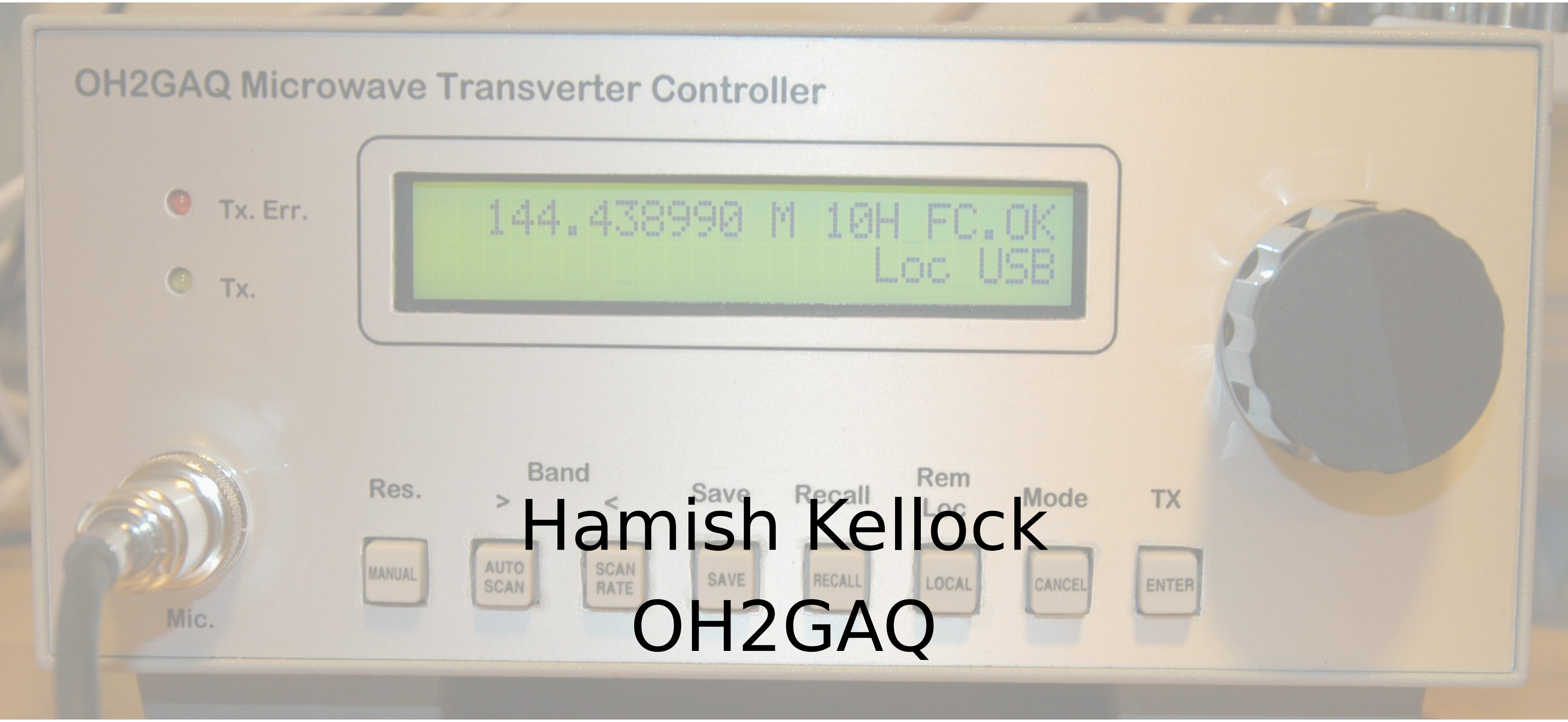
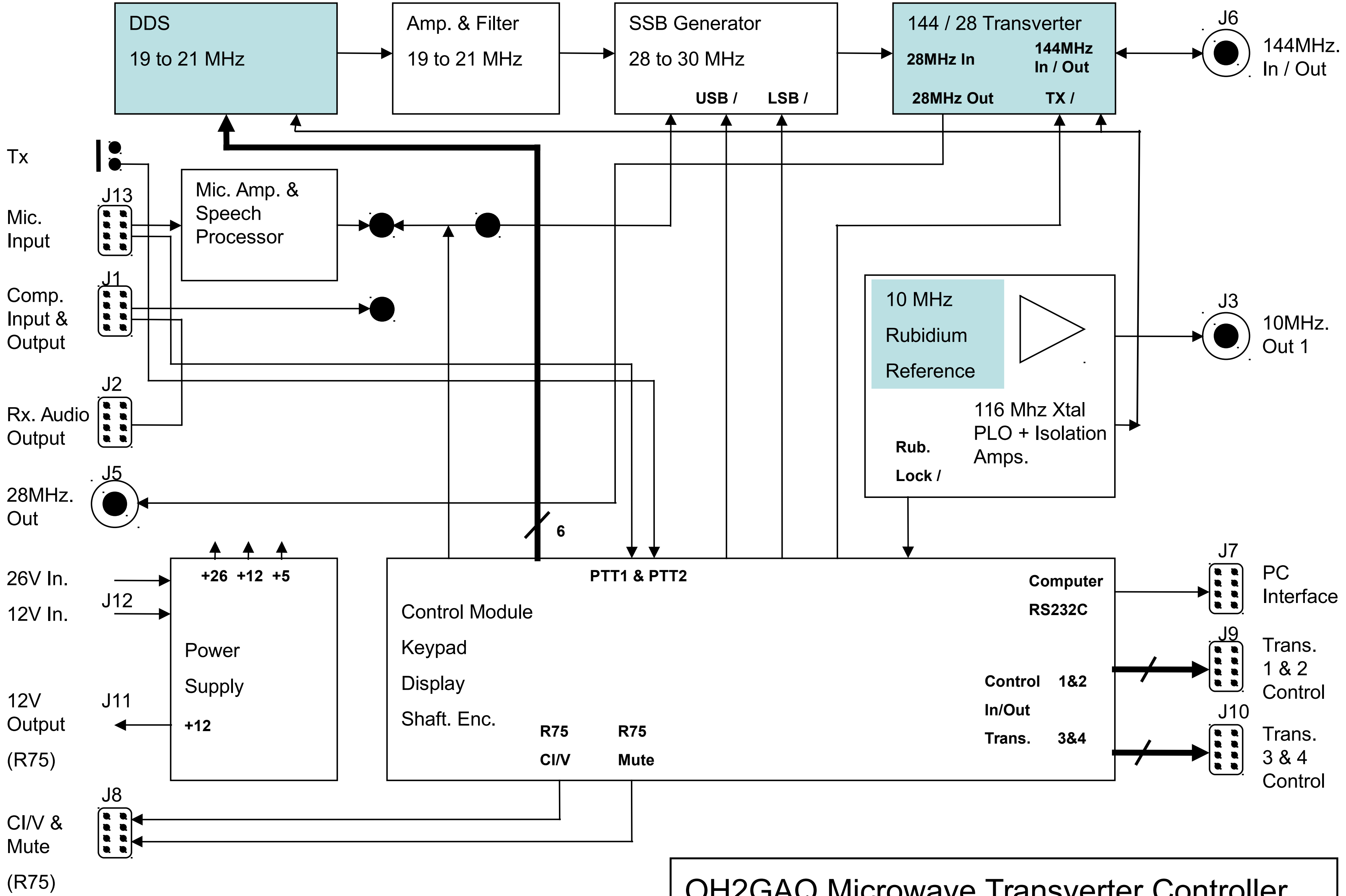
Using SDR Technology in a portable Microwave Transceiver



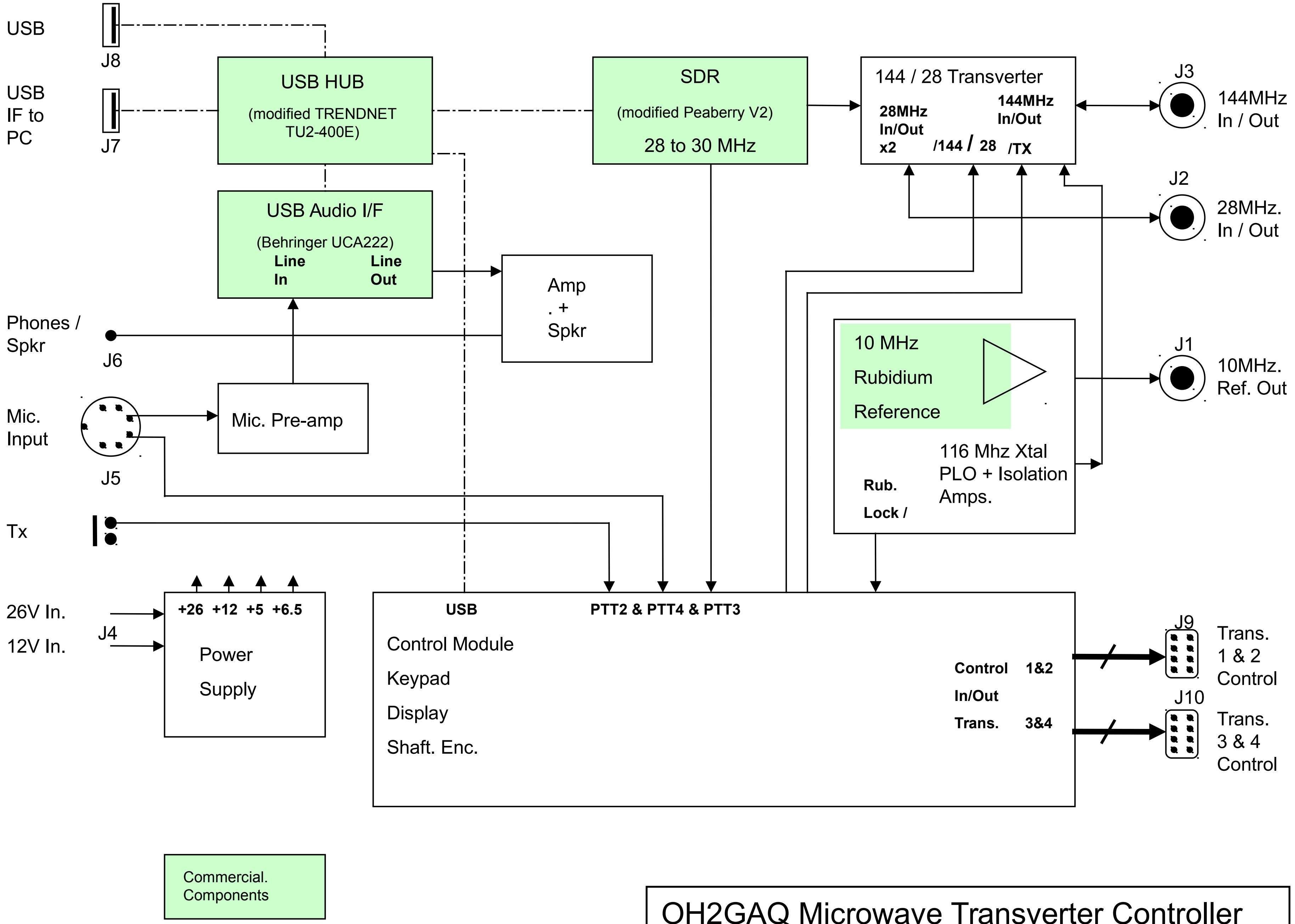
Hamish Kellock
OH2GAQ

Some requirements:

- Transmit and Receive SSB, FM and CW (at least) in the range 144 to 146 MHz. Frequency known to better than 10Hz.
- Provide direct readout of Rx and Tx frequencies to 1 Hz.
- Provide an accurate 10 MHz reference (1 part in 10^9) to remotely located transverters.
- Provide local or remote PTT (Tx./Rx.) switching functionality, with safety return to Rx.mode if Tx. error conditioins are detected (like high reflected power, PA overtemperature etc).
- Can be connected to (at least) 4 different Microwave Transverters at any one time, but also be useable stand-alone.
- Be powered from a 12.6 V battery for portable operation
- (Can be controlled from an external computer).
- Be programmable to accommodate changes in external environment.
- (Should be useable in a portable, mountain-top environment).

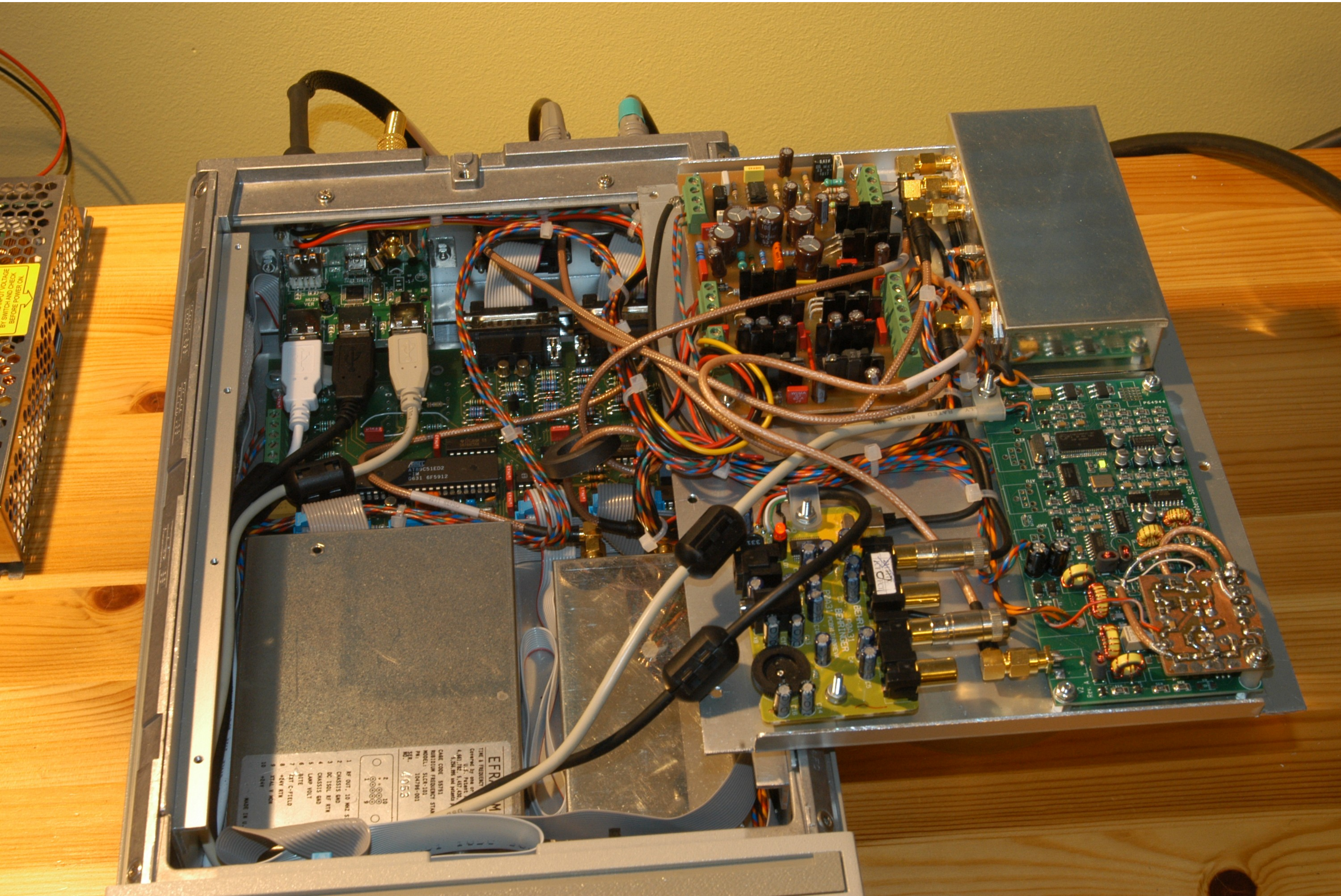


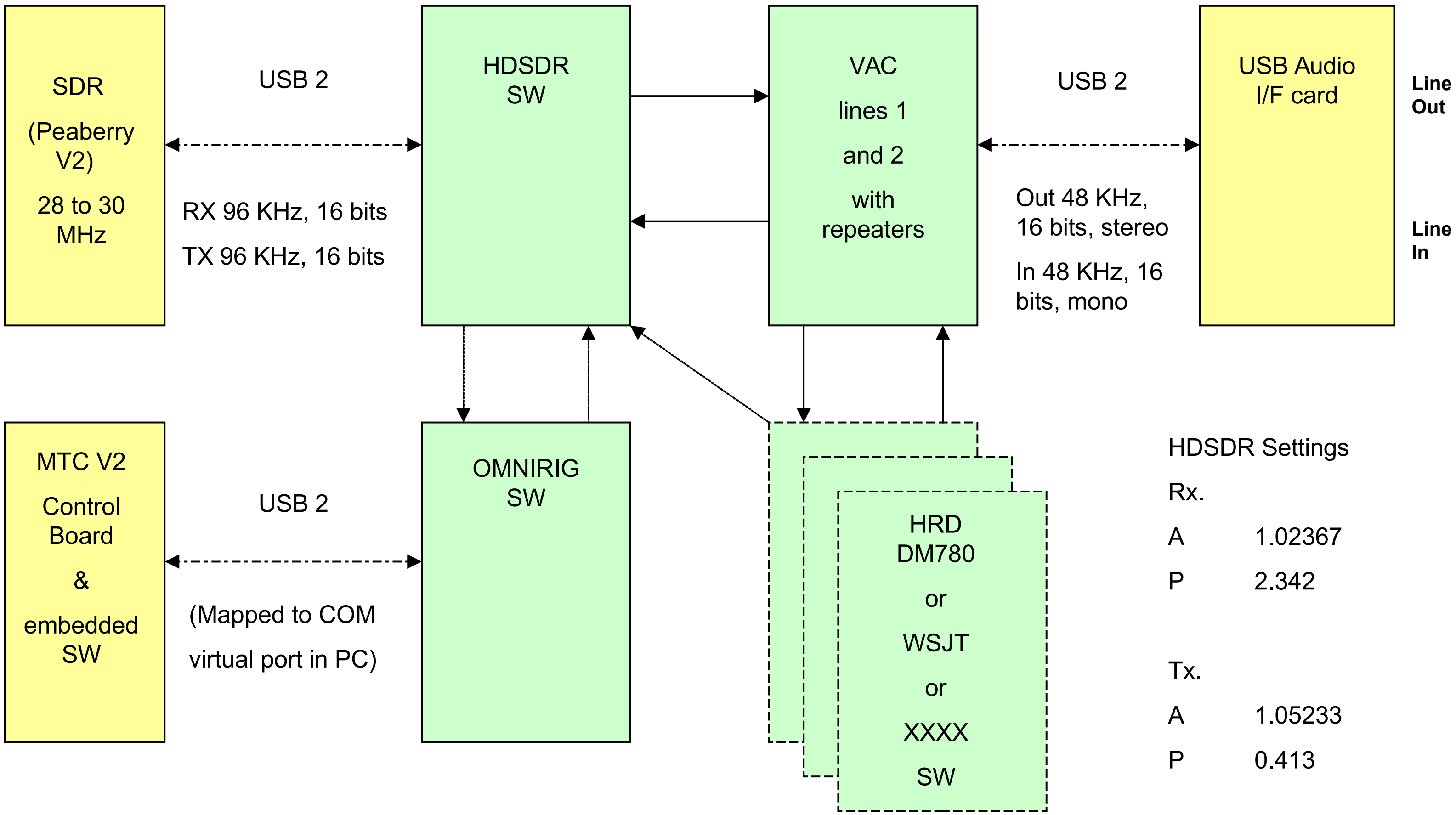
OH2GAQ Microwave Transverter Controller Mk1 Block Diagram



OH2GAQ Microwave Transverter Controller SDR Version Block Diagram v00

Physical Implementation of MTC V2





SDR

(Peaberry V2)

28 to 30 MHz

USB 2

RX 96 KHz, 16 bits

TX 96 KHz, 16 bits

HDSDR SW

VAC

lines 1

and 2

with repeaters

USB 2

Out 48 KHz, 16 bits, stereo

In 48 KHz, 16 bits, mono

USB Audio I/F card

Line Out

Line In

MTC V2

Control Board

&

embedded SW

USB 2

(Mapped to COM

virtual port in PC)

OMNIRIG SW

HRD DM780

or

WSJT

or

XXXX

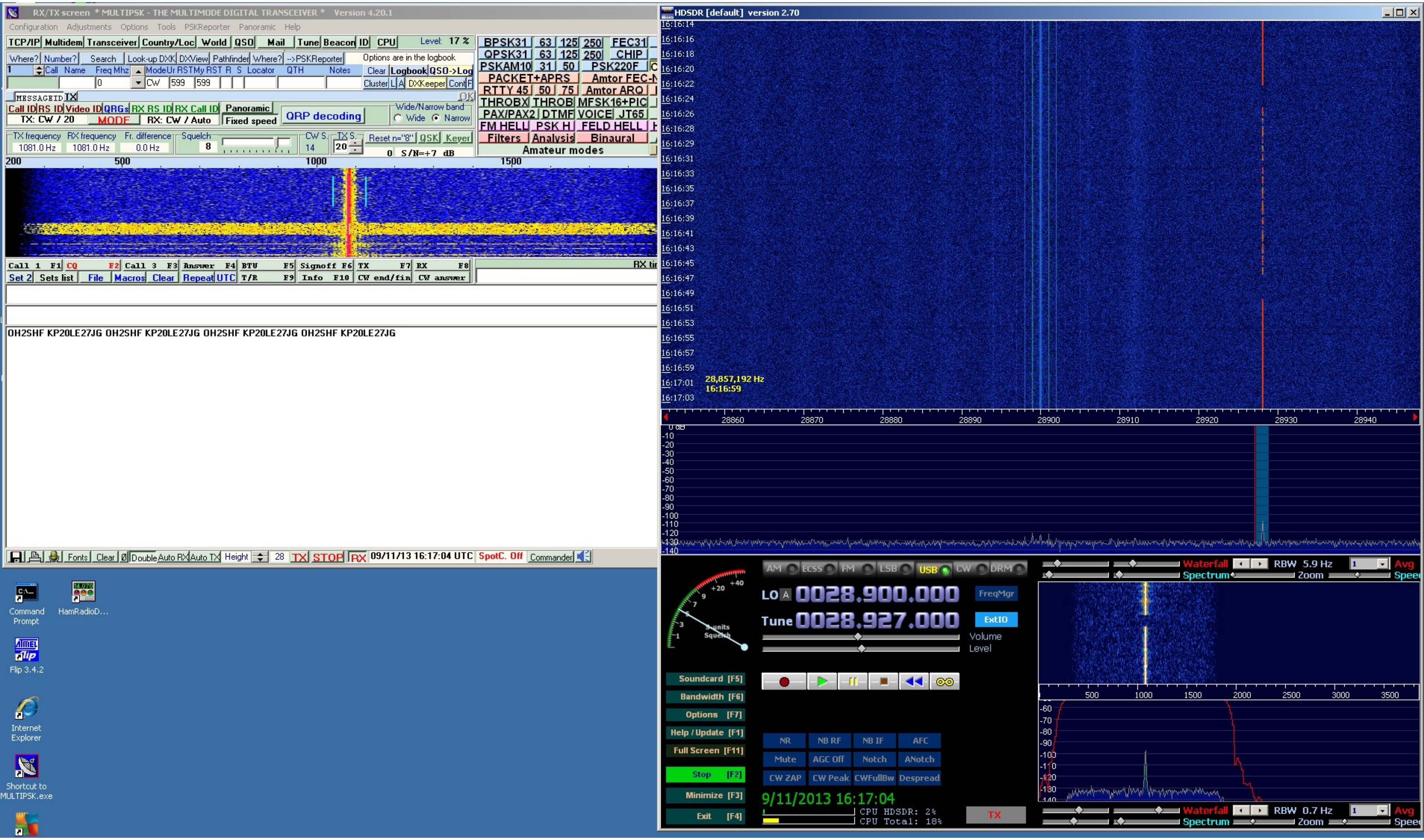
SW

HDSDR Settings

Rx.	
A	1.02367
P	2.342
Tx.	
A	1.05233
P	0.413

OH2GAQ Microwave Transverter Controller SDR Version SW Block Diagram

HDSDR, MultiPSK, 10 GHz Helsinki beacon



RX/TX screen * MULTIPSK - THE MULTIMODE DIGITAL TRANSCEIVER * Version 4.20.1

Level 17 %

TX: CW / 20 MODE RX: CW / Auto

TX frequency 1081.0 Hz RX frequency 1081.0 Hz Fr. difference 0.0 Hz Squelch 8 CW S. 14 TX S. 20 0 S/N=+7 dB

```
OH2SHF KP20LE27JG OH2SHF KP20LE27JG OH2SHF KP20LE27JG OH2SHF KP20LE27JG
```

09/11/13 16:17:04 UTC

HDSDR [default] version 2.70

28,857,192 Hz
16:16:59

LO A 0028.900.000
Tune 0028.927.000

USB

Waterfall RBW 5.9 Hz
Waterfall RBW 0.7 Hz

9/11/2013 16:17:04

CPU HDSDR: 2%
CPU Total: 18%

SDR MTC Notes

The SDR MTC uses a standard USB Hub (see later) plus a USB Sound Interface (Behringer U-Control UCA222 or similar) to provide the physical interface for a microphone and a loudspeaker/phones for receiver use.

The Control Module is fitted with the USB Interface option (which emulates a virtual COM port) and the SW emulates the required key functions of an Icom or other transceiver, which allows HRD or HDSDR alone to interrogate or set the state of the band selection of the MTC, and if required the MTC can be used to select LSB, USB etc. modes of the HRD or HDSDR.

PTT state of the MTC is controlled directly by the Peaberry PTT output line (normally used for keying an external amplifier).

All functionality for transmission and reception is provided by the PC attached via the USB port. At least USB 1.1 (12 mbs speed) is needed in order to provide sufficient speed for the SDR and Sound Card audio channels.

Some minus’s compared to the original MTC:

It needs a computer, and nothing comes in or out without the computer.

There is no easy way to stabilize the 28 MHz carrier frequency as it relies on a Si570 chip, which does have a crystal but may have a tempco. of up to +/- 50 ppm over the range of -40 to +85 celcius for the type used in the Peaberry SDR. Better versions are also available, but at a cost. The original MTC used an unlocked Xtal carrier osc. on 9 MHz, with slightly better stability.

Peaberry SDR has LO only about 35 to 40 db below the peak SSB output on transmit. The original MTC had better than 60 db suppression of LO, however DDS spurs were about 45 db down, dependent on tuning.

Some pluses’s compared to the original MTC:

Smaller, lighter, less interunit cabling, lower power consumption. Original needed an external receiver such as Icom R75. Uses same control module board as the original MTC. SW in the control module is simpler.

Additional transmission modes (original had only USB and LSB), HDSDR has CW, FM, AM, and some additional digital modes.

Less circuitry for audio processing is needed. Sound interface inbuilt in Peaberry, other sound interface card is low-cost.

New SW and functions can be easily (???) added.

PC only needs a single spare USB interface. There is no need for a high-quality sound card for digital modes.

How to get rid of the PC (1) ?

(a work still underway)

- Don’t build anything, just buy an FDM-Duo, etc, etc, stand-alone SDR type transceiver. Why do you want an SDR anyway !

- How do you define a Raspberry Pi V2 or V3 ? Is it a PC ??? (about 60€, 5v 1 to 2 A)

- Does it have enough processor capacity, I/O capacity for this type of application ?

- What about the SW for Linux / Raspbian variant ?

How to get rid of the PC (2) ?

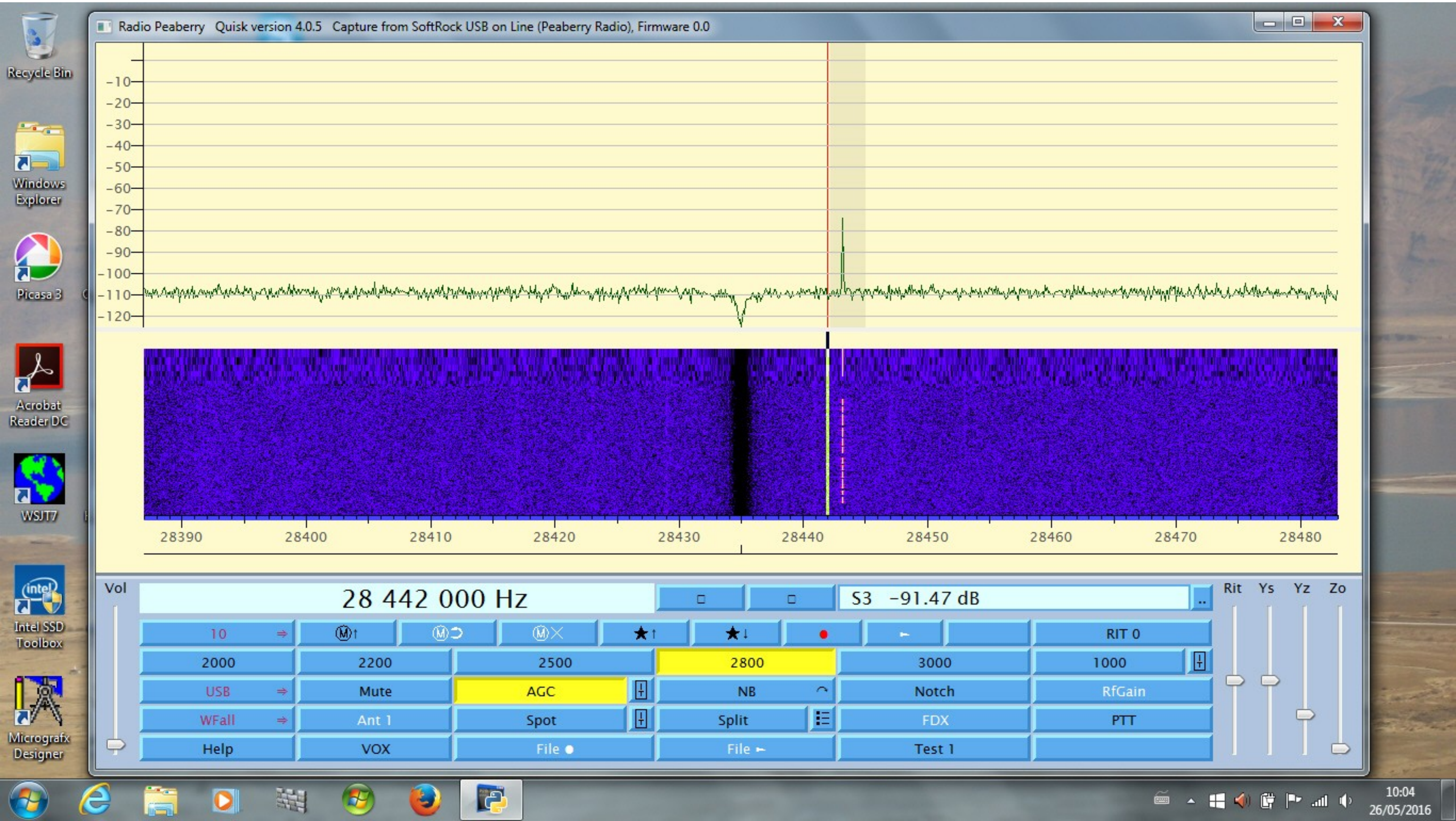
- Jim Ahlstrom, N2ADR and several other hams have produced Quisk which works with many SDR implementations including Softrock (Peaberry is a USB interfaced Softrock with inbuilt USB sound card), HiQSDR (Direct Conversion) hardware, etc. Runs on Linux and Windows. Can also perform other station control functions with added SW modules.

- There is the well-known Linrad from Leif, SM5BSZ. Also runs on Linux and Windows.

- There are various upper level Linux programs, like Fldigi (David Freese Jr. W1HKJ et al) which handle digital modes like PSK xx, Olivia etc in conjunction with Quisk/Linrad acting as the “Basic Transceiver”.

- Also the WSJT XX series of programs are available in Linux versions.

Quisk running on Windows PC, Small-Screen Mode, Waterfall, Inkoo 2M Beacon



Radio Peaberry Quisk version 4.0.5 Capture from SoftRock USB on Line (Peaberry Radio), Firmware 0.0

28 442 000 Hz

S3 −91.47 dB

How to get rid of the PC (3) ?

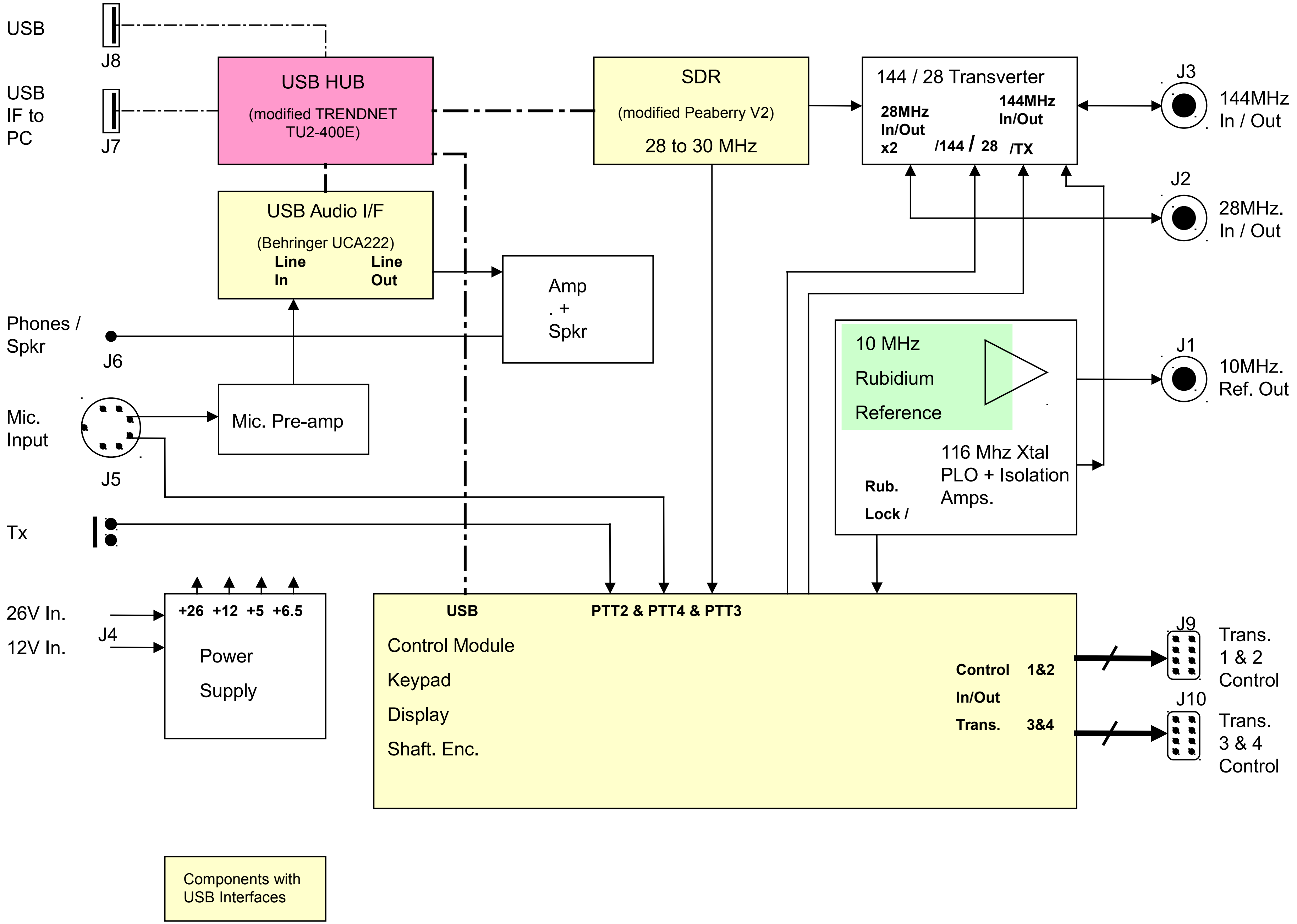
- So take a Raspberry Pi 2 or 3, load it with Raspbian, install Quisk as directed and you should have a working (non-PC ??) SDR application.

- Initially that’s what I did. As long as only the Rx. Audio Paths were defined, it more or less worked.
- As soon as the Tx. Paths were defined, Rx Audio “broke up” and frequency display sometimes/always distorted.
- Processor was not overloaded, according to load graphs.
- Periodic “Clicks” if Ethernet i/f was “up”.

- USB performance issues were found (as expected from comments on the web, many by audio enthusiasts). Lost data found in logs.
- Quisk has four audio streams, at least 3 running at the same time (In Rx. Mode we have Rx. I and Q data in, Audio out to LS/Phones, Mic. Audio in for VOX processing)

Interfacing to R Pi.

- USB is available, 4 ports.
- Also piggy-back cards, including audio A/D and D/A converters, not using USB.
- R Pi USB interface is not the same as that found on “normal PC” motherboards, but rather a reduced interface as used on smartphones. The extra functionality is provided by special driver SW. Interrupt timing issues.
- The single USB interface in the processor is expanded in the Ethernet chip (internal USB hub), to give the 4 external USB’s, but is also used by the wired Ethernet interface.


USB
J8
USB
IF to
PC
J7
USB HUB
(modified TRENDNET
TU2-400E)
SDR
(modified Peaberry V2)
28 to 30 MHz
144 / 28 Transverter
28MHz
In/Out
x2
144MHz
In/Out
/144 / 28
/TX
J3
144MHz
In / Out
J2
28MHz.
In / Out
USB Audio I/F
(Behringer UCA222)
Line
In
Line
Out
Amp
. +
Spkr
Phones /
Spkr
J6
10 MHz
Rubidium
Reference
116 Mhz Xtal
PLO + Isolation
Amps.
Rub.
Lock /
J1
10MHz.
Ref. Out
Mic.
Input
J5
Mic. Pre-amp
Tx
26V In.
12V In.
J4
+26 +12 +5 +6.5
Power
Supply
USB
PTT2 & PTT4 & PTT3
Control Module
Keypad
Display
Shaft. Enc.
Control 1&2
In/Out
Trans. 3&4
J9
Trans.
1 & 2
Control
J10
Trans.
3 & 4
Control
Components with
USB Interfaces

USB Hubs

- USB Hubs are generally poorly specified, at least to the consumer.
- If the USB “type” (USB 1, 1.1, 2) is not the same, then a “Transaction Translator” (TT) function is provided.
- The original USB hub used in the MTC (Trendnet TU2-400E) has only one TT, shared by all 4 downstream interfaces USB interfaces. The 3 interfaces in use are a combination of USB 1.1 and 2.
- The USB hub in the R Pi is a multi-TT hub, where all interfaces have their own TT.
- R Pi does not work well if single-TT hubs are used externally for a number of USB interfaces !

A Solution

- Use the internal R Pi USB hub – tested and works OK so far as is known.

- Use an external multi-TT USB hub. Cirrus at least makes the chipset and there are hubs available that use their chipset (in US at least). Can be bought through e-Bay. That way a single USB cable can be used between the R Pi and the SDR hw. Solution tried and working with a UH-BSC4-US 4-port multi-TT hub from Aya Group.

- R Pi has an optional 7 inch touch screen. So no need for Keyboard and/or mouse after configuring the system.

Continuing Story

- Touchscreen seems to work pretty well for most functions including small frequency tuning excursions (within the currently-displayed frequency span)
- Problen with the authors fingers (and/or using the touch screen for large frequency excursions). When interfaced to the MTC panel, then this is not needed on the screen. Can be solved by using a mouse also.
- Running upper level software not yet investigated by the author. Arnaud, F6BZG has been able to run Fldigi with some success in R Pi 2, now playing with R Pi 3.
- If you want Ethernet connection to the R Pi whilst using it for USB-interfaced SDR's, best to choose R Pi 3, which has a WLAN interface and does not load the USB subsystem.
- .