
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

LMS8001 Companion Board Mixer Channel Programs Configuration



The information contained in this document is subject to change without prior notice. Lime Microsystems assumes no responsibility for its use, nor for infringement of patents or other rights of third parties. Lime Microsystems' standard terms and conditions apply at all times.

Chip version:	LMS8001A
Chip revision:	2
Document version:	1.0
Document revision:	0

DOCUMENT REVISION HISTORY

The following table shows the revision history of this document:

Date	Version	Description of Revisions
09/02/2026	1.0.0	Initial version created.

CHIP AND BOARD REVISION HISTORY

The following table shows the current chip and the board version:

Chip/Board	Version /Revision
Chip version	LMS8001A
Chip revision	2
Board version	3
Board revision	2

Contents

1	Overview	1
2	Hardware Description	2
2.1	RF mixer channel configuration - LMS8001A.....	2
2.2	RF mixer channel configuration - LMS8001B.....	3
3	Software setup	5
3.1	Setting RF mixer channel programs for LMS8001A	6
3.2	Setting RF mixer channel programs for LMS8001B.....	12
4	Testing the board	17
5	Appendix.....	21
5.1	Arduino IDE RP2040 C-code.....	21

1

Overview

This document describes how to setup different predefined configurations on LMS8001 mixer channel. These predefined configurations, named Programs 0, 1, 2 and 3 can be loaded via SPI signal or, which is described here, via external signals which are connected on GPIO pins. Although this document describes a procedure on LMS8001A chip, very similar procedure could be done on LMS8001B chip.

Chapter 2, **Hardware Description**, explains hardware configuration related to the RF channel mixer. Both chip variants, LMS8001A and LMS8001B are described.

Chapter 3, **Software setup**, outlines the board setup in GUI for both chip variants, LMS8001A and LMS8001B.

Chapter 4, **Testing the board**, provides a description of the test bench and includes screenshots from the spectrum analyser taken during different mixer channel configurations.

Chapter 5, **Appendix**, provides the C-code for the Arduino IDE that was used in the test bench for the microcontroller board, which generates the external signals connected to the GPIO pins of the Companion Board.

It should be noted that the LMS8001 Companion Board is supplied with the LMS8001A chip. Nevertheless, this document also describes the same fast mixer channel configuration switching procedure for the LMS8001B chip, please see the chapters 2.2 and 3.2. There are two reasons for this:

1. Some of our custom-made Companion Boards are specifically manufactured with the LMS8001B chip.
2. The LMS8001 Application Boards use the same GUI (lms8suite) for configuration, and these boards are available with both chip versions, LMS8001A and LMS8001B. Therefore, this document can also be used as an instruction manual for the Application Board.

2

Hardware Description

2.1 RF mixer channel configuration - LMS8001A

A structure of LMS8001A channel is given in Figure 2.1.

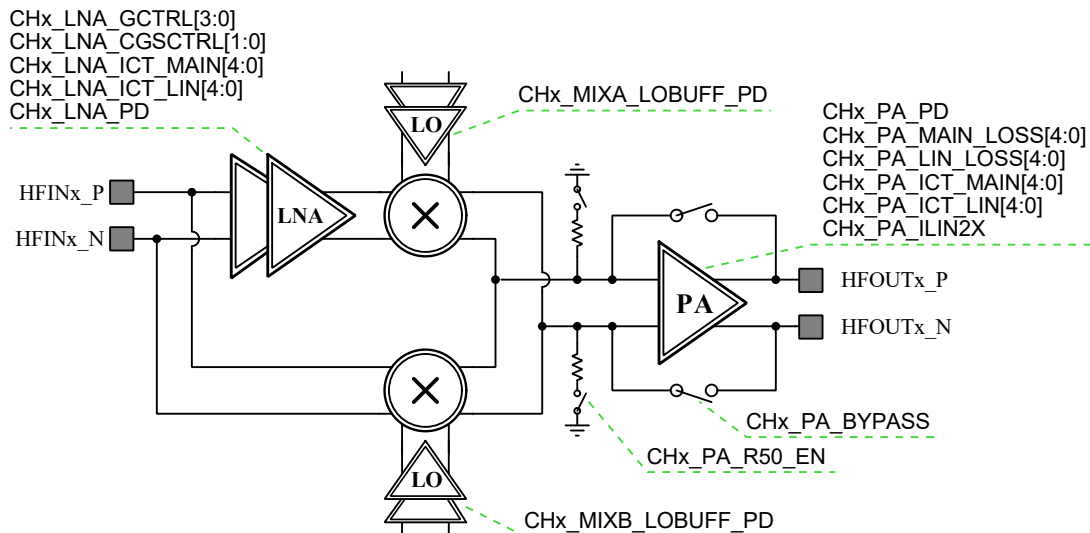


Figure 2.1: RF Channel - LMS8001A

Block diagram which depicts control signals of mixer channel digital circuit is given in Figure 2.2. Control signals are divided into three groups, each with four sets of values, which results in total of 64 possible configurations. Signal group multiplexer control signals are generated by MUXSEL macro, allowing the control via GPIO pins, SPI registers, or combination of them. Registers relevant to RF Channel configuration are grouped into register banks Channel_x (x=A,B,C,D).

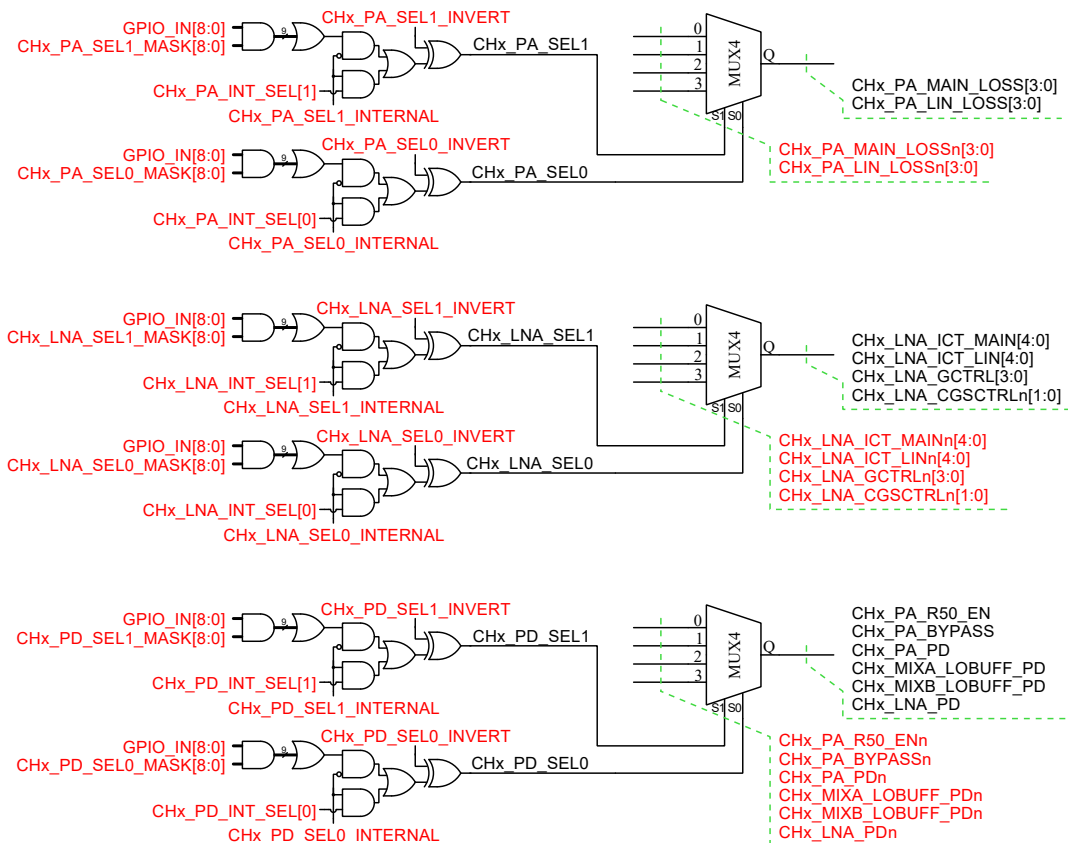


Figure 2.2: RF Channel control signal multiplexing

2.2 RF mixer channel configuration - LMS8001B

A hardware configuration of LMS8001 mixer, which enables mixer channel programs, will be provided in this chapter. A structure of LMS8001B channel is given in Figure 2.3.

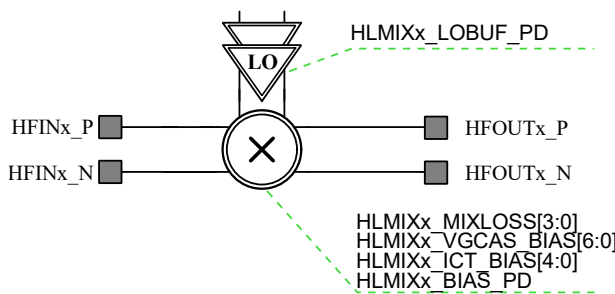


Figure 2.3: High-Linearity Mixer (HLMIX) – LMS8001B

Channel control signals are generated by digital circuit shown in Figure 2.4 **Error! Reference source not found.** Signal group multiplexer control signals are generated by MUXSEL macro, allowing the control via GPIO pins, SPI registers, or combination of them. Registers relevant to HLMIX configuration are grouped into register banks HLMIXx (x=A,B,C,D).

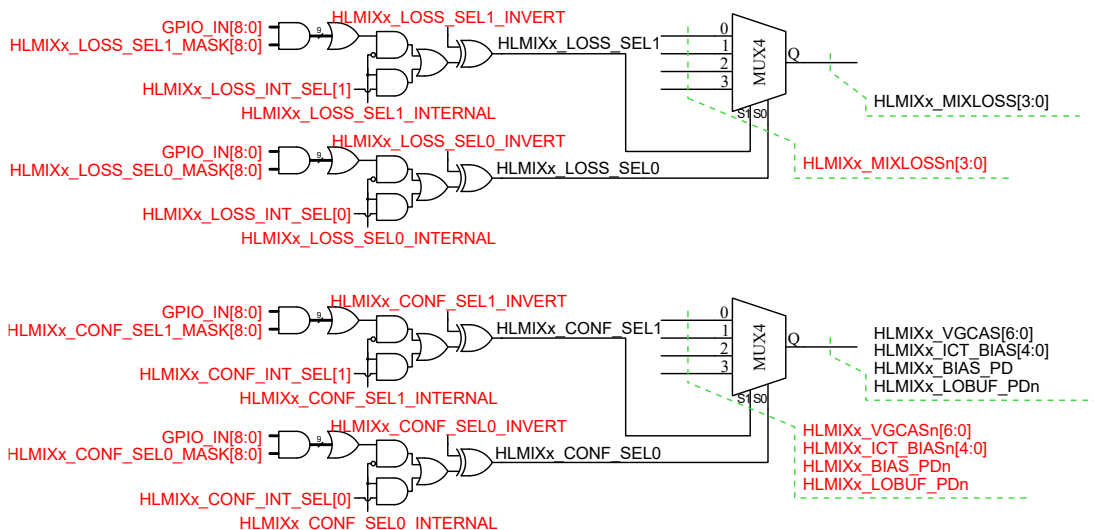


Figure 2.4: HLMIX control signal multiplexing

3

Software setup

The software setup consists of several steps, which are described below. This chapter describes the setup of an LMS8001 mixer channel.

Since GPIO signals will be used, it is necessary to connect the appropriate external signals to the P2 connector, shown in Figure 3.1.

In chapter 0, Channel B is used as a demonstration example, therefore, RF connector J6 should be connected to the signal generator and J9 should be connected to the spectrum analyser.

In chapter 0, Channel C is used as a demonstration example, therefore, RF connector J7 should be connected to the signal generator and J10 should be connected to the spectrum analyser.

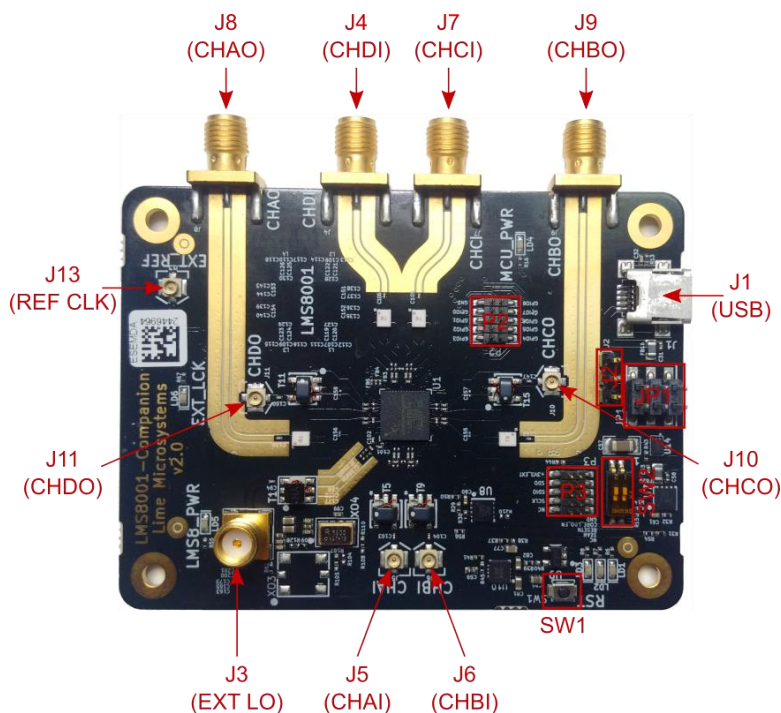


Figure 3.1: LMS8001 Companion Board RF connectors

3.1 Setting RF mixer channel programs for LMS8001A

An example will demonstrate how to create four different configurations for RF Mixer Channel B. Fast configuration selection is achieved using two external signals connected to the P2 connector pins GPIO0 and GPIO1. The configuration procedure for the LMS8001A chip is described below:

1. Connect the appropriate DC supply via connector J2 and select the correct DC supply source using connector JP1, shown in Figure 3.1.
2. Connect the board to the PC using a USB cable. Refer to the USB Type-C connector J1 shown in Figure 3.1.
3. Connect two external signals to the GPIO0 and GPIO1 pins of connector P2. These signals control the current state of RF Mixer Channel.
Pay attention to the DC voltage levels of these signals. They must be 3.3 V when the LMS8001 Companion Board is used in the experiment.
4. Start the GUI application, lms8suite. After this step, the startup window shown in Figure 3.2 should appear.

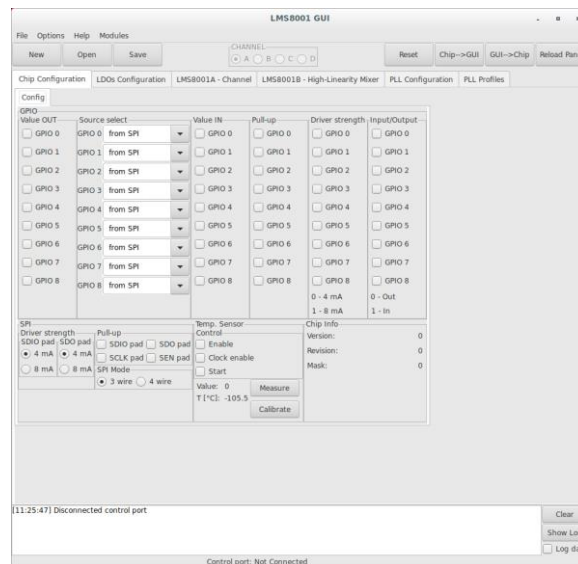


Figure 3.2: Startup window

5. Connect to the board via serial port
 - a. Click the *Options* button and select *Connection Settings*
 - b. The *Connection Settings* window will appear. Select the appropriate serial port
 - c. Click the *Connect* button

LMS8001A RF Mixer Channel Programs Configuration

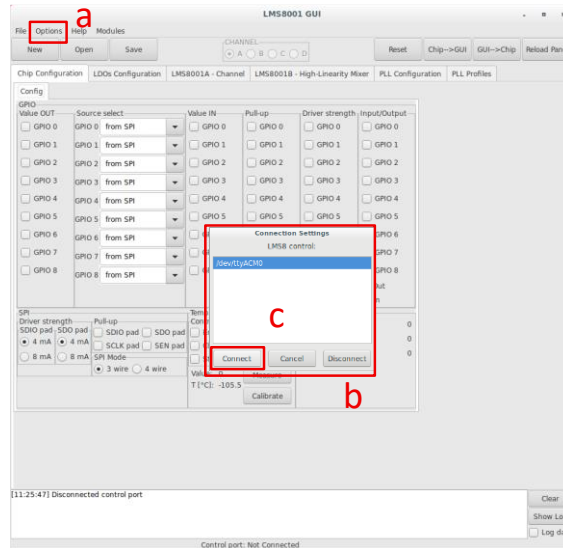


Figure 3.3: Connect to the LMS8001 Companion Board

6. Select the RF Mixer Channel
 - a. Select the *LMS8001A – Channel* tab
 - b. Select *Channel B*

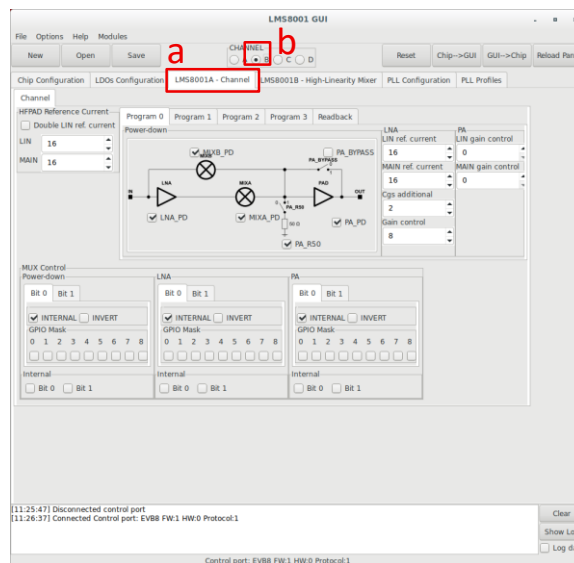


Figure 3.4: Channel selection

7. Set the four predefined configurations – Programs
 - a. Navigate to Programs 0, 1, 2 and 3
 - b. Enable or disable mixer channel subblocks according to the specified configuration

In the figures below, only the states that differ from the startup state are emphasized. Table 1,

Table 2 and Table 3 provide detailed information about each subblock configuration.

LMS8001A RF Mixer Channel Programs Configuration

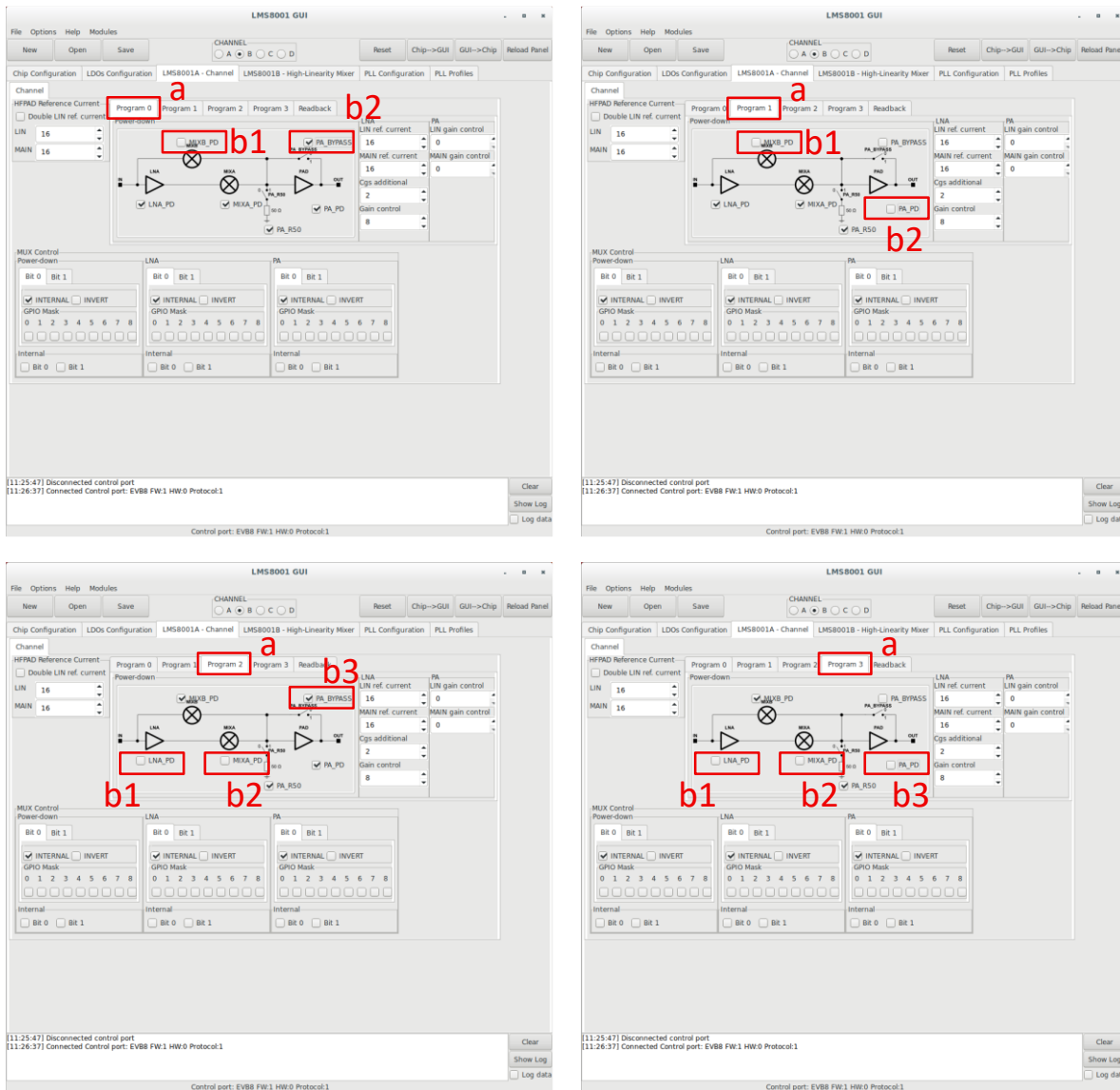


Figure 3.5: Mixer channel Programs configurations

Table 1 – RF Mixer Channel Programs Configurations: Set 1

	LNA_PD	MIXA_PD	MIXB_PD	PA_R50	PA_PD	PA_BYPASS
Program 0	Checked	Checked	Unchecked	Checked	Checked	Checked
Program 1	Checked	Checked	Unchecked	Checked	Unchecked	Unchecked
Program 2	Unchecked	Unchecked	Checked	Checked	Checked	Checked
Program 3	Unchecked	Unchecked	Checked	Checked	Unchecked	Unchecked

Table 2 – RF Mixer Channel Programs Configurations: Set 2

	LIN ref. current	MAIN ref. current	Cgs additional	Gain Control
Program 0	16	16	2	8
Program 1	16	16	2	8
Program 2	16	16	2	8
Program 3	16	16	2	8

Table 3 – RF Mixer Channel Programs Configurations: Set 3

	LIN gain control	MAIN gain control
Program 0	0	0
Program 1	0	0
Program 2	0	0
Program 3	0	0

This example demonstrates how GPIO signals can be used to control the active RF Mixer channel configuration. Four predefined configurations are provided in the GUI and are named Program 0, Program 1, Program 2, and Program 3.

Each program actually contains three independent sets of parameters, as shown in Table 1,

Table 2, Table 3 and in Figure 3.6, resulting in a total of 64 possible predefined configurations. However, since the parameters in Set 2 and Set 3 do not vary in this example (see

Table 2 and Table 3), only four different configurations are demonstrated here.

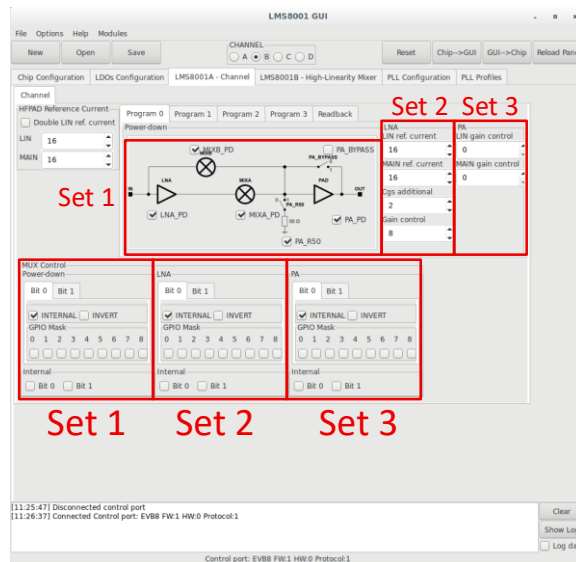


Figure 3.6: Independent sets for RF mixer channel configuration

8. Configuring the control GPIO signals
 - a. Select Bit 0 or Bit 1
 - b. Uncheck box INTERNAL
 - c. Connect control Bit 0 or Bit 1 to the desired GPIO pins by selecting the corresponding checkboxes in the GPIO Mask

The same procedure should be applied to all Bit 0 and Bit 1, in all three sets, as shown in Figure 3.7.

LMS8001A RF Mixer Channel Programs Configuration



Figure 3.7: Setting control bits (Bit 0 and Bit 1 for all three SETs) and connecting them with GPIO pins

9. Finalizing the setup procedure
 - a. Select the *PLL Profiles* tab
 - b. Set the LO frequency
 - c. Click the *Smart Tune* button
 - d. Check if LO frequency is successfully generated
 - e. Connect the LO signal to Channel B and verify its operation

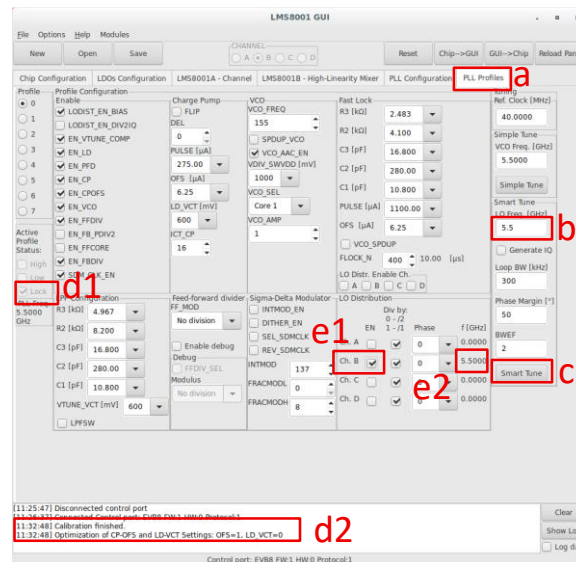


Figure 3.8: Setting LO frequency

After completing all the steps described above, the LMS8001A chip is properly configured. The signal should appear on the spectrum analyser, and its changes should be synchronized with the two external signals connected to the GPIO0 and GPIO1 pins of connector P2.

3.2 Setting RF mixer channel programs for LMS8001B

An example will demonstrate how to create four different configurations for RF Mixer Channel C. Fast configuration selection is achieved using two external signals connected to the P2 connector pins GPIO0 and GPIO1. The configuration procedure for the LMS8001B chip is described below:

1. Connect the appropriate DC supply via connector J2 and select the correct DC supply source using connector JP1, shown in Figure 3.1.
2. Connect the board to the PC using a USB cable. Refer to the USB Type-C connector J1 shown in Figure 3.1.
3. Connect two external signals to the GPIO0 and GPIO1 pins of connector P2. These signals control the current state of RF Mixer Channel.
Pay attention to the DC voltage levels of these signals. They must be 3.3 V when the LMS8001 Companion Board is used in the experiment.
4. Start the GUI application, lms8suite. After this step, the startup window shown in Figure 3.9 should appear.

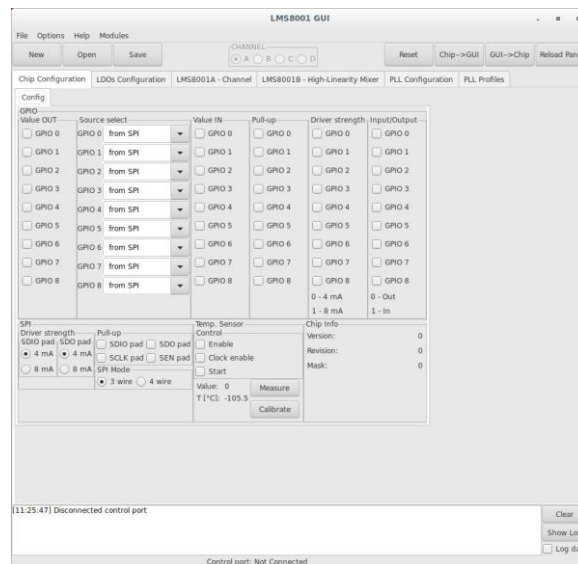


Figure 3.9: Startup window

5. Connect to the board via serial port
 - a. Click the *Options* button and select Connection Settings
 - b. The *Connection Settings* window will appear. Select the appropriate serial port
 - c. Click the *Connect* button

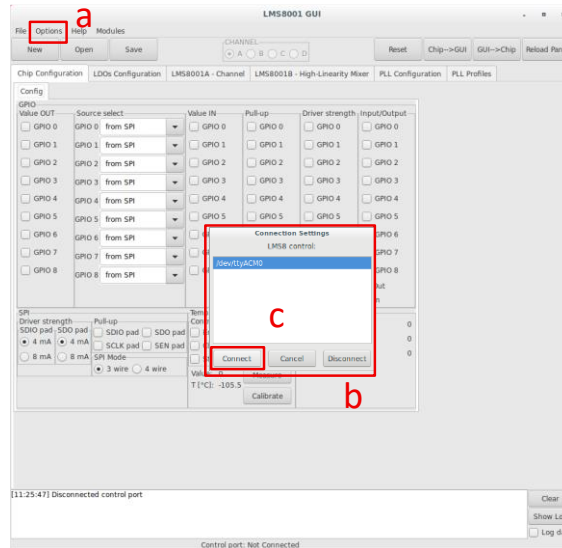


Figure 3.10: Connect to the LMS8001 Companion Board

6. Select the RF Mixer Channel
 - a. Select the *LMS8001B – High-Linearity Mixer* tab
 - b. Select *Channel B*

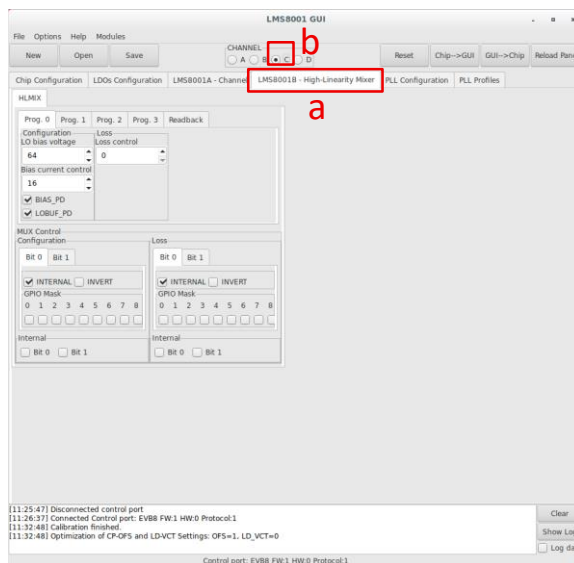


Figure 3.11: Channel selection

7. Set the four predefined configurations – Programs
 - a. Navigate to Programs 0, 1, 2 and 3
 - b. Turn on BIAS and LOBUF by unchecking the corresponding boxes
 - c. Change the Loss control for Programs 0, 1, 2 and 3 on the values 4, 8, 12 and 15, respectively.

In the figures below, only the states that differ from the startup state are emphasized. Table 4 and Table 5 provide detailed information about each subblock configuration.

LMS8001A RF Mixer Channel Programs Configuration

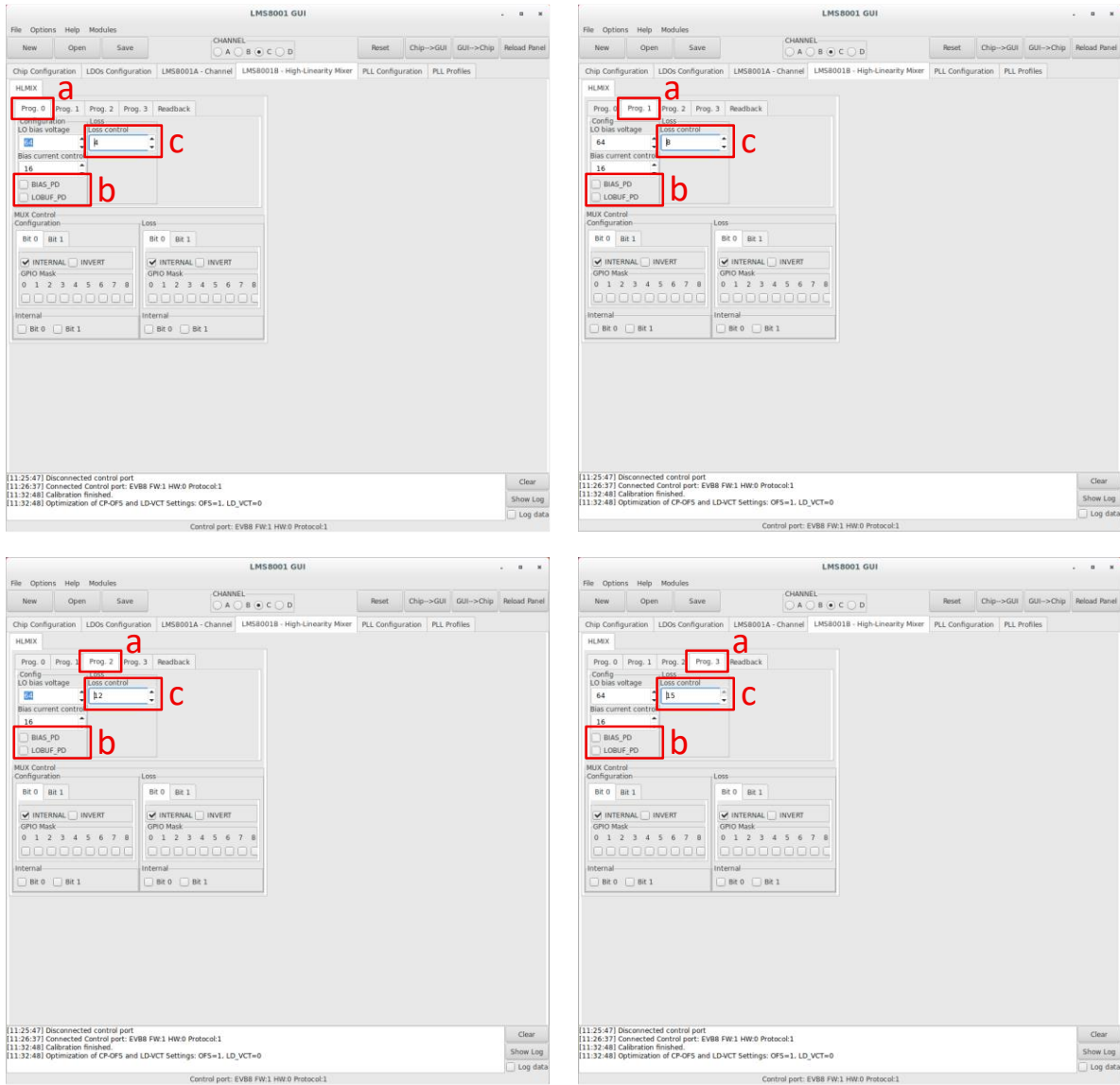


Figure 3.12: Mixer channel Programs configurations

Table 4 – RF High Linearity Mixer Channel Programs Configurations: Set 1

	BIAS_PD	LOBUF_PD	LO bias voltage	Bias current control
Program 0	Unchecked	Unchecked	64	16
Program 1	Unchecked	Unchecked	64	16
Program 2	Unchecked	Unchecked	64	16
Program 3	Unchecked	Unchecked	64	16

Table 5 – RF High Linearity Mixer Channel Programs Configurations: Set 2

	Loss control
Program 0	4
Program 1	8
Program 2	12
Program 3	15

LMS8001A RF Mixer Channel Programs Configuration

This example demonstrates how GPIO signals can be used to control the active RF High Linearity Mixer channel configuration. Four predefined configurations are provided in the GUI and are named Program 0, Program 1, Program 2, and Program 3.

Each program actually contains two independent sets of parameters, as shown in Table 4 and Table 5 and in Figure 3.13, resulting in a total of 16 possible predefined configurations. However, since the parameters in Set 1 do not vary in this example (see Table 4), only four different configurations are demonstrated here

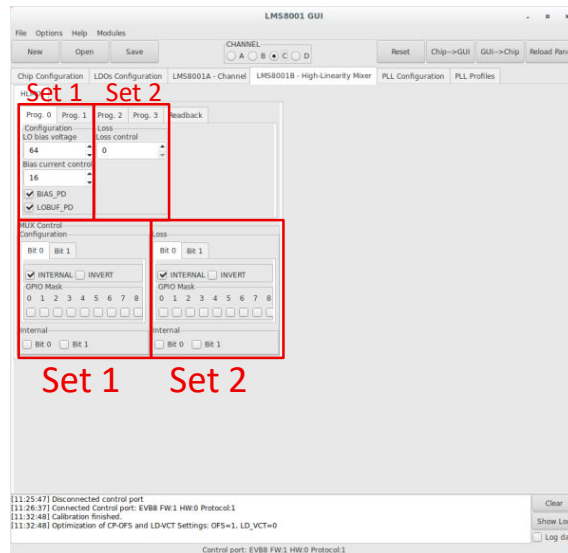
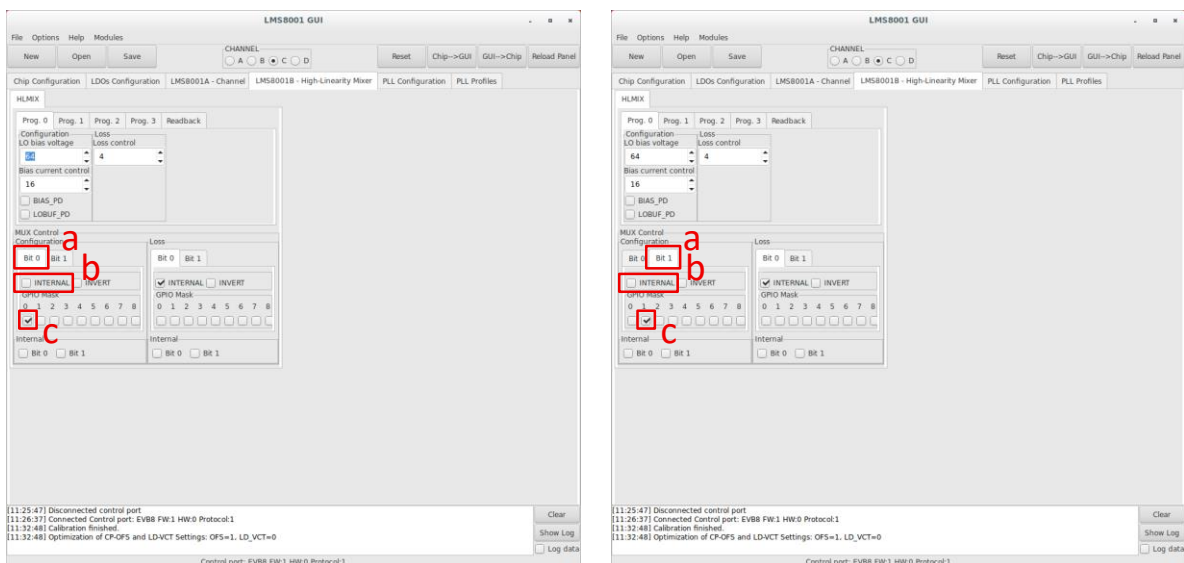


Figure 3.13: Independent sets for RF mixer channel configuration

8. Configuring the control GPIO signals
 - a. Select Bit 0 or Bit 1
 - b. Uncheck box INTERNAL
 - c. Connect control Bit 0 or Bit 1 to the desired GPIO pins by selecting the corresponding checkboxes in the GPIO Mask

The same procedure should be applied to all Bit 0 and Bit 1, in all three sets, as shown in Figure 3.14.



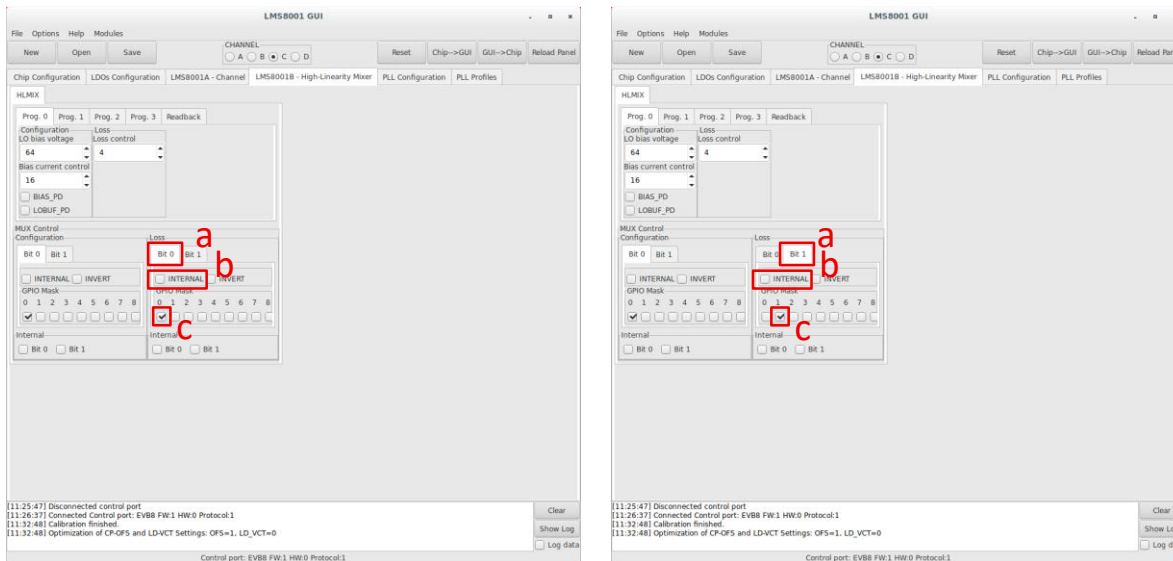


Figure 3.14: Setting control bits (Bit 0 and Bit 1 for all three SETs) and connecting them with GPIO pins

9. Finalizing the setup procedure

- a. Select the *PLL Profiles* tab
- b. Set the LO frequency
- c. Click the *Smart Tune* button
- d. Check if LO frequency is successfully generated
- e. Connect the LO signal to Channel B and verify its operation

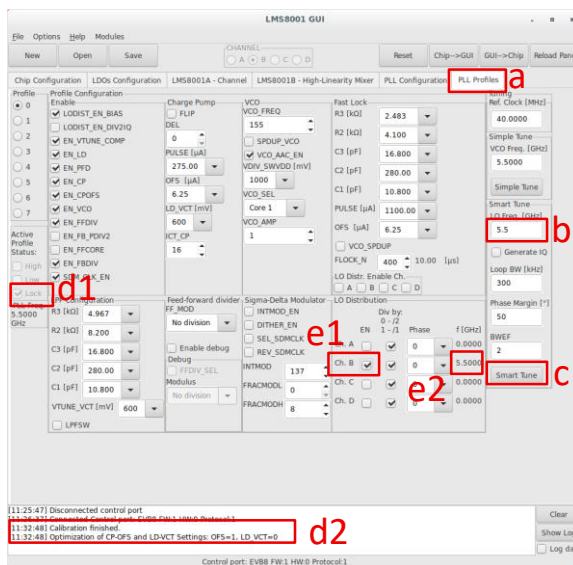


Figure 3.15: Setting LO frequency

After completing all the steps described above, the LMS8001A chip is properly configured. The signal should appear on the spectrum analyser, and its changes should be synchronized with the two external signals connected to the GPIO0 and GPIO1 pins of connector P2.

4

Testing the board

This test was designed to verify fast switching of the mixer channel configuration on the Companion Board. The mixer channel configuration is controlled by two external signals, which are generated by an Arduino microcontroller board. A detailed list of the hardware devices used in this test is provided below:

1. RF mixer board: The LMS8001 Companion Board 3v2
2. Micro-controller board: Arduino Nano RP2040 Connect board
3. Spectrum analyser: Agilent E4440A
4. Signal generator: Agilent E8267D
5. DC supply unit: Rigol DP832
6. Computer: Any PC with lms8suite and Arduino IDE installed on it

For detailed information about the connectivity between devices, please refer to the block diagram shown in Figure 4.1.

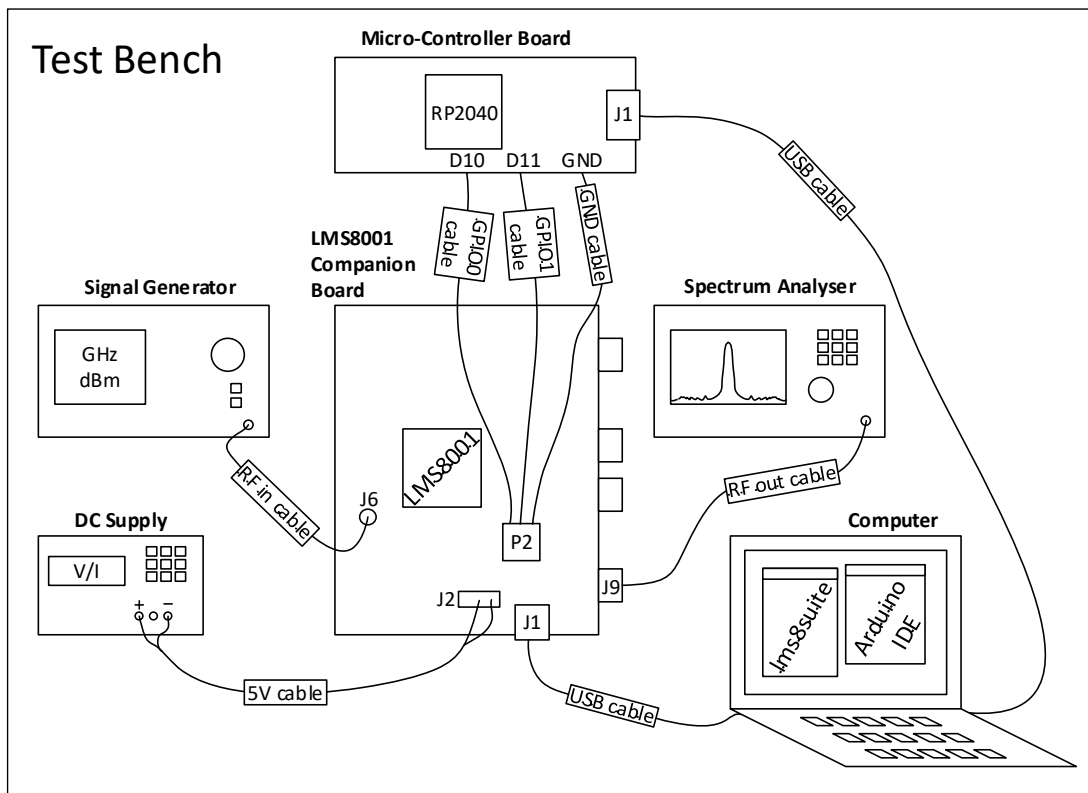


Figure 4.1: Block diagram of the test bench
(for detailed info about the Companion Board connectors please see Figure 3.1)

Connectors from the Figure 4.1 above are described in following text:

1. J1 USB connector, for the Companion Board 3v2 it is Type-C connector
2. J6: Channel B input
3. J9: Channel B output
4. P2: GPIO connector
5. J2: 5V DC power connector

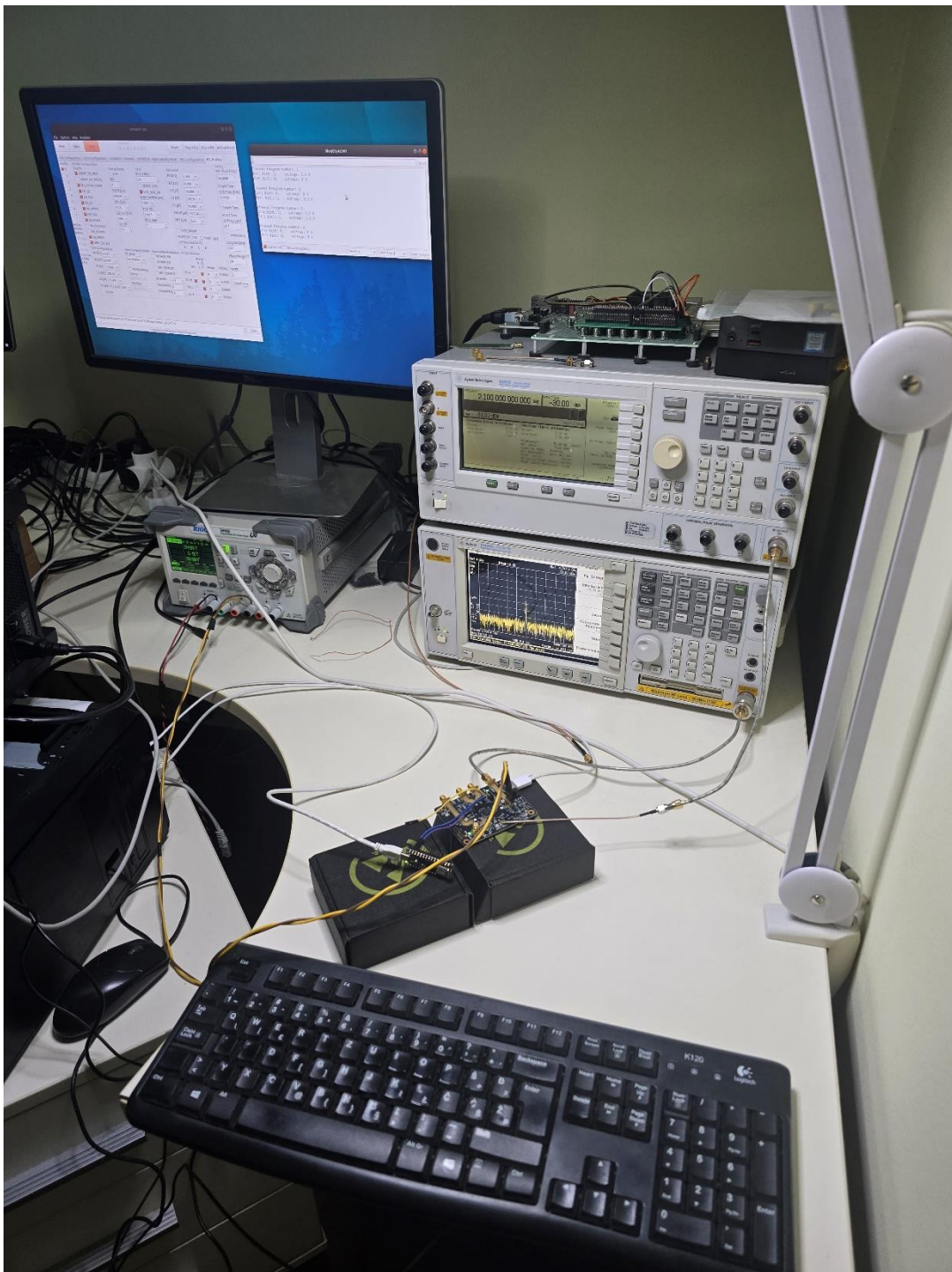


Figure 4.2: Photo of the test bench

The most relevant parameters for the test:

1. Signal generator output power: -30 dBm
2. Signal generator output frequency: 2.1 GHz
3. Spectrum analyser central frequency: 6.8 GHz
4. LO frequency, generated with Smart Tune: 4.7 GHz *
5. External signal switching speed: 3 seconds *
6. The Companion Board Channel B is used: up-conversion case

* Please keep in mind that, for the purpose of this test, some parameters differ from those described in the previous chapters:

1. The LO frequency is set to 4.7 GHz instead of 5.5 GHz, as shown in Chapter 3.
2. The switching period is increased to 3 seconds instead of 1 second, as specified in the C-code in the Appendix.

The change in mixer channel configuration can be observed in spectrum analyser. Screenshots for four different mixer channel configurations are given in Figure 4.3.

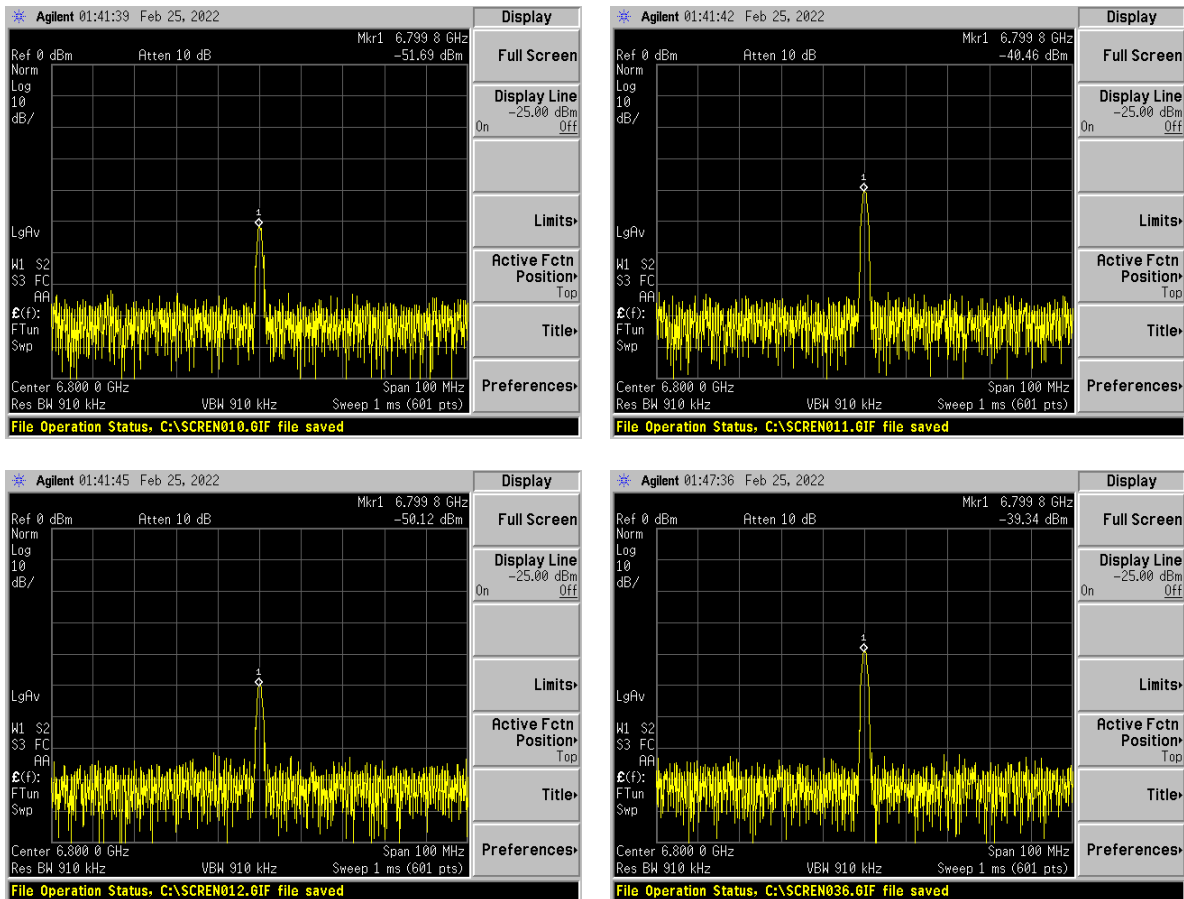


Figure 4.3: Measurement results, spectrum analyser screenshots taken for predefined Programs (predefined programs are described in Table 1, Table 2, Table 3 and Figure 3.5)

Table 6 – Measurement results for four mixer channel configurations

	Measured power
Program 0	-51.7 dBm
Program 1	-46.5 dBm
Program 2	-50.1 dBm
Program 3	-39.3 dBm

Programs 0 and 1 include mixer channel branch B, which contains only MIXB in its RF signal path, while Programs 2 and 3 include mixer channel branch A, which contains LNA and MIXA in its RF signal path, Figure 2.1. As a result, Programs 2 and 3 are expected to have higher gain than their corresponding Programs 0 and 1. Additionally, Programs 1 and 3 include the PA in their RF signal path, while Programs 0 and 2 bypass the PA. Therefore, Programs 1 and 3 are expected to achieve higher gain than their corresponding Programs 0 and 2. Keep in mind that the LNA in mixer branch A is operating far from its optimal condition; therefore, its gain is barely noticeable during this test, approximately 1 dB.

Please keep in mind that the purpose of this test was not to achieve optimized RF mixer performance in terms of gain, linearity, or similar parameters. This test demonstrates an advanced feature of the LMS8001 chip: fast channel configuration switching using external signals.

5

Appendix

5.1 Arduino IDE RP2040 C-code

```
#define lms_gpio_0 10
#define lms_gpio_1 11

int t_ms = 1000; // delay in ms
int counter = 0;
int bit0 = 0;
int bit1 = 0;
String msg0 = "0 V";
String msg1 = "0 V";

void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
  pinMode(lms_gpio_0, OUTPUT);
  pinMode(lms_gpio_1, OUTPUT);

  Serial.begin(115200);
}

// Update pins
void updateGPIOs(int b0, int b1) {
  digitalWrite(LED_BUILTIN, b0);
  digitalWrite(lms_gpio_0, b0);
  digitalWrite(lms_gpio_1, b1);
}

// Update msgs
String updateMSG(int b0) {
  String msg = "0 V";
  if(b0 == 0){msg = "0 V";}
  else {msg = "3.3 V";}

  return msg;
}
```

```
void loop() {
  delay(t_ms);

  // Control bits calculation
  bit0 = (counter >> 0) & 0x01; // LSB
  bit1 = (counter >> 1) & 0x01;

  // MSG update
  msg0 = updateMSG(bit0);
  msg1 = updateMSG(bit1);

  // GPIO pins update
  updateGPIOs(bit0, bit1);

  // Serial output
  Serial.println("");
  Serial.print("Channel Program number: ");
  Serial.println(counter);

  Serial.print("Ctrl Bit0: ");
  Serial.print(bit0);
  Serial.println("\t voltage: " + msg0);

  Serial.print("Ctrl Bit1: ");
  Serial.print(bit1);
  Serial.println("\t voltage: " + msg1);

  // counter update (0-3)
  counter++;
  if (counter > 3) {
    counter = 0;
  }
}
```