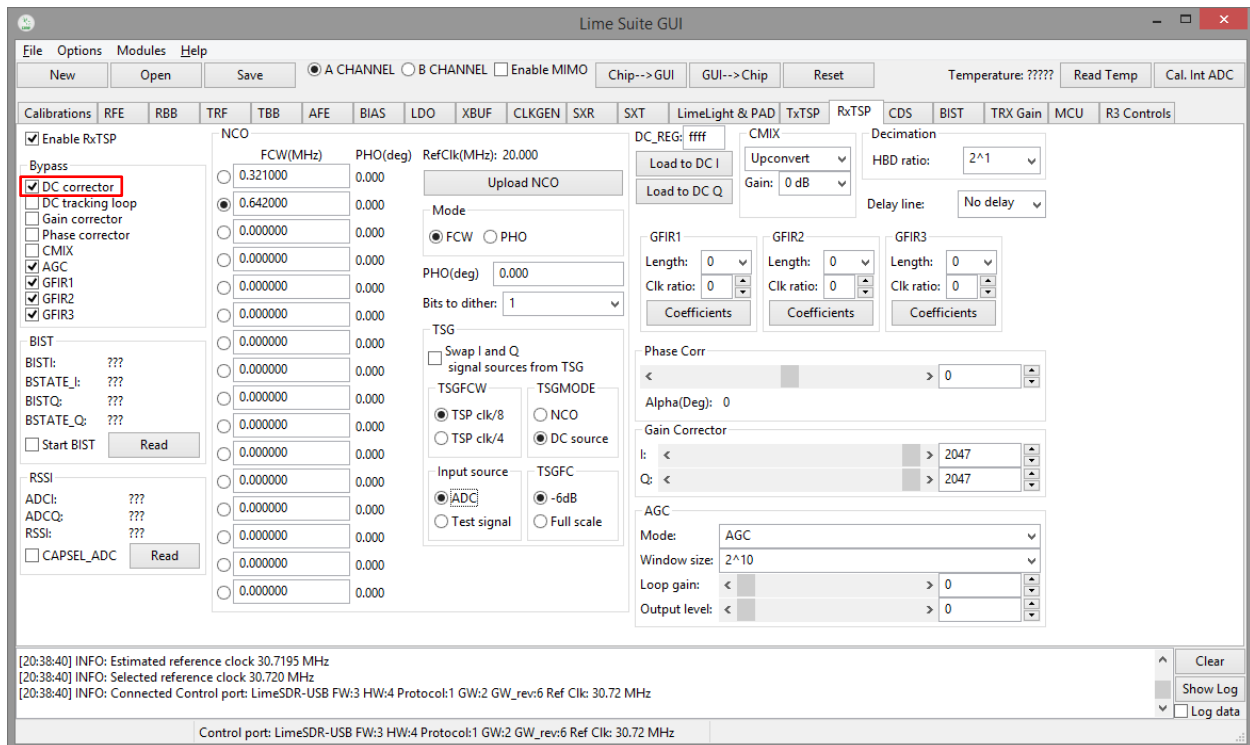


Figure 34 Enable DC corrector in RxTSP

The unwanted SSB can be seen on FFTviewer window by applying signal to one of the transceiver inputs. See Figure 35.

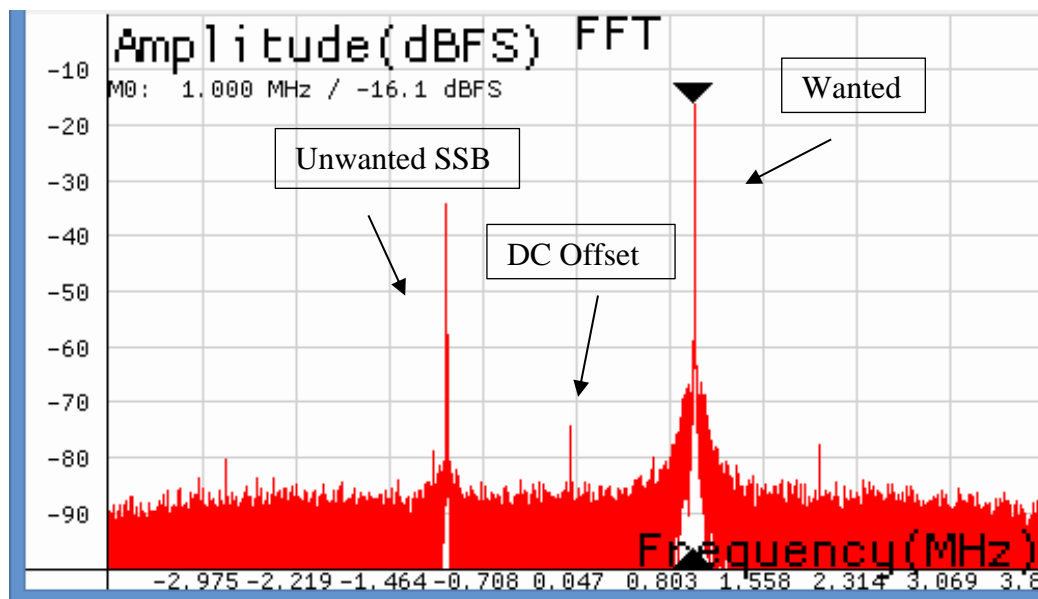


Figure 35 Receiver spectrum with un-calibrated IQ imbalance

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

3.11 Calibrate 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. The LO leakage calibration is done 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, as showed in figure below.

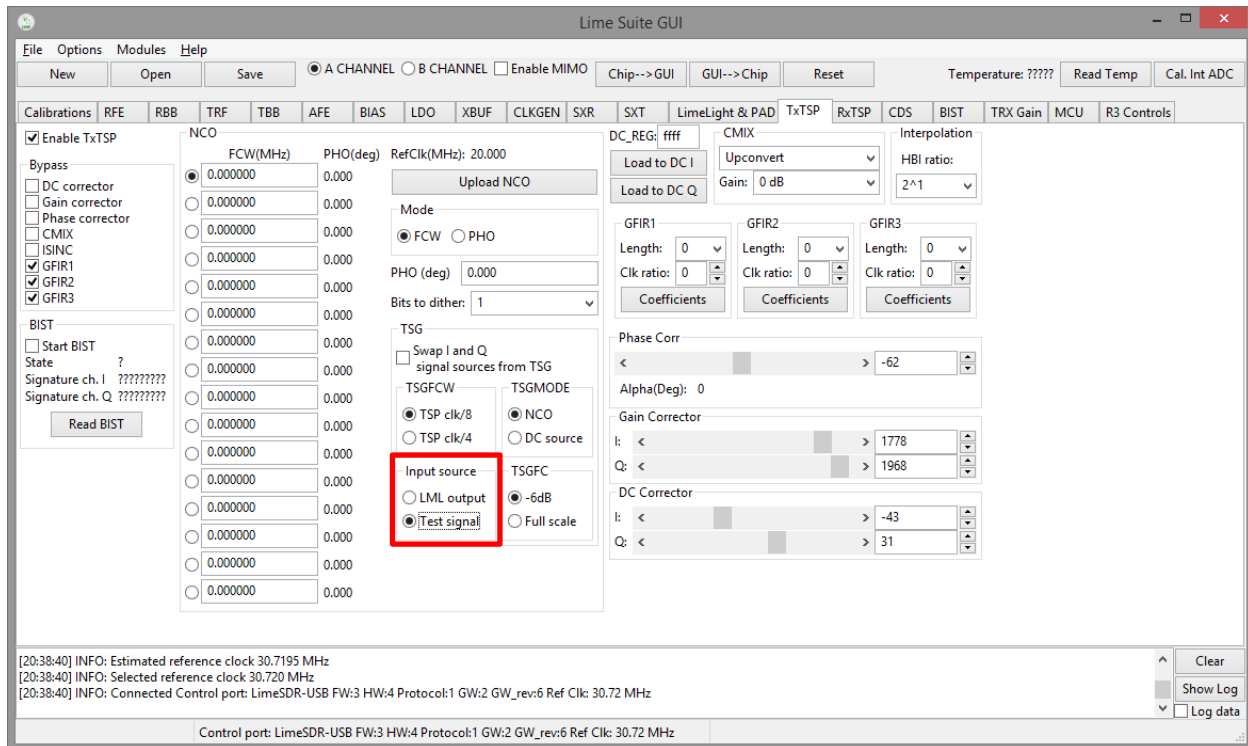


Figure 38 Enable the test NCO

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 39*.

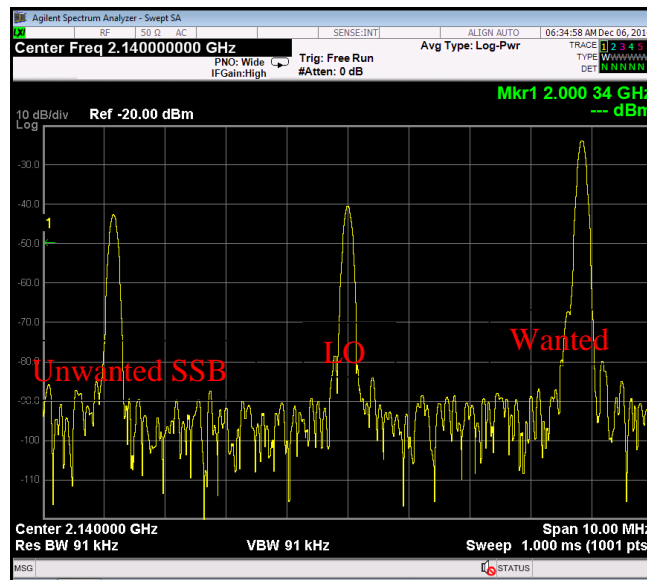


Figure 39 Not calibrated Tx Output

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

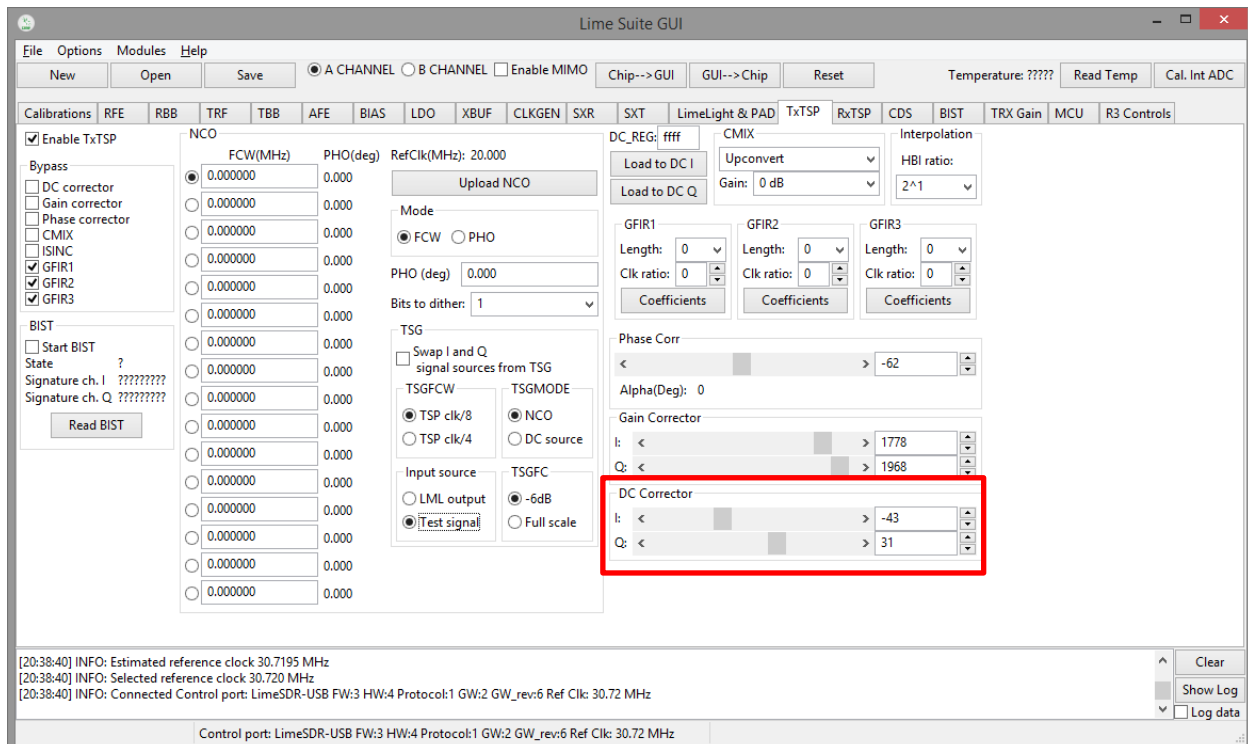


Figure 40 DC offset block control

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.

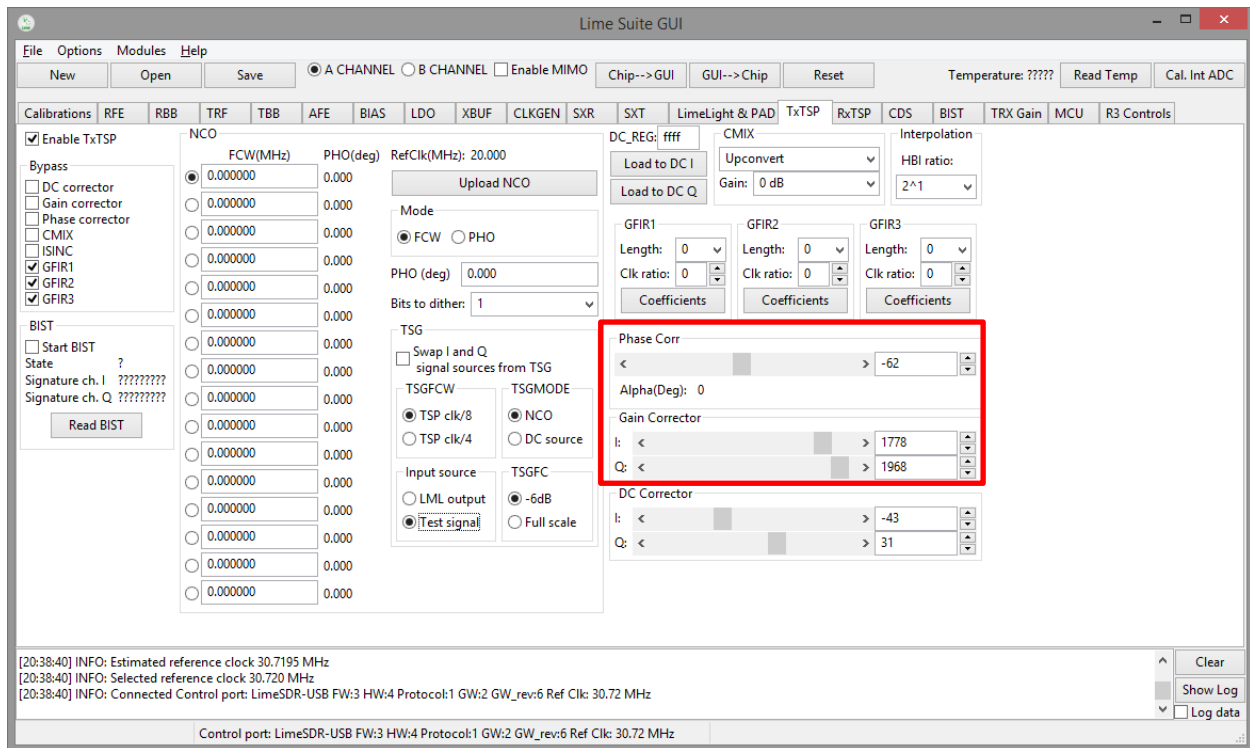


Figure 41 IQ Corrector block control

Calibrated Transceiver TX output should look like in the *Figure 42*.

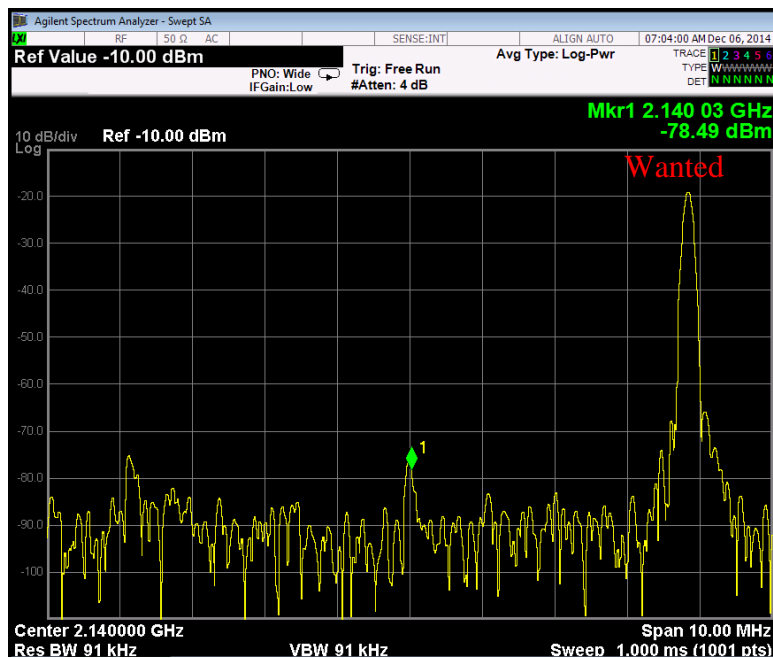


Figure 42 Calibrated Tx output

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 38*.

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 “2.2.7 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. More information about LEDs can be found on section “2.2.4 Indication LEDs”.

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 43*.

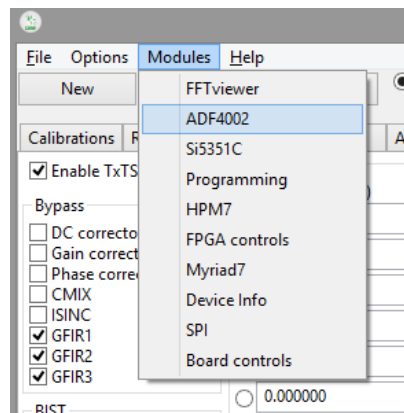


Figure 43 LMS7 suite module menu to select ADF4002 configuration tool

New control section should appear in the bottom of the main window, as shown in the *Figure 44*.

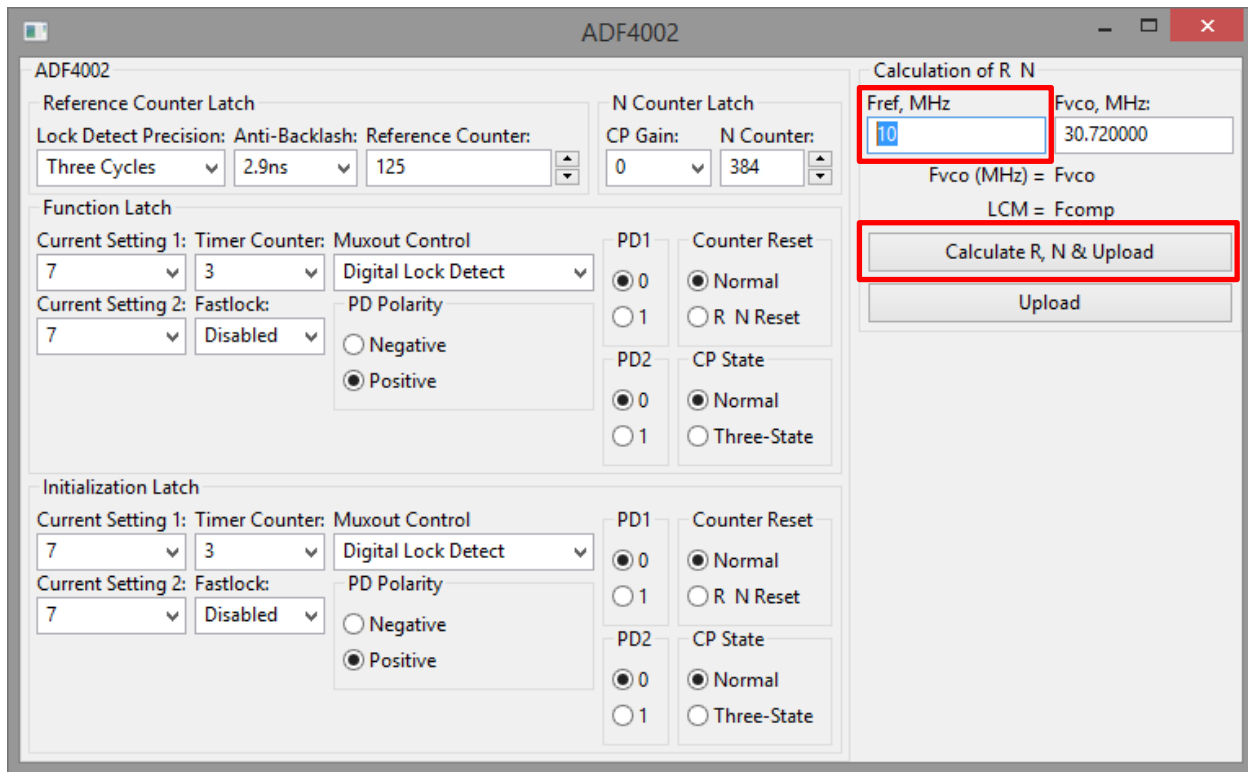


Figure 44 ADF4002 configuration window

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 45*.

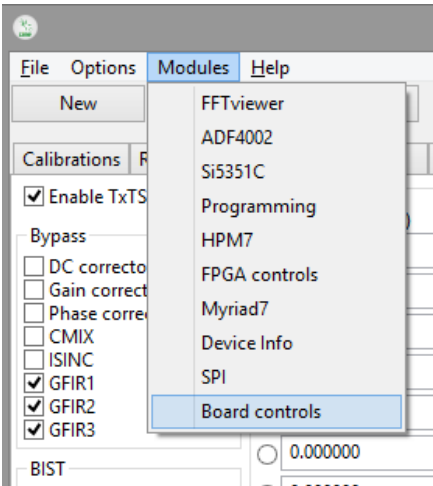


Figure 45 LMS7 suite module menu to select DAC configuration tool

New control section should appear in the bottom of the main window, as shown in the Figure 46.

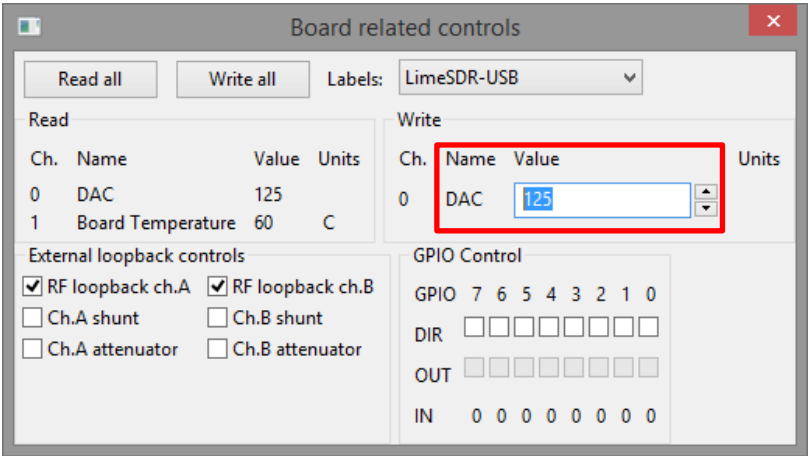


Figure 46 board related controls

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

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.

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 47*.

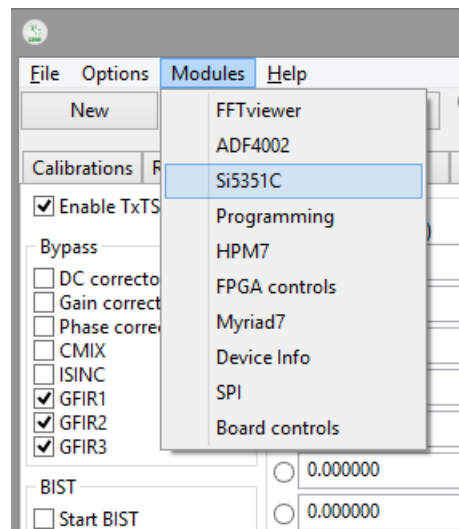


Figure 47 LMS7 suite module menu to select Si5351C configuration tool

New control section should appear in the bottom of the main window, as shown in the Figure 48.

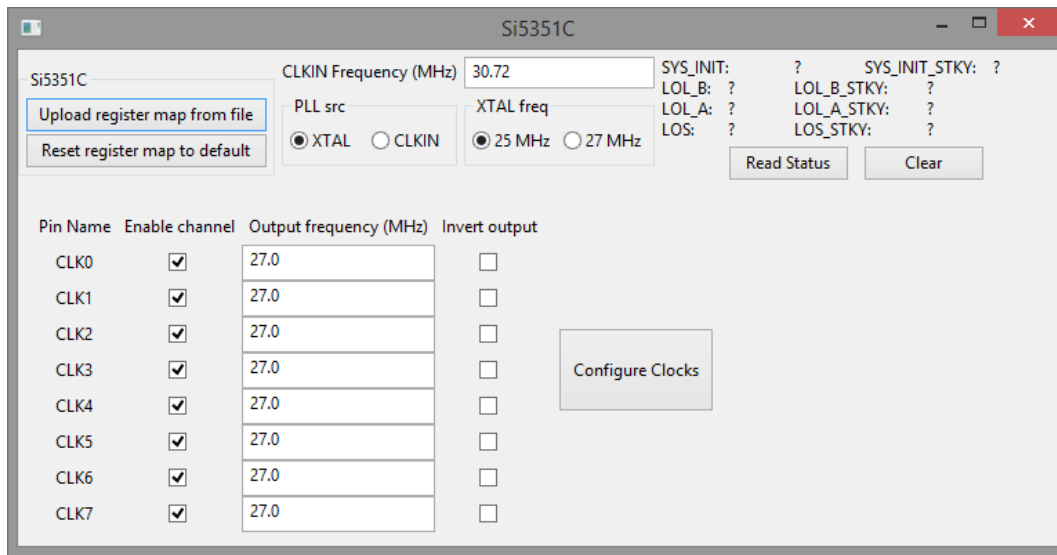


Figure 48 Si5351C configuration window

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 49*.

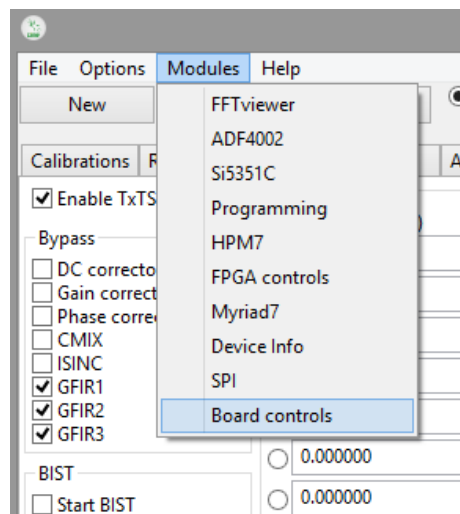


Figure 49 LimeSuiteGUI module menu to select DAC configuration tool

New control section should appear in the bottom of the main window, as shown in the *Figure 50*.

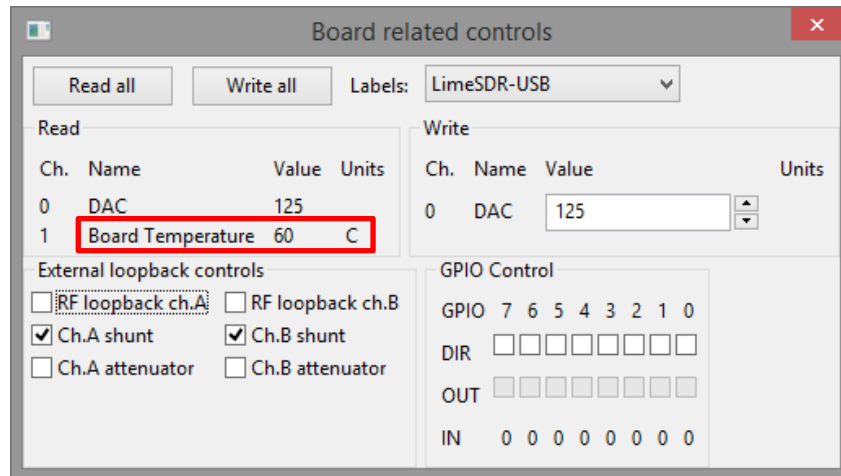


Figure 50 board related controls

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-USB board and PC is done via the USB3 interface. Initially, LimeSDR-USB 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-USB 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 51. The name of the archive will be *Windows-drivers-master.zip* by default. Extract the archive.

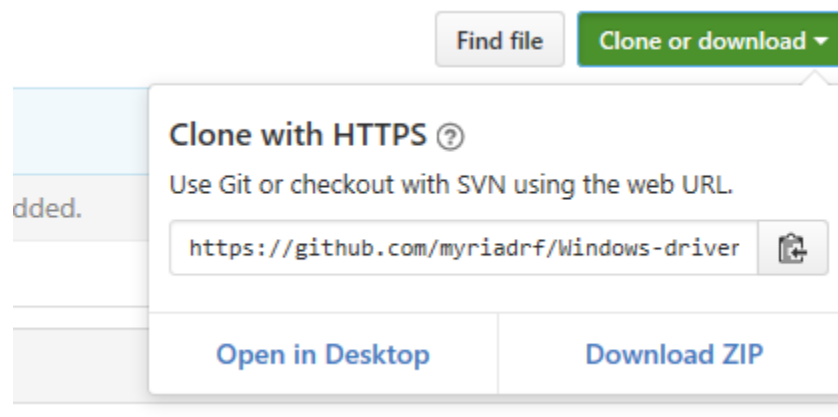


Figure 51 Driver download from GitHub

First time LimeSDR-USB 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**.

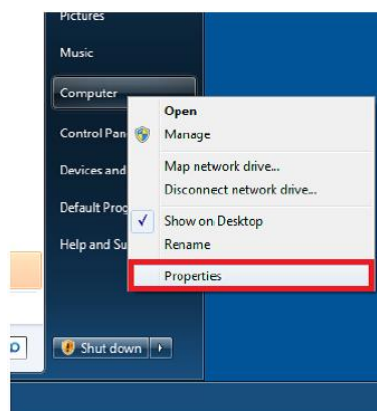


Figure 52 Open computer properties

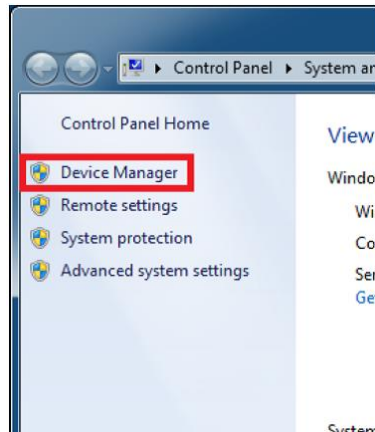


Figure 53 Open device manager

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

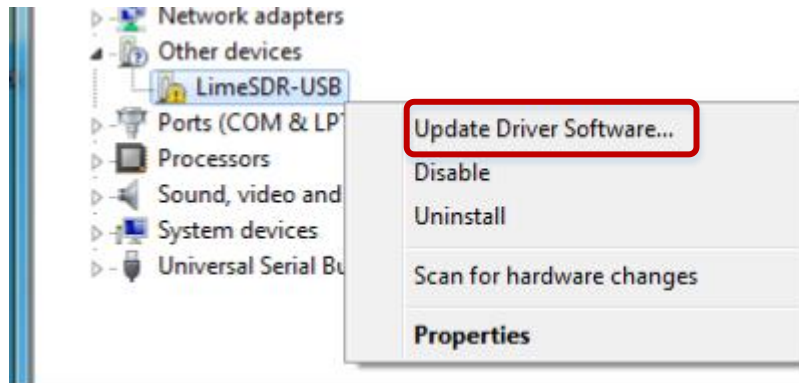


Figure 54 Update driver software

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

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)

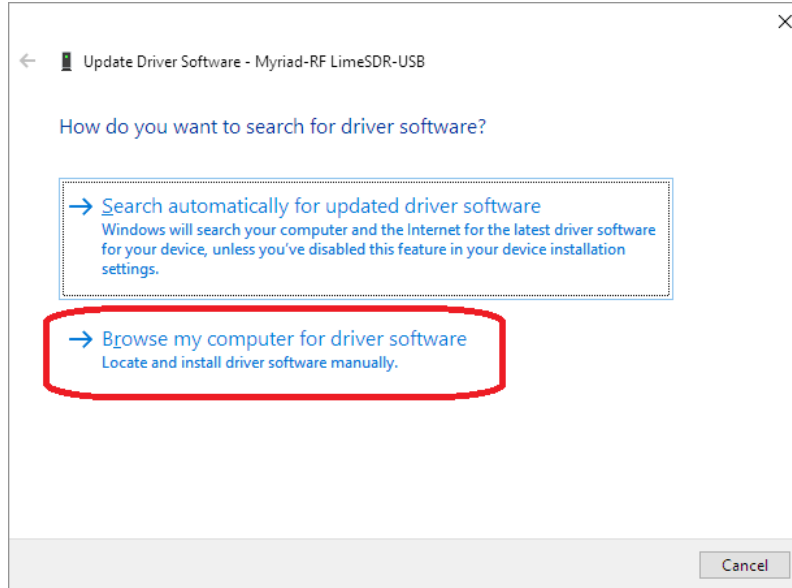


Figure 55 Browse for driver software

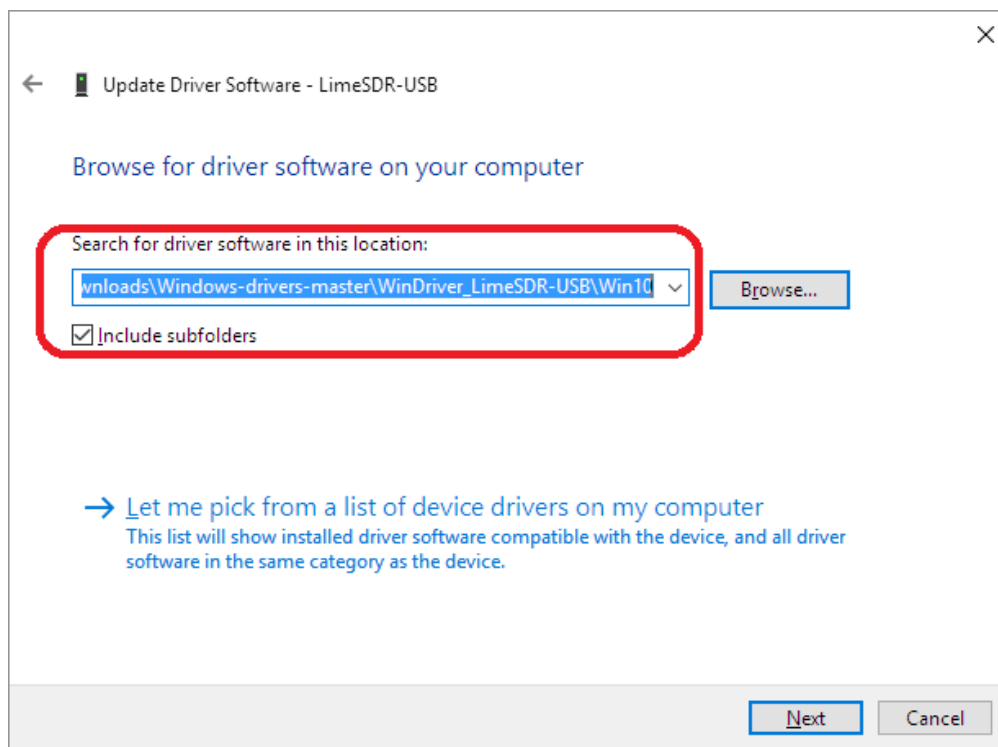


Figure 56 Select driver location

4. After successful installation “*Cypress USB BootLoader*” will appear under USB controller devices.

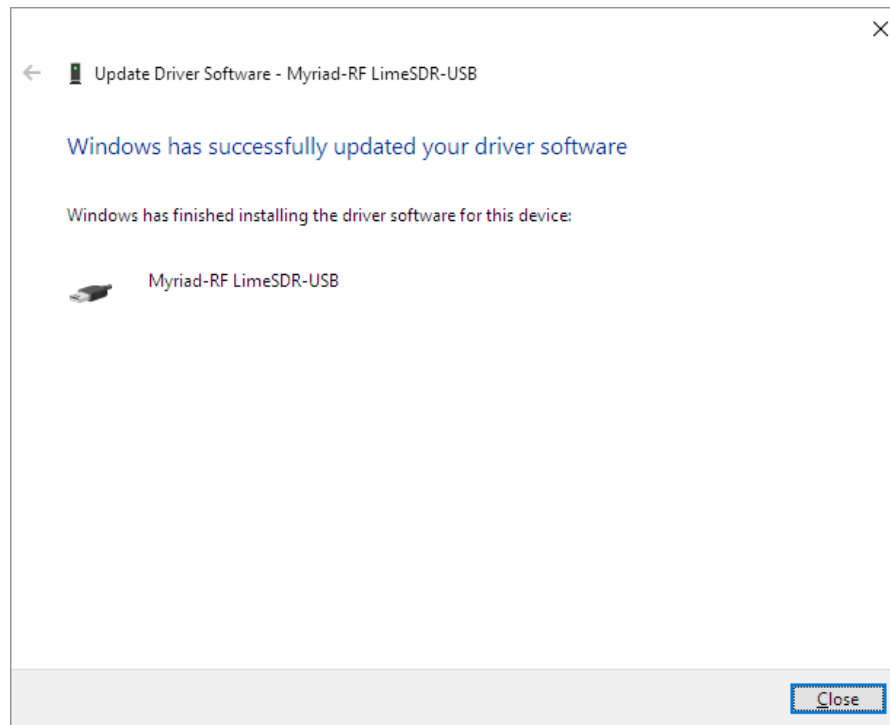


Figure 57 Successful LimeSDR-USB installation

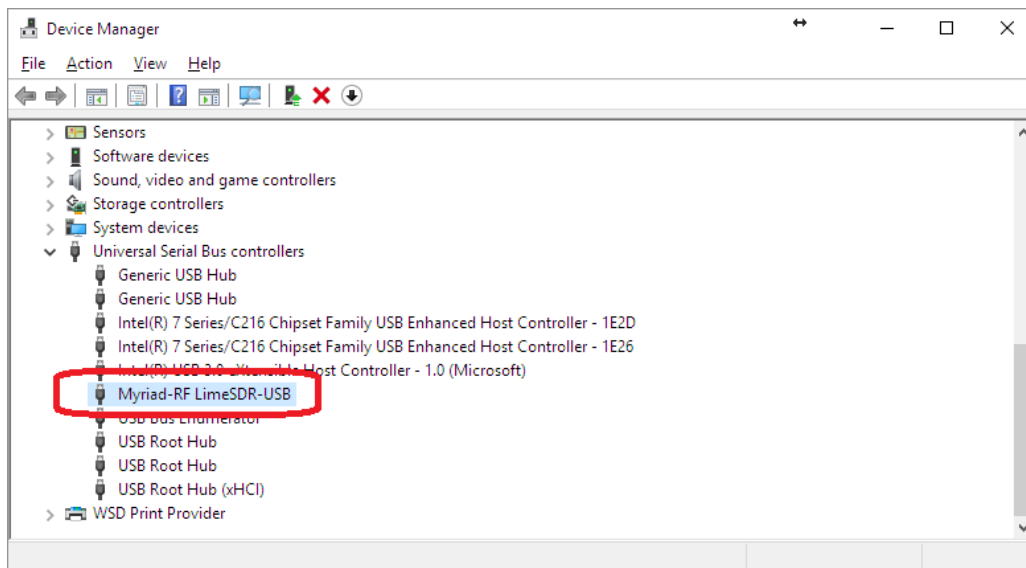


Figure 58 Device manager window after installation

4.2 Linux USB drivers

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

5. Board programming

This section describes how to use LimeSDR-USB board with LMS7 suite software.

5.1 Connecting to the board

To load the file to FPGA, launch *LimeSuiteGUI* application and select: Options->Connection Settings from the menu as shown in Figure 59.

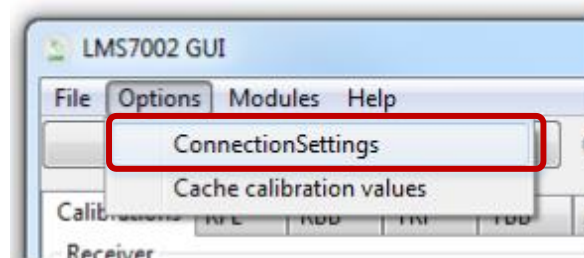


Figure 59 Opening connection settings

Select the board in the dialog window and press Connect button.

5.2 Updating FX3 firmware (normal mode)

The USB controller firmware can be updated using “*LimeSuiteGUI*” software. To call FPGA programming function, go to **Modules** form top menu and select **Programing** form the drop down menu, as shown in Figure 60.

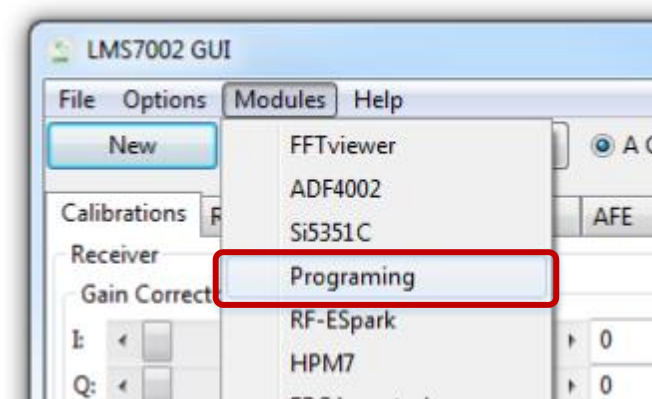


Figure 60 LimeSuiteGUI module menu to select FX3 programming tool

New control section should appear in the bottom of the main window, as shown in the Figure 61.

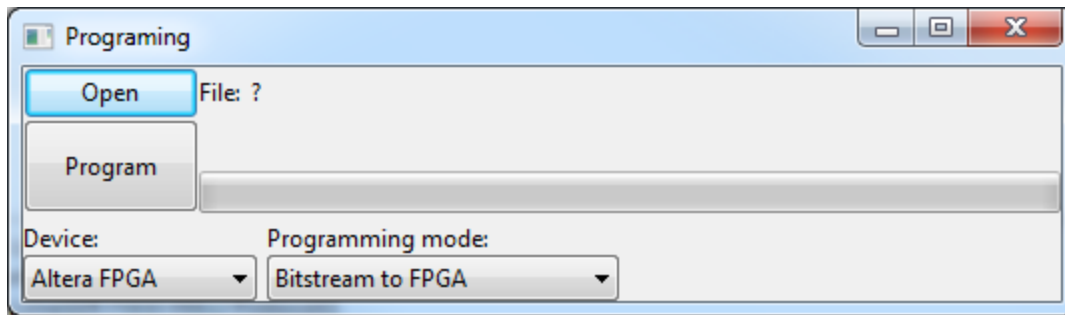


Figure 61 FX3 programing tool interface

Change device to “FX3” and press “Open” to load firmware file.

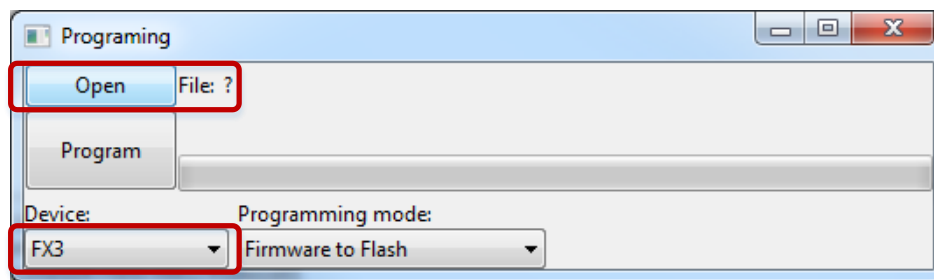


Figure 62 FX3 programing options

Initiate FLASH memory programming by clicking on **Program**.

The new message will come up when the programming is finished, as shown in Figure 63.

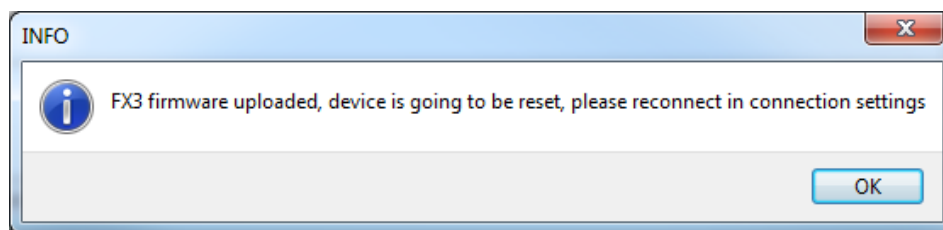


Figure 63 Successfully FX3 programing message

After successful firmware update, connect to board again as described in chapter 3.1.

5.3 Uploading FPGA bitstream to Flash

This section describes how to load custom bitstream to LimeSDR-USB board FPGA Flash memory.

The Altera Cyclone IV FPGA which sits on the LimeSDR-USB board can be programmed using “*LimeSuiteGUI*” software. To call FPGA programming function, go to **Modules** form top menu and select **Programing** form the drop down menu, as shown in *Figure 64*.

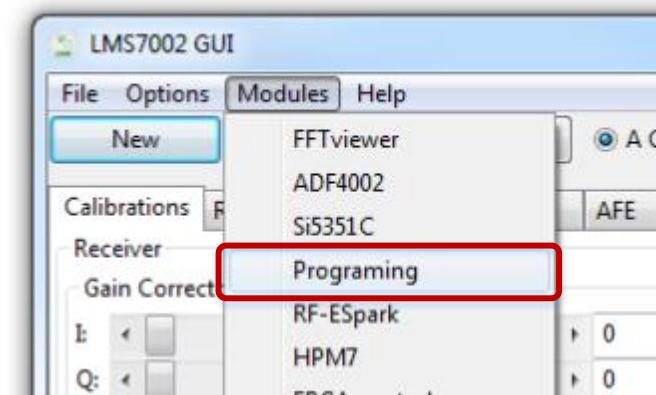


Figure 64 LimeSuiteGUI module menu to select FPGA programming tool

New control section should appear in the bottom of the main window, as shown in the *Figure 65*.

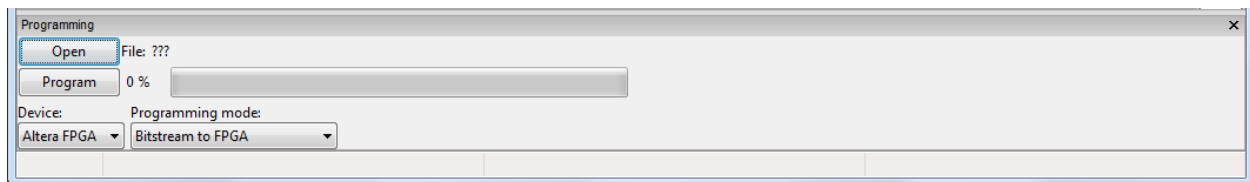


Figure 65 FPGA programming tool interface

Software loads raw binary files (*.rbf) [\[link\]](#) to FPGA and it offers couple options to do that, see *Figure 66*.

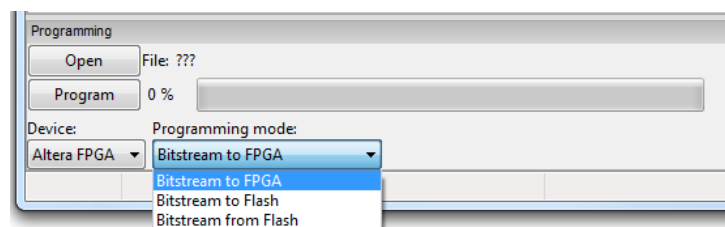


Figure 66 FPGA programming options

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

The new message will come up when the programming is finished, as shown in *Figure 67*.

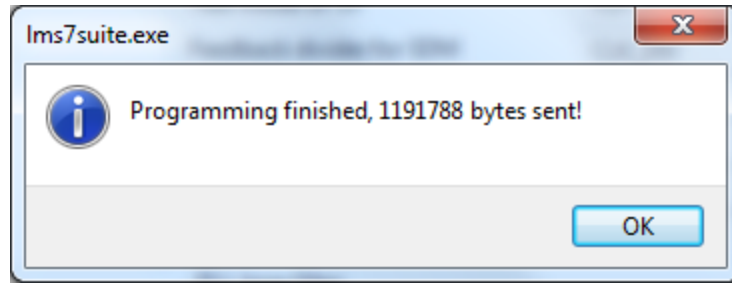


Figure 67 Successfully FPGA programming message (bytes shown may differ)

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

5.4 Flashing USB3 microcontroller firmware in bootloader mode

Cypress FX3 USB microcontroller has an integrated boot loader, which starts automatically after power-up or reset.

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 J13 (on LimeSDR-USB board) is open, USB3 microcontroller boots-up into bootloader mode. For the first time drivers must be installed as described in paragraph “4.1 Windows USB driver installation procedure”. Run the “*USB Control Center*” application and in the menu select **Cypress USB BootLoader** line as shown in *Figure 68*.

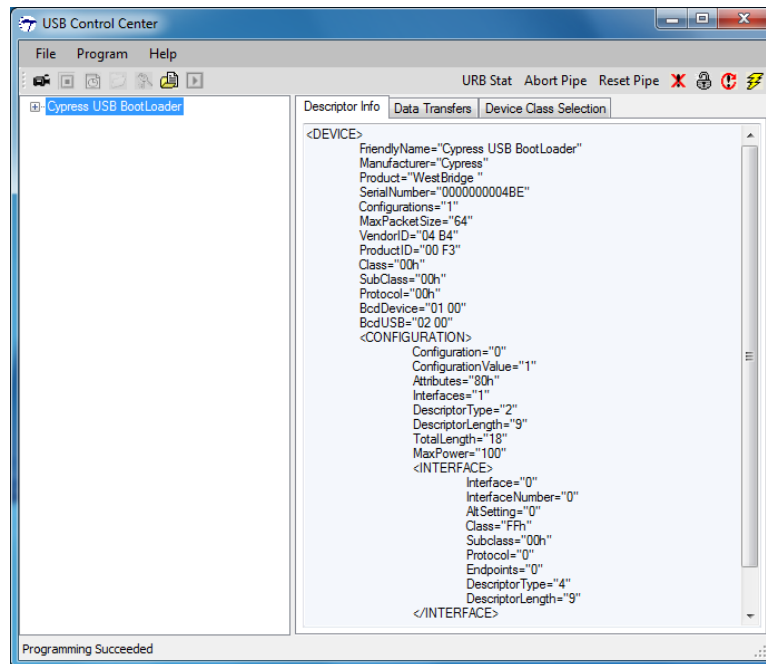


Figure 68 Default FX3 firmware, supplied by internal logic

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 “5.5 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 “5.6 Uploading firmware to the FX3 RAM”. The memory will be cleared after first power cycle.

5.5 Uploading firmware to SPI FLASH

Short the jumper J13 and connect LimeSDR-USB board to the PC. Start “CyControl.exe” application and select **Cypress USB BootLoader** as shown in *Figure 68*. 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 69*.

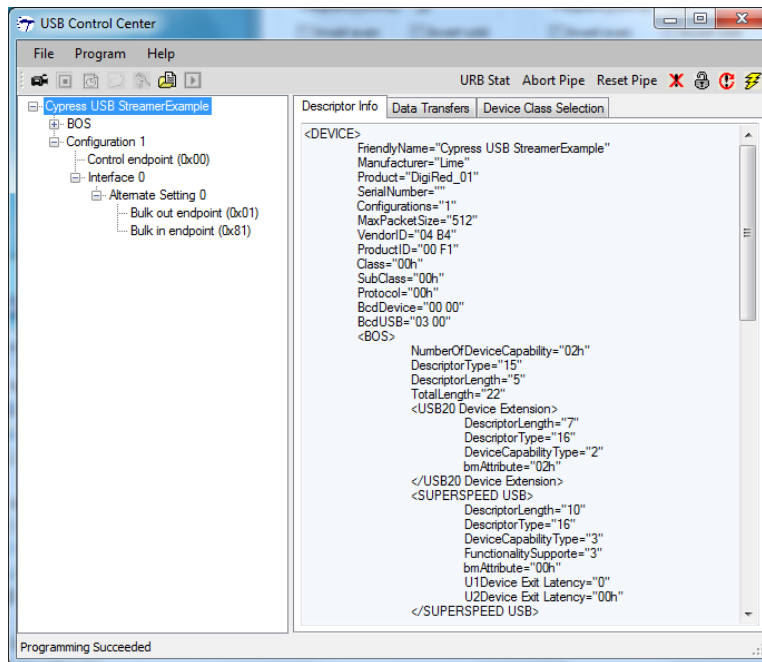


Figure 69 FX3 after custom firmware is downloaded

NOTE: USB3 microcontroller will boot firmware uploaded to FLASH each time after power-on if jumper J3 is shorted.

5.6 Uploading firmware to the FX3 RAM

Start “CyControl.exe” application and select **Cypress USB BootLoader** as shown in *Figure 68*. 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.