

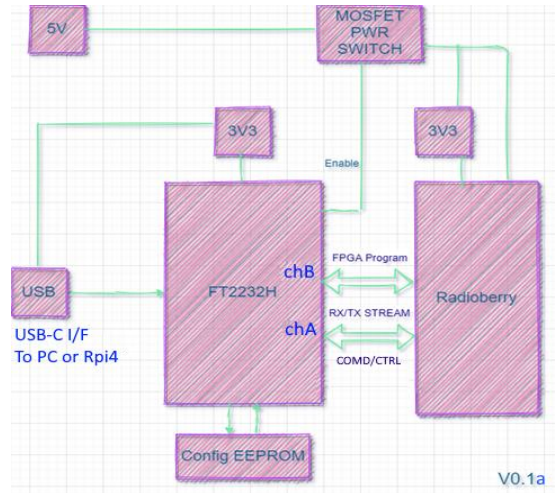
Trying the Radioberry Juice board.

Trying the Radioberry-Juice, which supports up to 6 rx slices with 384 KHz sampling.

<https://groups.google.com/g/radioberry/c/vh8BfdKJzVg/m/nIOUOODXBgAJ>

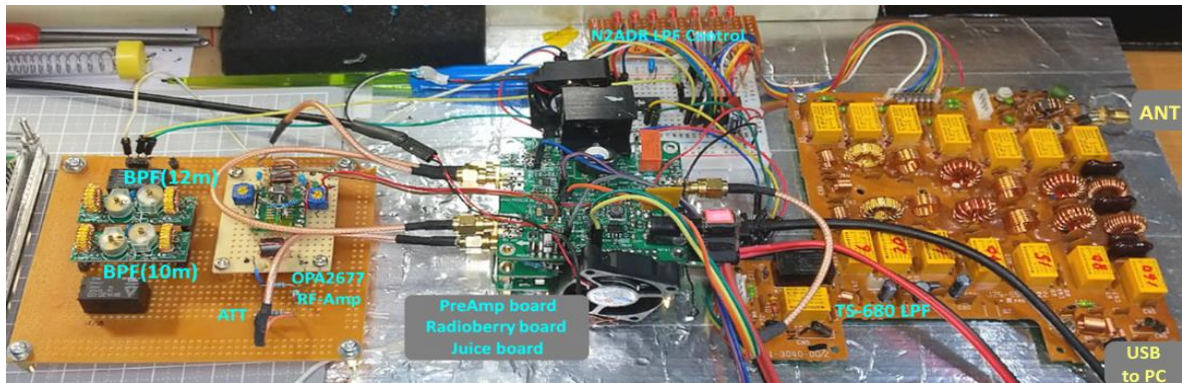
<http://www.pa3gsb.nl/2021/05/31/radioberry-juice/>

<http://www.pa3gsb.nl/2021/07/22/radioberry-juice-part-2/>

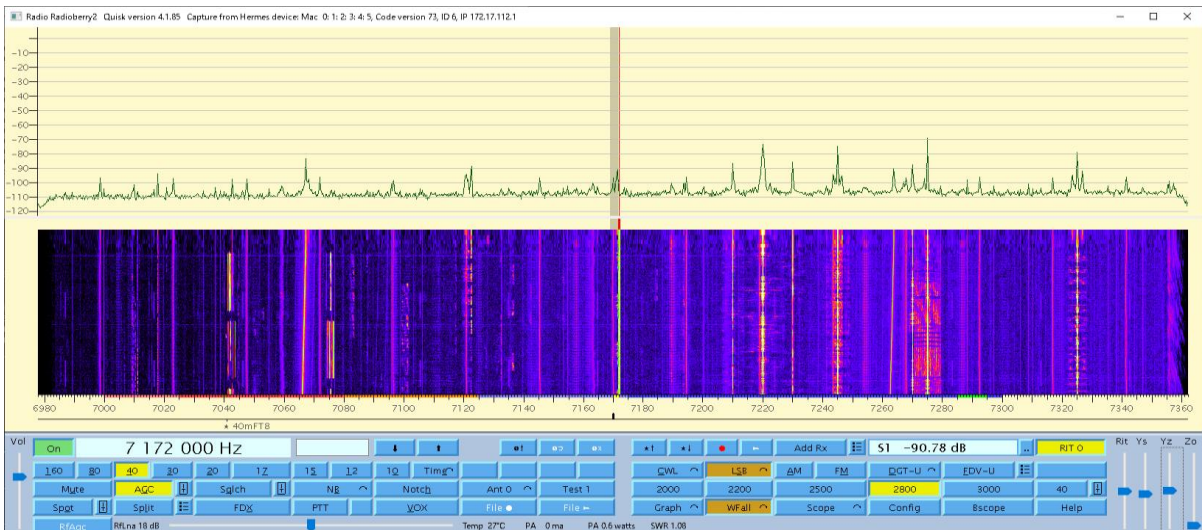


The FT2232H uses the 254 protocol; having a clock of 60Mhz. (a max transfer of $60\text{Mhz} * 8 \text{ bits} \Rightarrow 480\text{Mbps}$)
→ USB effective transfer speed is 240~320Mbps (USB2.0 PC).

Trying the Radioberry-juice TRX system

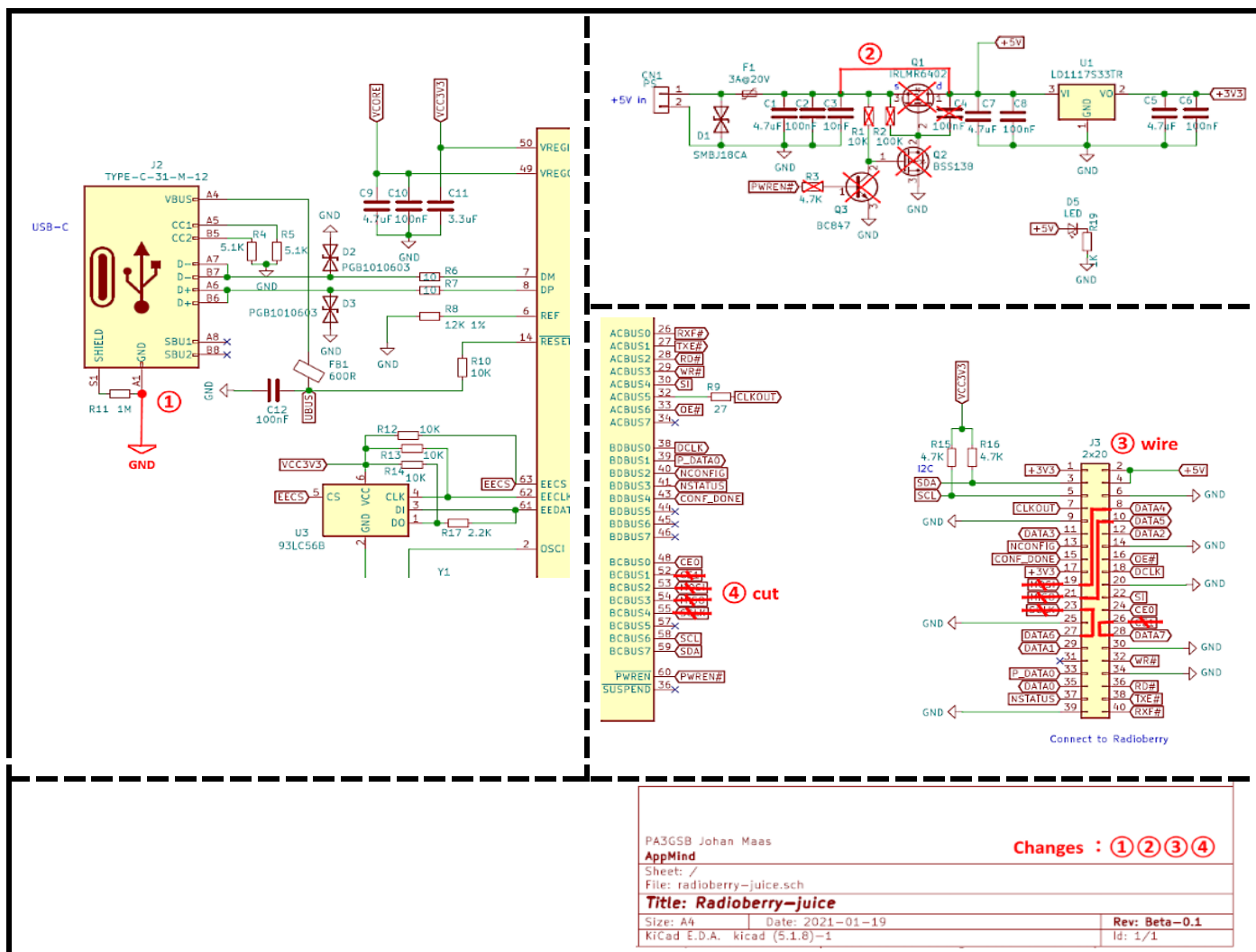


Quisk RX(384kHz): Covers 6.98MHz~7.36MHz



Juice hardware changes. (Beta0.1)

See <https://groups.google.com/g/radioberry/c/15BMIses-FM>
 → Setup Software and Hardware mods.pdf



Juice Software

Firmware-gateway (6RX, 1TX)

Commits on Aug 28, 2021: "juice tx without preamp"

<https://github.com/pa3gsb/Radioberry-2.x/tree/master/juice/firmware>

w10-Makefile for Win10, linux-Makefile for Linux(RPi4)
 radioberry.rbf is gateway for Radioberry-2(CL025)

Gateway source

Commits on Aug 28, 2021: "juice tx without preamp"

https://github.com/softerhardware/Hermes-Lite2/tree/master/gateway/variants/radioberry_juice_cl025

FTDI USB Driver

Download URL: <https://ftdichip.com/drivers/d2xx-drivers/>

- 2.12.36.4 for Win10 64bit
- 1.4.24 ARMv7 hard-float *** and "ReadMe" Info for Linux_RPi4

Note:

The way of preventing ftdi_sio from controlling FTDI devices.

FTDI d2xx conflict with ftdi_sio on linux. :

<https://stackoverflow.com/questions/33981056/>

ftdi-d2xx-conflict-with-ftdi-sio-on-linux-how-to-remove-ftdi-sio-automatically

GCC compiler on Windows : Mingw-w64

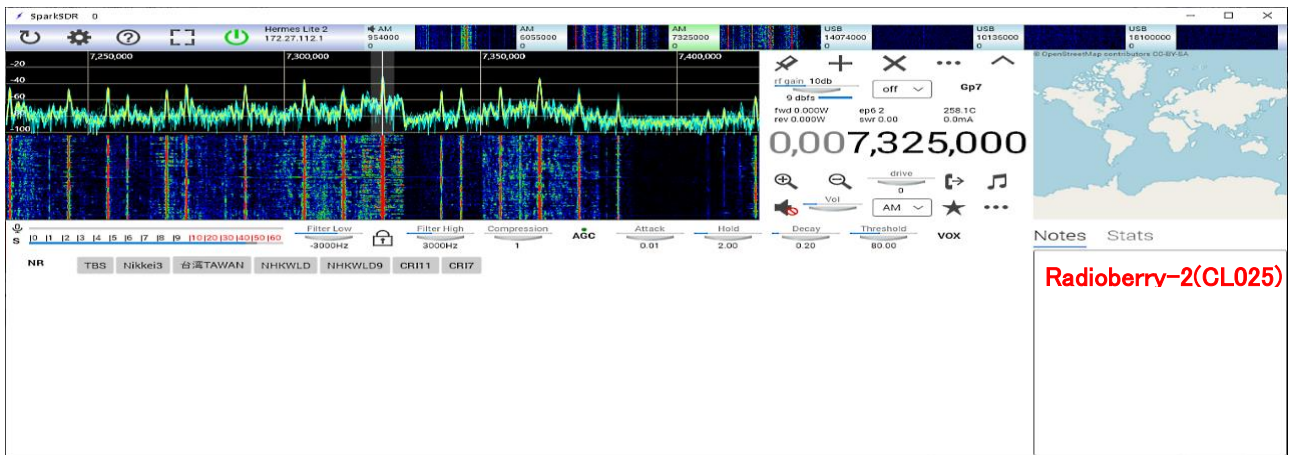
Download URL: <https://www.mingw-w64.org/downloads/>

- MingW-W64-builds 7.2.0/5.0.3

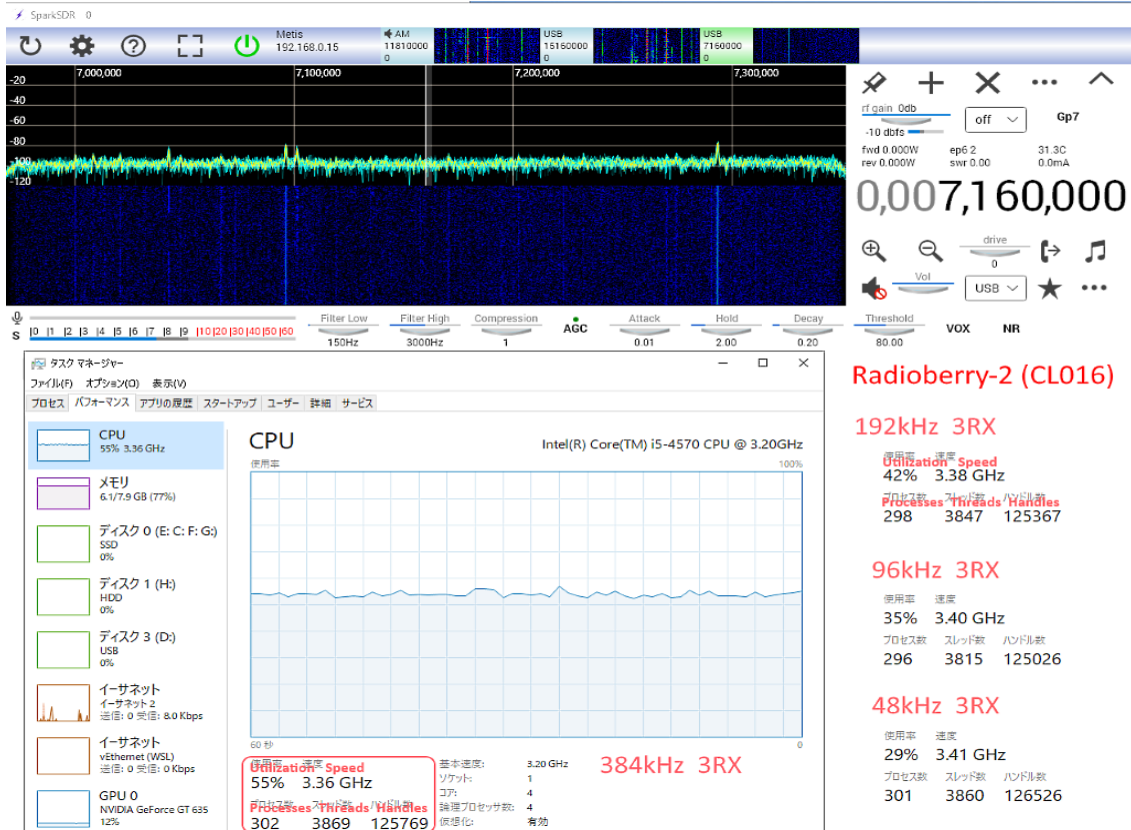
RX performance

(1) Win10-PC(ver.21H1) Core-i5-4570 CPU@3.2GHz RAM:8GB

SparkSDR_2.0.8.6 192kHz 6receivers : Good receiving ! (Some minor cracks and drops at 6x384 kHz receivers)



SparkSDR_2.0.8.6 384kHz 3receivers : Good receiving ! ← Using CL016 because I broke my Radioberry's FPGA CL025.



Radioberry-2 Power consumption.

3receivers

fsampling	I [A]	Power [W]
48kHz	0.533	2.7
96kHz	0.566	2.8
192kHz	0.593	3.0
384kHz	0.645	3.2

1receiver

fsampling	I [A]	Power [W]
48kHz	0.548	2.7
96kHz	0.562	2.8
192kHz	0.598	3.0
384kHz	0.642	3.2

Radioberry Juice

(2) RPi4B_2GB

piHPSDR 384kHz 2receivers : Some minor cracks and drops (Good Receiving at 2x192 kHz receivers)

```
$ sudo taskset -o3 ./radioberry-juice016_0829
```

piHPSDR: HermesLite V2 (Protocol 1 v7.3) 127.0.0.1 (00:01:02:03:04:05) on TCP
VFO A: 7.053000 VFO B: 9.750000
Zoom: 1 Pan: []
AP: 55 AGC: 103 RX-GAIN: 21
Mic: 23 Drive: 90 Squelch: 0

```
top - 23:09:49 up 47 min, 1 user, load average: 3.68, 3.17, 2.89
Tasks: 158 total, 4 running, 154 sleeping, 0 stopped, 0 zombie
Cpu(s): 41.9 us, 18.2 sy, 0.0 ni, 35.2 id, 0.0 wa, 0.0 hi, 4.6 si, 0.0 st
Mem: 1724.2 total, 1089.6 free, 310.5 used, 324.1 buff/cache
MiB Swap: 100.0 total, 100.0 free, 0.0 used, 1263.6 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
6123	pi	20	0	554516	167208	31780	S	109.9	9.5	7:49.56	pihpsdr.org
1402	root	20	0	99272	1820	1608	S	70.9	0.1	24:28.74	radioberry-juic
538	root	20	0	3444	22232	14568	S	37.1	1.3	0:33.88	vncserver-x11-c
591	root	20	0	228848	81262	62970	R	33.4	4.6	21:22.82	Xorg
232	root	-2	0	0	0	0	S	1.8	0.0	0:36.46	v3d_bin
233	root	-2	0	0	0	0	S	1.0	0.0	0:38.06	vnd_render
608	root	20	0	13152	9440	8964	S	1.0	0.5	0:28.19	vncagent
75	root	-1	-19	0	0	0	S	0.7	0.0	0:09.01	vchiq-slot/0
1307	pi	20	0	3680	2188	1928	S	0.7	0.1	0:13.17	watch
12	root	20	0	0	0	0	R	0.3	0.0	0:04.17	rcu_sched
17	root	20	0	0	0	0	S	0.3	0.0	0:01.23	kcfirad/1
285	root	20	0	0	0	0	S	0.3	0.0	0:02.47	brcmf_wdog/smc1
1306	pi	20	0	97784	30548	23756	S	0.3	1.7	0:07.83	lxterminal
25431	pi	20	0	6956	3128	2600	R	0.3	0.2	0:01.50	top

piHPSDR 384kHz 2receivers without "taskset" : NG, Many cracks and drops

```
./radioberry-juice016_0829 ( without "taskset" )
```

piHPSDR: HermesLite V2 (Protocol 1 v7.3) 127.0.0.1 (00:01:02:03:04:05) on TCP
VFO A: 7.074500 VFO B: 9.750000
Zoom: 1 Pan: []
AP: 57 AGC: 101 RX-GAIN: 12
Mic: 23 Drive: 48 Squelch: 0

```
top - 22:40:42 up 10 min, 1 user, load average: 3.28, 2.10, 1.09
Tasks: 160 total, 2 running, 158 sleeping, 0 stopped, 0 zombie
Cpu(s): 42.7 us, 18.7 sy, 0.0 ni, 35.4 id, 0.0 wa, 0.0 hi, 3.2 si, 0.0 st
Mem: 1724.2 total, 1084.0 free, 307.7 used, 321.7 buff/cache
MiB Swap: 100.0 total, 100.0 free, 0.0 used, 1269.0 avail Mem
```

PID	USER	PR	NI	VIRT	RES	SHR	S	%CPU	%MEM	TIME+	COMMAND
11180	pi	20	0	553380	167360	32128	S	105.0	9.5	0:42.55	pihpsdr.org
11128	pi	20	0	99272	1876	1664	S	73.8	0.1	0:45.23	radioberry-juic
527	root	20	0	34072	22112	14592	S	37.5	1.3	1:24.96	vncserver-x11-c
589	root	20	0	228744	82088	64780	R	33.2	4.6	1:47.89	Xorg
69	root	20	0	0	0	0	I	1.8	0.0	0:01.52	kworkeu/us:1-events+
145	root	20	0	0	0	0	I	1.8	0.0	0:03.14	kworkeu/us:2-brcmf_w+
207	root	-2	0	0	0	0	S	1.0	0.0	0:03.46	v3d_bin
606	root	20	0	13152	9504	9024	S	1.0	0.5	0:05.28	vncagent
1216	pi	20	0	3816	2352	1940	S	1.0	0.1	0:04.46	watch
32	root	20	0	0	0	0	I	0.7	0.0	0:00.69	kworkeu/g:1-events
205	root	-2	0	0	0	0	S	0.7	0.0	0:02.84	v3d_bin

Receiving Power consumption of RPi4B and Sound.

pihpsdr : 2receivers with "taskset -c 3"

fsampling	Irspi4b [A]	Power [W]	Sound
48kHz	0.83	4.2	○
96kHz	0.87	4.4	○
192kHz	0.91	4.6	○
384kHz	0.96	4.8	△

○ : Good

△ : Some minor cracks and drops.

pihpsdr : 2receivers, without "taskset"

fsampling	Sound
48kHz	○
96kHz	△
192kHz	×
384kHz	×

× : cracks and drops.

* The use of the taskset command to improve the receive in RPi4B was taught by Johan's email. Following is a summary of what Johan's email.

- Raspberry Pi4 typically idles at 600MHz and switches to the maximum speed (1500MHz) when needed. Note: The CPU frequency will down to 600MHz due to a drop in the 5V supply voltage or a rise in temperature.

• To improve the receive in RPi.

Way1: Overclocking → No big improvement at 2000MHz.

Way2: with "taskset", Dedicating one core to a Radioberry_Juice process. → Big improvement!, see table above..

Way2 avoid network performance problems of RPi4.

how to :

\$ sudo vi /boot/cmdline.txt

add "isolcpus=3" ← RPi4 CPU has 4cores(0-3). Isolate CPU Core#3 from the general scheduler Linux.

\$ sudo reboot

\$ cat /sys/devices/system/cpu/isolated ← this must return 3. Make sure that CPU 3 is currently isolated

\$ sudo taskset -c 3 ./radioberry-juice ← Dedicating Core#3 to a Radioberry_Juice process.

• Maximum Speed budgets for 2RX with RPi4

1) USB performance test. Result: 75Mbps (RPi4B_2GB)

```

=====
=====
                                Radioberry V2.0

Checking USB speed using juice board.

Build version: Linux-Juice USB performance

Have fun Johan PA3GSB

Report requests or bugs to <pa3gsb@gmail.com>.

=====
=====
                                :
FTDI driver information: 01.04.24
Init device for iq stream handling...
Init device succeeded for iq streaming using FT245 protocol.
Please wait for performance measurement to end ....
Performance test ended succesfull.

Read 1GB from the radioberry using juice board in 114.6 seconds.

Average read speed: 75.0 Mbps.
    
```

2) Calculation

Measured average read speed: 75Mbps

Within the 1032 bytes of an ethernet packet in the openHPSDR protocol.

The openHPSDR protocol has two IQ streams and communication in the gateway is simplex!

• Upstream 2RX: 864byte/pkt (72 IQ samples) ... 1RX:63, 2RX:72, 3RX:75, 4RX:76
 75Mbps/(2*24bit) * ((864byte/pkt)/(1032byte/pkt)) = 1.30Mhz IQ bandwidth (I:24bit, Q:24bit)
 and Bandscope.

Radioberry Juice

• Downstream 1TX: 504byte/pkt (63 IQ samples fixed 48kHz)

$48\text{kHz} * 2 * 16\text{bit} / 8\text{bit} / (504\text{byte} / \text{pkt}) * (1032\text{byte} / \text{pkt}) = 393\text{kHz}$ IQ bandwidth (I:16bit, Q:16bit)

• Calculation up-down stream

$1300\text{kHz} - 393\text{kHz} = 907\text{kHz}$ IQ bandwidth

-> 384kHz = 2 receivers max. (907kHz/384kHz)

-> 192kHz = 4 receivers max. (907kHz/192kHz)

//

Reference:

Raspberry Pi 4 and SDR Software for HL2

<https://groups.google.com/g/hermes-lite/c/At-Olx59LBA/m/FpoAZYapCAAJ>

** pihpsdr

April 4 pull from <https://github.com/dl1ycf/pihpsdr>

92% CPU usage (Could only enable 2 384kHz receivers with pihpsdr)

34% Xorg usage

Native ALSA backend

Some minor cracks and drops at 2X384 kHz receivers, better than the same with linhpsdr, but not as good as Quisk or SparkSDR

FAN PWM and TX-enable feature.

Results.(band_voits_enabled = 'b0')

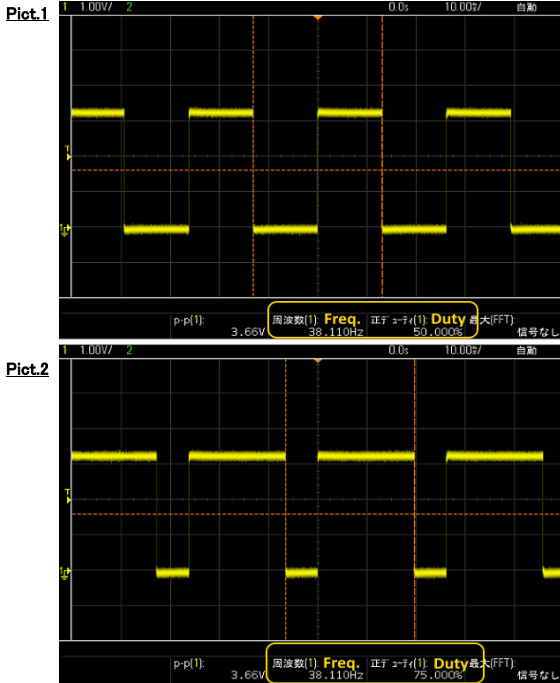
PWM Pin = P133(CN301/P10), TX enable Pin = P136(envpa: CN301/P8), P137(enbias: CN301/P7)

TX(Tune) Start

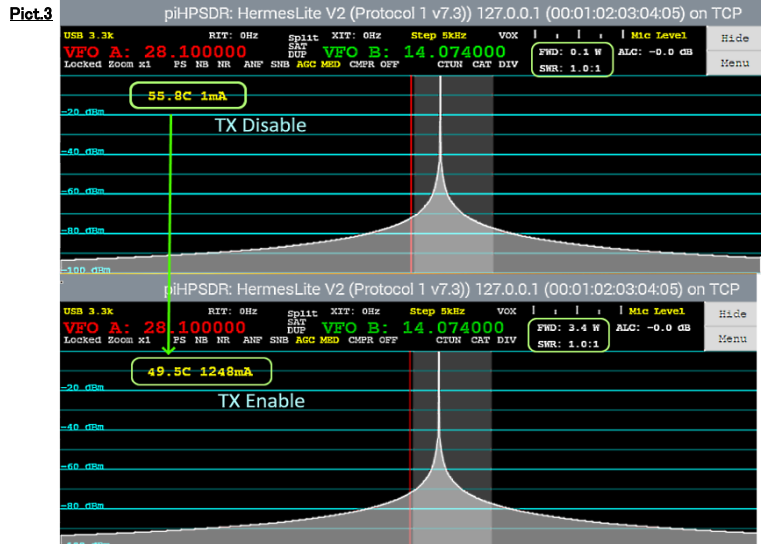
Temp.[°C]	Output[V]	duty[%]	TX enable	Pict.
less 37	0	0	Enable	-
over 37	3.3	50	Enable	1
over 40	3.3	75	Enable	2
over 45	3.3	100	Enable	-
over 55	3.3	100	Disable	3
less 50	3.3	100	Enable	3

TX(Tune) Stop

Temp.[°C]	Output[V]	duty[%]	TX
less 50	3.3	100	off
less 40	3.3	75	off
less 37	3.3	50	off
less 35	0	0	off



* Freq = 2.5MHz / 2^16



The fan_pwm signal to output from FPGA.

~/Hermes-Lite2-master/gateway/rtl/radioberry/juice/ radioberry_core.v , radioberry.v.

(1) radioberry_juice_core.v

```

module radioberry_juice_core(
//RF Frontend
output rffe_ad9866_rst_n,
:
output io_pwm,
:
inout io_sda
);
control #(
.VERSION_MAJOR(VERSION_MAJOR),
.FAN (FAN ),
.CW (CW )
) control_i (
.clk(clk_internal),
:
.fan_pwm(io_pwm), //not used yet; only shutting down pa if temp too high
:
.sda_t(sda_t)
);

```

(2) radioberry.v

```

module radioberry (
//RF Frontend
output rffe_ad9866_rst_n,
:
output io_pwm,
:
inout io_sda
);
radioberry_juice_core #(
.NR (2 ),
:
.io_pwm (io_pwm),
:
.io_sda (io_sda)
);
endmodule

```

~/Hermes-Lite2-master/gateway/variants/radioberry_juice_cl025/radioberry.qsf

(3) radioberry.qsf

```

:
set_global_assignment -name DEVICE 10CL016YE144C8G

set_global_assignment -name VERILOG_MACRO "USE_ALTSYNORAM=1"
set_global_assignment -name PROJECT_OUTPUT_DIRECTORY build
source ../../boards/radioberry-juice/general.tcl
source ../../boards/radioberry-juice/pins.tcl
source ../../boards/radioberry-juice/files.tcl

set_global_assignment -name VERILOG_FILE ./radioberry.v
set_global_assignment -name TOP_LEVEL_ENTITY radioberry
set_global_assignment -name SDC_FILE "../../boards/radioberry-juice/timing.sdc"

set_global_assignment -name LAST_QUARTUS_VERSION "20.1.1 Lite Edition"

set_location_assignment PIN_133 -to io_pwm
set_location_assignment PIN_115 -to io_scl
set_location_assignment PIN_119 -to io_sda

```

I use Radioberry(CL016) board now
Because I broke the CL025 FPGA in applied 15V !
A Hot summer night in Tokyo caused the mistake. :(

2. N2ADR-LPF feature for mcp23018.

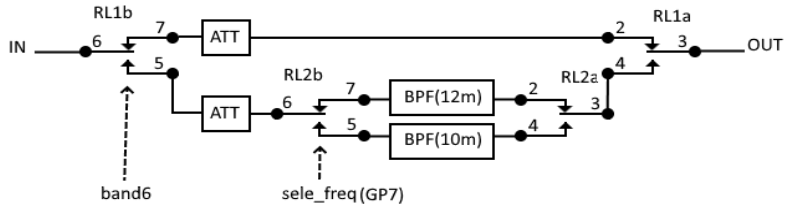
Added "select_freq" signal for BPF switching and initialize IODIRA to Output for mcp23018.

~/Hermes-Lite2-master/gateway/rtl/radioberry/juice/control.v, i2c bus.v

```

1  module control(
2  input  clk,          //10 MHz
  :
99  logic  sele_freq = 1'b0; // 0: (30m|17m|12m), 1: (20m|15m|10m) <-- Added
  :
114 i2c_bus i2c_bus_i (
134 .sele_freq(sele_freq), // Added
  :
488 if (freq[25:16] >= FREQ_25MHZ) begin
    volt_mark = VOLT_10M;
    sele_freq = 1'b1; // 1: 10m <-- Added
end else if (freq[25:16] >= FREQ_23MHZ) begin
    volt_mark = VOLT_12M;
    sele_freq = 1'b0; // 0: 12m
end else if (freq[25:16] >= FREQ_20MHZ) begin
    volt_mark = VOLT_15M;
    sele_freq = 1'b1; // 1: 15m
end else if (freq[25:16] >= FREQ_16MHZ) begin
    volt_mark = VOLT_17M;
    sele_freq = 1'b0; // 0: 17m
end else if (freq[25:16] >= FREQ_12MHZ) begin
    volt_mark = VOLT_20M;
    sele_freq = 1'b0; // 0: 20m
end else if (freq[25:16] >= FREQ_8MHZ) begin
    volt_mark = VOLT_30M;
    sele_freq = 1'b0; // 0: 30m
end else if (freq[25:16] >= FREQ_6MHZ) begin
    volt_mark = VOLT_40M;
    sele_freq = 1'b0; // 0: 40m
end else if (freq[25:16] >= FREQ_4MHZ) begin
    volt_mark = VOLT_60M;
    sele_freq = 1'b0; // 0: 60m
end else if (freq[25:16] >= FREQ_2MHZ) begin
    volt_mark = VOLT_80M;
    sele_freq = 1'b0; // 0: 80m
end else begin
    volt_mark = VOLT_160M;
    sele_freq = 1'b0; // 0: 160m
507 end
518 endmodule

```



BPF switching for TX (to reduce non-harmonic spurious)

~/Hermes-Lite2-master/gateway/rtl/radioberry/juice/i2c bus.v

```

3  module i2c_bus
4  (
5  clk,
  :
24  sele_freq, // added
  :
52  input  sele_freq;
  :
93  logic  flg_fini = 1'b0; // added for mcp23018
  :
198 // Filter select update
199 if (cmd_addr == 6'h00) begin
200   if (flg_fini == 1'b1) begin
201     if ((cmd_data[23:17] != filter_select_reg) | (sele_freq != rx_antenna_reg)) begin
202       // Must send
203       if ("busy") begin
204         filter_select_next = cmd_data[23:17];
205         rx_antenna_next = sele_freq;
206         cmd_next = 'h20; // i2c addr 20h
207         data0_next = 'h12; // GPIOA, mcp23008: 0x0a(OLAT)
208         // Alex rx antenna option passed to GP7 on MCP23008 GP7
209         data1_next = |sele_freq, cmd_data[23:17]];
210         state_next = STATE_FCMDADDR;
211       end else begin
212         cmd_ack_next = 1'b0; // Missed
213       end
214     end
215   end else begin // IODIRA=00 to Output 1st.
216     if ((cmd_data[23:17] != filter_select_reg) | (sele_freq != rx_antenna_reg)) begin
217       if ("busy") begin
218         filter_select_next = 'h48; // band7, band4
219         rx_antenna_next = 'h0;
220         cmd_next = 'h20; // i2c addr 20h
221         data0_next = 'h0; // mcp23018 IODIRA (Output)
222         data1_next = 'h0;
223         state_next = STATE_FCMDADDR;
224       end else begin
225         cmd_ack_next = 1'b0; // Missed
226       end
227     end
228   end
229   flg_fini = 1'b1;
230 end
231 end

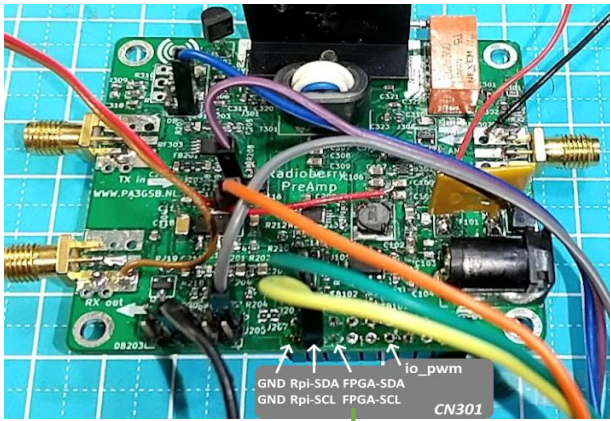
```

Changed a signal GP7 to select the frequency. This can separate the 10m and 12m band for BPF (also 15m/17m, 20m/30m, 40m/60m)

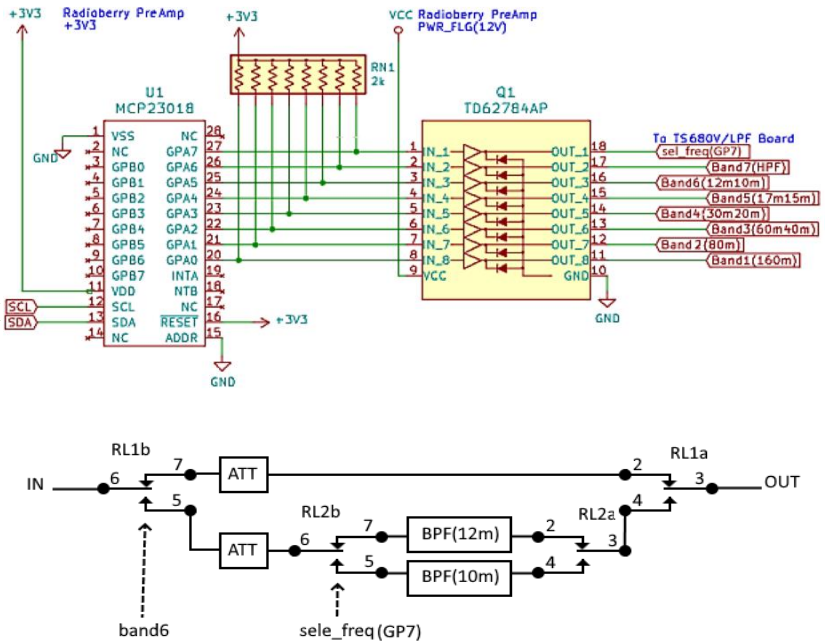
The initial setting of IODIRA=00 is the same as MC23008, but MC23018 was not initialized. Therefore, this initialization has been added.

pi-HPSDR - Open Collector Output															
Band	1	2	3	4	5	6	7	1	2	3	4	5	6	7	Tune (ORed with TX)
GEN	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	1
136kHz	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	2
472kHz	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	3
160	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	4
80	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	5
60	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	6
40	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	7
30	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Full Tune(ms): 2800
20	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	Memory Tune(ms): 550
17	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
15	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
12	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
10	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	

BAND	160m	80m	60m	40m	30m	20m	17m	15m	12m	10m
1	○									
2		○								
3			○	○						
4					○	○				
5							○	○		
6									○	○
7 (HPF)		○	○	○	○	○	○	○	○	○
8 (GP7)								○		○



CN301	
P15	FPGA-SDA
P16	FPGA-SCL
(P10 io_pwm)	

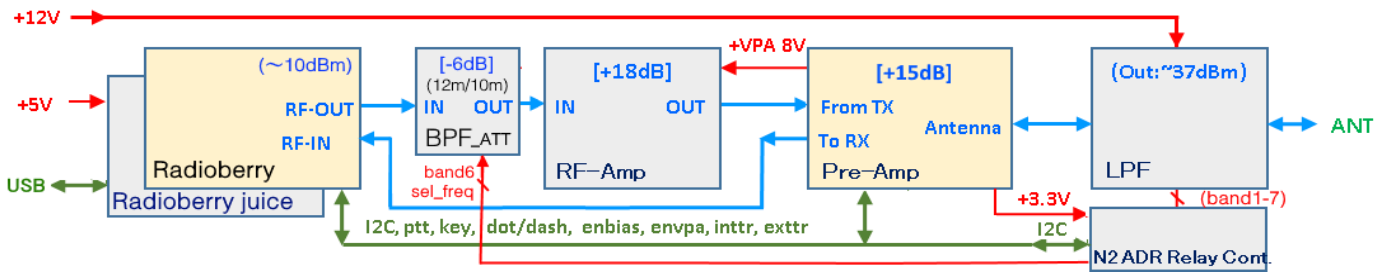


How to Compile HDL for Radioberry-Juice FPGA

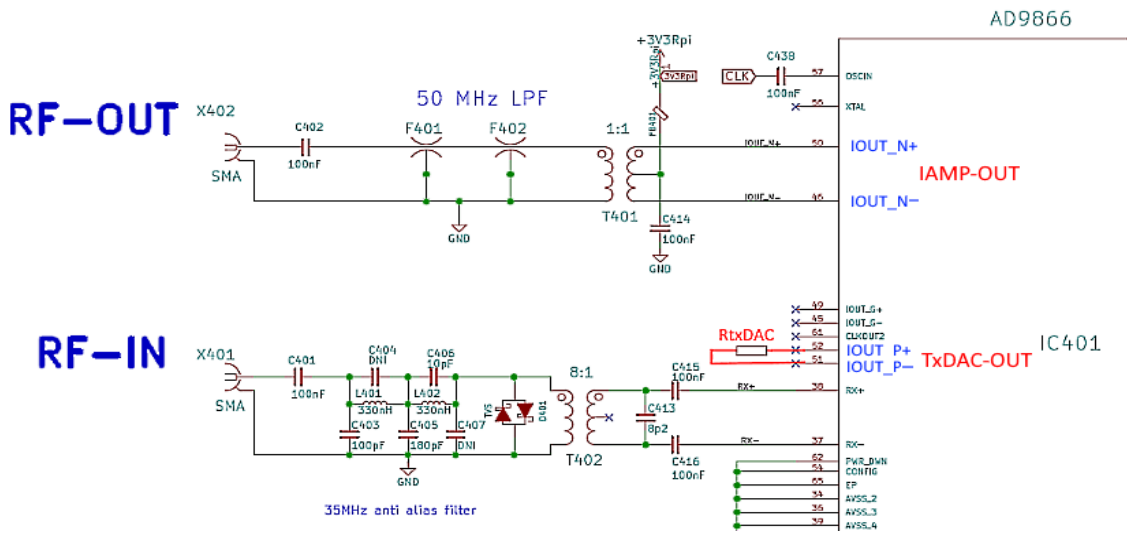
- (1) Install Quartus Prime Lite 20.1 or later into Linux-PC (Ubuntu 18.04)
 Download URL: <https://fpgasoftware.intel.com/20.1/?edition=lite> select Linux
 Individual Files Tab: (or Combined Files Tab)
 • Cyclone 10 LP device support (•Quartus-lite-20.1.0.711-linux.tar)
 • Quartus Prime (includes Nios II EDS)
 Install:
`$./QuartusLiteSetup-20.1.0.711-linux.run`
- (2) Edit .profile and add PATH.
`$ vi ~/.profile`
`export QSYS_ROOTDIR="/home/yado-san/intelFPGA_lite/20.1/quartus/sopc_builder/bin"`
`PATH=$PATH:$HOME/intelFPGA_lite/20.1/quartus/bin <-- Add this line (pse change your directory)`
- (3) Download Hermes-Lite2 (Radioberry) gateway.
`$ git clone https://github.com/softerhardware/Hermes-Lite2.git`
- (4) Compile
`$ cd ~/Hermes-Lite2/gateway/variants/radioberry_juice_cl025`
`$ make`
 → Results here: ~/Hermes-Lite2/gateway/variants/radioberry_juice_cl025/build/radioberry.rbf

TX spurious reduction experiment

Test System



Add "RtxDAC" between AD9866-TxDAC (pin 52: IOUT_P+, pin 51: IOUT_P-) to reduce non-harmonic spurious generated by AD9866-IAMP.



Note:

non-harmonic spurs.

- $f_{spurs} = nF_s \pm mF_{tx}$ $F_s = 76.8\text{MHz} (=38.4\text{MHz} * 2)$
- following 12m and 10m are likely to be a problem.
 - 24.9MHz spur : 29.1MHz ($=2 * 76.8 - 5 * 24.9$)
 - 28.1MHz spur : 13.1MHz ($=2 * 76.8 - 5 * 28.1$)

See attached pdf.

<https://groups.google.com/g/radioberry/c/IIDGZeUKBLc>
07.04 PreAmp measurements and IAMP Spurs and BPF effects.



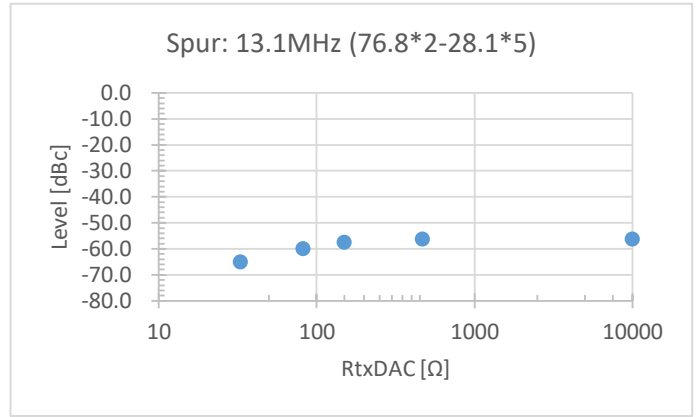
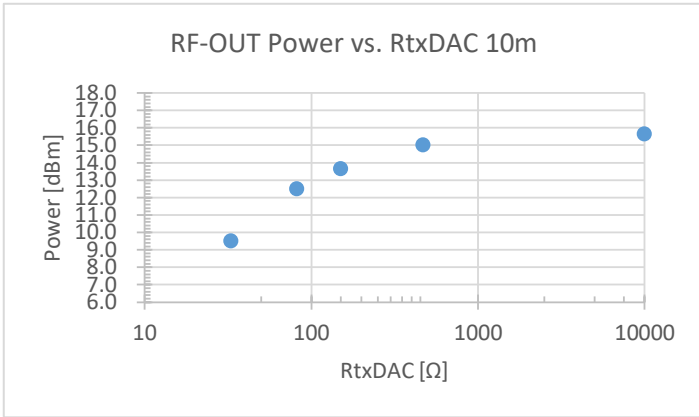
28.1MHz

piHPSDR TX(Tune) Drv.100

RtxDAC [Ω]	10m			
	28.1	10.3	13.1	48.7
10000	15.6	-58.6	-56.4	-59.4
470	15.0	-59.0	-56.2	-58.9
150	13.7	-58.5	-57.5	-59.9
82	12.5	-58.8	-59.9	-59.2
33	9.5	-59.0	-65.1	-59.1

Spurs.

- 10.3MHz = 38.4 - 28.1
- 13.1MHz = 76.8x2 - 28.1x5
- 48.7MHz = 76.8 - 28.1



24.9MHz

piHPSDR TX(Tune) Drv.100

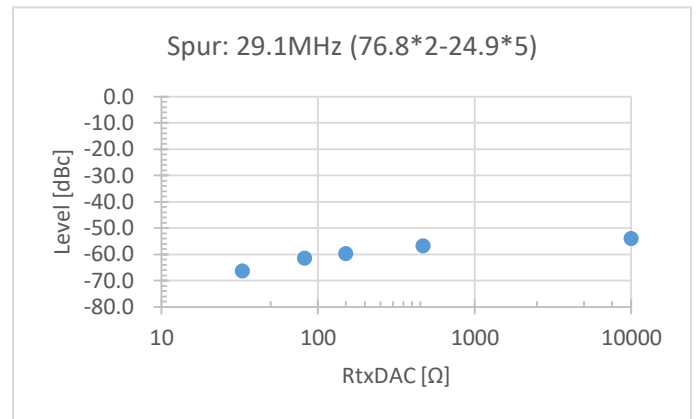
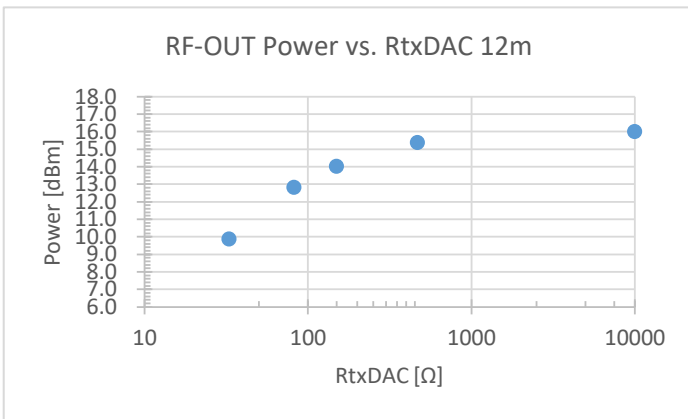
RtxDAC [Ω]	12m			
	24.9	13.5	29.1	51.9
10000	16.0	-60.0	-54.2	-61.2
470	15.4	-60.7	-56.9	-60.6
150	14.0	-60.6	-59.9	-61.8
82	12.8	-59.9	-61.6	-60.8
33	9.8	-60.7	-66.5	-60.5

Spurs.

13.5MHz = 38.4 - 24.9

29.1MHz = 76.8x2 - 24.9x5

51.9MHz = 76.8 - 24.9



21.1MHz

piHPSDR TX(Tune) Drv.100

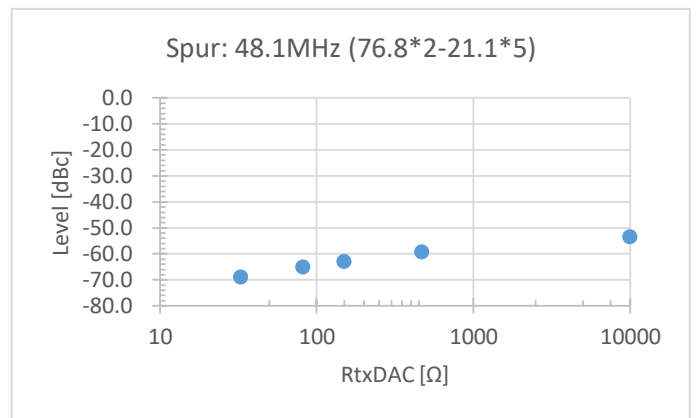
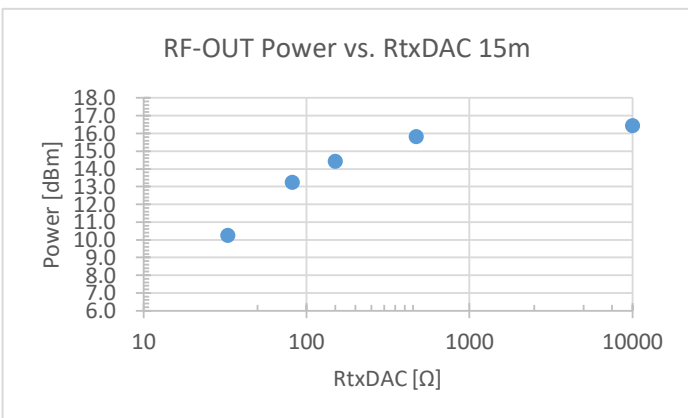
RtxDAC [Ω]	15m			
	21.1	17.3	42.2	48.1
10000	16.5	-61.1	-59.7	-53.4
470	15.8	-60.1	-63.6	-59.2
150	14.4	-61.1	-72.9	-62.9
82	13.3	-61.8	-75.9	-64.9
33	10.3	-61.2	-70.6	-68.9

Spurs.

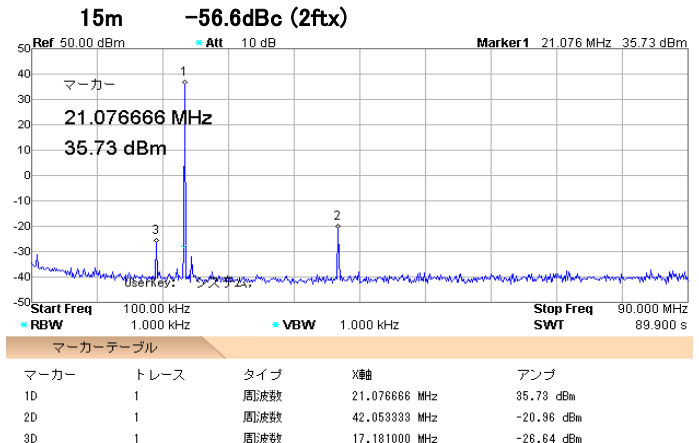
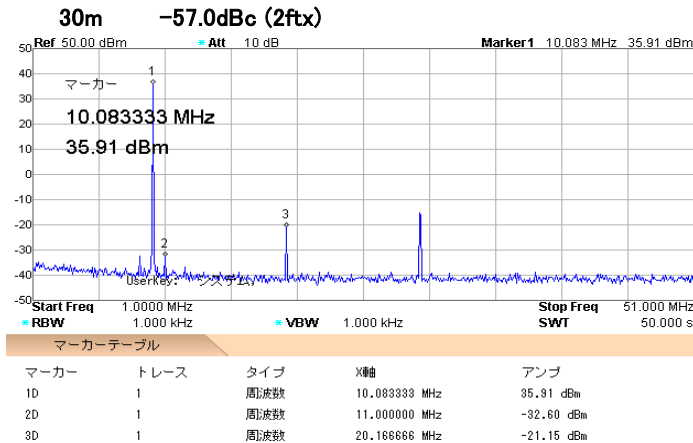
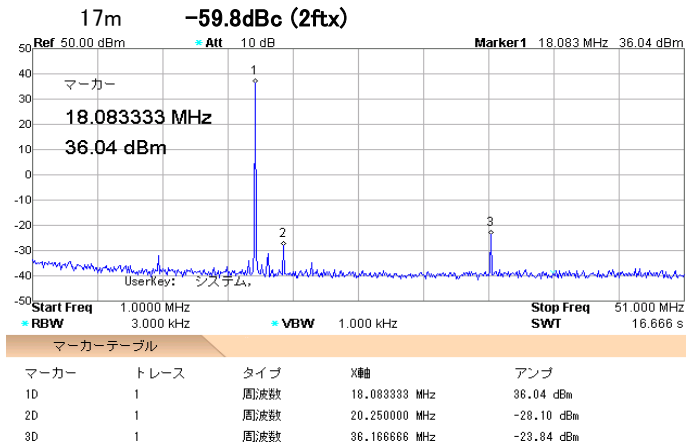
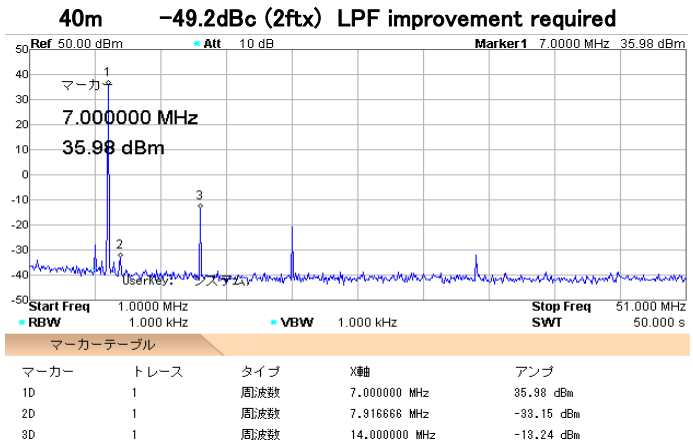
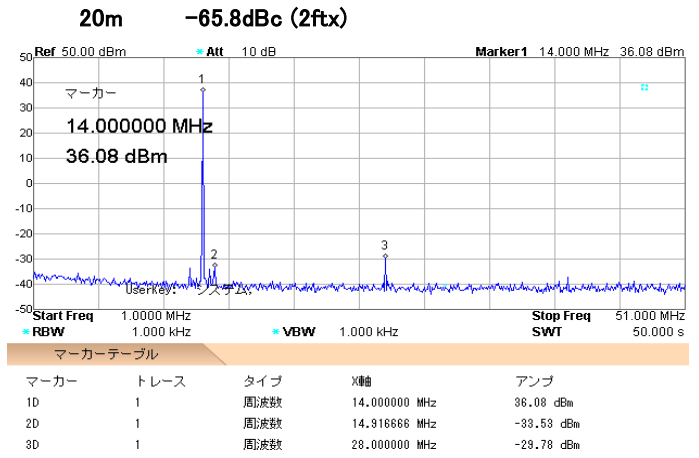
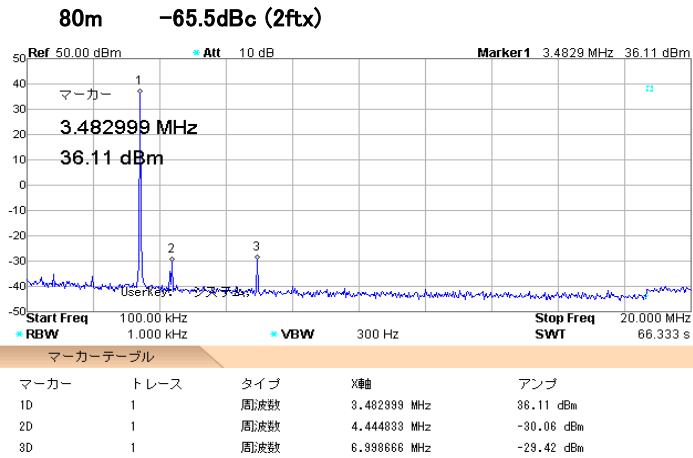
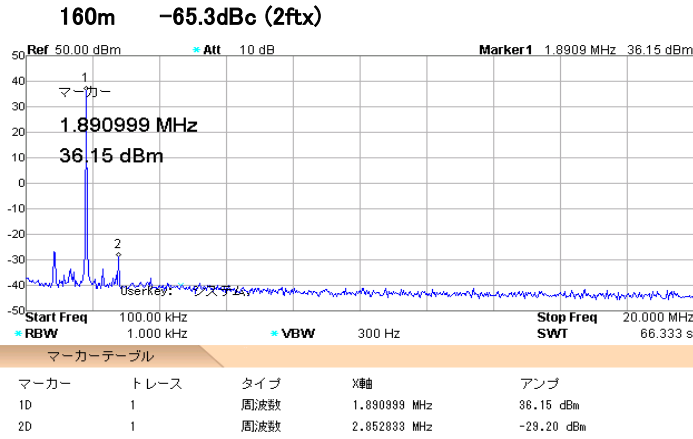
17.3MHz = 38.4 - 21.1

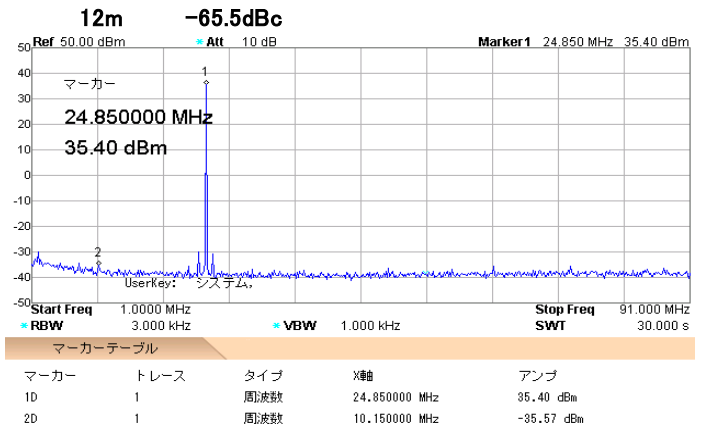
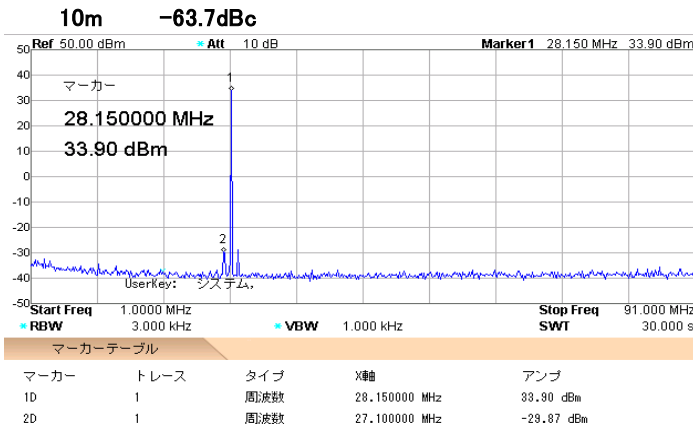
48.1MHz = 76.8 - 21.1

42.2MHz = 21.1 x 2

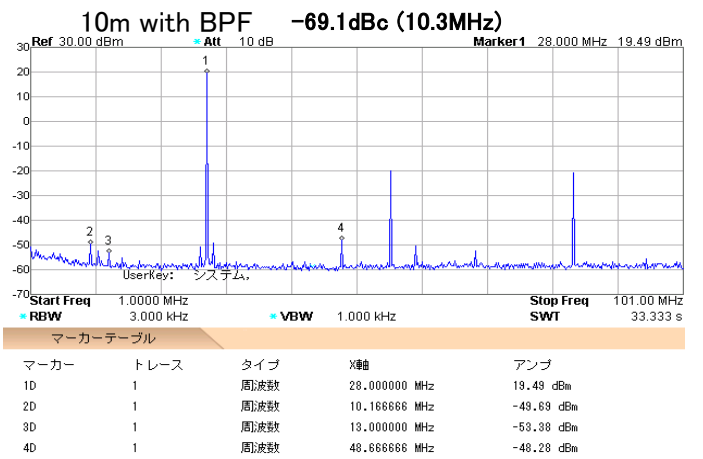
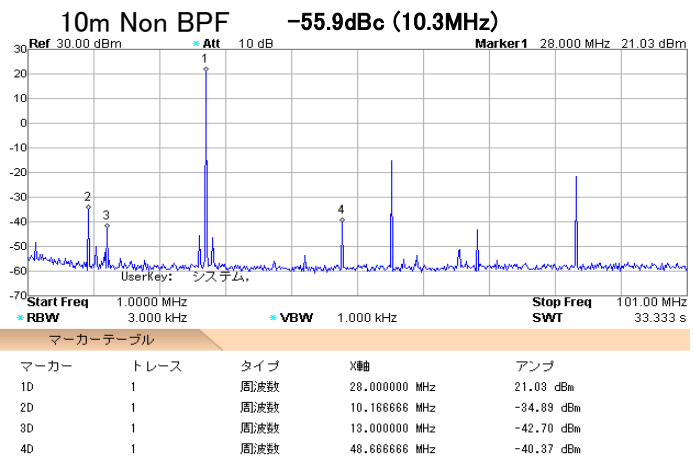
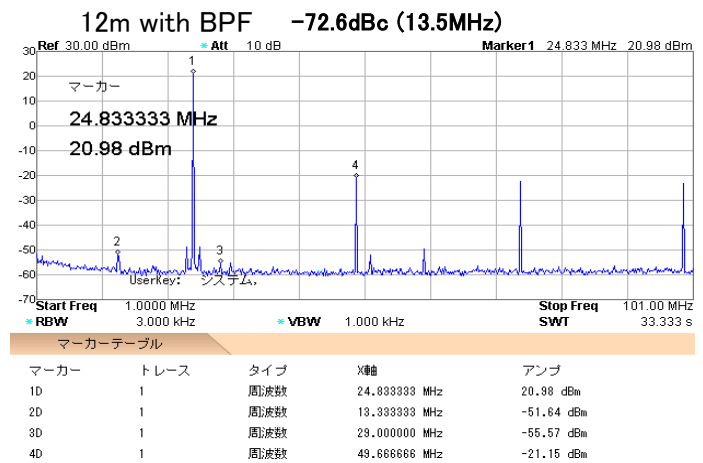
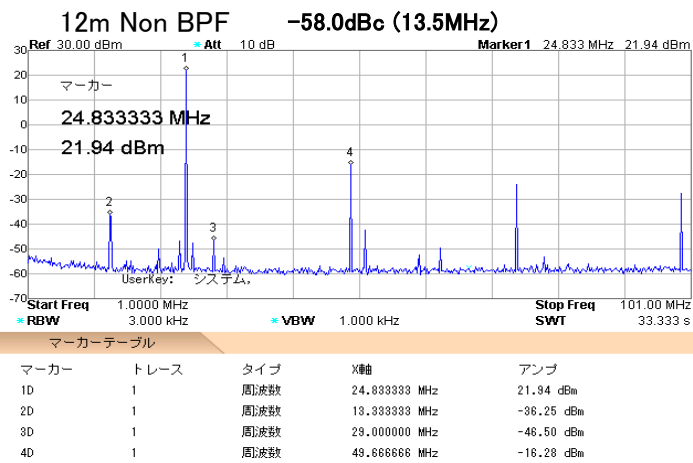


TX Spectrum (RtxDAC=33Ω): PreAmp-LPF OUT





BPF Effect (RtxDAC=33Ω): RF-Amp OUT, without LPF

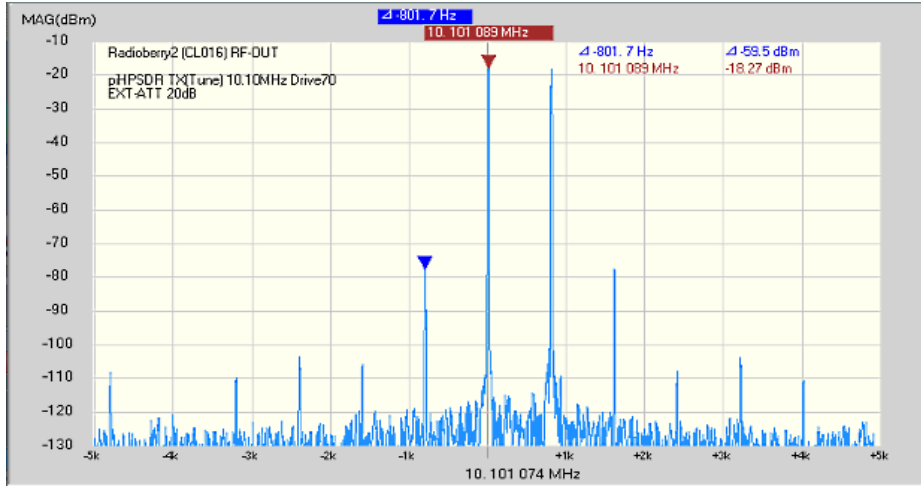


QRPLabs. BPF kit <https://www.qrp-labs.com/images/bpkit/bpf2.pdf>

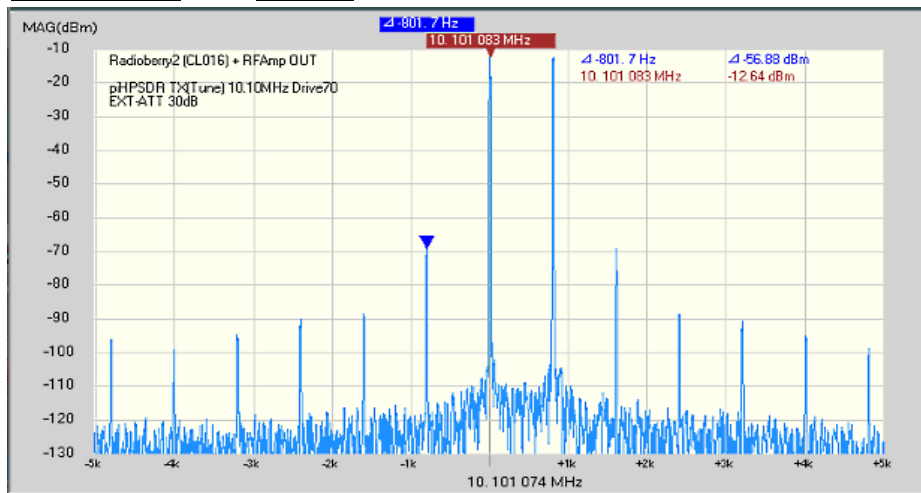
Band	Centre frequency	Bandwidth	Insertion
15m	21.00MHz	1.538MHz	-1.10dB
12m	25.53MHz	2.87MHz	-1.55dB
10m	28.99MHz	3.01MHz	-2.52dB

IMD

RF-OUT 58.5dBc



RF-Amp OUT 56.8dBc



Pre-Amp OUT 24.2dBc

