Evaluation of SDR Boards and Toolchains

Final Report

Authors: Alexandru Csete
Sheila Christiansen

Date: 2020-08-29
Version: 1.1

Revision History

Date	Version	Sections	Changes
2020-03-20	1.0	All	First version for LSF review.
2020-08-29	1.1	2.2.4 2.2.8.3	Updated with new LimeSDR Mini measurements and info.

Table of Contents

List of Figures

List of Tables

Appendices

List of Abbreviations and Symbols

ADC	Analog-to-Digital Converter
API	Application Programming Interface
BDR	Blocking Dynamic Range
CW	Carrier Wave or Continuous Wave
DDC	Digital Down Converter
DR	Dynamic Range
FFT	Fast Fourier Transform
FPGA	Field Programmable Gate Array
GPU	Graphical Processing Unit
MDS	Minimum Detectable Signal
MER	Modulation Error Ratio
NF	Noise Figure
RF	Radio Frequency
RFIC	Radio Frequency Integrated Circuit
SDR	Software Defined Radio
SNR	Signal to Noise Ratio
SFDR	Spurious Free Dynamic Range

References

[1] SDRplay RSPduo datasheet: https://www.sdrplay.com/wp-content/uploads/2018/05/RSPduoDatasheetV0.6.pdf

[2] Keysight Technologies, Noise Figure Measurement Accuracy: The Y-Factor Method, (application note, local copy in Git): https://www.keysight.com/dk/en/assets/7018-06829/application-notes/5952-3706.pdf

[3] Wikipedia article on Minimum Detectable Signal: https://en.wikipedia.org/wiki/Minimum_detectable_signal

[4] Wikipedia article on Johnson-Nyquist noise: https://en.wikipedia.org/wiki/Johnson%E2%80%93Nyquist_noise

[5] Wikipedia article on Boltzmann Constant: https://en.wikipedia.org/wiki/Boltzmann_constant

[6] Rohde & Schwarz, The Y Factor Technique for Noise Figure Measurements (application note, local copy in Git): https://www.rohde-schwarz.com/pl/applications/the-y-factor-technique-for-noise-figure-measurements-application-note_56280-15484.html

[7] Walt Kester, Taking the Mystery out of the Infamous Formula, "SNR = 6.02N + 1.76dB," and Why You Should Care, Analog Devices MT-001: https://www.analog.com/media/en/training-seminars/tutorials/MT-001.pdf

[8] Error vector magnitude on Wikipedia: https://en.wikipedia.org/wiki/Error_vector_magnitude

[9] ETSI EN 302 307-1 V1.4.1

[10] Softrig Git repository: https://gitlab.com/csete/softrig

[11] GNU Radio: https://gnuradio.org

[12] GrSoapy SDR device wrapper module for GNU Radio: https://gitlab.com/librespacefoundation/gr-soapy

[13] Git repository for this sub-activity: https://gitlab.com/librespacefoundation/sdrmakerspace/sdreval

[14] SATSAGEN http://www.albfer.com/en/author/wp_10278758/

[15] RTL-SDR wiki page at Osmocom: https://osmocom.org/projects/rtl-sdr/wiki/Rtl-sdr

[16] Airspy Mini product page: https://airspy.com/airspy-mini/

[17] SDRplay RSPduo Technical Information: https://www.sdrplay.com/wp-content/uploads/2018/06/RSPDuo-Technical-Information-R1P1.pdf

[18] LimeSDR Mini product page: https://www.crowdsupply.com/lime-micro/limesdr-mini

[19] LMS7002M data sheet v3.1: https://limemicro.com/app/uploads/2017/07/LMS7002M-Data-Sheet-v3.1r00.pdf

[20] Lime Microsystems projects on GitHub: https://github.com/myriadrf

[21] LMS7002M RF and Analog Measurement Results by Lime Microsystems: https://limemicro.com/app/uploads/2015/08/LMS7002M_Measurements-v1_05.pdf

[22] BladeRF 2.0 micro product page: https://www.nuand.com/bladerf-2-0-micro/

[23] Nuand BladeRF project on GitHub: https://github.com/Nuand/bladeRF

[24] BladeRF 2.0 micro schematics: https://www.nuand.com/bladeRF-micro.pdf

[25] Ettus USRP B210 product page: https://www.ettus.com/all-products/ub210-kit/

[26] Ettus Research on GitHub: https://github.com/EttusResearch

[27] USRP B210 schematics: https://files.ettus.com/schematics/b200/b210.pdf

[28] USRP B2xx Environmental Specifications: https://kb.ettus.com/B200/B210/B200mini/B205mini#Environmental_Specifications

[29] USRP B200 Performance Data: https://kb.ettus.com/images/c/cb/B200_RF_Performance.pdf

[30] PlutoSDR Wiki page: https://wiki.analog.com/university/tools/pluto

[31] PlutoSDR System Issues: https://wiki.analog.com/university/tools/pluto/users/name#system_issues

[32] ADALM-PLUTO Hardware: https://wiki.analog.com/university/tools/pluto/hacking/hardware

[33] SatNOGS Open Source Ground Station Network: https://satnogs.org/

[34] SatNOGS ground station 1353: https://network.satnogs.org/stations/1353/

[35] SatNOGS ground station 1354: https://network.satnogs.org/stations/1354/

1 Introduction

1.1 Purpose and Scope

During recent years, we have seen tremendous development in the availability of low cost, consumer-grade SDR hardware. Unfortunately, the low cost and the fast pace at which SDR hardware is developed does not leave much room for delivering complete turnkey solutions to the users. Hardware is often delivered as “evaluation boards” with little or no RF performance specifications, and software support is often left to third parties.

The purpose of this activity is to evaluate a selection of currently available SDR hardware and software from a satellite communications point of view, in order to provide prospective users with independent data that is often not available from the manufacturers. The evaluation includes both analytical assessments as well as lab tests supplemented by over the air usage of the systems.

The work was executed in three parts:

1. Evaluation of the technical specifications, where we tested the SDR devices for key parameters such as noise figure, dynamic range, and transmit power. This part is covered in chapter 2 of the report.
2. On-air evaluation, where two SDR devices were compared side-by-side as satellite telemetry receivers. This part is covered in chapter 3 of the report.
3. SDR software evaluation, where we took a look at how well the SDR devices are supported by the most common SDR software. This is covered in chapter 4 of the report.

1.2 Selected SDR devices

The SDR devices under test were selected according to the following criteria:

- Relevant for satellite communications, most notably for CubeSat projects
- Affordable and easily available
- Variety in architecture and expected performance

The list of devices that were tested is shown in Table 1.1 below.

Device	Type	Frequency range	Bandwidth	Bits
RTL-SDR Blog V3	RX	0.1 – 1700 MHz	3.2 MHz	8
Airspy Mini	RX	24 – 1700 MHz	6 MHz	12
SDRplay RSPduo	RX	1 kHz – 2 GHz	10 MHz	8-14
LimeSDR Mini	RX/TX	10 MHz – 3.5 GHz	30 MHz	12
BladeRF 2.0 Micro	RX/TX	70 MHz – 6 GHz	56 MHz	12
Ettus USRP B210	RX/TX	70 MHz – 6 GHz	56 MHz	12
PlutoSDR	RX/TX	70 MHz – 6 GHz	20 MHz	12

Table 1.1: Selected SDR devices under test.

Notes:

- In practice, RTL-SDR devices are limited to 2.4 MSPS continuous transfer to the host computer [15].
- SDRPlay RSPduo has 14-bits resolution up to 6.048 MSPS, 12-bits up to 8.064 MSPS, 10-bits up to 9.216 MSPS, and 8-bits above 9.216 MSPS [1].

1.3 Evaluation Criteria

Given the scope of the activity, the evaluation of the devices was limited to the frequency bands used for satellite communications:

- VHF: 145-146 MHz
- UHF: 435-438 MHz
- L-band: 1.26-1.30 GHz
- S-band: 2.0-2.1 GHz TX and 2.2-2.3 GHz RX
- S-band: 2.40-2.45 GHz
- C-band: 5.8 GHz

The evaluation criteria according to which the devices are evaluated are grouped in three groups:

1. Generic parameters (extracted from datasheets):
   - Frequency range, stability, and phase noise
   - Maximum bandwidth
   - Operating temperature range
   - Cost and availability
   - Size and weight
2. Receiver-specific parameters (tested):

   - Noise figure (NF) and minimum detectable signal (MDS)
   - Receiver dynamic range
   - Spectral purity
3. Transmitter-specific parameters (tested):
   - Maximum output power
   - Spectral purity
   - Achievable MER

1.4 Evaluation Methods

One of our objectives during this activity was to provide useful and reproducible results. To that end, we have defined our test procedures according to the following criteria:

1. Use simple methods that do not require expensive RF facilities or laboratory equipment.
2. Carry out the tests under realistic conditions that are representative for a setup under which the SDR devices are normally used.

Such methods can not provide results with the same accuracy as measurements carried out in a professional RF laboratory. However, in order to increase confidence in our results, we have cross-check the measurements using different test methods and tools. These cross-checks are described in section 2.3.

Furthermore, since we use the same methods to test all devices, the results can be used for a relative comparison between the devices, even if the measurements are not accurate on an absolute scale.

2 Evaluation of Technical Specifications

2.1 Test Setup and Procedures

2.1.1 Receiver Noise Figure

2.1.1.1 Background

The receiver sensitivity determines the smallest signal that a receiver can reliably detect [2]. Therefore, it is also called the minimum detectable signal [3], MDS, and specifies the strength of the smallest signal at the input of the receiver that causes the output signal power to be M times the output noise power.

We use M = 1 and measure the MDS as the input signal that leads to a 3 dB increase in output power.

The receiver noise figure and noise factor are measures of degradation in signal-to-noise ratio in the receiver caused by internal thermal noise in the receiver [4]. The noise figure (NF) is the noise factor (F) expressed in dB:

$$NF = 10\log_{10}(F)$$

The noise factor is closely related to the MDS of the receiver [2]:

$$MDS = k_B \times T_0 \times F \times B \times M$$

where k_B is the Boltzmann constant in units of J/K [5], T_0 is the temperature of the device, B is the detection bandwidth in Hz, and M is the power ratio used in the MDS measurement.

Using T0 = 290 K and expressing the MDS in dBm:

$$MDS = -174\,dBm + NF + 10\log_{10}(B) + 10\log_{10}(M)$$

Using M = 1, we can calculate the noise figure from the MDS:

$$NF = MDS + 174\,dBm - 10\log_{10}(B)$$

Another common way to measure the noise figure is the Y-factor method. The Y-factor method uses a calibrated noise source with well known excess noise ratio (ENR). The noise source is connected to the device under test and the noise power output density from the device is then measured with the noise source ON and OFF. The noise figure of the device is then given by [6]:

$$NF = 10\log_{10}\left(\frac{10^{\frac{ENR}{10}}}{10^{\frac{Y}{10}}-1}\right) = ENR - 10\log_{10}\left(10^{Y/10}-1\right)$$

where *ENR* is the noise figure of the noise source and *Y* is the difference between output power density from the device under test when the noise source is ON and OFF.

When the noise figure of the device becomes larger than the *ENR* of the noise source, the difference between noise source ON and OFF becomes difficult to measure. Therefore, we will be using both the MDS method and the Y-factor method to measure the noise figure. Moreover, the MDS method is simpler because it does not require a calibrated ENR source and is therefore interesting to compare it to the more accurate Y-factor method.

2.1.1.2 Test Procedure

The device under test is connected to a calibrated signal generator or *ENR* source, and the output power from the receiver is measured using the test SDR software described in section 2.1.7.

Settings:

- Sample rate and analog bandwidth are set to what is considered optimal for the device
- The DDC is tuned as to avoid DC and other spurious signals in the pass-band
- All automatic gain control in hardware and software is turned OFF
- The measurement bandwidth is set to 500 Hz

The measurement is carried out according to the following steps listed below.

MDS method:

1. With the signal generator OFF, measure the output power in 500 Hz bandwidth
2. With the signal generator ON, find the input power in dBm that gives a 3 dB increase in output power; this is our *MDS*
3. Calculate the noise figure using the formula $NF = MDS + 147$
4. Repeat steps 1-3 for different gain settings in each frequency band of interest listed in section 1.3

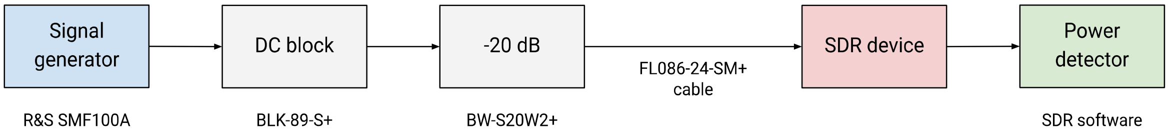


Figure 2.1: Test setup for noise figure measurements using the MDS method.

The MDS value obtained using the method described above has to be corrected for the insertion losses introduced by the DC block and the coaxial cable, as well as the deviation of the attenuator from the nominal 20 dB attenuation. The correction is performed by subtracting the insertion losses to the *MDS* obtained from the signal generator:

$$MDS_{actual} = MDS - IL_{DCBLK} - IL_{ATT} - IL_{cable}$$

The insertion losses of the individual parts in the setup are listed in Appendix I.

Y-factor method:

1. With the noise source OFF, measure the output power in 500 Hz bandwidth
2. With the noise source ON, measure the output power in 500 Hz bandwidth
3. Calculate the Y-factor using the formula $Y = P_{ON} - P_{OFF}$
4. Calculate the noise figure using the formula $NF = ENR - 10\log_{10}(10^{Y/10} - 1)$
5. Calculate the *MDS* using the formula $MDS = NF - 147$
6. Repeat steps 1-5 for different gain settings in each frequency band of interest listed in section 1.3

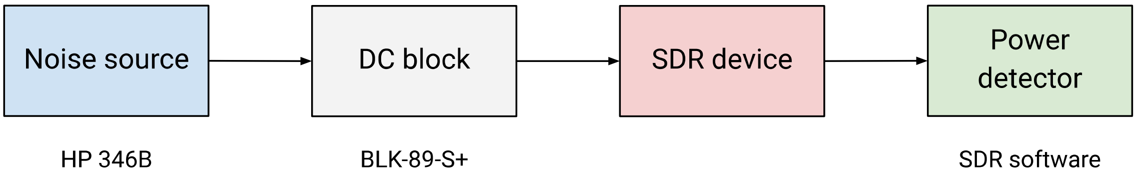


Figure 2.2: Test setup for noise figure measurements using the Y-factor method.

To avoid unnecessary losses, the SDR device is connected directly to the DC block. Therefore, only the insertion loss of the DC block has to be taken into account and this has the effect of increasing the measured noise figure.

2.1.2 Receiver Dynamic Range

2.1.2.1 Background

The dynamic range of a receiver tells us the largest signal-to-noise ratio (SNR) that we can achieve in the receiver. In other words, it expresses the difference between the strongest and the weakest signal a receiver can receive at the same time and is therefore of great importance when receiving weak signals from satellites in the presence of strong local interference.

A theoretical estimate for the dynamic range of an SDR receiver can be obtained from the bit resolution and the quantization noise of the analog to digital converter (ADC) in the receiver [7]:

$$SNR[dB] = 6.02 \times Q + 1.76$$

where Q is the bit resolution of the ADC. However, analog circuitry in front of the ADC as well as digital signal processing after the ADC can enhance or degrade the dynamic range of the complete system.

There are different metrics that can be used to characterize the dynamic range of a receiver system. We will limit our measurements to finding the blocking dynamic range, *BDR*, and the 1 dB compression point P_{1dB}.

Blocking occurs when the receiver performance is degraded by a strong input signal. This often happens because of:

1. Reciprocal mixing, i.e. when a strong signal outside of the received pass-band mixes with the phase noise of the receiver's local oscillator or another strong signal, generating extra spurs or noise in the receiver, or
2. A strong input signal overloads an analog component (amplifier, mixer, ADC) making it behave in a non-linear fashion

In both cases, the degradation can be measured either as a change in noise level, see Figure 2.3, or as a clearly noticeable "receiver breakdown" in the form of spurs in the receiver spectrum, see Figure 2.4.

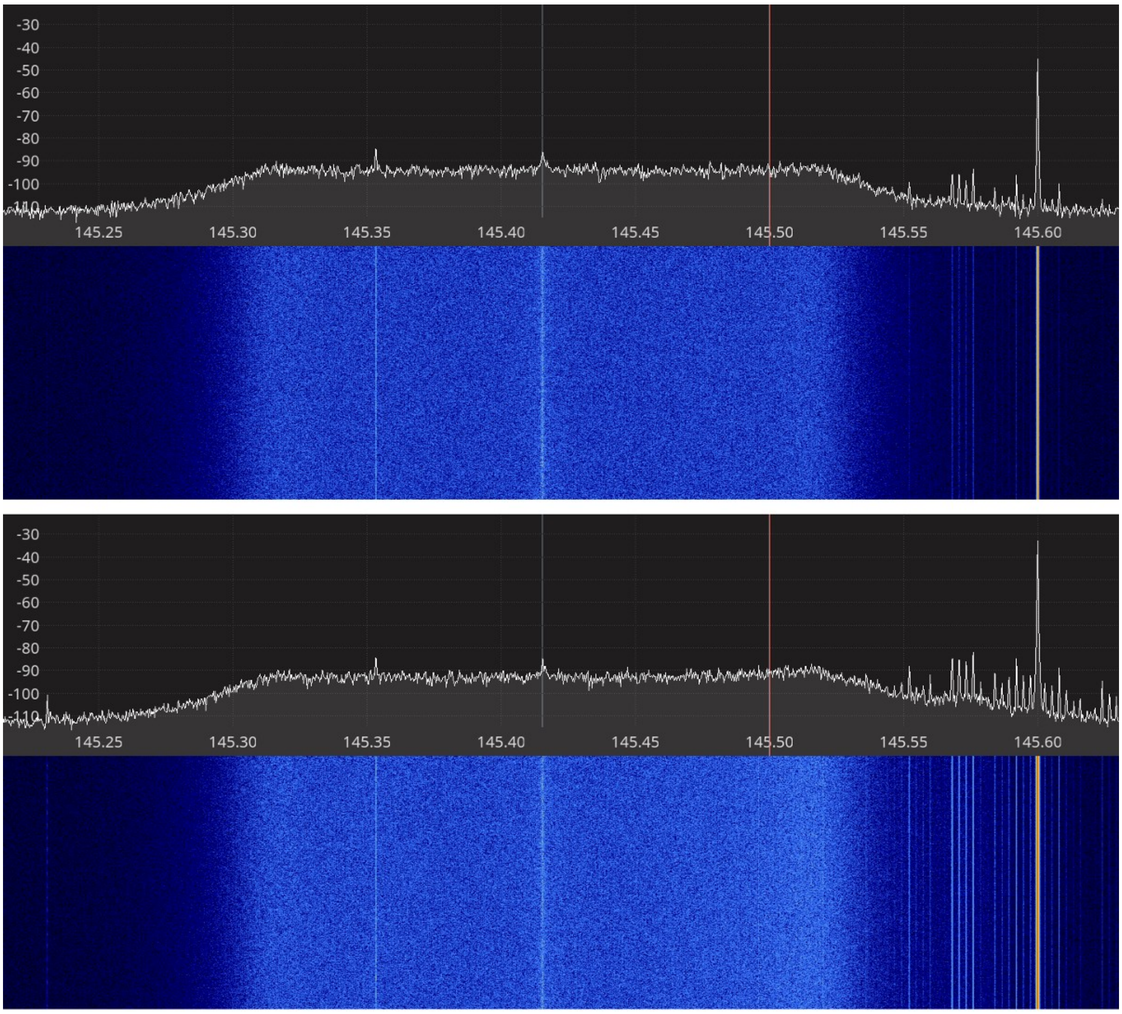


Figure 2.3: Disturbing signal causing increase in the noise level at 100 kHz away.

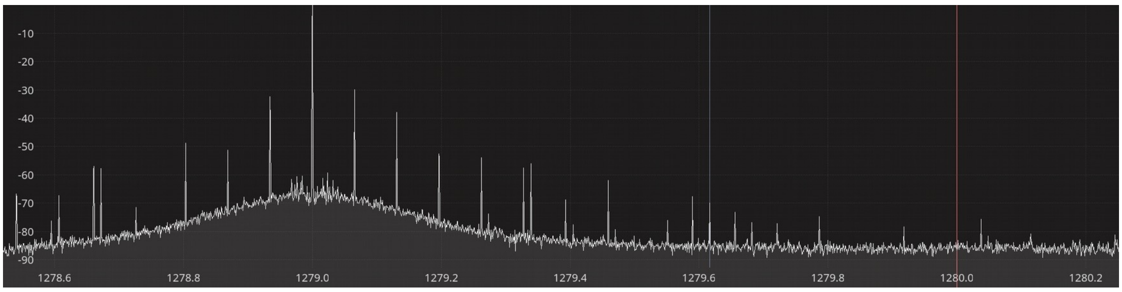


Figure 2.4: Disturbing signal causing spurs as far as 1 MHz away.

The power level just before the reciprocal mixing or overload, P_{in}, is the maximum power the receiver can handle. The blocking dynamic range is then found as the difference between the power of this input signal, P_{in}, and the minimum detectable signal, *MDS*, found during the noise figure measurements:

$$BDR[dB] = P_{in} - MDS$$

The **1 dB compression point**, P_{1dB}, is the input power level at which an increase in input power by X dB leads to an increase in output power by X-1 dB, hence the 1 dB compression. When this occurs, the dynamic range of the receiver is calculated as the difference between the P_{1dB} and the *MDS* found during noise figure measurements:

$$DR[dB] = P_{1dB} - MDS$$

2.1.2.2 Test Procedure

The dynamic range is measured in a similar setup that was used for noise figure measurements, however, we use both a power detector and spectrum plot at the same time.

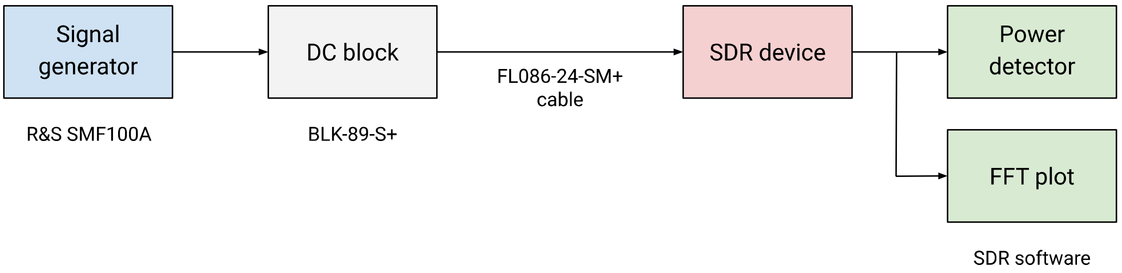


Figure 2.5: Test setup for dynamic range measurements.

The blocking dynamic range is measured as follows:

1. Tune the receiver to the desired frequency f_0 (closed to what was used during the noise figure)
2. Tune the signal generator (unwanted signal) to a frequency 5 kHz away from f_0
3. With the signal generator OFF, measure the output power in 500 Hz bandwidth
4. With the signal generator ON, find the input power P_{in} that gives a 3 dB increase in output noise power, or until spurs appear in the vicinity of f_0 and the strongest spur is 3 dB above the noise floor, c.f. figures 2.3 and 2.4
5. Calculate the dynamic range using the formula $BDR = P_{in} - MDS$
6. Repeat steps 2-5 with a signal **100 kHz** and **1 MHz** away

Since this setup also uses a DC block and coaxial cable, P_{in} has to be corrected for the insertion losses of these components:

$$P_{actual} = P_{in} - IL_{DCBLK} - IL_{cable}$$

The receiver gain compression, P_{1dB}, is measured as follows:

1. Tune the receiver and the signal generator to the desired frequency f_0 (closed to what was used during the noise figure)
2. With the signal generator OFF, measure the output power in 500 Hz bandwidth
3. With the signal generator ON, increase the input power P_{in} by 2-3 dB at a time until 1 dB compression is observed in the output power
4. Calculate the dynamic range using the formula $DR = P_{1dB} - MDS$

Note:

Receiver gain compression could not be measured on any of the devices under test because the ADC got saturated before any gain compression could be observed. Therefore, there are no P_{1dB} measurements.

2.1.3 Receiver Spectral Purity

The spectral purity of the receiver is tested through visual observation of the spectrum using an FFT in the SDR software. The tests are carried out in the frequency bands listed in section 1.3:

1. Connect a 50-ohm terminator to the receiver input
2. Adjust the receiver frequency, sample rate, and decimation to cover as much of the band of interest as possible
3. Set the receiver to maximum gain
4. Take a screenshot

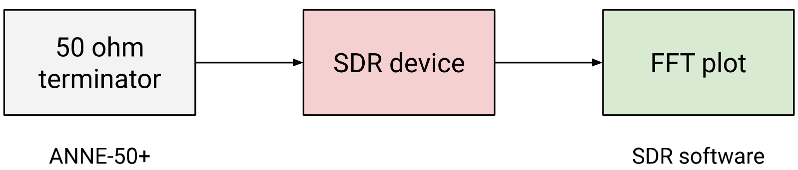


Figure 2.6: Test setup for receiver spectral purity measurements.

2.1.4 Transmitter Spectral Purity

The transmitter spectral purity is tested by transmitting a CW test signal in the frequency bands of interest listed in section 1.3. The output is checked using a spectrum analyzer.

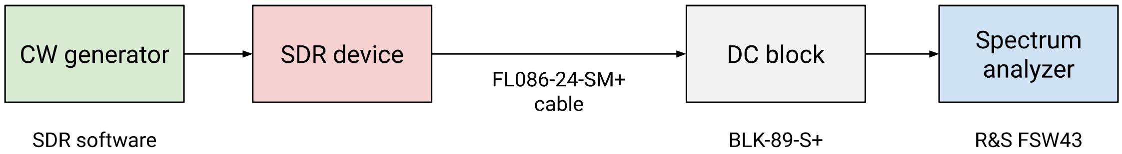


Figure 2.7: Test setup for transmitter power and spectral purity measurements.

2.1.5 Transmitter Output Power

The transmitter output power is measured by transmitting a CW test signal at various gain settings and measuring the output power using a spectrum analyzer. The setup used for this is the same as the setup used for transmitter spectral purity measurements shown in Figure 2.9. Since this setup also uses a DC block and a coaxial cable, the measured CW power is corrected for the insertion losses from these components:

$$P_{actual} = P_{FSW} + IL_{DCBLK} + IL_{cable}$$

where P_{FSW} is the power measured by the spectrum analyzer.

2.1.6 Transmitter Modulation Error Ratio

The transmitter modulation error ratio (MER) is a measure used to quantify how much the transmitted symbols deviate from their ideal value as illustrated on the constellation diagram in Figure 2.8 and given by the formula:

$$MER(dB) = 10 \times \log_{10}\left(\frac{P_{signal}}{P_{error}}\right)$$

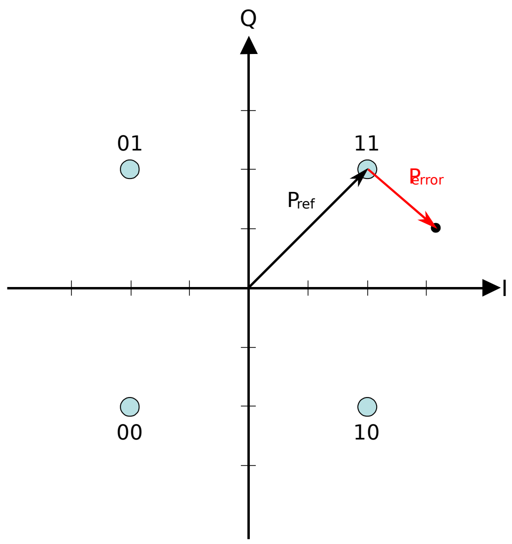


Figure 2.8: Constellation with error [8].

Transmitter MER is measured by transmitting a DVB-S2 signal and measuring the SNR and MER using a DVB-S/S2 signal analyzer and using the following settings:

- 6.25 Msymbols / sec
- Modulation: 8-PSK
- FEC rate: 9/10
- FEC frame: Normal
- Roll-off factor: 0.20
- Pilots: ON

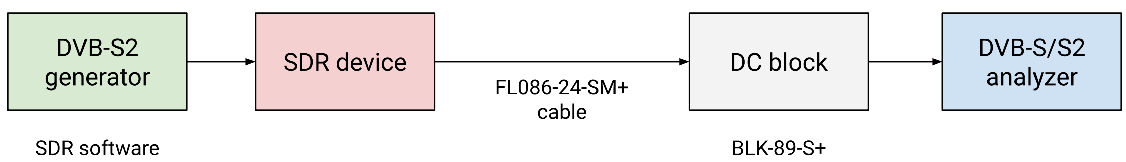


Figure 2.9: Test setup for transmitter MER measurements.

The analyzer used in our setup was a ROVER HD Flash DVB-S/S2 signal analyzer. It measures the signal power, MER, and noise margin in dB, the latter being presumably a margin above the E_S/N_0 that is required for error free reception, which is 11 dB for the modulation and FEC-rate used in our setup, c.f. Table 13 in ETSI-EN-302-307-1 [9].

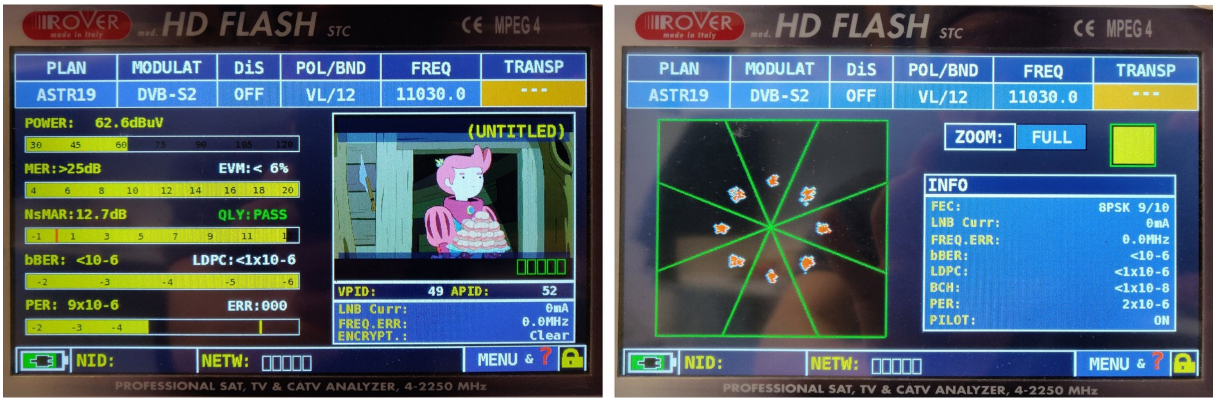


Figure 2.10: Screen captures of the ROVER DVB-S/S2 analyzer.

2.1.7 Test Software

For the receiver tests (noise figure, dynamic range, spectral purity), a dedicated Qt-based application [10] with the following components has been devloped:

1. Device interface to configure, control and read samples from the SDR device under test.
2. Spectrum plot using FFT and Qt graphics.
3. Digital down-converter to select a channel with adjustable bandwidth.
4. RMS power detector with adjustable averaging time.
5. Basic AM, FM, CW, and SSB demodulators.

A functional diagram for the test software used for receiver tests is shown on Figure 2.11 below. In order to ensure sufficient support for the SDR devices, the test software does not use any SDR device wrappers but interfaces directly to the SDR device driver libraries.

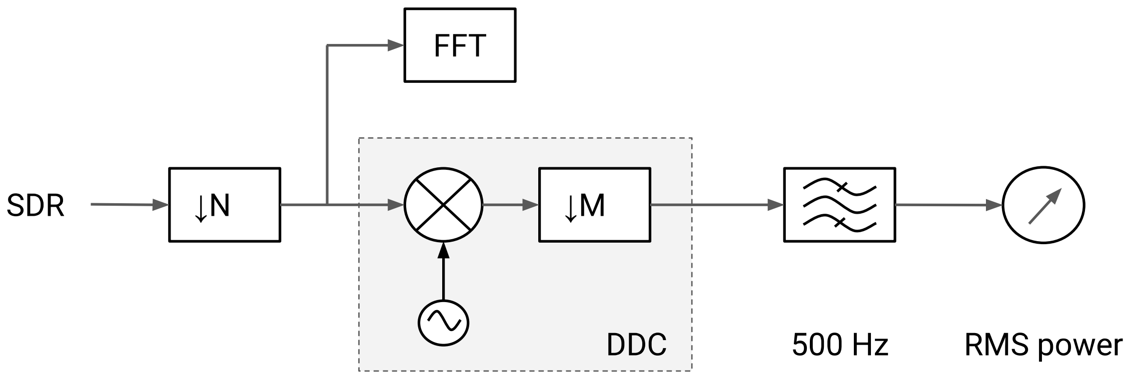


Figure 2.11: Functional diagram for the test software used for receiver tests.

For transmitter tests (output power, MER, spectral purity), simple applications based on GNU Radio 3.8 [11] and the gr-soapy [12] device interface module were created. These applications are available in the project repository [13] and two examples using the BladeRF 2.0 micro are shown on Figure 2.12 and Figure 2.13 below.

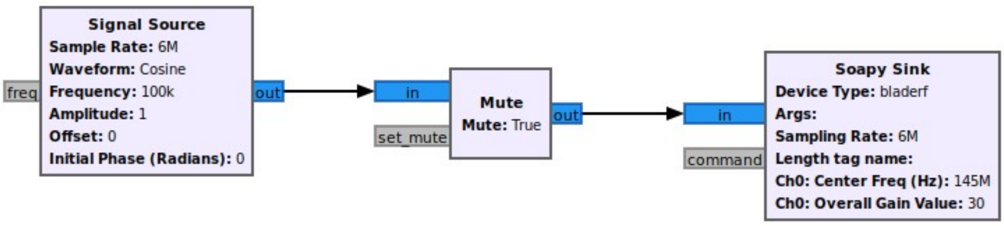


Figure 2.12: GNU Radio application used for output power measurements.

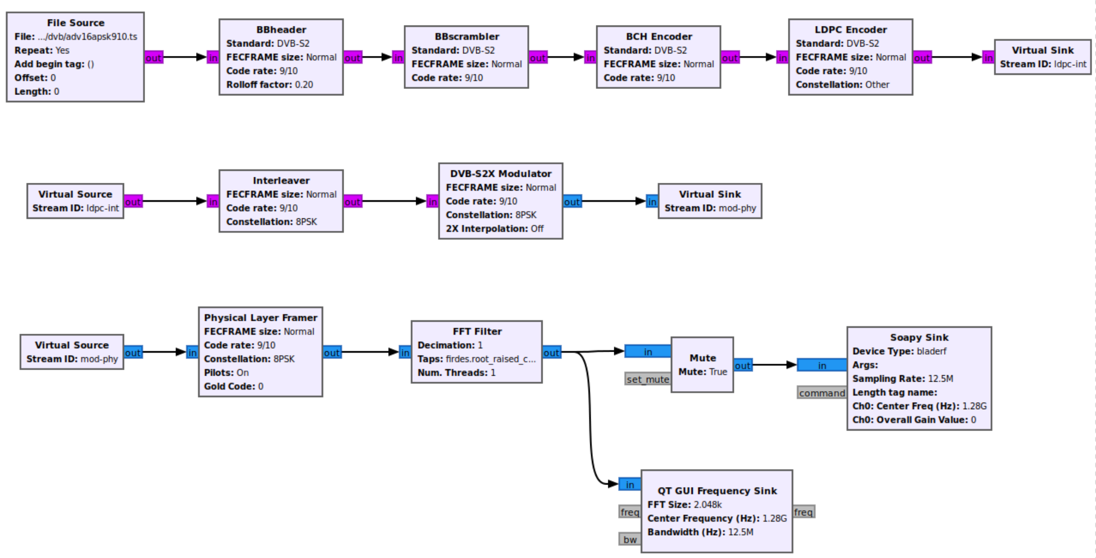


Figure 2.13: GNU Radio application used for MER measurements.

Unfortunately, our GNU Radio transmitter scripts using the gr-soapy SDR API did not work with the PlutoSDR. The CW output couldn't generate any CW-like signal and te DVB-S2 output was oscillating in amplitude. In order to perform at least some CW power measurement, we decided to use a new tool called SATSAGEN [14], which is specifically written for the PlutoSDR.

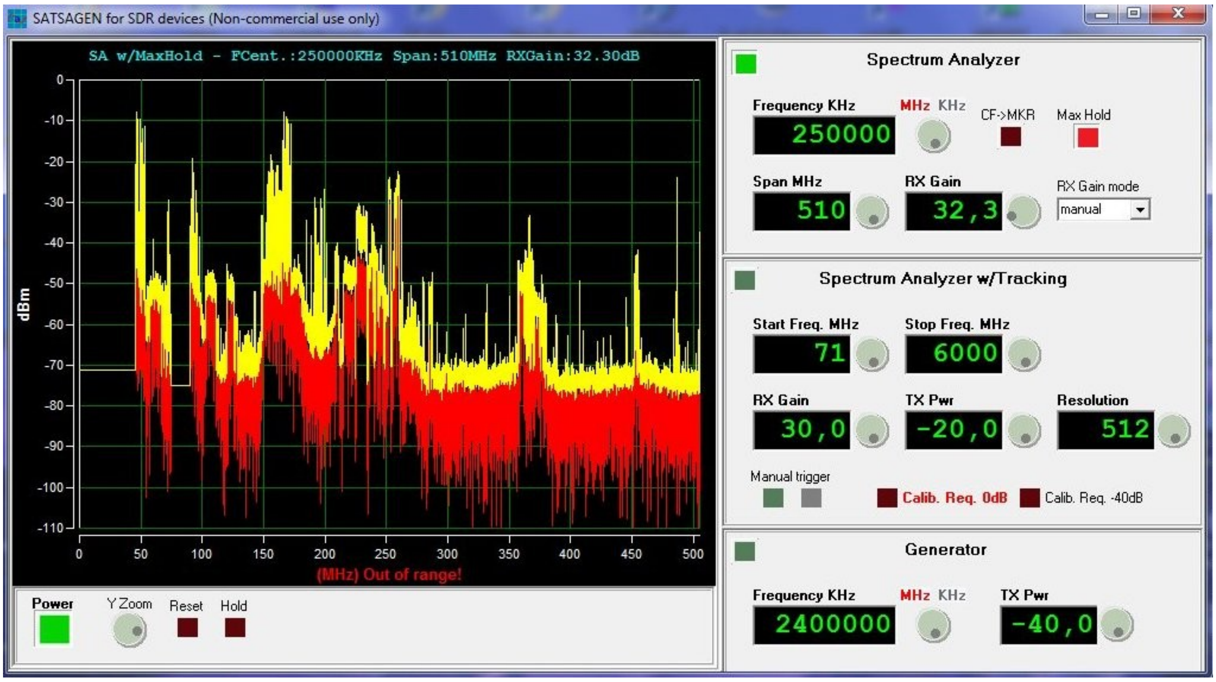


Figure 2.14: The SATSAGEN spectrum analyzer software for PlutoSDR.

2.2 Test Results

The following sections present the test results from our lab measurements. For each device tested, we begin with a brief description and the technical specifications of the device under test, followed by the receiver and transmitter performances measured during our tests. Finally, in section 2.2.8 we provide some comparisons between devices.

The data that was used to generate the graphs presented in this report is available in plain text format in the project GIT repository.

2.2.1 RTL-SDR Blog V3

RTL-SDR dongles are based on the Realtek RTL2832U DVB-T demodulator chip that was found to be capable of transferring raw I/Q samples to the host computer. This characteristic was discovered already in 2012 and has made the RTL2832U-based dongles the most popular SDR receivers to date. Compared to generic RTL-SDR devices, the RTL-SDR Blog V3 dongle has been redesigned with SDR users in mind providing significantly better performance than the generic RTL-SDR dongles.

Figure 2.15: The RTL-SDR blog V3 receiver.

The RTL-SDR Blog V3 dongle uses a Rafael R820T2 24-1700 MHz RF tuner, which delivers analog I/Q samples to the RTL2832U demodulator chip. The RTL2832U chip acts as 8-bit analog to digital converter and USB interface to the host computer and can also function as a direct sampling shortwave receiver, see figure 2.16 below.

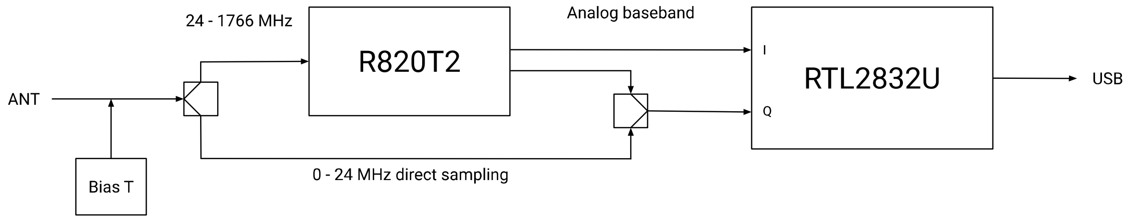


Figure 2.16: RTL-SDR Blog V3 dongle block diagram.

Software support for RTL-SDR devices is available through librtlsdr, a light-weight, open-source user space library written in C.

A summary of the technical specifications for the RTL-SDR Blog V3 dongle is listed in Table 2.1 below.

Frequency Range	500 kHz – 1766 MHz
Sample rate	2.4 MSPS
RF bandwidth	350 kHz – 8 MHz
Frontend filters	Tracking RF filters in the R820T2 tuner
RX paths	1
RX inputs	1
ADC resolution	8-bit
Claimed noise figure	-
Clained dynamic range	-
Reference clock	1 PPM TCXO
Other features	Bias T Direct sampling Expansion ports
Temperature range	-
Size	10 cm × 2.5 cm × 1.4 cm
Weight	30 g
Approximate price	22 USD
Product page	https://www.rtl-sdr.com/buy-rtl-sdr-dvb-t-dongles/

Table 2.1: RTL-SDR Blog V3 technical specifications.

2.2.1.1 Receiver Noise Figure

The settings used during the RTL-SDR noise figure measurements are listed in Table 2.2 and the results are shown on Figure 2.17 below.

Input rate	Decimation	Sample rate
2.4 MHz	8	300 kHz

Table 2.2: RTL-SDR V3 settings used during noise figure measurements.

We note that:

- There is very good consistency between the noise figures calculated from the MDS and the noise figures measured using the Y-factor method.
- The noise figure at high gains is significantly worse on 1280 MHz than on 145 and 437 MHz. This is not unexpected as insertion losses in the signal path usually increase with frequency. Furthermore, the datasheet of the R820T2 tuner only specifies the performance parameters up to 1002 MHz and anything above this frequency can be considered out of spec.

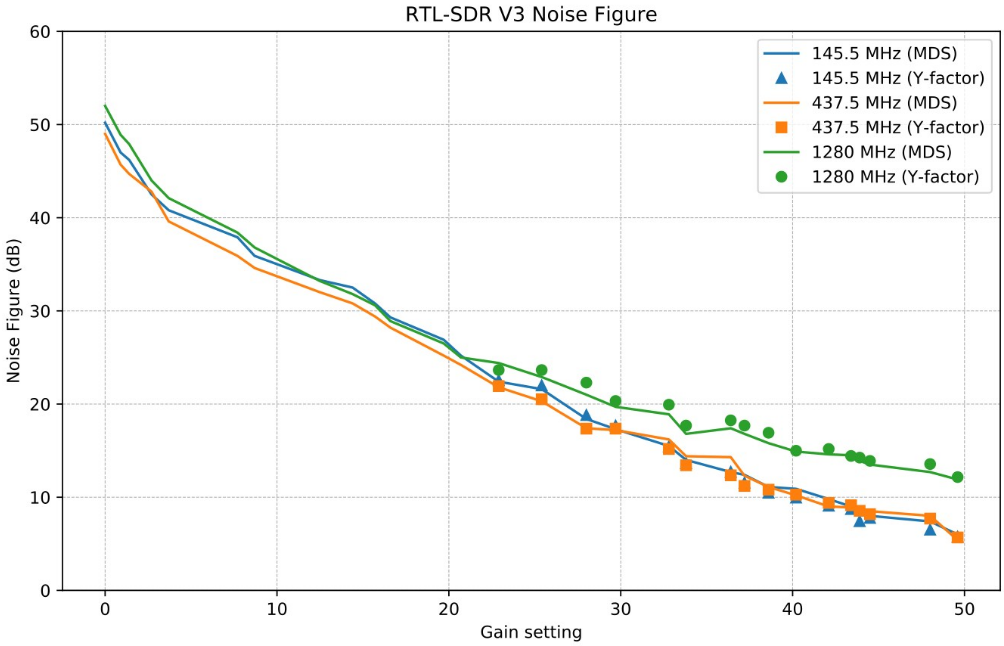


Figure 2.17: RTL-SDR V3 noise figure measurements.

2.2.1.2 Receiver Blocking Dynamic Range

The settings used during the RTL-SDR dynamic range measurements are listed in Table 2.3 and the results are shown on Figure 2.18 below.

Input rate	Decimation	Sample rate
2.4 MHz	1	2.4 MHz

Table 2.3: RTL-SDR V3 settings used during BDR measurements.

We note that:

- Ax expected, the highest input power that the device can tolerate without overload, P_{in}, is decreasing as the gain is increased. The relationship is mostly linear except on 1280 MHz.
- The dynamic range is mostly constant over the entire gain spectrum, slightly dropping at the highest gains.
- The measured dynamic range is significantly higher than what we would expect from a device with an 8-bit ADC (48 dB). The only way we can explain this is high oversampling by the ADC and that non-linearity artifact are below the quantization noise of the ADC.

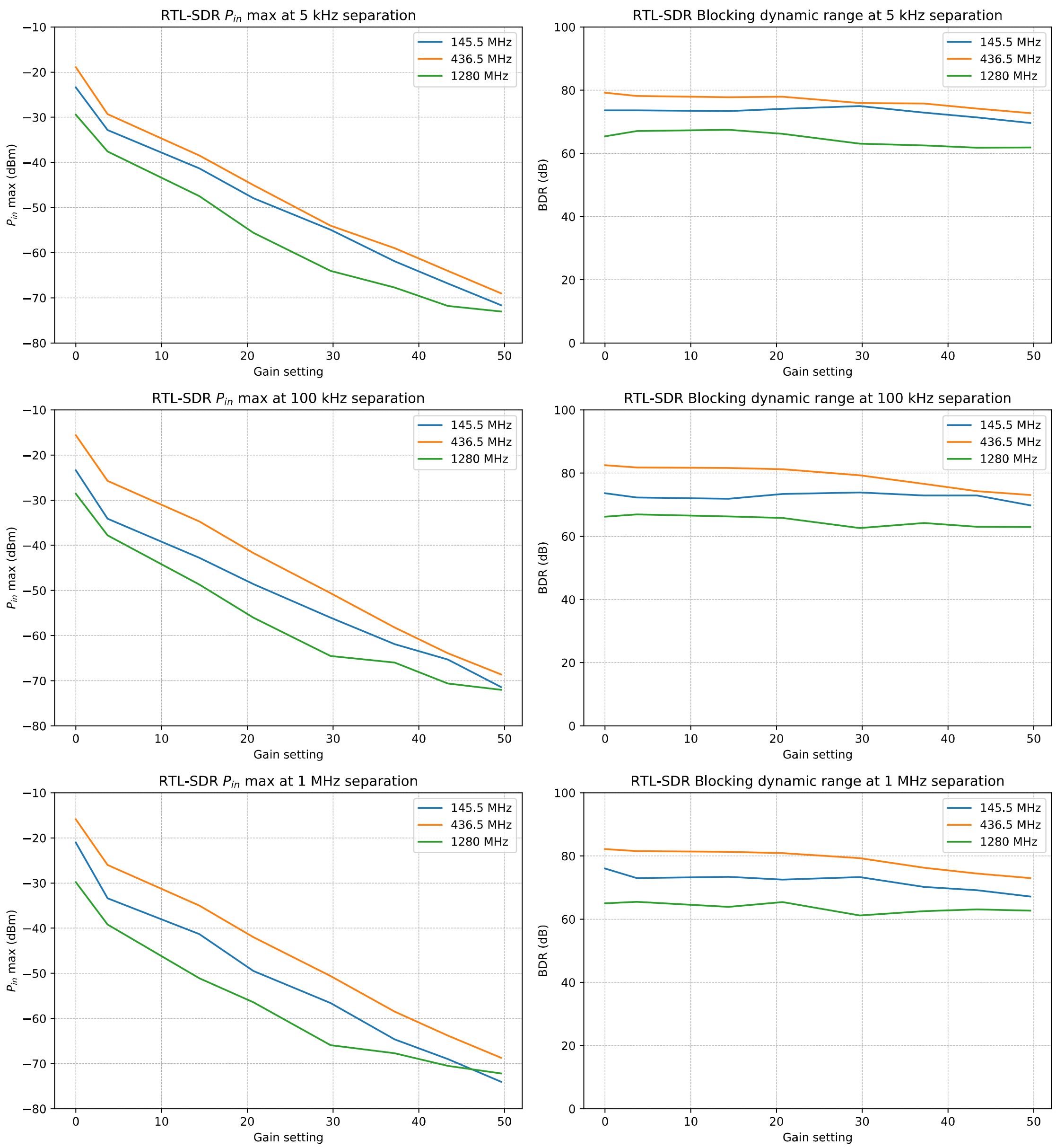


Figure 2.18: RTL-SDR V3 blocking dynamic range measurements.

In order to verify that the measured dynamic range is realistic, we performed a test where the signal from an antenna and the signal from the signal generator were combined using a 3 dB power splitter. The receiver was tuned to a weak beacon transmitting constant power and the signal from the signal generator was tuned to 5 kHz away from the beacon and the power was increased to the measured P_{in} + 3 dB. We could then verify that the signal from the signal generator did not decrease the signal to noise ratio of the beacon.

2.2.1.3 Receiver Spectral Purity

The receiver spectra of the RTL-SDR V3 receiver are shown on Figures 2.19 to 2.26 below. As we can see from the spectra, there are very few spurs generated by or picked up by the device when connected to 50 ohm terminator.

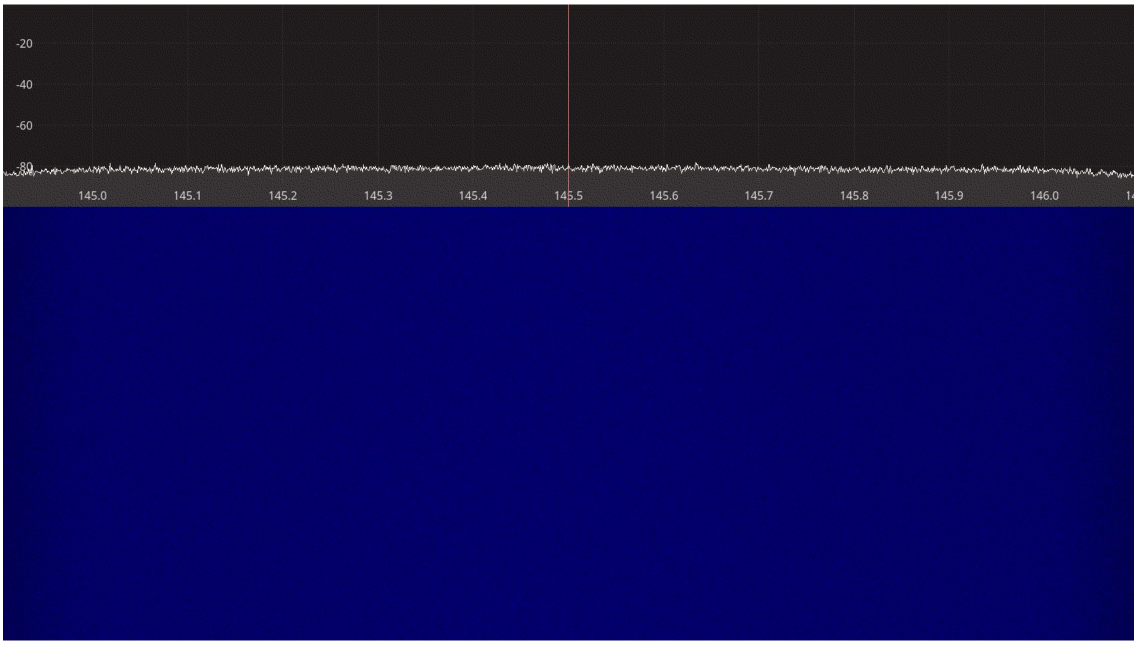


Figure 2.19: RTL-SDR receiver spectrum between 145 – 146 MHz.

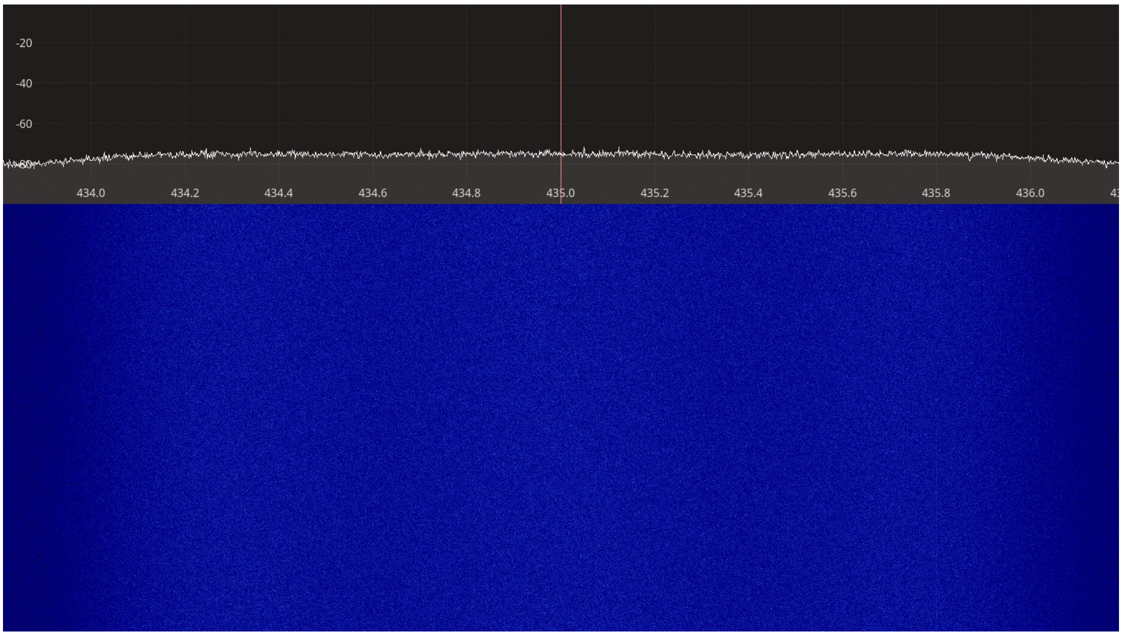


Figure 2.20: RTL-SDR receiver spectrum between 434 – 436 MHz.

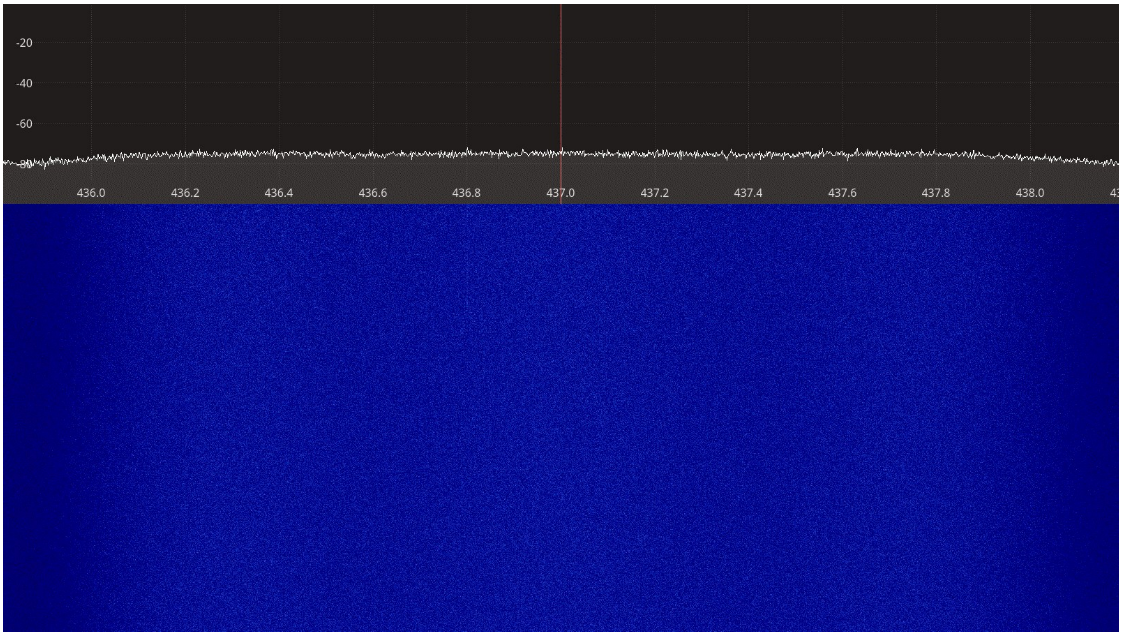


Figure 2.21: RTL-SDR receiver spectrum between 436 – 438 MHz.

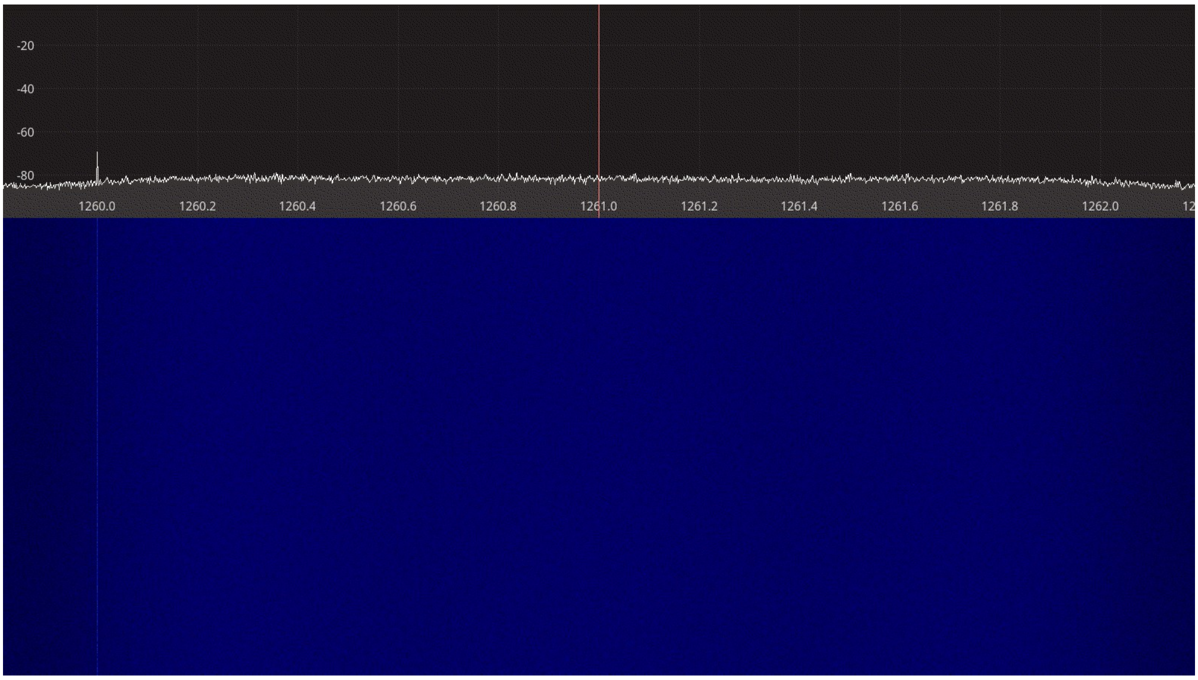


Figure 2.22: RTL-SDR receiver spectrum between 1260 – 1262 MHz.

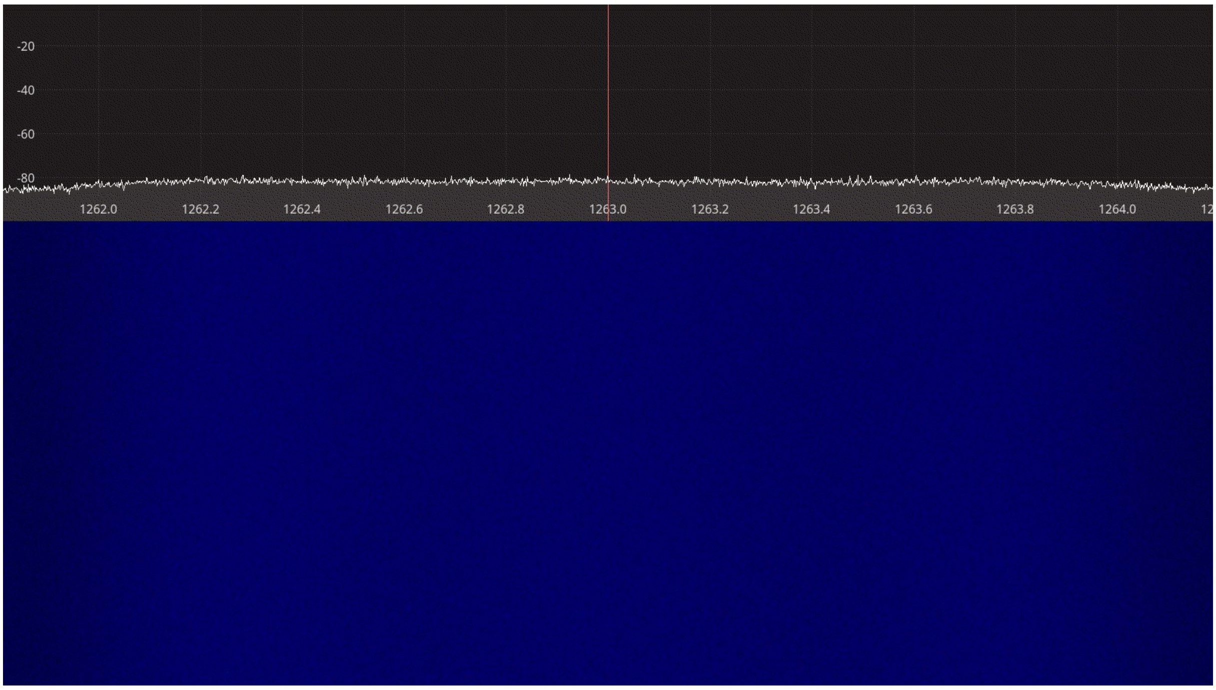


Figure 2.23: RTL-SDR receiver spectrum between 1262 – 1264 MHz.

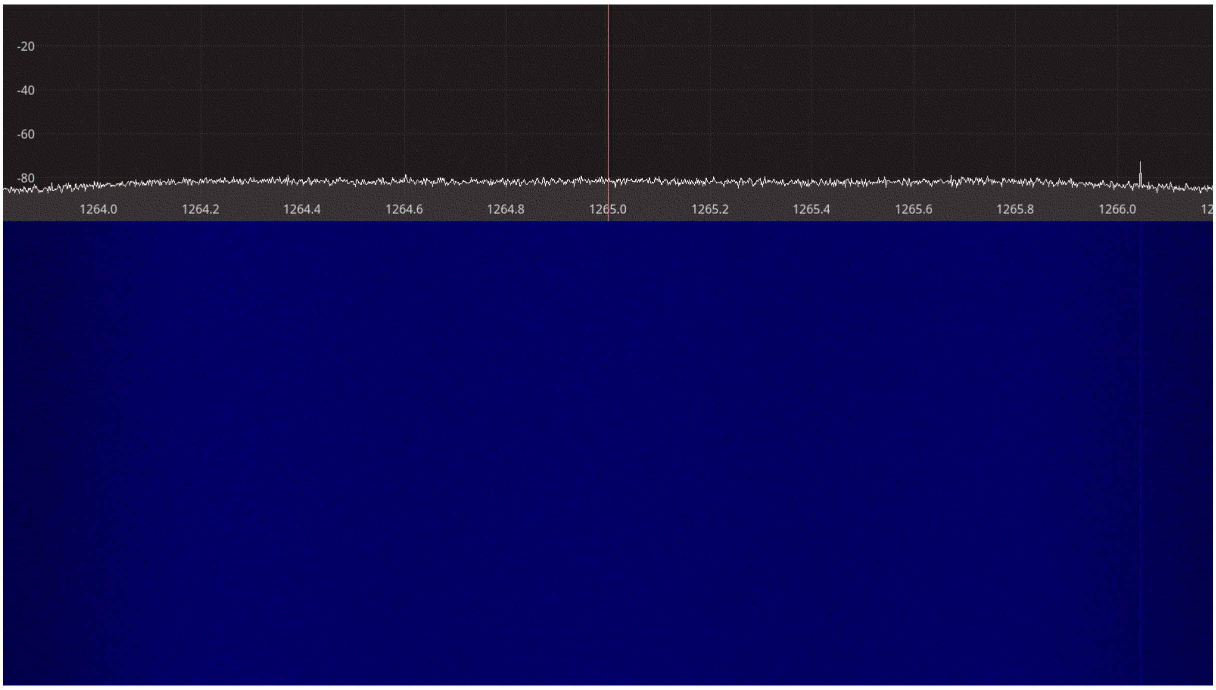


Figure 2.24: RTL-SDR receiver spectrum between 1264 – 1265 MHz.

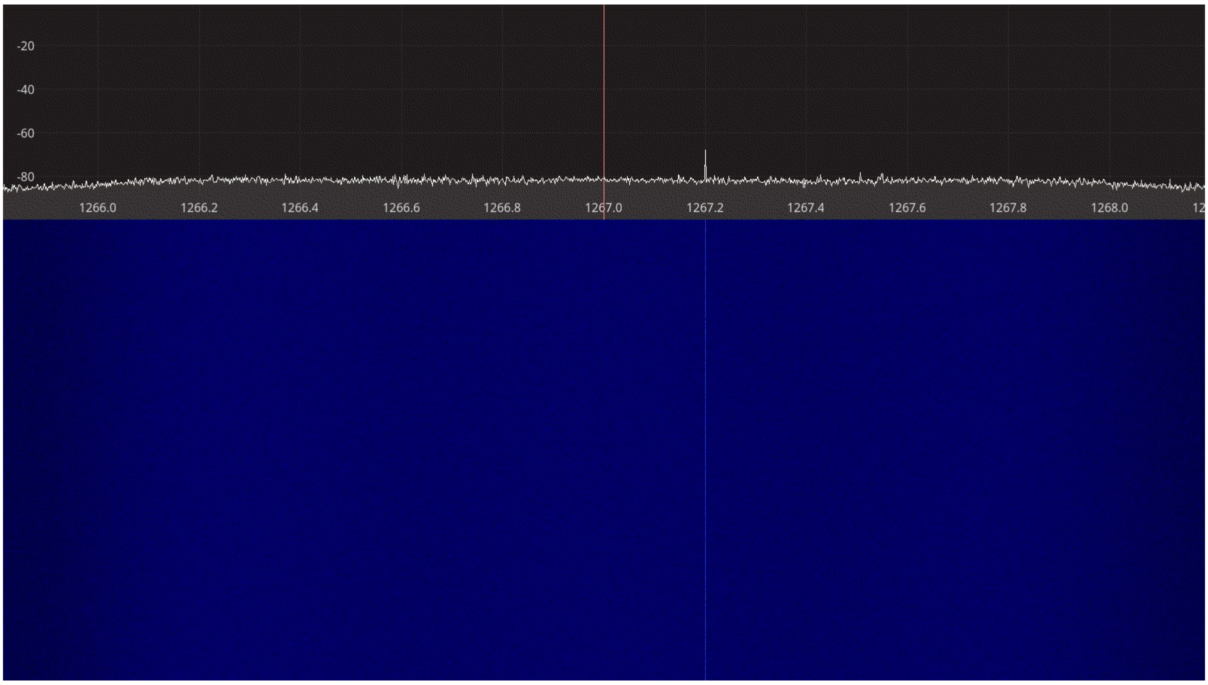


Figure 2.25: RTL-SDR receiver spectrum between 1266 – 1268 MHz.

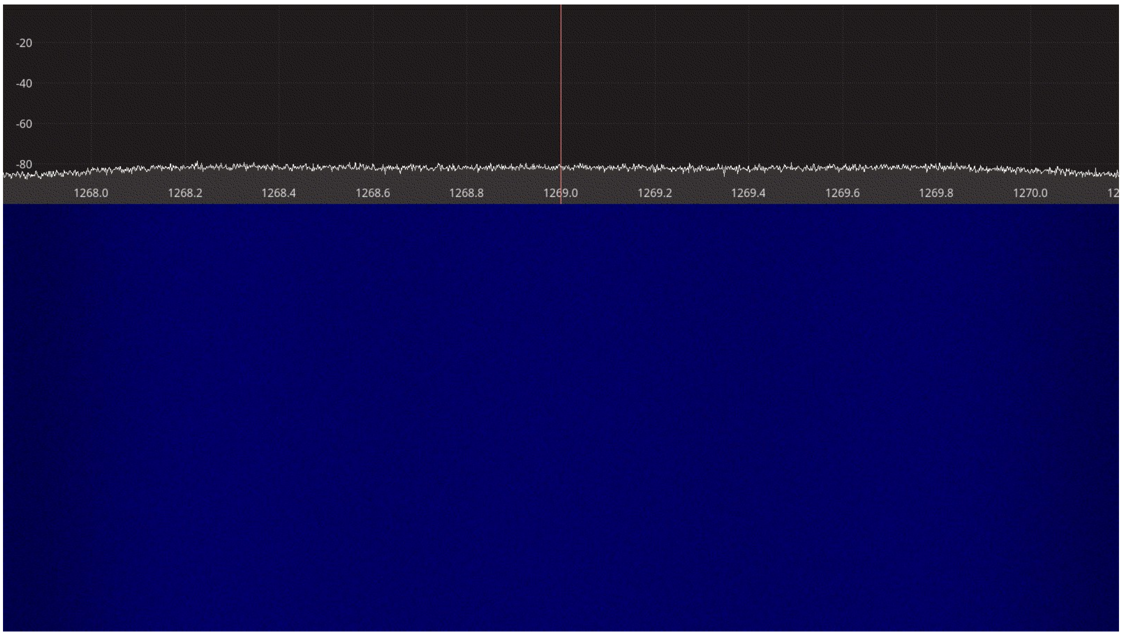


Figure 2.26: RTL-SDR receiver spectrum between 1268 – 1270 MHz.

2.2.2 Airspy Mini

The Airspy Mini has similar architecture as the RTL-SDR, except that the RTL2832U DVB-T demodulator chip is replaced with an LPC4370 micro-controller with integrated 12-bit, 20 MSPS analog to digital converter, providing both higher sample rates and potentially better dynamic range.

Figure 2.27: The Airspy Mini receiver.

The Airpsy Mini uses IF sampling to capure the bandwidth of interest. Conversion to quadrature samples is done by the driver running on the host PC. Thus, the Airspy Mini does not suffer from the I/Q imbalance, DC offset, and 1/F noise often seen in SDR receivers that use direct conversion [16].

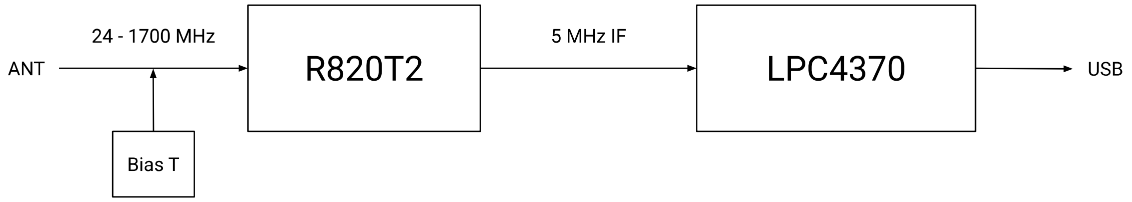


Figure 2.28: Airspy Mini block diagram.

Software support for Airspy devices is available through libairspy, a light-weight, open-source user space library written in C.

A summary of the technical specifications for the Airspy Mini is listed in Table 2.4 below.

Frequency Range	24 – 1700 MHz
Sample rate	3, 6, 10 MSPS
RF bandwidth	2.4 – 6 MHz
Frontend filters	Tracking RF filters
RX paths	1
RX inputs	1
ADC resolution	12-bit (10.4 ENOB)
Claimed noise figure	3.5 dB between 42 – 1002 MHz
Claimed dynamic range	70 dB SNR, 95 dB SFDR for the ADC
Reference clock	0.5 PPM
Other features	Bias T
Temperature range	-10 to 40 °C
Size	8 cm × 2.7 cm × 1 cm
Weight	20 g
Approximate price	99 USD
Product page	https://airspy.com/airspy-mini/

Table 2.4: Airspy Mini technical specifications.

2.2.2.1 Receiver Noise Figure

The settings used during the Airspy Mini noise figure measurements are listed in Table 2.5 and the results are shown on Figure 2.29 below.

Input rate	Decimation	Sample rate	Gain mode
6 MHz	16	375 kHz	Linearity

Table 2.5: Airspy Mini settings used during noise figure measurements.

We note that:

- There is very good consistency between the noise figure calculated from the MDS measurements and the noise figure measured using the Y-factor method.
- Like with the RTL-SDR, the noise figure on 1280 MHz is worse that the noise figure on 145 and 437 MHz.
- We could not reproduce the claimed noise figure of 3.5 dB.

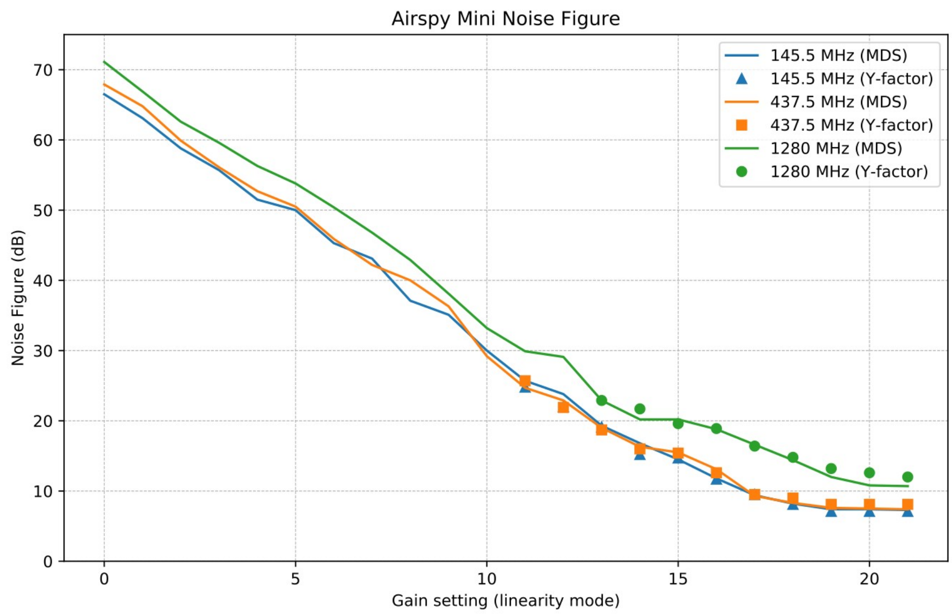


Figure 2.29: Airspy Mini noise figure measurements.

2.2.2.2 Receiver Blocking Dynamic Range

The settings used during the Airspy Mini dynamic range measurements are listed in Table 2.6 and the results are shown on Figure 2.30 below.

Input rate	Decimation	Sample rate	Gain mode
6 MHz	2	3 MHz	Linearity

Table 2.6: Airspy Mini settings used during BDR measurements.

We note that:

- Ax expected, the highest input power that the device can tolerate without overload, P_{in}, is decreasing as the gain is increased. The relationship is mostly linear except on 1280 MHz.
- The dynamic range is mostly constant over the entire gain spectrum, slightly dropping at the highest gains.
- The measured dynamic range is in the neighborhood of the claimed dynamic range of 70 dB SNR and 95 dB SFDR for the ADC.

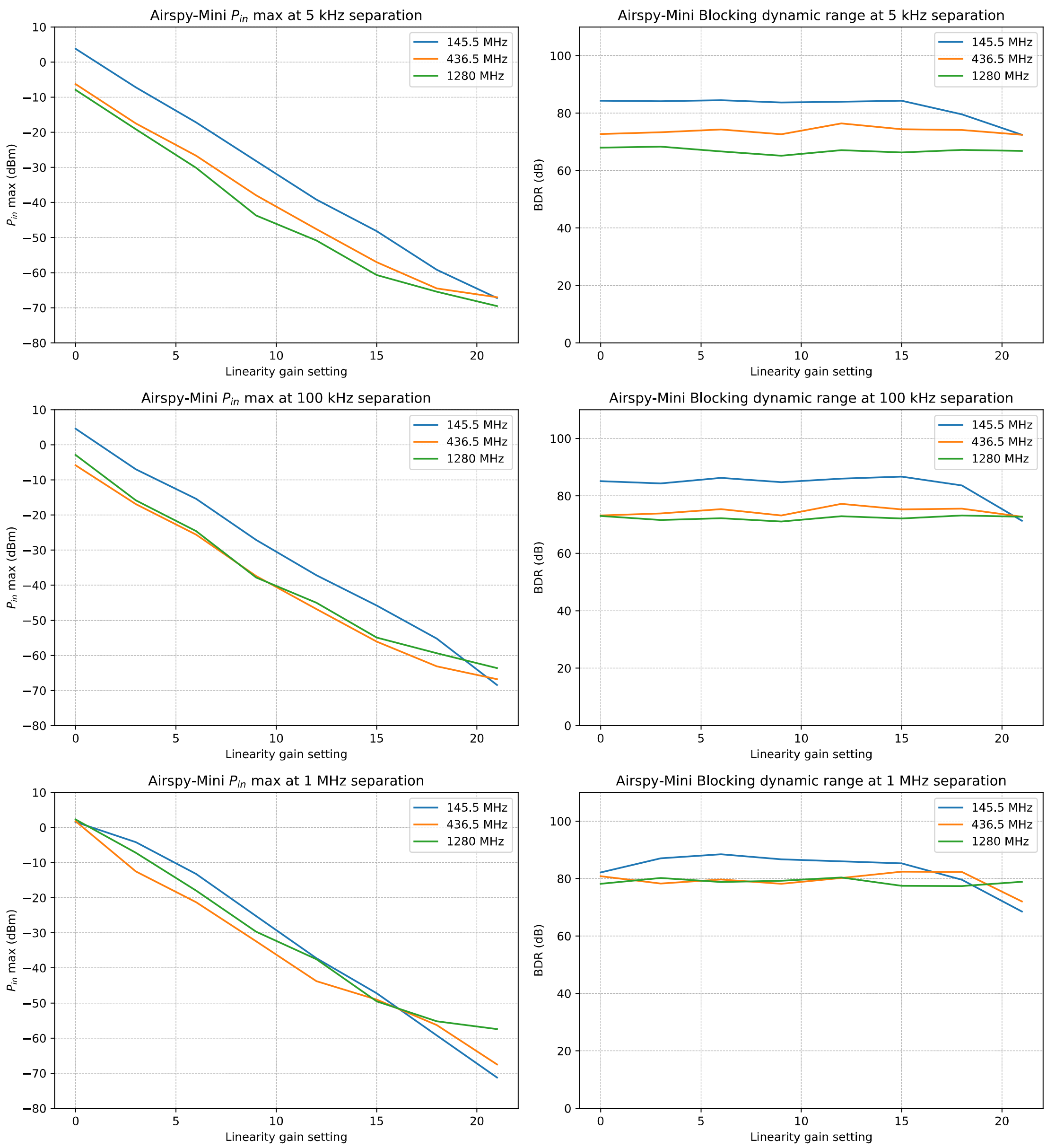


Figure 2.30: Airspy Mini blocking dynamic range measurements.

2.2.2.3 Receiver Spectral Purity

The receiver spectra of the Airspy Mini receiver are shown on Figures 2.31 to 2.35 below. As we can see from the spectra, there are very few spurs generated by or picked up by the device when connected to 50 ohm terminator.

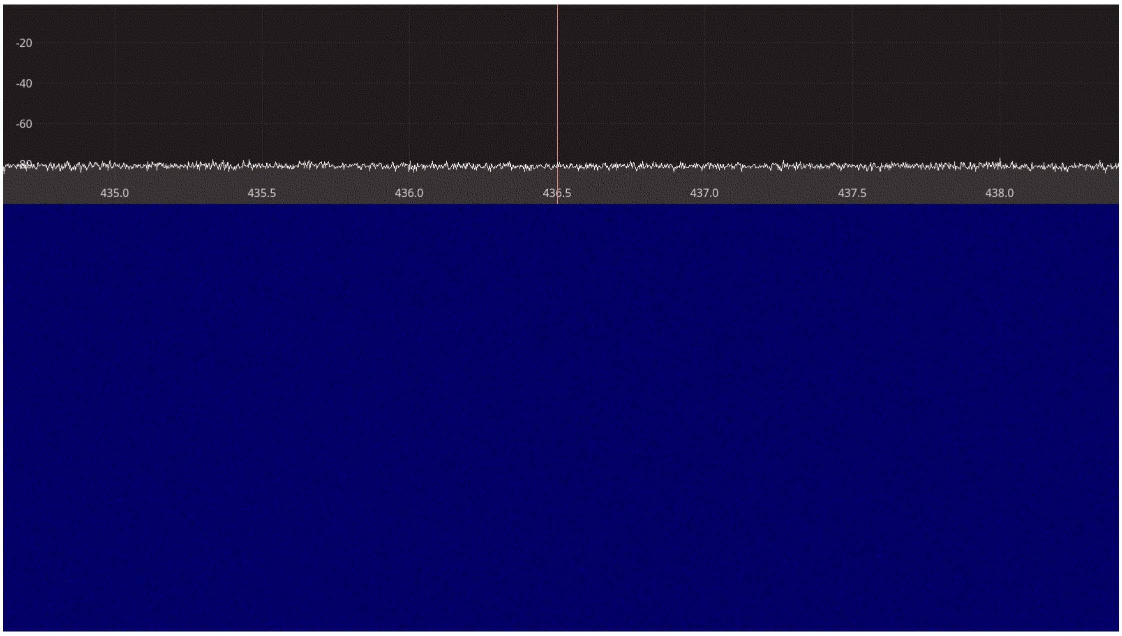


Figure 2.32: Airspy Mini receiver spectrum between 435 – 438 MHz.

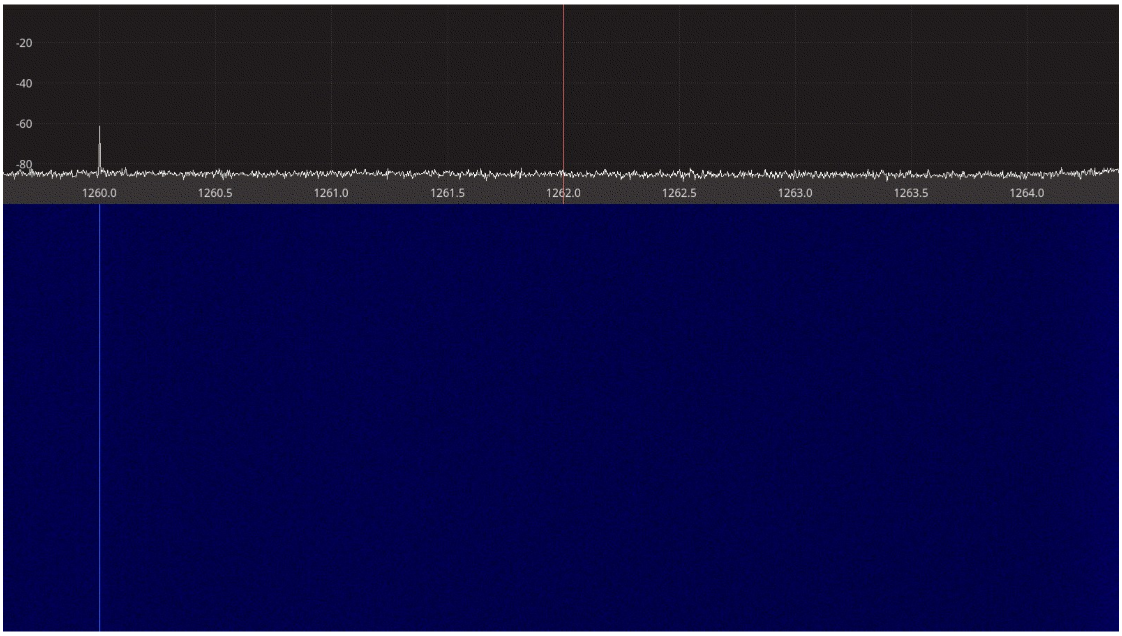


Figure 2.33: Airspy Mini receiver spectrum between 1260 – 1264 MHz.

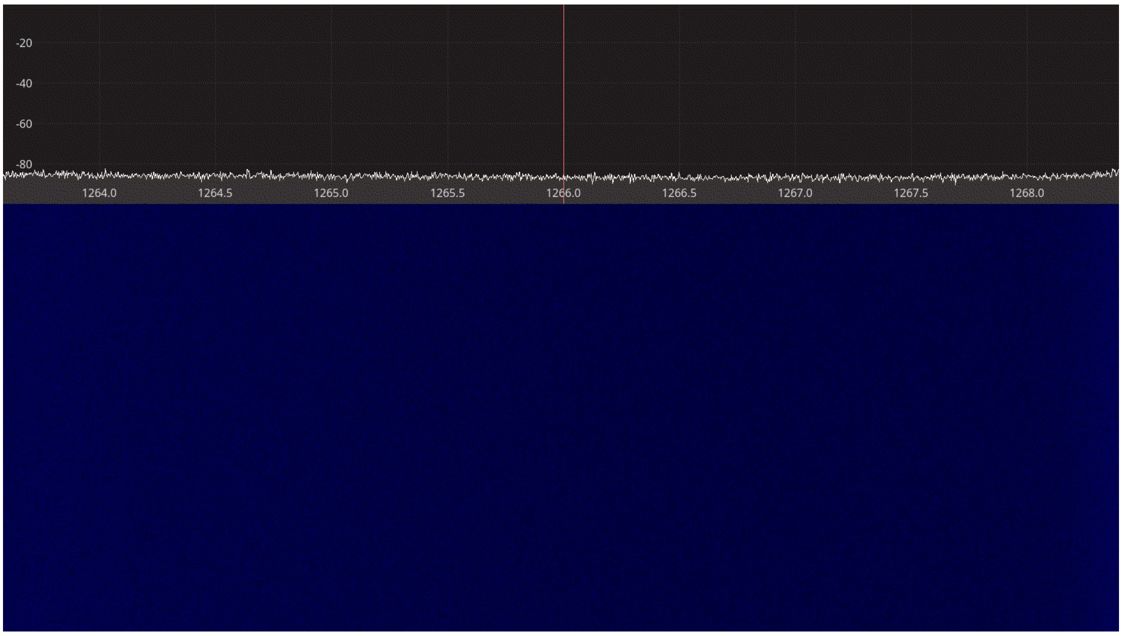


Figure 2.34: Airspy Mini receiver spectrum between 1264 – 1268 MHz.

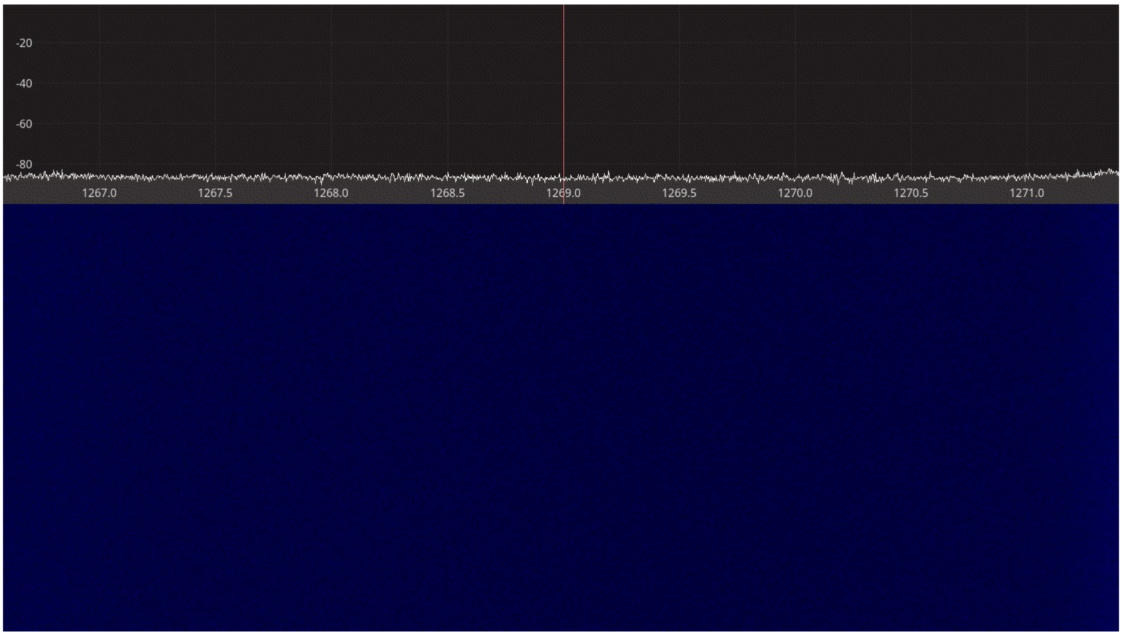


Figure 2.35: Airspy Mini receiver spectrum between 1267 – 1271 MHz.

2.2.3 SDRplay RSPduo

The SDRplay RSPduo is a dual-tuner 14-bit receiver based on a tuner and USB chipset from Mirics, covering the entire spectrum from 1 kHz to 2 GHz with up to 10 MHz maximum bandwidth [1].

Unlike the other devices with multiple receive paths, the two tuners in the SDRplay RSP duo function independently of each other and can receive different frequency bands at the same time. This makes them particularity interesting for applications such as a SatNOGS ground station, where VHF and UHF could be received simultaneously using just one device.

Other noteworthy features of SDRplay devices include the extensive set of band-pass and notch filters as well as the extensive documentation of receiver performance measurements available on the website [17].

Figure 2.36: The SDRplay RSPduo receiver.

Software support for SDRplay devices is available through a user space library called libmirsdrapi-rsp. Unlike all the drivers for other SDR devices tested during this activity, the SDRplay driver library is closed-source and available as binary download only. Furthermore, the software API provided by the driver is not as simple and intuitive as the ones provided by e.g. RTL-SDR or Airspy.

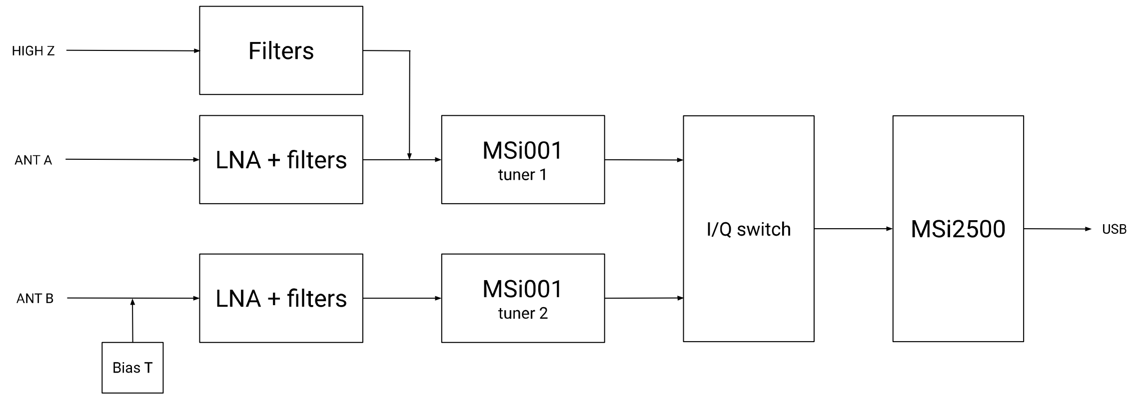


Figure 2.37: The SDRplay RSPduo block diagram.

A summary of the technical specifications for the SDRplay RSPduo is listed in Table 2.7 below.

Frequency Range	1 kHz – 2 GHz
Sample rate	2 – 10.66 MSPS
RF bandwidth	200 – 8000 kHz
Frontend filters	2 MHz low-pass 2-12 MHz 30-60 MHz 60-120 MHz 120-250 MHz 250-300 MHz 300-380 MHz 380-420 MHz 420-1000 MHz 1000 MHz high-pass
RX paths	2
RX inputs	3
ADC resolution	14-bit up to 6.048 MSPS 12-bit between 6.048-8.064 MSPS 10-bit between 8.064-9.216 MSPS 8-bit above 9.216 MSPS
Claimed noise figure	~ 2.5 dB below 500 MHz ~ 4.5 dB below 1.3 GHz
Clained dynamic range	-
Reference clock	0.5 PPM TCXO
Other features	Independent tuners Bias T External clock
Temperature range	-
Size	9.5 cm × 8.0 cm × 3 cm
Weight	315 g
Approximate price	240 €
Product page	https://www.sdrplay.com/rspduo/

Table 2.7: SDRplay RSPduo technical specifications.

2.2.3.1 Receiver Noise Figure

The settings used during the SDRplay RSPduo noise figure measurements are listed in Table 2.8 and the results are shown on Figure 2.38 below.

Input rate	Decimation	Sample rate	IF Bandwidth	IF gain
4 MHz	8	500 kHz	600 kHz	-20 dB

Table 2.8: SDRplay RSPduo settings used during noise figure measurements.

We note that:

- As with the previous devices, the noise figure on 1280 MHz is worse than on 145 and 437 MHz.
- The lowest noise figure we measured was 2.5 dB using the MDS method and 2.1 dB using the Y-factor method. This indicates that our test setup is capable of measuring such low noise figures.
- The measured noise figures are overall consistent with the data available on the SDR-play website, although direct 1:1 comparison is not possible due to the complicated gain API provided by the driver.

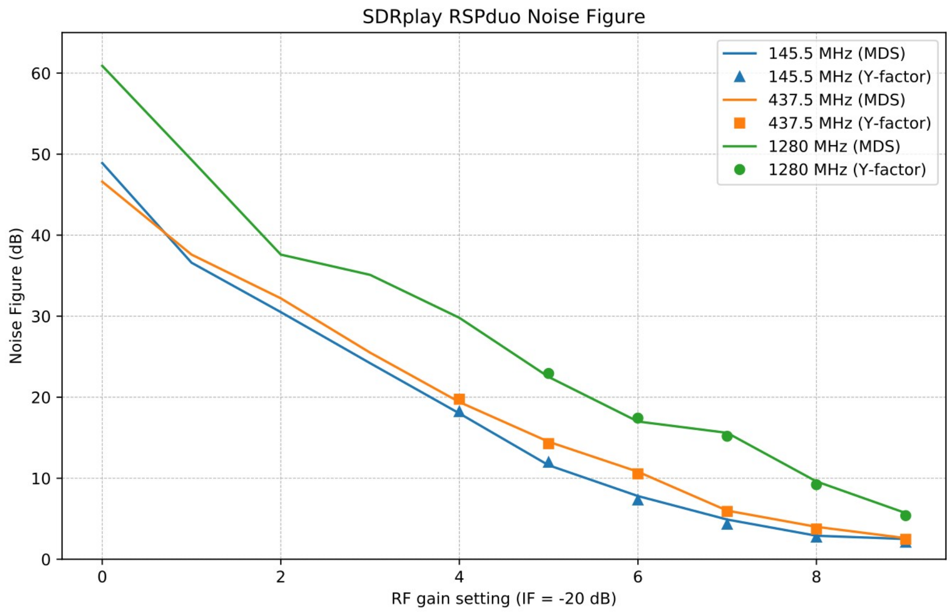


Figure 2.38: SDRplay RSPduo noise figure measurements.

2.2.3.2 Receiver Blocking Dynamic Range

The settings used during the SDRplay RSPduo dynamic range measurements are listed in Table 2.9 and the results are shown on Figure 2.39 below.

Input rate	Decimation	Sample rate	IF Bandwidth	IF gain
4 MHz	2	2 MHz	200 kHz	-40 dB

Table 2.9: SDRplay RSPduo settings used for BDR measurements.

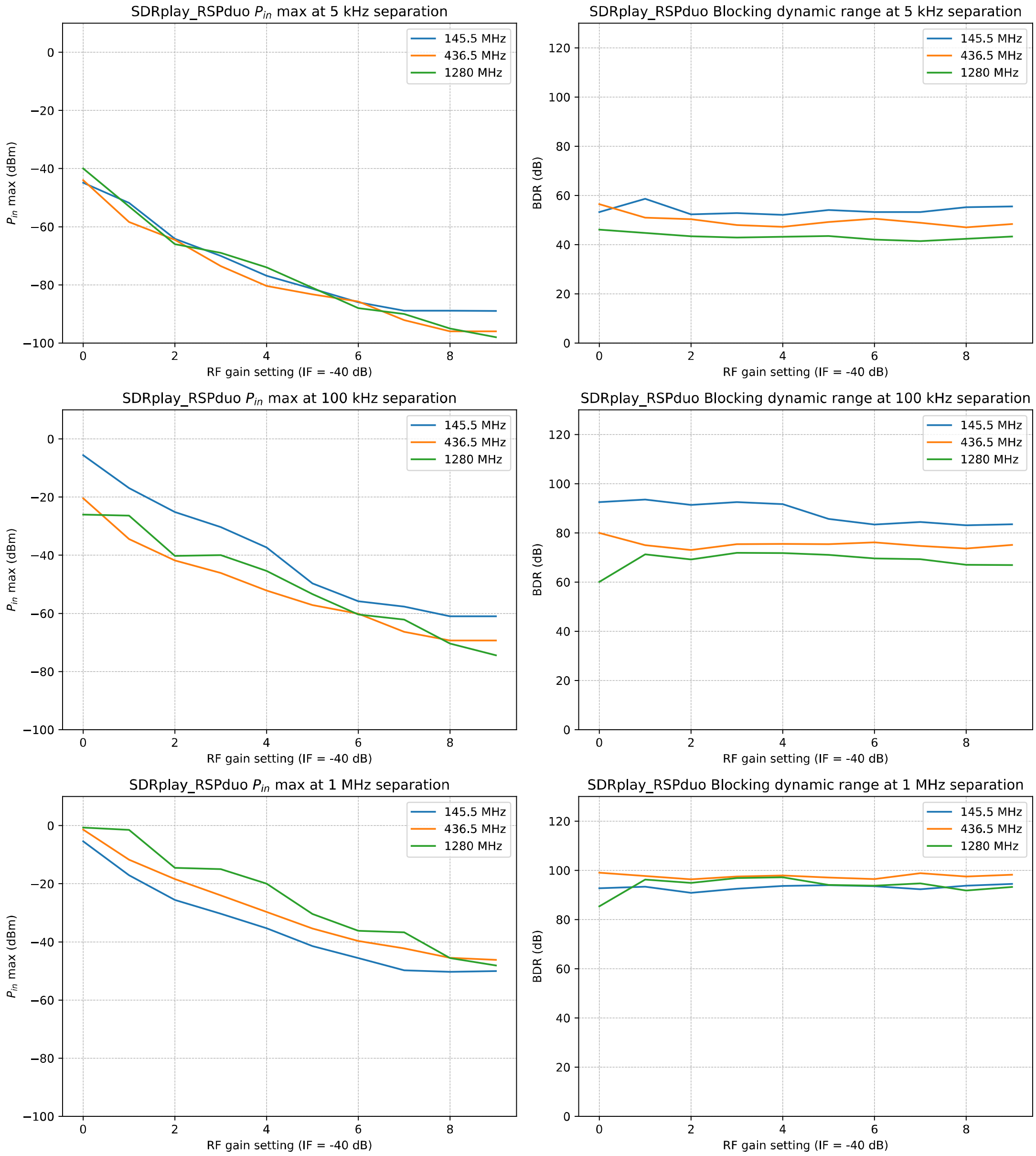


Figure 2.39: SDRplay RSPduo blocking dynamic range measurements.

We note that:

- Ax expected, the highest input power that the device can tolerate without overload, P_{in}, is decreasing as the gain is increased. The relationship is mostly linear except on 1280 MHz.
- The dynamic range is mostly constant over the entire gain spectrum, slightly dropping at the highest gains.

- The receiver has a rather poor dynamic range when the disturbing signal is only 5 kHz away but improves significantly at larger separations, thanks to the narrow IF filter.
- Changing the IF gain can have large influence on the dynamic range and, as we have seen, not so much on the noise figure. That is why we chose -40 dB instead of the -20 dB used during the noise figure measurements.

2.2.3.3 Receiver Spectral Purity

The receiver spectra of the SDRplay RSPduo receiver are shown on Figures 2.40 to 2.43 below. As we can see from the spectra, there are very few spurs generated by or picked up by the device when connected to 50 ohm terminator.

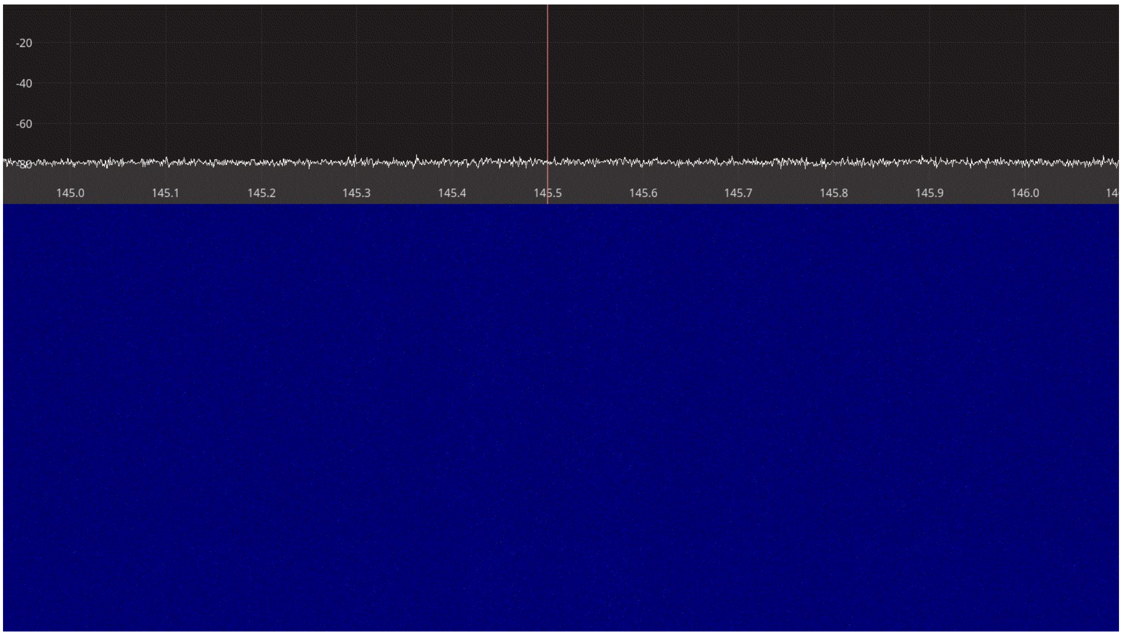


Figure 2.40: SDRplay RSPduo receiver spectrum between 145 – 146 MHz.

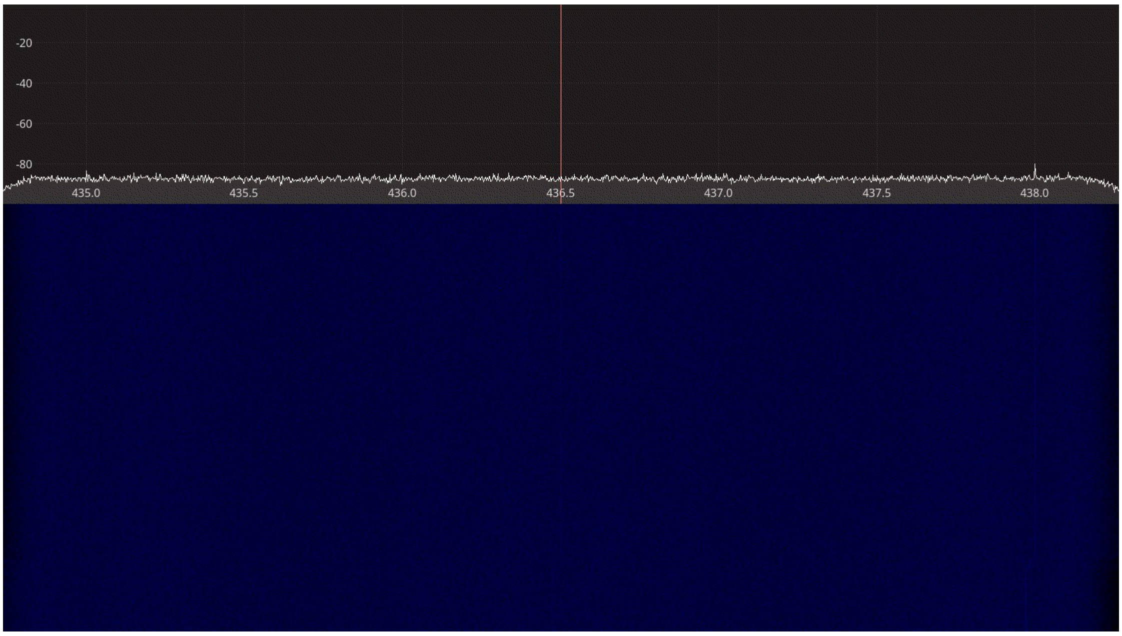


Figure 2.41: SDRplay RSPduo receiver spectrum between 435 – 438 MHz.

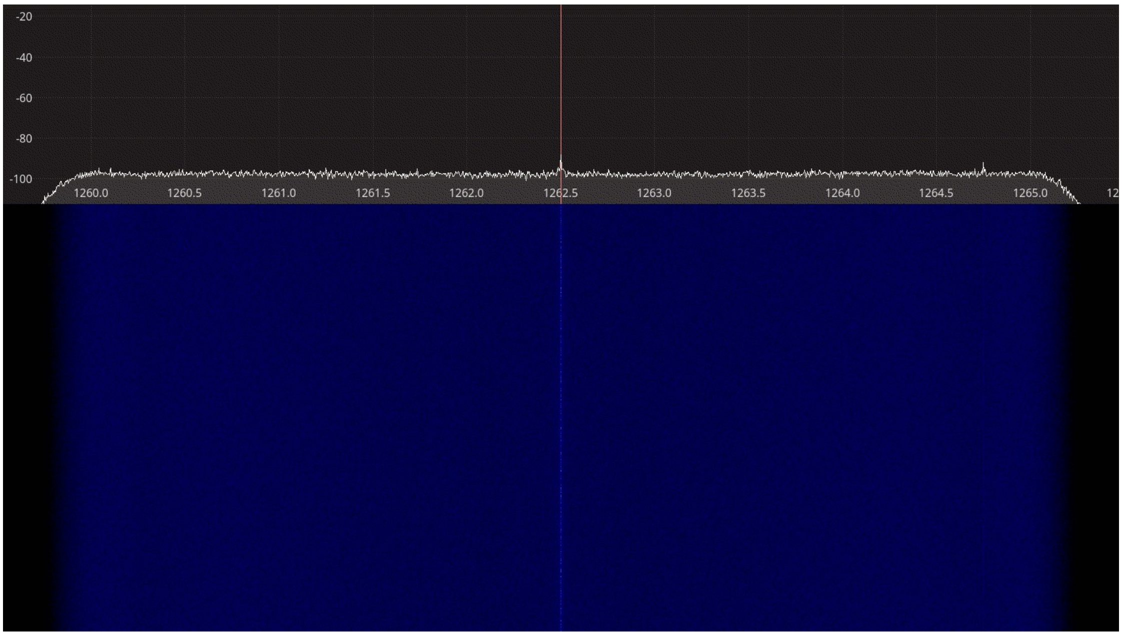


Figure 2.42: SDRplay RSPduo receiver spectrum between 1260 – 1265 MHz.

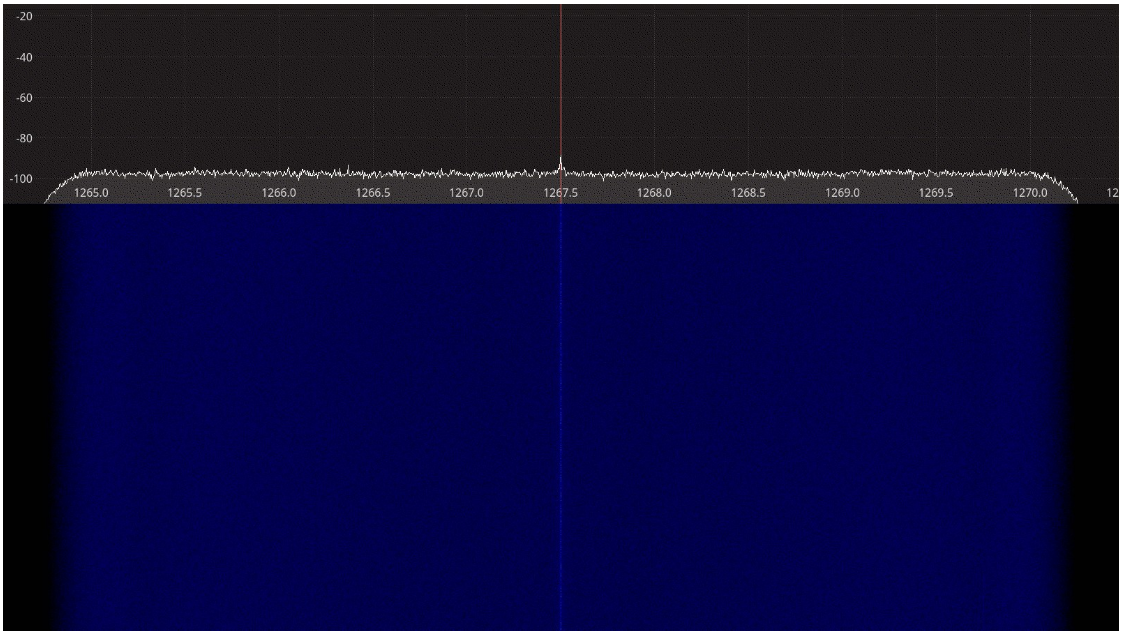


Figure 2.43: SDRplay RSPduo receiver spectrum between 1265 – 1270 MHz.

2.2.4 LimeSDR Mini

The LimeSDR Mini is a small 10 MHz – 3.5 GHz full-duplex transceiver based on the LMS7002M integrated transceiver from Lime Microsystems [18]. Its small size and low cost makes it a popular choice among hobbyists looking for entry-level SDR transceiver hardware. Although the LMS7002M is capable of delivering bandwidths up to 96 MHz [19], the components used in the LimeSDR Mini limit its capabilities to 30.72 MHz.

Both the LimeSDR Mini hardware and accompanying software come as open-source with design files and source code available on GitHub [20]. All LimeSDR devices are controlled through a LimeSuite library, which provides a simple API to user applications.

Figure 2.44: LimeSDR Mini transceiver.

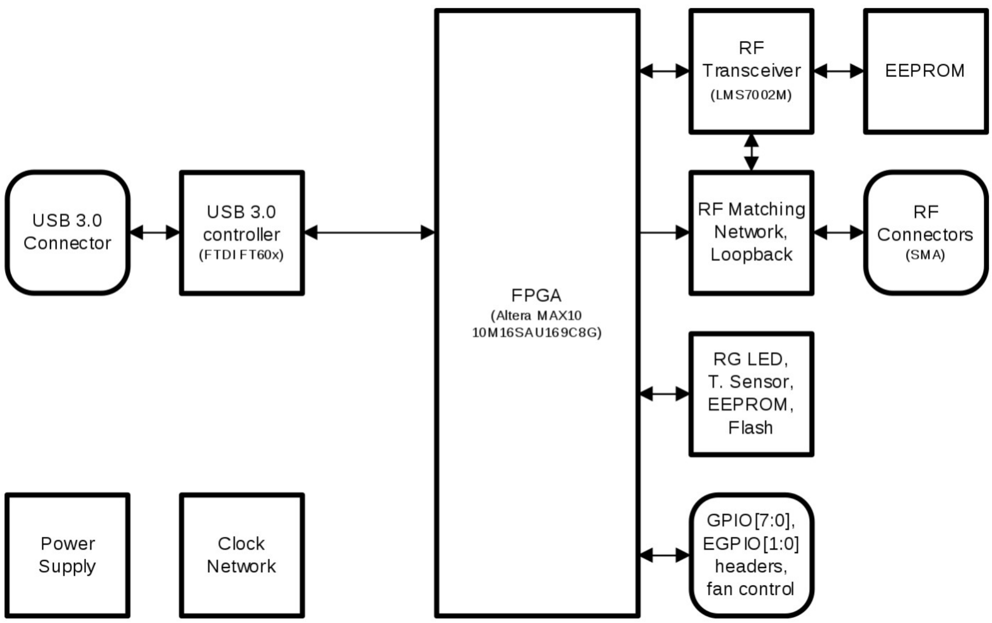


Figure 2.45: LimeSDR Mini block diagram.

A summary of the technical specifications for the LimeSDR Mini is listed in Table 2.10 below.

Frequency Range	10 – 3800 MHz
Sample rate	30.72 MSPS
RF bandwidth	30.72 MHz
Frontend filters	Variable bandwidth low-pass filter before ADC
RX paths	RX1_W for 10 MHz – 2 GHz RX1_H for 2 – 3.5 GHz
RX inputs	1
TX paths	TX1_1 for 2 – 3.5 GHz TX1_2 for 10 MHz – 2 GHz
TX outputs	1
ADC resolution	12
DAC resolution	12
Claimed noise figure	-
Clained dynamic range	-
Claimed transmit power	-
Reference clock	VCTCXO (±1 ppm initial, ±4 ppm stable)
Other features	On-board EEPROM and flash for firmware and FPGA image GPIO through header External clock through uFL connectors Open source design
Temperature range	-
Size	9.5 cm × 3.6 cm × 1.3 cm (with enclosure)
Weight	42 g
Approximate price	159 USD (board only) 299 USD (with aluminum enclosure)
Product page	https://limemicro.com/products/boards/limesdr-mini/

Table 2.10: LimeSDR Mini technical specifications.

2.2.4.1 Receiver Noise Figure

The settings used during the LimeSDR Mini noise figure measurements are listed in Table 2.11 and the results are shown on Figure 2.46 below.

Input rate	Decimation	Sample rate	RF Path	Bandwidth	LPF
4 MHz	8	500 kHz	Auto	Auto	ON

Table 2.11: LimeSDR Mini settings used during noise figure measurements.

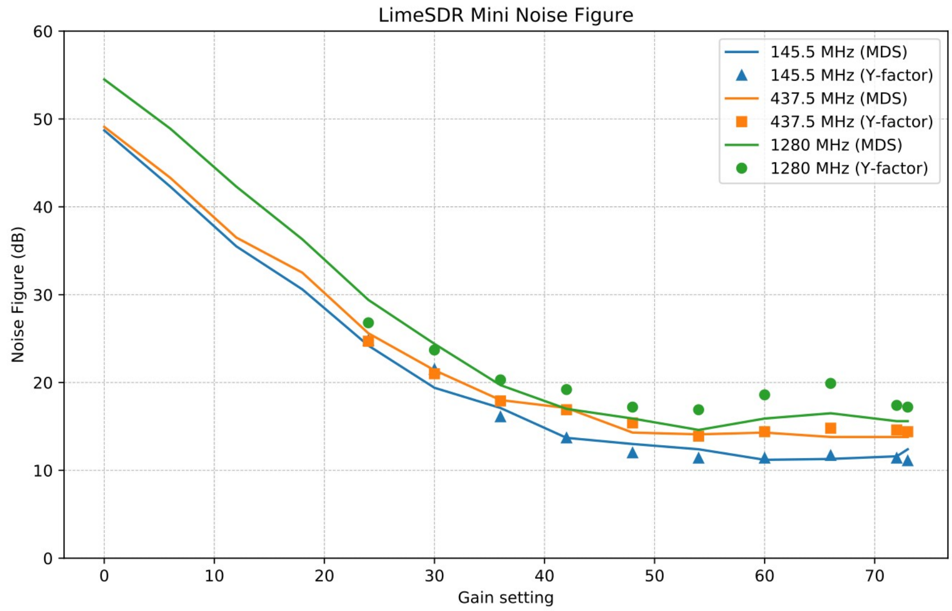


Figure 2.46: LimeSDR Mini noise figure measurements.

We note that:

- The LimeSDR Mini was in general suffering from excessive noise, which is clearly reflected by the poor noise figure we measured even at the highest gain.
- Above 2 GHz, the noise was so high and sensitive to touching the enclosure that we didn't find it meaningful to include the results.
- We had two units at our disposal and both of them behaved the same way.
- We have verified that Analog LFP ON/OFF does not impact the noise figure (within 1 dB).

The noise figure of the LMS7002M RFIC is around 2 dB [21] and it is unlikely that the difference is caused by insertion losses between the antenna connector and the RFIC. Because the noise level was sensitive to touching the case, we suspected that the case might be the cause of the poor noise figure. We took the board out of its aluminum case and repeated the noise figure measurements using the Y-factor method. Figure 2.47 shows these measurements on the LimeSDR Mini board only vs. the board inside the aluminum case.

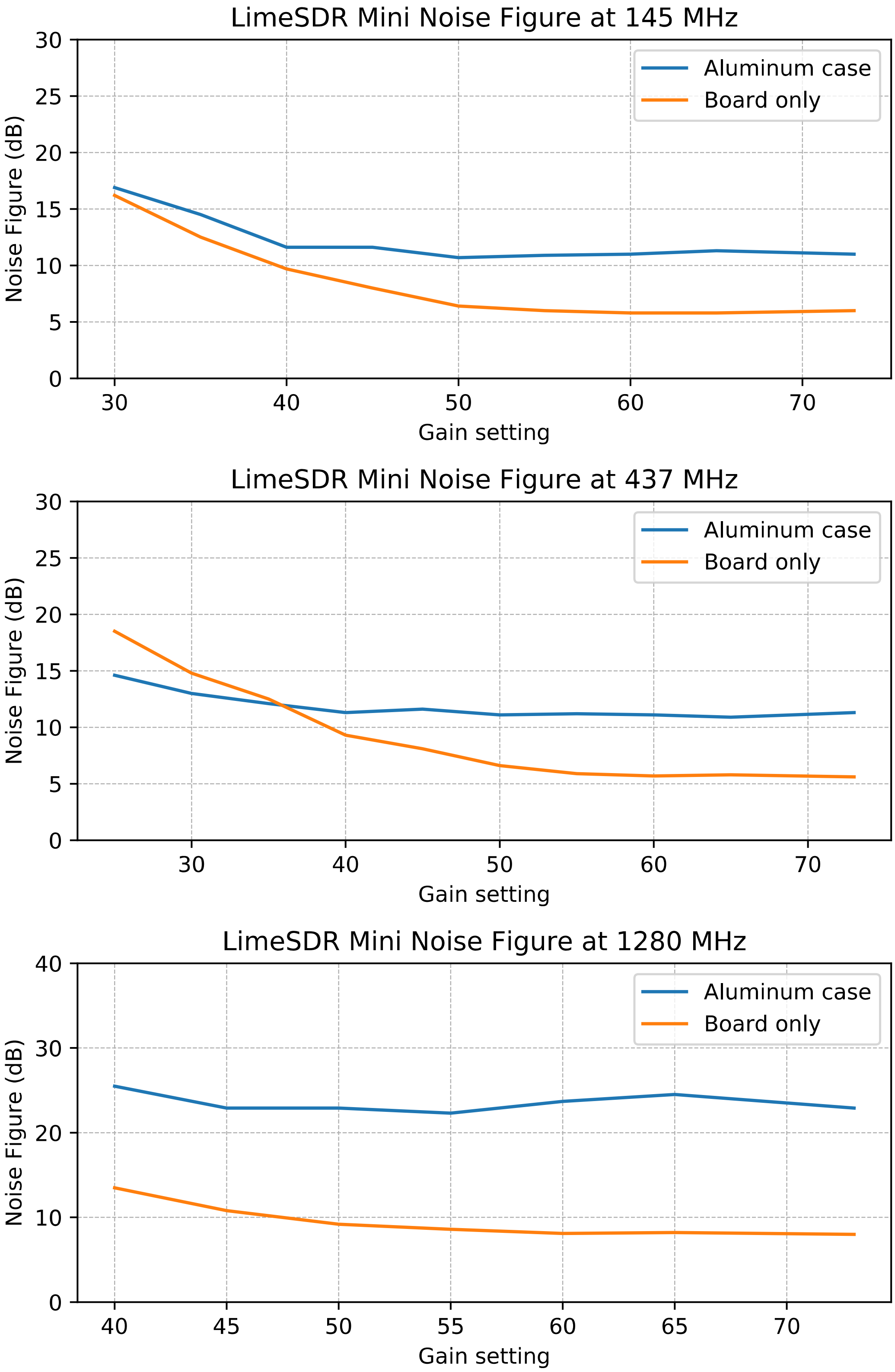


Figure 2.47: LimeSDR Mini noise figure measured with and without the aluminum case.

The measurements show clearly that the receiver noise figure is severely degraded by the aluminum case. Lime Microsystems investigated the issue and concluded that the issue could be caused by coupling via the case to the receive port SMA, where there was a scratch in the case paint. This can be resolved by isolating the SMA connector ground by adding a small piece of heat shrink around it.

The subsequent tests use the measurements on the board inside the case as reference since this is how our units were shipped from the factory.

2.2.4.2 Receiver Blocking Dynamic Range

The settings used during the LimeSDR Mini dynamic range measurements are listed in Table 2.12 and the results are shown on Figure 2.48 below.

Input rate	Decimation	Sample rate	RF path	Bandwidth	LPF
2 MHz	1	3 MHz	Auto	1401 kHz	ON

Table 2.12: LimeSDR Mini settings used for BDR measurements.

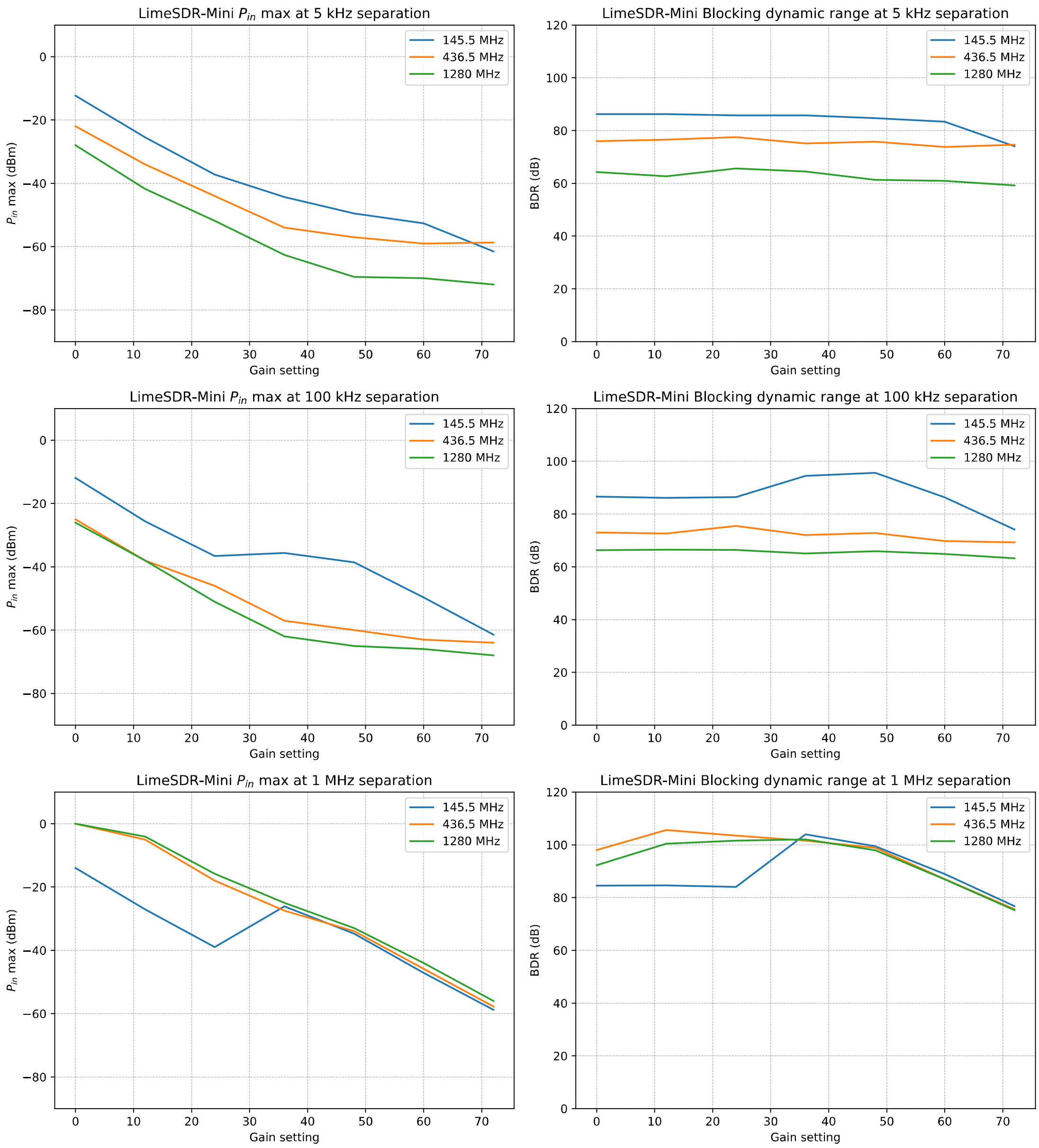


Figure 2.48: LimeSDR Mini blocking dynamic range measurements.

We note that:

- Ax expected, the highest input power that the device can tolerate without overload, P_{in}, is decreasing as the gain is increased.
- At 5 kHz and 100 kHz separation, the dynamic range is mostly constant over the entire gain spectrum, but varies more than 20 dB at 1 MHz separation.

2.2.4.3 Receiver Spectral Purity

The receiver spectra of the LimeSDR Mini receiver are shown on Figures 2.49 to 2.57 below. As we can see from the spectra, there are relatively few spurs generated by or picked up by the device when connected to 50 ohm terminator. However, the noise floor becomes significantly higher as we go above 2 GHz, which was also noted during the noise figure measurements.

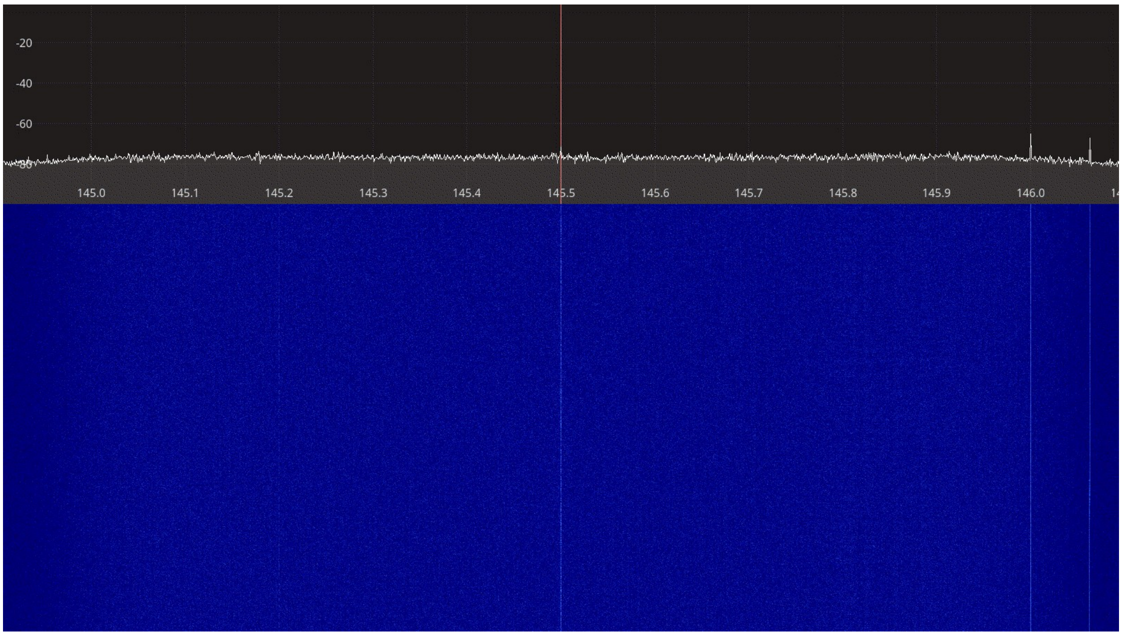


Figure 2.49: LimeSDR Mini receiver spectrum between 145 – 146 MHz.

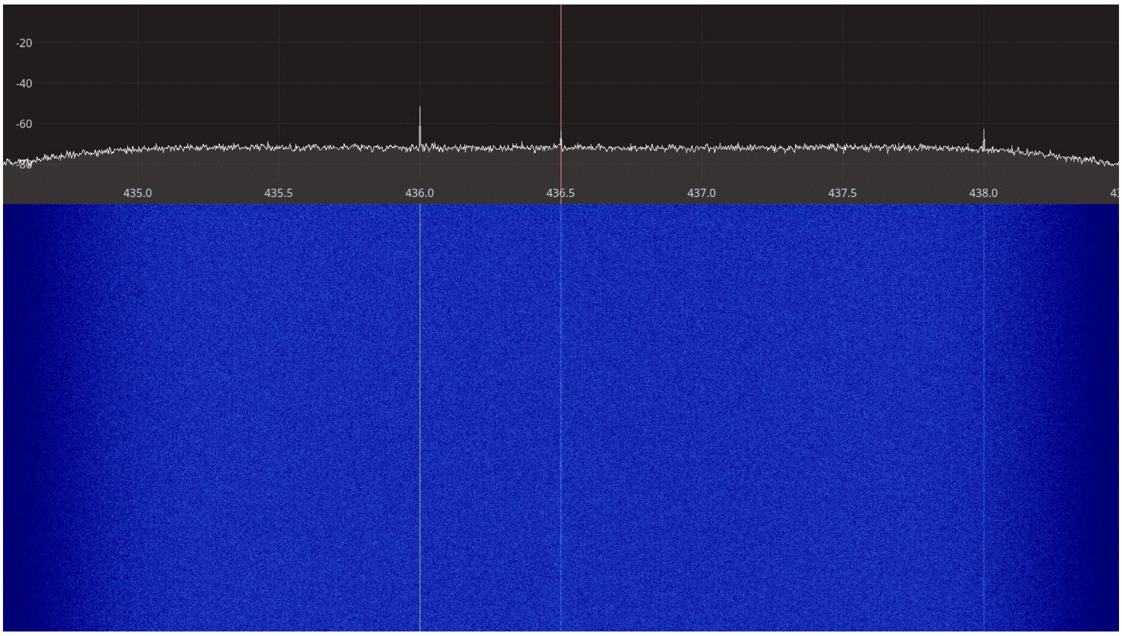


Figure 2.50: LimeSDR Mini receiver spectrum between 435 – 438 MHz.

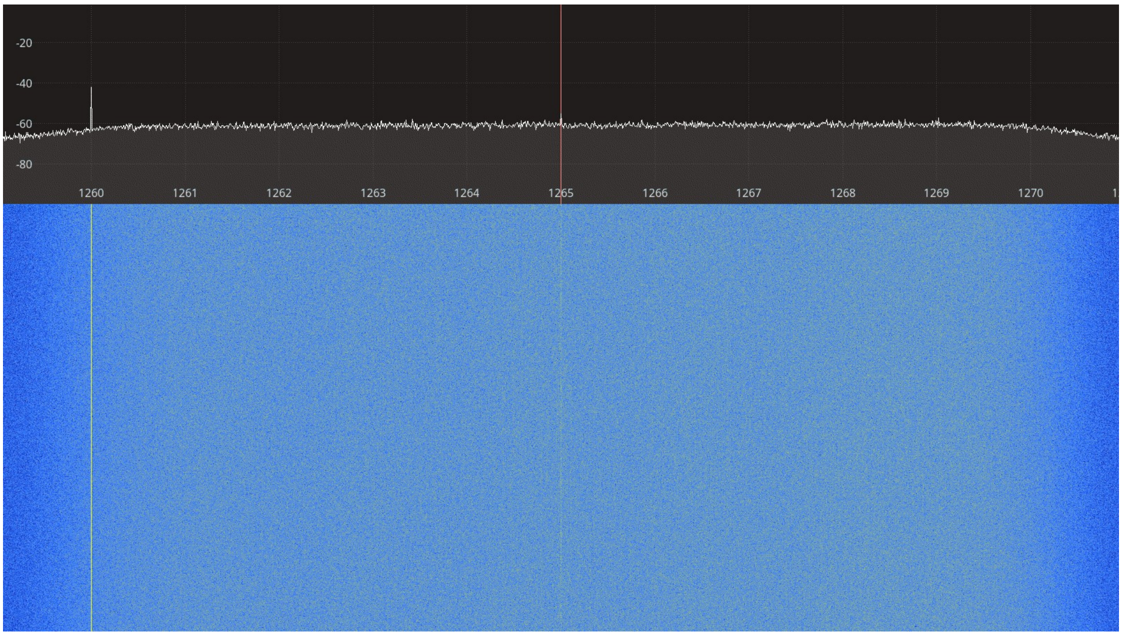


Figure 2.51: LimeSDR Mini receiver spectrum between 1260 – 1270 MHz.

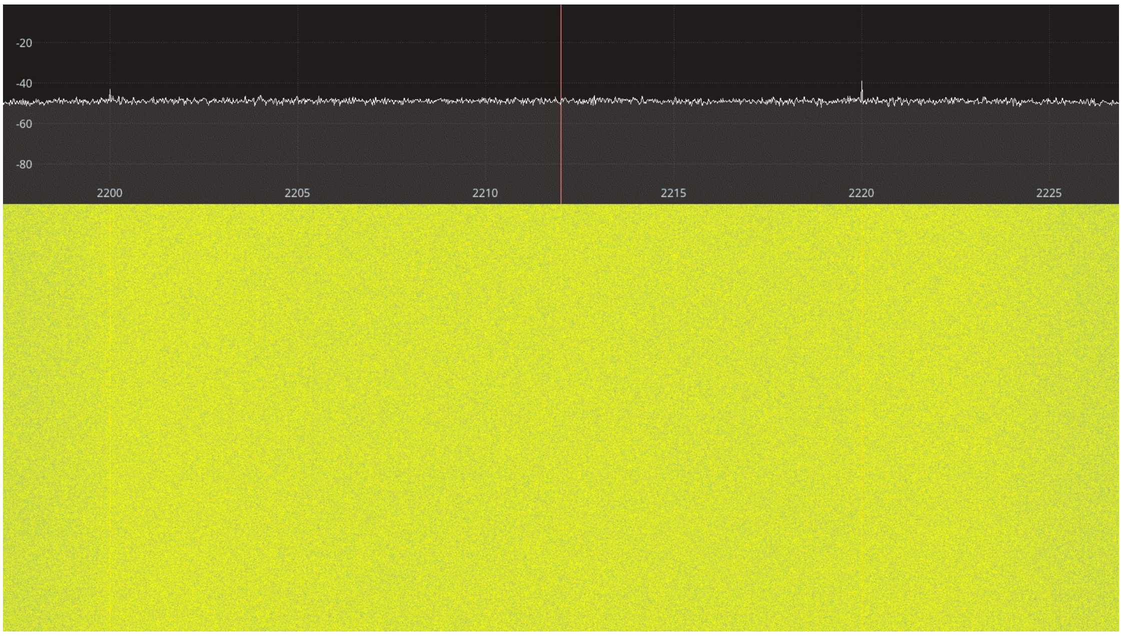


Figure 2.52: LimeSDR Mini receiver spectrum between 2200 – 2225 MHz.

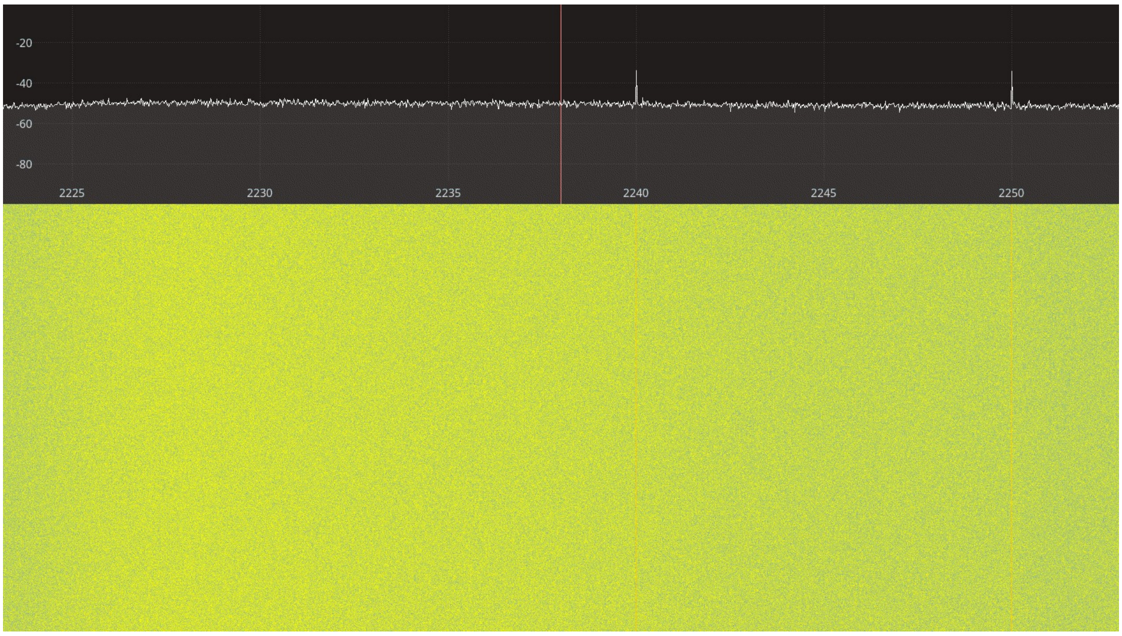


Figure 2.53: LimeSDR Mini receiver spectrum between 2225 – 2250 MHz.

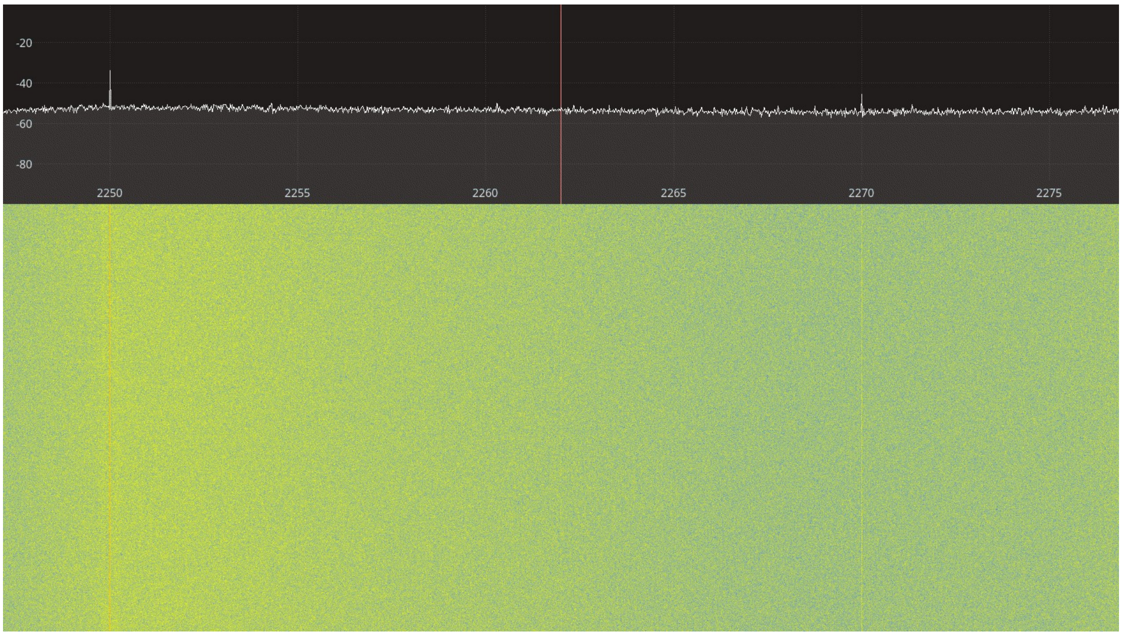


Figure 2.54: LimeSDR Mini receiver spectrum between 2250 – 2275 MHz.

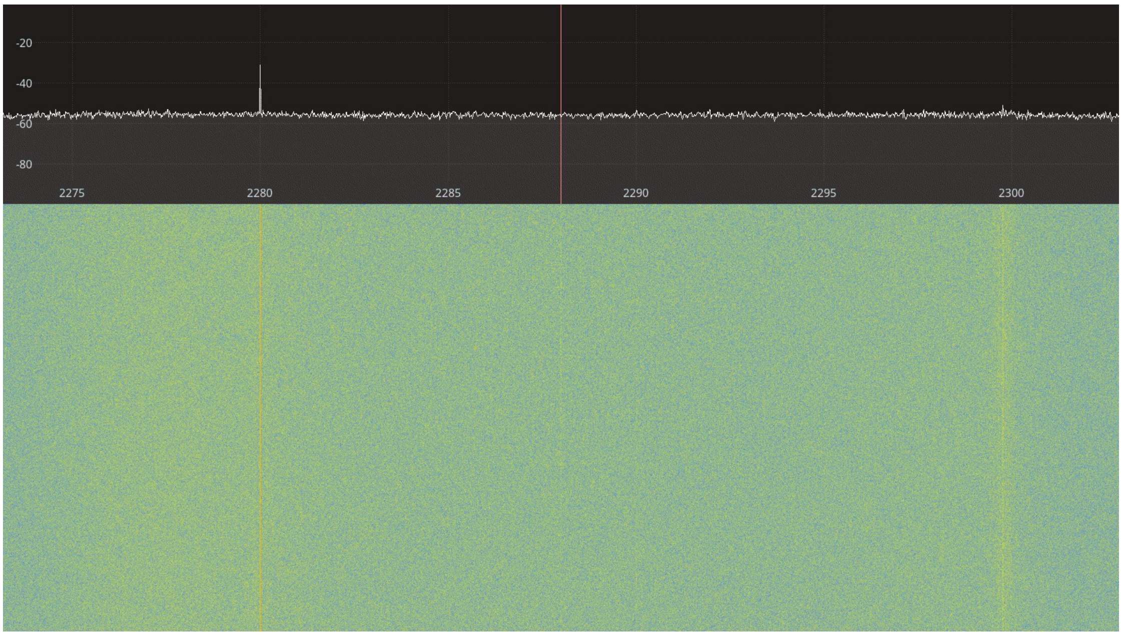


Figure 2.55: LimeSDR Mini receiver spectrum between 2275 – 2300 MHz.

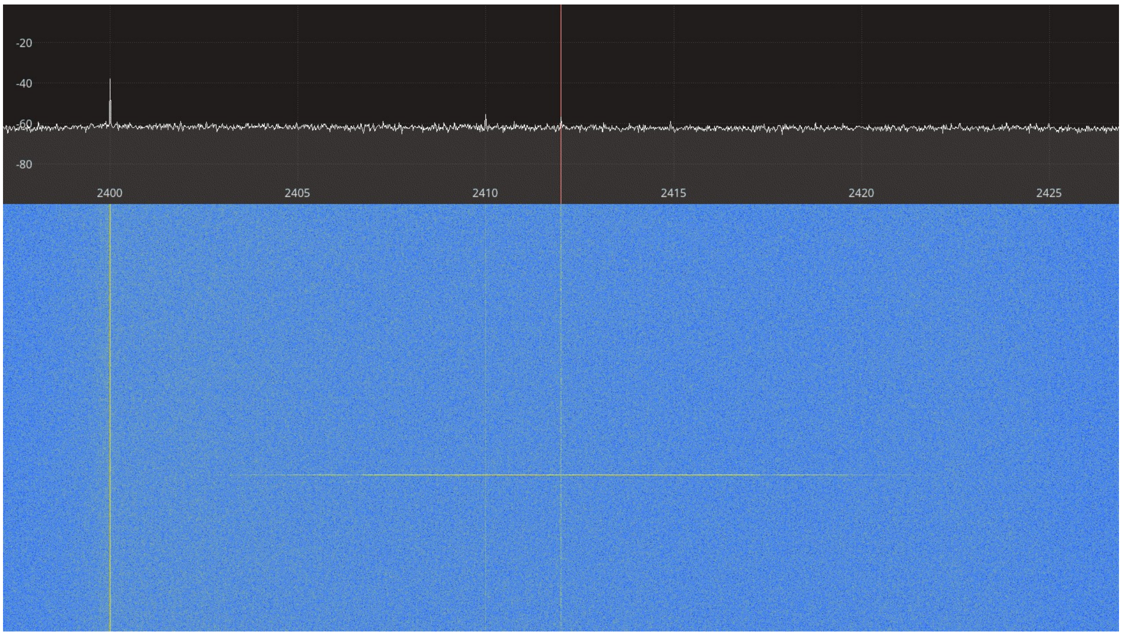


Figure 2.56: LimeSDR Mini receiver spectrum between 2400 – 2425 MHz.

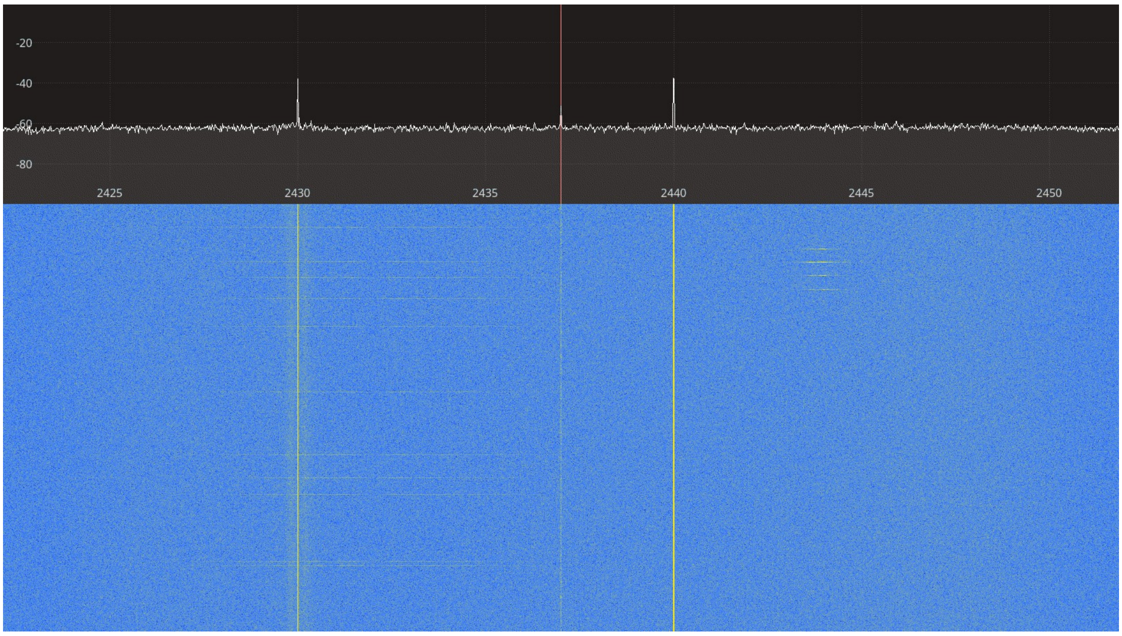


Figure 2.57: LimeSDR Mini receiver spectrum between 2425 – 2450 MHz.

2.2.4.4 Transmitter Spectral Purity

The transmitter spectra for the LimeSDR Mini transmitting a CW signal are shown on Figures 2.58 to 2.65 below. For each frequency, two spectra are shown, one at a gain 0 and one at gain 10. The tests were carried out with a modified gr-soapy interface, where the initial call to the IQ balancing function was disabled in order to prevent a large IQ imbalance caused by the default settings in gr-soapy.

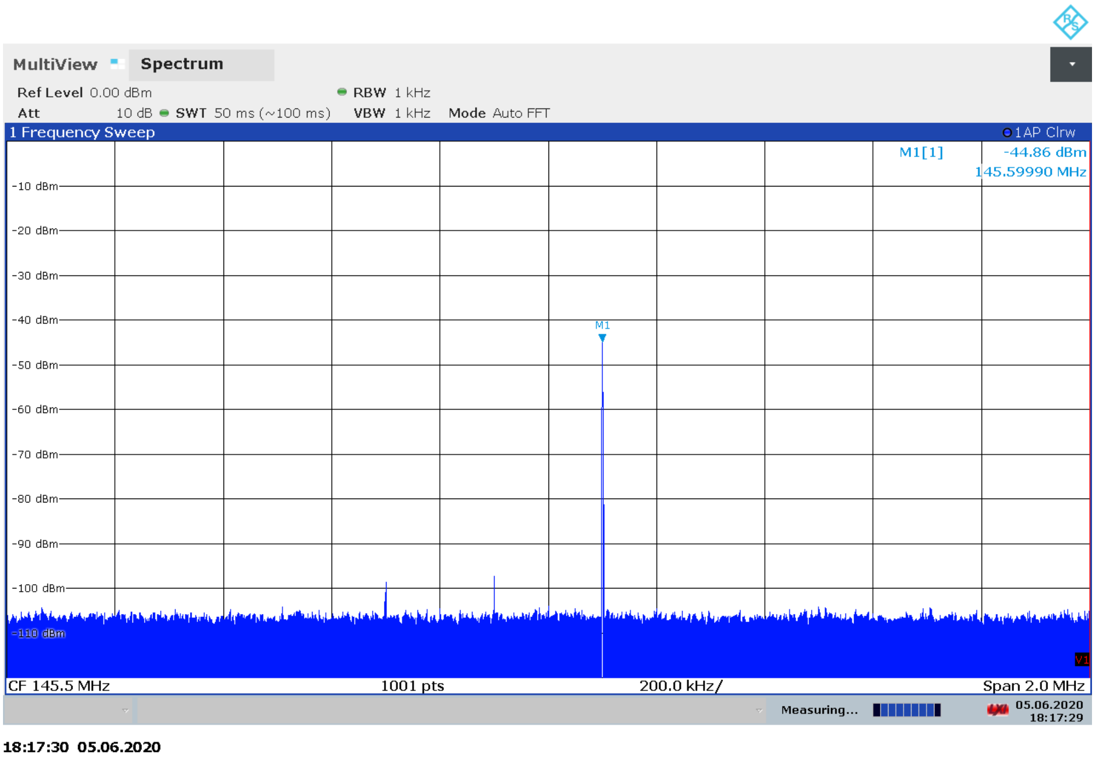


Figure 2.58: LimeSDR Mini transmitter spectrum at 145.6 MHz, gain 0.

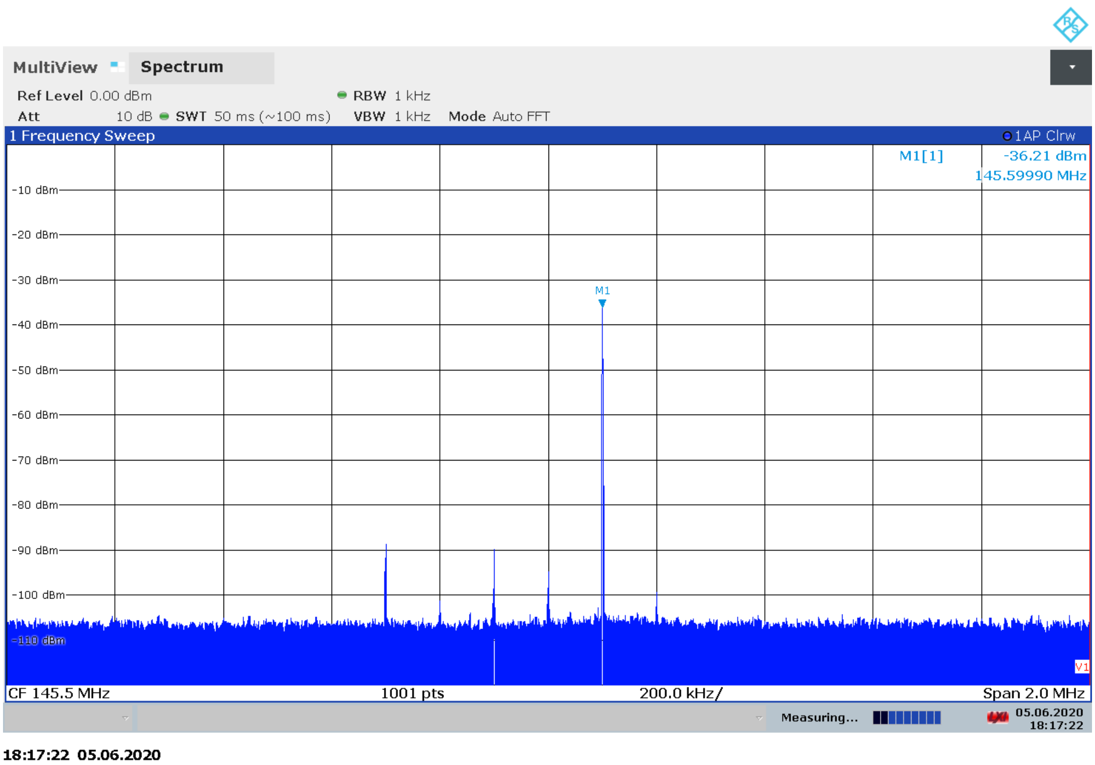


Figure 2.59: LimeSDR Mini transmitter spectrum at 145.6 MHz, gain 10.

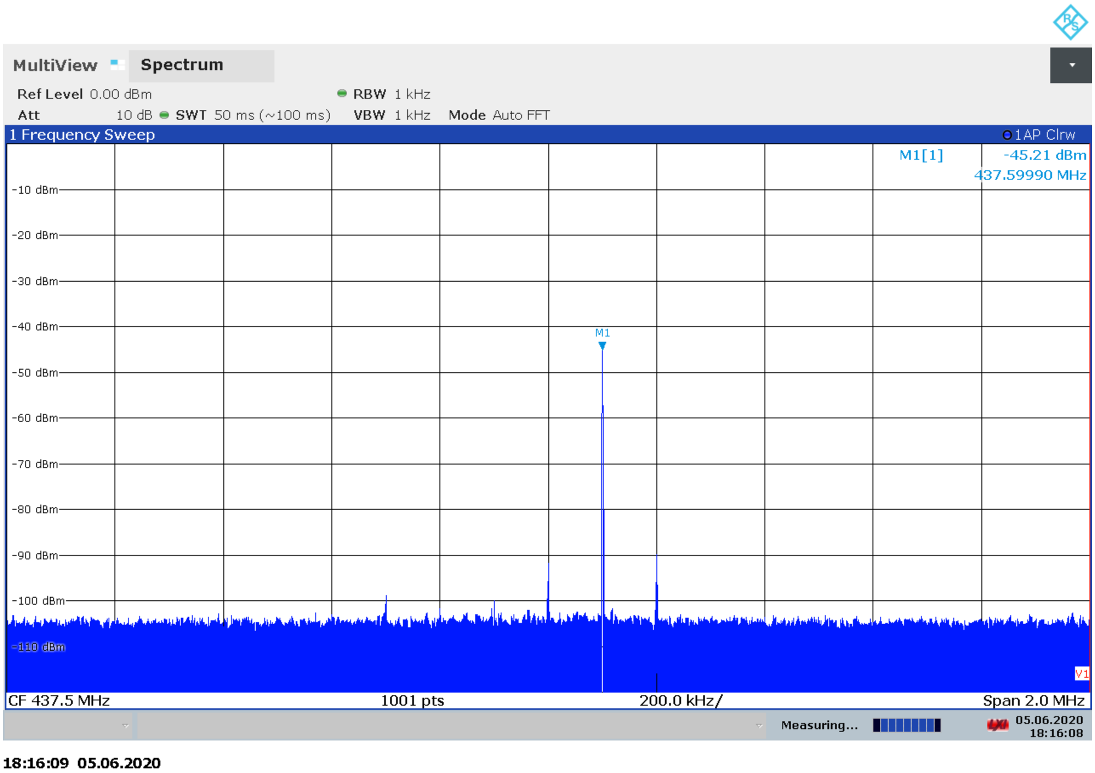


Figure 2.60: LimeSDR Mini transmitter spectrum at 437.6 MHz, gain 0.

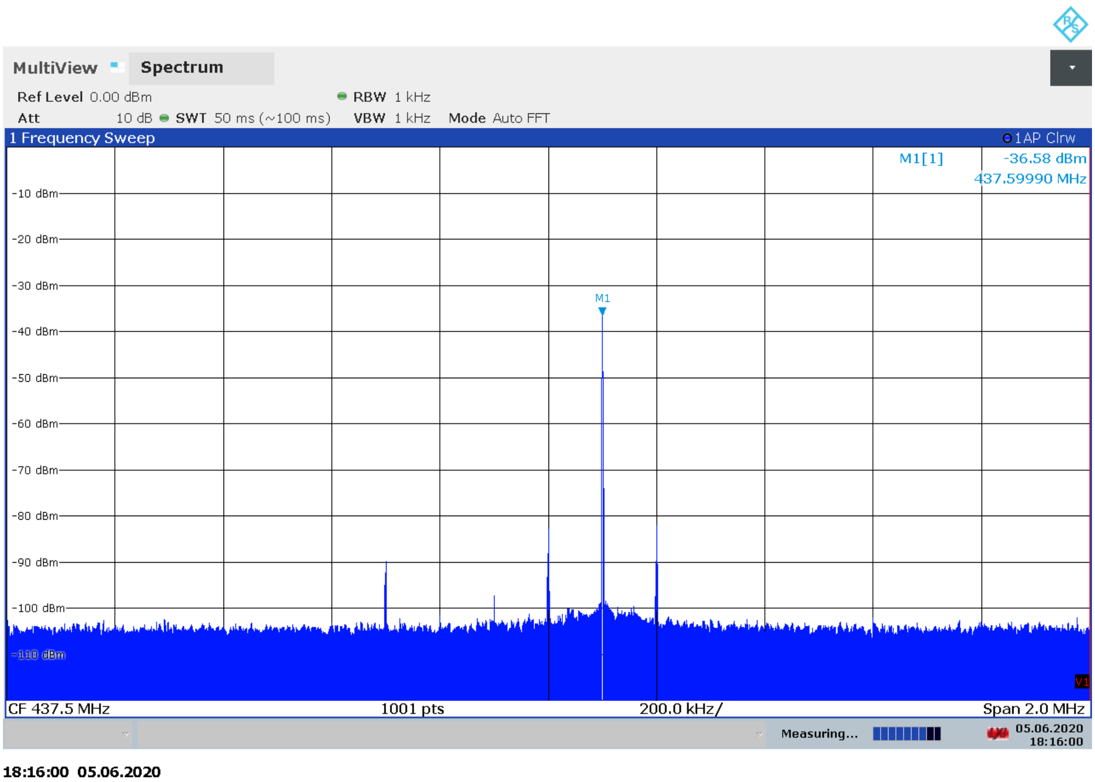


Figure 2.61: LimeSDR Mini transmitter spectrum at 437.6 MHz, gain 10.

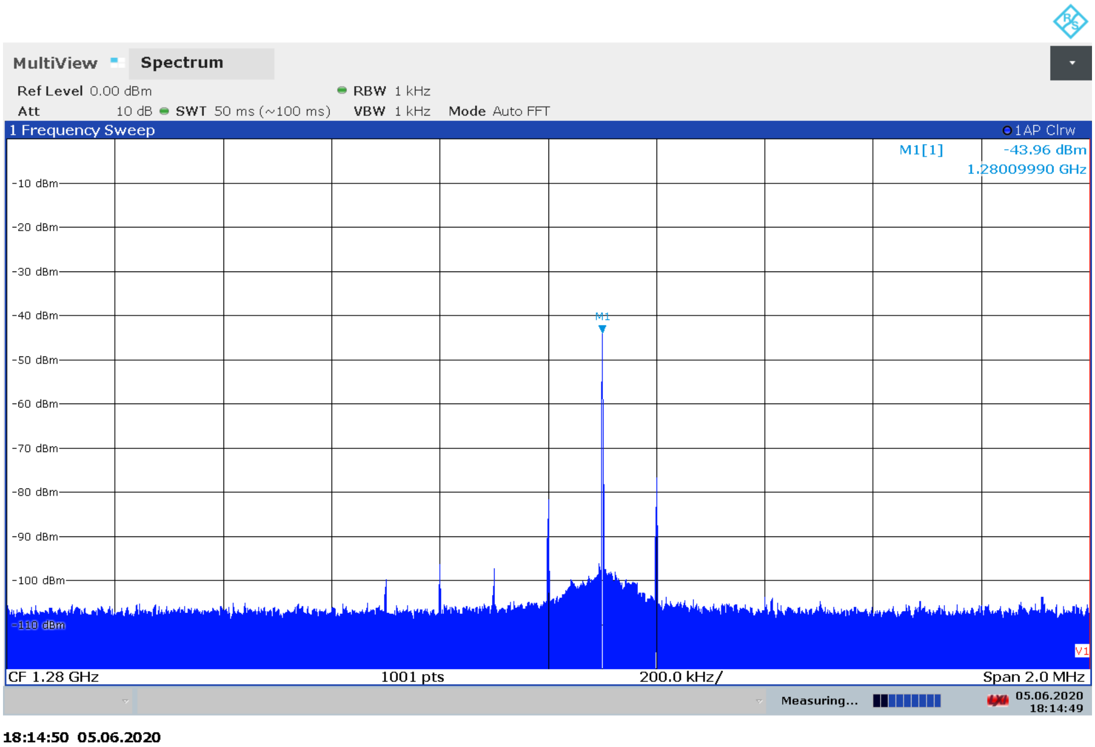


Figure 2.62: LimeSDR Mini transmitter spectrum at 1280 MHz, gain 0.

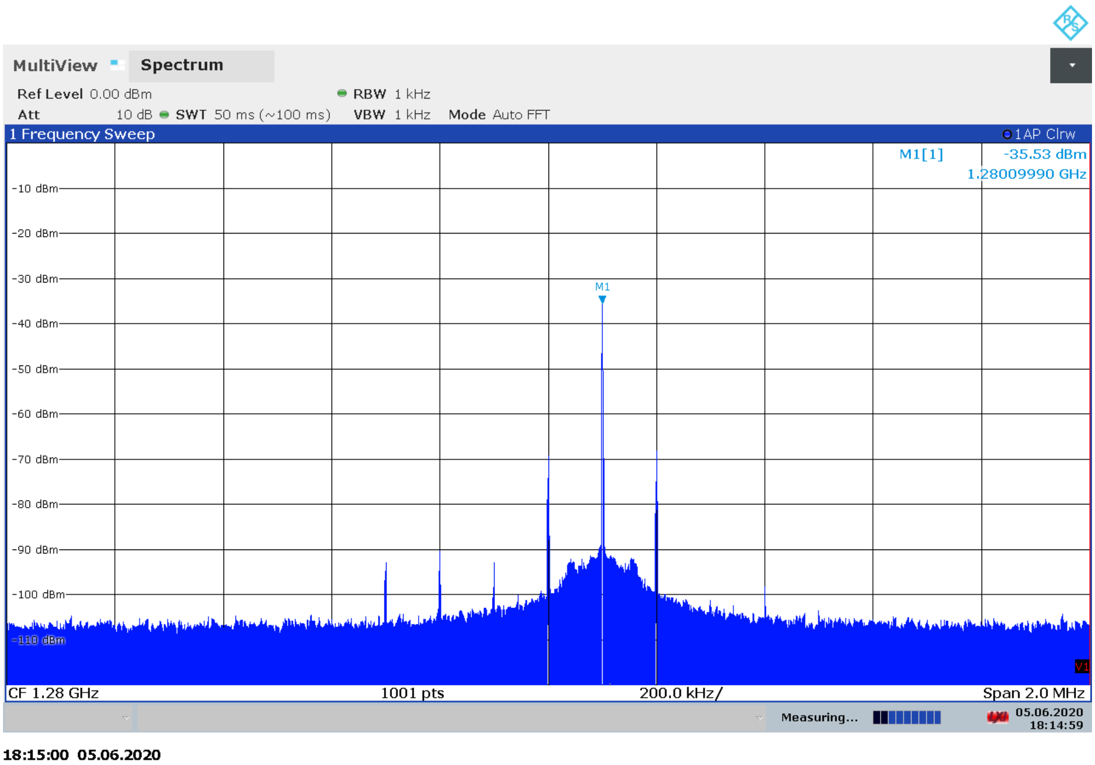


Figure 2.63: LimeSDR Mini transmitter spectrum at 1280 MHz, gain 10.

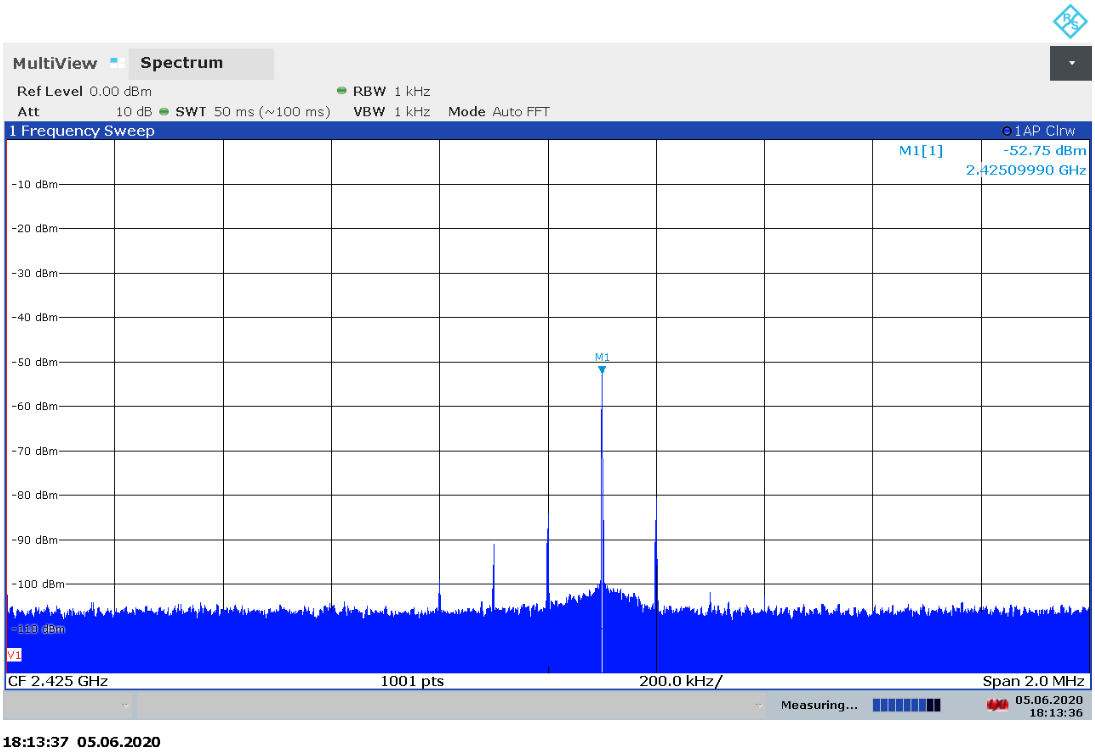


Figure 2.64: LimeSDR Mini transmitter spectrum at 2425 MHz, gain 0.

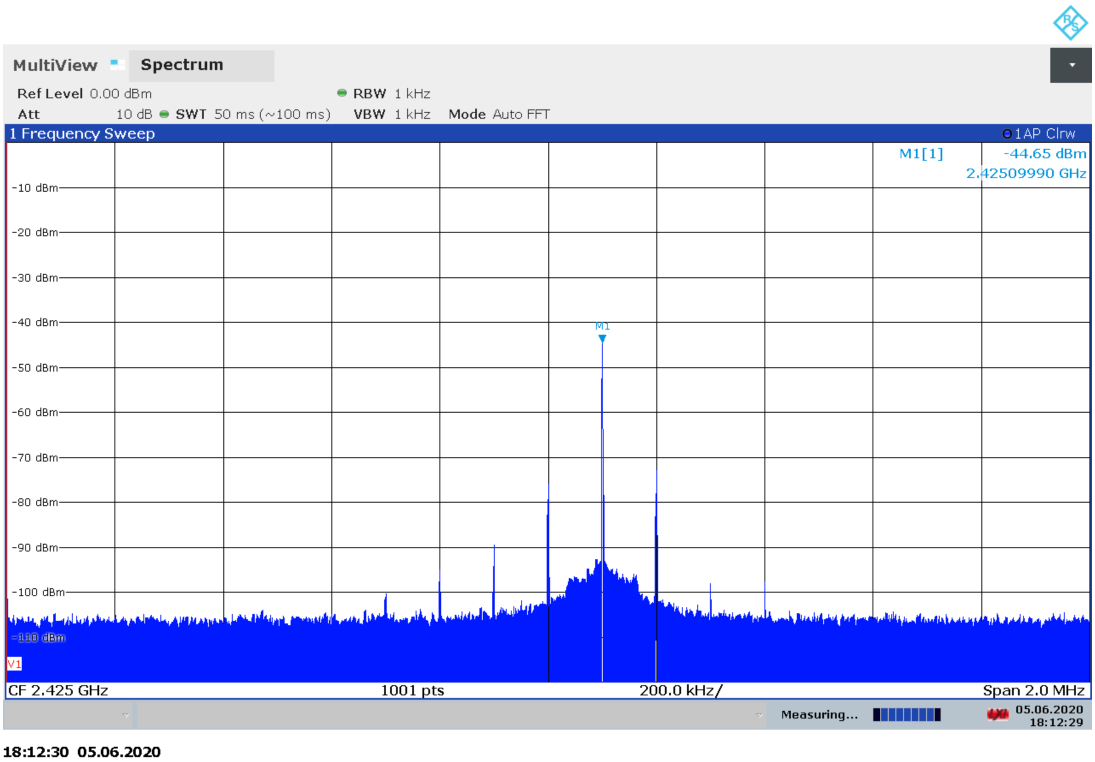


Figure 2.65: LimeSDR Mini transmitter spectrum at 2425 MHz, gain 10.

2.2.4.5 Transmitter Output Power

The measured transmitter output for the LimeSDR Mini is shown on Figure 2.66 below. As we can see there is a linear relationship between the transmitter gain setting and the output power, except at the lower and upper limits where the curve begins to flatten out.

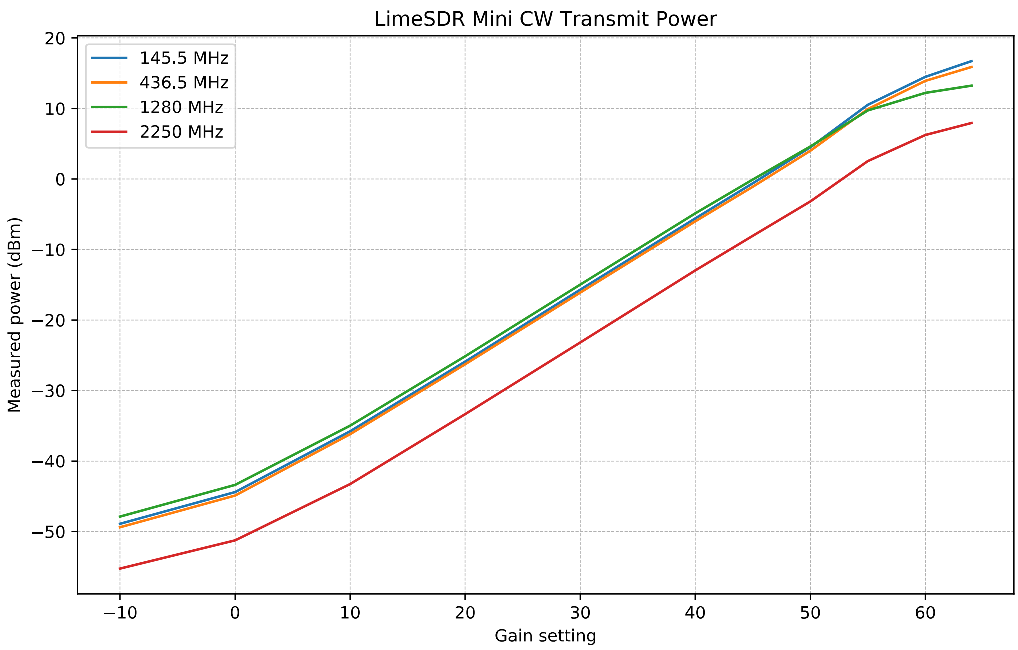


Figure 2.66: LimeSDR Mini CW transmitter output power as function of gain.

2.2.4.6 Transmitter MER

The measured output power and modulation quality for a DVB-S2 signal transmitted with the Mini are shown in Table 2.13 below. As we noted in test procedure description in section 2.1.6, we believe that the NsMAR parameter is a margin above the required E_S/N_0, which is 11 dB for the modulation and FEC settings used in our tests. Also note that the saturation in NsMAR and MER is due to the limitation of the signal analyzer and does not necessarily represent a limitation in the SDR device.

TX Gain	Power (dBm)	NsMAR (dB)	MER (dB)
-10	-60.9	7.4	19.8
-5	-57.3	12.7	25.0
0	-52.1	12.7	25.0
10	-42.7	11.5	25.0
20	-35.3	12.7	25.0
30	-22.9	12.7	25.0
40	-15.3	12.7	25.0
50	-3.0	12.7	25.0

Table 2.13: LimeSDR Mini transmitter MER as function of gain.

2.2.5 BladeRF 2.0 Micro

The BladeRF 2.0 micro is a 2×2 MIMO transceiver based on the AD9361 integrated RFIC from Analog Devices. It covers 70 MHz – 6 GHz with up to 61 MHz sample rate and 56 MHz analog bandwidth [22].

The wide bandwidth and frequency range make the BladeRF 2.0 micro an interesting choice for satellite communications, both as a receiver and as a transmitter.

BladeRF products are supported through an open-source driver library, which provides a simple and intuitive programming interface to the hardware. Firmware and FPGA sources are avaialble as well [23].

On the hardware side, the schematics are available in PDF format on the website [24].

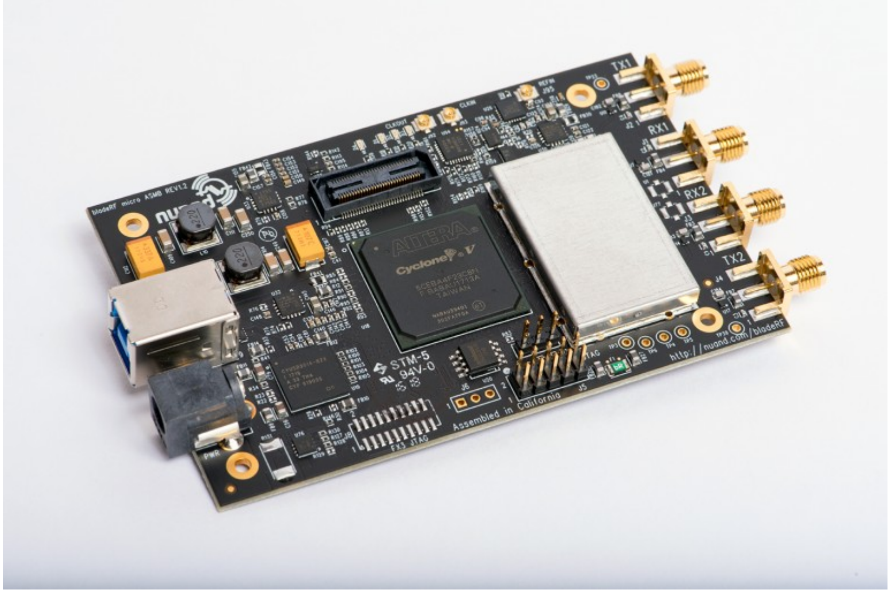


Figure 2.67: The BladeRF 2.0 micro transceiver.

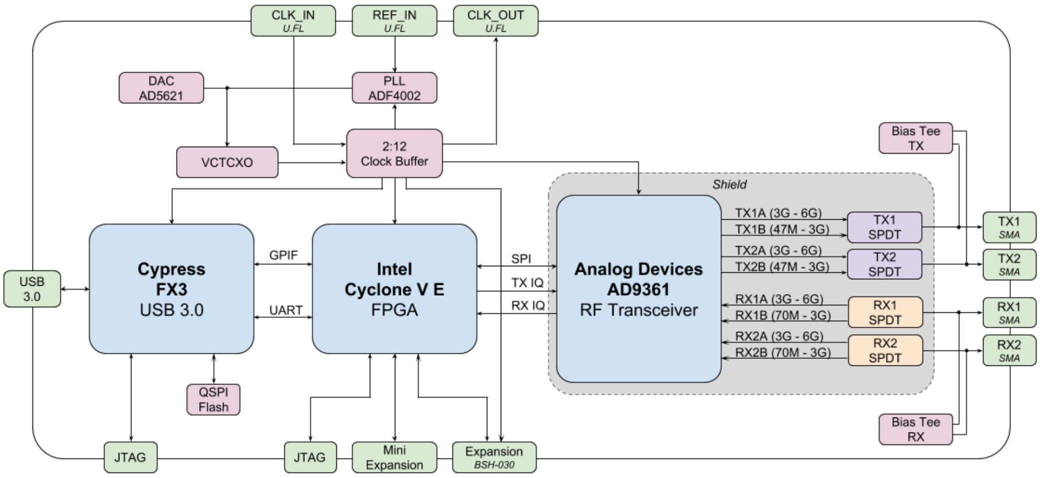


Figure 2.68: BladeRF 2.0 micro block diagram.

A summary of the technical specifications for the BladeRF 2.0 micro is listed in Table 2.14 below.

Frequency Range	70 MHz – 6 GHz
Sample rate	61.44 MSPS
RF bandwidth	200 kHz – 56 MHz
Frontend filters	None
RX paths	2
RX inputs	2
TX paths	2
TX outputs	2
ADC resolution	12
DAC resolution	12
Claimed noise figure	-
Claimed dynamic range	-
Claimed transmit power	8 dBm (CW)
Reference clock	Factory calibrated VCTCXO
Other features	Automatic IQ and DC offset correction 2×2 MIMO Bias-T on both TX and RX ports External clock through uFL connectors on the board
Temperature range	0-70 °C
Size	10.2 cm × 6.3 cm × 1.8 cm (board only) 11.1 cm × 7.3 cm × 2.4 cm (in acrylic case)
Weight	90 g (board only) 125 g (in acrylic case covered with copper tape)
Approximate price	480 USD (xA4) 720 USD (xA9)
Product page	https://www.nuand.com/bladerf-2-0-micro/

Table 2.14: BladeRF 2.0 micro technical specifications.

2.2.5.1 Receiver Noise Figure

The settings used during the BladeRF 2.0 micro noise figure measurements are listed in Table 2.15 and the results are shown on Figure 2.69 below.

Input rate	Decimation	Sample rate	Bandwidth
2 MHz	4	500 kHz	Auto

Table 2.15: BladeRF 2.0 micro settings used during noise figure measurements.

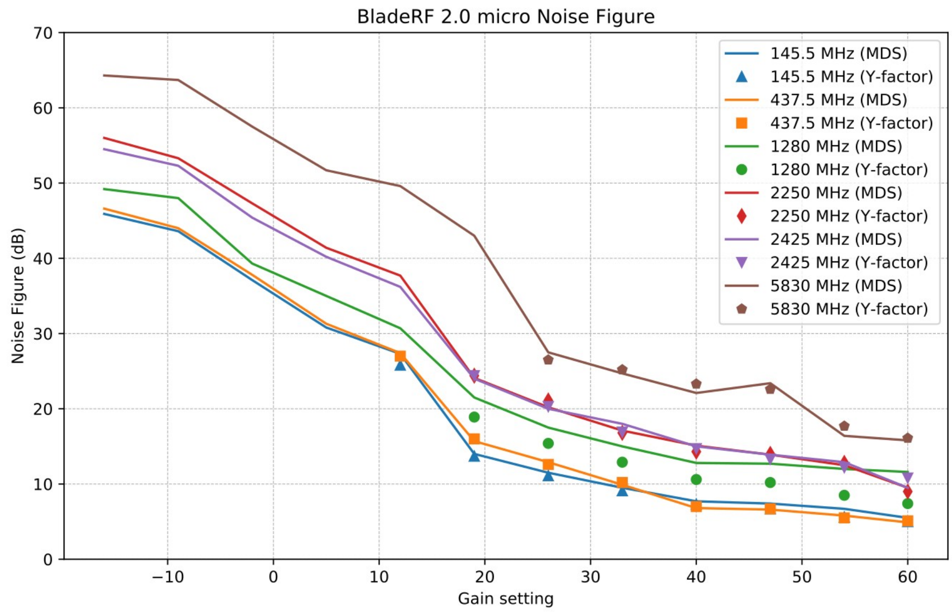


Figure 2.69: BladeRF 2.0 micro noise figure measurements.

We note that:

- There is in general good agreement between the noise figure calculated from the MDS and the noise figure measured using the Y-factor method, except on 1.28 GHz.
- We consider the MDS measurements at 1.28 GHz to be unreliable because the device was picking up a large amount of noise from somewhere while connected to the signal generator.
- The noise level at 5.8 GHz was very sensitive to how the USB connector was positioned while mated.
- Although no RF official performance data is available, the measurements look as we would expect it. Relatively good noise figure on VHF and UHF, and slowly increasing noise figure with increasing frequency.

2.2.5.2 Receiver Blocking Dynamic Range

The settings used during the BladeRF 2.0 micor dynamic range measurements are listed in Table 2.16 and the results are shown on Figure 2.70 below.

Input rate	Decimation	Sample rate	Bandwidth
3 MHz	1	3 MHz	Auto

Table 2.16: BladeRF2.0 micro settings used for BDR measurements.

We note that:

- Ax expected, the highest input power that the device can tolerate without overload, P_{in}, is decreasing as the gain is increased. The relationship is mostly linear.
- The dynamic range is mostly constant over the entire gain spectrum, slightly dropping on some frequencies at the highest gains.
- There is a large (30 dB) variation in the dynamic range as we go from 145 MHz to 5.8 GHz.

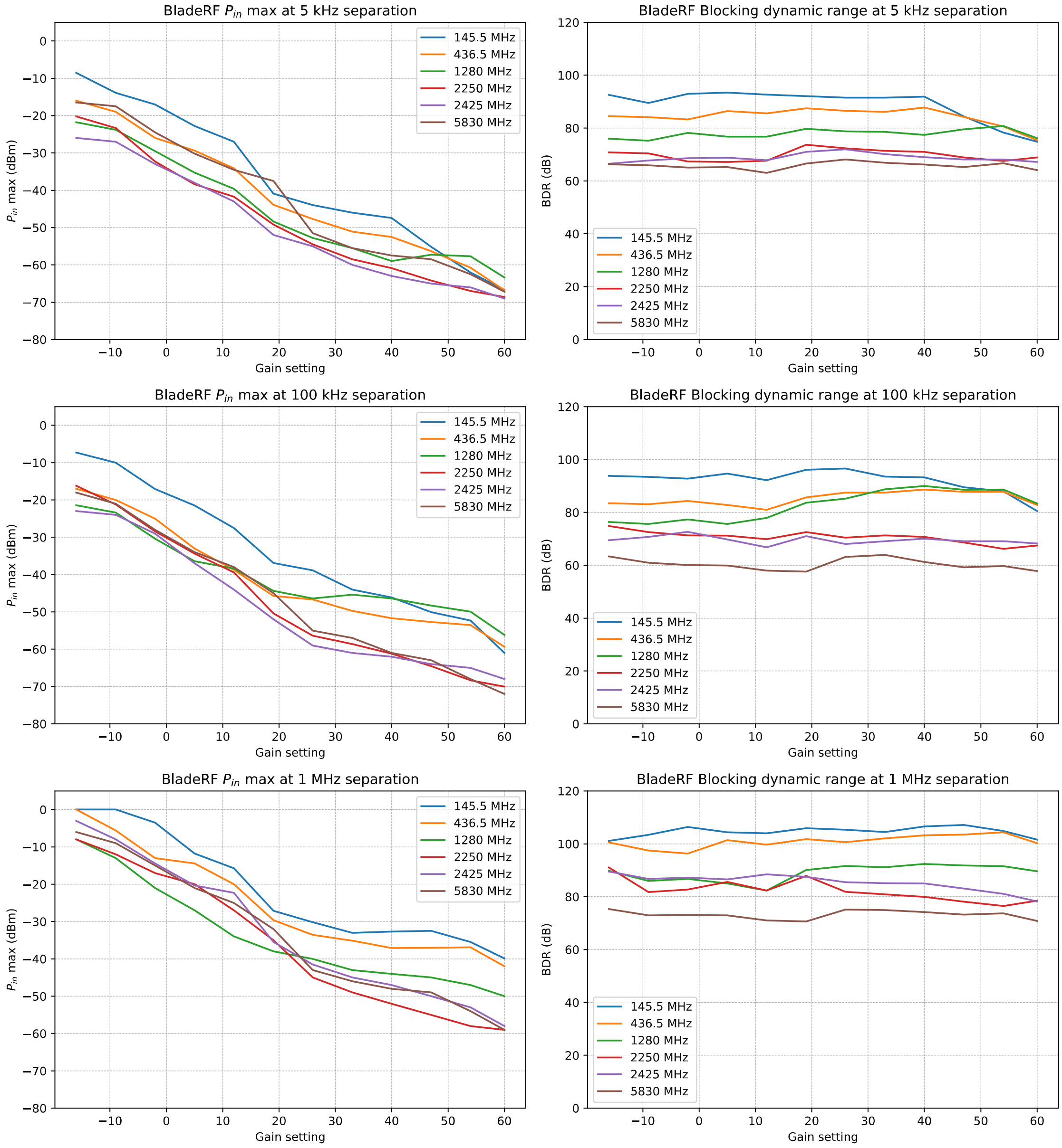


Figure 2.70: BladeRF 2.0 micro blocking dynamic range measurements.

2.2.5.3 Receiver Spectral Purity

The receiver spectra of the BladeRF 2.0 micro receiver are shown on Figures 2.71 to 2.81 below. As we can see from the spectra, there are some spurs generated by or picked up by the device when connected to 50 ohm terminator. We can also note that the device is sensitive to external noise on 145 MHz and 2.4 GHz. This is not surprising as the device comes with an acrylic enclosure.

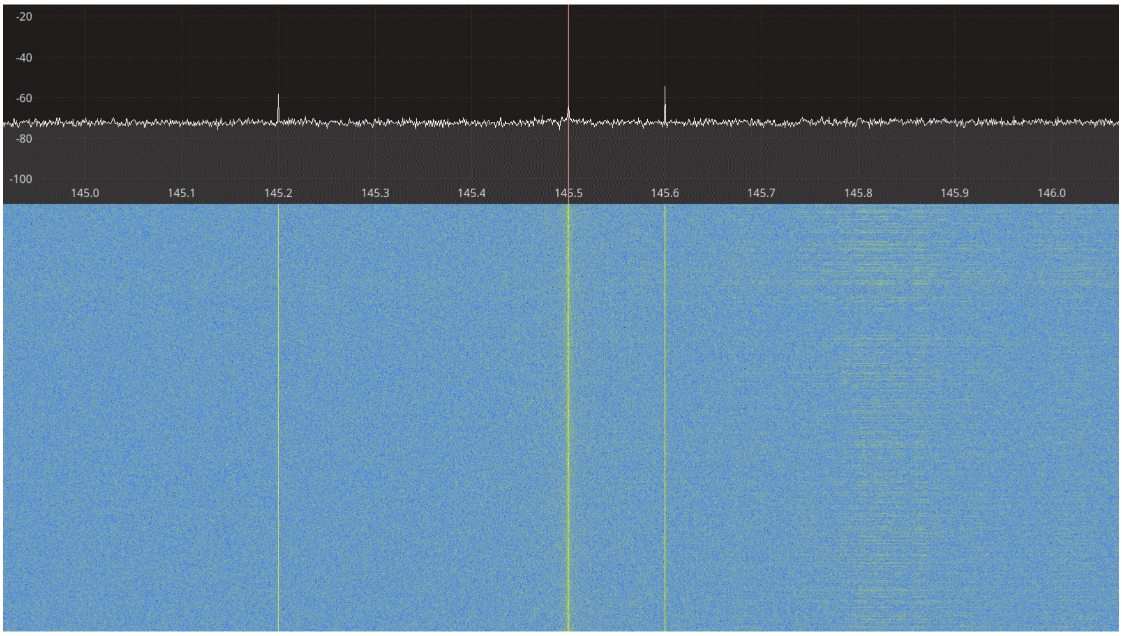


Figure 2.71: BladeRF 2.0 micro receiver spectrum between 145 – 146 MHz.

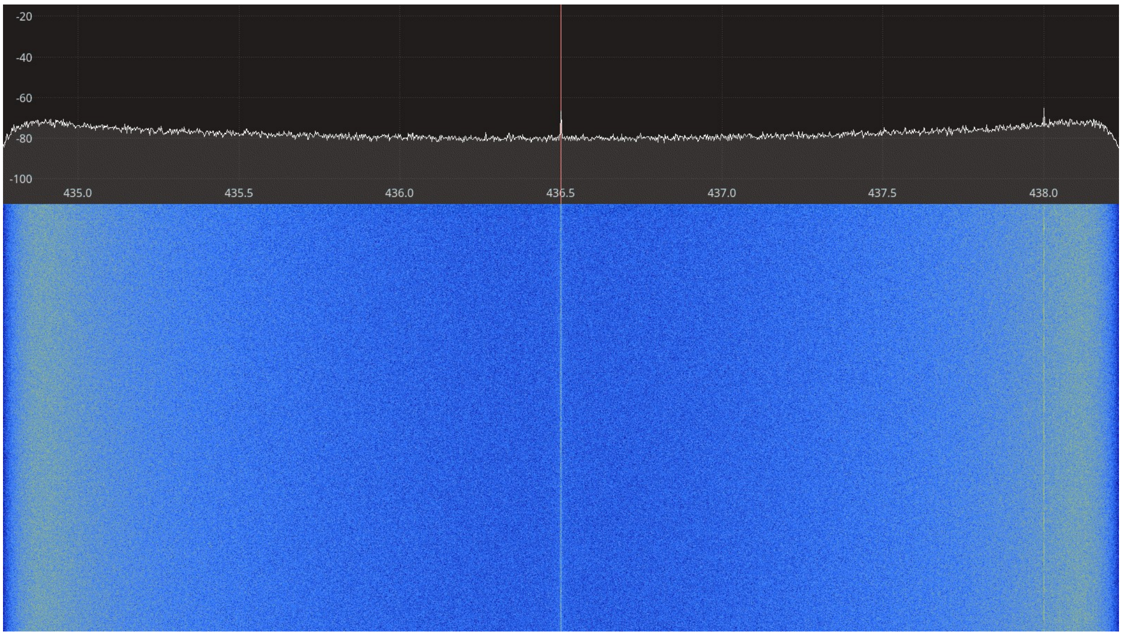


Figure 2.72: BladeRF 2.0 micro receiver spectrum between 435 – 438 MHz.

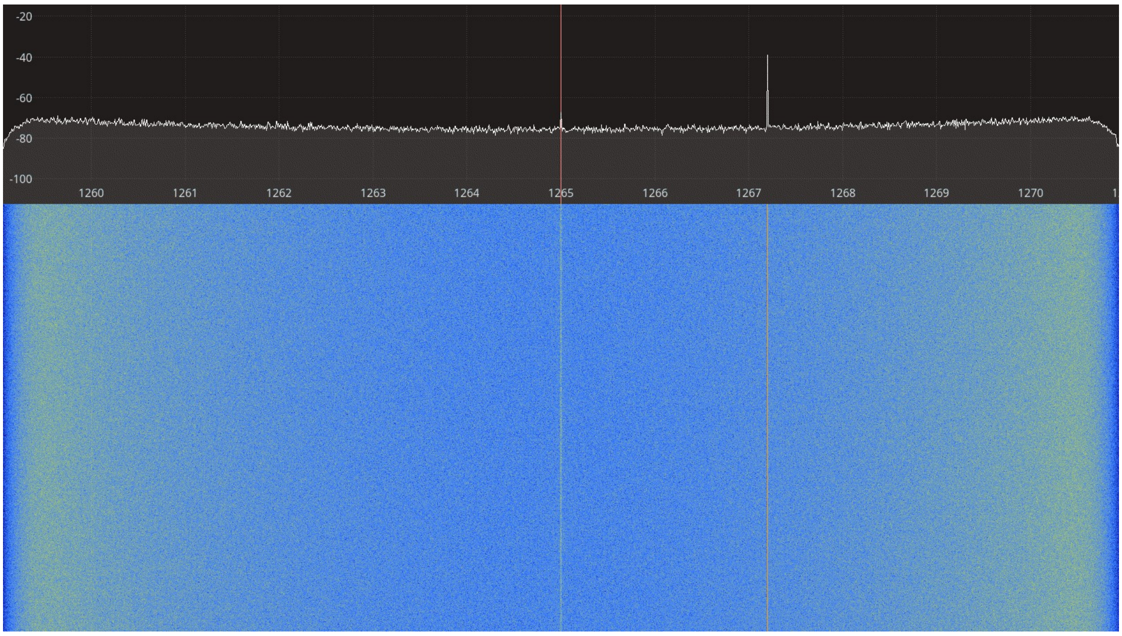


Figure 2.73: BladeRF 2.0 micro receiver spectrum between 1260 – 1270 MHz.

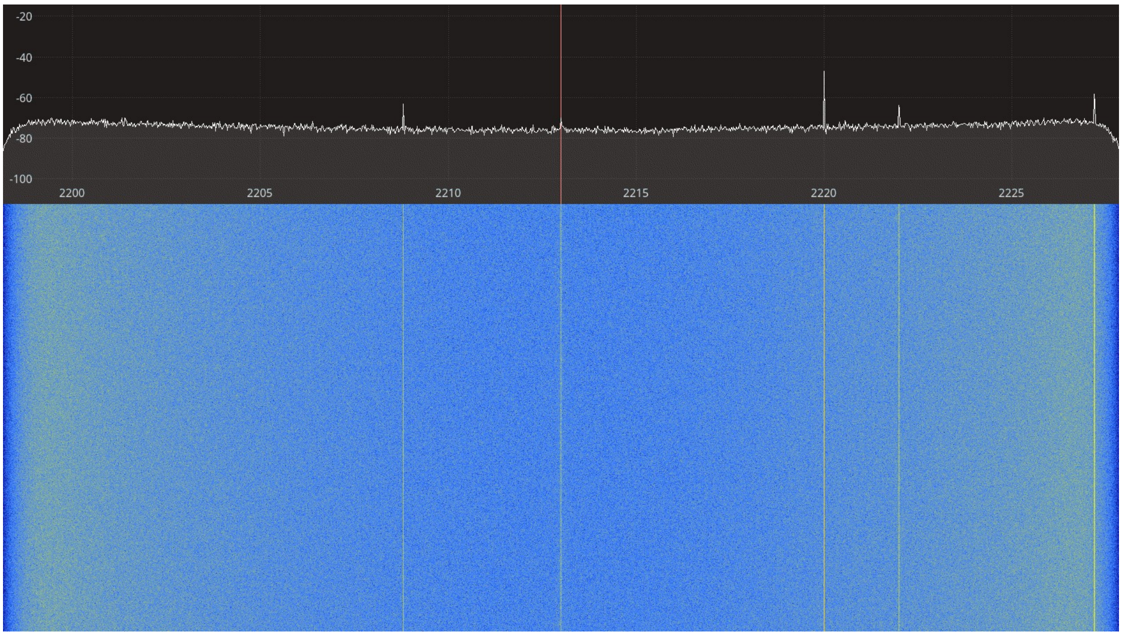


Figure 2.74: BladeRF 2.0 micro receiver spectrum between 2200 – 2225 MHz.

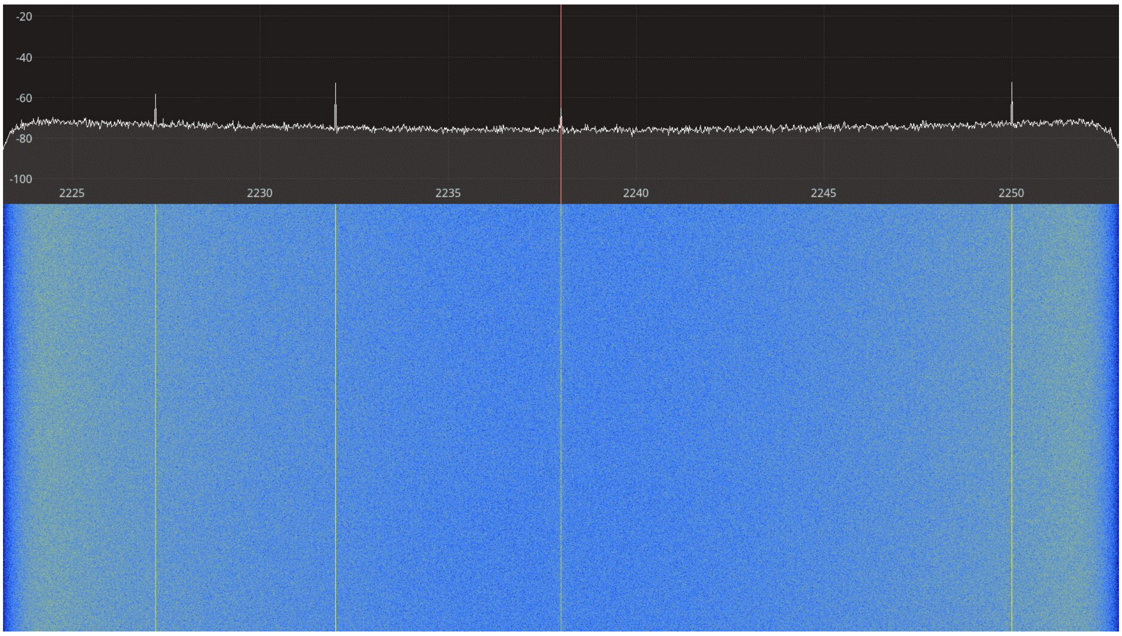


Figure 2.75: BladeRF 2.0 micro receiver spectrum between 2225 – 2250 MHz.

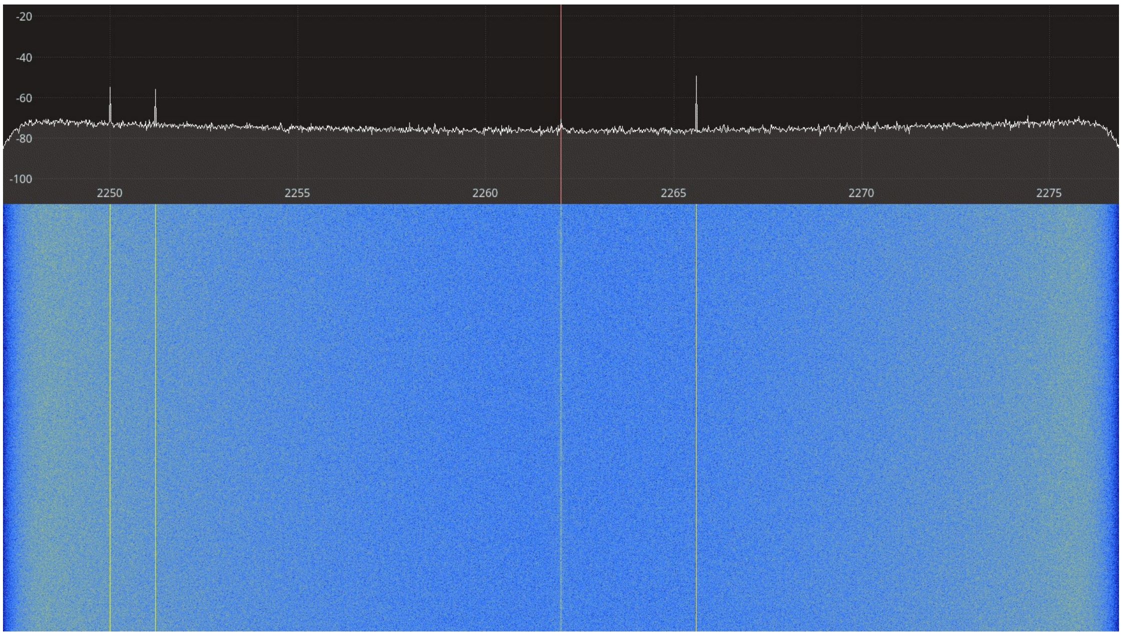


Figure 2.76: BladeRF 2.0 micro receiver spectrum between 2250 – 2275 MHz.

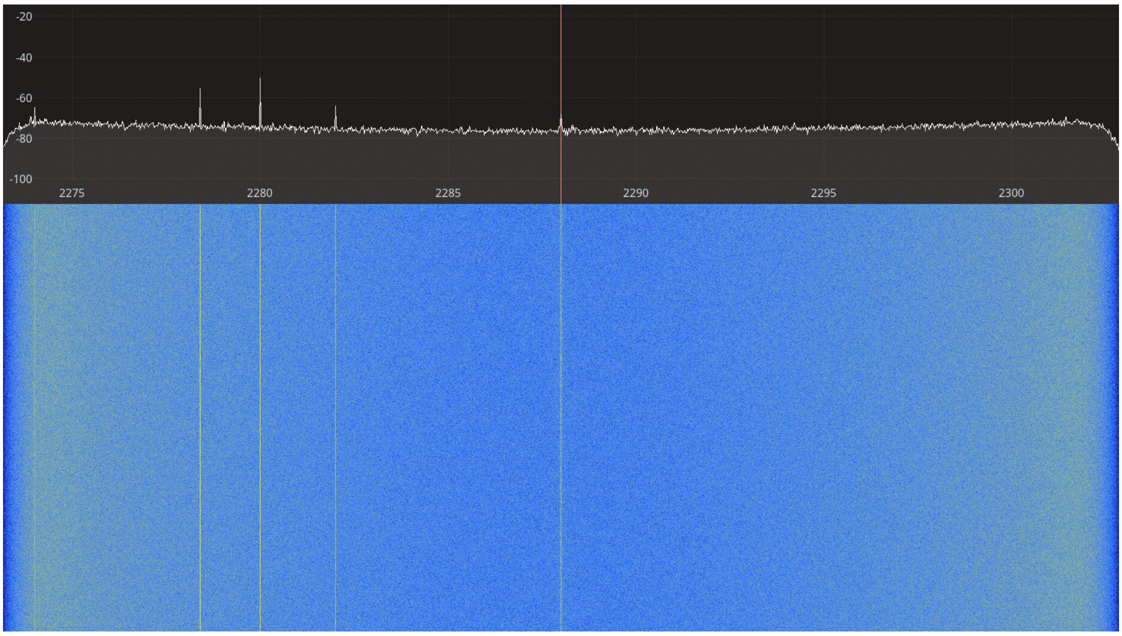


Figure 2.77: BladeRF 2.0 micro receiver spectrum between 2275 – 2300 MHz.

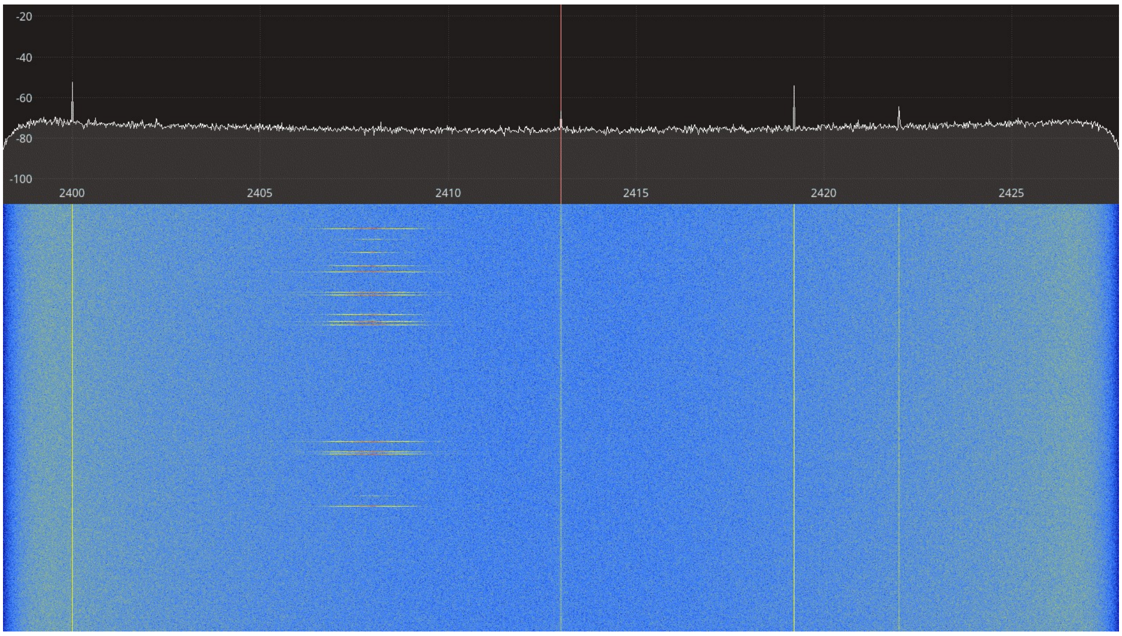


Figure 2.78: BladeRF 2.0 micro receiver spectrum between 2400 – 2425 MHz.

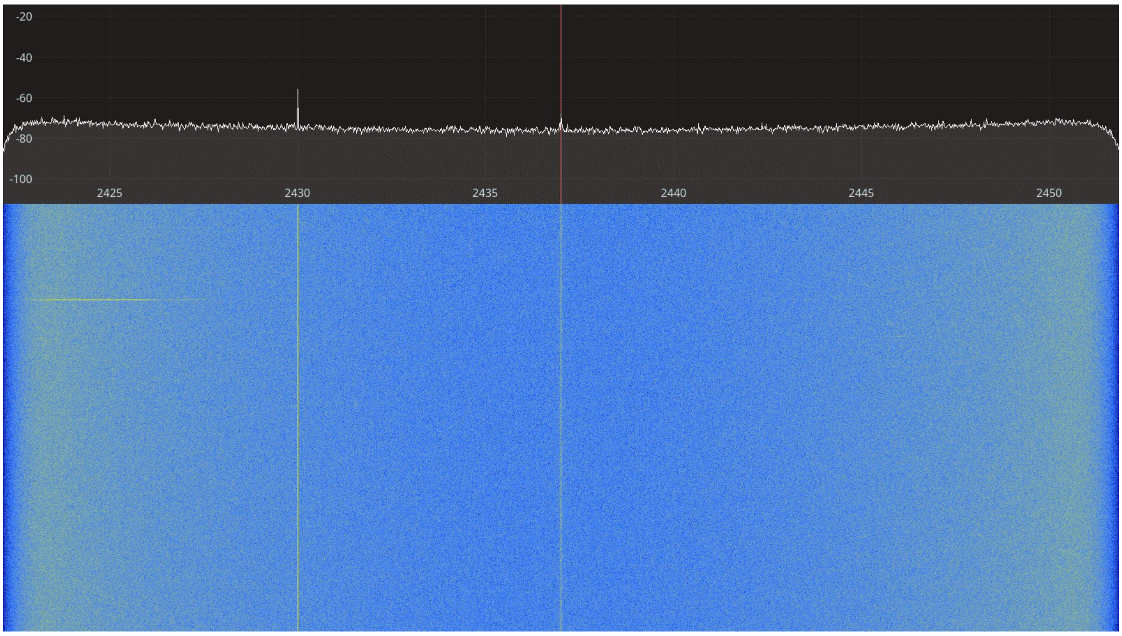


Figure 2.79: BladeRF 2.0 micro receiver spectrum between 2425 – 2450 MHz.

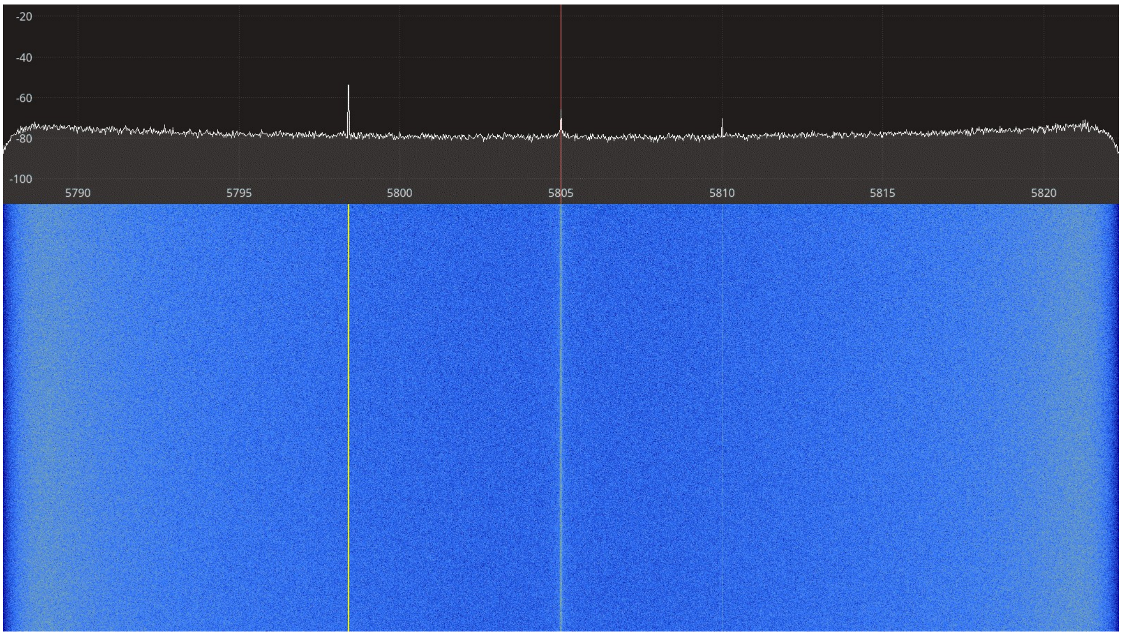


Figure 2.80: BladeRF 2.0 micro receiver spectrum between 5790 – 5820 MHz.

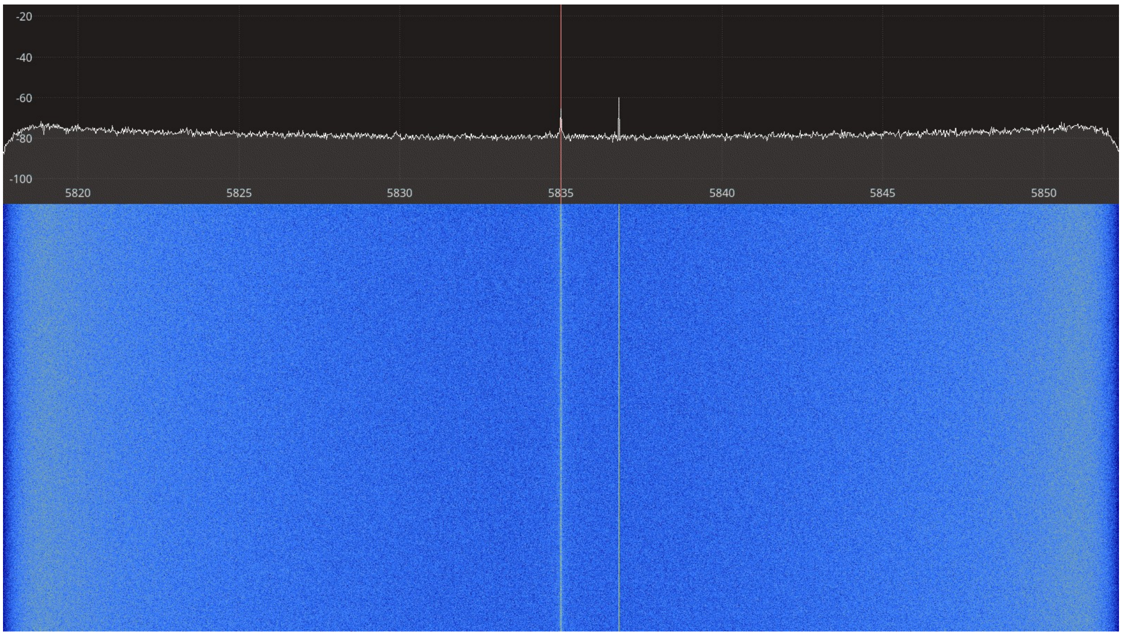


Figure 2.81: BladeRF 2.0 micro receiver spectrum between 5820 – 5850 MHz.

2.2.5.4 Transmitter Spectral Purity

The transmitter spectra for the BladeRF 2.0 micro transmitting a CW signal are shown on Figures 2.82 to 2.93 below. For each frequency, two spectra are shown, one at a gain setting where the spectrum is clean only showing the CW signal, and one at a higher gain setting where spurious artifacts appear on the spectrum.

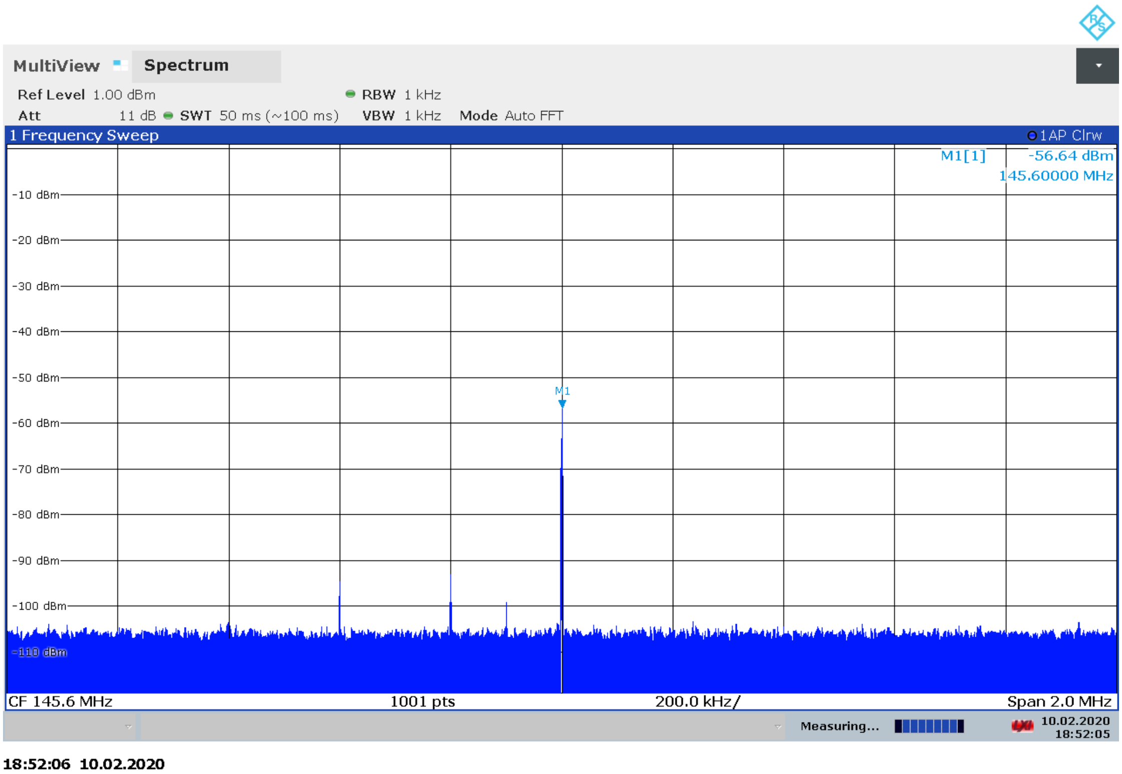


Figure 2.82: BladeRF 2.0 micro transmitter spectrum at 145.5 MHz, gain 0.

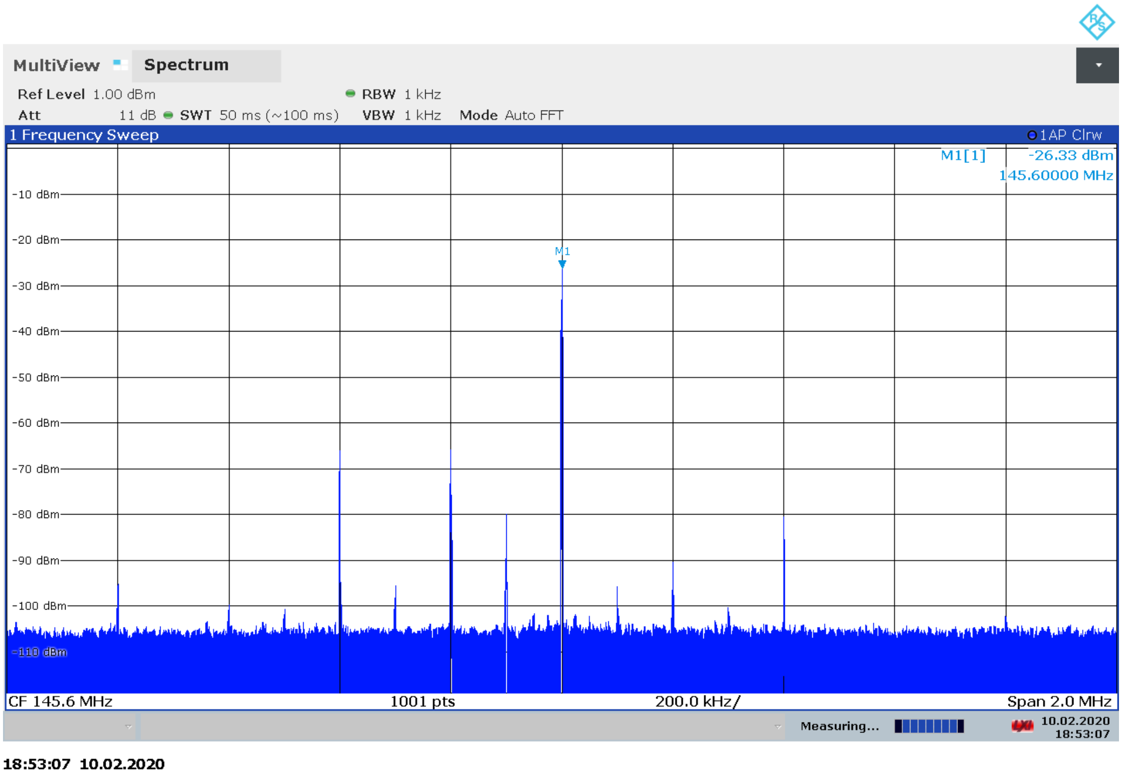


Figure 2.83: BladeRF 2.0 micro transmitter spectrum at 145.5 MHz, gain 30.

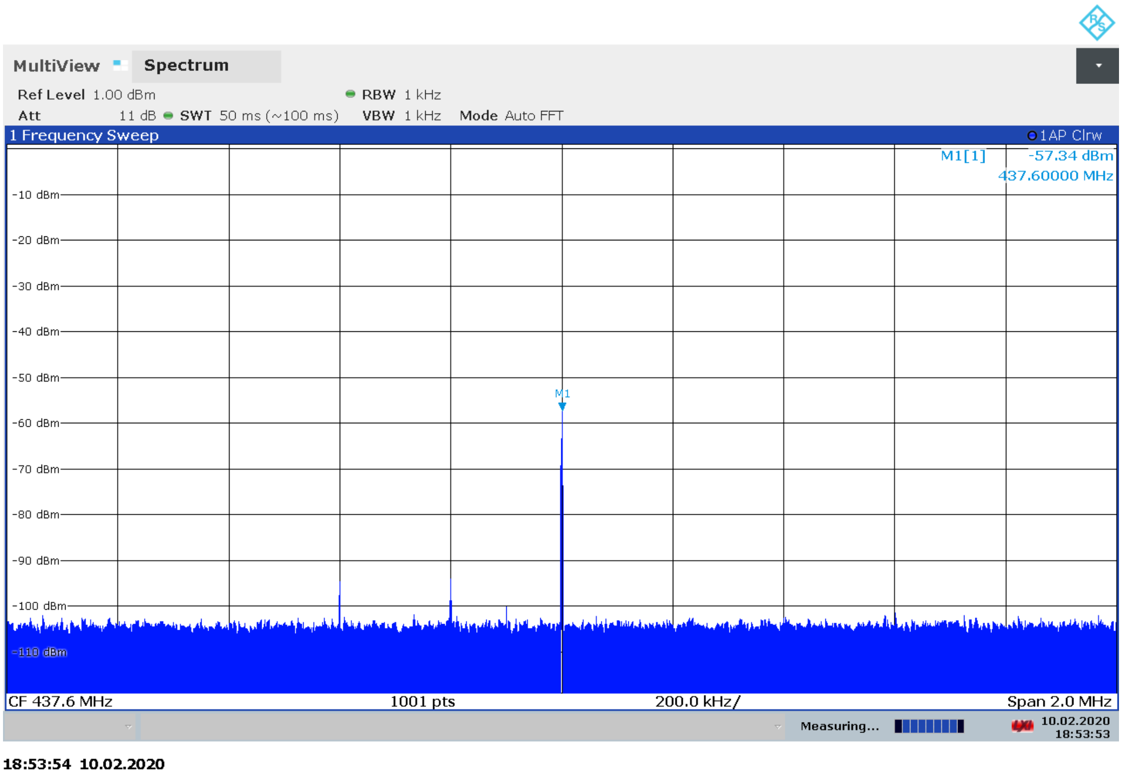


Figure 2.84: BladeRF 2.0 micro transmitter spectrum at 437.5 MHz, gain 0.

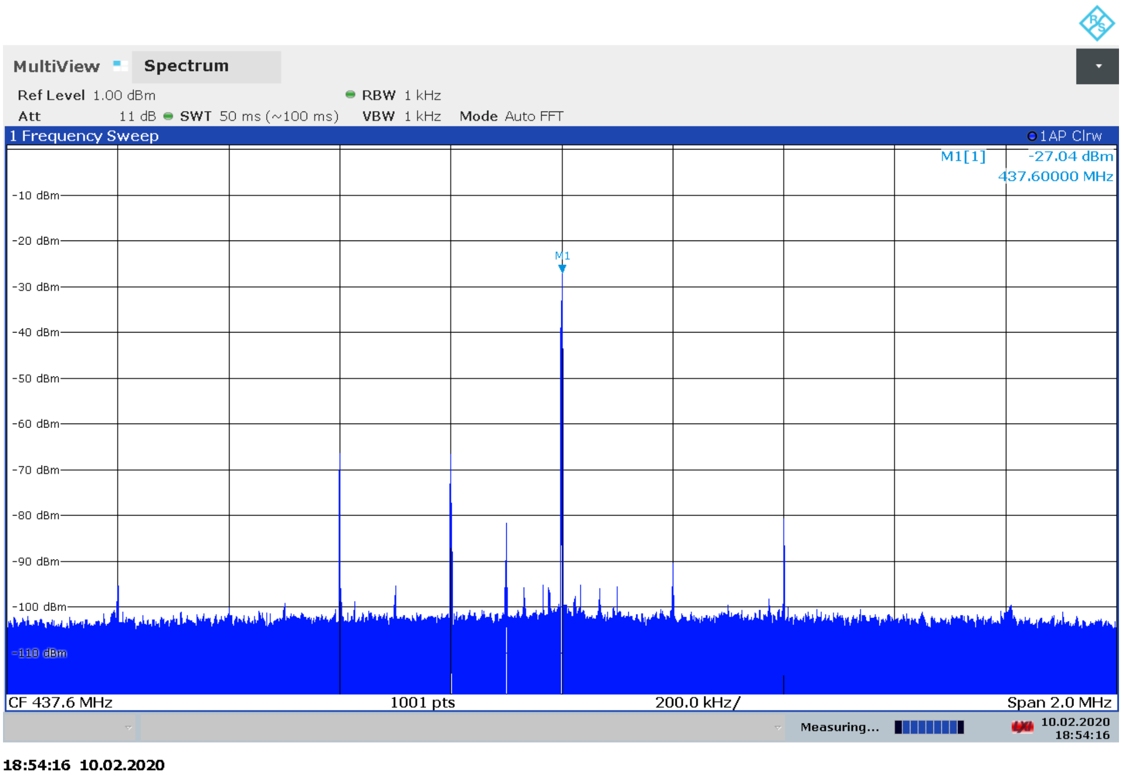


Figure 2.85: BladeRF 2.0 micro transmitter spectrum at 437.5 MHz, gain 30.

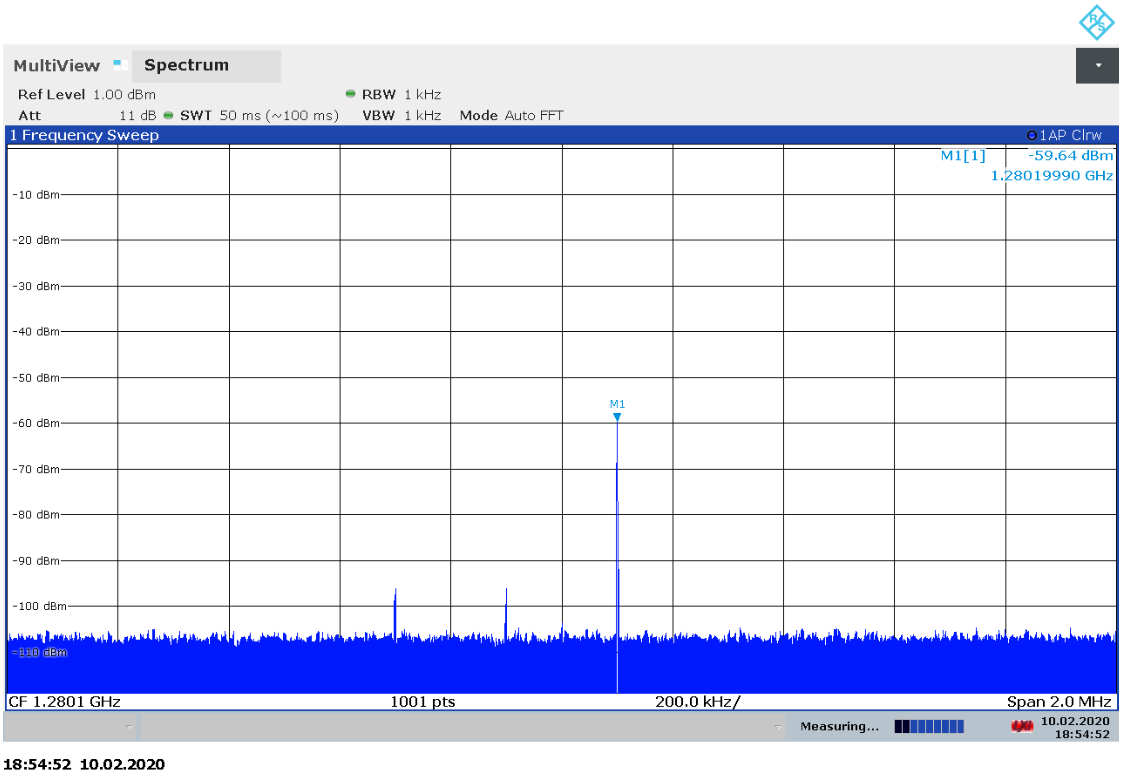


Figure 2.86: BladeRF 2.0 micro transmitter spectrum at 1280 MHz, gain 0.

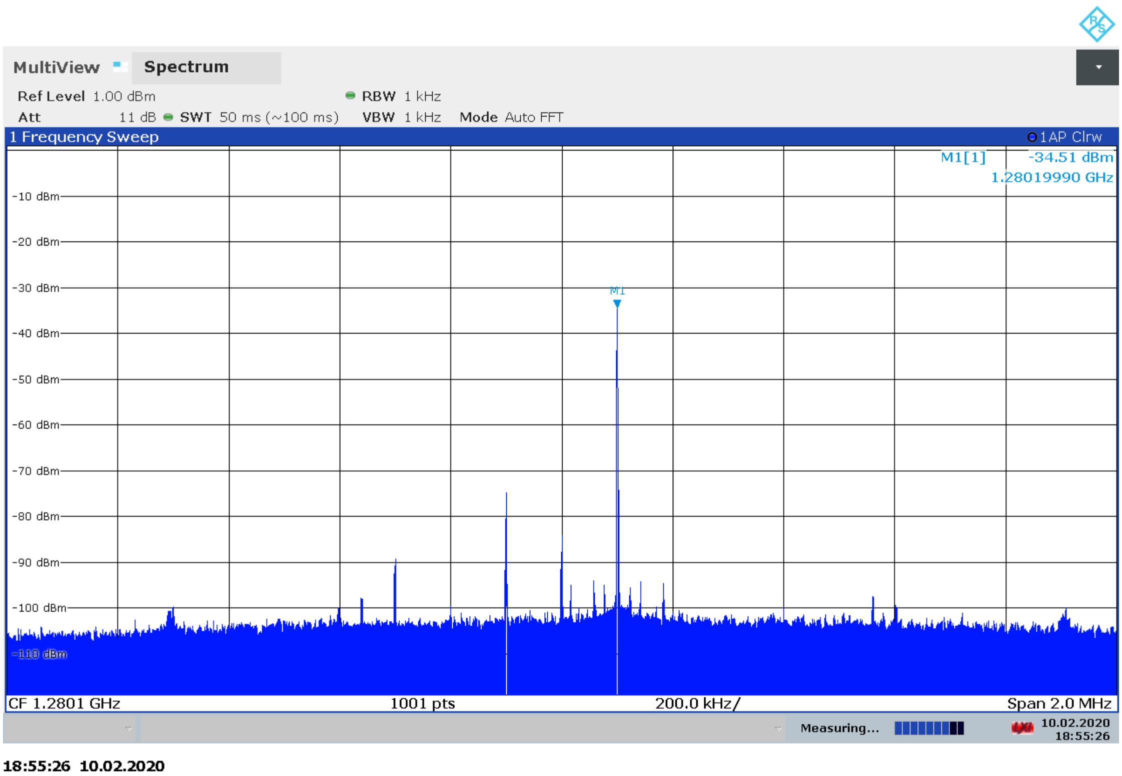


Figure 2.87: BladeRF 2.0 micro transmitter spectrum at 1280 MHz, gain 25.

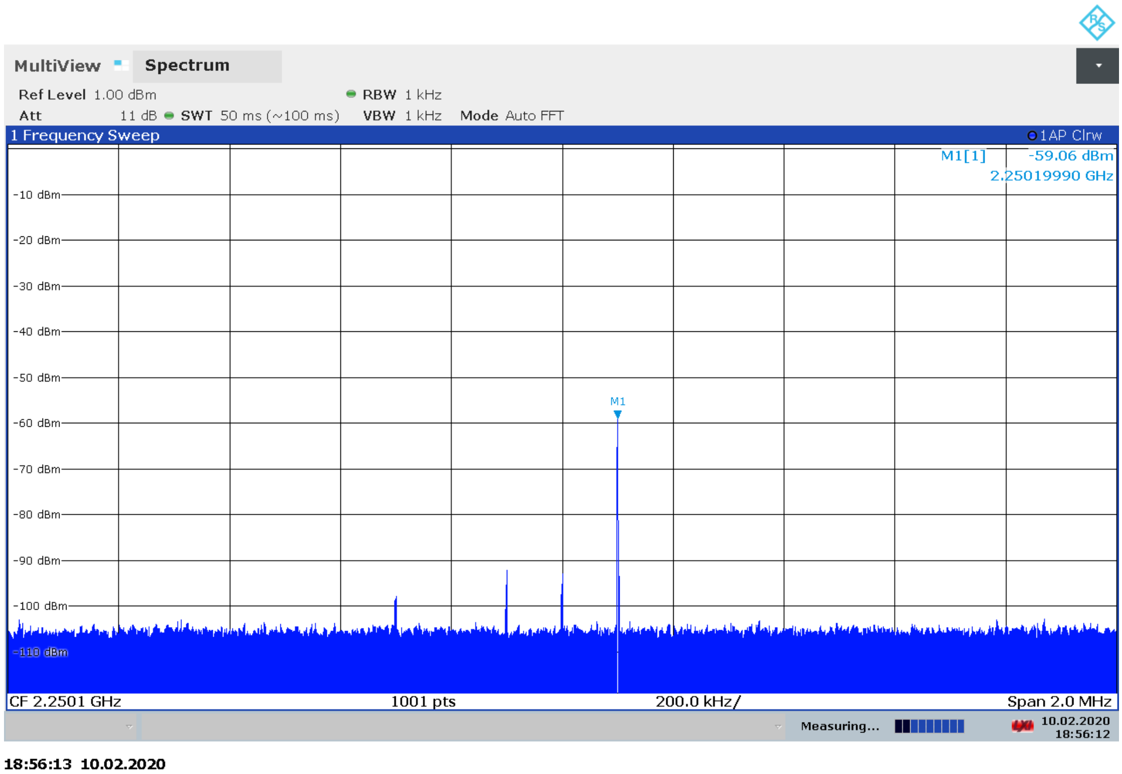


Figure 2.88: BladeRF 2.0 micro transmitter spectrum at 2250 MHz, gain 5.

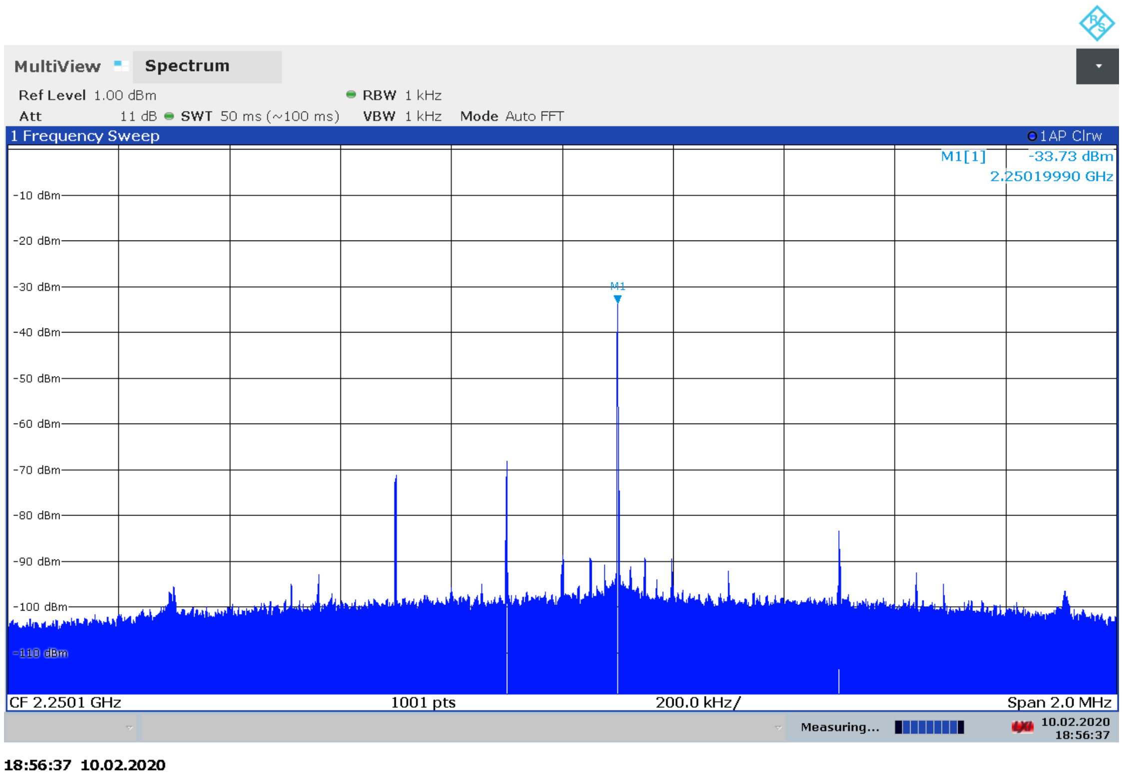


Figure 2.89: BladeRF 2.0 micro transmitter spectrum at 2250 MHz, gain 30.

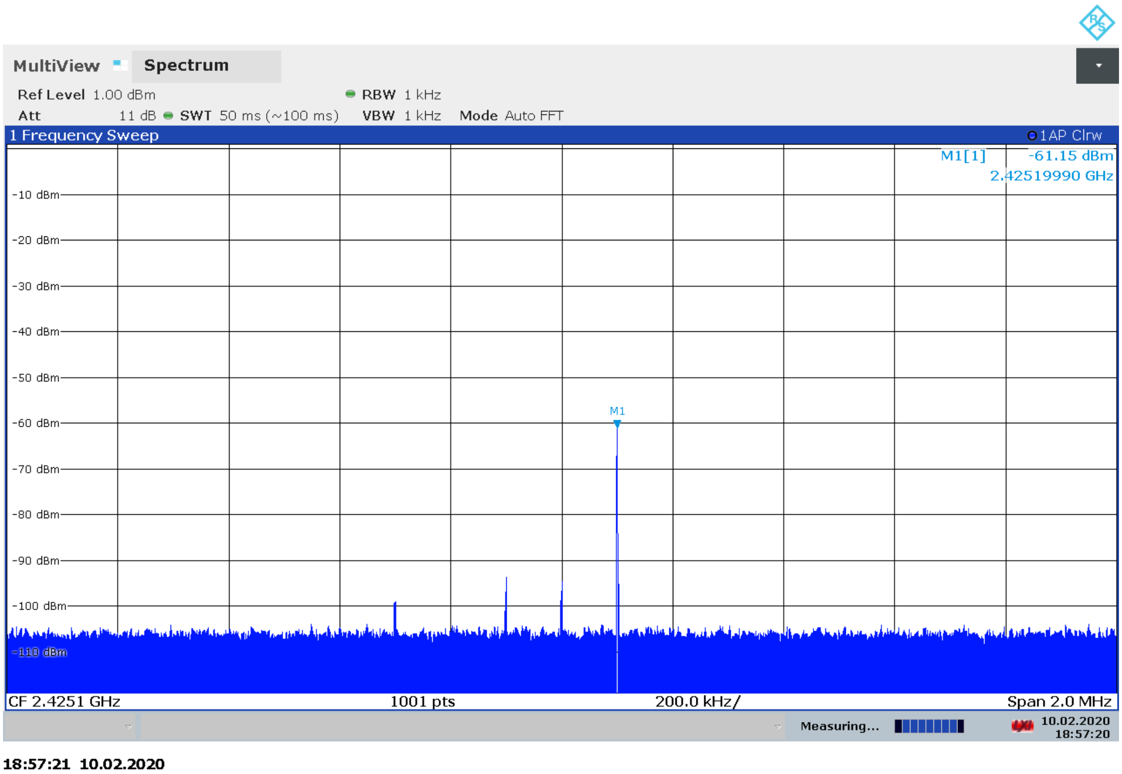


Figure 2.90: BladeRF 2.0 micro transmitter spectrum at 2425 MHz, gain 5.

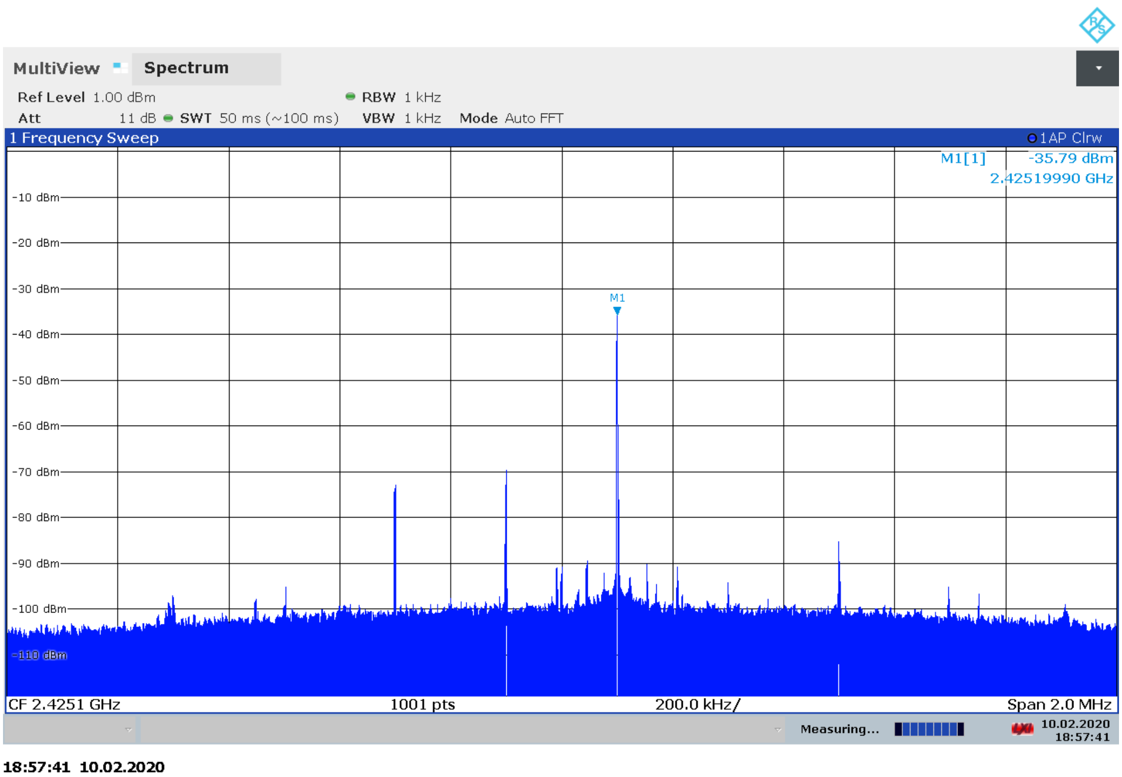


Figure 2.91: BladeRF 2.0 micro transmitter spectrum at 2425 MHz, gain 30.

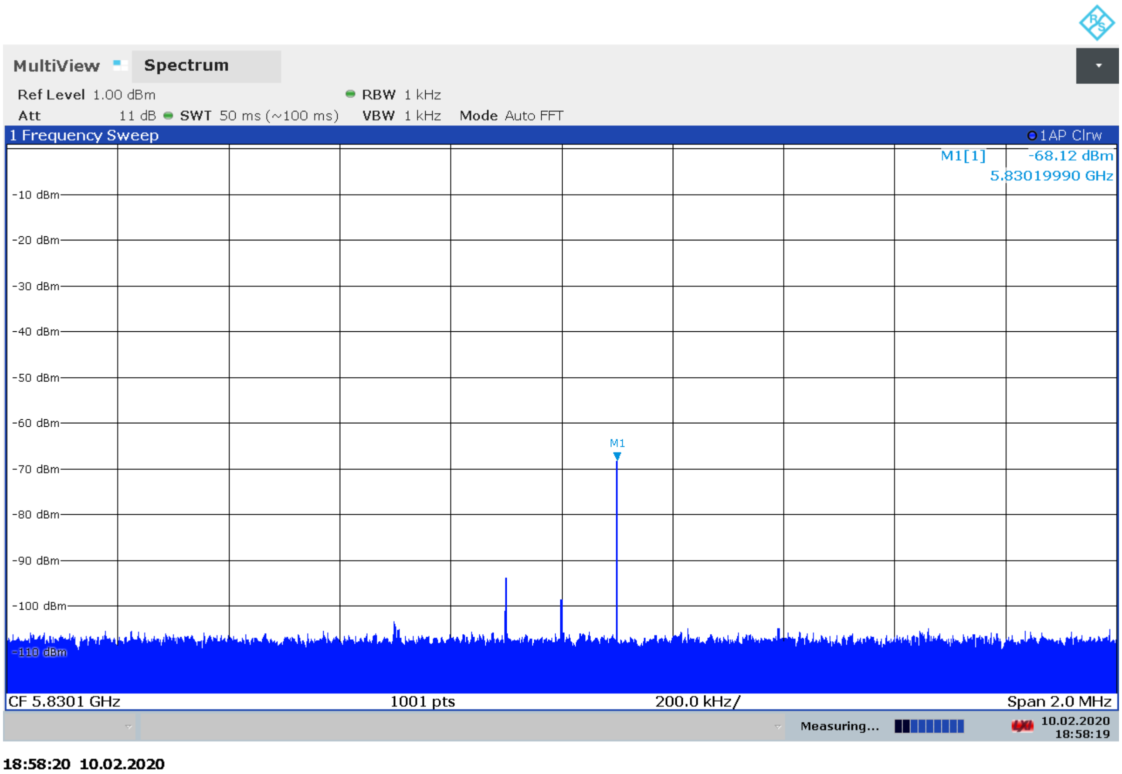


Figure 2.92: BladeRF 2.0 micro transmitter spectrum at 5830 MHz, gain 5.

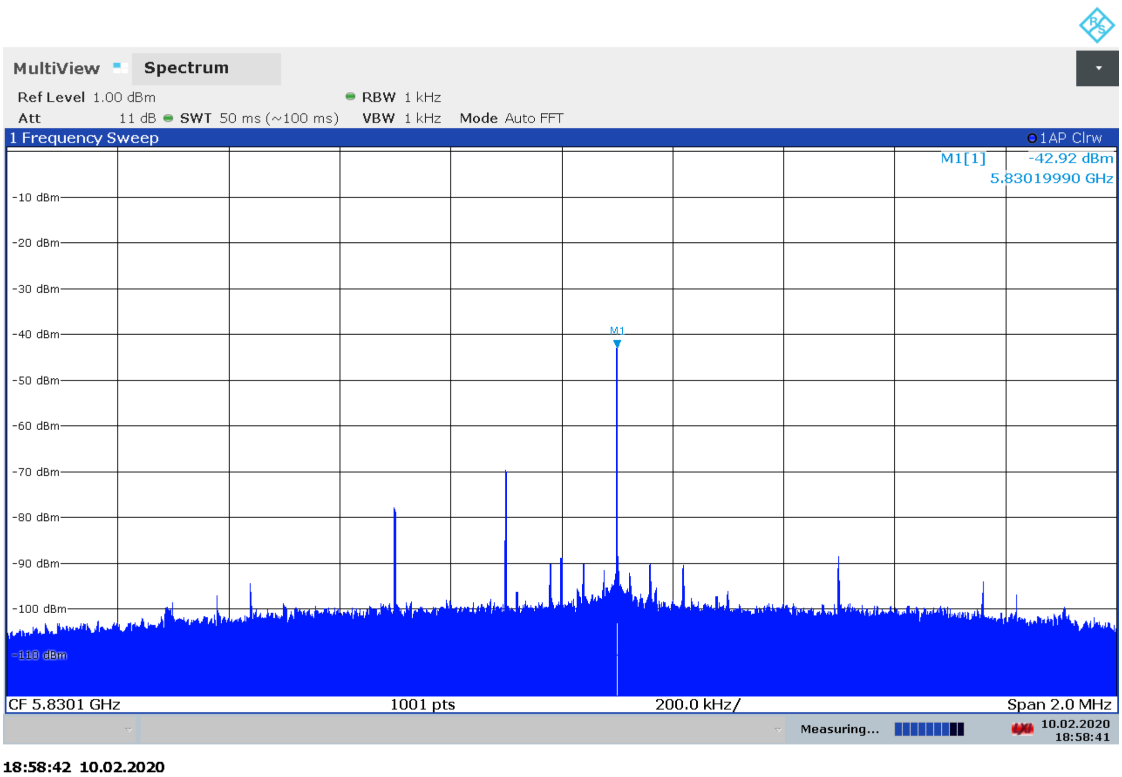


Figure 2.93: BladeRF 2.0 micro transmitter spectrum at 5830 MHz, gain 30.

2.2.5.5 Transmitter Output Power

The measured transmitter output for the BladeRF 2.0 micro is shown on Figure 2.94 below. As we can see there is a linear relationship between the transmitter gain setting and the output power.

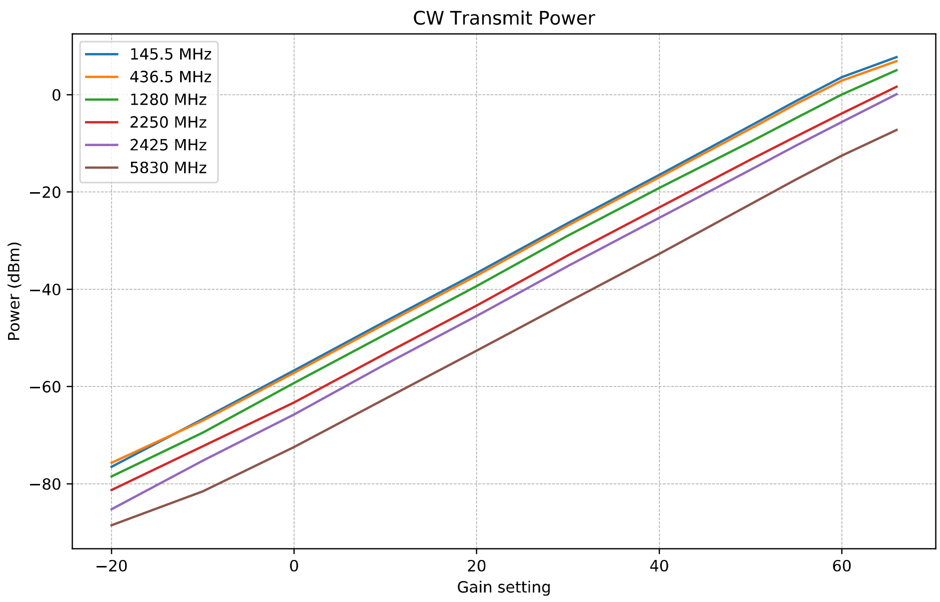


Figure 2.94: BladeRF 2.0 micro CW transmitter output power as function of gain.

2.2.5.6 Transmitter MER

The measured output power and modulation quality for a DVB-S2 signal transmitted with the BladeRF 2.0 micro are shown in Table 2.17 below. As we noted in test procedure description in section 2.1.6, we believe that the NsMAR parameter is a margin above the required E_S/N_0, which is 11 dB for the modulation and FEC settings used in our tests. Also note that the saturation in NsMAR and MER is due to the limitation of the signal analyzer and does not necessarily represent a limitation in the SDR device.

TX Gain	Power (dBm)	NsMAR (dB)	MER (dB)
0	-66.1	2.5	14.8
10	-56.6	11.5	24.1
20	-50.3	12.7	25.0
30	-37.4	12.7	25.0

TX Gain	Power (dBm)	NsMAR (dB)	MER (dB)
40	-27.0	12.7	25.0
50	-17.9	12.7	25.0

Table 2.17: BladeRF 2.0 micro transmitter MER as function of gain.

2.2.6 Ettus USRP B210

The The USRP B210 from Ettus Research is a 2×2 MIMO transceiver based on the AD9361 integrated RFIC from Analog Devices. It covers 70 MHz – 6 GHz with up to 61 MHz sample rate and 56 MHz analog bandwidth [25].

The wide bandwidth and frequency range make the USRP B210 an interesting choice for satellite communications, both as a receiver and as a transmitter.

USRP products are supported through an open-source Universal Hardware Driver (UHD) library, which provides a relatively simple and intuitive programming interface to the hardware. Firmware and FPGA sources are available as well [26].

On the hardware side, the schematics are available in PDF format on the website [27].

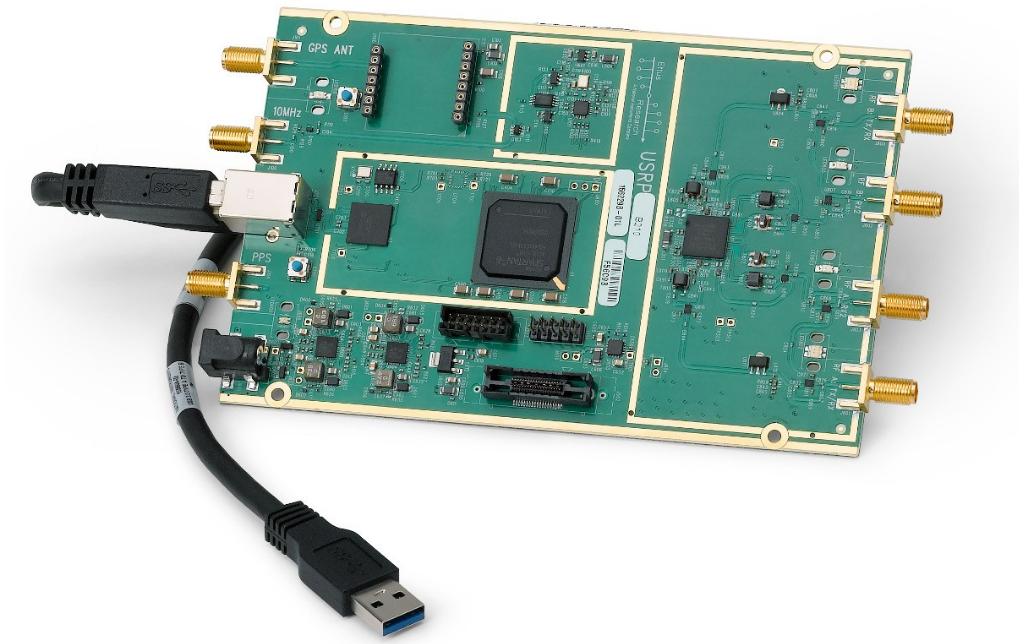


Figure 2.95: The Ettus USRP B210 transceiver.

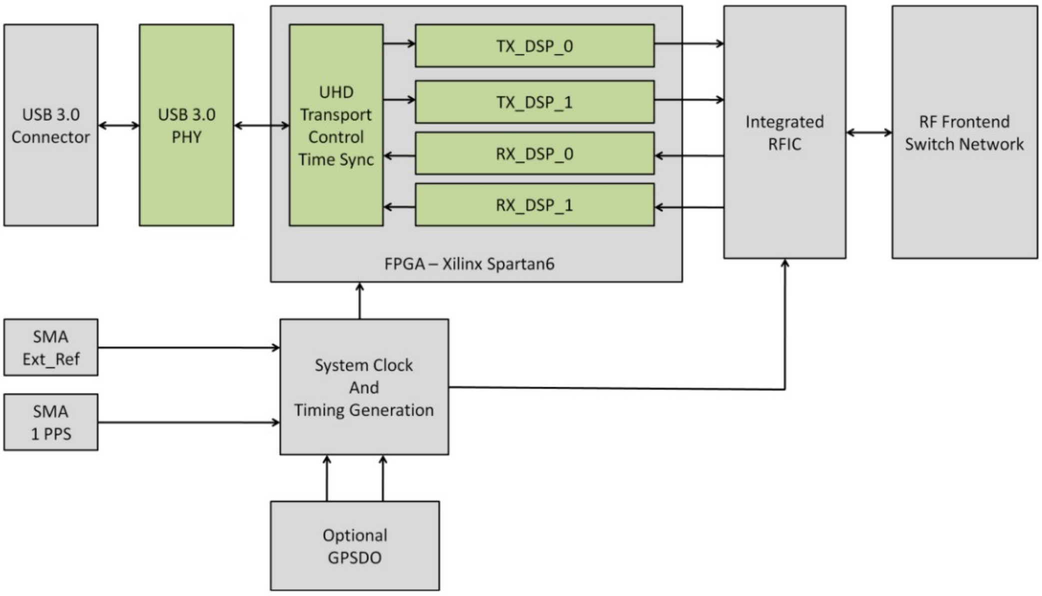


Figure 2.96: USRP B210 block diagram.

A summary of the technical specifications for the Ettus USRP B210 is listed in Table 2.18 below.

Frequency Range	70 MHz – 6 GHz
Sample rate	61.44 MSPS
RF bandwidth	200 kHz – 56 MHz
Frontend filters	None
RX paths	2
RX inputs	2
TX paths	2
TX outputs	2
ADC resolution	12
DAC resolution	12
Claimed noise figure	8 dB
Claimed dynamic range	-
Claimed transmit power	10 dBm
Reference clock	2 ppm
Other features	Automatic IQ and DC offset correction 2×2 MIMO External clock Optional GPSDO
Temperature range	25 °C [28]
Size	15.5 cm × 9.7 cm × 1.5 cm
Weight	350 g (board only)
Approximate price	1200 € (board only)
Product page	https://www.ettus.com/all-products/ub210-kit/

Table 2.18: Ettus USRP B210 technical specifications.

2.2.6.1 Receiver Noise Figure

The settings used during the USRP B210 noise figure measurements are listed in Table 2.19 and the results are shown on Figure 2.97 below.

Input rate	Decimation	Sample rate	Bandwidth
3 MHz	4	750 kHz	Auto

Table 2.19: USRP B210 settings used during noise figure measurements.

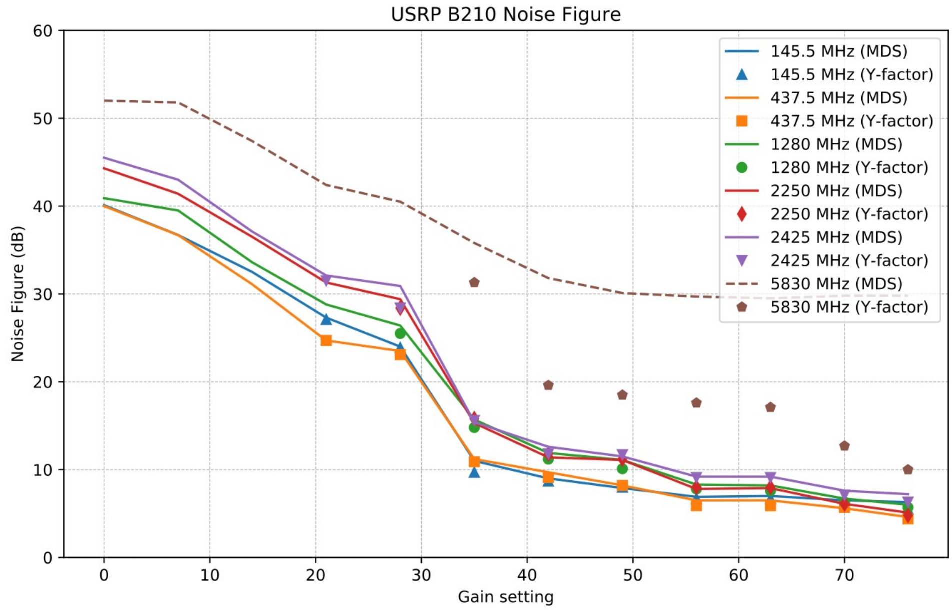


Figure 2.97: USRP B210 noise figure measurements.

We note that:

- On 5.83 GHz, the USRP was picking up leaked signal from the signal generator making MDS measurements on this frequency practically impossible at high gains. Therefore, the MDS data for 5.83 GHz is not reliable.
- With the exception of 5.83 GHz, our noise figure measurements are in good agreement with the B200 performance data available from Ettus Research [29].

2.2.6.2 Receiver Blocking Dynamic Range

The settings used during the USRP B210 dynamic range measurements are listed in Table 2.20 and the results are shown on Figure 2.98 below.

Input rate	Decimation	Sample rate	Bandwidth
3 MHz	1	3 MHz	Auto

Table 2.20: USRP B210 settings used for BDR measurements.

We note that:

- Ax expected, the highest input power that the device can tolerate without overload, P_{in}, is decreasing as the gain is increased. The relationship is mostly linear.

- The dynamic range is mostly constant over the entire gain spectrum, slightly dropping on some frequencies at the highest gains.
- There is a large (30 dB) variation in the dynamic range as we go from 145 MHz to 5.8 GHz.

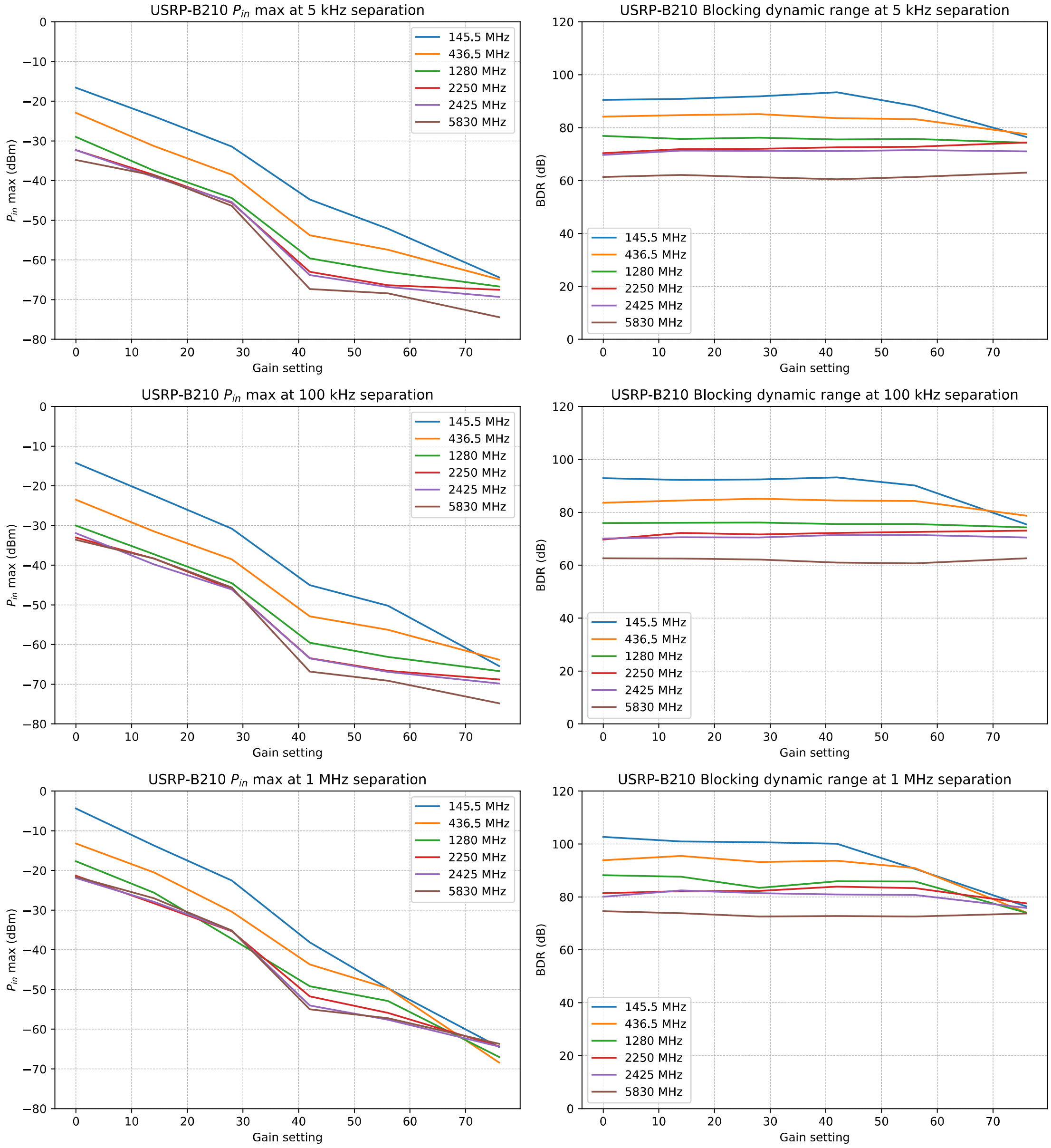


Figure 2.98: USRP B210 blocking dynamic range measurements.

2.2.6.3 Receiver Spectral Purity

The receiver spectra of the USRP B210 receiver are shown on Figures 2.99 to 2.109 below. As we can see from the spectra, there are very few spurs generated by or picked up by the device when connected to 50 ohm terminator.

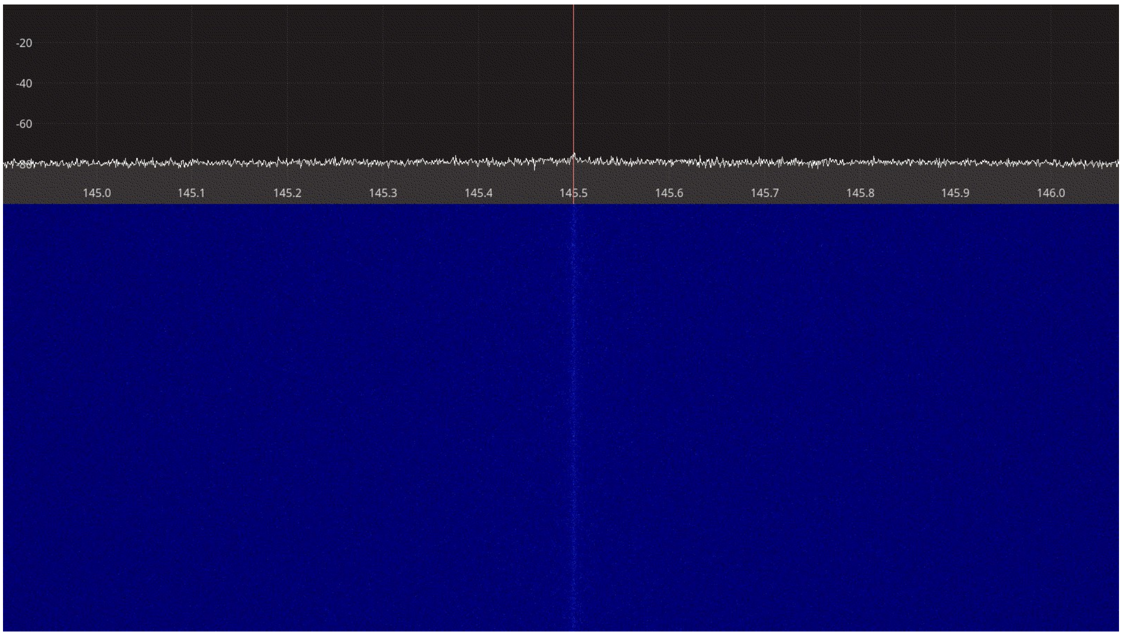


Figure 2.99: Ettus USRP B210 receiver spectrum between 145 – 146 MHz.

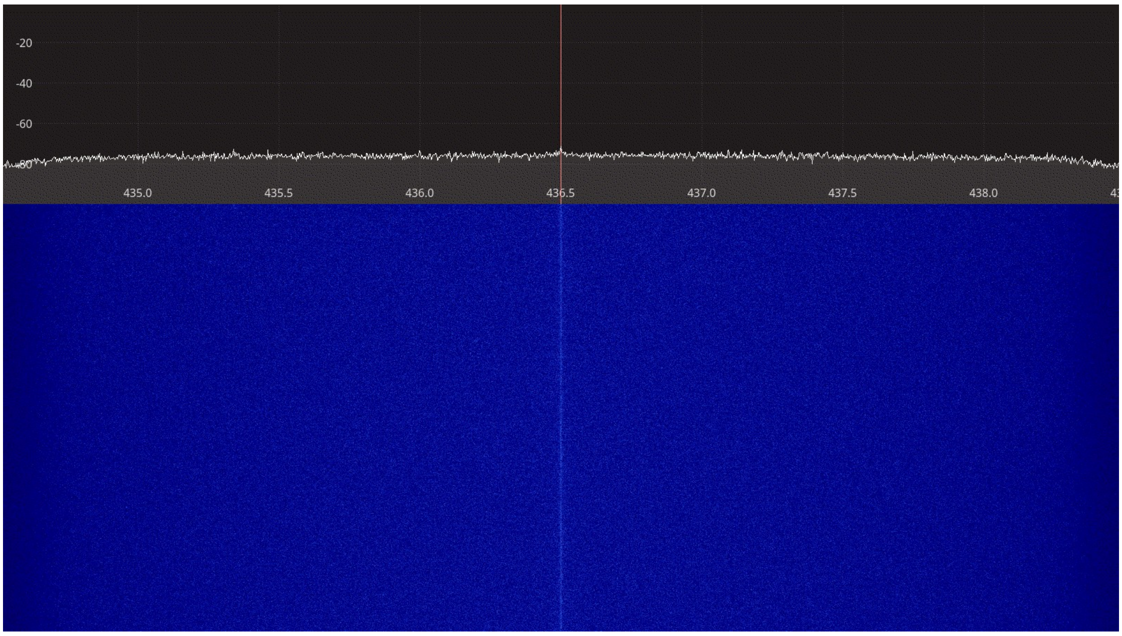


Figure 2.100: Ettus USRP B210 receiver spectrum between 435 – 438 MHz.

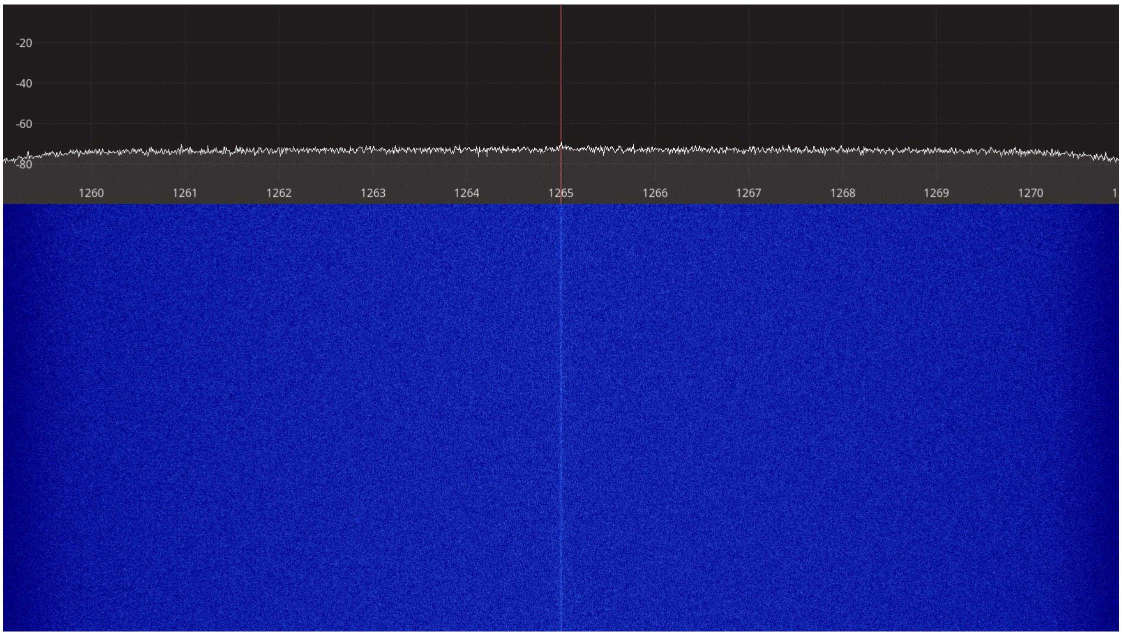


Figure 2.101: Ettus USRP B210 receiver spectrum between 1260 – 1270 MHz.

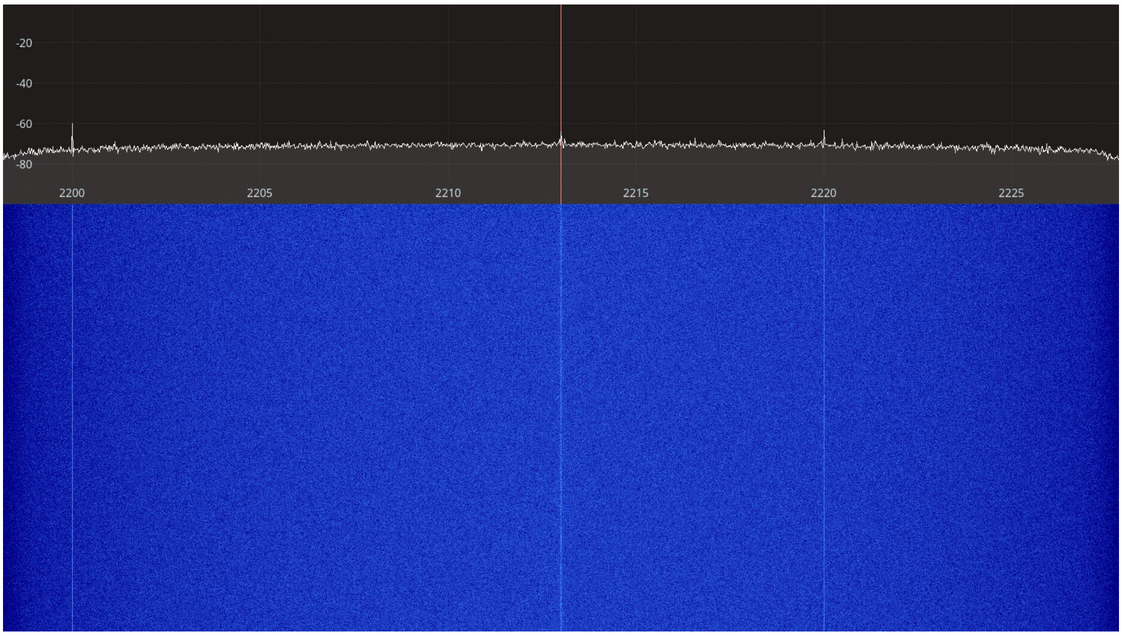


Figure 2.102: Ettus USRP B210 receiver spectrum between 2200 – 2225 MHz.

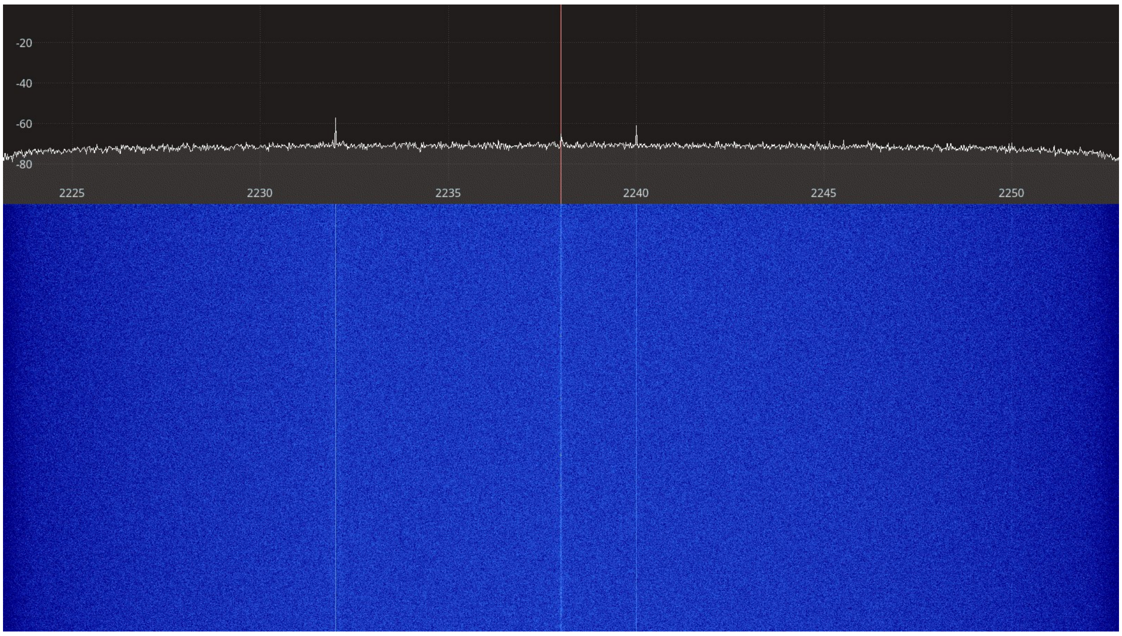


Figure 2.103: Ettus USRP B210 receiver spectrum between 2225 – 2250 MHz.

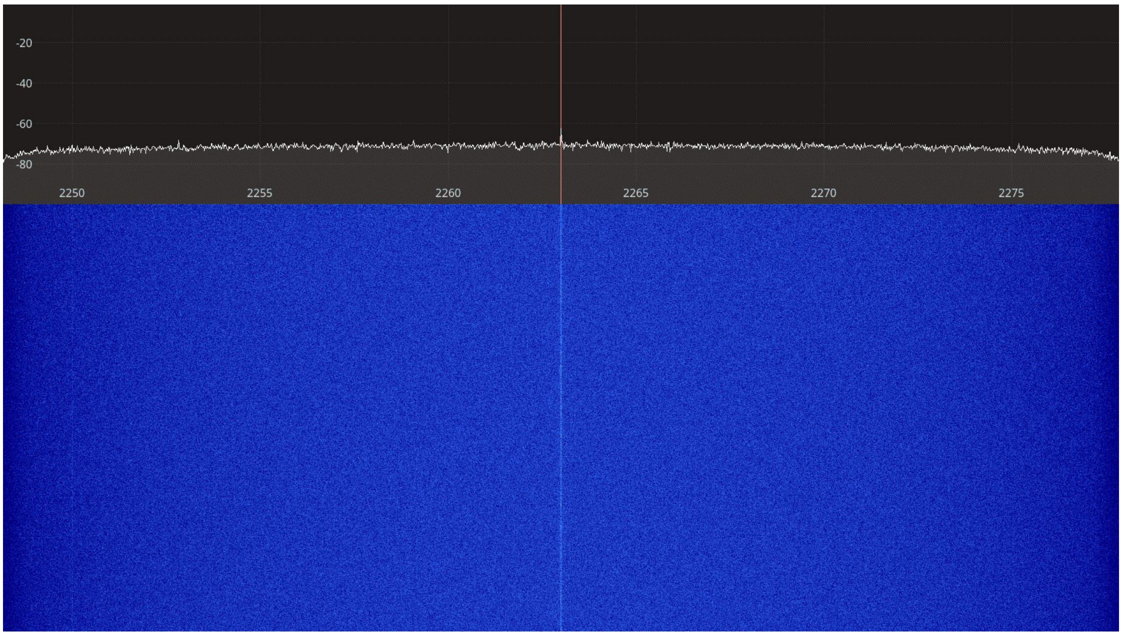


Figure 2.104: Ettus USRP B210 receiver spectrum between 2250 – 2275 MHz.

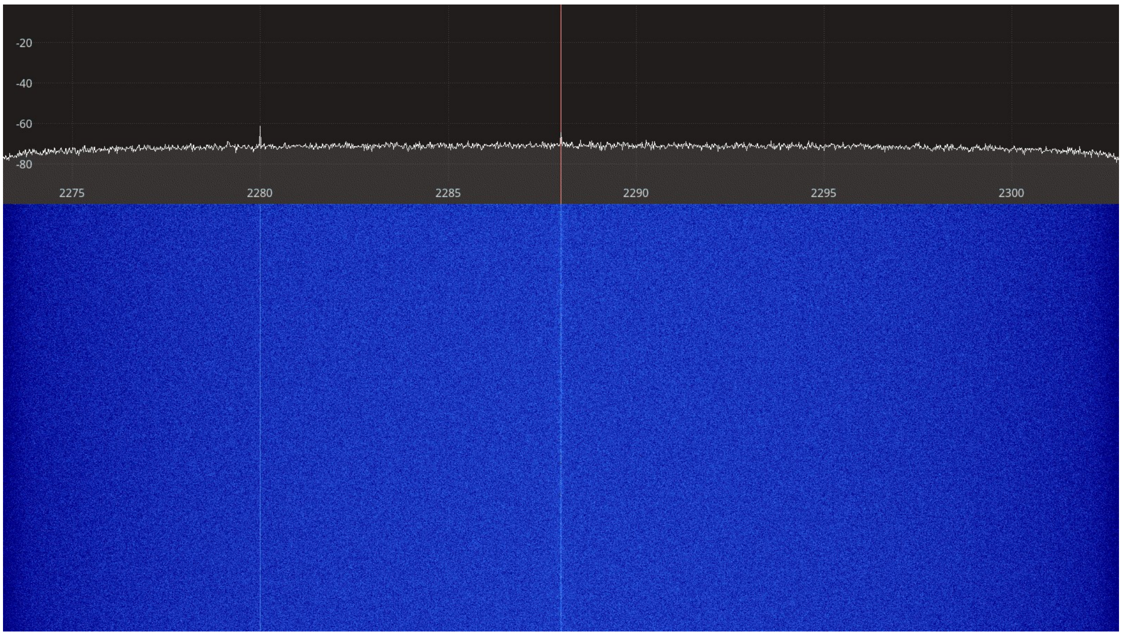


Figure 2.105: Ettus USRP B210 receiver spectrum between 2275 – 2300 MHz.

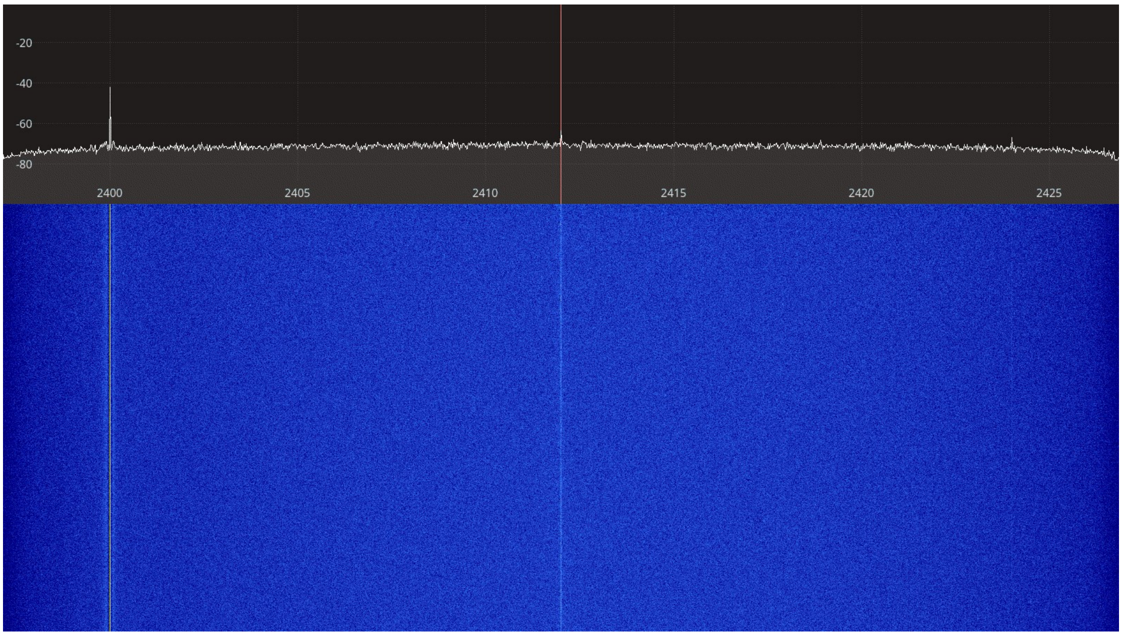


Figure 2.106: Ettus USRP B210 spectrum between 2400 – 2425 MHz.

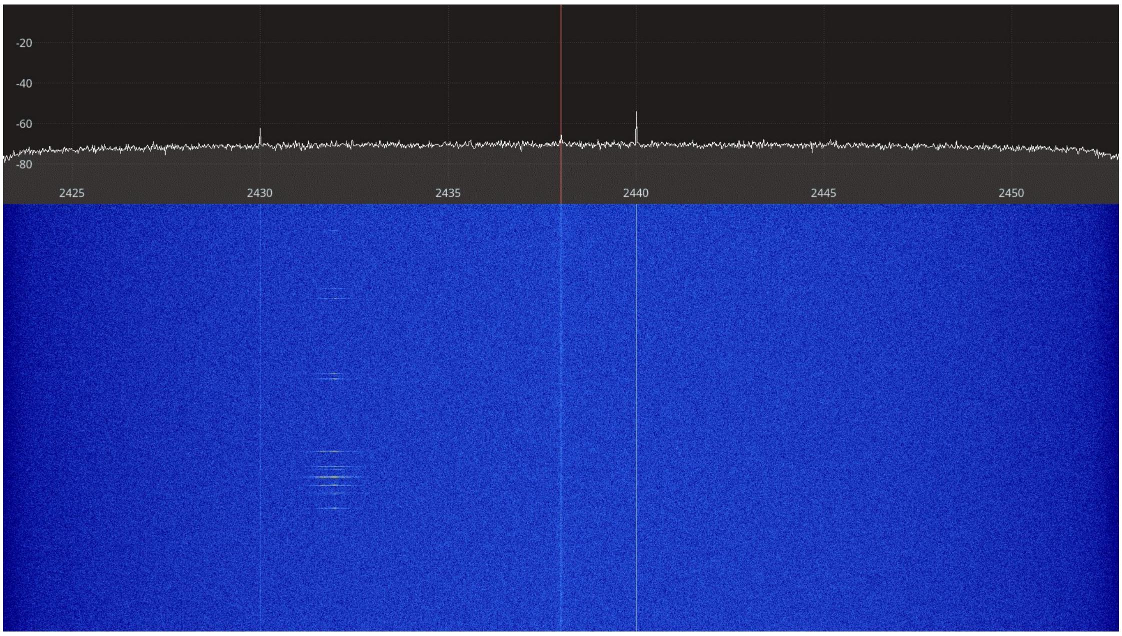


Figure 2.107: Ettus USRP B210 receiver spectrum between 2425 – 2450 MHz.

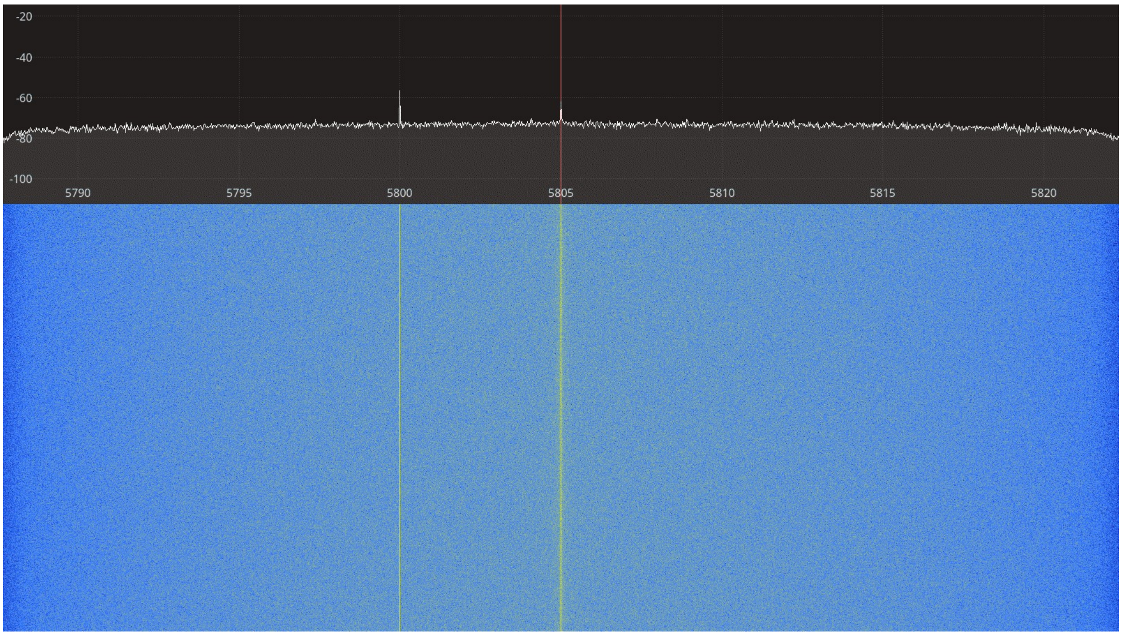


Figure 2.108: Ettus USRP B210 receiver spectrum between 5790 – 5820 MHz.

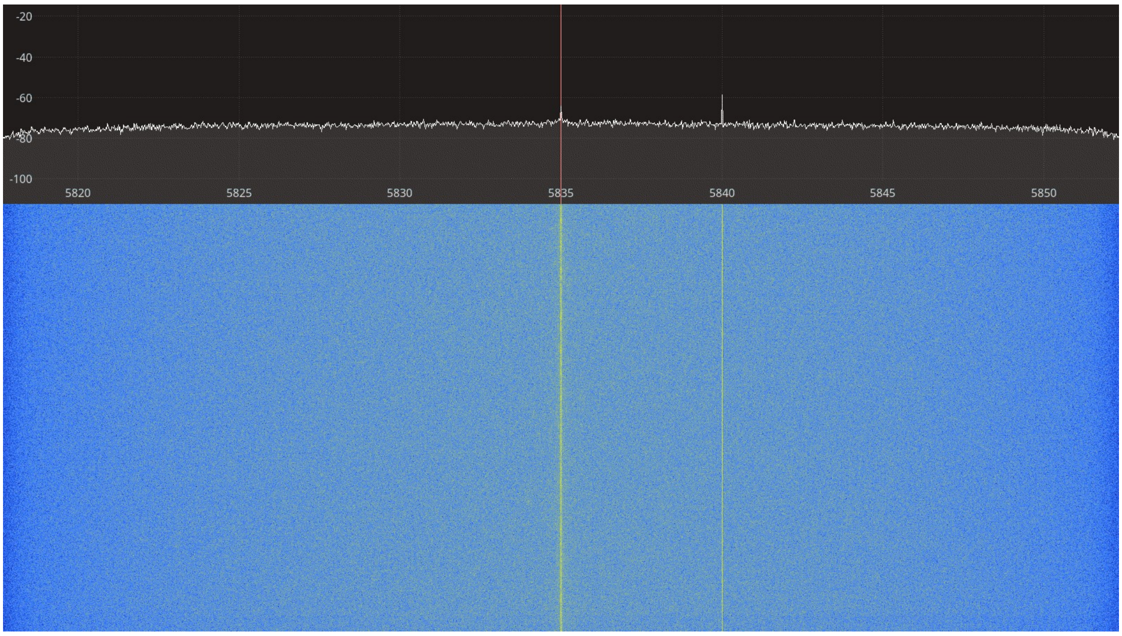


Figure 2.109: Ettus USRP B210 receiver spectrum between 5820 – 5850 MHz.

2.2.6.4 Transmitter Spectral Purity

The transmitter spectra for the USRP B210 transmitting a CW signal are shown on Figures 2.110 to 2.121 below. For each frequency, two spectra are shown, one at a gain setting where the spectrum is clean only showing the CW signal, and one at a higher gain setting where spurious artifacts appear on the spectrum.

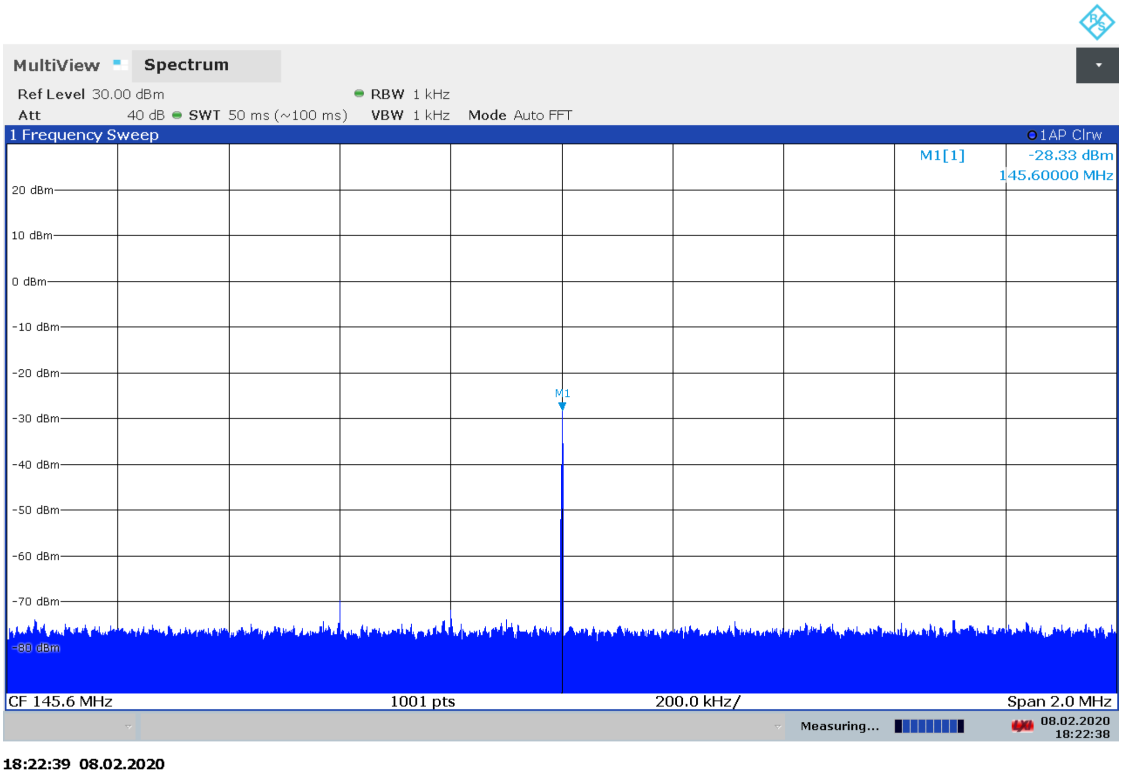


Figure 2.110: Ettus USRP B210 transmitter spectrum at 145.5 MHz, gain 35.

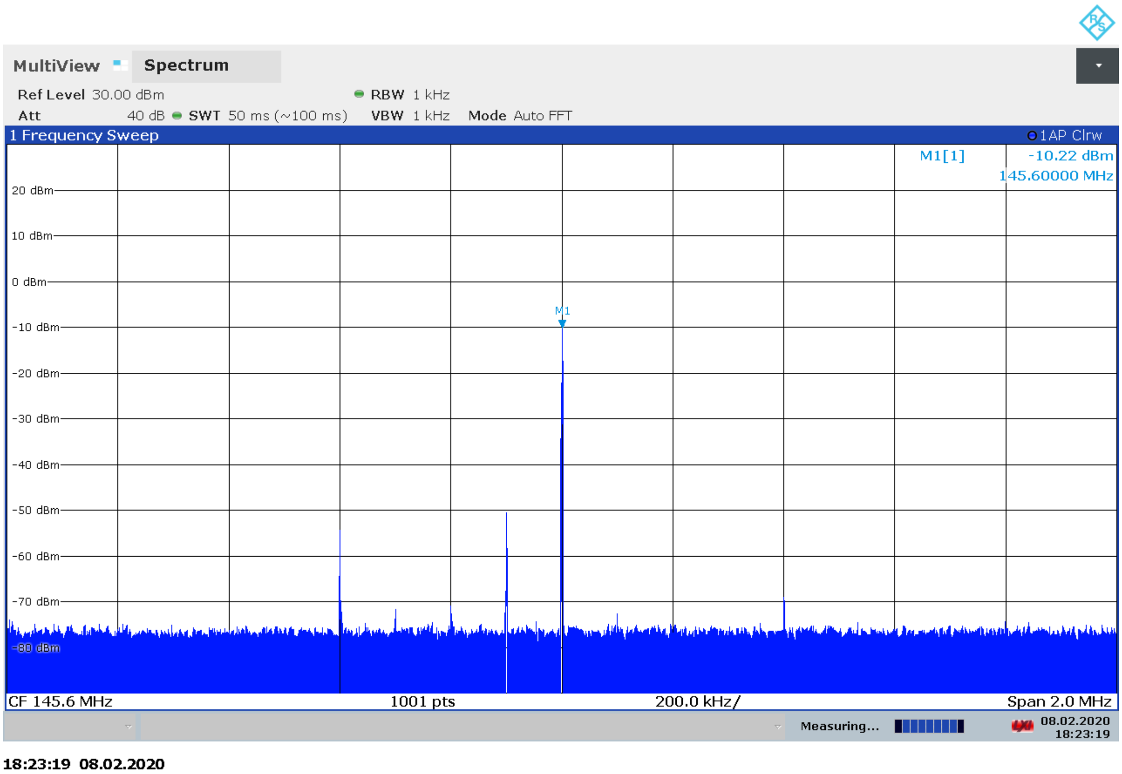


Figure 2.111: Ettus USRP B210 transmitter spectrum at 145.5 MHz, gain 53.

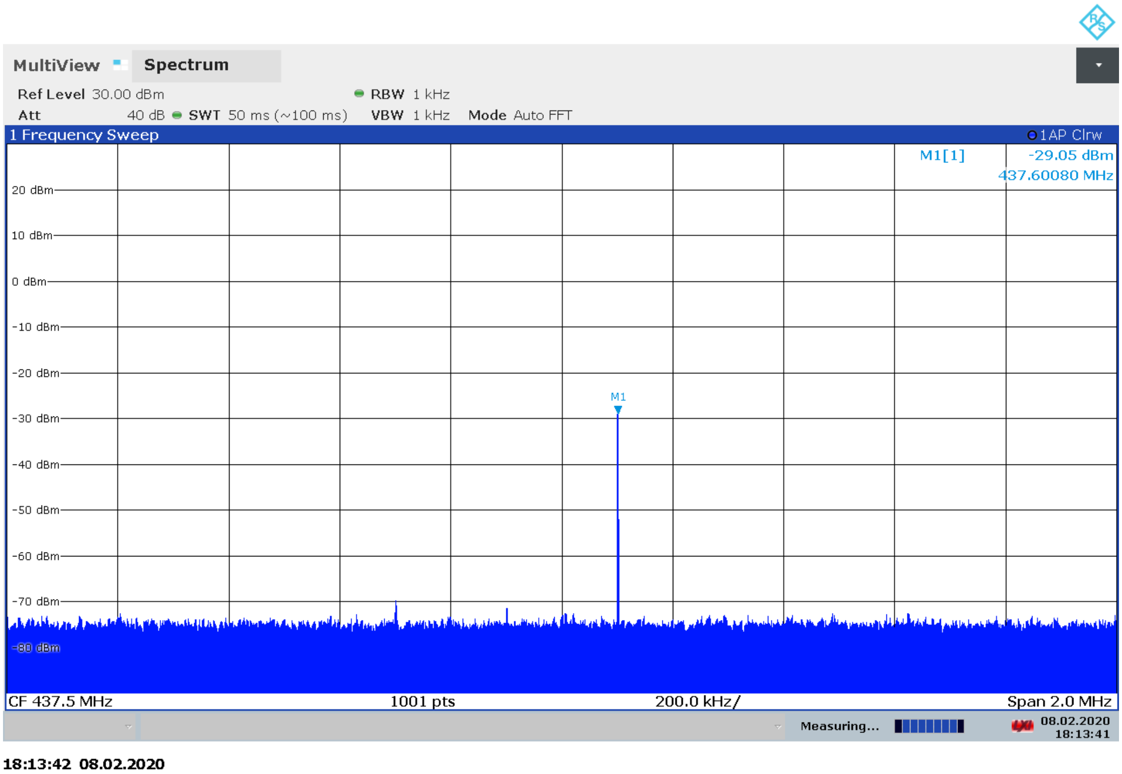


Figure 2.112: Ettus USRP B210 transmitter spectrum at 437.5 MHz, gain 35.

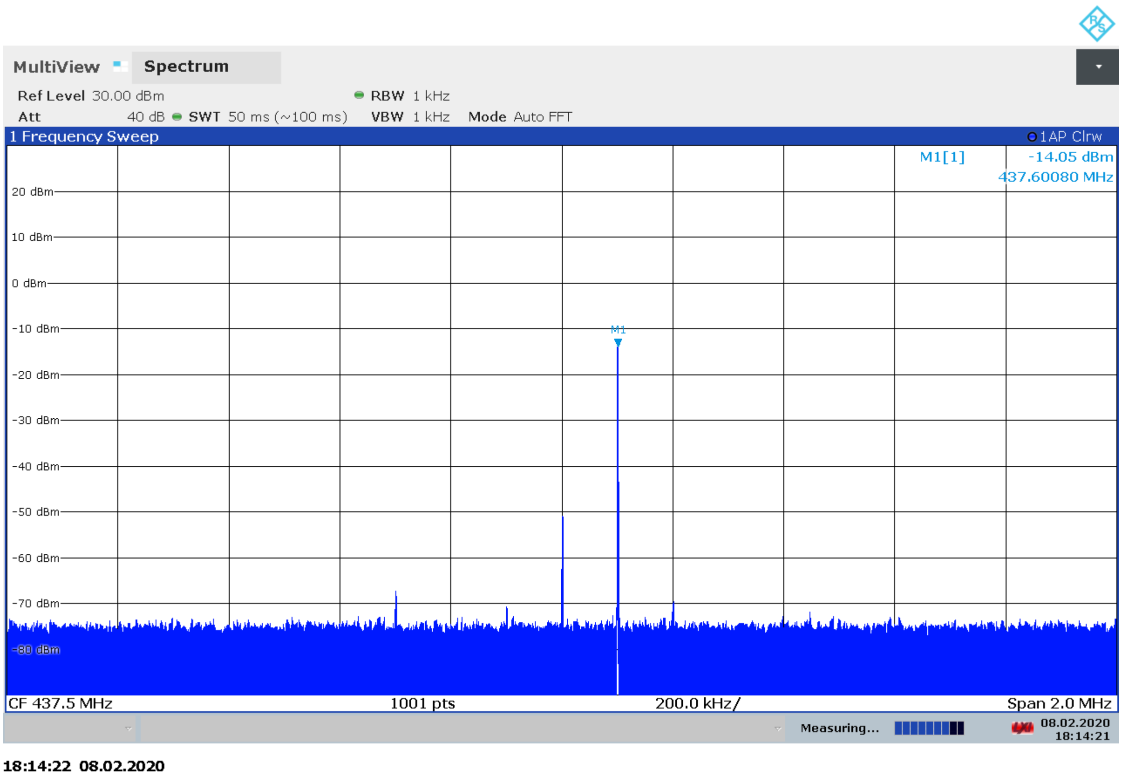


Figure 2.113: Ettus USRP B210 transmitter spectrum at 437.5 MHz, gain 50.

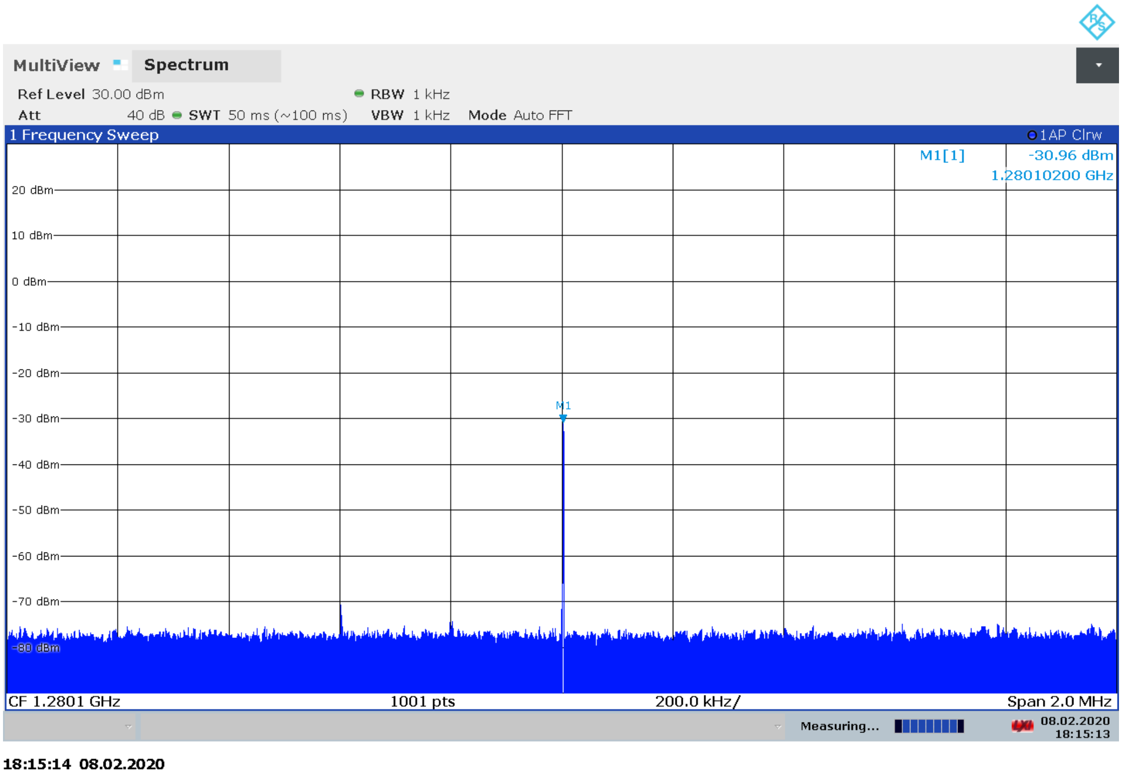


Figure 2.114: Ettus USRP B210 transmitter spectrum at 1280 MHz, gain 35.

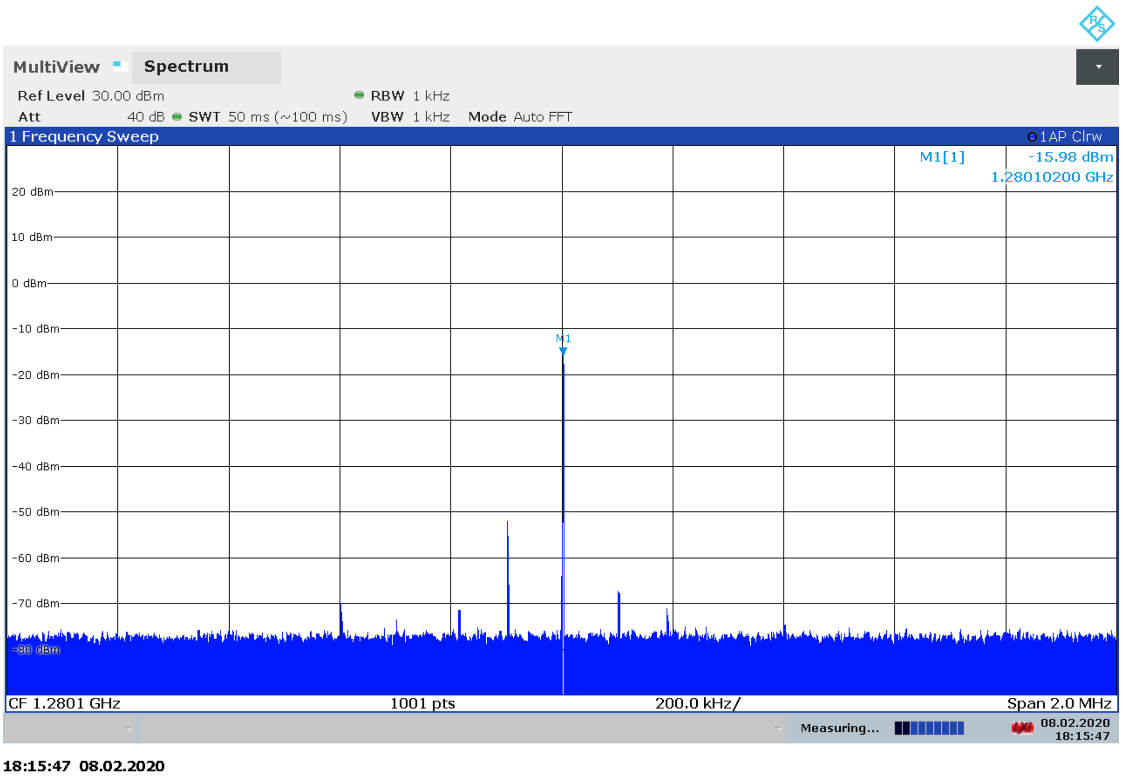


Figure 2.115: Ettus USRP B210 transmitter spectrum at 1280 MHz, gain 50.

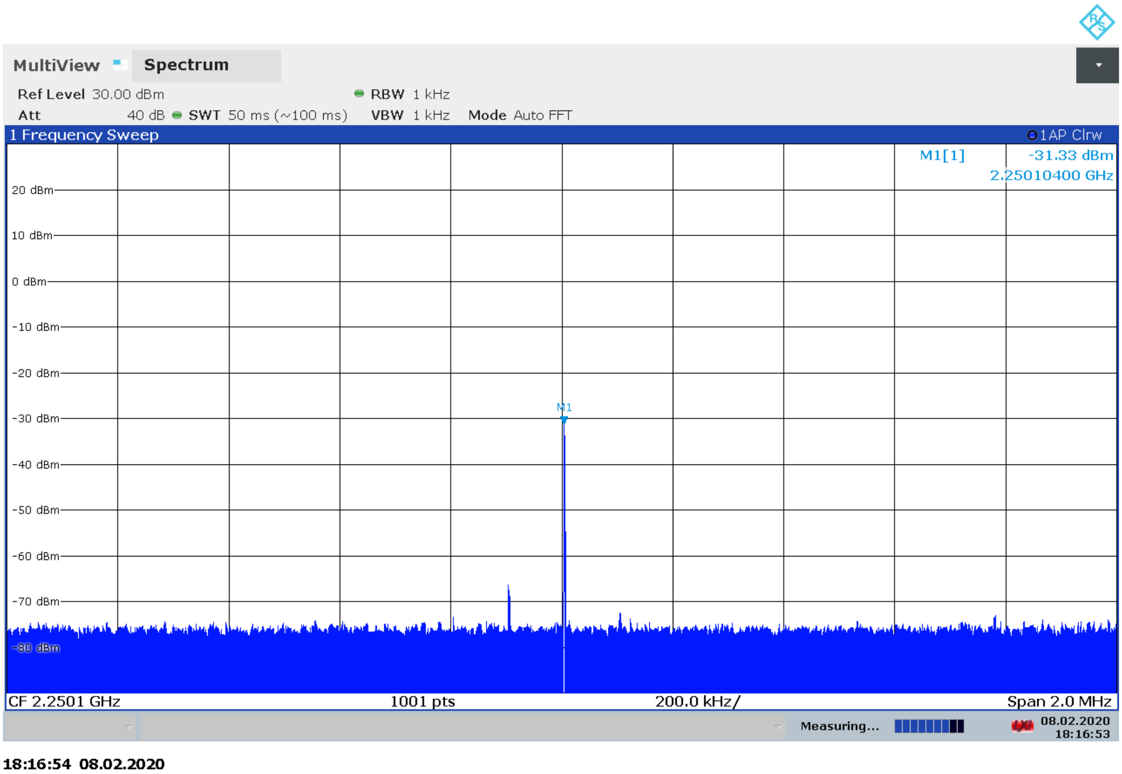


Figure 2.116: Ettus USRP B210 transmitter spectrum at 2250 MHz, gain 43.

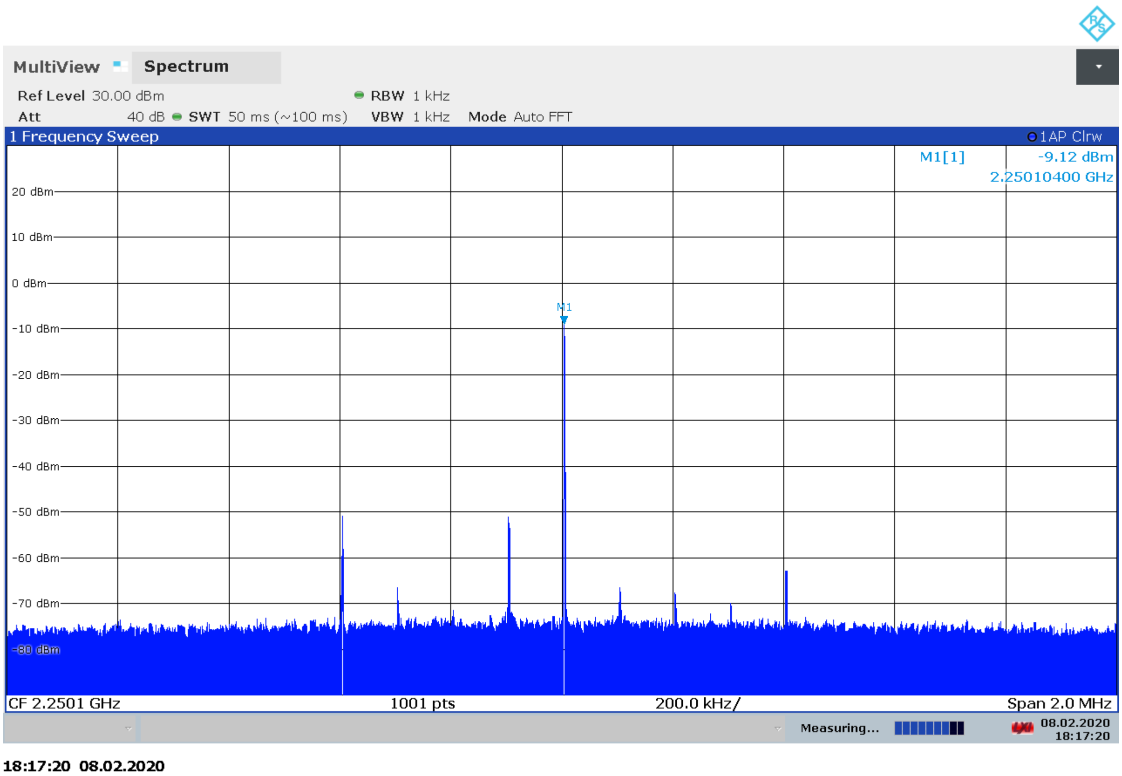


Figure 2.117: Ettus USRP B210 transmitter spectrum at 2250 MHz, gain 65.

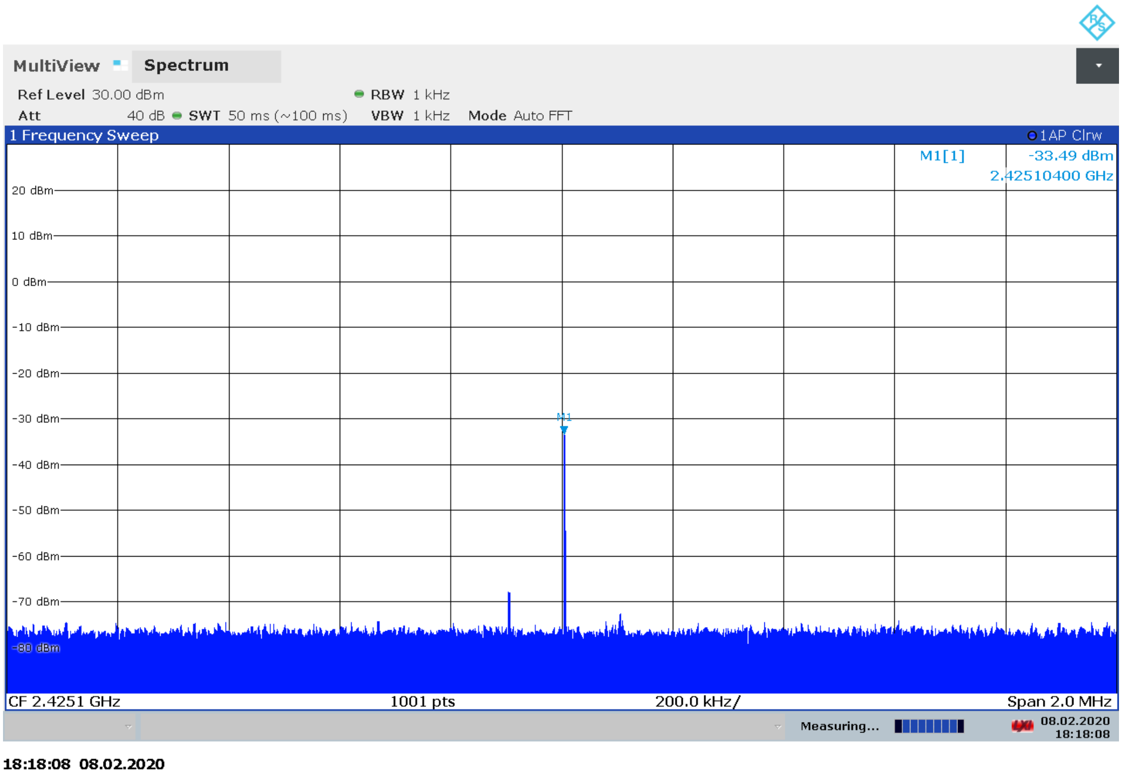


Figure 2.118: Ettus USRP B210 transmitter spectrum at 2425 MHz, gain 43.

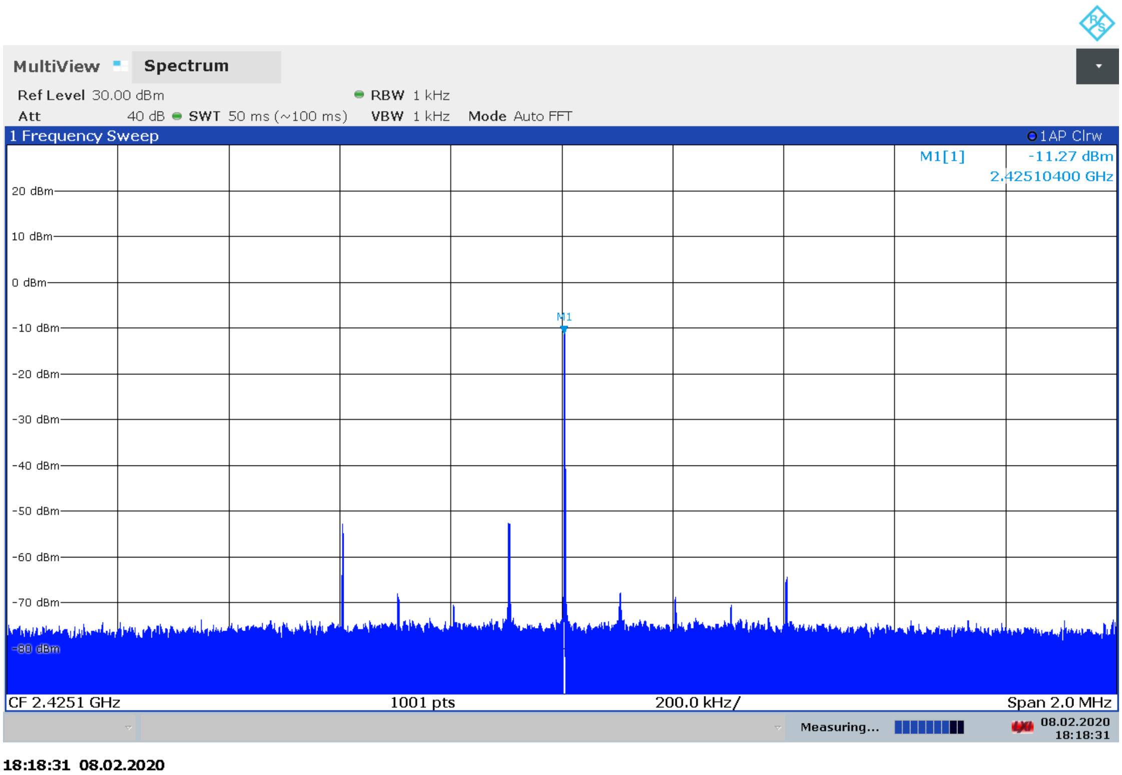


Figure 2.119: Ettus USRP B210 transmitter spectrum at 2425 MHz, gain 65.

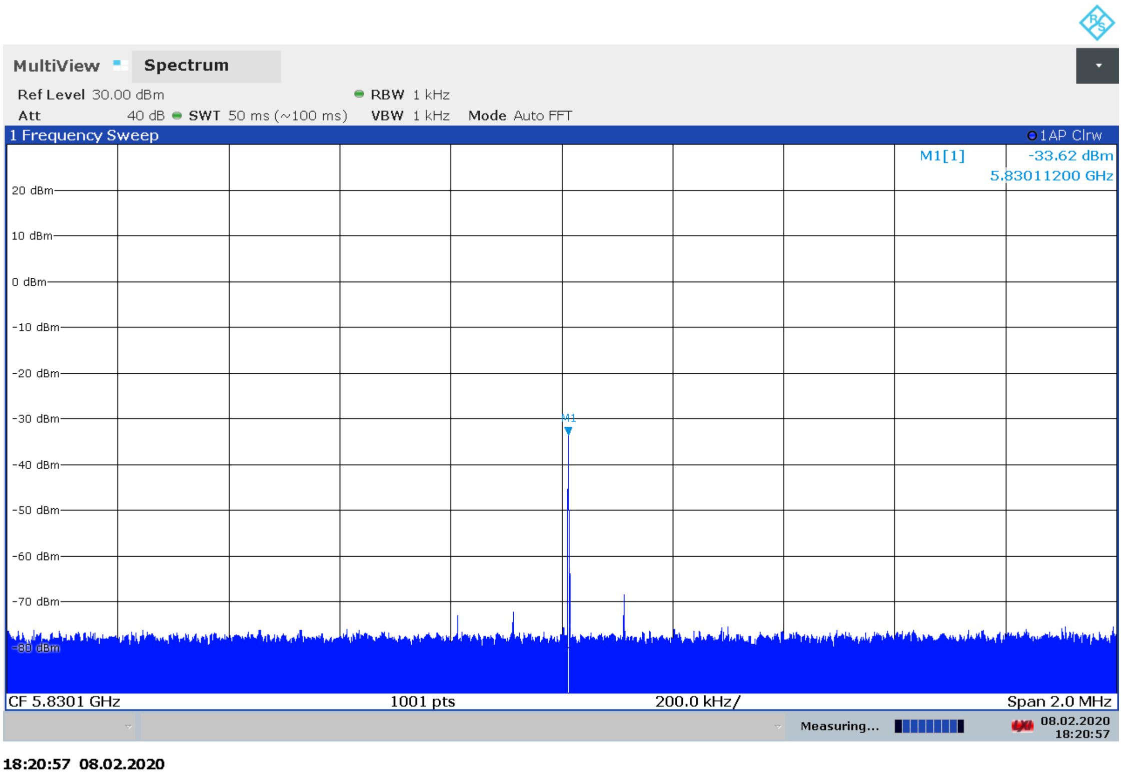


Figure 2.120: Ettus USRP B210 transmitter spectrum at 5830 MHz, gain 43.

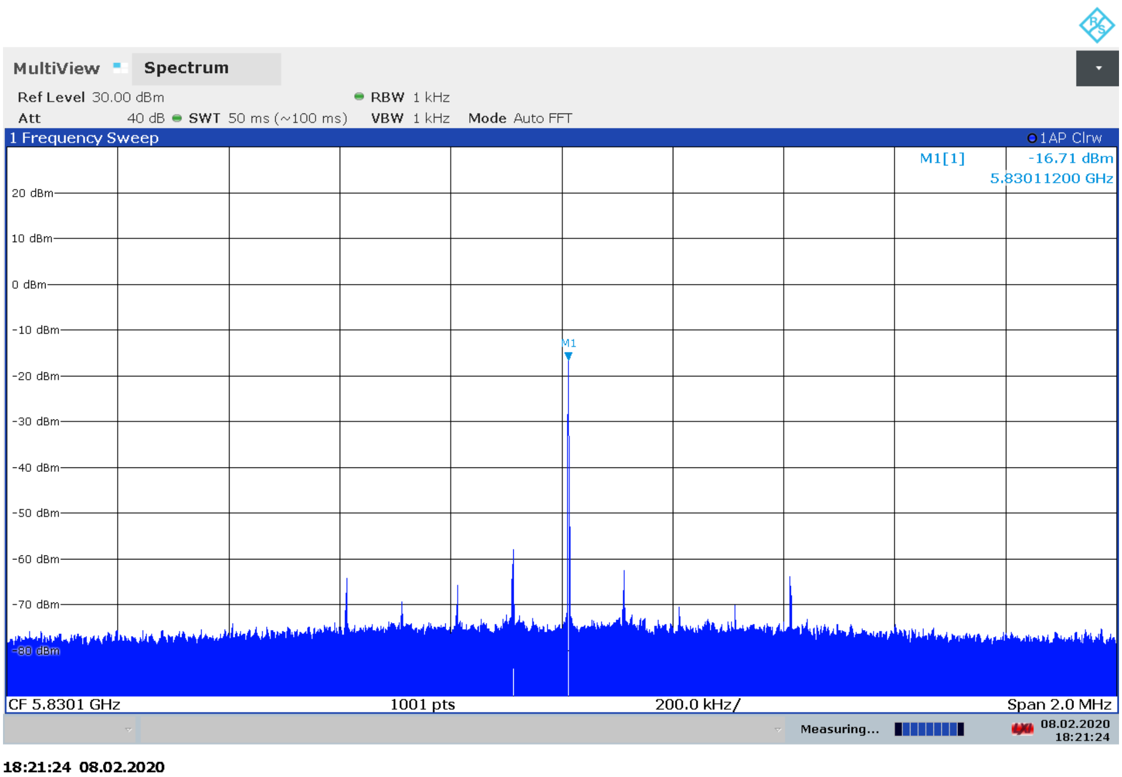


Figure 2.121: Ettus USRP B210 transmitter spectrum at 5830 MHz, gain 60.

2.2.6.5 Transmitter Output Power

The measured transmitter output for the USRP B210 is shown on Figure 2.122 below. As we can see there is a linear relationship between the transmitter gain setting and the output power.

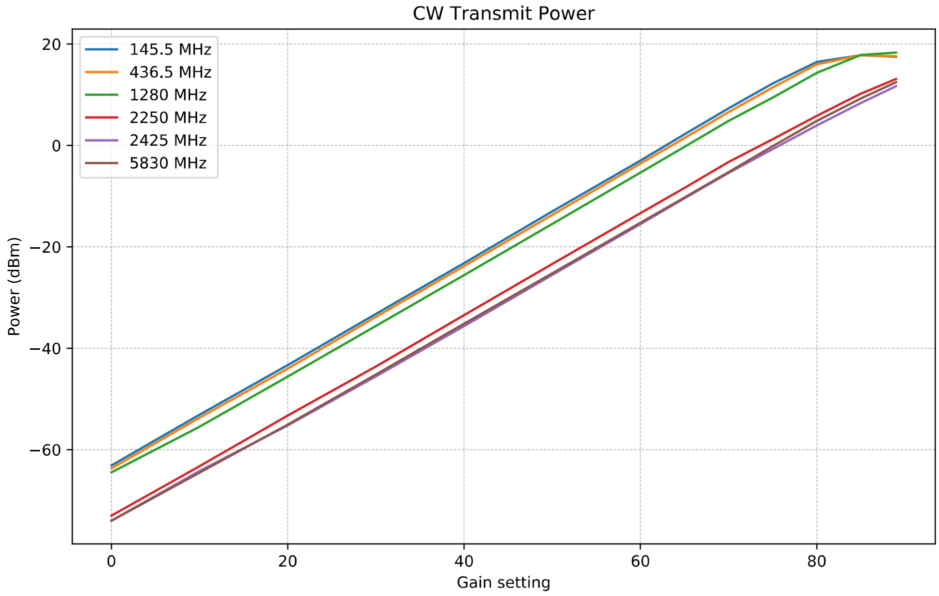


Figure 2.122: Ettus USRP B210 CW transmitter output power as function of gain.

2.2.6.6 Transmitter MER

The measured output power and modulation quality for a DVB-S2 signal transmitted with the USRP B210 are shown in Table 2.21 below. As we noted in test procedure description in section 2.1.6, we believe that the NsMAR parameter is a margin above the required E_S/N_0, which is 11 dB for the modulation and FEC settings used in our tests. Also note that the saturation in NsMAR and MER is due to the limitation of the signal analyzer and does not necessarily represent a limitation in the SDR device.

TX Gain	Power (dBm)	NsMAR (dB)	MER (dB)
0	-73.1	2.9	15.4
10	-62.6	12.7	25.0
20	-53.2	12.7	25.0
30	-42.3	12.7	25.0

TX Gain	Power (dBm)	NsMAR (dB)	MER (dB)
40	-33.3	12.7	25.0
50	-22.4	12.7	25.0
60	-14.5	12.7	25.0

Table 2.21: USRP B210 transmitter MER as function of gain.

2.2.7 PlutoSDR

The PlutoSDR (officially called ADALM-PLUTO Active Learning Module) is a low-cost software defined radio transceiver from Analog Devices Inc., created to be used as a tool for teaching RF and radio communications at all levels [30]. It is based on the Analog Devices AD9363 integrated RFIC and can be used as a full-duplex transceiver between 325 – 3800 MHz.

By default, the PlutoSDR is set up to stream raw I/Q samples to/from a host computer through a USB2 connection. However, the devices contains a ZYNQ processor running embedded Linux, allowing advanced users and developer to run SDR applications on the device. The PlutoSDR comes as an open-source hardware and software with schematics and PCB layout files available through the Analog Devices website [32].

Figure 2.123: The Analog Devices PlutoSDR transceiver.

PlutoSDR devices are controlled via the Linux Industrial I/O Subsytem (IIO). Compared to simple SDR-specific APIs, IIO provides a somehow obscure and unintuitive interface for controlling SDRs and streaming samples. Clear and concise programming documentation is hard to find. Mastering Google and reading the source code of existing applications are must-have skills if one wants to write an application that interfaces to the PlutoSDR.

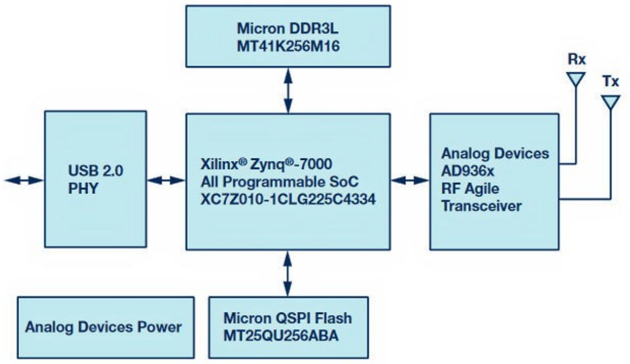


Figure 2.124: PlutoSDR block diagram.

A summary of the technical specifications for the PlutoSDR is listed in Table 2.22 below.

Frequency Range	325 – 3800 MHz (70 MHz – 6 GHz through bootloader config)
Sample rate	7.5 – 12 MSPS (depending on USB2 host [31])
RF bandwidth	20 MHz
Frontend filters	None
RX paths	1
RX inputs	1
TX paths	1
TX outputs	1
ADC resolution	12
DAC resolution	12
Claimed noise figure	-
Claimed dynamic range	-
Claimed transmit power	-
Reference clock	25 ppm
Other features	ZYNQ processor running embedded linux Onboard flash Open source design
Temperature range	10 – 40 °C
Size	
Weight	127 g (in original case covered with copper tape)
Approximate price	140 €
Product page	https://www.analog.com/en/design-center/evaluation-hardware-and-software/evaluation-boards-kits/adalm-pluto.html

Table 2.22: PlutoSDR technical specifications.

2.2.7.1 Receiver Noise Figure

The settings used during the PlutoSDR noise figure measurements are listed in Table 2.23 and the results are shown on Figure 2.125 below.

Input rate	Decimation	Sample rate	Bandwidth
3 MHz	8	375 kHz	Auto

Table 2.23: PlutoSDR settings used during noise figure measurements.

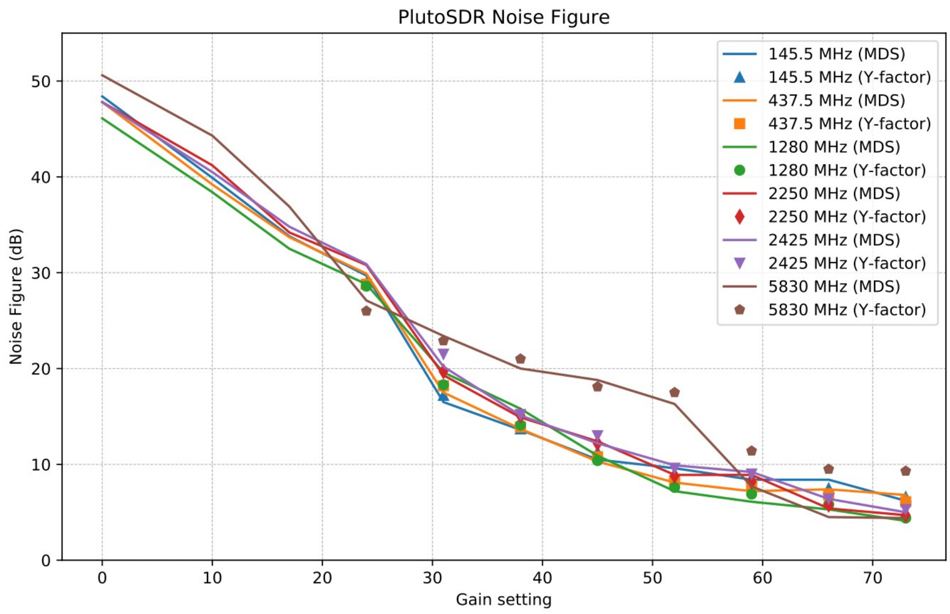


Figure 2.125: PlutoSDR noise figure measurements.

We note that:

- Although no performance data is available for the PlutoSDR, we notice similar noise figure and trends as we measured for the USRP B210.
- We can not explain the large difference between the noise figured calculated from the MDS and the noise figure measured using the Y-factor method on 5.8 GHz. In this case, we trust the Y-factor measurement since they are more consistent with the trend and expectation of higher NF at higher frequencies.

2.2.7.2 Receiver Blocking Dynamic Range

The settings used during the PlutoSDR dynamic range measurements are listed in Table 2.24 and the results are shown on Figure 2.126 below.

Input rate	Decimation	Sample rate	Bandwidth
3 MHz	1	3 MHz	Auto

Table 2.24: PlutoSDR settings used for BDR measurements.

We note that:

- Ax expected, the highest input power that the device can tolerate without overload, P_{in}, is decreasing as the gain is increased. The relationship is mostly linear.
- The dynamic range is mostly constant over the entire gain spectrum, slightly dropping on some frequencies at the highest gains.
- There is a large (30 dB) variation in the dynamic range as we go from 145 MHz to 5.8 GHz.

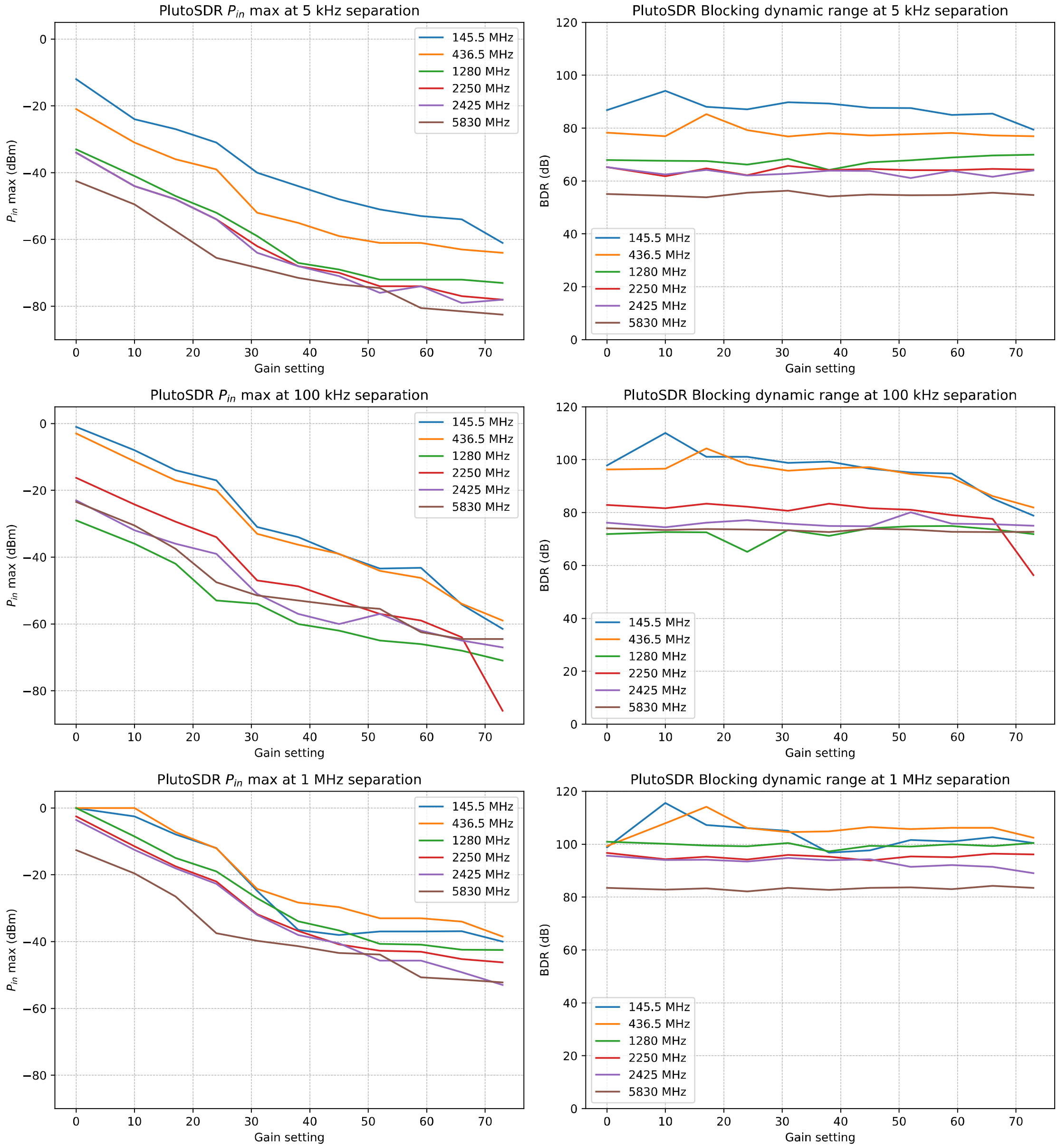


Figure 2.126: PlutoSDR blocking dynamic range measurements.

2.2.7.3 Receiver Spectral Purity

The receiver spectra of the PlutoSDR receiver are shown on Figures 2.127 to 2.139 below. As we can see from the spectra, there are very few spurs generated by or picked up by the device when connected to 50 ohm terminator. The only band where the PlutoSDR seems sensitive to exaternal noise is on 145 MHz and 2.4 GHz (WiFi).

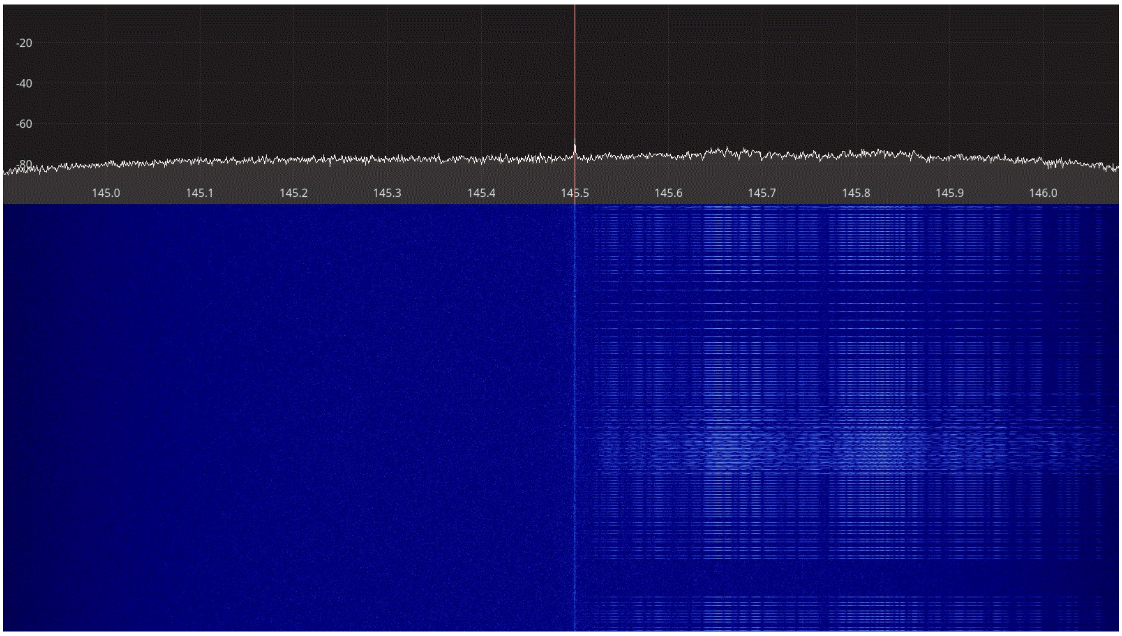


Figure 2.127: PlutoSDR receiver spectrum between 145 – 146 MHz.

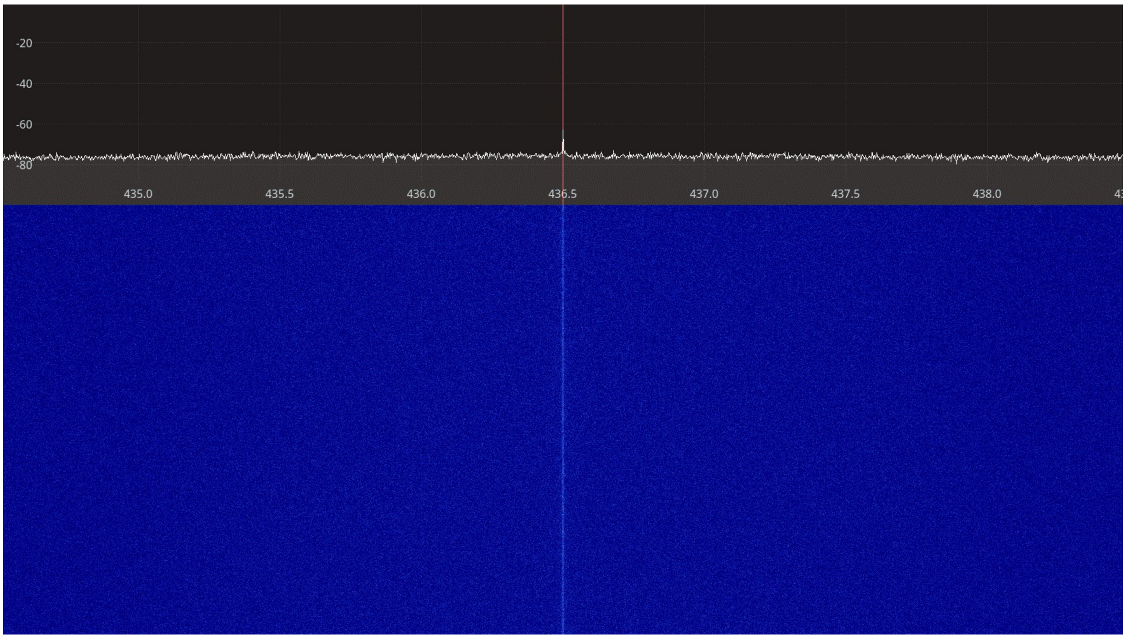


Figure 2.128: PlutoSDR receiver spectrum between 435 - 438 MHz.

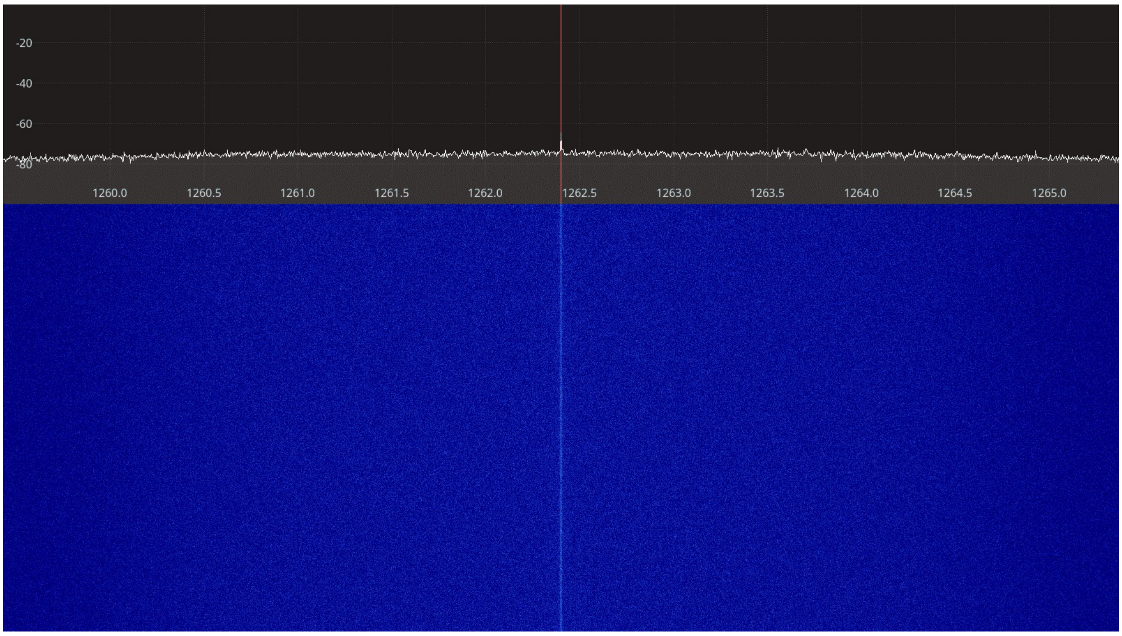


Figure 2.130: PlutoSDR receiver spectrum between 1260 – 1265 MHz.

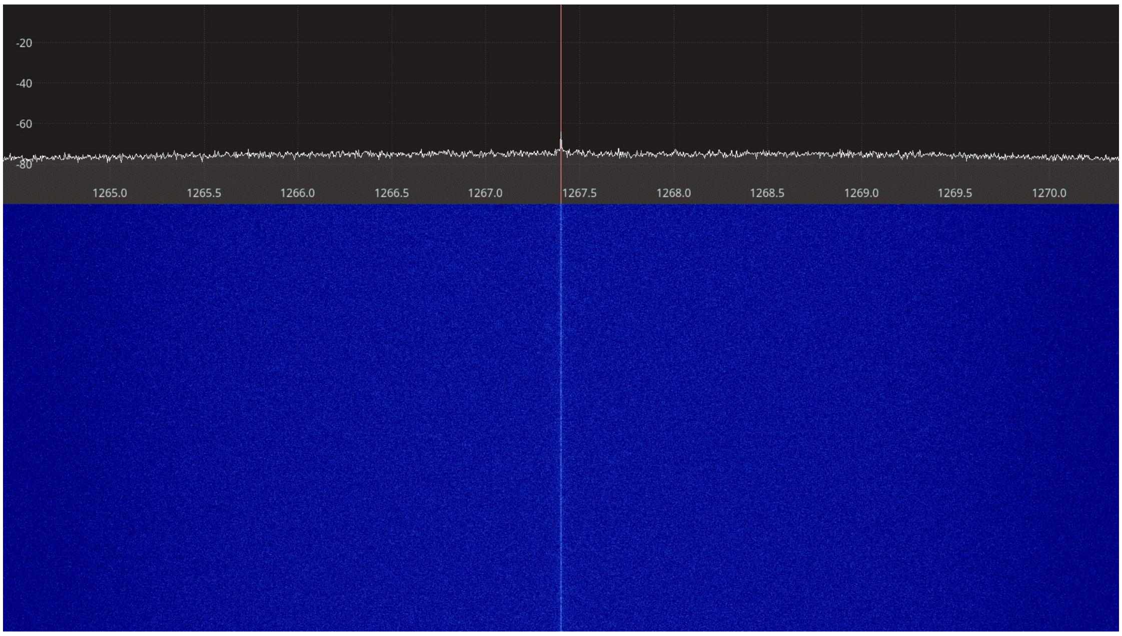


Figure 2.131: PlutoSDR receiver spectrum between 1265 – 1270 MHz.

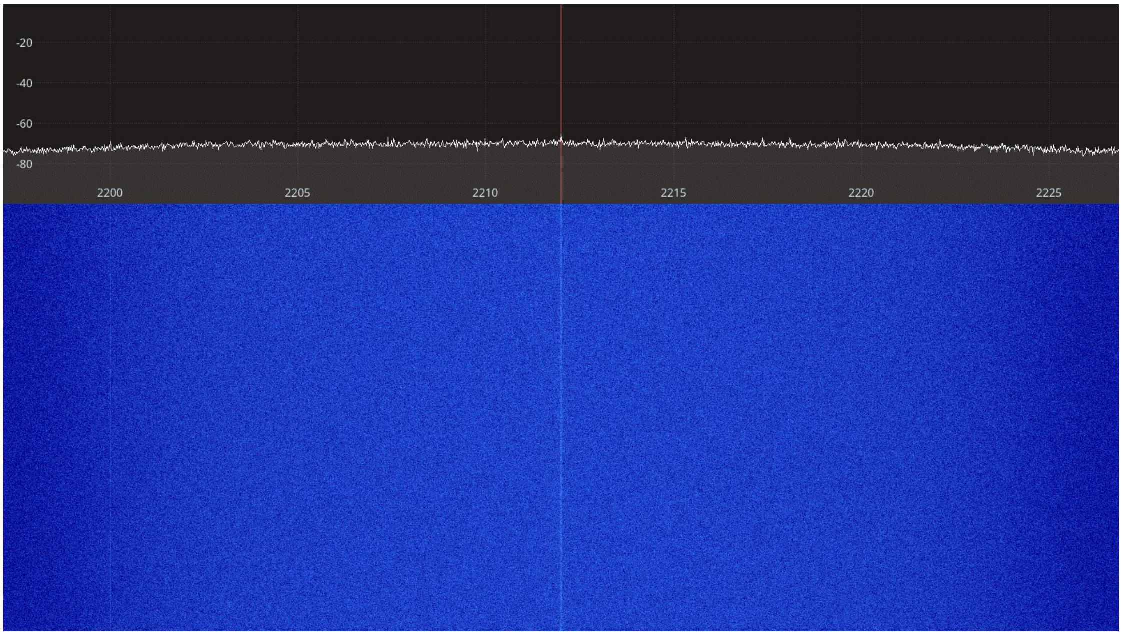


Figure 2.132: PlutoSDR receiver spectrum between 2200 – 2225 MHz.

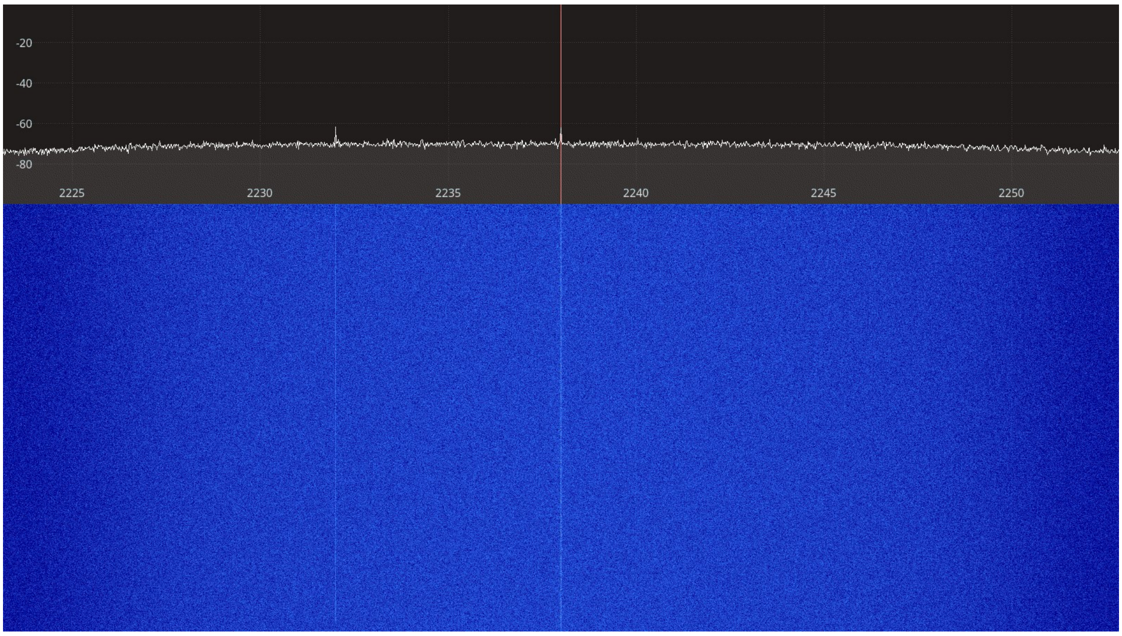


Figure 2.133: PlutoSDR receiver spectrum between 2225 – 2250 MHz.

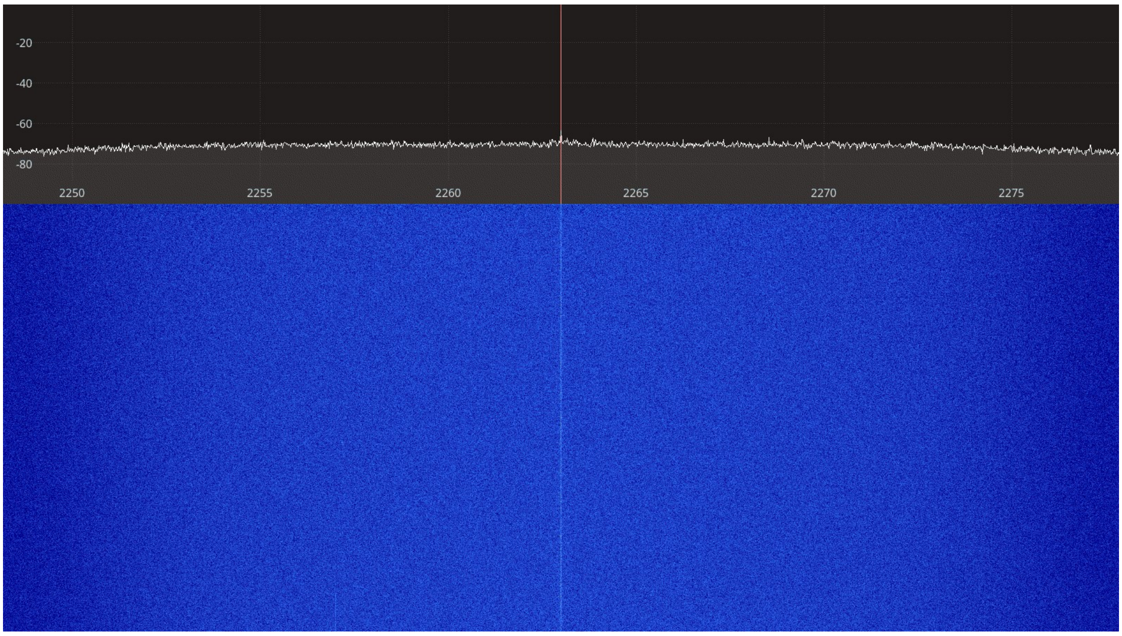


Figure 2.134: PlutoSDR receiver spectrum between 2250 – 2275 MHz.

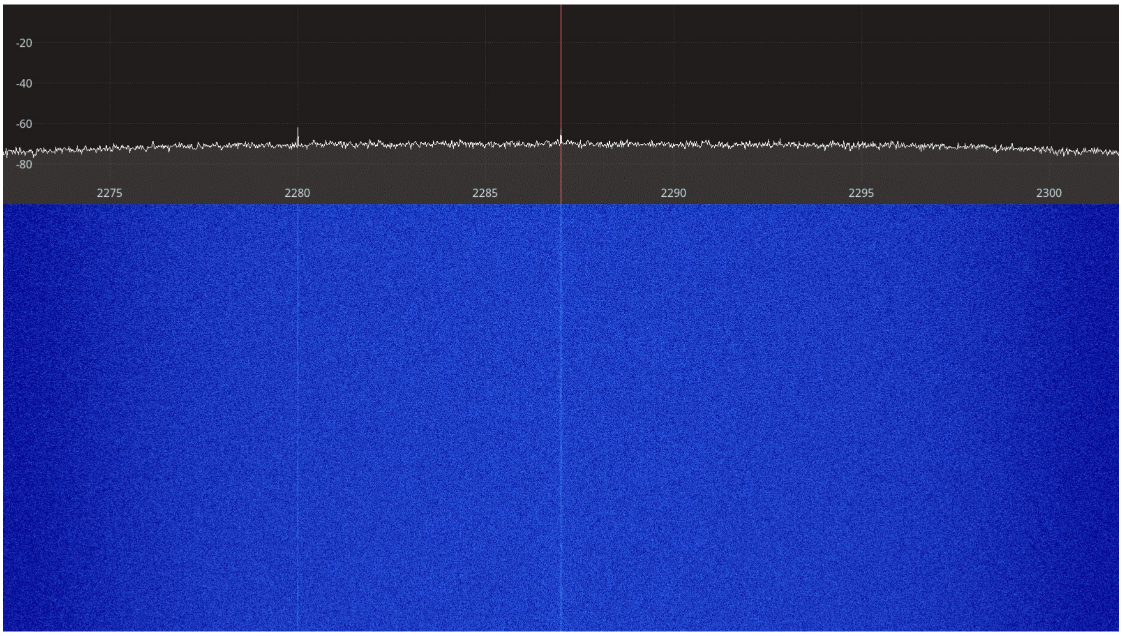


Figure 2.135: PlutoSDR receiver spectrum between 2275 – 2300 MHz.

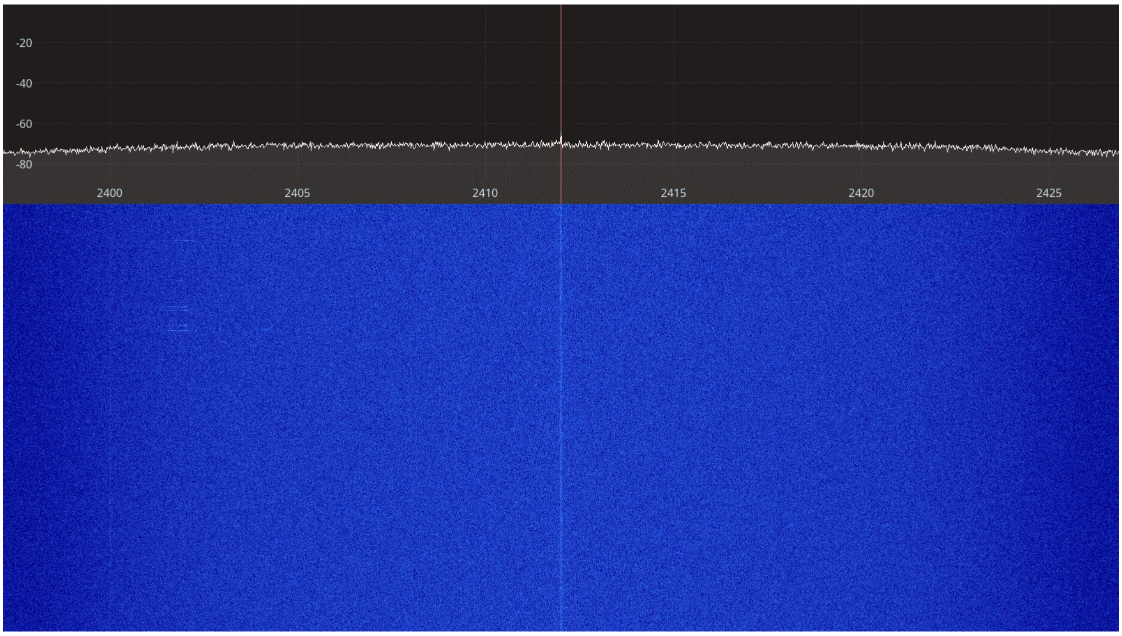


Figure 2.136: PlutoSDR receiver spectrum between 2400 – 2425 MHz.

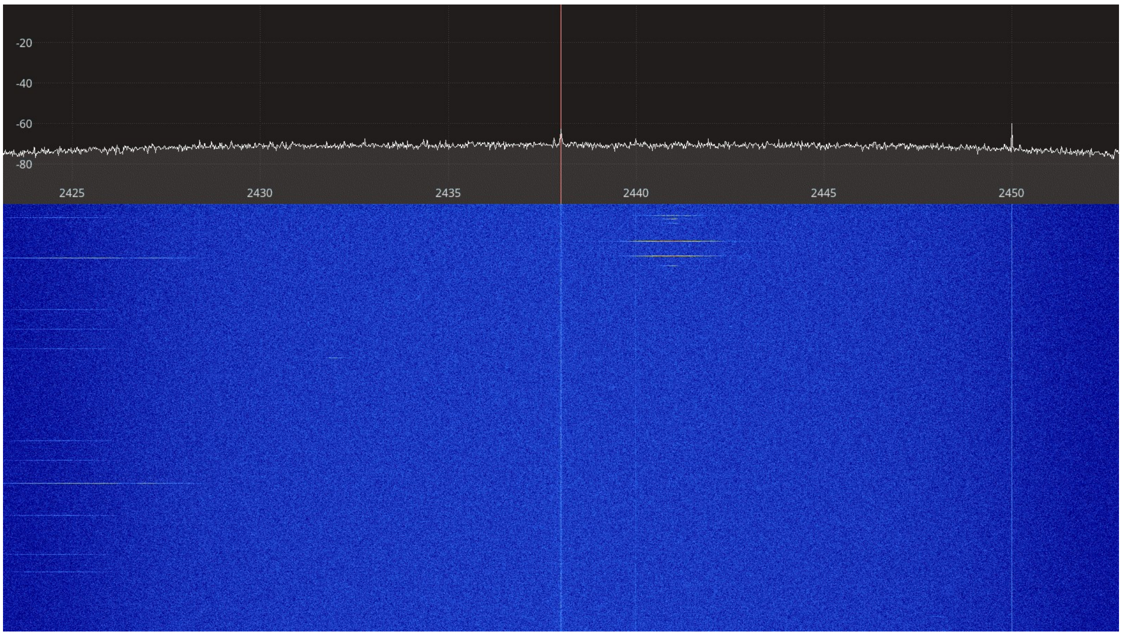


Figure 2.137: PlutoSDR receiver spectrum between 2425 – 2450 MHz.

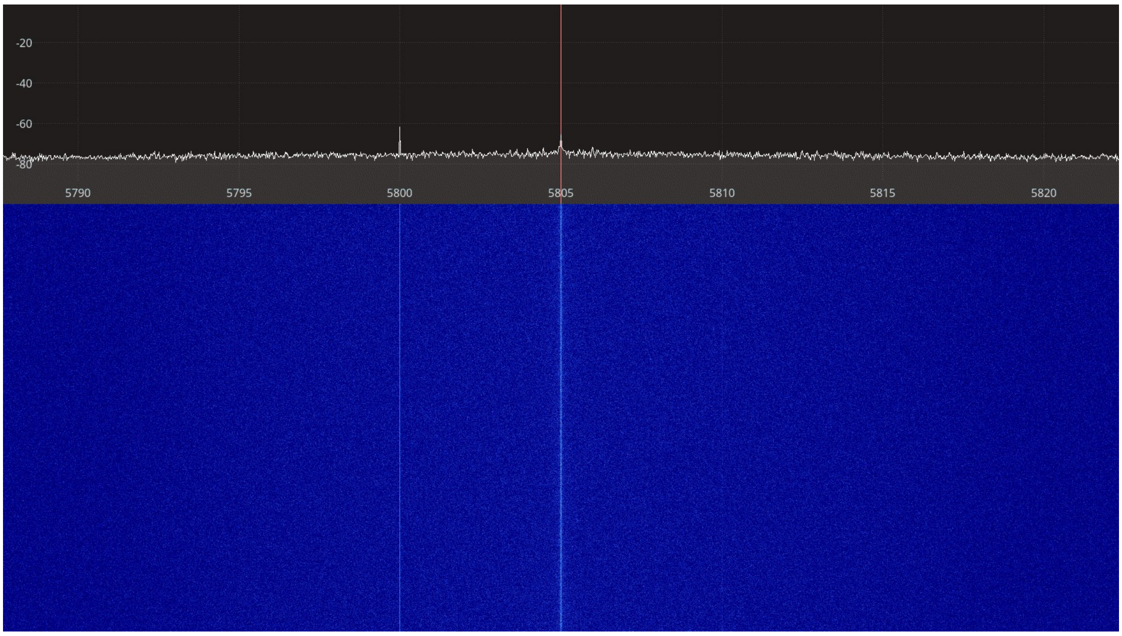


Figure 2.138: PlutoSDR receiver spectrum between 5790 – 5820 MHz.

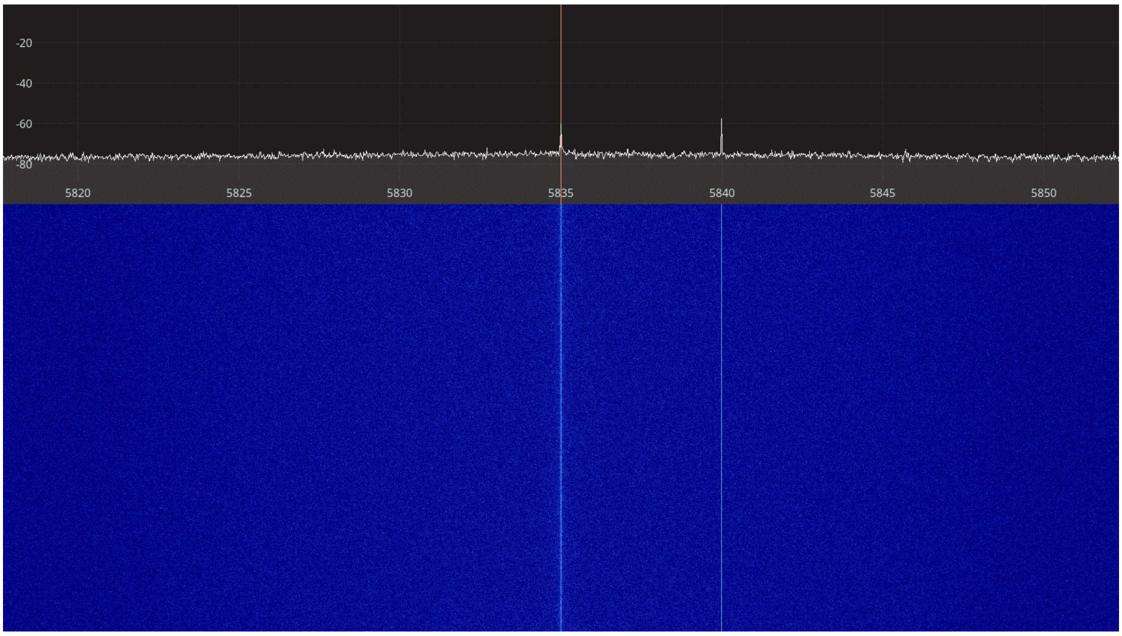


Figure 2.139: PlutoSDR receiver spectrum between 5820 – 5850 MHz.

2.2.7.4 Transmitter Spectral Purity

The transmitter spectra for the PlutoSDR transmitting a CW signal are shown on Figures Figure 2.140 to Figure 2.152 below. For each frequency, two spectra are shown, one at a gain setting where the spectrum is clean only showing the CW signal, and one at a higher gain setting where spurious artifacts appear on the spectrum.

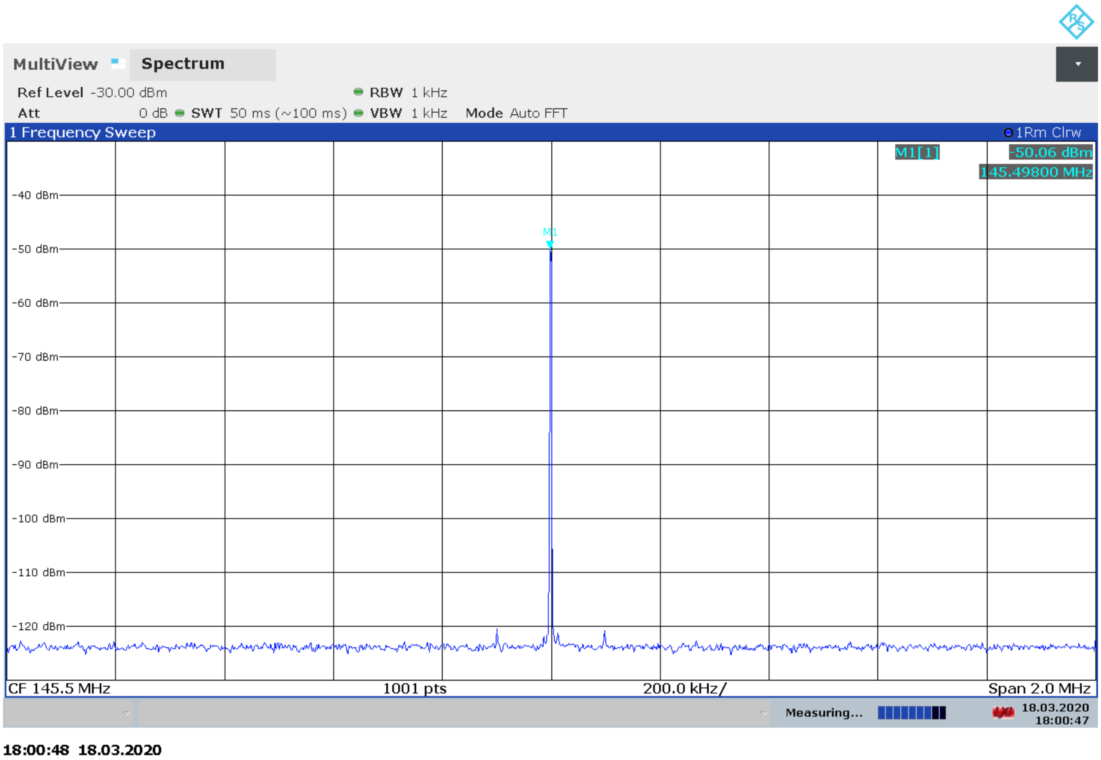


Figure 2.140: PlutoSDR transmitter spectrum at 145.5 MHz, power -40 dBm.

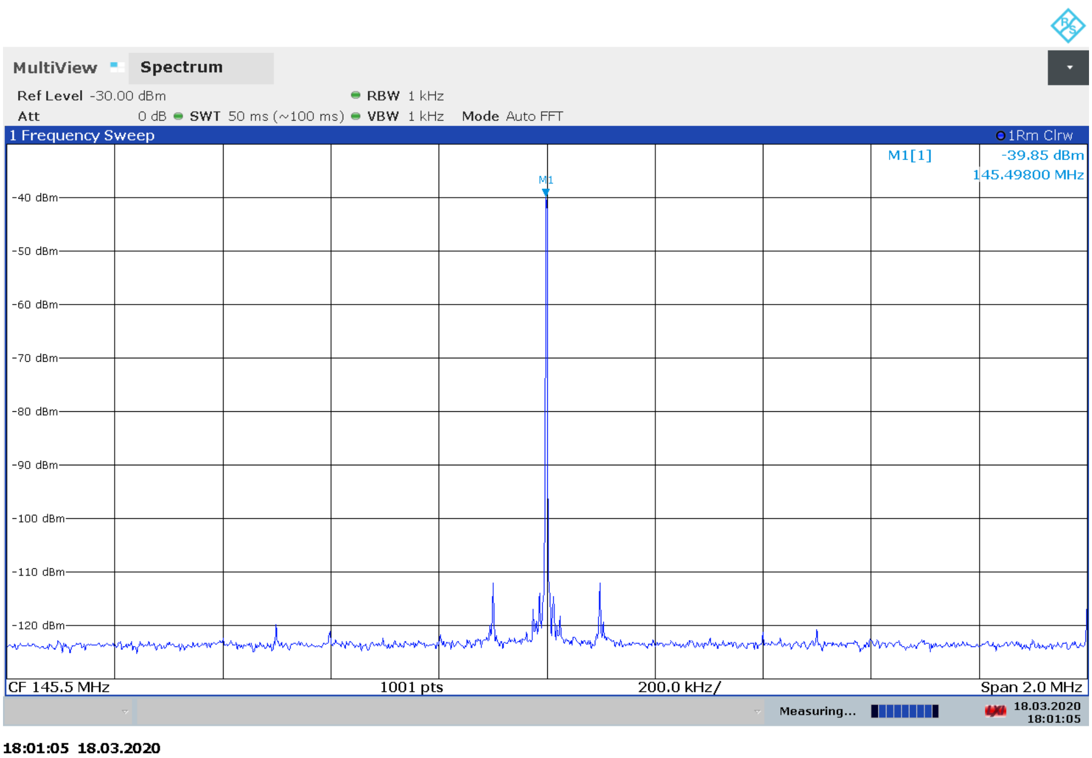


Figure 2.141: PlutoSDR transmitter spectrum at 145.5 MHz, power -30 dBm.

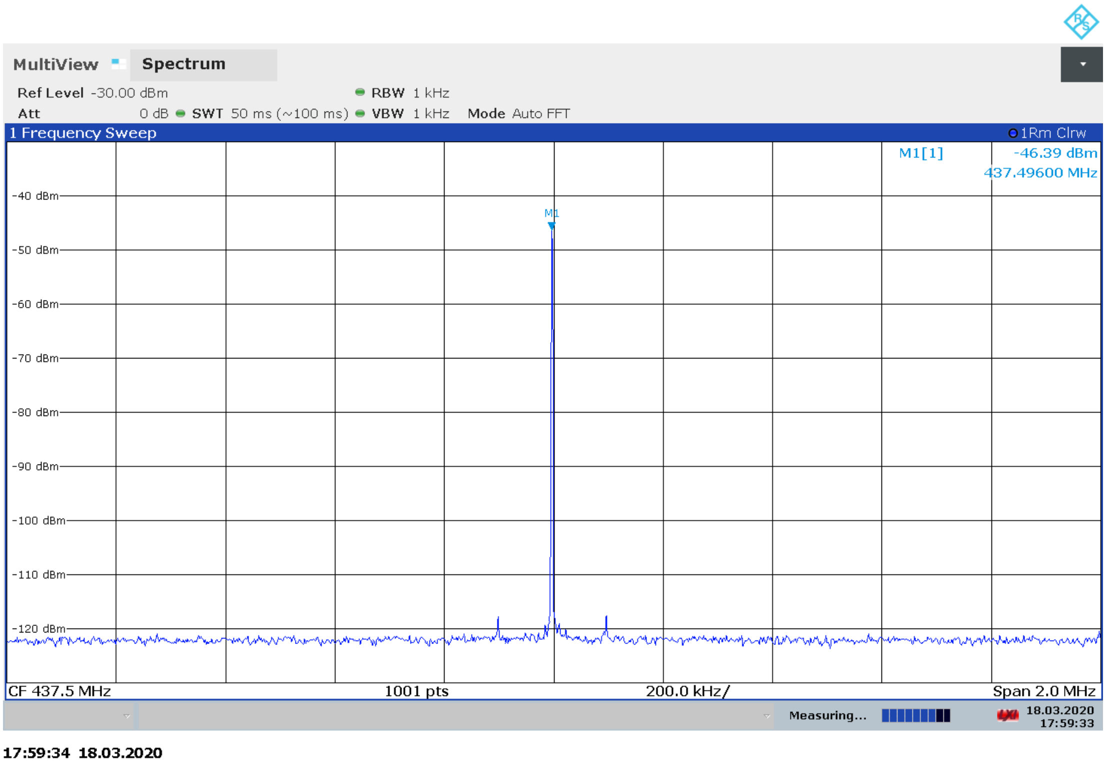


Figure 2.142: PlutoSDR transmitter spectrum at 437.5 MHz, power -40 dBm.

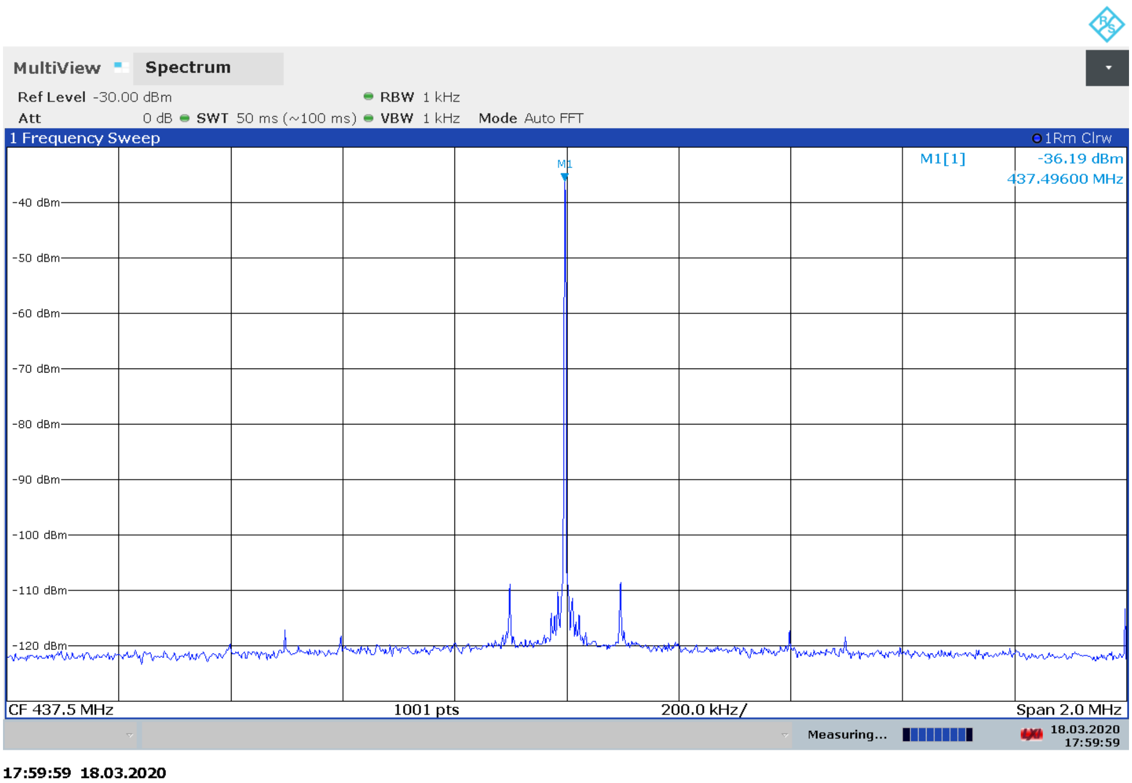


Figure 2.143: PlutoSDR transmitter spectrum at 437.5 MHz, power -30 dBm.

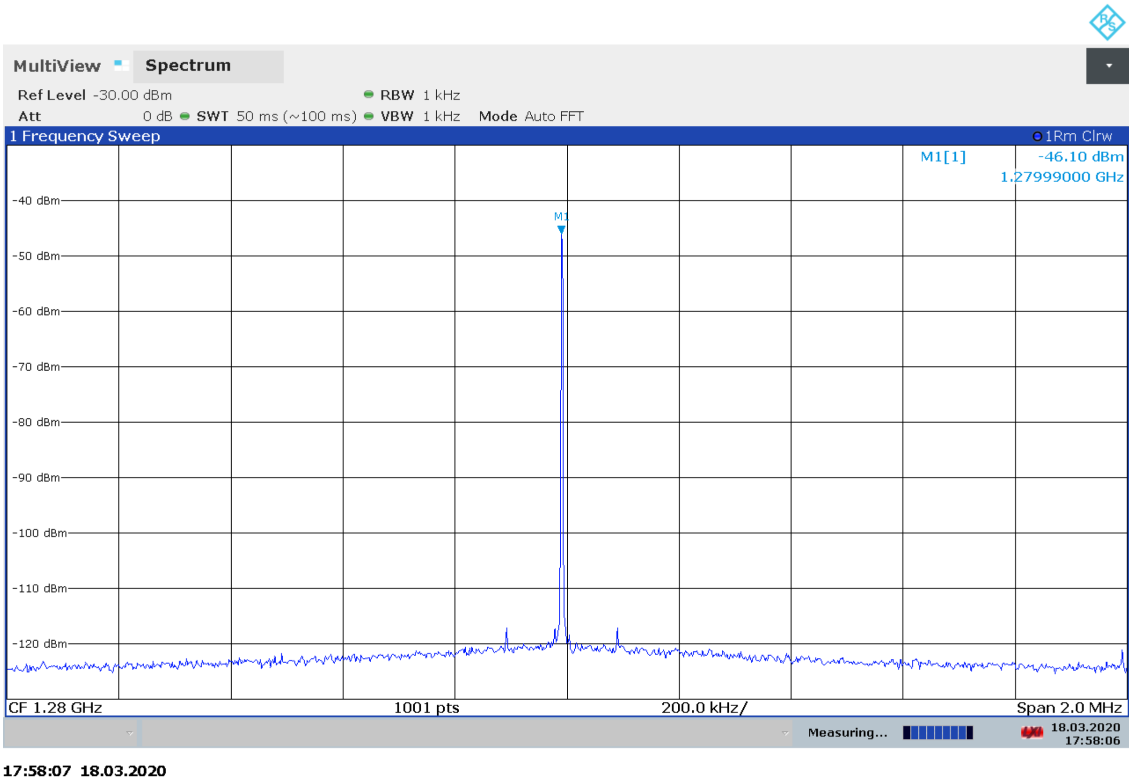


Figure 2.144: PlutoSDR transmitter spectrum at 1280 MHz, power -40 dBm.

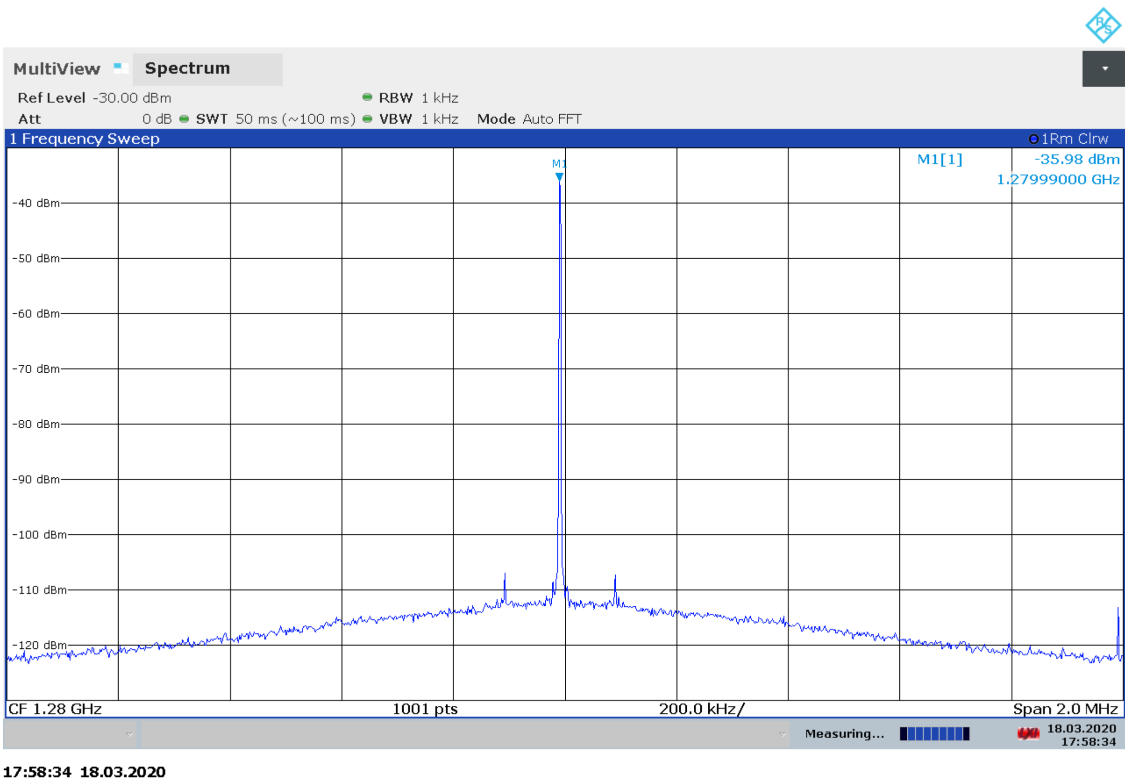


Figure 2.145: PlutoSDR transmitter spectrum at 1280 MHz, power -30 dBm.

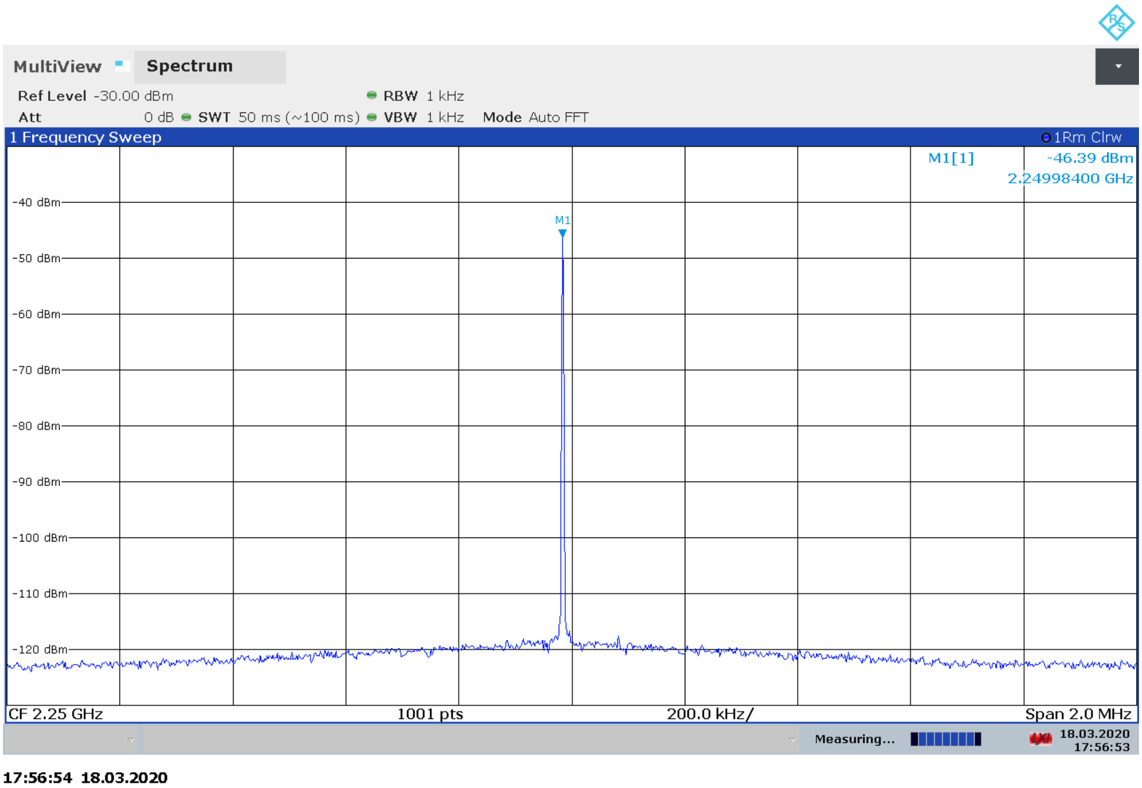


Figure 2.146: PlutoSDR transmitter spectrum at 2250 MHz, power -40 dBm.

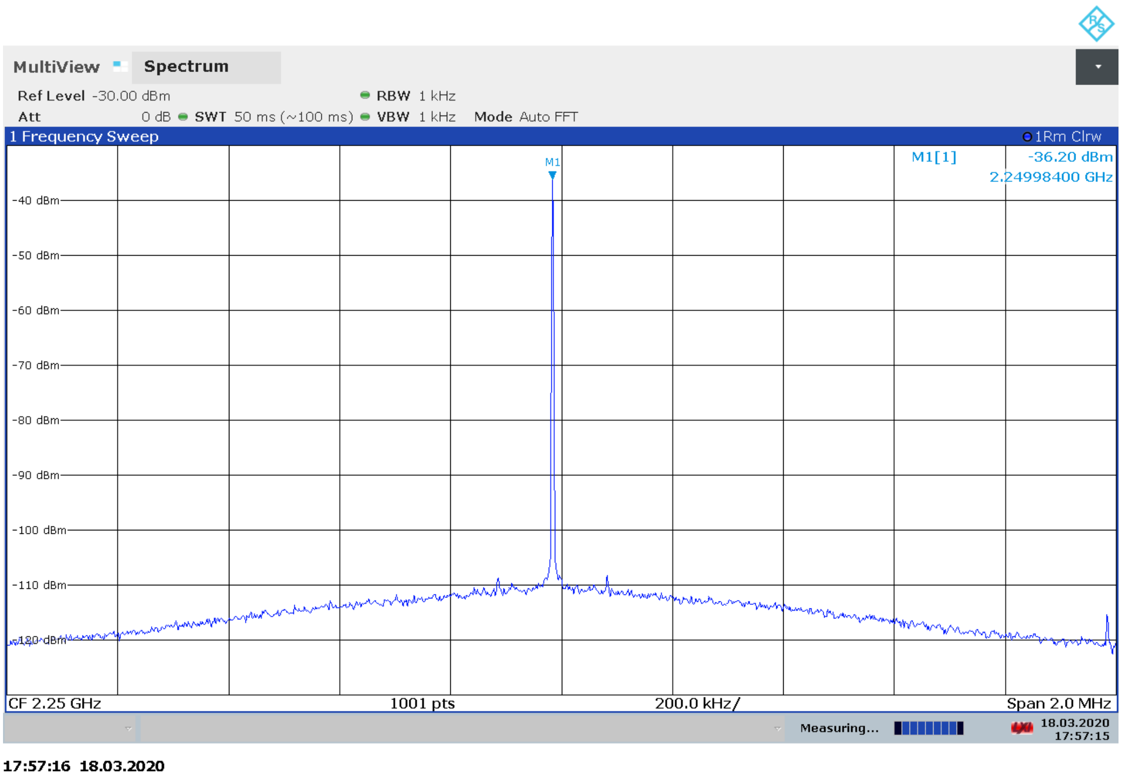


Figure 2.147: PlutoSDR transmitter spectrum at 2250 MHz, power -30 dBm.

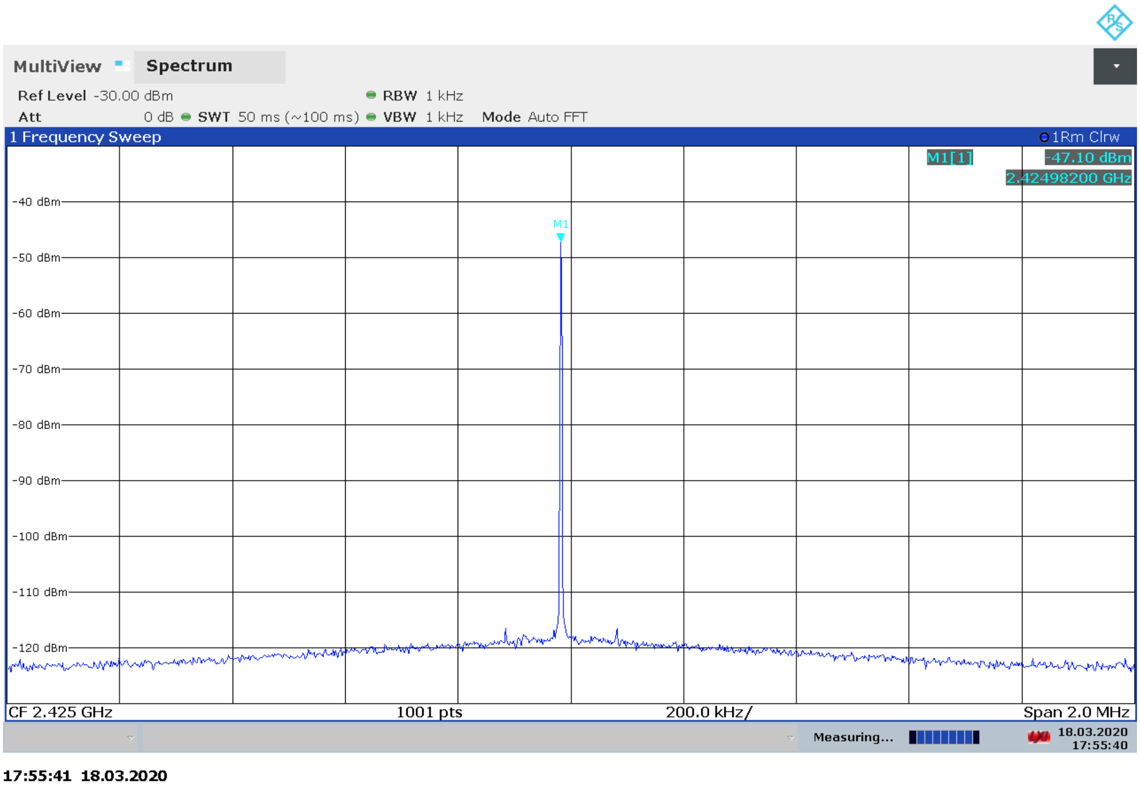


Figure 2.148: PlutoSDR transmitter spectrum at 2425 MHz, power -40 dBm.

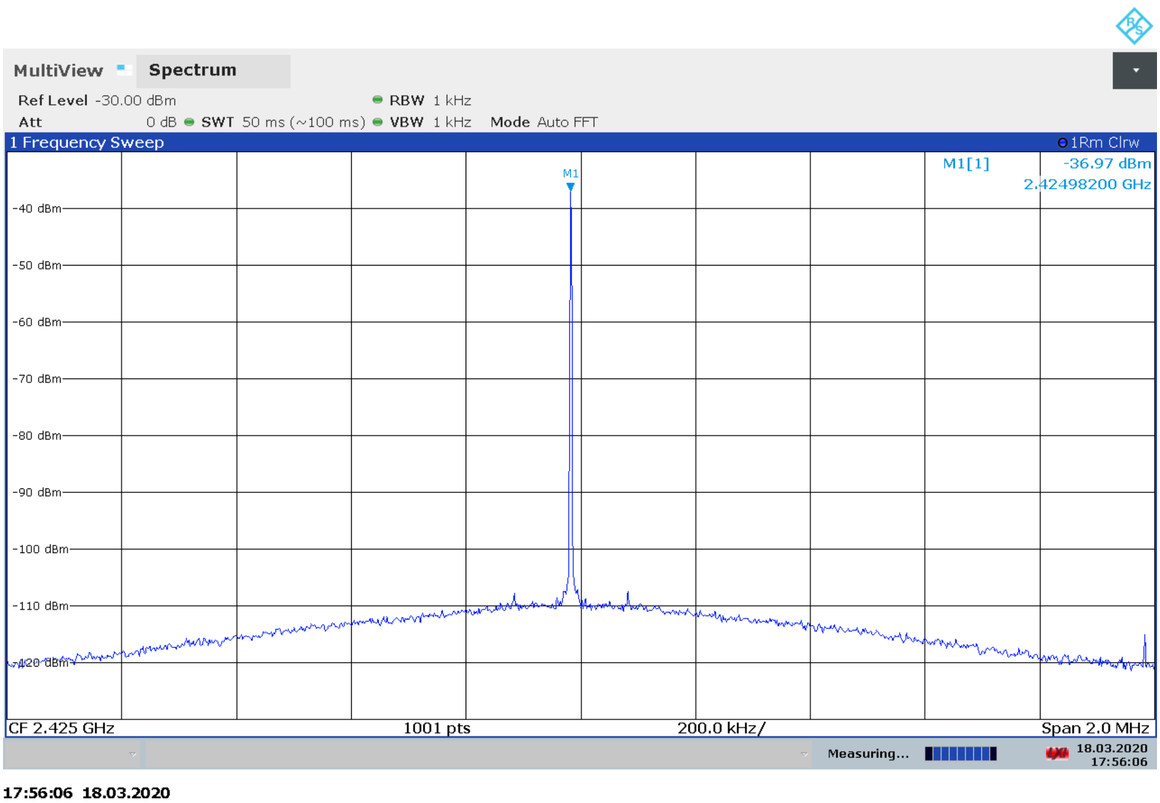


Figure 2.149: PlutoSDR transmitter spectrum at 2425 MHz, power -30 dBm.

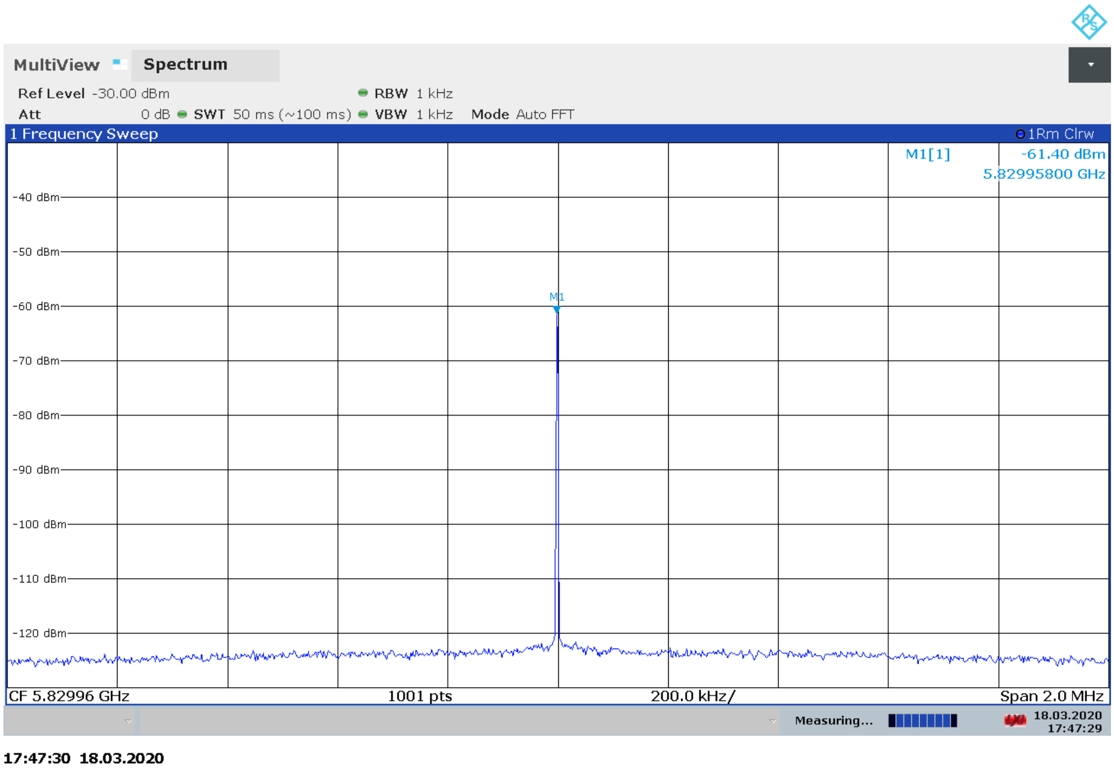


Figure 2.150: PlutoSDR transmitter spectrum at 5830 MHz, power -50 dBm.

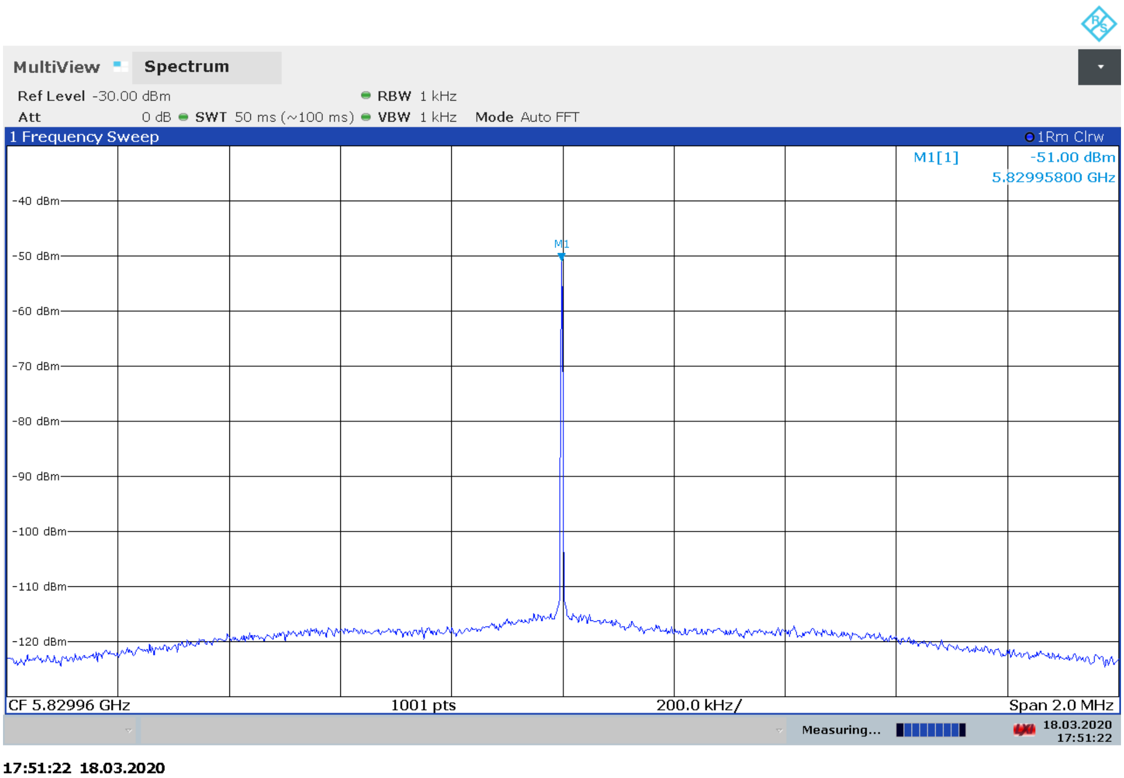


Figure 2.151: PlutoSDR transmitter spectrum at 5830 MHz, power -40 dBm.

Although the spectra on Figures 2.140 to 2.151 look nice and clean, we noticed several spurs occurring when at a wider span. Figure 2.152 shows a wider spectrum using 20 MHz span around the CW carrier. We believe that the first spur on the left side of the CW carrier is the DC component, the second being an image, and the third being a mix product between the carrier and its image. If this is so, these artifacts can be corrected through software.

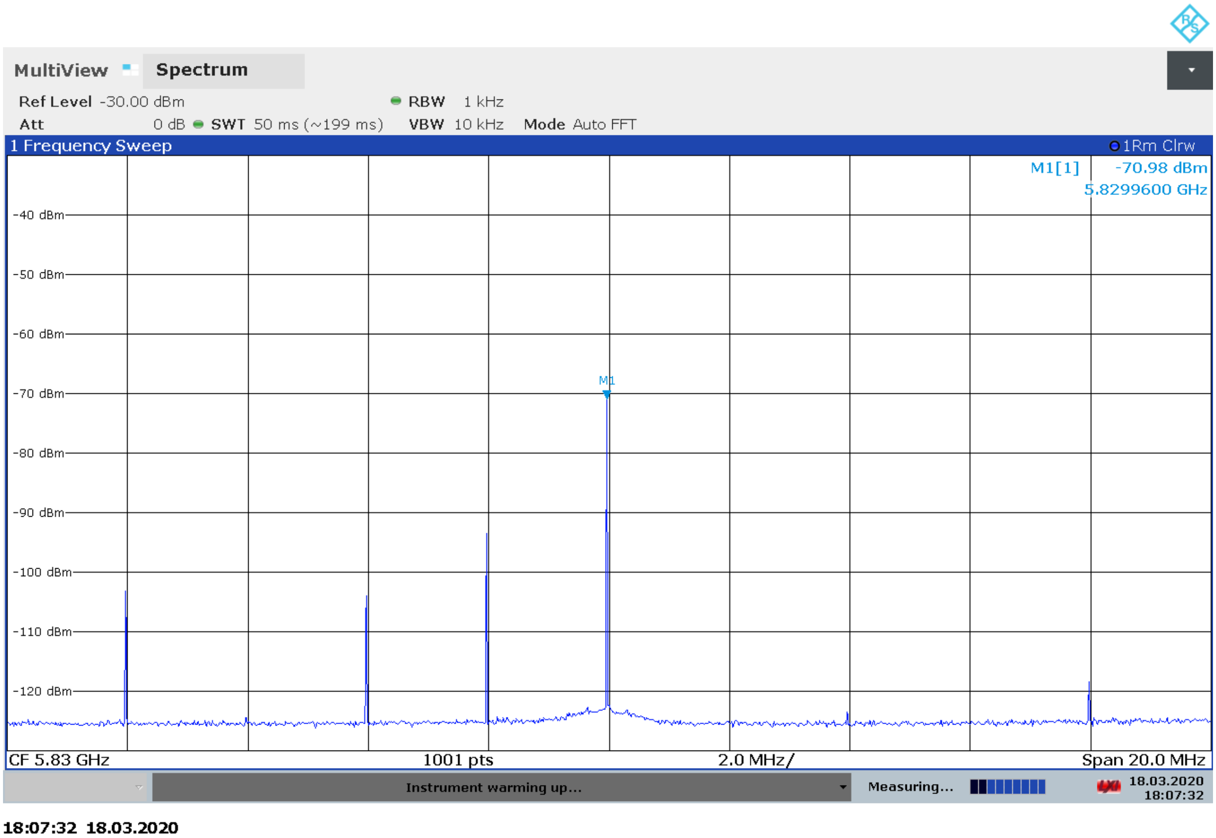


Figure 2.152: PlutoSDR transmitter spectrum at 5830 MHz with 20 MHz span.

2.2.7.5 Transmitter Output Power

As mentioned in section 2.1.5, we were unable to get a useful CW signal out of the PlutoSDR using GNU Radio and the gr-soapy interface. Instead, we used a new PlutoSDR software tool called SATSAGEN, which turns the PlutoSDR into a tracking spectrum analyzer and a signal generator. Unlike our GNU Radio scripts, the user is required to enter a desired power output in dBm, so in this case we do not know the exact relationship between PlutoSDR gain setting and output power.

The measured transmitter output for the PlutoSDR is shown on Figure 2.153 below. As we can see there is a linear relationship between the SATSAGEN power setting setting and the output power.

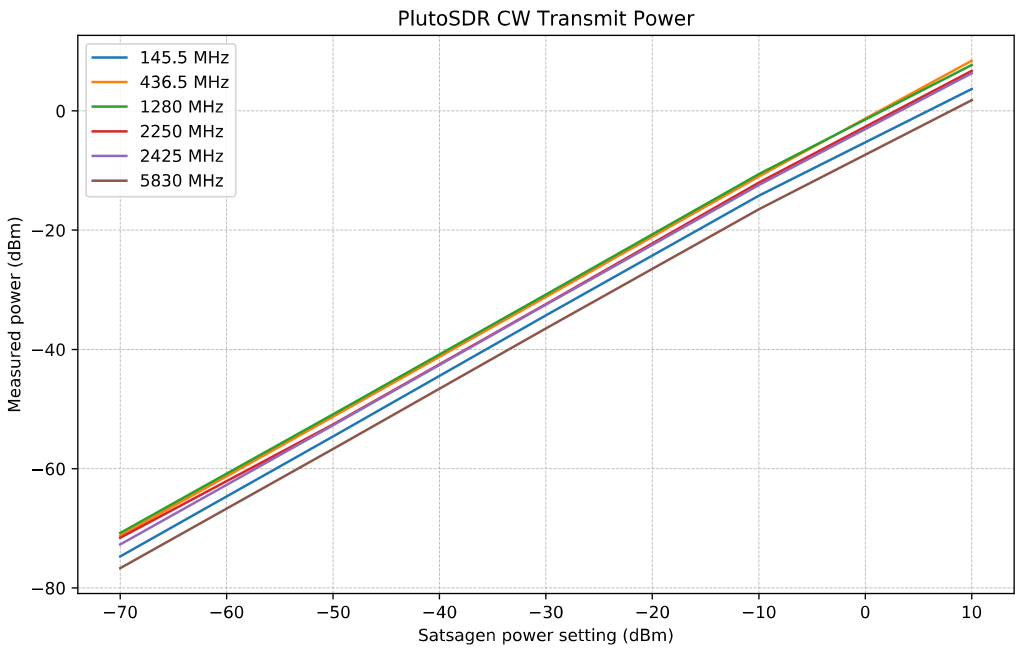


Figure 2.153: PlutoSDR output power as function of Satsagen power setting.

2.2.7.6 Transmitter MER

Since our GNU Radio and gr-soapy based DVB-S2 transmitter did not work with the PlutoSDR, we could not perform the transmitter MER measurements for this device.

2.2.8 Device Comparisons

2.2.8.1 Lowest Noise Figure

Table 2.25 below shows the lowest noise figure measured for each SDR device in each frequency band. Although the lowest noise figure of an SDR device is often not relevant in practical applications because

comparing it to the noise figure of the frontend component used in the device might give an indication of how well the RF front end is matched at the frequencies of interest:

- The RTL-SDR and Airspy Mini use the R820T2 tuner, which has a noise figure of 3.5 dB below 1002 MHz.
- The SDRplay RSPduo has filters and LNA in front of the tuner. Notch filters are placed in front of the LNA while band pass filters are placed between the LNA and the tuner. Both the FM and the DAB notch filters have higher insertion loss at 145 and 437 MHz than what we measured, so these filters are presumably off by default and the measured noise figure is that of the LNA plus switching circuitry in front of the LNA.
- The LimeSDR Mini is based on the LMS7002M RFIC which itself has 2 dB noise figure. The values we measured are far from it.
- The USRP B210, BladeRF 2.0 micro, and the PlutoSDR are all based on the AD936x RFIC, which has 2 dB noise figure below 1 GHz and 3.8 dB at 5.5 GHz.

Device	Lowest Noise Figure (dB)					
	145 MHz	437 MHz	1.28 GHz	2.25 GHz	2.425 GHz	5.83 GHz
RTL-SDR	5.8	5.4	11.9	-	-	-
Airspy Mini	7.1	7.4	10.7	-	-	-
SDRplay RSPduo	2.1	2.5	5.4	-	-	-
LimeSDR Mini	11.1	13.8	15.6	-	-	-
BladeRF 2.0 micro	5.0	4.9	7.4	9.0	9.5	15.8
USRP B210	5.5	4.4	5.7	4.8	6.3	9.6
PlutoSDR	6.2	6.1	4.1	4.7	5.0	9.3

Table 2.25: Lowest measured noise figure.

2.2.8.2 Dynamic Range

A way to compare the dynamic range of the tested devices without having a figure of merit such as the third order intercept point, it to plot the maximum input power the receiver can tolerat vs. the noise figure. This gives a good indication of which device has the highest dynamic range at any given noise figure. Figures 2.154 to 2.156 below show this data for all devices at 437 MHz.

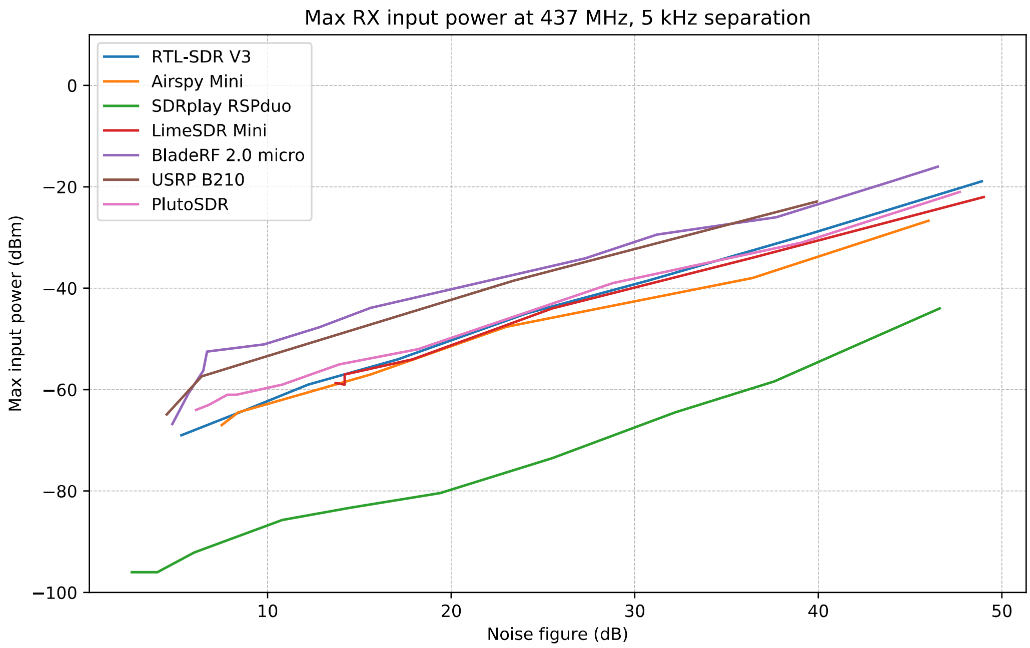


Figure 2.154: Max input power vs. noise figure at 5 kHz separation.

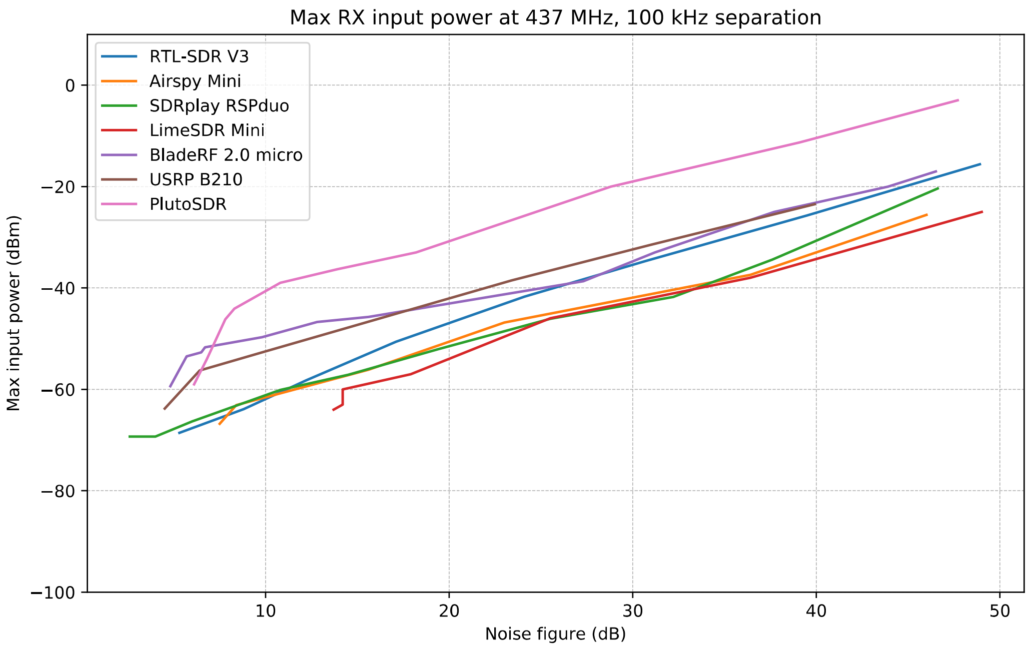


Figure 2.155: Max input power vs. noise figure at 100 kHz separation.

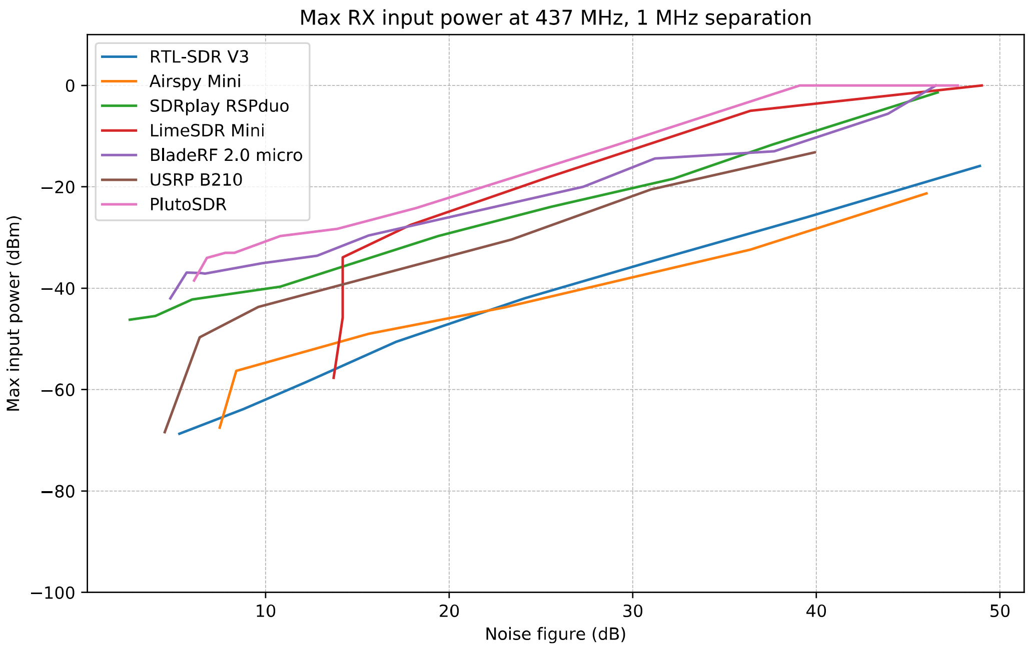


Figure 2.156: Max input power vs. noise figure at 1 MHz separation.

2.2.8.3 Transmitter Power

Table 2.26 below shows the highest measured CW power transmitted by each SDR device in each frequency band. Note that this is the highest power at maximum gain setting, not taking into account any spurious emissions the devices might be emitting at this setting.

Device	Output Power (dBm)					
	145 MHz	437 MHz	1.28 GHz	2.25 GHz	2.425 GHz	5.83 GHz
LimeSDR Mini	16.7	15.9	13.2	7.9	-	-
BladeRF 2.0 micro	7.7	6.9	5.0	1.6	0.1	-7.2
USRP B210	17.5	17.6	18.3	13.1	11.7	12.5
PlutoSDR	3.7	8.4	7.7	6.7	6.3	1.8

Table 2.26: Highest measured output power.

2.3 Test Setup Verification

Since our test setups and methods were relatively simple, we carried out several cross-checks to ensure that our measurements are reliable:

1. We used two different methods to measure the noise figure: MDS and Y-factor.
2. We used two different signal generators and verified that they provide the same output power.
3. We cross-checked the noise figure measurements using another SDR software (SDR Radio V3).

2.3.1 Signal Generator Output Power Comparison

We used two different signal generators during the MDS measurement:

1. Marconi Instruments 2031 that can support measurements up to 2.7 GHz
2. Rohde & Schwartz SMF 100A generator that covers all frequency bands of interest

The Marconi 2031 is of older date with last calibration date in 2008 whereas the SMF 100A is being calibrated at the prescribed intervals at a certified calibration facility. Although the Marconi 2031 was only used for a short time period during the initial noise figure measurements, we decided to compare the two generators against each other in order to ensure that we could use them interchangeably, should that be necessary.

The comparison was carried out by connecting the output of the signal generator to a Rohde & Schwartz FSP3 spectrum analyzer using the following settings:

- Span: 10 kHz
- Resolution bandwidth: 1 kHz
- Video bandwidth: 10 kHz (auto)

The generators were connected to the spectrum analyzer using a 141-24SMNM+ coaxial cable from Mini Circuits with the insertion losses listed in Table 2.27 below.

Frequency	Insertion Loss
145.5 MHz	0.09 dB
437.5 MHz	0.12 dB
1280 MHz	0.23 dB
2250 MHz	0.33 dB
2450 MHz	0.35 dB

Table 2.27: Coaxial cable insertion loss.

These losses are taken into account in the measurements shown in the tables below. The measurements done at -100 dBm signal power are more uncertain than the others since this power level is rather close to the noise floor of the spectrum analyzer.

Generator setting (dBm)	Measured Power (dBm)		Difference (dBm)
	R&S SMF 100A	Marconi 2031	
-100	-100.2	-100.1	0.10
-80	-80.29	-80.39	0.10
-60	-60.27	-60.41	0.14
-40	-40.27	-40.38	0.11
-20	-20.28	-20.39	0.11

Table 2.28: Signal generator output power comparison at 145.5 MHz.

Generator setting (dBm)	Measured Power (dBm)		Difference (dBm)
	R&S SMF 100A	Marconi 2031	
-100	-100.3	-100.1	0.20
-80	-80.52	-80.62	0.10
-60	-60.55	-60.67	0.12
-40	-40.56	-40.68	0.12
-20	-20.56	-20.70	0.14

Table 2.29: Signal generator output power comparison at 437.5 MHz.

Generator setting (dBm)	Measured Power (dBm)		Difference (dBm)
	R&S SMF 100A	Marconi 2031	
-100	-100.5	-101.2	0.70
-80	-80.83	-80.93	0.10
-60	-60.83	-61.04	0.21
-40	-40.82	-40.97	0.15
-20	-20.82	-20.97	0.15

Table 2.30: Signal generator output power comparison at 1280 MHz.

Generator setting (dBm)	Measured Power (dBm)		Difference (dBm)
	R&S SMF 100A	Marconi 2031	
-100	-100.8	-101.3	0.50
-80	-81.03	-81.05	0.02
-60	-61.03	-61.11	0.08
-40	-41.07	-41.10	0.03
-20	-21.05	-21.08	0.03

Table 2.31: Signal generator output power comparison at 2250 MHz.

Generator setting (dBm)	Measured Power (dBm)		Difference (dBm)
	R&S SMF 100A	Marconi 2031	
-100	-100.8	-101.3	0.50
-80	-81.13	-81.15	0.02
-60	-61.12	-61.23	0.11
-40	-41.12	-41.21	0.09
-20	-21.13	-21.18	0.05

Table 2.32: Signal generator output power comparison at 2425 MHz.

2.3.2 Test Software Verification

In order to verify that our test software used for noise figure and dynamic range measurements works correctly, we cross-checked the noise figure measurements using SDR Radio V3. The cross-checks were done with the Airspy Mini using the Y-factor method on 145.5 MHz and the results are shown on Figure 2.157 below. As we can see from the graph, the two measurement sets are virtually identical.

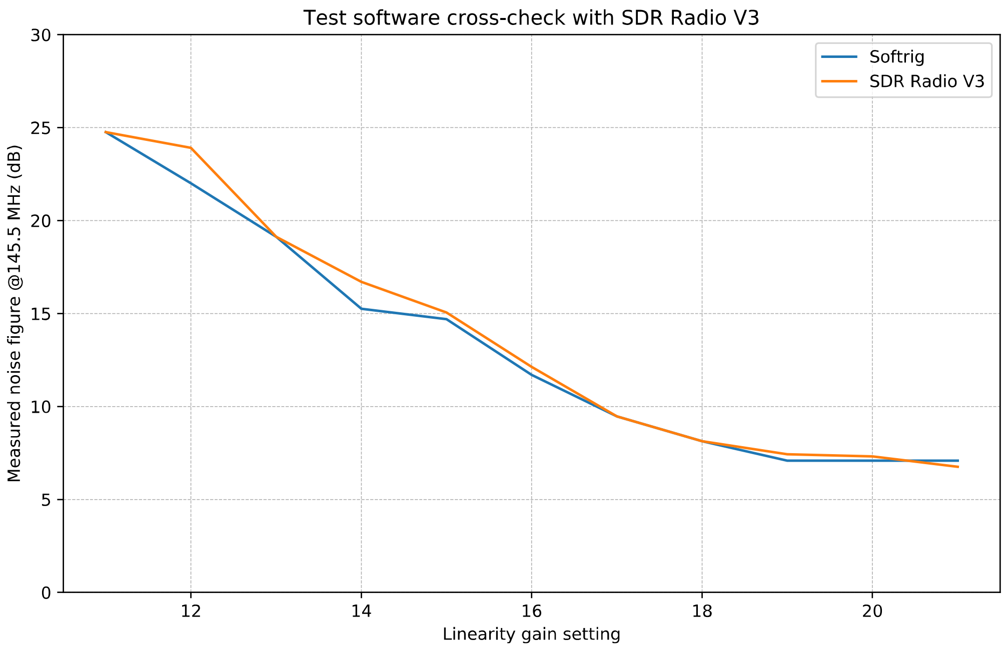


Figure 2.157: Noise figure measurements for cross-checking the test software.

3 On-Air Tests

Based on the technical specifications and the results obtained from the lab measurements, two SDR devices were selected for further side-by-side performance testing over the air:

1. SDRplay RSPduo, a low cost dual-tuner receiver, which performed quite well during the lab tests. The two independent tuners in the device makes it an appealing option for receiving satellite telemetry simultaneously on two bands.
2. Ettus USRP B210, the most expensive device on our list with dual transceiver capabilities.

3.1 Test Setup

The two devices were installed in a SatNOGS setup [33] at the AMSAT OZ premises. The devices were configured to run as separate SatNOGS ground stations, namely station 1353 (USRP) [34] and station 1354 (SDRplay) [35], sharing the same antenna and receiving the same satellite passes. This way the performance of the two SDR devices can be compared based on the number of packets received and decoded during the test campaign.

Figure 3.1 shows a diagram of the on-air test setup. After installation, the stations were running for 24 hours during which the device settings were adjusted for best performance. Once the optimal settings were found, the stations were left running unattended for a few days, just like they would run in a regular SatNOGS ground station setup. The final settings that were used during the the on-air tests are listed in Table 3.1 below.

	Station 1353 **USRP B210**	**Station 1354** **SDRplay RSPduo**
SATNOGS_RX_SAMP_RATE	2 MHz	2 MHz
SATNOGS_RX_BANDWIDTH	600 kHz	600 kHz
SATNOGS_RF_GAIN	50	34
SATNOGS_ANTENNA	RX2	Tuner 1 50 ohm

Table 3.1: SDR device settings used during the on-air tests.

The ground stations were running the latest GNU Radio 3.8 based SatNOGS client software available at the time of testing:

- satnogs-client git revision d9ff0e8509df12889d25bf4c1d8403c798d3ce04
- gr-satnogs git revision 0ba9e0e601f1fe561ba4d74f349bb45530fa58c7
- gr-soapy git revision a7399c5e7b586c90f971b47280ac92eb64349bc8

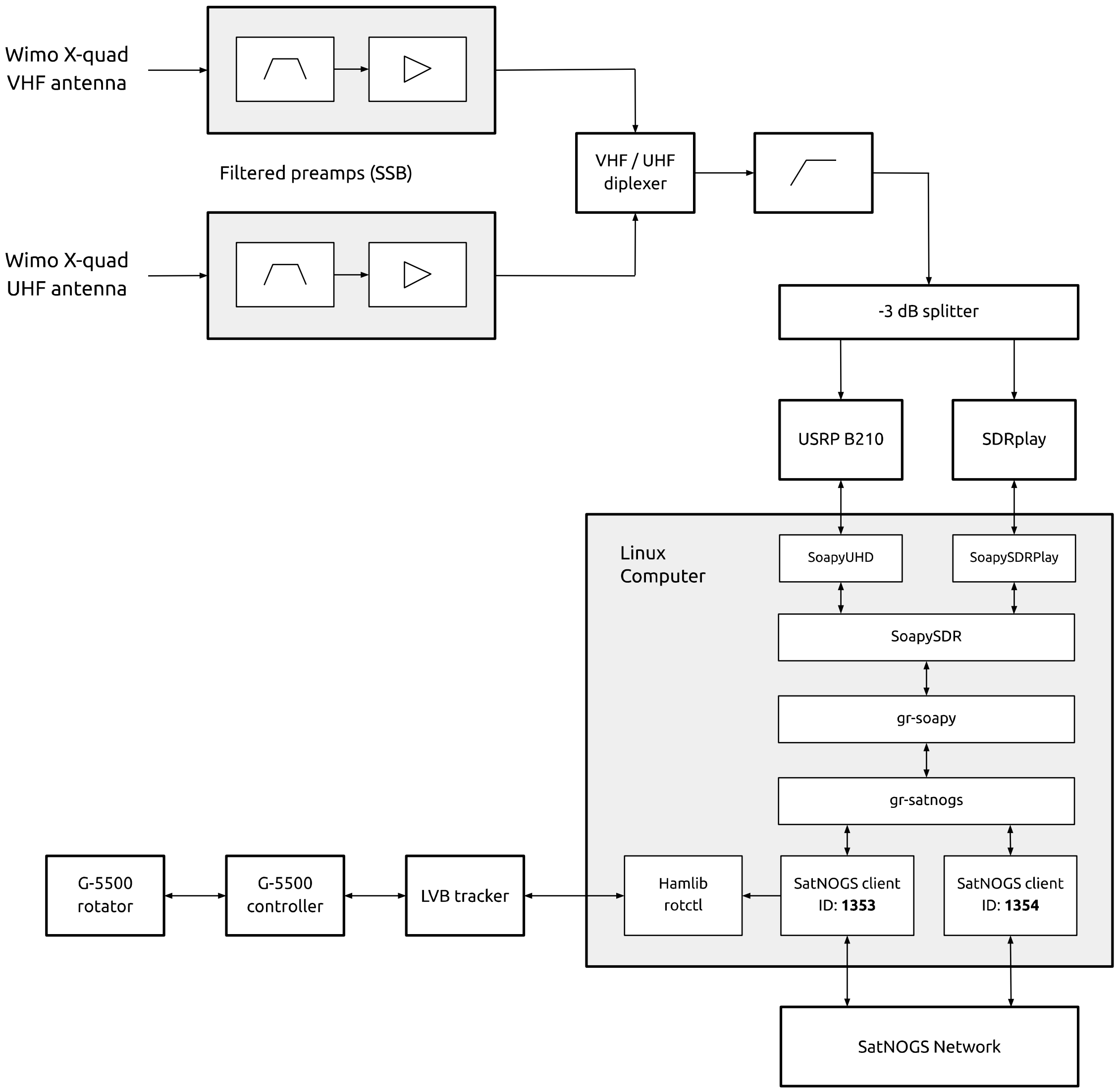


Figure 3.1: On-air test setup.

3.2 Test Results

The final on-air tests were run over a period of 48 hours beginning at 2020-03-22 00:00 UTC until 2020-03-23 23:59 UTC. The results are listed in Table 3.2 below showing the satellite name, the nominal downlink frequency, modulation, and the number of packets received by the two ground stations during the 48 hour test period.

Satellite	Frequency MHz	Modulation	Packets (USRP B210)	Packets (SDRplay RSPduo)
OPS-SAT	437.200	GMSK 9600	1483	1537
UNISAT-6	437.421	FSK 9600	2180	2271
TIGRISAT	435.001	FSK 9600	1010	999
CubeBel-1	436.990	FSK 9600	209	218
MCUBED-2	437.480	FSK 9600	134	139
BUGSAT-1	437.445	FSK 9600	187	191
FOX-1B	145.960	DUV 200	246	220
FOX-1D	145.880	DUV 200	192	181
		Total	**5641**	**5756**

Table 3.2: Number of packets decoded during the on-air test campaign.

As can be seen from the results, the two receivers performed more or less equally, with the SDRplay RSPduo having received about 2% more packets the the USRP B210. However, this difference is rather low and would likely even out if the test campaign would be running over a longer time period. It is still very impressive that in a SatNOGS setup, a low cost SDRplay performs equally well as the higher cost USRP B210. On the other hand, we have to remember that the USRP is a wide-band dual transceiver with frequency coverage up to 6 GHz, whereas the SDRplay cover only up to 2 GHz and has to some extent been optimized for reception in the amateur radio bands.

4 Software Support

In this chapter we give a brief overview of the most popular SDR software available and to which extent they suport the SDR devices evaluated during this activity. We have divided the SDR software into three categories:

1. End user applications – relevant to people who just want to use the SDR hardware with basic functionality avaialble in common SDR software. In this category we have included:

- SDR-Radio V3
- SDR#
- Gqrx
- CubicSDR
- SDR Angel

2. SDR development frameworks – relevant to people who want to write SDR applications in a convenient, high-level environment that does not require programming skills. In this category we have included:

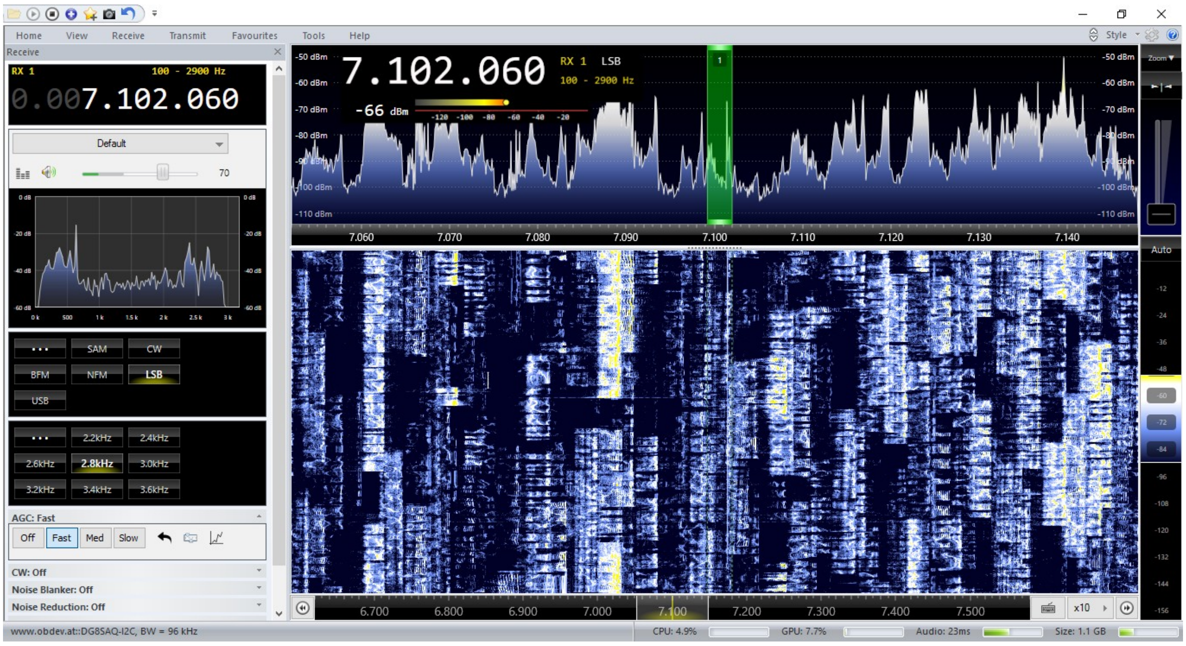
- GNU Radio
- Pothos SDR

3. SDR API wrappers – relevant to people who want to write SDR applications from the ground up but without worrying about interfacing to different SDR device drivers. SDR API wrappers provide a uniform programming interface to most commonly available SDR hardware. In this category we have included:

- SoapySDR
- gr-soapy
- gr-osmosdr

4.1 SDR-Radio V3

Software type:	Receive, transmit, satellite tracking
Supported OS:	Windows
License:	Closed source commercial (free for amateur radio use)
Hardware support:	RTL-SDR, Airspy, SDRplay, LimeSDR, USRP, BladeRF, PlutoSDR
Website:	https://www.sdr-radio.com/
Description:	SDR-Radio V3 is one of the most popular SDR applications for the Windows operating system. It supports most of the SDR hardware available on the market and is one of the few end user SDR applications with built-in transmit support. In addition to providing all of the basic radio functionality needed, it also has built-in satellite tracking and Doppler tuning functionality.



4.2 SDR#

Software type: Receive

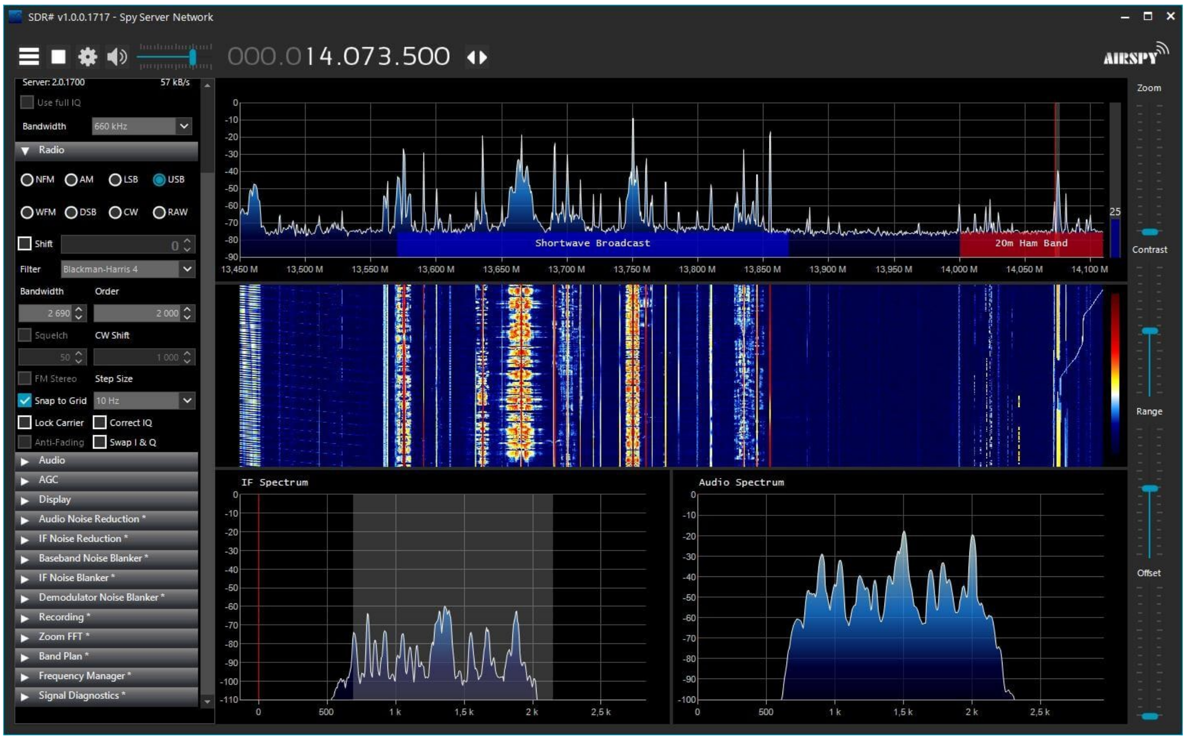
Supported OS: Windows

License: Closed source

Hardware support: RTL-SDR, Airspy, others might be supported via plug-ins

Website: https://airspy.com/download/

Description: SDR# is another very popular SDR application for Windows. One of its core features is plug-in support allowing third parties to extend the functionality of the application without having access to the source code. Officially SDR# only supports RTL-SDR and Airtspy from the list of tested devices, support for other devices might be available throgh plug-ins.
SDR# does not have built-in satellite tracking functionality, however, plug-ins exist to interface to third party satellite applications such as Gpredict.



4.3 Gqrx

Software type: Receive

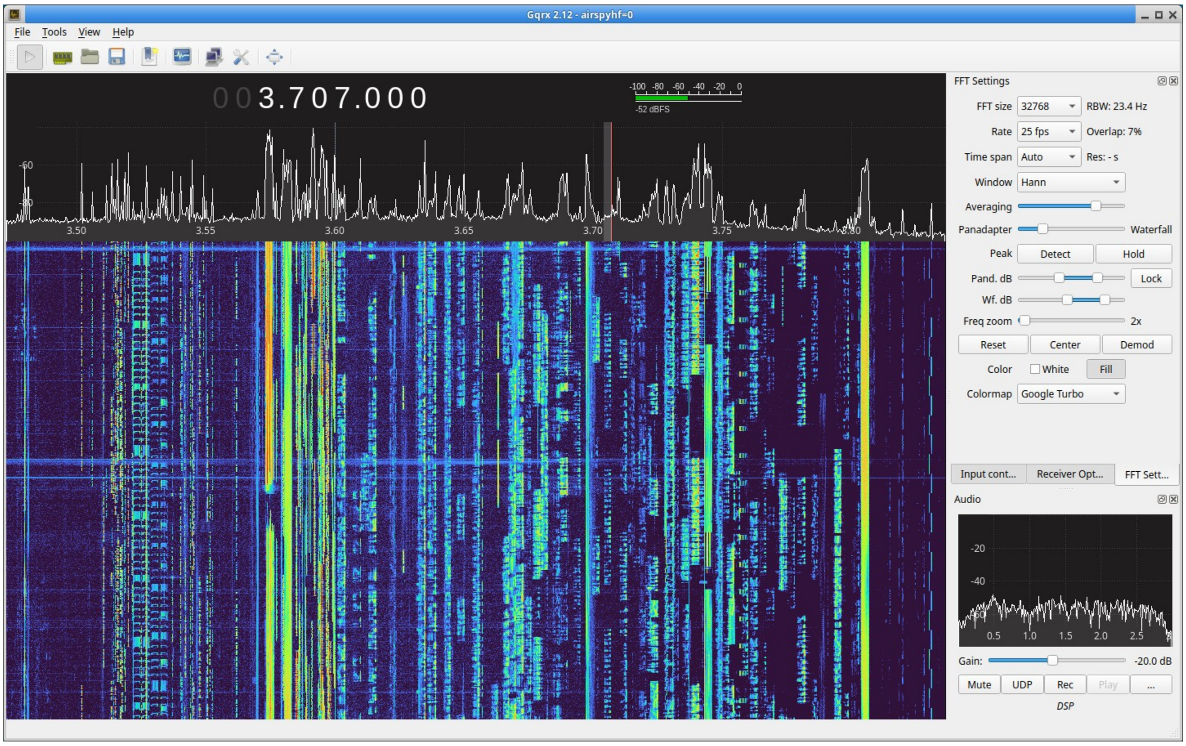
Supported OS: Linux, Mac OS X, Windows

License: Open source GPL v2

Hardware support: RTL-SDR, Airspy, LimeSDR, USRP, BladeRF, PlutoSDR

Website: https://gqrx.dk/

Description: Gqrx is a simple SDR application written in C++ using the Qt and GNU Radio toolkits. Although it is primarily targeted to the Linux operating system, binary packages are also available for Mac OS X and Windows. Gqrx can interface to third party satellite tracking applications, such as Gpredict, through its network based remote control control interface. Gqrx uses the gr-osmosdr SDR driver wrapper for accessing SDR devices and will therefore work with all devices supported by gr-osmosdr and SoapySDR.



4.4 CubicSDR

Software type: Receive

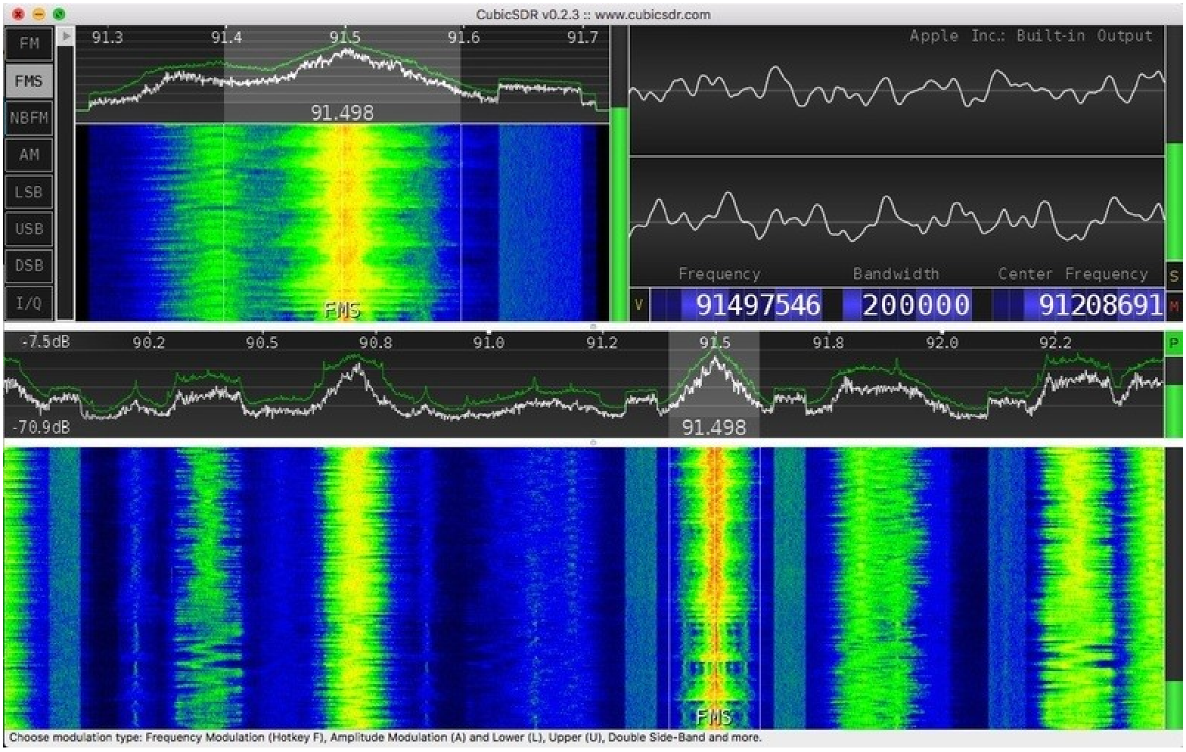
Supported OS: Linux, Windows, Mac OS X

License: Open source GPL v2

Hardware support: RTL-SDR, Airspy, SDRplay, LimeSDR, USRP, BladeRF, PlutoSDR

Website: https://cubicsdr.com/

Description: CubicSDR is another popular, cross-platform SDR application with basic spectrum view and demodulation functionality. CubicSDR uses the SoapySDR SDR driver wrapper for accessing SDR devices and will therefore work with all devices supported by SoapySDR. We are not aware of any built-ion functionality or external interface to provide satellite tracking and Doppler tuning functionality.



4.5 SDR Angel

Software type: Receive, transmit

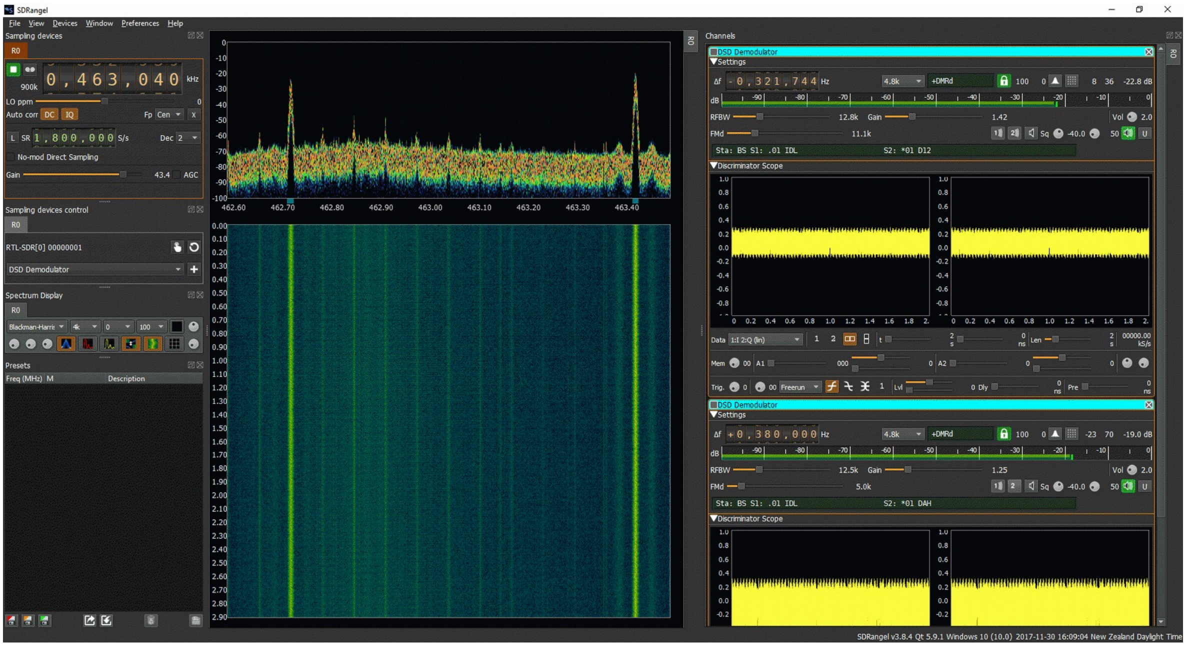
Supported OS: Linux, Windows

License: Open source GPL v3

Hardware support: RTL-SDR, Airspy, SDRplay, LimeSDR, USRP, BladeRF, PlutoSDR

Website: https://github.com/f4exb/sdrangel

Description: SDR Angel is another popular SDR application available for both Linux and Windows. In addition to the basic spectrum view and demodulator functionalities, it also provides more advanced SDR functions such as signal diagnostics, digital voice, video demodulation, and transmitting. SDR Angel does not use any SDR device wrapper libraries, nontheless it has built-in support for most of the commonly available SDR devices, including the ones tested in this activity.



4.6 GNU Radio

Software type: SDR development framework

Supported OS: Linux, Windows, Mac OS X

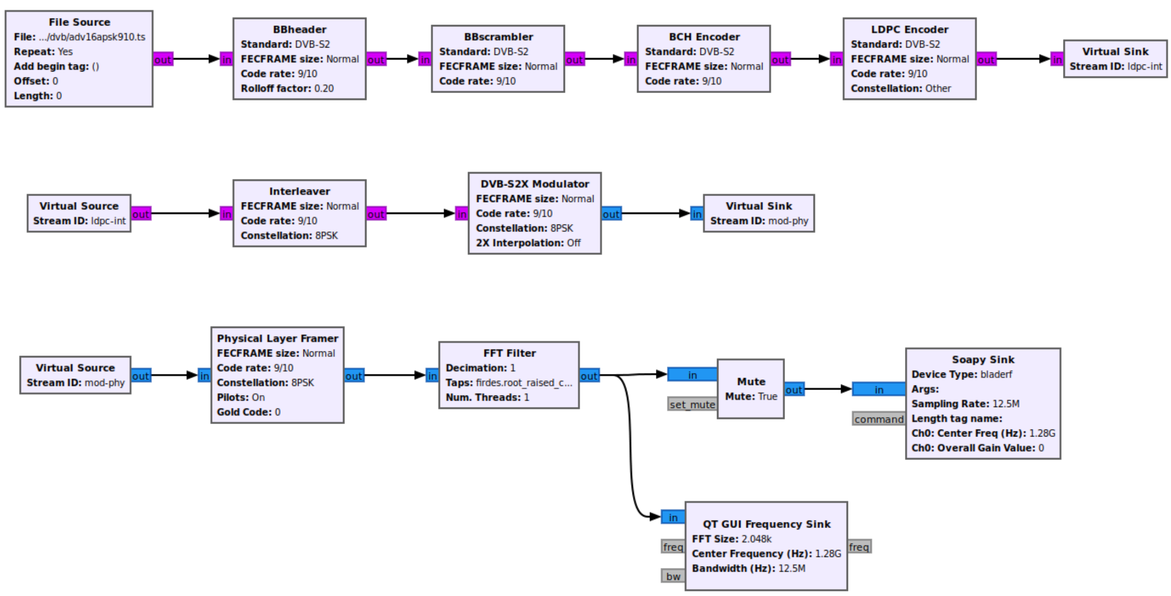
License: Open source GPL v3

Hardware support: RTL-SDR, Airspy, SDRplay, LimeSDR, USRP, BladeRF, PlutoSDR

Website: https://www.gnuradio.org/

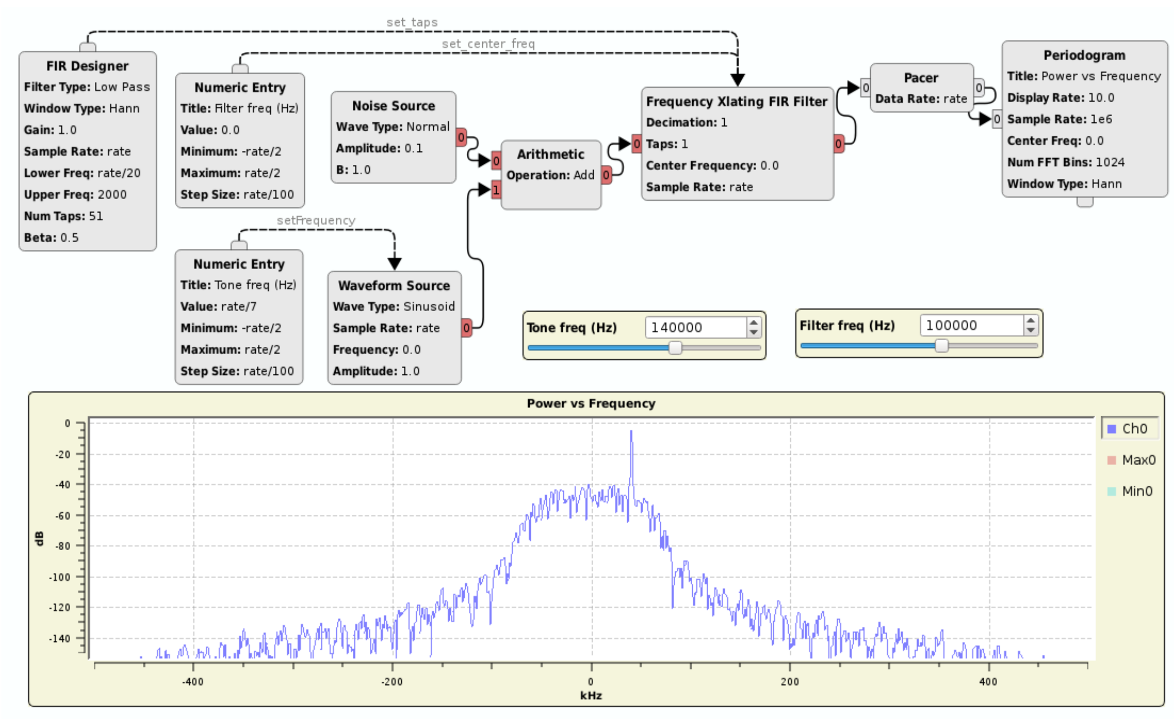
Description: GNU Radio is a very popular SDR development framework providing a convenient graphical environment for creating SDR applications. It includes blocks for creating signal processing pipelines with or without a graphical user interface, as well as composite blocks for more advanced functionality such as digital television.

SDR hardware support in GNU Radio is available through both SDR driver wrappers like gr-osmosdr and gr-soapy, as well as vendor specific out of tree modules like gr-uhd, gr-limesdr, and gr-iio.



4.7 Pothos SDR

Software type:	SDR development framework
Supported OS:	Linux, Windows
License:	Open source, Boost Software License 1.0
Hardware support:	RTL-SDR, Airspy, SDRplay, LimeSDR, USRP, BladeRF, PlutoSDR
Website:	https://github.com/pothosware/PothosCore/wiki
Description:	Pothos SDR os a graphical SDR development framework similar to GNU Radio. In addition to providing the framework for creating SDR applications in a convenient graphical environment, Pothos supports execution of applications in a distributed environment, including networked hosts and hardware accelerators such as FPGAs and GPUs. The primary SDR hardware support in Pothos is provided by SoapySDR, therefore it supports most of the commonly available SDR devices.



4.8 SoapySDR

Software type:	SDR device API
Supported OS:	Linux, Mac OS X, Windows
License:	Open source, Boost Software License 1.0
Hardware support:	RTL-SDR, Airspy, SDRplay, LimeSDR, USRP, BladeRF, PlutoSDR
Website:	https://github.com/pothosware/SoapySDR/wiki
Description:	SoapySDR provides a vendor-neutral interface to SDR devices available through C, C++ and Python API. SDR device support in SoapySDR is provided through plugins making it easy to add new hardware support to any application that uses SoapySDR as its device interface.

4.9 Gr-soapy

Software type:	SDR device API
Supported OS:	Linux, Mac OS X, Windows
License:	Open source GPL v3
Hardware support:	RTL-SDR, Airspy, SDRplay, LimeSDR, USRP, BladeRF, PlutoSDR
Website:	https://gitlab.com/librespacefoundation/gr-soapy
Description:	Gr-soapy is a GNU Radio out-of-tree module, making SDR hardware supported by SoapySDR available to GNU Radio applications. The initial version of gr-soapy was developed as part of the SDR Makerspace project and has since been maintained by Libre Space Foundation as part of the SatNOGS client software ecosystem.

4.10 Gr-osmosdr

Software type: SDR device API

Supported OS: Linux, Mac OS X, Windows

License: Open source GPL v3

Hardware support: RTL-SDR, Airspy, LimeSDR, USRP, BladeRF, PlutoSDR

Website: https://osmocom.org/projects/gr-osmosdr/wiki

Description: Gr-osmosdr is the Osmocom GNU Radio block prividing a uniform interface to GNU Radio applications an thus has the same role as SoapySDR + gr-soapy. The most significant difference compared to SoapySDR + gr-soapy is that SDR device support in gr-osmosdr is added at compile time rather than runtime loadable plugins. This makes the framework somehow more difficult to distribute to end users in that distribution packages must be accompanied by the driver libraries in order to satisfy the dependencies that have been added during compile time. Because of this and the fact the SDRplay driver are closed source, most distributions of gr-osmosdr do not include SDRplay support.

5 Conclusion

We have evaluated seven SDR devices, the RTL-SDR V3 dongle, the Airspy Mini, the SDR-play RSPduo, the LimeSDR Mini, the BladeRF 2.0 micro, the USRP B210, and the PlutoSDR. These devices were chosen based on the scope of the SDR Makerspace project, namely satellite communications. During our evaluation, we have measured the receiver noise figure, the receiver dynamic range, the transmitter output power, and achievable transmitter modulation error ratio. The measurements were carried out under realistic conditions that are representative for an RF environment in which these devices are typically used.

Our tests have provided us with key performance data that is not readily available for most SDR devices under test. In fact, out of the seven tested devices only Ettus Research and SDR-play have published detailed performance data for their devices. For these devices our measurements are in good agreement with the published data.

We have defined some relatively simple test setups and test procedures for the receiver tests that involve only a signal generator, then cross checked the measurements using a calibrated noise source. The procedures turned out to be sufficiently accurate for the purpose of comparing the performance of wide band SDR devices. During the test campaign we have also developed open-source test software to support the test procedures.

Two of the SDR devices under test, SDRplay RSPduo and USRP B210, have also been tested on-air in a SatNOGS satellite telemetry receiver setup. Both devices performed very well receiving virtually the same amount of packets.

Finally, we have taken a brief look at some of the most popular SDR software for Linux, Windows, and Mac OS X in order to evaluate how well the tested SDR devices are supported by existing software. All devices are well supported by most software with the exception of SDRplay, which due its closed-source driver only has limited support in pre-packaged Linux software.

6 Appendices

Appendix I: Insertion Losses

Frequency (MHz)	Loss (dB)
145	0.03
435	0.06
1280	0.10
2250	0.14
2425	0.15
5830	0.46

Table 6.1: Insertion loss for BLK-89-S+ DC block.

Frequency (MHz)	Loss (dB)
145	19.65
435	19.67
1280	19.71
2250	19.74
2425	19.74
5830	19.78

Table 6.2: Insertion loss for BW-S20W2+ attenuator.

Frequency (MHz)	Loss (dB)
145	0.09
435	0.12
1280	0.23
2250	0.33
2425	0.35
5830	0.57

Table 6.3: Insertion loss for 141-24SMNM+ coaxial cable.

Frequency (MHz)	Loss (dB)
145	0.12
435	0.21
1280	0.43
2250	0.59
2425	0.62
5830	1.00

Table 6.4: Insertion loss for FL086-24SM+ coaxial cable.