BreadBoard RF103

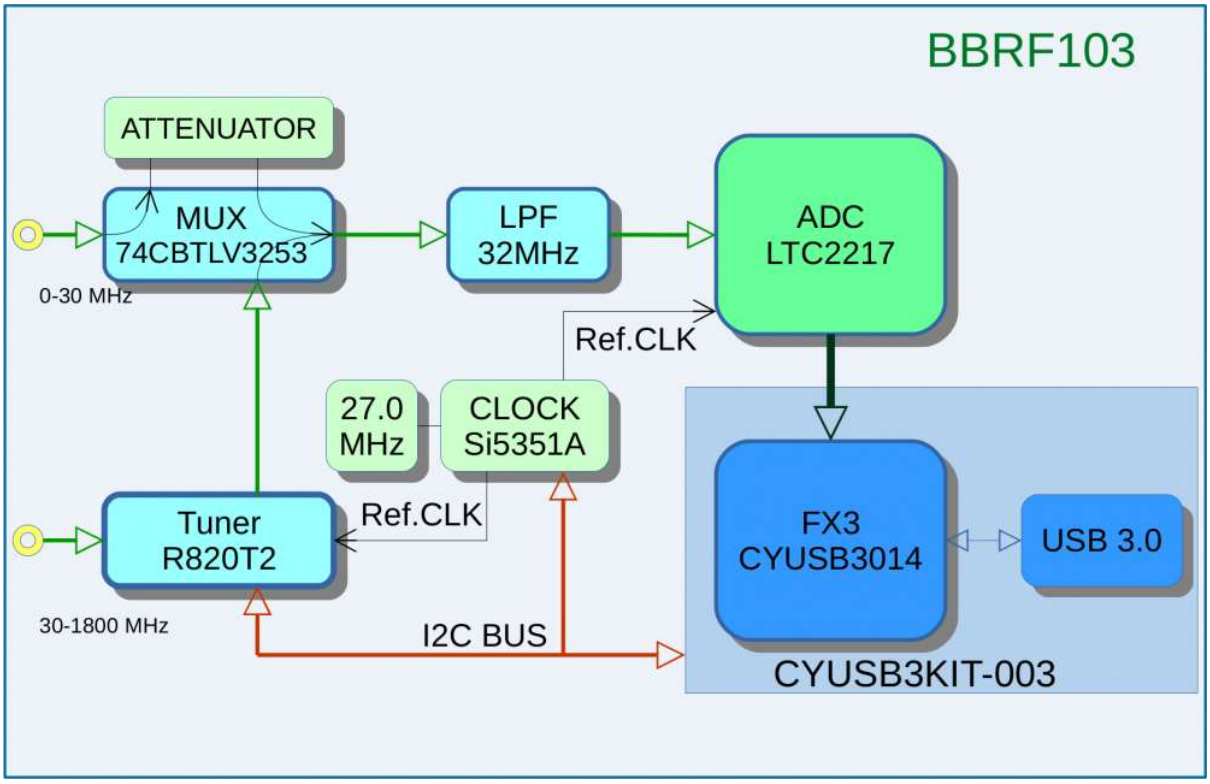
20 June, 2017

Twitter Facebook Google+ Linkedin

These are exciting times for homemade construction of Software Designed Radio (SDR). Our laptop and desktop have more computing power. Better compilers simplify multi thread programming. Computer interfaces run at higher throughput rate.

I designed the breadboard BBRF103 to learn how to use and to test the following components :

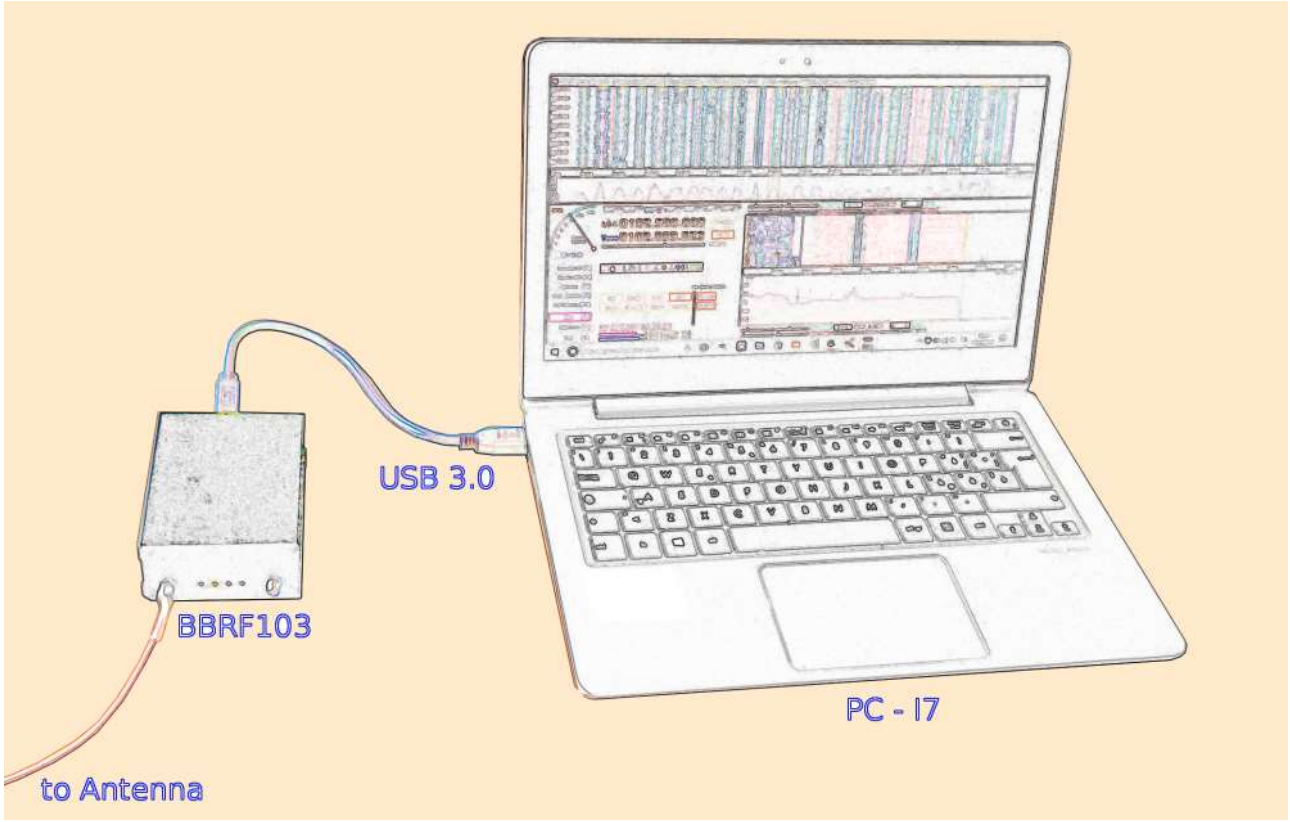
- FX3 SuperSpeed Explorer Kit USB3.0 transfers the ADC sample stream to the PC.
- ADC (LTC2217) samples the real data at 16 bit up to 105 Msps.
- 0-30MHz input, attenuator (0,-10,-20 dB) and LPF transfer antenna signal to the ADC.
- Tuner (R820T2) down converts signals in the 30-1800 MHz range to the ADC.
- Clock generator (Si5351A) outputs the clocks to the ADC and the R820T2.



In other words the idea is to avoid the Digital Down Converter (DDC) Custom or FPGA chip in between ADC and PC. The full HF radio spectrum is processed by the host computer connected via an USB3.0 port.

BBRF103 is placed in series between Antenna and Computer. A modern pc (I5-I7 CPU or higher) equipped with USB 3.0 is required.

The R820T2 chip has been added to look at its performance with a 16 bit ADC and wide bandwidth.



Hardware

The hardware uses two separate antenna connectors 0-30MHz and 30MHz-1.8GHz

I made the schematic using cut and paste of the main components test circuits.

The HF input (0-30 MHz) is routed to a multiplexer circuit. Some resistors implement a step attenuator with value 0, -10 , -20 dB. The attenuator's output goes to a low pass filter and then to the ADC input via a balancing transformer. The ADC parallel output bus is routed to the FX3 SuperSpeed Explorer Kit using the kit IO connectors. The Cypress kit uses some GPIOs as control of multiplexer and ADC while a I2C bus is used to program the Si5351a clock generator and the R820T2 tuner.

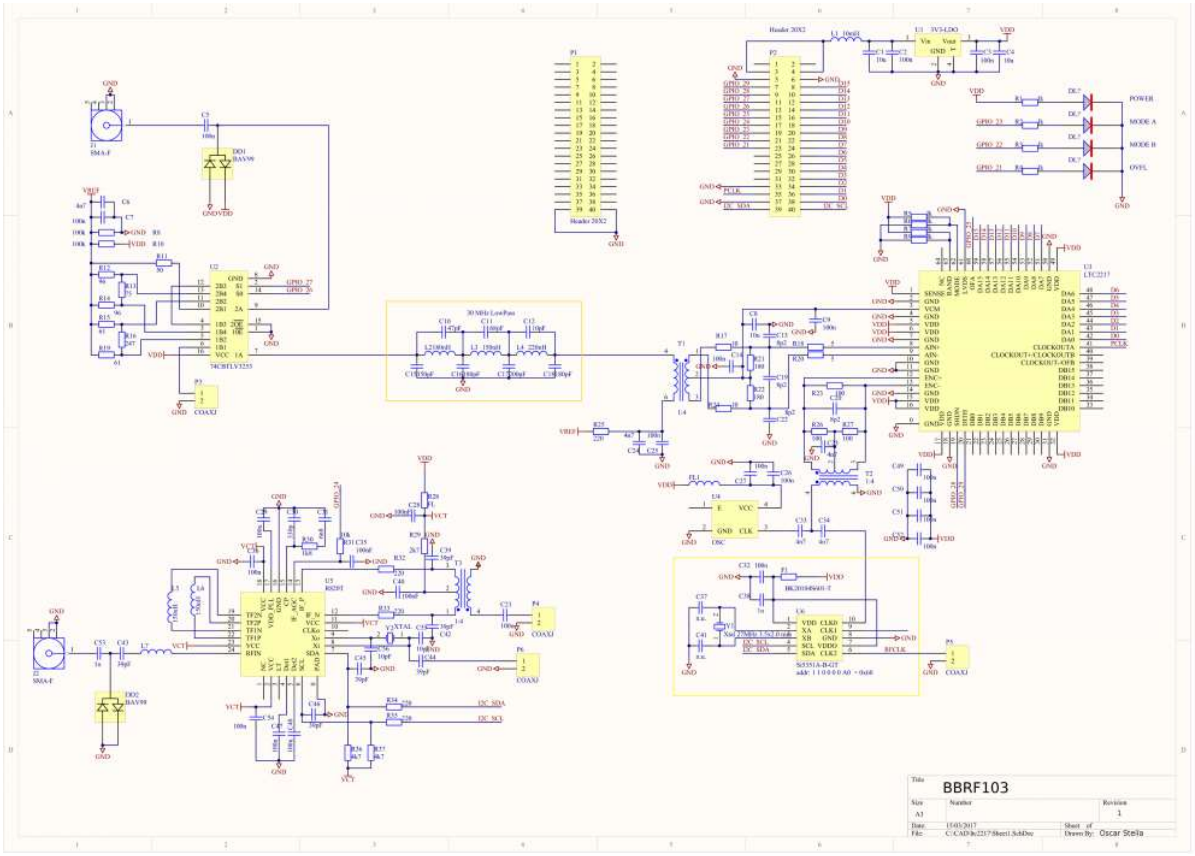
The input multiplexer other than the HF input selects the R820T2 tuner output.

The R820T2 uses an indipendent input (30MHz - 1.8GHz) connector. The Si5351a tuner generates by the tuner reference clock; the first software prototype setup uses a 32 MHz. The software may program different reference frequency to move out of band spur signals.

The Si5351a generates also the ADC clock . The pcb previews an optional backup alternative with a fixed frequency oscillator.

The Si5351a's reference Xtal is 27.000MHz. Another frequency may be used. This is the only frequency reference of all the hardware. The software will be able to compensate the accuracy of this xtal with a correction coefficient.

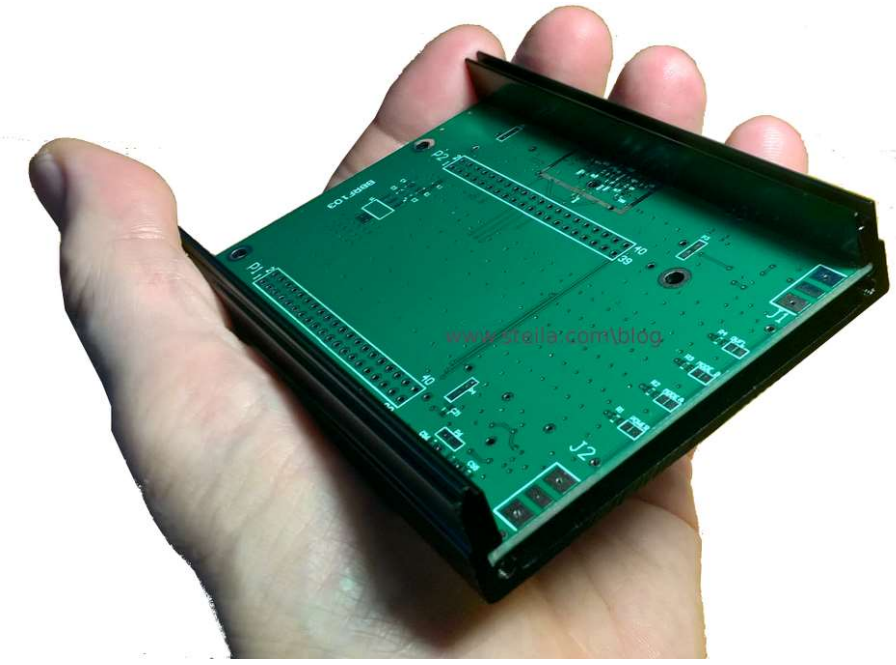
The clock is coupled to the ADC LTC2217 using a rf balancing transformer.



https://github.com/ik1xpv/BBRF103/blob/master/HARDWARE/BBRF103_scheme.pdf

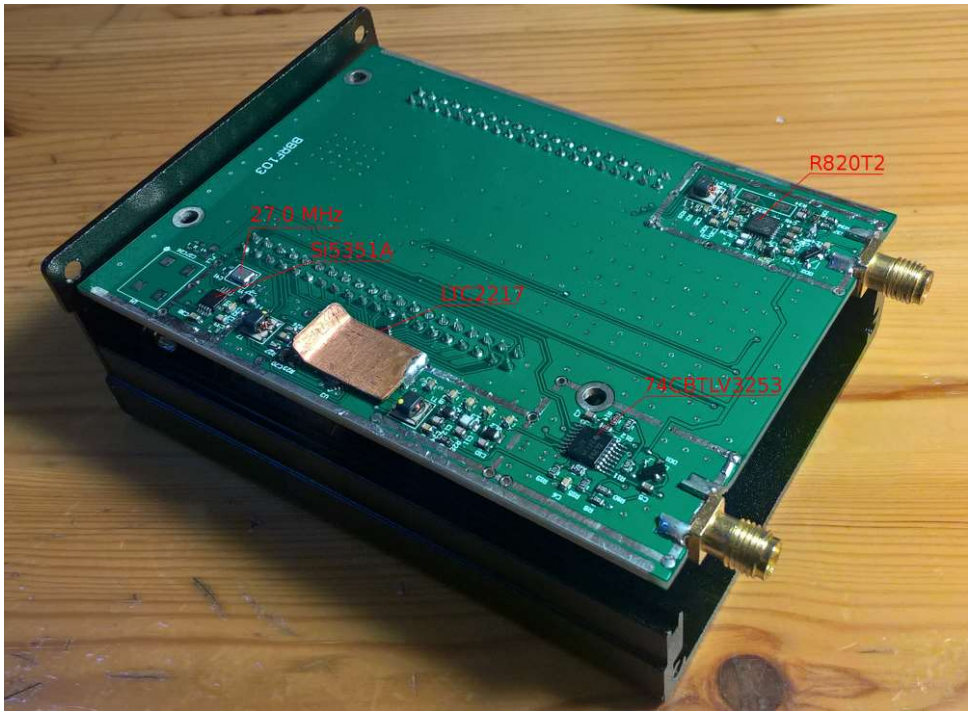
An extruded aluminum box of 100 * 76 * 35 mm is large enough to accommodate the FX3 SuperSpeed Explorer kit and board PCB.

The size of the PCB is about 100x70 mm. Two 40x2 headers connect the FX3 SuperSpeed Explorer kit.

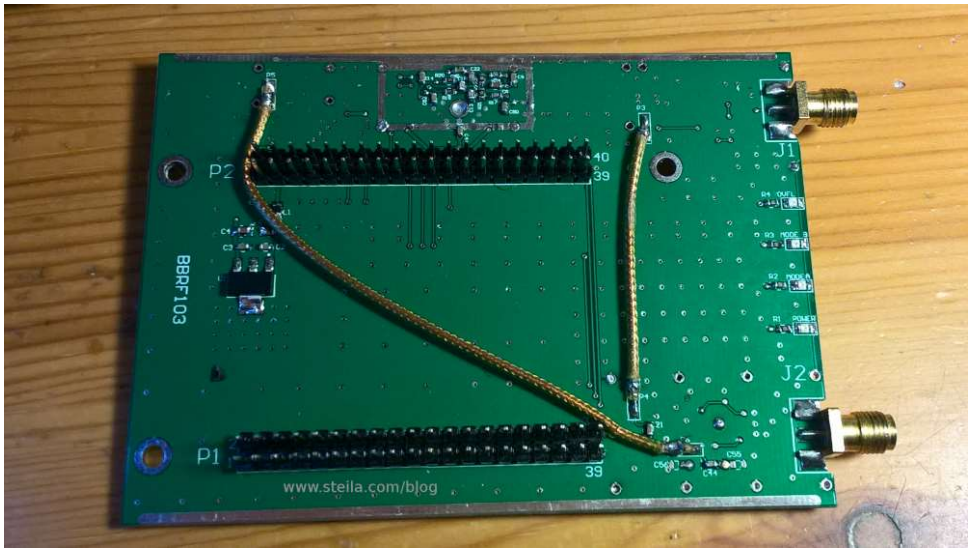


The PCB board contains the main components on the underside. The ADC has a copper radiator on the top. It taps the aluminum box to dissipate part of the ADC heat.

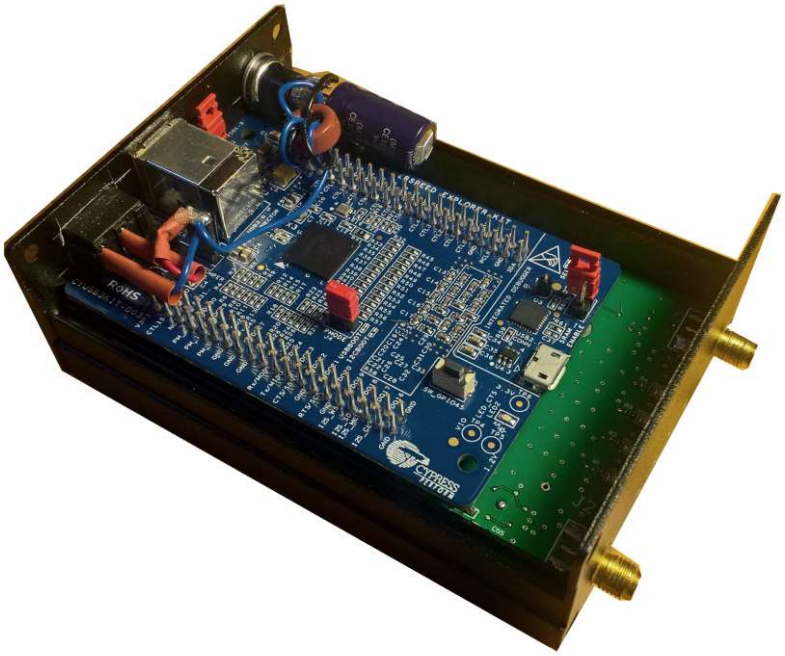
The two RF input connectors are SMA.



On the upper side there are the power regulator, pin connectors and two jumper cables in coaxial cable for the R820T2 clock and the IF signal.



The final assembly of the prototype shows the FX3 kit at the top of the BBRF103 board.



The prototype has a 5 volt Auxiliary Power Connector that was used during the first tests. It is not necessary because the required current is less than 800mA and can be supplied by the standard USB3.0 connection.

The hardware scheme and pcb layout at https://github.com/ik1xpv/BBRF103/tree/master/HARDWARE .

Firmware

The FX3 firmware is a modification of Cypress streaming examples. Some Vendor commands have been added to control I2C bus and to control GPIO e PWM output.

SDK comes with the Cypress Eclipse development enviroment. It's a nice exercise to learn how to use FX3.

BBRF103 uses the Cypress USB driver that comes with FX3 Kit.

The firmware source repository is https://github.com/ik1xpv/BBRF103/tree/master/FX3Firmware

Software

The prototype has been tested with the HDSDR application that i like a lot (THANKS to Mario Taeubel and Alberto di Bene).

I designed an ExtIO_sddc.dll. The name stands for ExtIO software digital down convertion. The dll task is to tune and to downconvert the SDR real samples, generating a IQ complex downsampled stream that is processed by the HDSDR application.

The software source repository is https://github.com/ik1xpv/BBRF103/tree/master/ExtIO_sddc .

Here a video of BBRF103 prototype with a 1 meter wire antenna receiving local FM band in Turin. The PC used is a I5-3350P CPU @3.10GHz desktop

https://www.youtube.com/embed/V4i-ekfWK-k

I prepared a portable setup for summer holidays using a Laptop i7-7500U CPU @2.70 GHZ 2.90 GHz

https://www.youtube.com/embed/jcFJuISBV9U

To be continued.

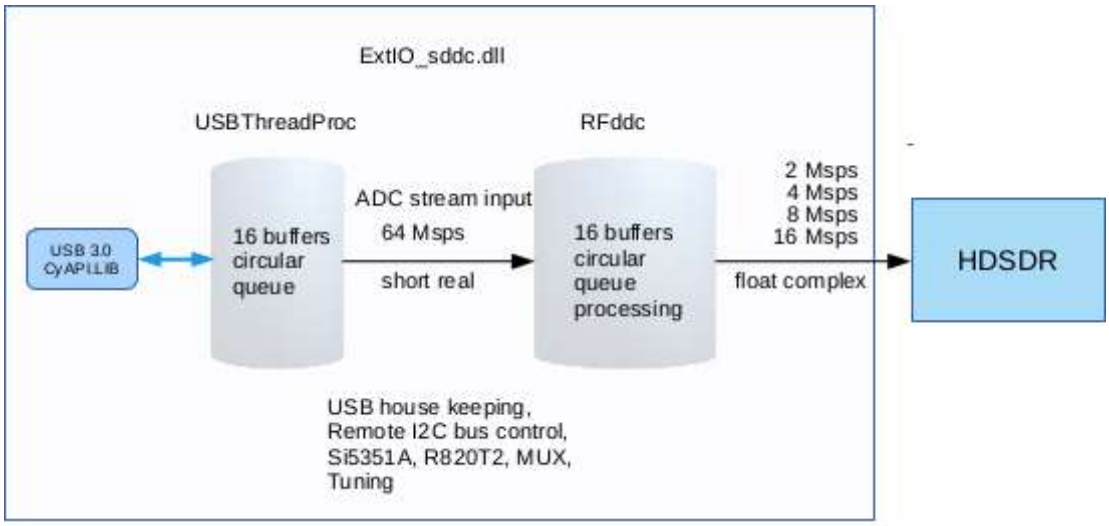
Some variations of ExtIO_sddc.dll architecture

31 August, 2017

Twitter Facebook Google+ Linkedin

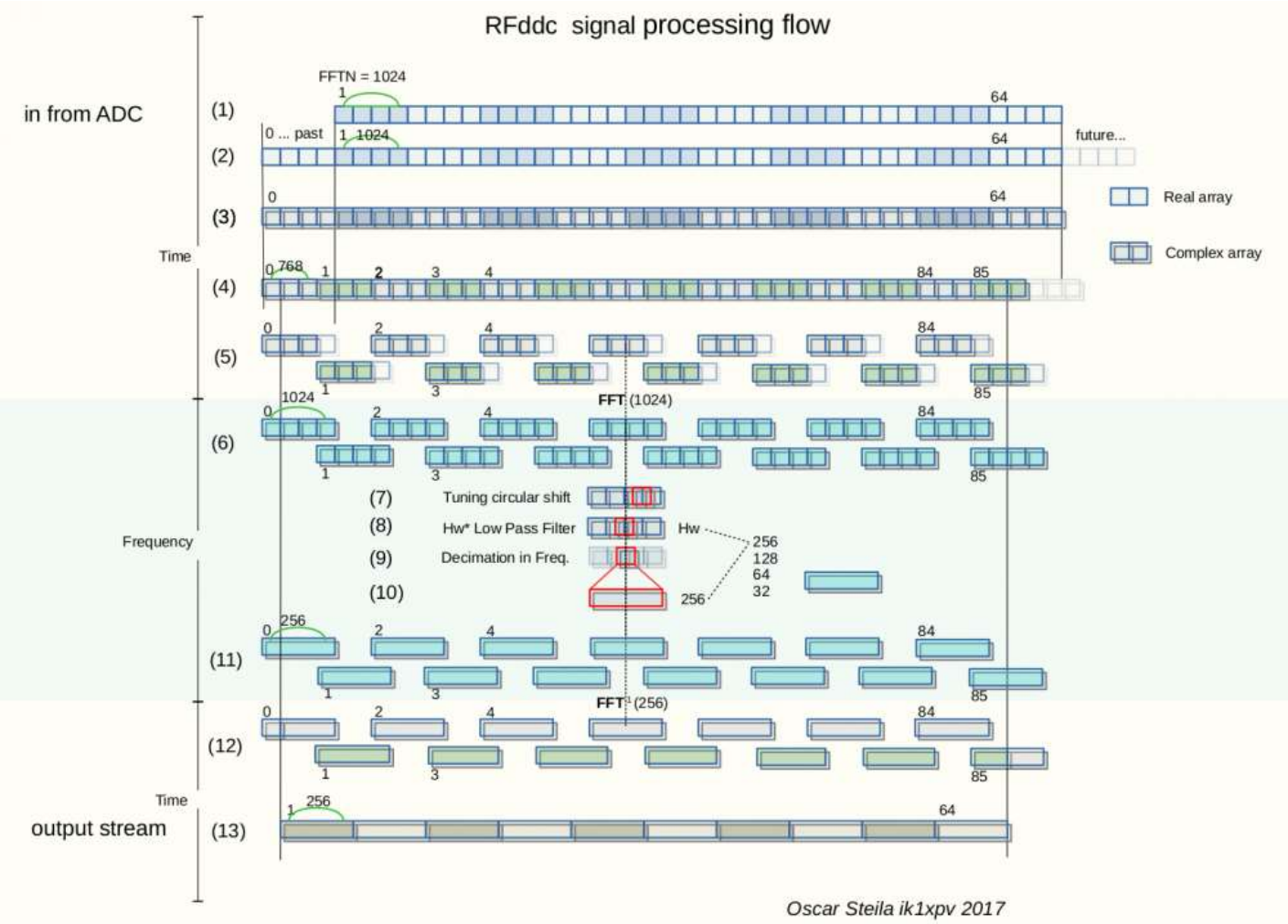
During Summer holidays I experimented about decimation scheme of BreadBoard RF103.

The ExtIO_sddc.dll processes the ADC real signal stream. The sampling rate is 64 MHz.
The output is a decimated I&Q complex signal stream at 16 MHz, 8MHz, 4MHz or 2MHz.
Filtering and tuning are integrated.



The dll uses CyAPI library to connect BBRF103 hardware via USB3.0.
A USBthreadProc thread uses 16 buffers queue to receive data chunk of 65536 samples (short).
One buffer's time duration is about 1 ms at 64 MHz.
I configured USBthreadProc to run at high priority to be responsive to hardware timing.
The USBthreadProc output buffer is processed by the class RFddc that has an own buffer array of 16 elements.
The time of a circular turn of 16 buffers queue allows the digital down conversion algorithm of each chunk to complete on a separate thread.
The previously computed output vector is returned while the signal processing of the buffer is started. A priority for these threads below normal seems good enough.
Frequency domain signal processing is used. An overlap and add FFT scheme processes the buffer 65536 sample frame and overlap and save scheme glues the buffers together.

The following diagram represents the processing of one buffer frame and it is implemented in every thread of the RFddc pool of 16.



notes:

1) It is the input sample array of short. It is a real signal. The frame buffer is a sequence of 65536 samples (it can be seen as a sequence of 64 *1024 slice).

2) The last 1024 slice in the past is copied at the beginning of the array to form a frame of 65 *1024 = 66560 samples. This is the overlap and save scheme.

3) A complex array 66560 samples long is obtained from (2) adding a zero imaginary component.

4) Starting from 0 the array is divided into slices of 768 samples to implement override and add (Overlap and add method) with an overlap of 256 and a fast Fourier transform (FFT) of 1024 sample frame.

5) Every 768 slice is copied into a 1024 one adding a tail of 256 sample zero filled.
85 slice of 1024 complex samples.

6) A FFT forward is applied to the every 1024 slice. 85 slices in frequency domain are computed.

7) For every slide a circular shift of the FFT's bins is used to implement tuning to the IQ carrier frequency.
The resolution is 64000000/1024 = 62500 Hz. This coarse step is good enough for HDSDR tuning that uses its own fine adjustment. A phase adjustment is required depending on the tuning bin position.

8) A low pass filter of the signal is implemented as fast convolution (https://en.wikipedia.org/wiki/Convolution#Fast_convolution_algorithms) multiplying the FFT bins by the complex conjugate (https://en.wikipedia.org/wiki/Complex_conjugate) frequency response of the filter (Hw*). To use the fast convolution approach the length of the time filter response ht is limited to (1024 -768) +1 = 257 samples.

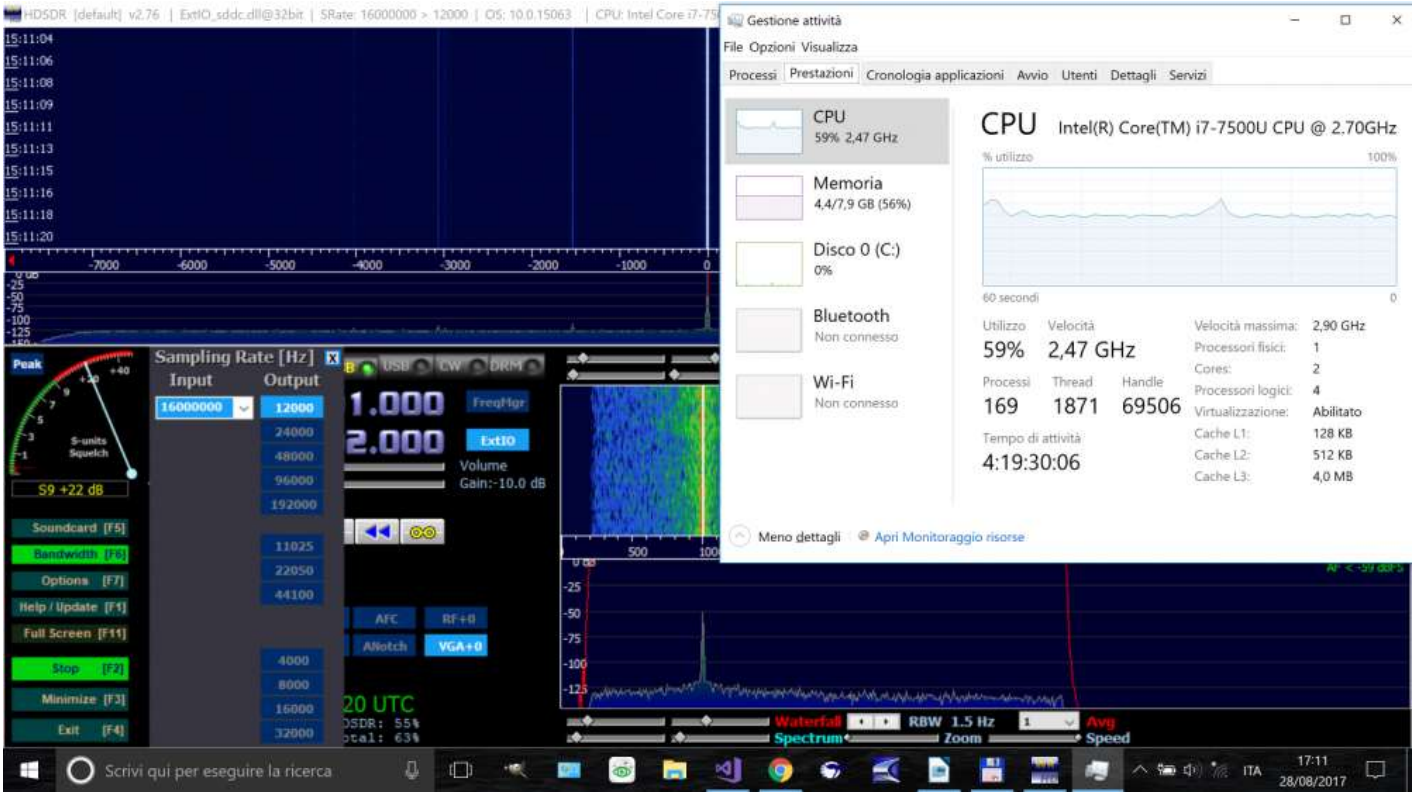
9) Decimation in frequency is used. The implemented output lengths are 256,128,64,32 that obtains output rate of 16MHz, 8MHz, 4MHz, 2MHz. It is made just copying the decimated FFT bins chunk near zero frequency.

10) The resulting output is a sequence of decimated FFT (256,…) bins. Steps (7) (8) (9) are implemented together in a copy and modify loop.

11) The 85 slices of decimated FFT.

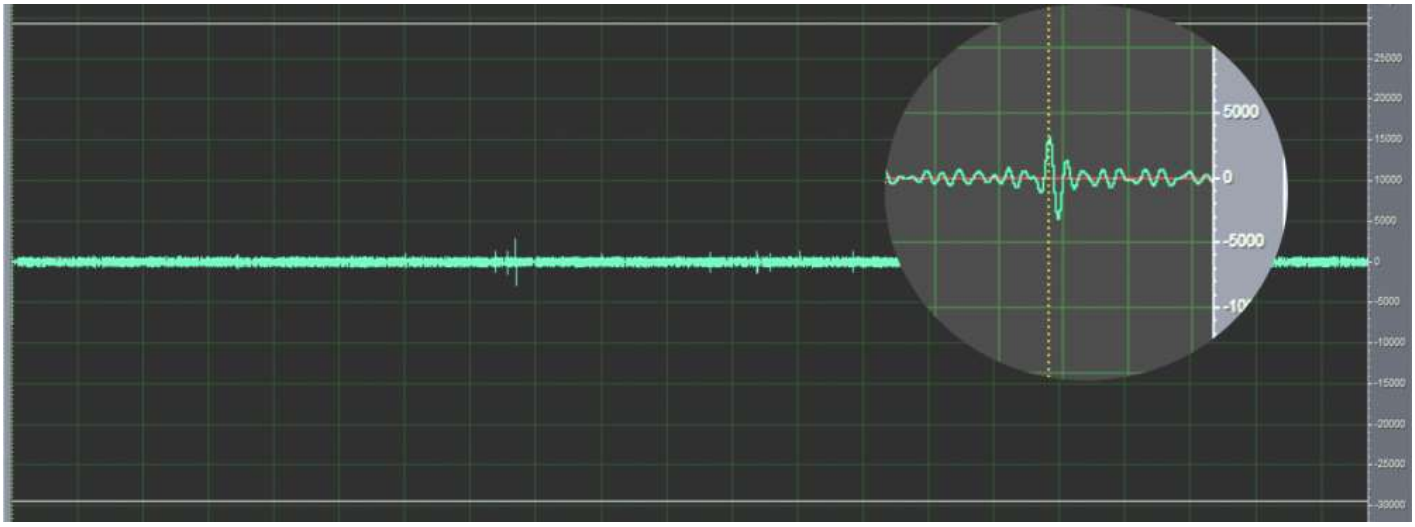
12) The FFT inverse is computed.

13) The time output is computed with overlap and add of the 85 slices. The first 256 samples and the latest 512 are dropped using overlap and save (Overlap save method) of the 64 * decimated FFTN samples frames.
The function has an array of 65536 samples input and returns an array of 16.384 samples at 16MHz, 8.192 at 8MHz, 4.096 at 4MHz, 2.048 at 2MHz.
A separate thread processes the signal from each one buffer.



CPU use of HDSDR and ExtIO_sddc, V0.95 ADC 64MHz, IQ 16Msps ; 60 s plot

I made some debug measuring the time jitter of USBthreadProc 16 buffer cycle.
The theoretical time is 16 * 1.024 ms = 16.384 ms.
Here after a plot of the measured duration time - 16.384 ms.
The plot shows that the peak jitter is within +/- 3mS.



USBthreadProc 16 buffer circle timing jitter running at 64 MHz in 16MHz sampling output, 60 s plot.

I named this release version 0.95 and I save it in a separate GitHub repository at:

https://github.com/ik1xpv/ExtIO_sddc

I switched to the integrated Visual studio Git and it was simpler to me to keep it separate from other BBRF103 project components.

To be continued.

Troubleshooting BBR103

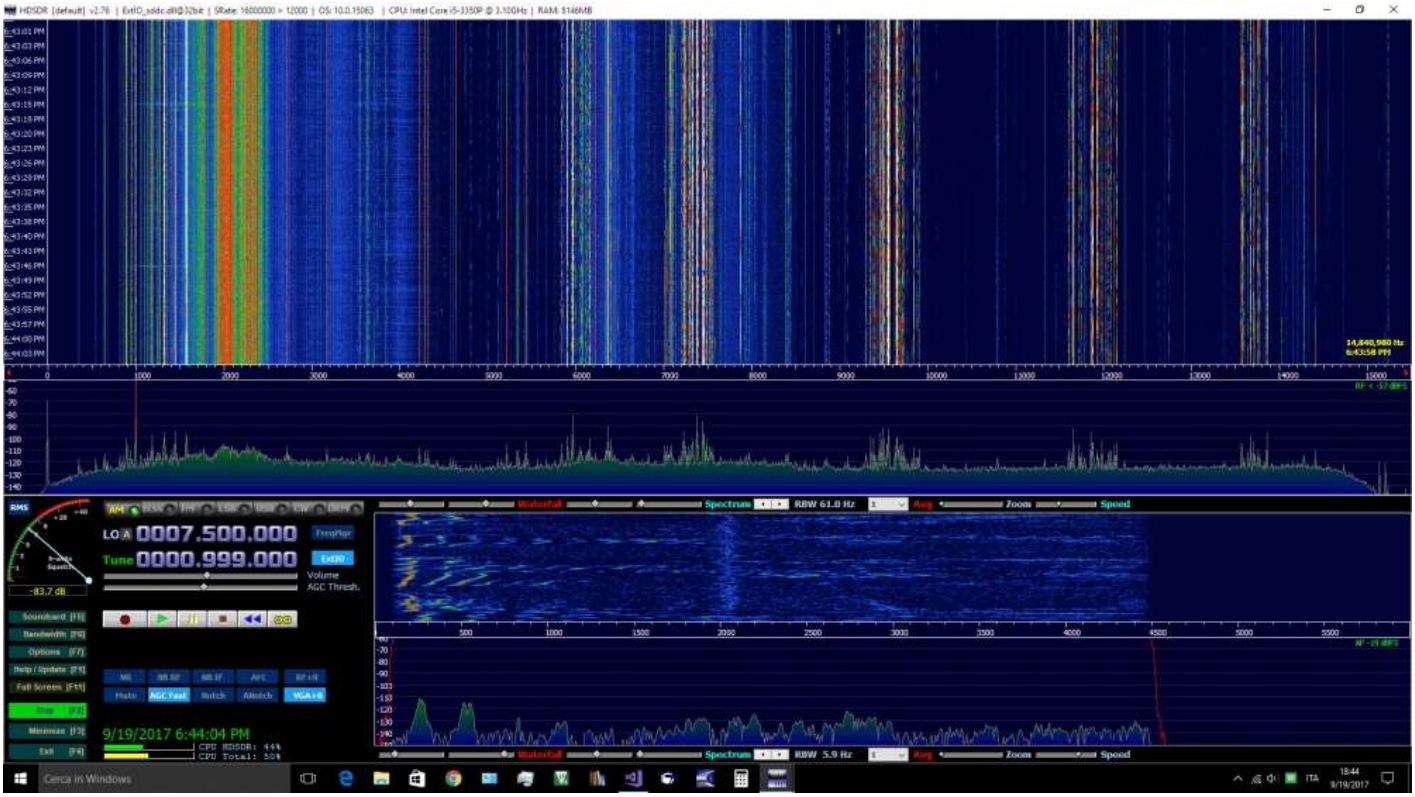
22 September, 2017

Twitter Facebook Google+ Linkedin

"*Troubleshooting or dépanneuring is a form of problem solving, often applied to repair failed products or processes on a machine or a system. It is a logical, systematic search for the source of a problem in order to solve it, and make the product or process operational again. Troubleshooting is needed to identify the symptoms. Determining the most likely cause is a process of elimination—eliminating potential causes of a problem. Finally, troubleshooting requires confirmation that the solution restores the product or process to its working state.*"
(from:https://en.wikipedia.org/wiki/Wikipedia:Troubleshooting).

BBR103 prototype is alive, nevertheless some bugs limit the performance. Hereafter the syntoms and some investigation to hopefully solve them.

I focused on the HF operation using HDSDR with IF bandwidth of 16 MHz as shown in the following picture.



The BBR103 in the picture is connected to a 5 meter wire on my balcony in the city. The ADC performance seems good as espected.

Nevertheless I notice the following problems during its use:

~~001~~) (SOLVED) The signals had a periodic distortion (-50 dB down) at 10ms time distance. I noticed it using a 10MHz reference generator.

The bug was periodic with the period of the FRAMEN lenght buffer. Digging in the rfddc code I got the bug. It was mirroring the wrong past frame.

Correction of 63->64

Browse files

cleanup_A

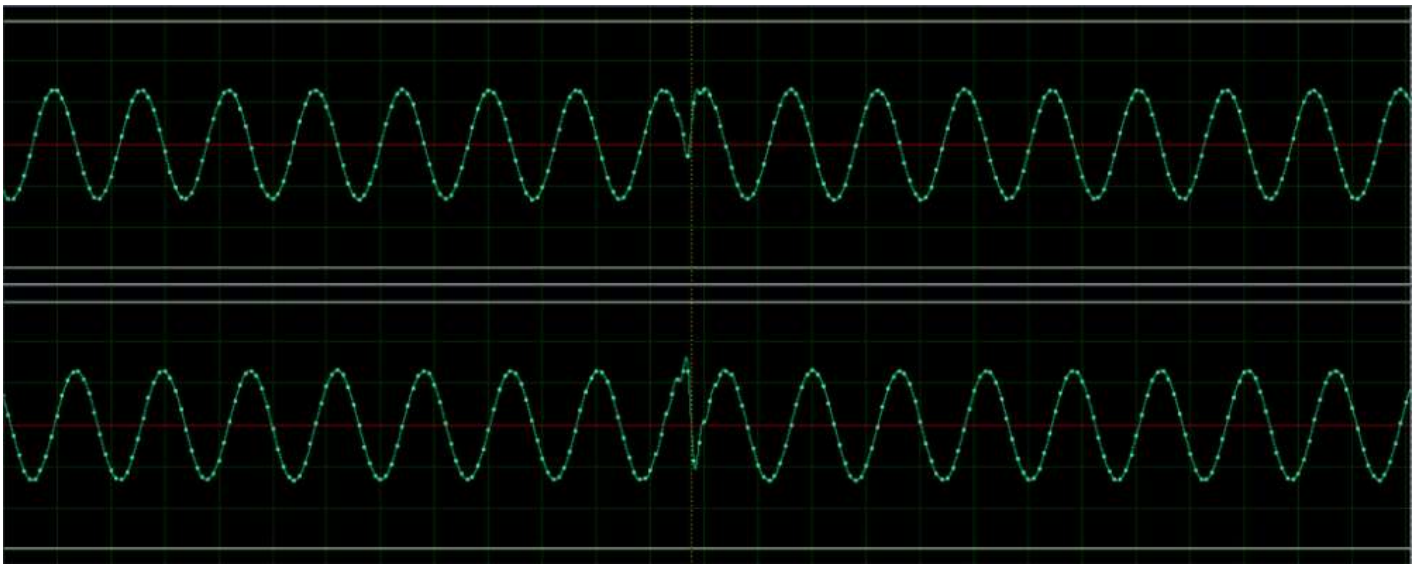
ik1xpv committed 3 days ago 1 parent 3dd871f commit cc0ba3928ce833e9effab92bf075e90554e7e785

Showing 1 changed file with 1 addition and 1 deletion. Unified Split

2 ExtIO_sddc/ExtIO_sddc.cpp View

```
@@ -426,7 +426,7 @@ DWORD WINAPI USBThreadProc(__in  LPVOID lpParameter)
426 426                                 Successes++;
427 427 #endif
428 428
429     -                               memcpy( mybuffer[(ii+1) % NVIA], &mybuffer[ii][1024 * 2 * 63], 1024 * 2);
    429 +                               memcpy( mybuffer[(ii+1) % NVIA], &mybuffer[ii][1024 * 2 * 64], 1024 * 2);
```

002) There are some distortion on the signal that are at random time like the following I got in a IQ waveform recorded with HDSDR.



The distortion looks like a 90° phase shift. I imagine that the cause is in the dll. The randomness of the behaviour keeps the search difficult.

003) The USB communication fails at random time (up to some tens minutes). I got the following debug message:

```
...
```

```
Xfer request rejected. NTSTATUS = c000000e
```

```
AbortXferLoop()
```

```
...
```

The USB device disconnects. It requires reset to start again.

My knowledge about USB is very low and I have to learn everything. I found some hits:

https://community.cypress.com/thread/27549

I run the attached https://community.cypress.com/servlet/JiveServlet/download/122270-26243/FX3GPIFnoise.zip

I used the following procedure:

1) Load the SDRx3B.img in BBRF103.

2) Run HDSDR and then close it (to initialize the BBRF103 hardware).

3) Run the FX3USBread console application from FX3GPIFnoise.zip.

I use a script file. In this example it has a 3 minutes run.

```
d:\DEV\FX3GPIFnoise>time /T
03:15 PM
```

```
d:\DEV\FX3GPIFnoise>FX3USBread
FX3USBread version 1.0
```

```
Press ESC for stop
Count:22982361088 Speed:120.571MB/s Max:122.792MB/s
DeviceIoControl failed (GetOverlappedResult error code=995)
Operazione di I/O terminata a causa dell'uscita dal thread oppure della richiesta di
un'applicazione.
```

```
d:\DEV\FX3GPIFnoise>time /T
03:19 PM
```

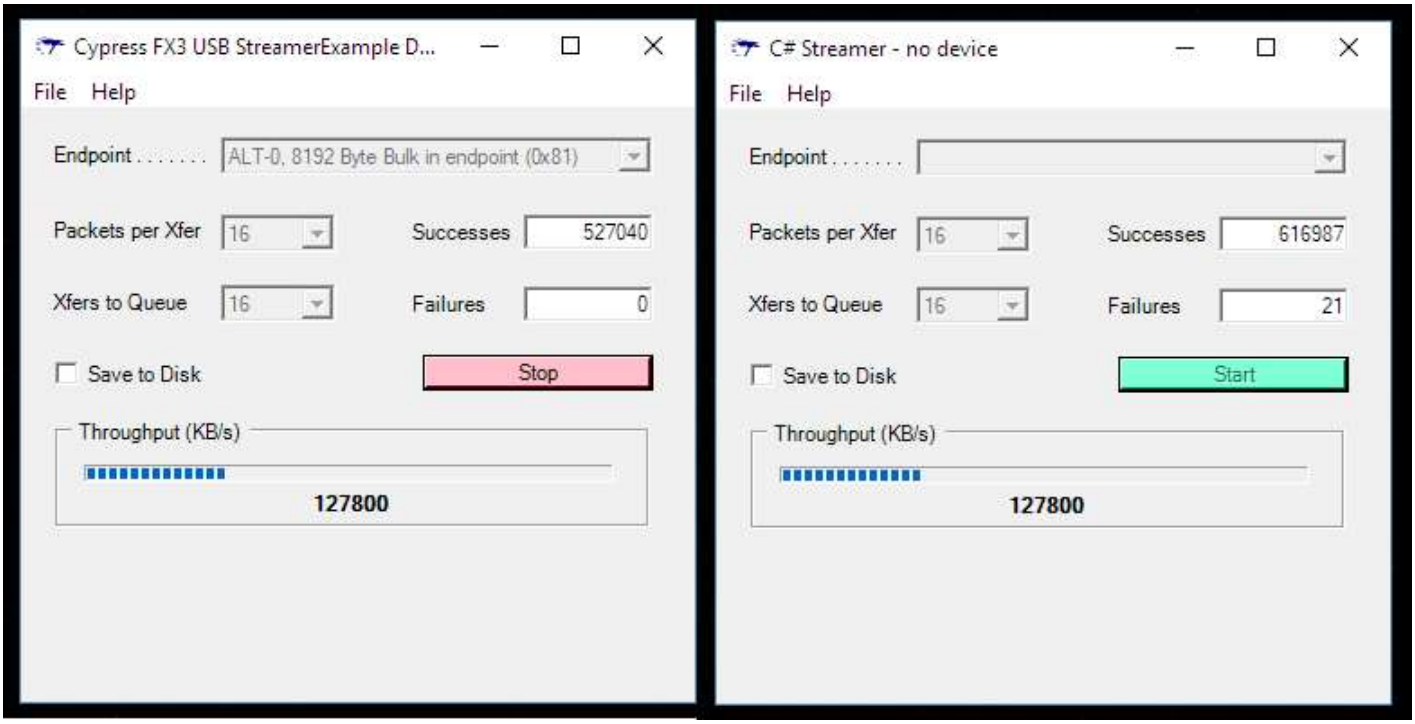
```
d:\DEV\FX3GPIFnoise>PAUSE
Premere un tasto per continuare . . .
```

It crashes at random time as it happens with HDSDR+ ExtIO_sddc.

I made the same test using Cypress stream application.

Power on of BBR103 with HDSDR + ExtIO_sddc.dll to program ADC clocks.

Run stream.exe I got the same problem after some time (5 minutes in the example)



The bug seems to be in the FX3 device firmware because it happens with 3 different application.

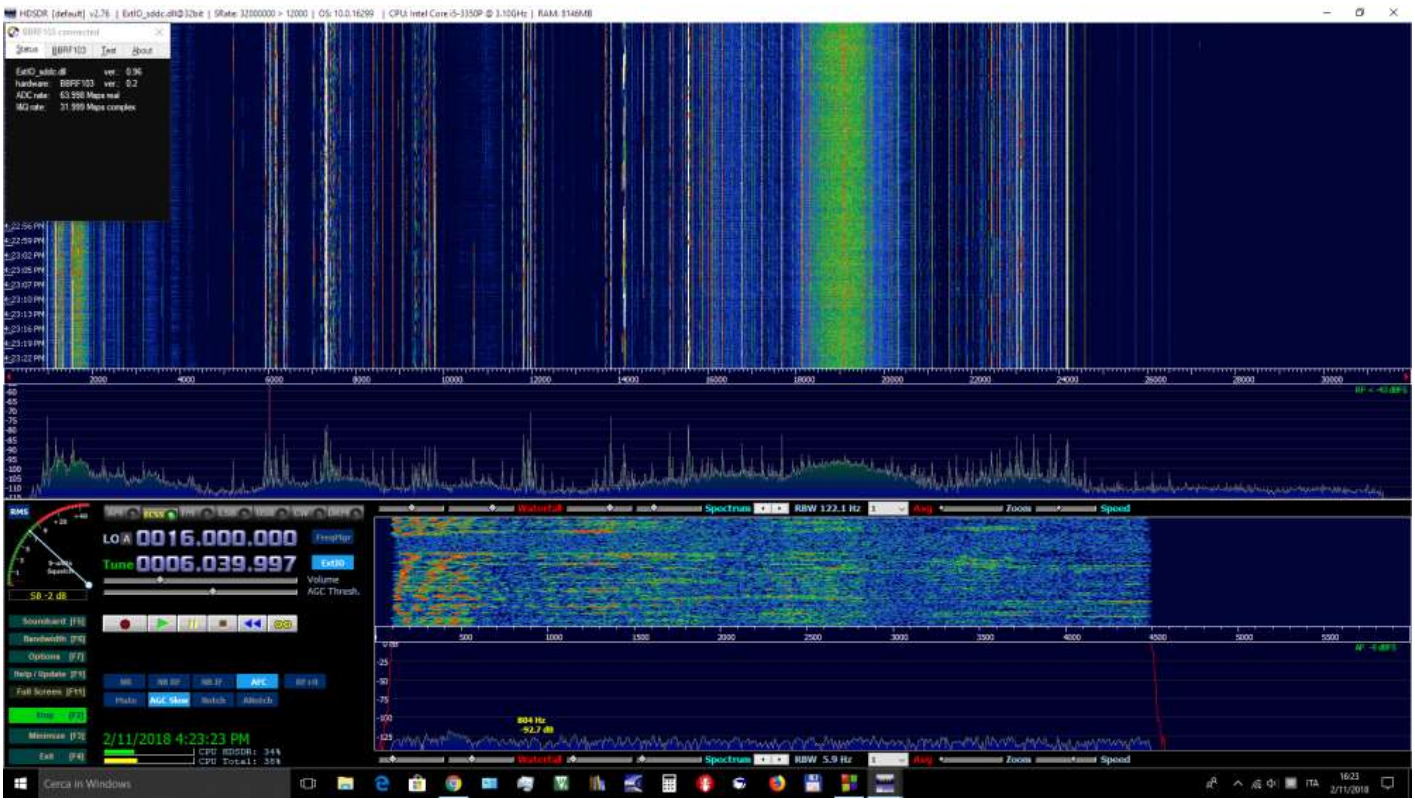
..continue..

Thanks in advance for advice at ik1xpv~gmail.com.

BBRF103 and ExtIO.dll ver. 0.96 get 32 MHz span

01 March, 2018

Twitter Facebook Google+ Linkedin



End of September 2017 I was googling “SDR and FX3” when I found a nice SDR project named Booya SDR (http://booyasdr.sourceforge.net/BooyaSDRDoc.pdf , http://booyasdr.sourceforge.net/).

It uses the same USB3 Cypress Explorer Kit Board while the ADC is a LTC2206.
Reginald Eisenblatt, gave a presentation on the BooyaSDR (January 23, 2017, at Linaspace see http://www.amrad.org/).
Some video at https://www.youtube.com/watch?v=uF6y0ETTJFA , notice the Waterfall with multiple rows!

The BooyaSDR application is very clever. It uses gcc compiler with pthreads and fftw library. The FX3 firmware is loaded at run time using the Cypress download protocol.

I decided to use the same software environment and compiler to test a version of ExtIO_sddc.dll for BBRF103 with pthreads lib.

To install CodeBlocks 12.11 IDE , https://sourceforge.net/projects/codeblocks/files/Binaries/12.11/Windows/ download codeblocks-12.11mingw-setup.exe -> Default installer WITH compiler (MinGW).

ExtIO_sddc.dll ver 0.96 project links to the following libraries:
pthreads :
https://sourceforge.net/projects/pthreads4w/files/pthreads-w32-2-9-1-release.zip/download
copy Pre-built.2 directory contens into /lib/pthreads/.
Fftw:http://www.fftw.org/
ftp://ftp.fftw.org/pub/fftw/fftw-3.3.5-dll32.zip

CyAPI_gcc:
a gcc compiled version of CyAPI.cpp.
library source can be downloaded at http://www.cypress.com/file/289981/download
(see License) .

The directories structure I used is:

ExtIO_sddc \	readme.txt ,
ExtIO_sddc \BBRF103_SE	FX3 firmware,
ExtIO_sddc \source\	sources, ExtIO_sddc.cbp,
ExtIO_sddc \Lib\fftw	fftw library,
ExtIO_sddc \Lib\pthreads	pthreads library ,
ExtIO_sddc \Lib\CyAPI_gcc	CyAPI gcc library ,
ExtIO_sddc \bin\debug	debug ExtIO_sddc.dll, HDSDR ,
ExtIO_sddc \bin\release	release ExtIO_sddc.dll, HDSDR.

The bin\release and \bin\debug directories contain:

HDSDR.exe	
ExtIO_sddc.dll	release or debug
BBRF103_SE.img	BBRF103 firmware image
libfftw3f-3.dll	
pthreadGC2.dll	

Sources repository :

https://github.com/ik1xpv/ExtIO_sddc-Ver0.96

An archive file of project with compiled binaries can be download at

http://www.steila.com/test/ExtIO_sddc_v096.zip

MD5 6efcd88bdc14107389cae5d6f7efe3dc

SHA-1 88ef3f3ba6276da694f6e92ebb71d987046de100

Hardware:

The BBR103 has been updated to version 0.2 with the following patch.
- RAND patch: a wire has been added to control the RAND pin of ADC using GPIO20 of FX3.
This option allows the control and test of RAND feature of ADC (see pg 14, 24
of http://cds.linear.com/docs/en/datasheet/2217f.pdf).
The wire connects RAND (U3-pin 63, R7,R9) to GPIO20 (BGA K7 = PIN25 J6 FX3 SS kit).

Firmware:

The Firmware source can be found into the archive file of project under \Firmware directory

Status:

The 0.96 version is still a preliminary release with some bugs. It operates in HF mode only.

Problems remain in use of R820T2 tuner, and a post on this argument will follow.

The digital signal processing frontend uses a the Halfcomplex-format DFT (http://www.fftw.org/fftw3_doc/The-Halfcomplex_002dformat-DFT.html).

The FFT output is sent to HDSDR with a selectable rate of 32 Msps or a decimated one at 16, 8, 4, 2 Msps.

Filtering and tuning is made using overlap and add with frame of 1024 as 768 +256 samples. The filter time responses are 257 sample long.

When 32Msps is used the local oscillator of HDSDR is fixed to 16 MHz at centre of spectrum and the fine tuning of HDSDR allows reception from 500 kHz to 31500 kHz, while with lower sample rates the local oscillator can be tuned with 125 kHz step while the fine tuning is made by HDSDR.

At this development stage ExtIO_sddc.dll has a GUI dialog with 4 tabs :

- Status - reports ADC rate and I&Q rate.
- BBRF103 - buttons :

  LW-MW this is used to modify the FFT output filtering to receive the low frequency band.
  HF - standard HF setup .
  VHF - enables R820T2 (it is disabled, to enable undefine _NO_TUNER_ in config.h and recompile)
  DITH - enables the ADC dither.
  RAND - enables the ADC randomize.

  TRACE - enabled in debug mode to trace some signals to log files.

- Test

  RF ADC stream: it requires BB103, ADC input ,default,
  RF virtual tone: it requires BB103, virtual tone,
  RF virtual sweep: it requires BB103, virtual sweep,
  IF virtual tone: NO hardware required, virtual tone ,
  IF virtual sweep: NO hardware required, virtual sweep,

- About

Hereafter a video recorded with a random wire antenna 5mt long on the balcony (in the city).

R820T2 update - BBRF103_2 PCB

10 April, 2018

Twitter Facebook Google+ Linkedin

A bug crashed the USB3.0 stream at random time while the R820T2 tuner was active. (Troubleshooting BBR103)

It was caused by a spurious coupling via the 3V3 power supply.

I thought of a problem in the firmware of FX3 while the cause was hardware.
I decoupled the R820T2 3V3 power supply using a separate LDO from the 5V USB bus to solve the problem in the prototype.

The R820T can be used to receive frequency band in the range 30MHz -1800MHz.
The tuner uses a clock generated by Si5351A at 32.000MHz, the REGDIV bit of reg 4 is set to 1 to divide it by two internally to the nominal 16MHz.
The IF output is selected at 5MHz (I used up to 7.5MHz) and it's sampled by the ADC at 64Msps and then decimated down to 8Msps or less. The R820T data sheet states that the standard IF filters are implemented for 6/7/8 MHz channel bandwidths.

The LNA, the mixer and the VGA gains can be set manually although their precise values are absent from the datasheet.
They can also be set automatically via automatic gain control (AGC) in order to optimize the signal to noise ratio (SNR).

A revision of PCB BBRF103_2 has been designed: (www.steila.com/test/BBRF103_2B.pdf)

- A separate LDO voltage regulator for R820T2 has been added

- 1000 uF capacitors with a mosfet delayed switch has been added to 5VBUS and 3V3 R820T2.

- Antenna power supply with software switch added to HF and VHF input

- SMD pad dimensions have been increased a little bit to simplify manual assembly.

- Board profile modified to house SMA connectors.

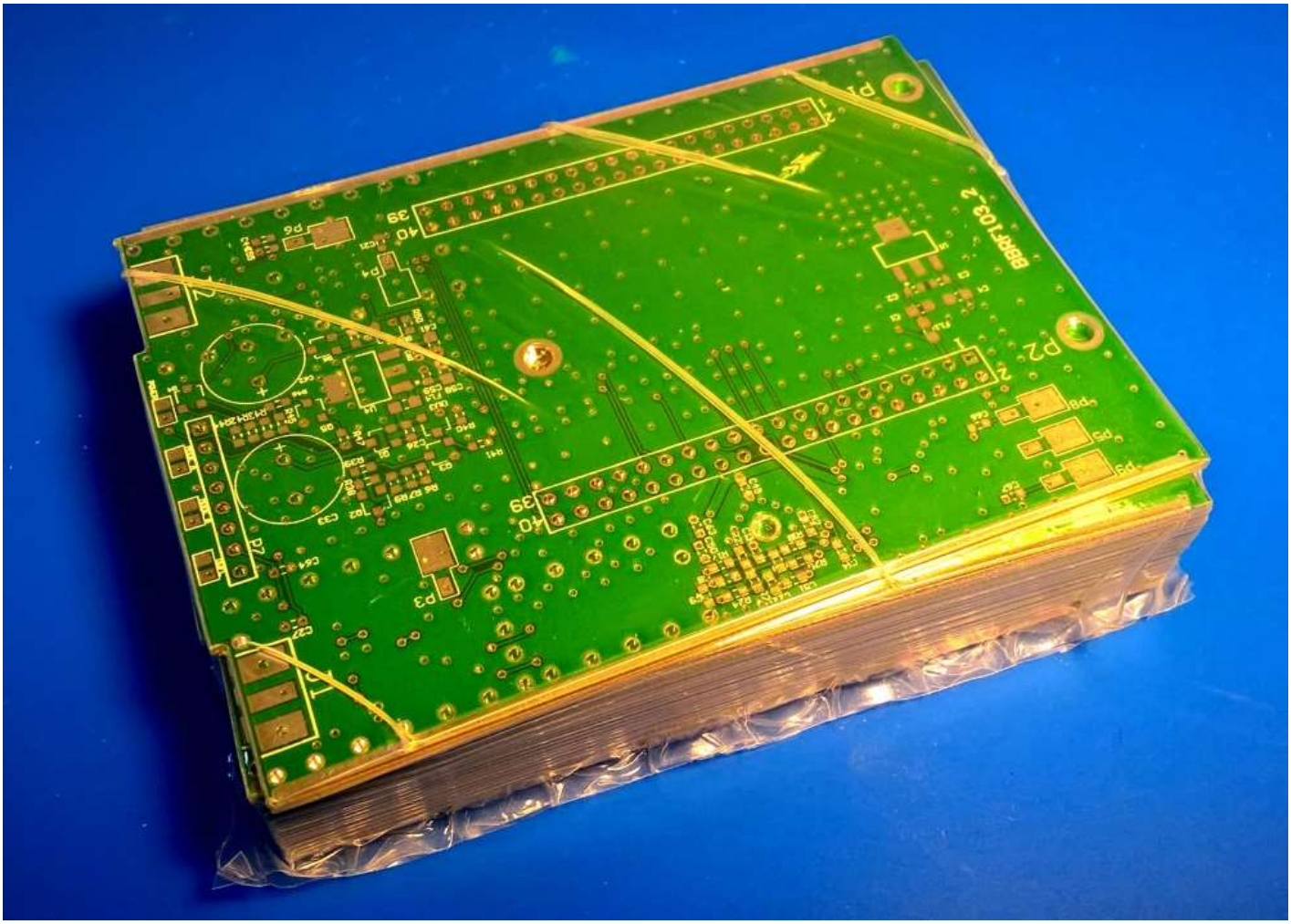
- BAV99 smd layout corrected.

- A header P7 with some GPIOs added.

- Possibility of external frequency reference input to Si5351a (P8).

- Aux clock ouput (P9).

I just received the manufactured pcb and possibly I will test it in the next month.


BBRF103_2
P2
P3
P4
P5
P6
39
40
RZ

BBRF103 - Band L reception

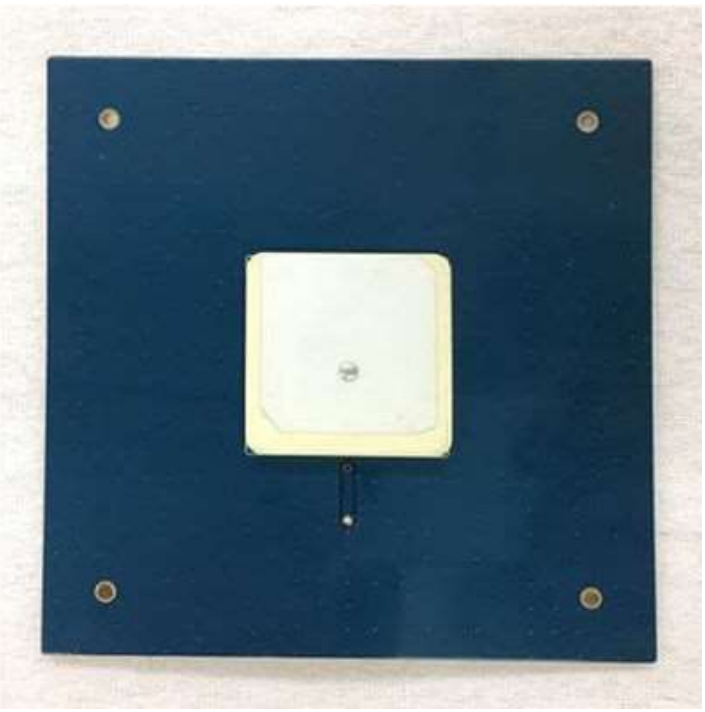
26 May, 2018

Twitter Facebook Google+ Linkedin

BBRF103_2 PCB adds a switchable LNA power supply through the antenna cable

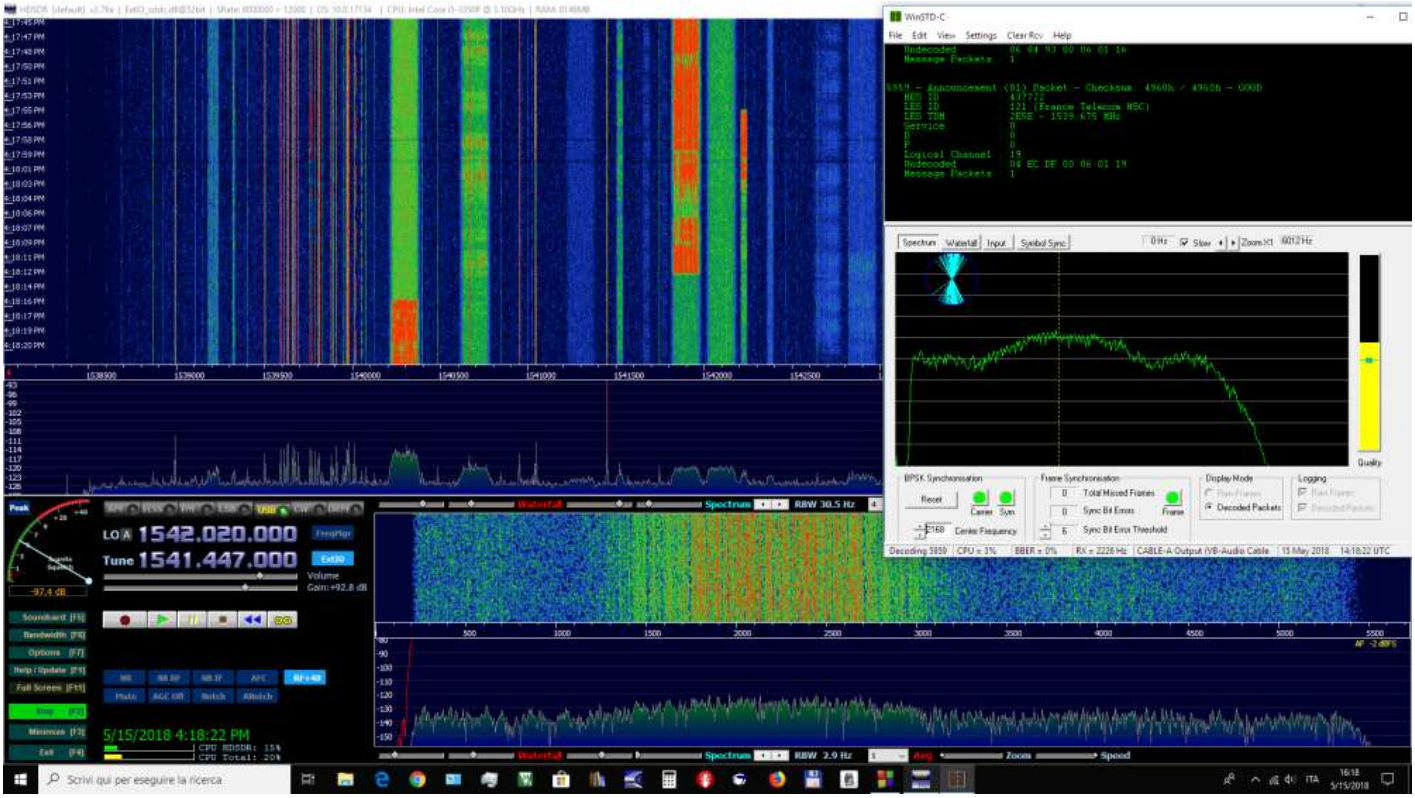
I tested it with an Outernet L-band ceramic patch antenna. This antenna requires power and can be connected to BBRF103_2 PCB. The antenna onboard filter helps to reduce problems from interfering signals and restricts reception to 1525 - 1559 MHz.

The antenna is a 12 by 12 cm square PCB. I placed it into a plastic radome (an empty IKEA FIXA series DIY kit).

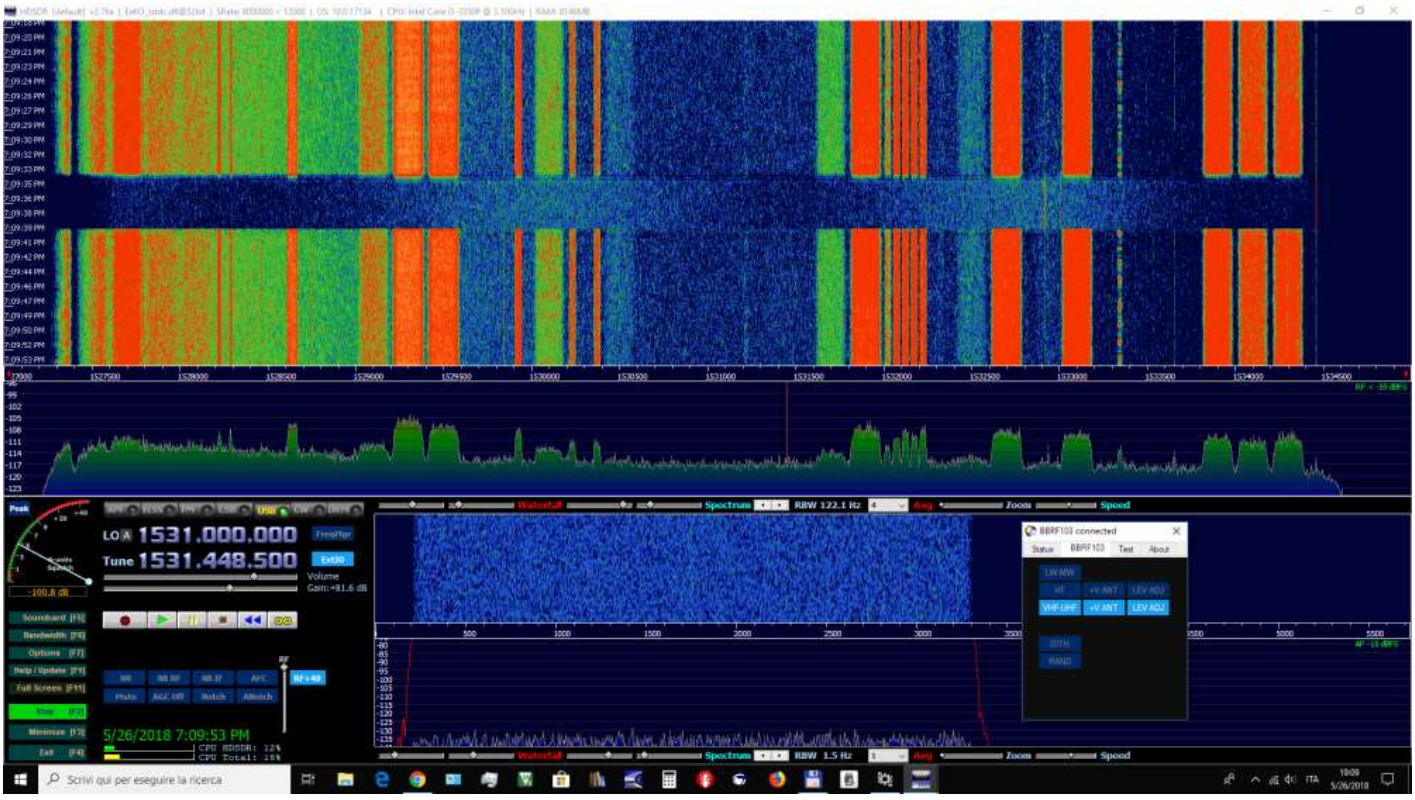


A picture of the installed radome.

Some screenshots of the band L:



Reception of Inmarsat C with WinSTD-C program.



Ten MHz down there are many other satellite signals. The gap in the spectrogram shows the BBRF103 noise when the antenna power is switched off; two spurious signals are visible on the right.

BBRF103_2 PCB notes

ADC: I mounted the BBRF103_2 prototype with a LTC2208 ADC instead of LTC2217. I wanted to check compatibility and I had not any other LTC2217 sample. The LTC2208 is a little noiser than LTC2217 by some 2 dB. The LTC2208 draws some mA more current.

Antenna power: The scheme uses 2N3906,2N3904 : Q2,Q3,Q4,Q5. The pcb footprint is wrong. Mount them upside down on the pc board.

I mounted R42,R43,R44 = 10 Ohm, it increases the output current.

R820T2: I added some bypass capacitors to the VCT line on the top layer.

Temperature: I made some measure of the temperature of R820T2 and LTC2208 with a small copper radiator.

The temperature of the ADC with a small copper radiator reaches 67 ° C while the R820T2 with the radiator reaches 45 ° C.

Preview: I will use MAX4995 50mA to 600mA Programmable Current-Limit Switch to control the LNA current in a new pcb's revision and some heat radiators will be added to ADC and RT820T2...

email: ik1xpv AT gmail DOT com

Just another BBRF103 version

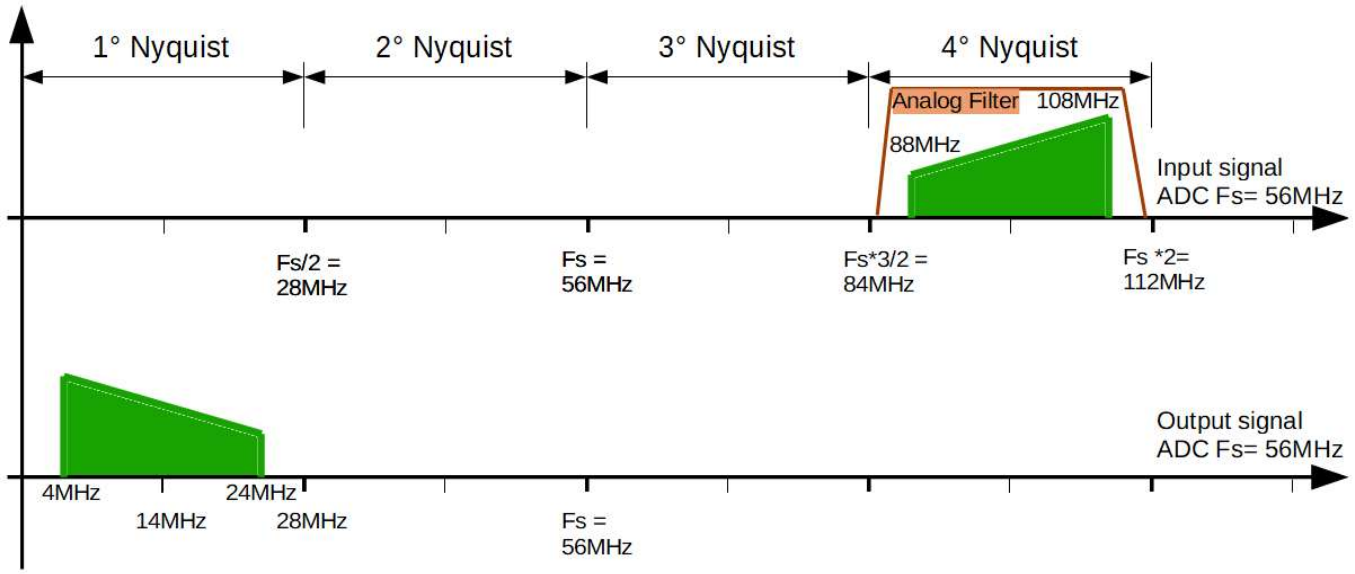
14 August, 2018

Twitter Facebook Google+ Linkedin

A new board to experiment with undersampling technique.

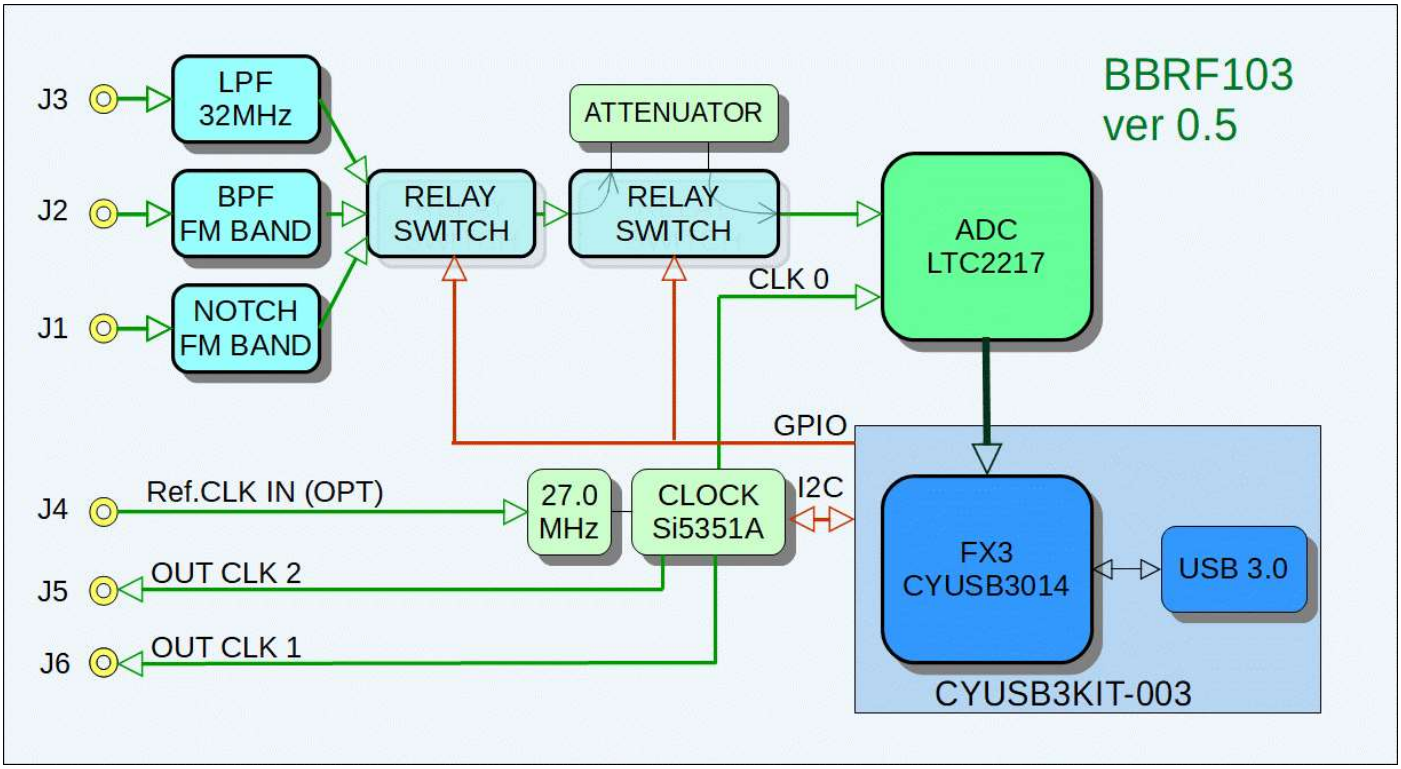
"..If we use the sampling frequency less than twice the maximum frequency component in the signal, then it is called undersampling. Undersampling is also known as band pass sampling, harmonic sampling or super-Nyquist sampling. Nyquist-Shannon Sampling theorem, which is the modified version of the Nyquist sampling theorem, says that the sampling frequency needs to be twice the signal bandwidth and not twice the maximum frequency component, in order to be able to reconstruct the original signal perfectly from the sampled version. If B is the signal bandwidth, then Fs > 2B is required where Fs is sampling frequency. The signal bandwidth can be from DC to B or from f1 to f2 where B = f2 – f1. The aliasing effect due to the undersampling technique can be used for our advantage. When a signal is sampled at a rate less than twice its maximum frequency, the aliased signal appears at Fs – Fin, where Fs is the sampling frequency and Fin in the input signal frequency. " from Why Use Oversampling when Undersampling Can do the Job? - Texas Instruments.

In an example case looking at FM band we sample the input 98 MHz with Fs = 56MHz and the aliased component will appear at 14 MHz (56*2 – 98).
As we know in advance that the signal is aliased, we can recover the actual frequency by using the N*Fs – Fin relationship. The undersampling technique allows the ADC to behave like a mixer or a down converter in the receive chain. For a band-limited signal of 98 MHz with a 20-MHz signal bandwidth, the sampling rate (Fs) of 56 Msps, the aliased component referred to 2*Fs will appear between 4MHz to 24 MHz (20 ±10 MHz.
An analog band pass filter is required at ADC input to avoid interference from other Nyquist band.



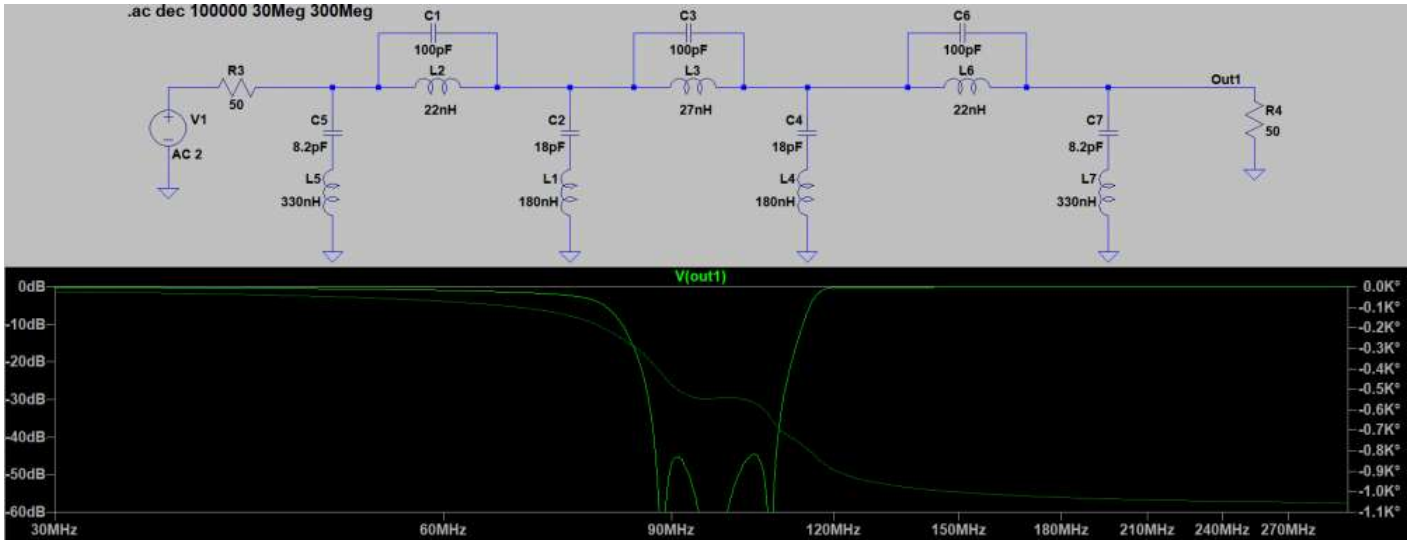
Undersampling Case of 98MHz Signal with 20MHz Bandwidth.

I designed a new breadboard with some modification. The PCB is named **BBRF103 ver 0.5**.



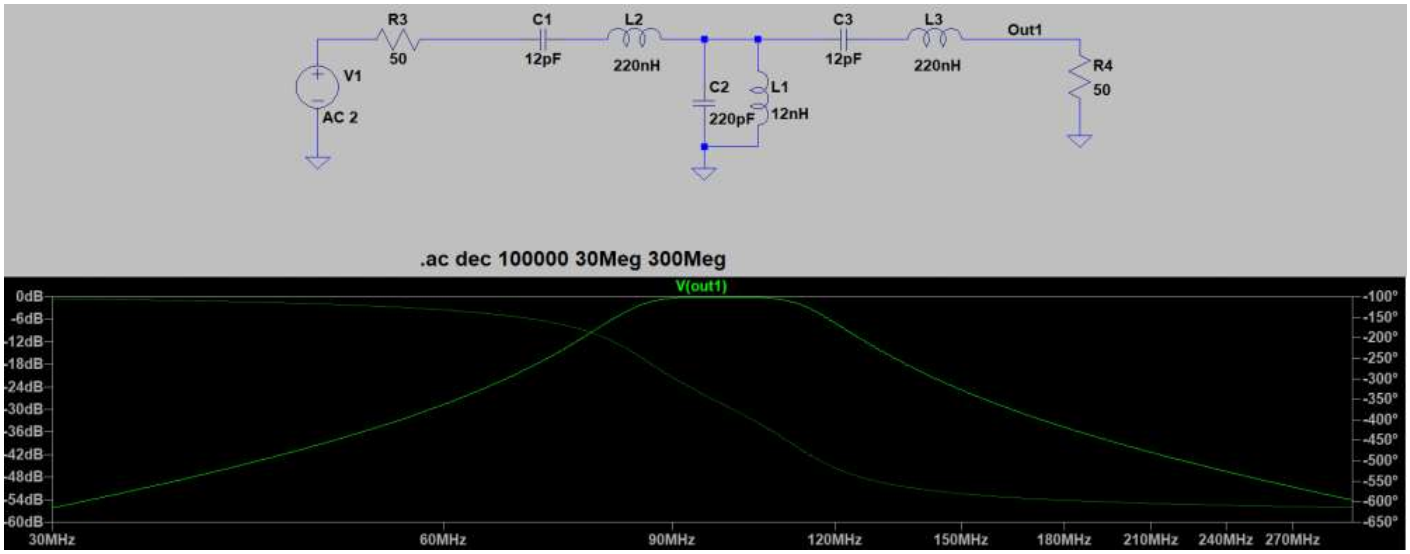
BBRF103 ver 0.5 - block diagram.

The J1 input uses a band stop filter for the FM Band 88-108 MHz.
This input is planned for experimental use within 120-500 MHz frequency range. Some specific band pass filter and LNA will be externally added for 50MHz, 144MHz or 432MHz band.



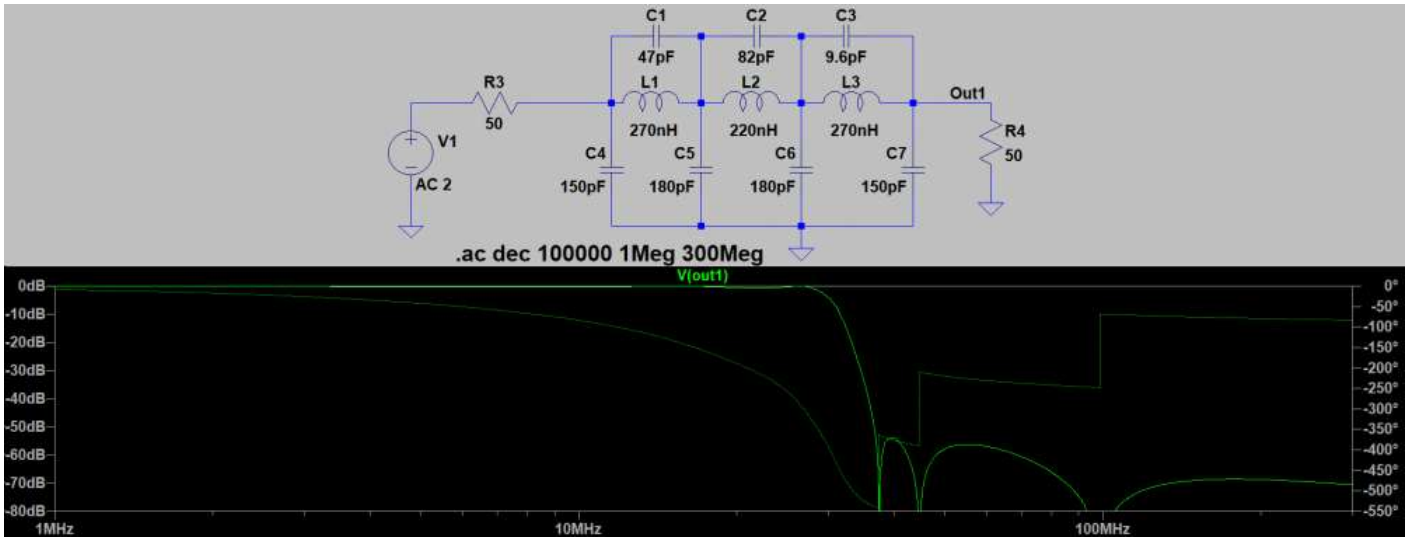
The FM Band Stop Filter - LTspice simulation.

The **J2** input has a band pass filter for the FM band to analyze the full 88-108 MHz band spectrum at once. This filter is quite simpler as the FM signals are very strong versus possible interference.



FM band filter response.

Finally **J3** is the HF input. The filter components values are changed from first version of BBRF103.



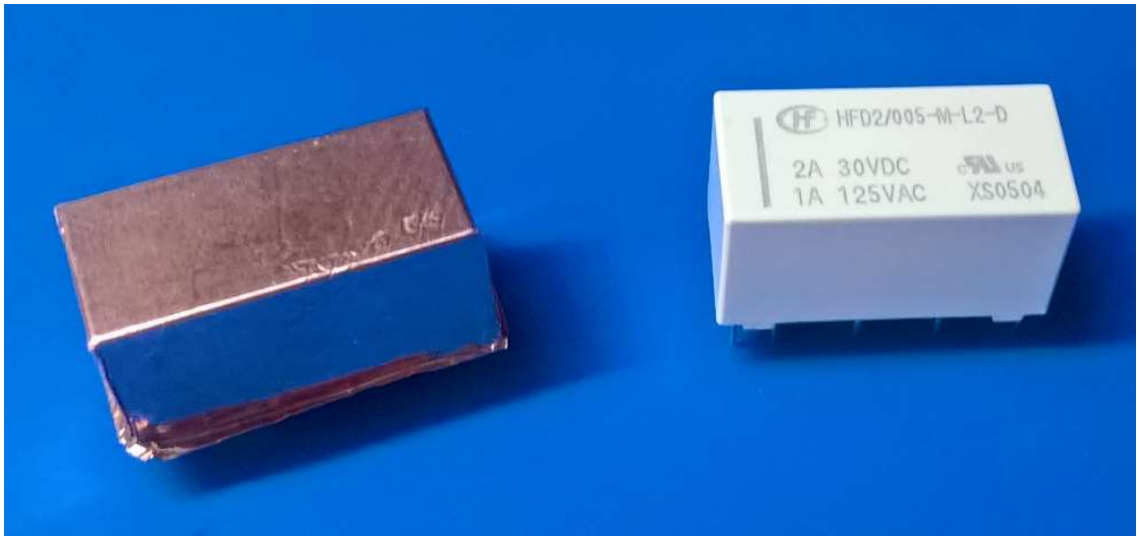
HF low pass filter.

The connector **J4** is an optional input for an external reference signal. A capacitor must be mounted to enable this signal.

J5 and **J6** are two programmable clock output from the Si5351 generator. May be used to synchronize external tuner oscillator.

RF switch type

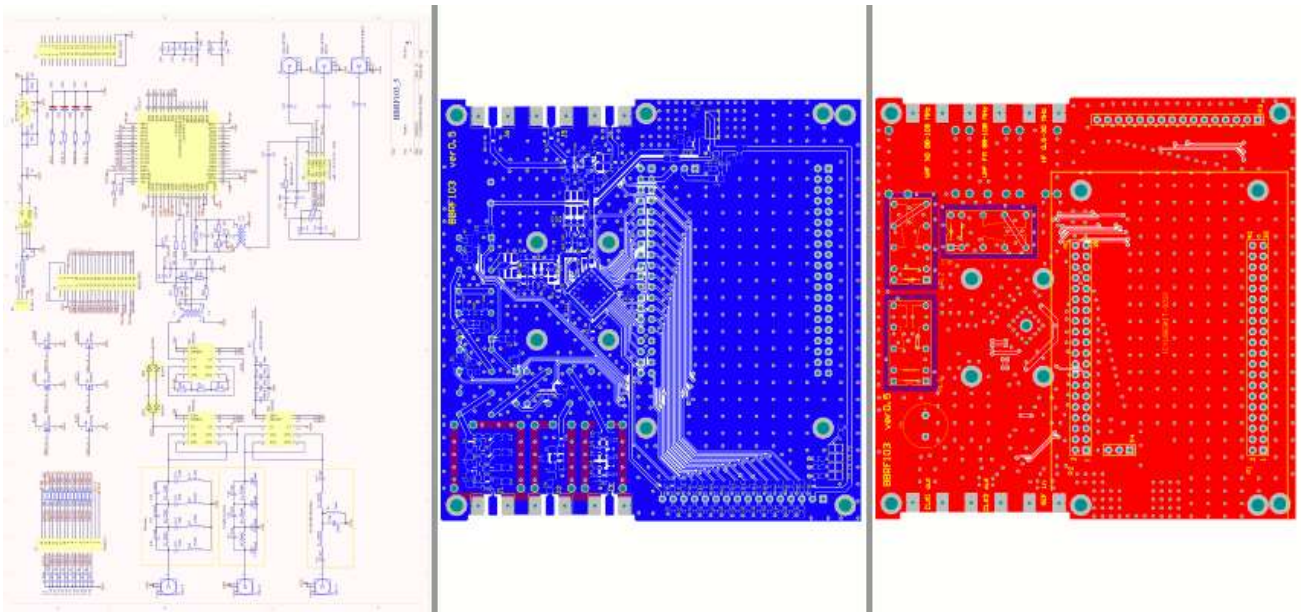
I like to test a bi-stable subminiature DIP relay type HFD2/005-M-L2-D to switch RF instead of active switches.
The relay is a 5Volt dual coil latched one. I shielded it using adhesive copper tape that will be soldered to the PCB ground plane.



Relays used. The left one with a copper tape shield added.

The board scheme and PCB layout is at link http://www.steila.com/test/BBRF103_5.pdf

24/04/2019 - here http://www.steila.com/test/BBRF103_5B.pdf annotated scheme circuit to keep relays off during power on.



I preview to receive the PCB within September and then to start testing.

24/04/2019 - My prototype shows that the PCB layout around the PCB is worse and noisier than the previous PCBs. A makeover is needed.

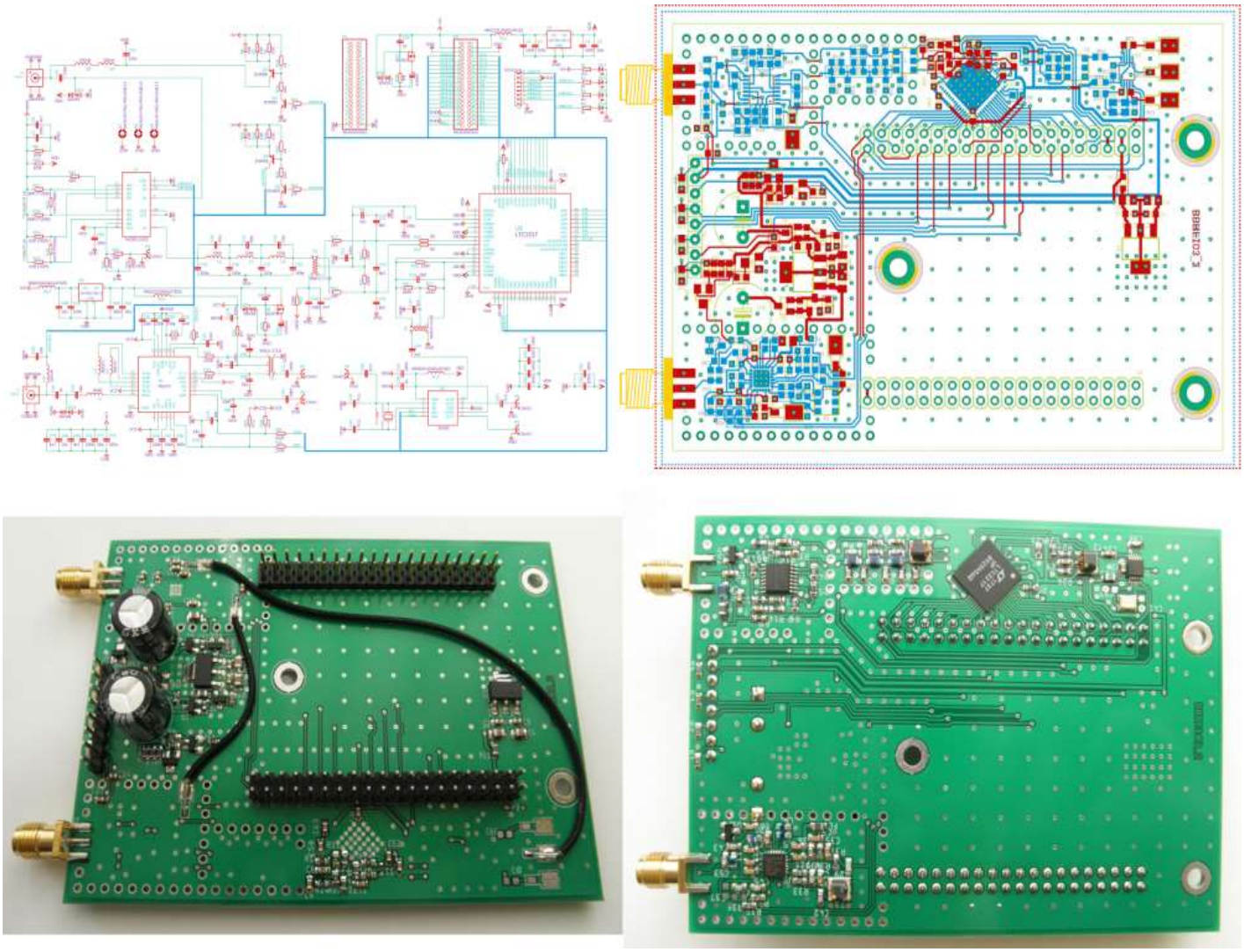
email: ik1xpv AT gmail DOT com

BBRF103-2 RC3 is here!

01 September, 2018

Twitter Facebook Google+ Linkedin

Radek Haša is a shortwave and airband listener living in Czech republic.
He decided to redesign the PCB layout and assembled a prototype of the BBRF103-2.
Thanks for allowing his project to be published.



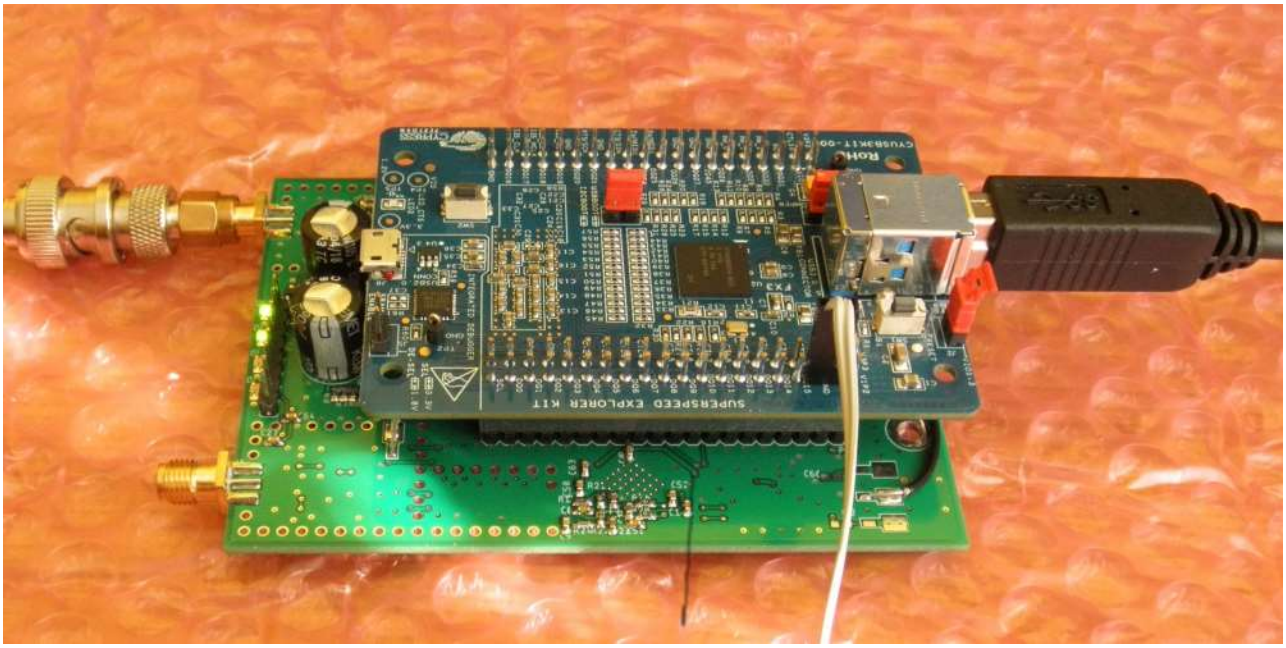
While designing the PCB in Eagle 8.x format, he noticed and fixed some bugs in my layout.

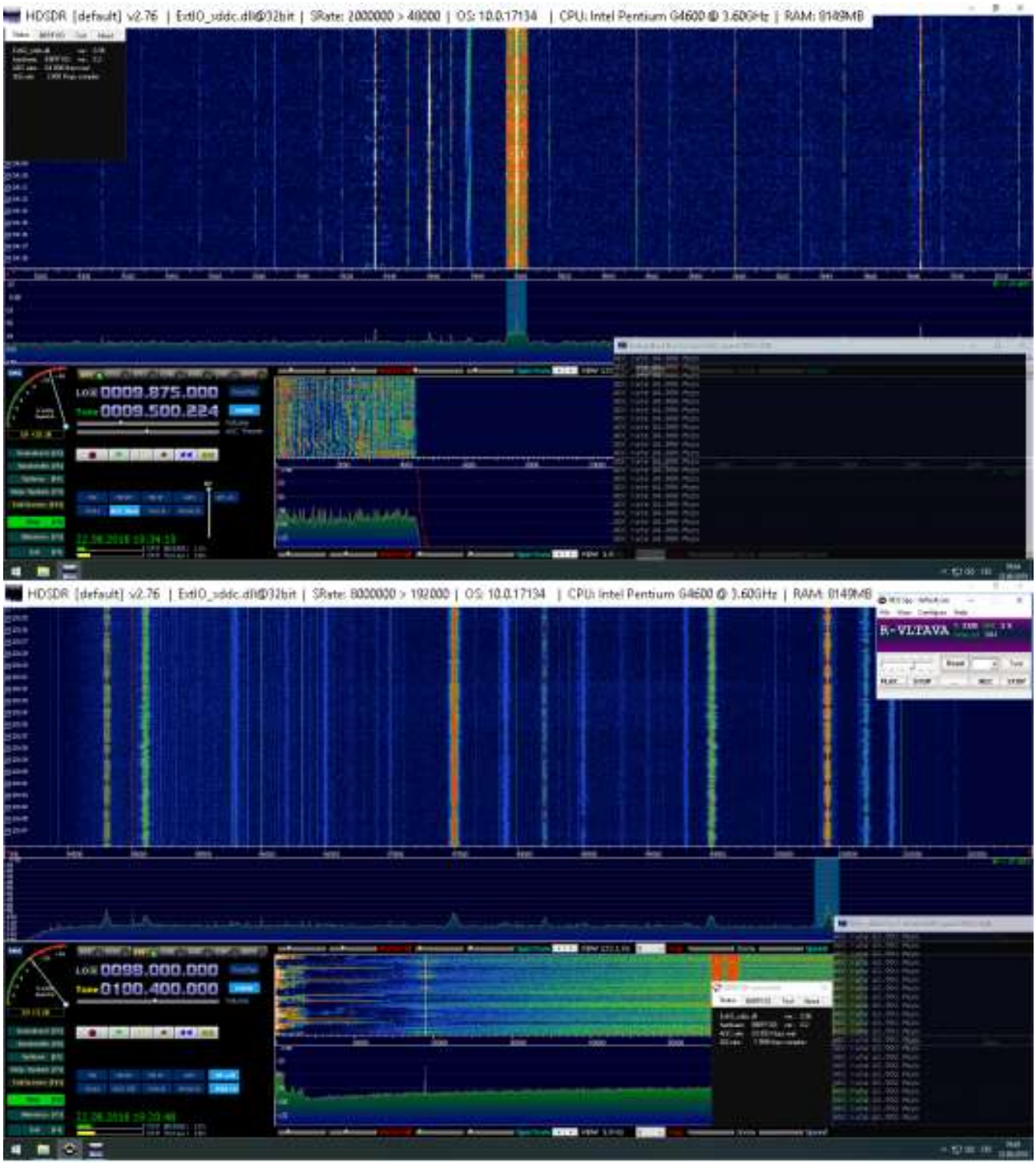
- Referring to transformers T1,T2,T3. The Coilcraft WBC4-6TL type is better than the WBC4-1TL. The insertion loss is 0.65 dB instead of 1 dB. (WBC datasheet)
Note that terminals 4 and 6 of the primary winding are interchanged in the electric scheme and the SMD footprint used in the PCB. This does not affect performance and will be corrected in the future PCB version to match the original SMD footprint.

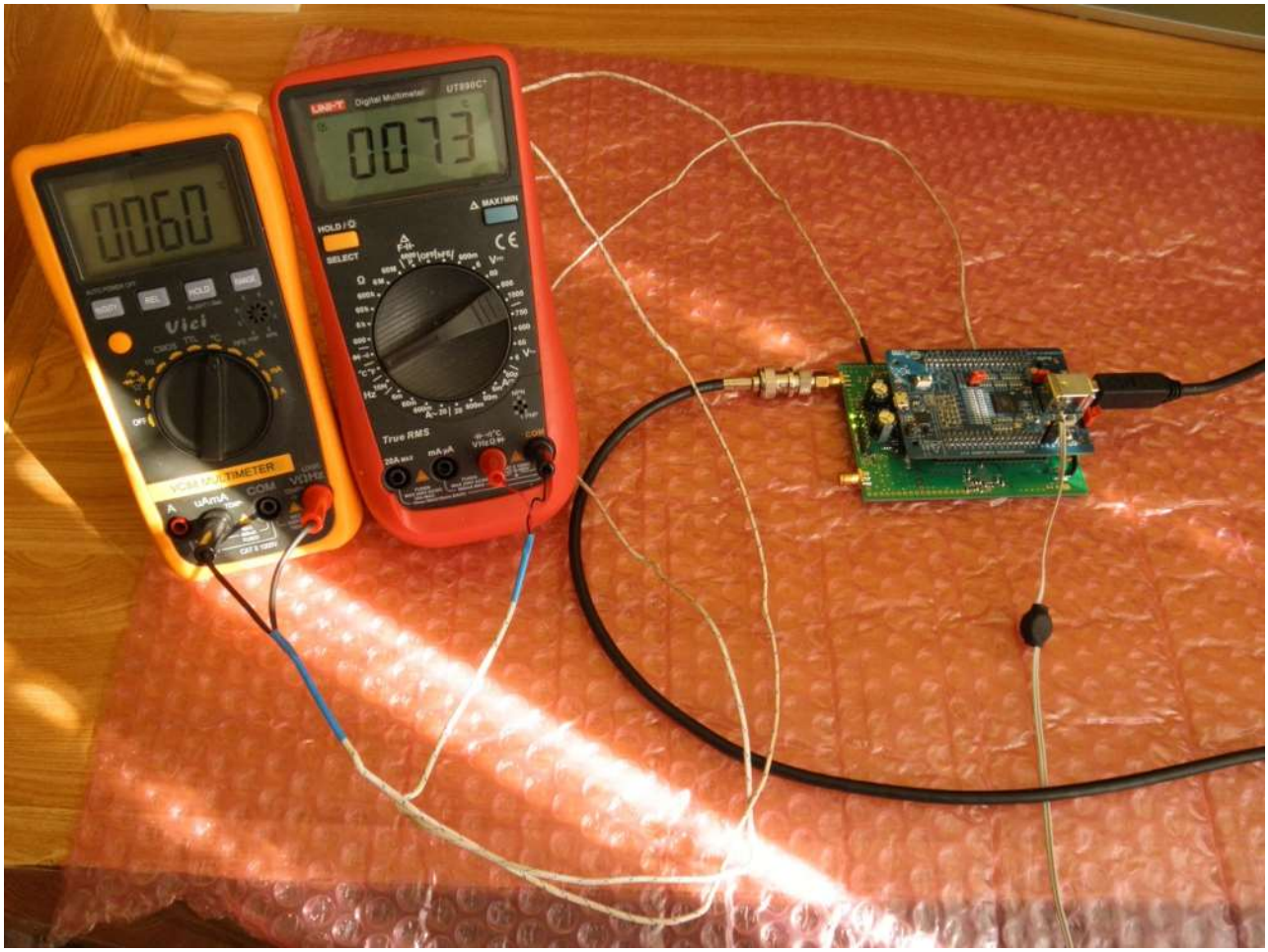
- The Q1,Q2,Q3,Q4 SMD footprint is corrected in RC3 PCB.

- C44 and C55 silkscreen locations are swapped in the BBRF103-2 original PCB. Footprint is corrected in RC3 PCB layout.

Prototype was tested on desktop PC equipped with Intel Pentium G4600 CPU and B150 chipset USB 3.0 hub controller.
Hereafter some pictures of prototype under testing.


SUPERSPEED EXPLORER KIT


HDSDR [default] v2.76 | ExtIO_sddc.dll@32bit | SRate: 2000000 > 48000 | OS: 10.0.17134 | CPU: Intel Pentium G4600 @ 3.60GHz | RAM: 0149MB
LOF 0009.875.000
Tune 0009.500.224
HDSDR [default] v2.76 | ExtIO_sddc.dll@32bit | SRate: 8000000 > 192000 | OS: 10.0.17134 | CPU: Intel Pentium G4600 @ 3.60GHz | RAM: 0149MB
R-VLTAVA
LOF 0099.000.000
Tune 0100.400.000



He made some test of temperature of the ADC and R820T2 in VHF mode.
“There are no heatsinks on ADC and R820T2. The temperature of the ADC reached 73 ° C while the R820T2 reached 60 ° C at 100MHz. Room temperature was 27 ° C.“

WARNING: notice that this description is a BETA test version without any warranty and is intended for non-commercial purposes.

Radek’s PCB design files can be downloaded here: http://www.steila.com/radek/BBRF103_2_RC3.zip
The archive contains:
BBRF103-RC3.sch
BBRF103-RC3.brd
BBRF103.ods
license.txt

Finally please notice that ExtIOsddc.dll ver 0.96 software does not yet control the antenna power via dll panel window and to enable VHF (R820T2) mode you must undefine _NO_TUNER_ in config.h and recompile.

email: ik1xpv AT gmail DOT com

BBRF103 Construction notes

04 May, 2019

Twitter Facebook Google+ Linkedin

These days I assembled a second prototype of BBRF103 receiver. Here are some notes.

Power supply

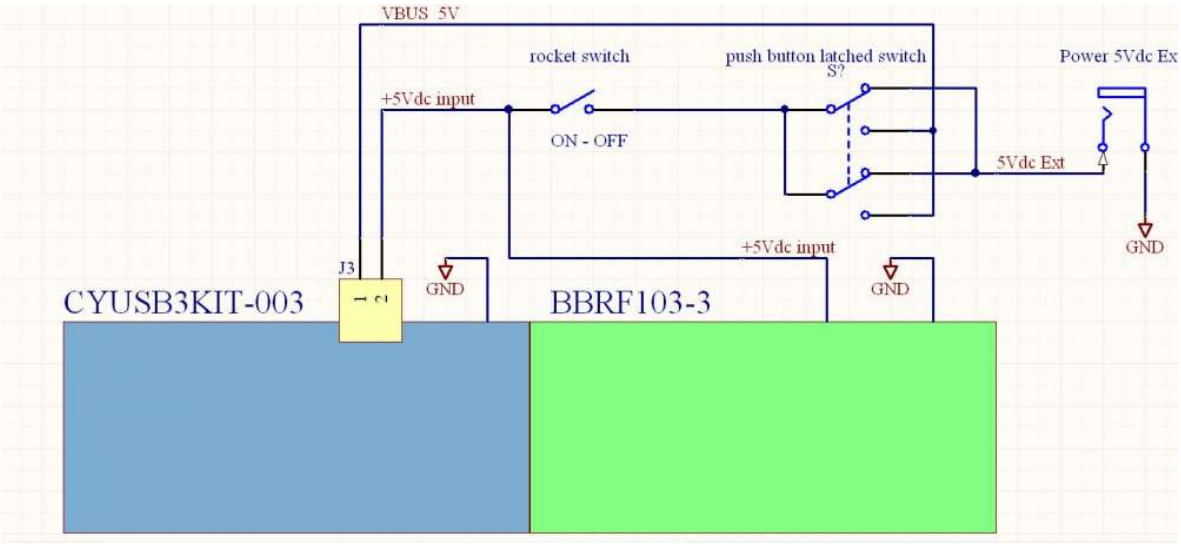
I made a different power harness. The purpose is to allow the circuit to be powered by the USB cable with VBUS or via a separate 5Volt power connector.

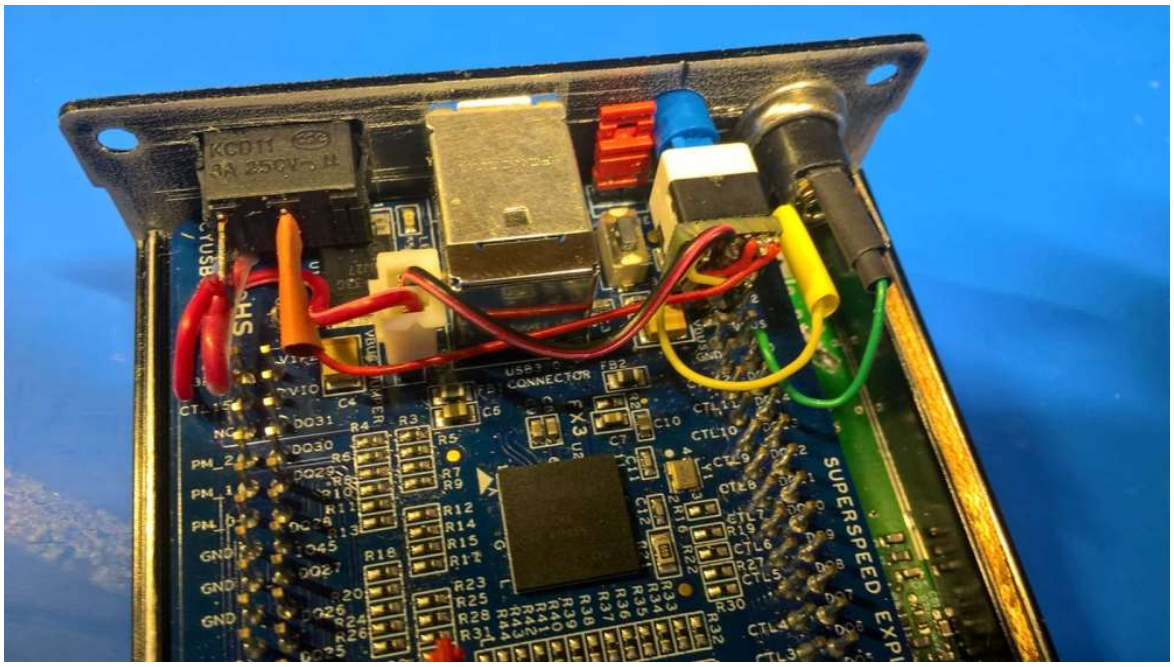
So far in the diagrams of BBRF103 the input of the 5V power supply to the PCB is taken from the VBUS of CYUSB3KIT-003.

I noticed on my two prototypes some problem related to the intervention of the protection circuit (U11 sheet 2 of 8, CYUSB3KIT scheme) on the Cypress plate and caused by the current increase required in some phases of power-up of the PCB BBRF103 and R820T2 use.

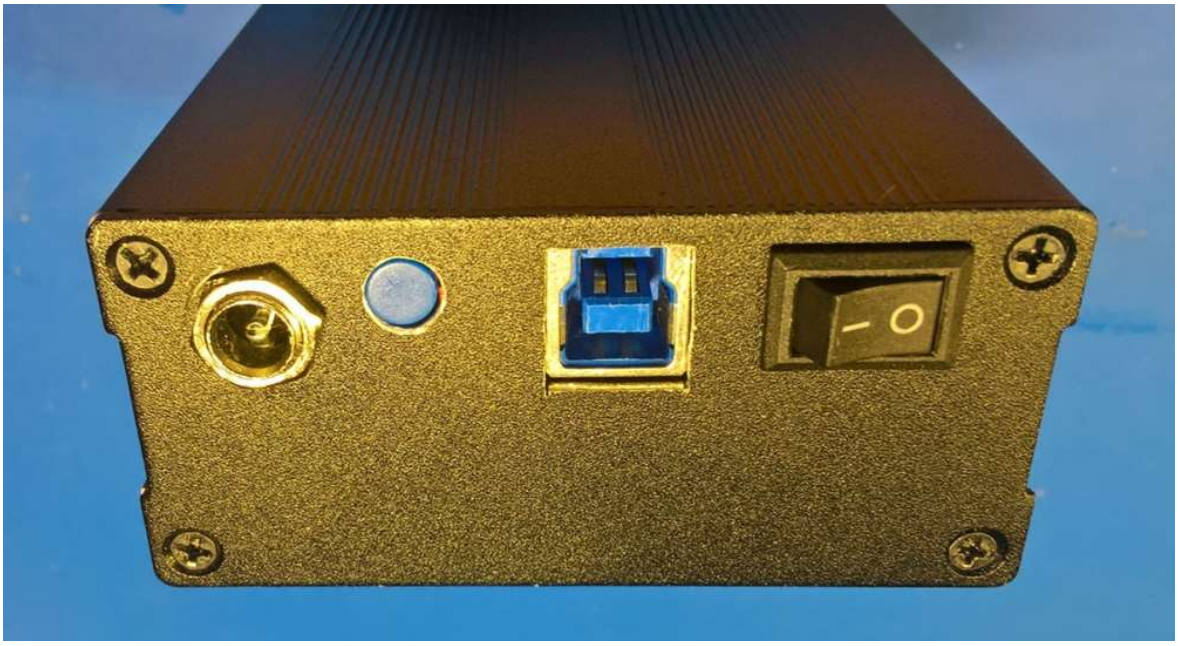
I modified the input of the 5Volt and connected it upstream of the protection circuit using the J3 jumper as a connector.

The scheme is as follows:





The current consumption of complete BBRF103 using Tuner R820T2 is about 530mA at 5Vdc.



From left to right

- 5Volt dc external input (optional)

- latched pushbutton VBUS / extenal 5V

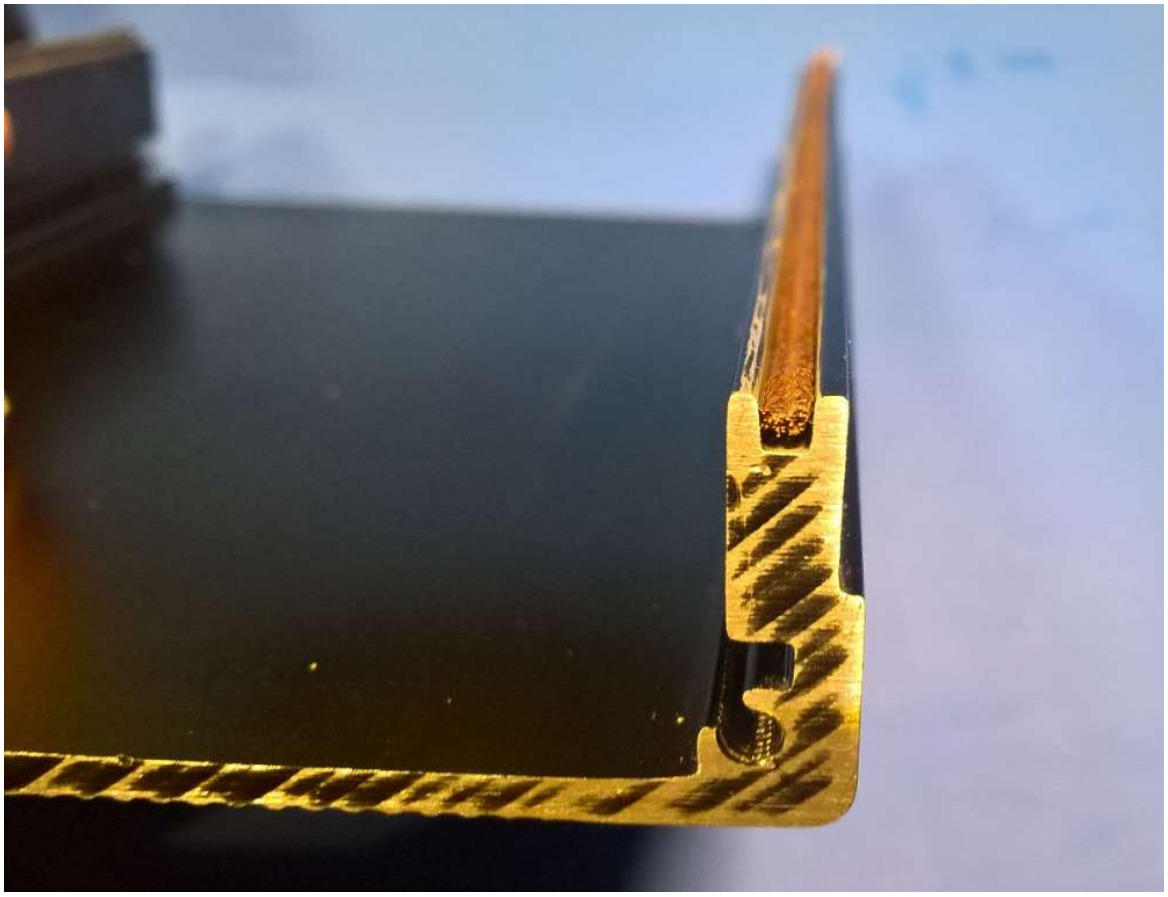
- USB3 connector

- ON / OFF

Shielding box

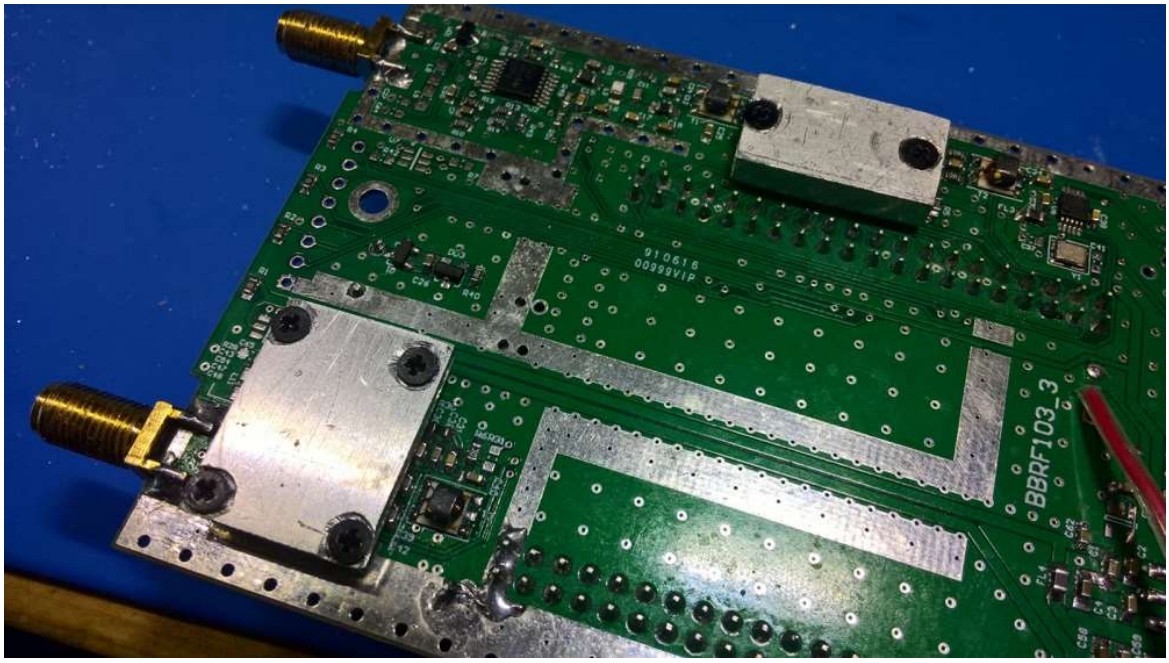
The box used is made of aluminum, it has dimensions 100x76x35mm, currently at Banggood it is available only in a golden color. The surface is brushed and treated with an insulating process so it is necessary to sandpaper the contact surfaces to obtain an electrical contact when it closes.

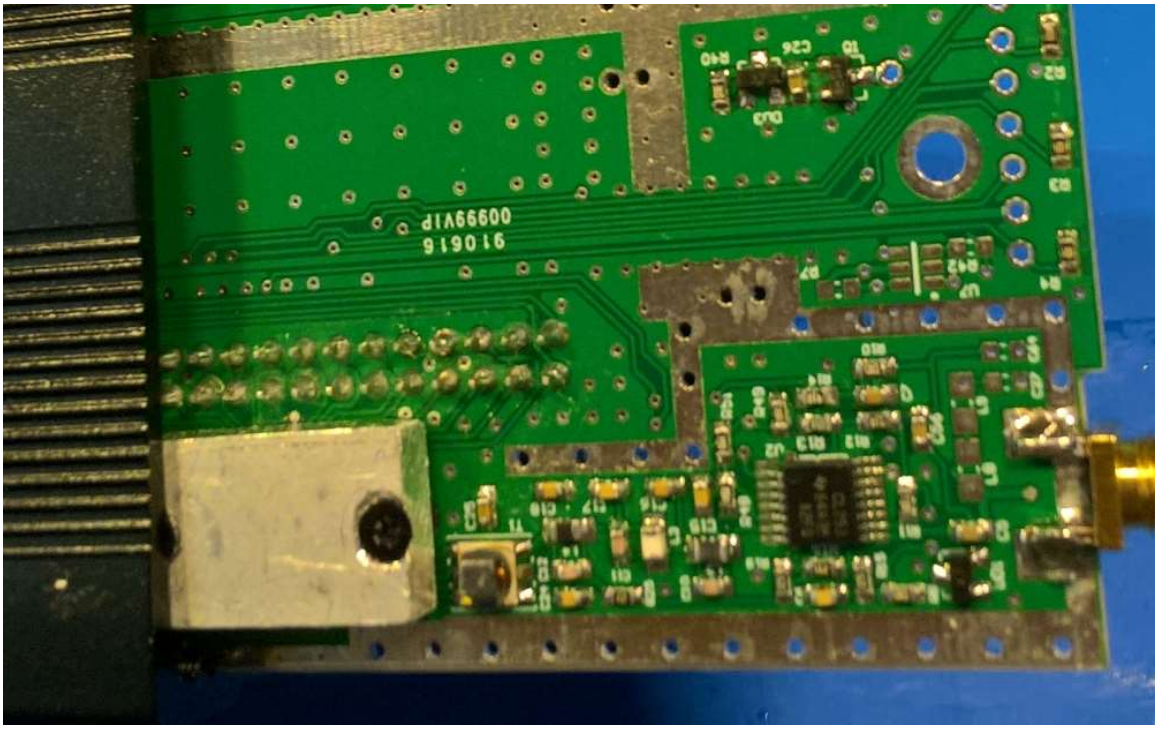
In the grooves on the long sides I inserted after scratching the rail a strip of Desoldering Copper Wick as a contact gasket.

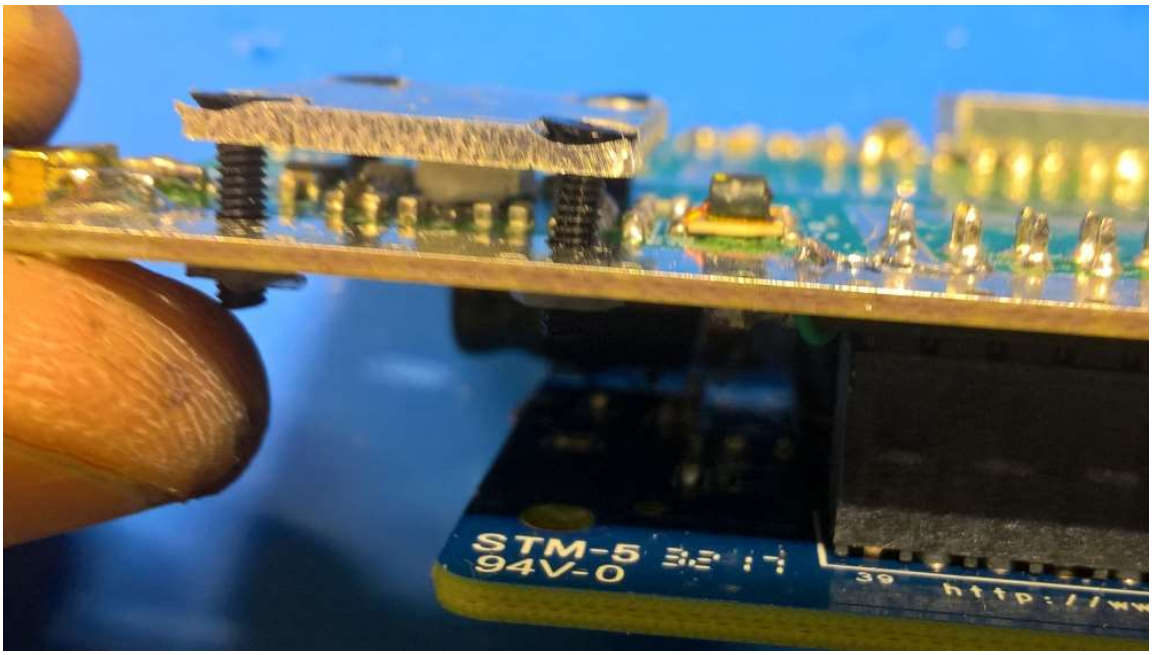


Heat dissipation

The ADC LTC2217 and the tuner R820T dissipate more than one Watt in heat. To keep the temperature of the chips lower, I tried two passive aluminium heat sinks that lead the heat to the aluminium box, which acts as a heat sink, reducing the temperature of the chips by 20-30° C.







The radiators are fixed with some M2 nylon insulating screws and nuts. I used a slightly modified PCB from BBRF103-2 layout.

Stand alone static current test

Assembling the pcb I have tried a simple method to verify errors of power supply rails.

Each block of the circuit is powered separately through a filter inductor. So not mounting these components it is easy to measure the current absorbed by various blocks on the PCB BBRF103-2 before connecting the CYUSB3KIT-003 board.

Here is a table with the measured power consumption of three blocks in a static way.
I have not mounted FL1, FL2, FL3.
I connected a 5Volt /500mA power supply to the LDO power supply of the plate and
I measured the current at the unassembled inductors by injecting an external voltage, getting:

Block	Current	Measuring point
Oscillator SI5351	15-16 mA	FL3 / external power supply 3.3V
Tuner R820T2	80-85 mA	FL2 / external power supply 3.3V

ADC LTC2217	269-210 mA	FL1 / external power supply 5V (ADC gets 3.3V via LDO)

In case, inspect the component mounting and check the welds with a microscope or lens.

BBRF103 Some measurements

20 May, 2019

Twitter Facebook Google+ Linkedin

In the construction of BBRF103 the evaluation kit of Cypress for the FX3 is used as it is and the ADC printed circuit board is realized with 2 layers as a compromise to reduce the cost of realization.

I try to evaluate now with homemade measurements how close the performance is and how good it is.

Let's start with a few readings from the technical data sheet of the LTC2217 (https://www.analog.com/media/en/technical-documentation/data-sheets/2217f.pdf)

"*(pag 1) ...The LTC2217 includes 81.3dBFS Noise Floor and 100dB spurious free dynamic range (SFDR)*

(pag 24) … Digital Output Randomizer
Interference from the ADC digital outputs is sometimes unavoidable. Interference from the digital outputs may be from capacitive or inductive coupling, or coupling through the ground plane. Even a tiny coupling factor can result in discernible unwanted tones in the ADC output spectrum.
By randomizing the digital output before it is transmitted off chip, these unwanted tones can be randomized, trading a slight increase in the noise floor for a large reduction in unwanted tone amplitude.
The digital output is "Randomized" by applying an exclusive-OR logic operation between the LSB and all other data output bits. To decode, the reverse operation is applied; that is, an exclusive-OR operation is applied between the LSB and all other bits. The LSB, OF and CLKOUT output are not affected. The output Randomizer function is active when the RAND pin is high.

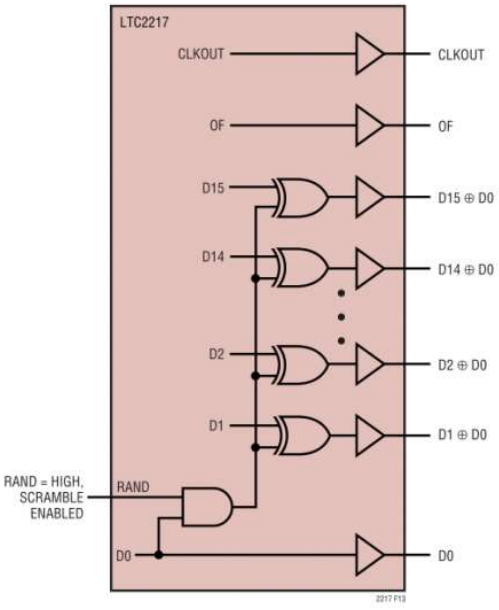


Figure 13. Functional Equivalent of Digital Output Randomizer

(pag 25)... Internal Dither
The LTC2217 is a 16-bit ADC with a very linear transfer function; however, at low input levels even slight imperfections in the transfer function will result in unwanted tones.
Small errors in the transfer function are usually a result of ADC element mismatches. An optional internal dither mode can be enabled to randomize the input location on the ADC transfer curve, resulting in improved SFDR for low signal levels.
As shown in Figure 15, the output of the sample-and-hold amplifier is summed with the output of a dither DAC. The dither DAC is driven by a long sequence pseudo-random number generator; the random number fed to the dither DAC is also subtracted from the ADC result. If the dither DAC is precisely calibrated to the ADC, very little of the dither signal will be seen at the output. The dither signal that does leak through will appear as white noise. The dither DAC is

calibrated to result in typically less than 0.5dB elevation in the noise floor of the ADC as compared to the noise floor with dither off, when a suitable input termination is provided (see Demo Board schematic DC996B).

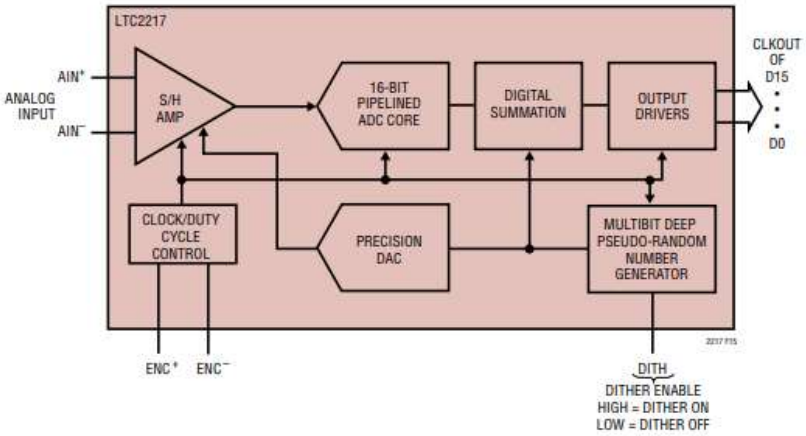


Figure 15. Functional Equivalent Block Diagram of Internal Dither Circuit

".

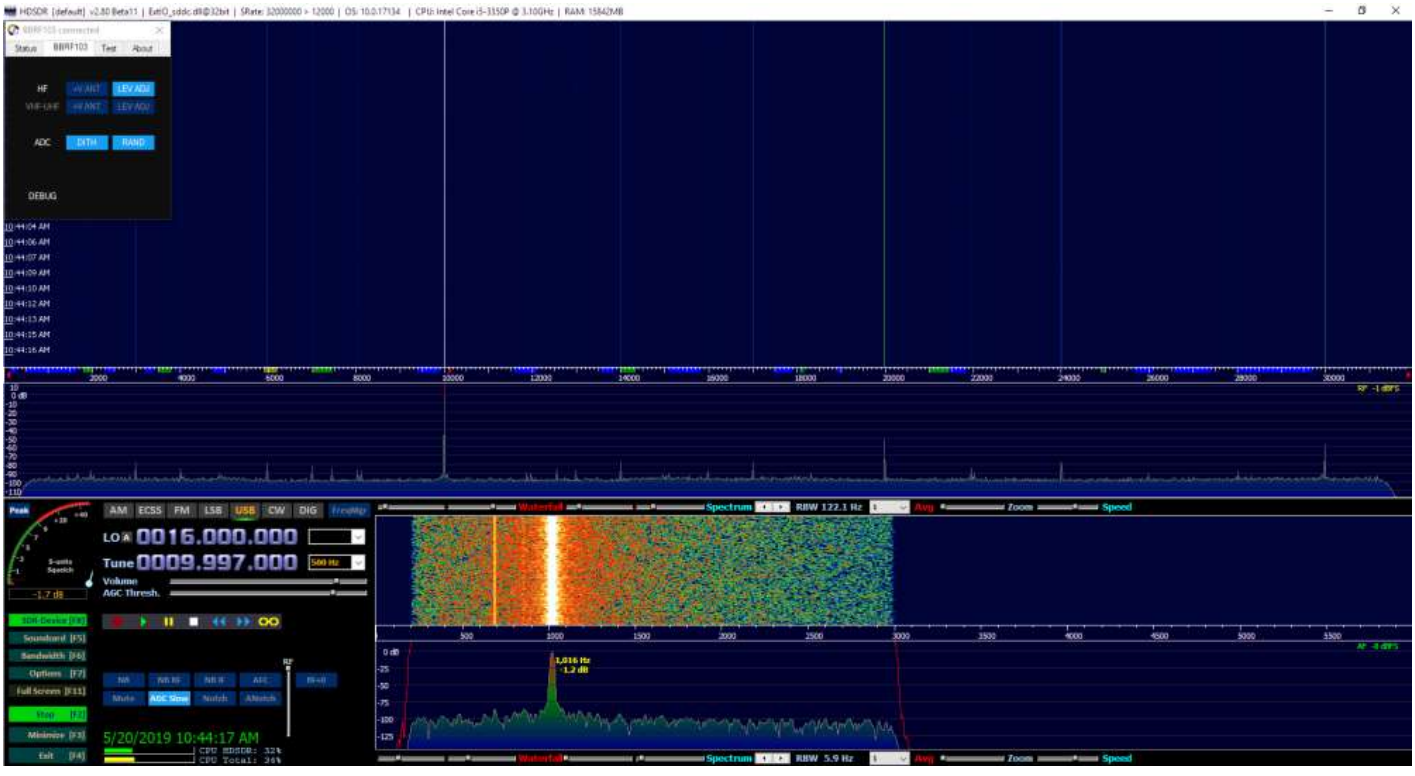
I made a sinusoidal generator using an old 10MHz TCXO followed by a ladder quartz filter made with cheap 10MHz quartz.

I calibrated the TCXO at 9.9981 MHz to pass through the quartz filter.
Before the filter, an SBF5089Z amplifier allows to obtain after the filter a level of +2dBm on 50 Ohm.
Finally, a series of resistive attenuators allows you to adjust the output level.
The aim is to obtain a generator with good dynamics and low noise. The result has 2nd and 3rd harmonic level at -50dB.

Here are a few measures

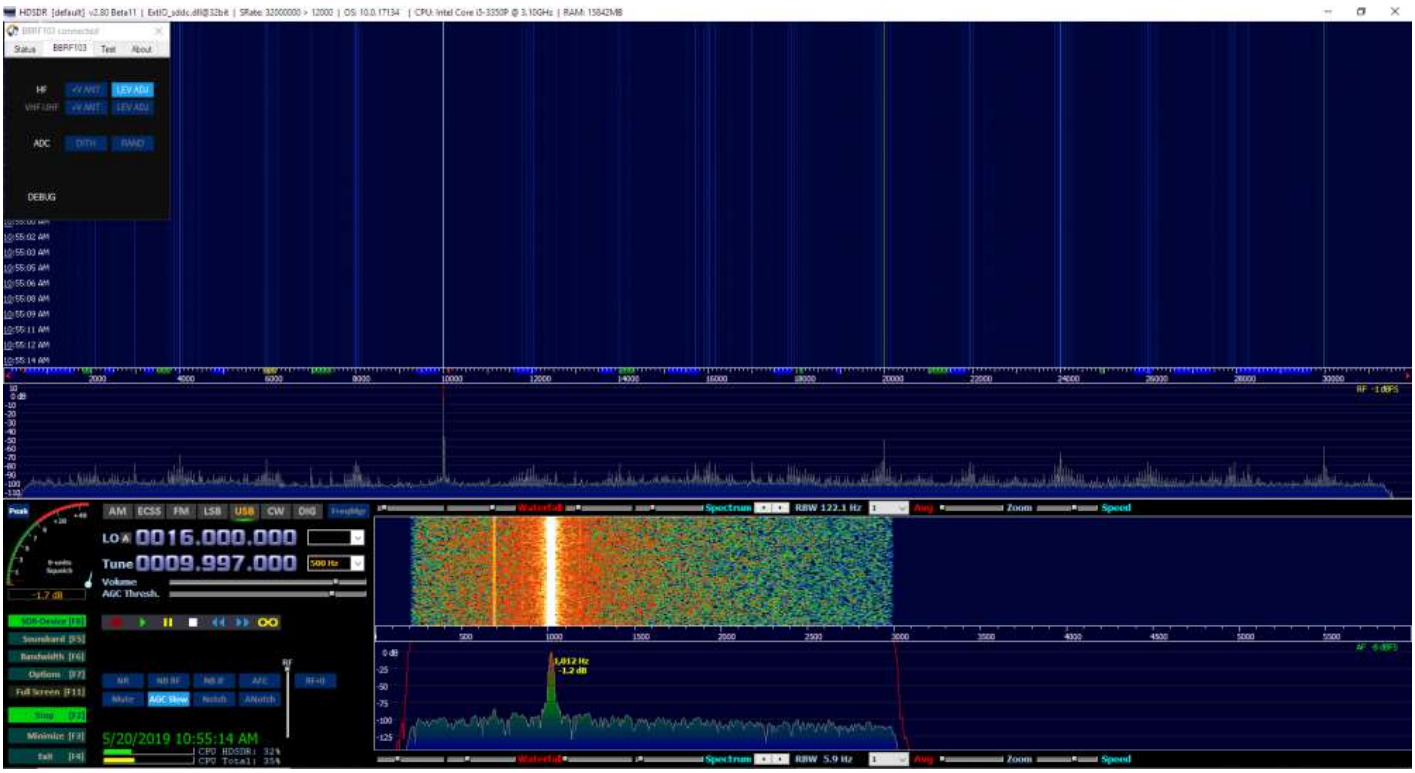
BBRF103 HF antenna is connected to the attenuator output.
BBRF103 control panel allows to activate dither and randomize.

Hereafter the level is -1dBFS, DITH and RAND are active.



High resolution image link

Hereafter the same -1dBFS level, DITH and RAND are OFF



High resolution image link

With strong and stationary signals the interference of the data bus is evident and the use of dither and randomize is effective.
Referring to the ADC datasheet with reference to these approximate measurements, it seems to me that the current layout of the 2-layer PCB and the adc databus, which also continues in the evaluation PCB of the FX3, worsen the spurious performance by 10-15 dB compared to the optimal layout of the PCB.

It would be possible to realize a single multilayer PCB including FX3 and ADC with better performances following the datasheet's layout indications.

Receivers similar to BBRF103 ?

08 June, 2020

Twitter Facebook Google+ Linkedin

Some receivers with an architecture similar to BBRF103 appeared in the web during this look down time.
I tried to contact the authors not for copyright issues, being BBRF103 completely open source but to exchange some experiences. A contacted person wrote me that the project is new and fully original, but I think I reached the re-seller and not the designer.
So I signal my interest in talking with the real authors of the devices.
I list the devices with links to images that are in these weeks at the following links.

If someone has a review or any other info please send it to me.

Thanks, everyone, ik1xpv AT gmail DOT com

Update 15 August, 2020

- If you want to give a try to BBRF103 software with these receivers, here the compiled ExtIO_sddc.dll : v.0.96 (HF only) and v.0.98 (HF and VHF).

- The Dragonfly RX-666 SDR © Bjarne Mjelde, arcticdx.blogspot.com

Update 19 August, 2020

- The RX888 SDR – Up Close Photos at https://swling.com/blog/2020/08/the-rx888-sdr-up-close-photos/

- The RX-888 Team sent me the link to they *RX-888 software* at

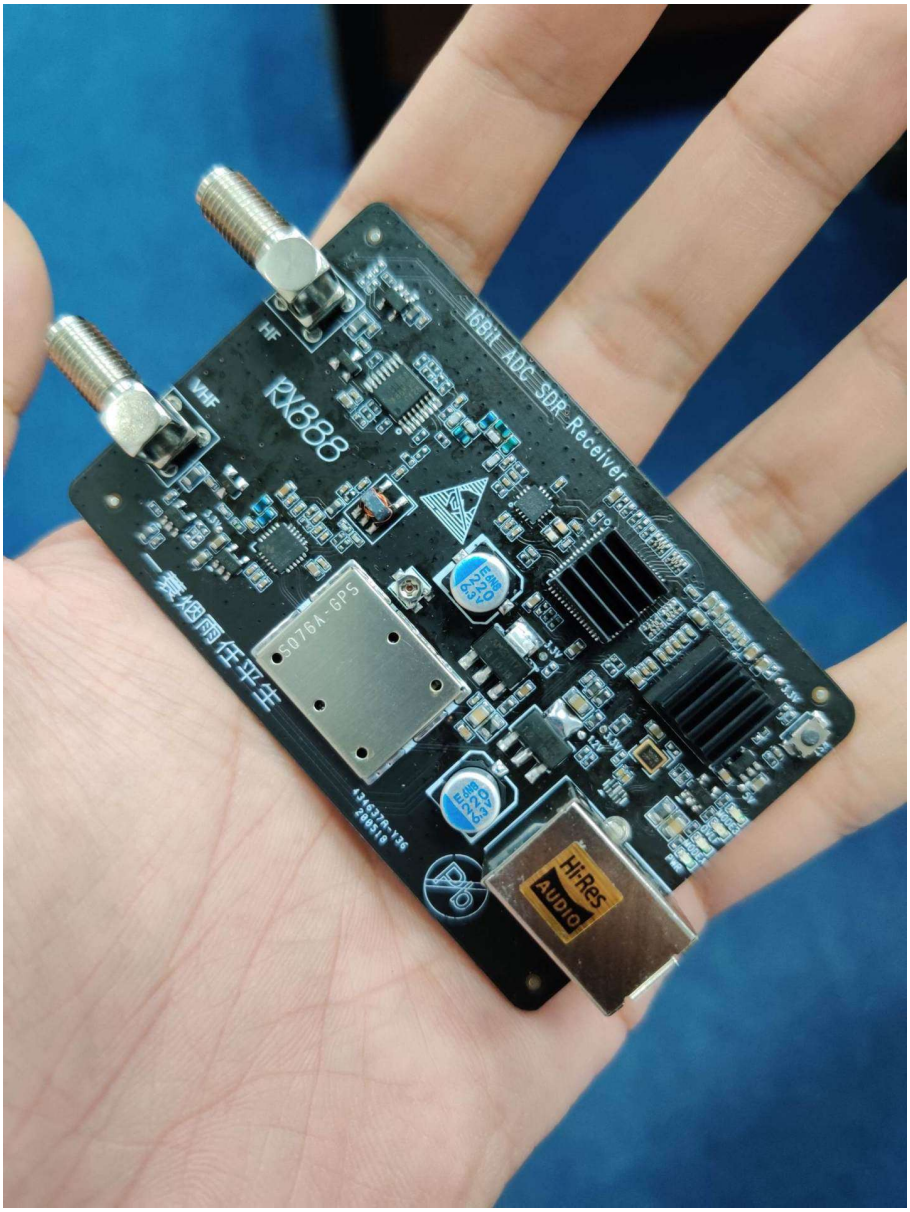
https://drive.google.com/file/d/1hqV2-ZCI1iDz7IvOsBS-YR5XGLpOd6ER/view?usp=sharing

RX-888

source:

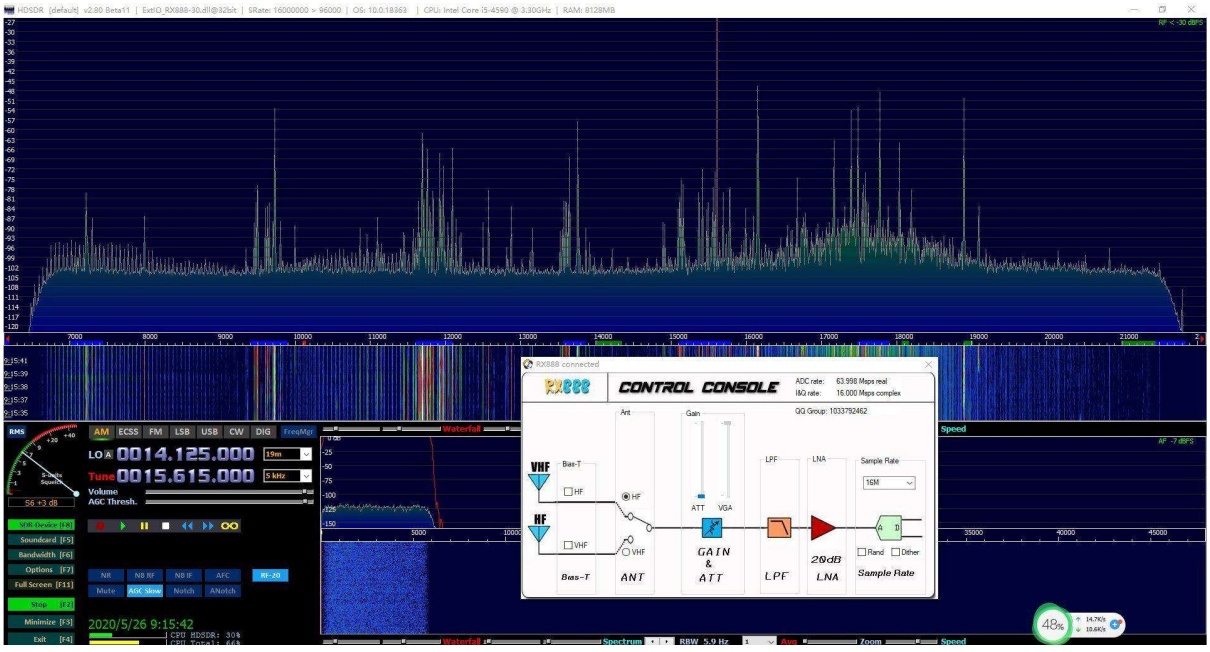
http://www.radioscanner.ru/forum/topic46950-28.html

https://twitter.com/fei666888



http://www.radioscanner.ru/uploader/2020/ezawtqsueaeoa-z.jpg

http://www.radioscanner.ru/uploader/2020/78934789iw.jpg



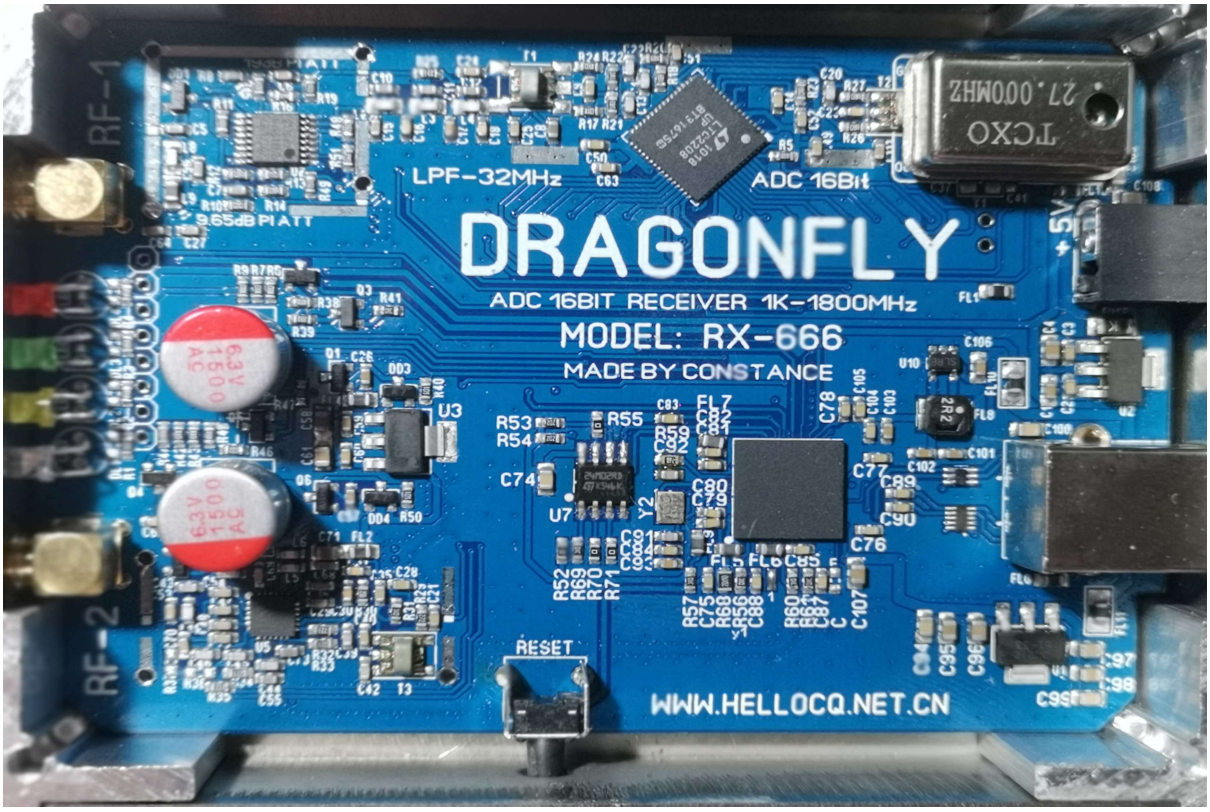
http://www.radioscanner.ru/uploader/2020/ezawdhruwaaz-gq.jpg

RX-666 ?

source:

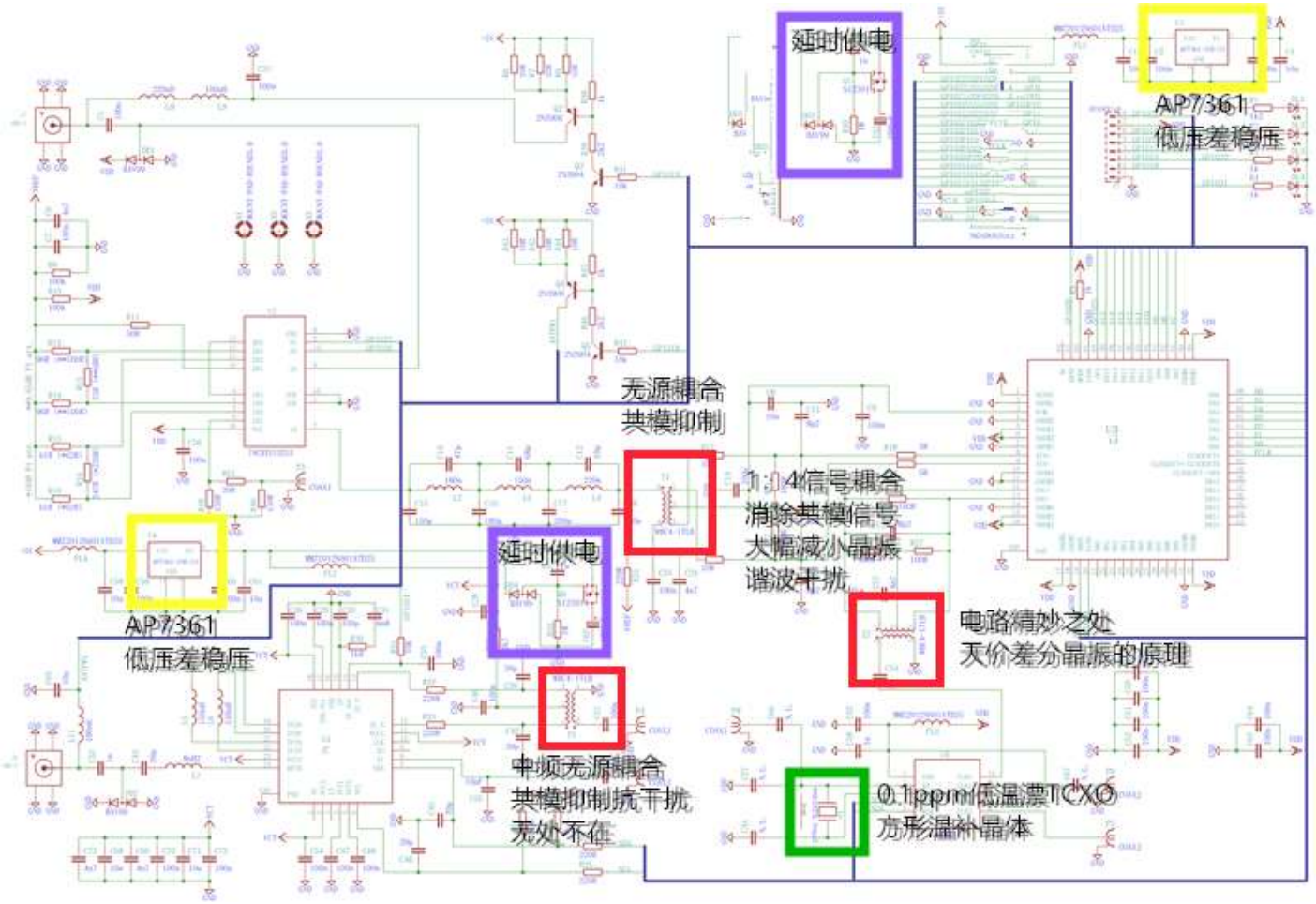
TAOBAO https://item.taobao.com/item.htm?spm=2013.1.w4023-1262356524.17.4ea334a9ZaAa2W&id=617774885587

Aliexpress https://www.aliexpress.com/item/4001080658440.html?spm=a2g0o.detail.1000023.9.6ccb4491vlqtWJ



https://img.alicdn.com/imgextra/i2/22088642/O1CN01wrOfUF2Di5PDC8KRa_!!22088642.jpg

https://img.alicdn.com/imgextra/i4/22088642/O1CN017bLC3v2Di5OpbZuEj_!!22088642.jpg



https://img.alicdn.com/imgextra/i3/22088642/O1CN01B86EWy2Di5P6pKtn4_!!22088642.jpg

https://img.alicdn.com/imgextra/i3/22088642/O1CN01hqZ7N92Di5OrG2ydq_!!22088642.jpg