Microwave Transverter Controller Updated

By Hamish Kellock, OH2GAQ

(A slightly expanded version of an article published in Finnish Radio Amateur Society journal 2/2024)

Overview

Amateurs operating in the microwave (Lets say 1 GHz and above) portion of the spectrum often require a tuneable IF (operating at 144 MHz, 432 MHz or maybe higher) and several other items like a very stable frequency reference, various software packages to pass raw received signals onwards for processing and decoding. Often the basic transceiver functions are provided by a "conventional" transceiver or SDR. Sometimes the tuneable IF is not needed with an SDR operating at the actual microwave band being used. Particularly for higher microwave bands (such as 24 GHz, 48 GHz, 76 GHz 121 GHz and higher) a tuneable IF of some sort is often still employed.

In August, 2013 in an article published in QEX about my earlier Microwave Transverter Controller (the abbreviation MTC will be used after this), a description of an integrated set of hardware functions for operating a microwave station was described, but this also required an external receiver to complete the functionality. Over the intervening years technology has advanced, and plenty of new requirements have arisen. One of the major new requirements is that of full duplex operation, not on the same "on air" frequency, but on separate uplink and downlink frequencies for geostationary satellite operation. A major reduction in size and cost of SDR transceiver technology has taken place, and lots of incredible SDR related SW has been published, some of it open source. Small, lo-cost hi-computational power single board computers and colour touch screens are available having relatively low power consumption and cost, allowing the replacement of laptop or tablet PC's with integrated computers and screens, with not too much mechanical complexity.

Figure 1 shows the results of an updated design for the MTC. The whole unit is contained in an enclosure measuring 300mm wide by 130mm high by 495mm deep. The touch screen is a "7inch" diagonal type, and takes up about half of the real-estate of the front panel. The screenshot shows most of the Narrow Band segment of the QO-100 transponder spectrum on a typical evening. The signals at 2400.000 and 2400.250 are two of the "pseudo-beacons" for the transponder, the third one is above the bandwidth displayed here.

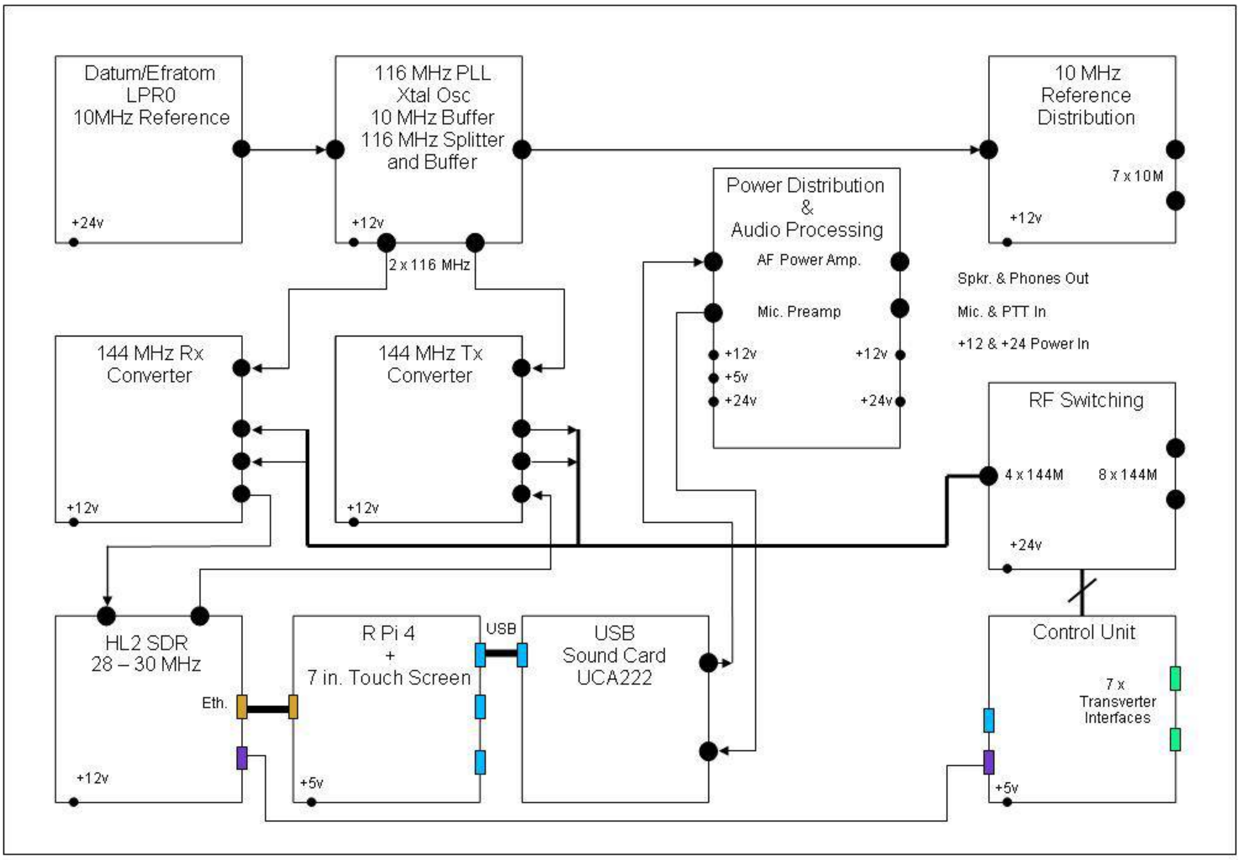

Figure 2. The block diagram of the MTC with all the major functional blocks and external interface connectors. The main external interfaces are collected to the right side of the diagram. The MTC requires only the addition of a +12.6 v and + 24 v power supply, and of course the transverters.

The basic frequency reference is an Rb disciplined oscillator at 10 MHz, which is distributed internally to lock the 116 MHz LO and 7 x 10 MHz outputs for up to 7 transverters. Some care is needed in selecting the Rb. source, the comments in reference 9 at the end of this article give some pointers about the issues, as some of the low cost Rb. sources have rather poor performance in the short term. The 144 MHz Up and Down converters connect directly to the 28 MHz input/low power output of the Hermeslite 2 SDR. The SDR is connected directly by Ethernet to the Raspberry Pi 4 SBC, although a small Ethernet switch can be interposed between them, allowing the connection to an external network if needed. The RF switching block handles the selection of 1 of 6 half-duplex and 1 full duplex transverters.

One of the design goals was to re-use standard HW and SW without requiring special functionality in the SW to support this application. A small microcontroller (Arduino Micro) was used to provide a small amount of extra functionality involving the translation of band selection data and the real-time status of key controller and transverter parameters. It is mounted on the Control board, which also handles the interface to each transverter, and some internal functions. The SDR SW applications used were either Quisk, written by Jim, N2ADR and others or piHPSDR which is currently maintained by Christoph, DL1YCF. The SDR used was the HL2 (Hermes Lite 2) designed by Steve, KF7O and others. It is really because of the availability of these three excellent items that implementing this version of the controller was so easily done.

The MTC provides 6 half-duplex interfaces for microwave transverters (IF in/out, 10 MHz reference, control and status signals) and 1 full-duplex interface (IF in, IF out, 10 MHz reference, control and status signals). The MTC also provides audio out to Speaker/Headphones, and a microphone input pre-amp. There is also an external USB interface available on the back panel, an HDMI video interface and the optional 1G Ethernet interface.

Details of some blocks.

In the following sections some details are given of the implementation of the various functional blocks. Schematic diagrams (.pdf) and Gerbers in "standard format" for producing the printed circuit boards are available for any interested reader. Some of the mechanical/enclosure solutions used by the author are based on the enclosures available in the authors' junkbox.

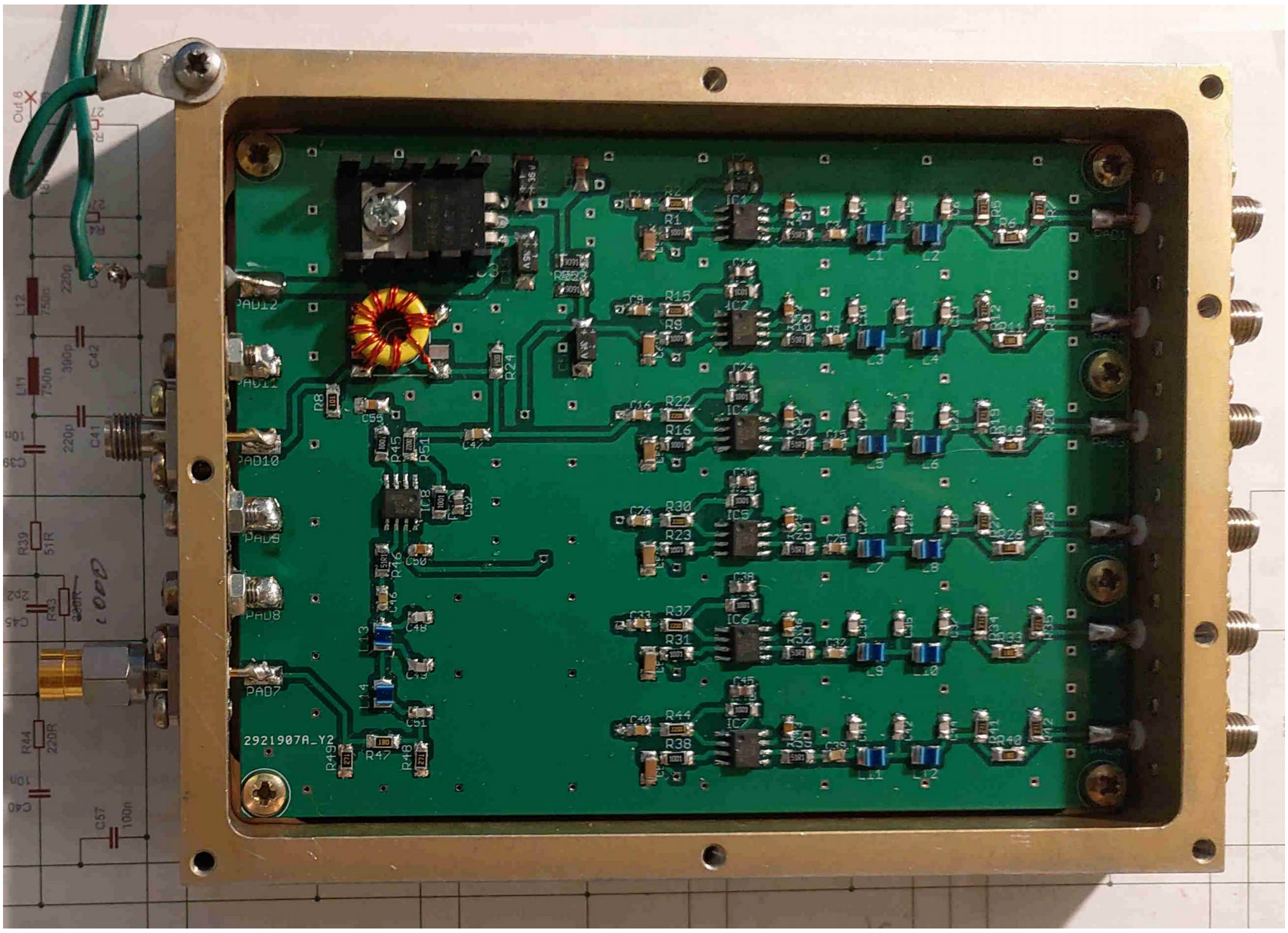

Figure 3. The 7-channel reference distribution amplifier. This version of the amp. uses high-speed op amps, which results in a significant lower power consumption than a typical MMIC implementation (about 90 mA total compared to 350 mA for a typical MMIC implementation).

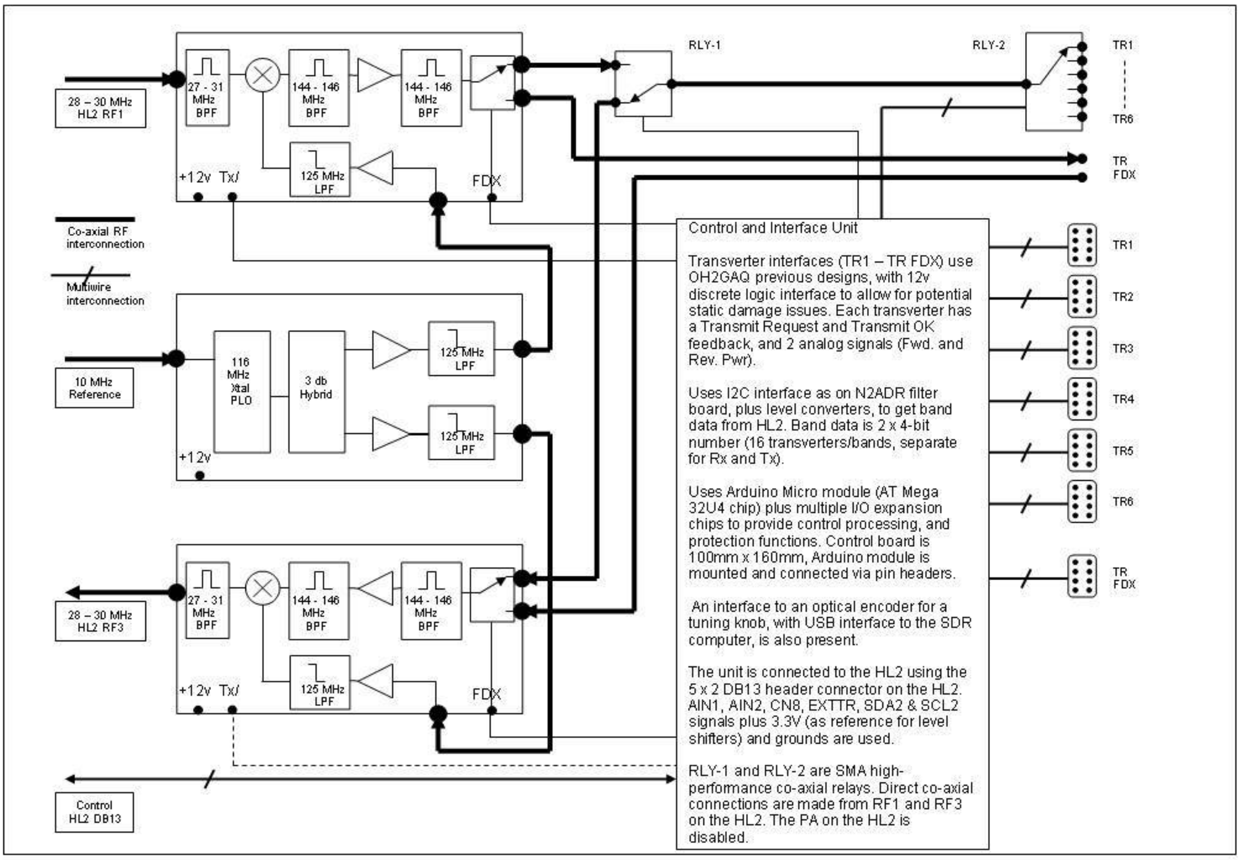

Figure 4. A more detailed block diagram of the 144/28 downconverter, 28/144 upconverter, 116 MHz LO, RF switching, showing also the main functions in the controller card.

The 144/28 and 28/144 converters are based on a common pcb design, so that by alternative equipping of a few minor parts the item can be built as either an upconverter or a downconverter. As the HL2 SDR main board does not contain bandpass filters at 28 to 31 MHz, this filter function is also incorporated into the converter. Also a small changeover relay is used to select between usage as a half-duplex function or full duplex function in the MTC. The PCB's are soldered into standard 111 mm x 55 mm x 30 mm tinplate enclosures.

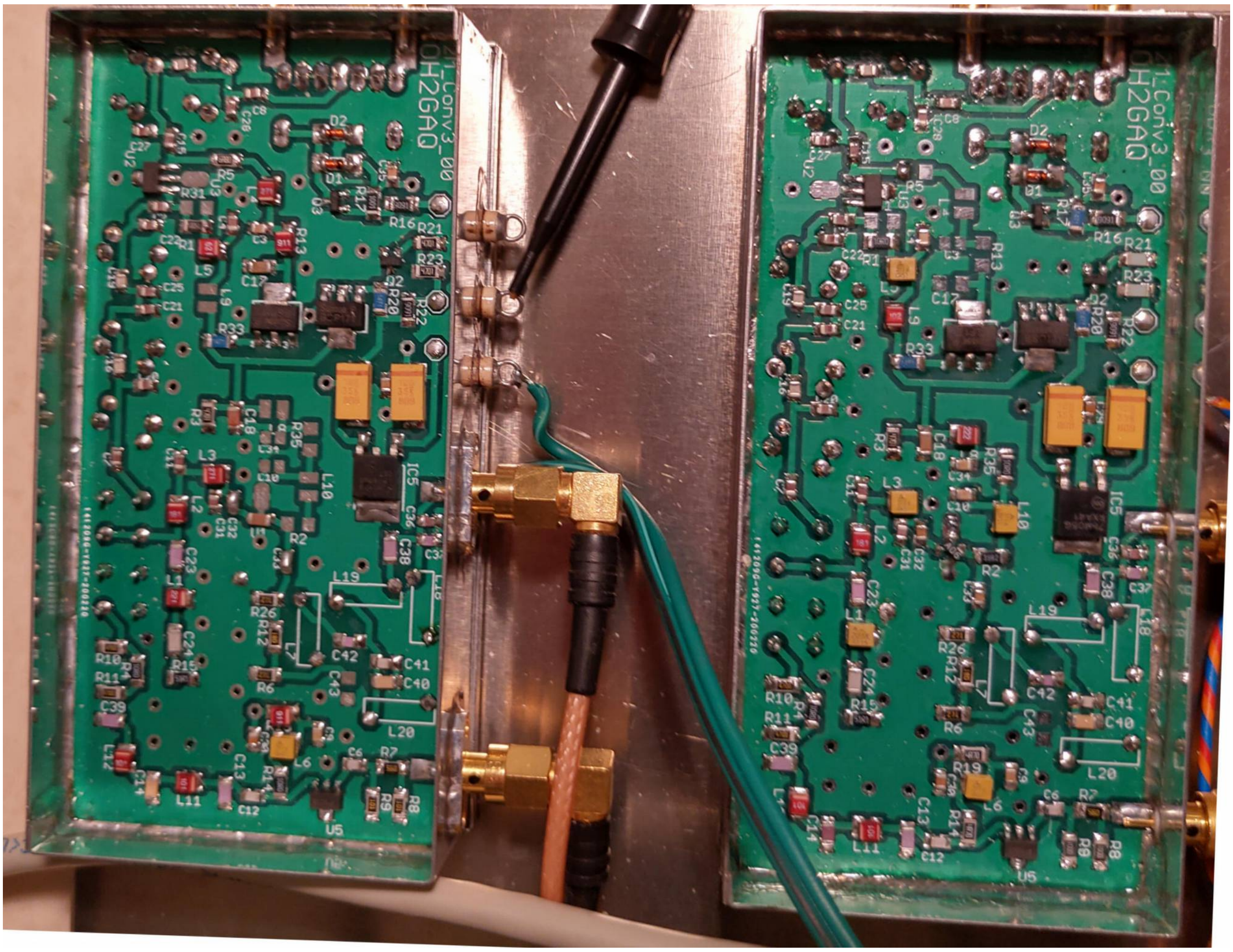

Figure 5. Upconverter (Tx.), left and Downconverter (Rx.), right during prototyping. Changeover relay is in the top left area of the pcb.

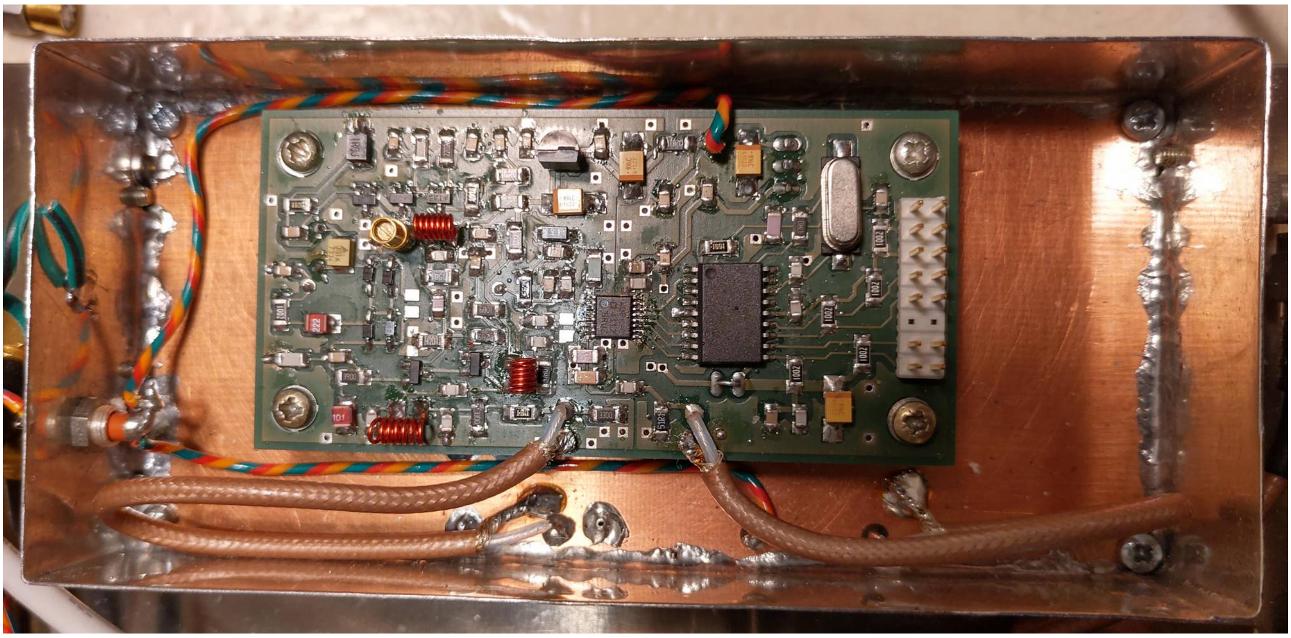

Figure 6. 116 MHz Xtal PLL oscillator. The lower pcb (soldered into the tinplate box) contains the 10 MHz reference buffer, 10 MHz squarer and 116 MHz distribution amplifiers.

The 116 MHz Xtal PLL (the same circuit with slightly different component values is used in all my Xtal PLL applications in various transverters) uses the stable and lo-noise oscillator circuit developed by John Hazell, G8ACE, which seems to have a similar philosophy to the work of John Stephensen, KD6OZH. To phase lock the Xtal oscillator, a simple PLL using the ADF4112 has been added, and a PIC 16F84A has been used to load the ADF4112 parameters at power up. The PCB is shown in Figure 6.The crystal is mounted on the backside of the PCB. The small HC-49 crystal seen here is for the PIC16F84A microcontroller (the largest chip on the board). The ADF4112 is the small chip almost in the middle of the board. Note the “cut” in the groundplane which extends under the ADF4112 chip. This separates the digital and analog ground planes on the component side of the pcb.

The PCB has a solid groundplane on the backside, the crystal and one power regulator are mounted there. There is a pin header on the right side of the board, which is used to program the PIC and later during operation to output the state of the PLL (locked/unlocked). The SW for the PIC is developed using Microchips' MPLAB IDE, where it is also possible to simulate the SW operation and debug it, including the state of the input pins.

This 116 MHz PLL Xtal Osc is mounted on a second pcb, soldered into the tinplate box. The second pcb contains a CMOS squarer for the PLL chip (which needs a square wave reference signal), buffer amp for the 10 MHz reference and a pair of simple resistive attenuators dividing the 116 MHz, which is then amplified with a couple of HP MSA2111 MMIC's to provide extra isolation between the Rx. and Tx. LO's. The whole system is shielded in a small tinplate box.

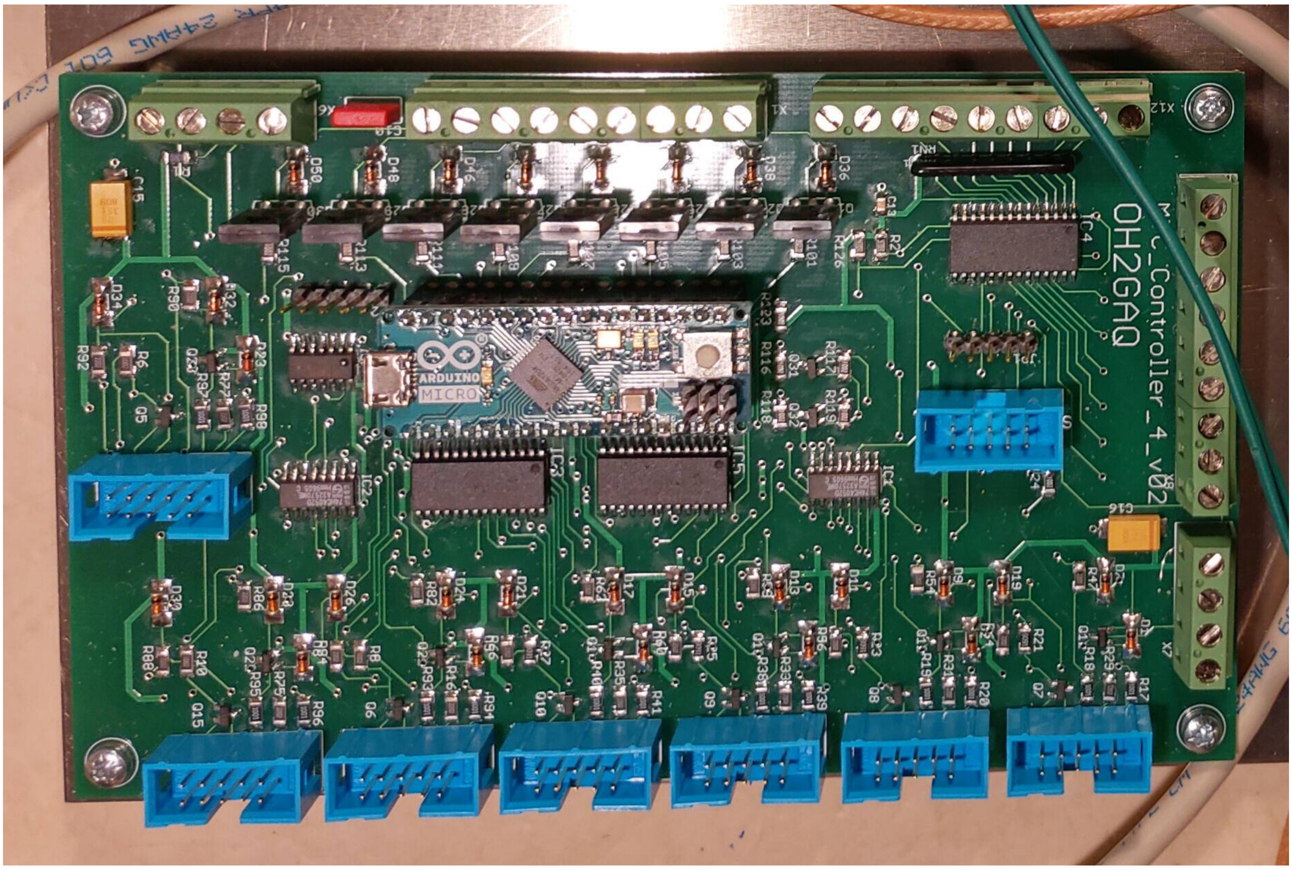

Figure 7. Controller board. Arduino micro seen in the approximate center of the board, power transistors with inductive load protection in the top for driving relays, and to the left and along the bottom are the interface connectors to the 7 transverters. The blue connector to the right center goes to the HL2 SDR.

The control signals to and from each transverter are:

To transverter: TX command.
From transverter: TX OK response.

The TX command is generated by the controller based on the band selected, and the PTT/Remote TX request. Following application of the TX command, the controller waits for the transverter to signal TX OK. If this is not received within the specified time interval (set to 250 ms by default), the TX command is de-activated, and the TX fault indicator is activated. Similarly, if a TX fault condition occurs in a transverter during a commanded transmission, the same happens. In the authors transverters, all the “fast” protection is self-contained in the transverter itself, to ensure that any expensive amplifier transistors are protected, e.g. from high VSWR, or other monitored conditions, and that the protection is not dependent on any remote signals.

The analog monitoring signals from each transverter are:

From transverter: Forward power (analog voltage)
From transverter: Reflected power (analog voltage)

The Controller board has the same interface as an earlier design to the microwave transverters, which is implemented with discrete components to facilitate easy maintenance should some interface circuits be damaged. Several signals from the HL2 come to the controller board (Band selection via I2C interface, Tx mode). The controller board also has outputs to control the RF relays, the 144 MHz up and down converters, and to read in PLL and Rb. oscillator status as well as the optical encoder data and PTT switch state from the microphone or front panel.

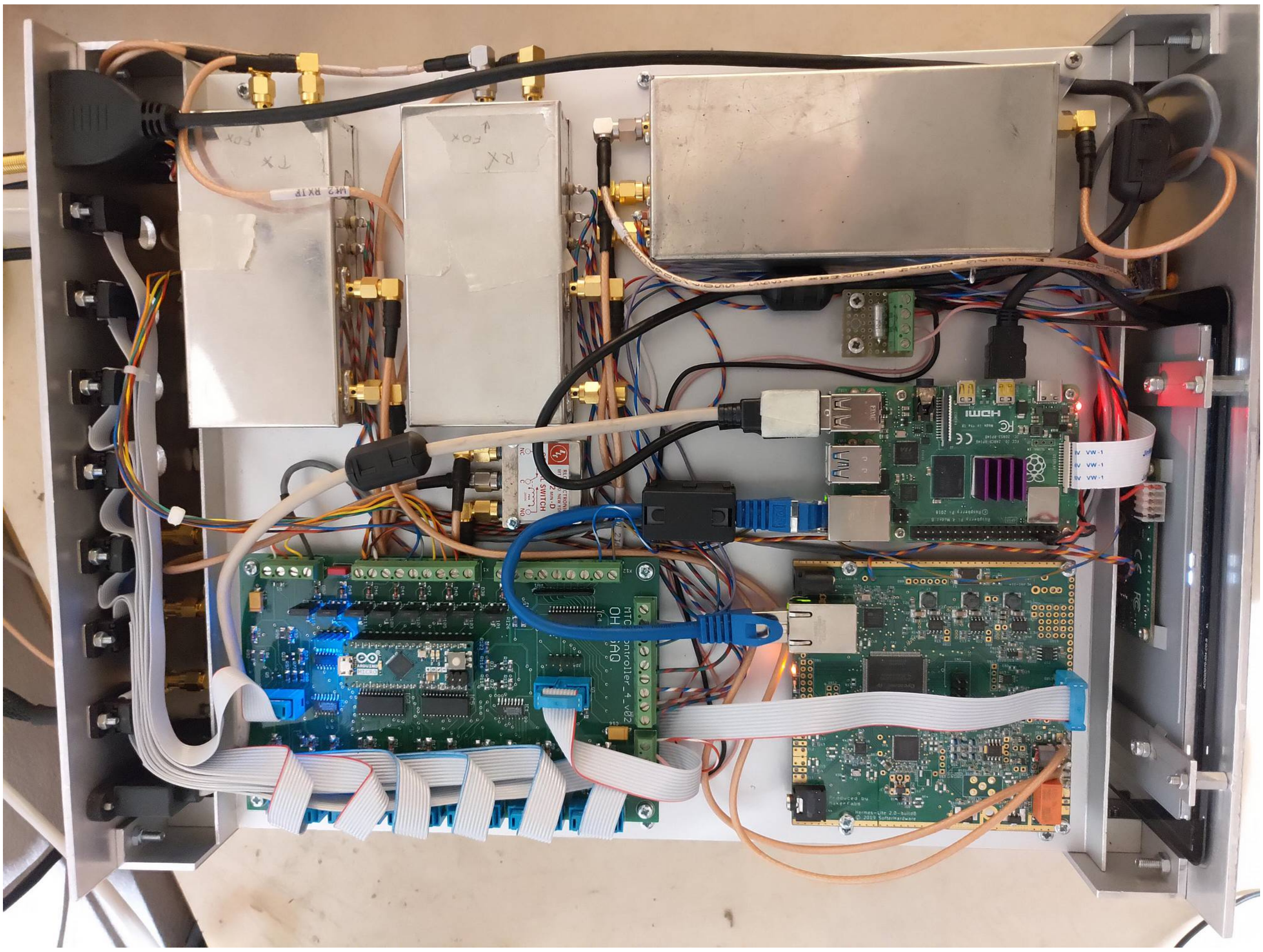

Figure 8. Top view of the MTC, showing clockwise from the top left, Tx 28 to 144 MHz converter, Rx 144 to 28 MHz converter, dual output 116 MHz PLL stabilized oscillator, Raspberry Pi 4 SBC, and display screen, Hermes Lite 2 SDR transceiver, Controller Interface board and cabling to the 7 Transverter interfaces.

The main mechanical structure in the MTC is provided by two 300mm x 330mm trays mounted on 20mm x 15mm "U" sections running from the front to rear panels, this structure can be partly seen in Figures 8 and 9. This allows the modules to be mounted independently on the top and bottom trays without interference. The space between the trays can be used to run cables from front to back. Modules are mainly mounted using tapped holes in the trays, allowing for easy removal and replacement with alternative implementations of the function.

The key low-level RF functions are in fully shielded tinplate boxes (144/28 up/down converters, 10 MHz references, 116 MHz LO's). Enough spare space was kept around the HL2 SDR for shielding, but it does not seem to be needed. All RF point-to-point connections are made using small diameter co-axial cables and SMA connectors. The 12V to 5V power converter uses special ITW Paktron capacitors on both the input and output side of the supply, to isolate the switching transients of the supply from the general DC distribution in the MTC.

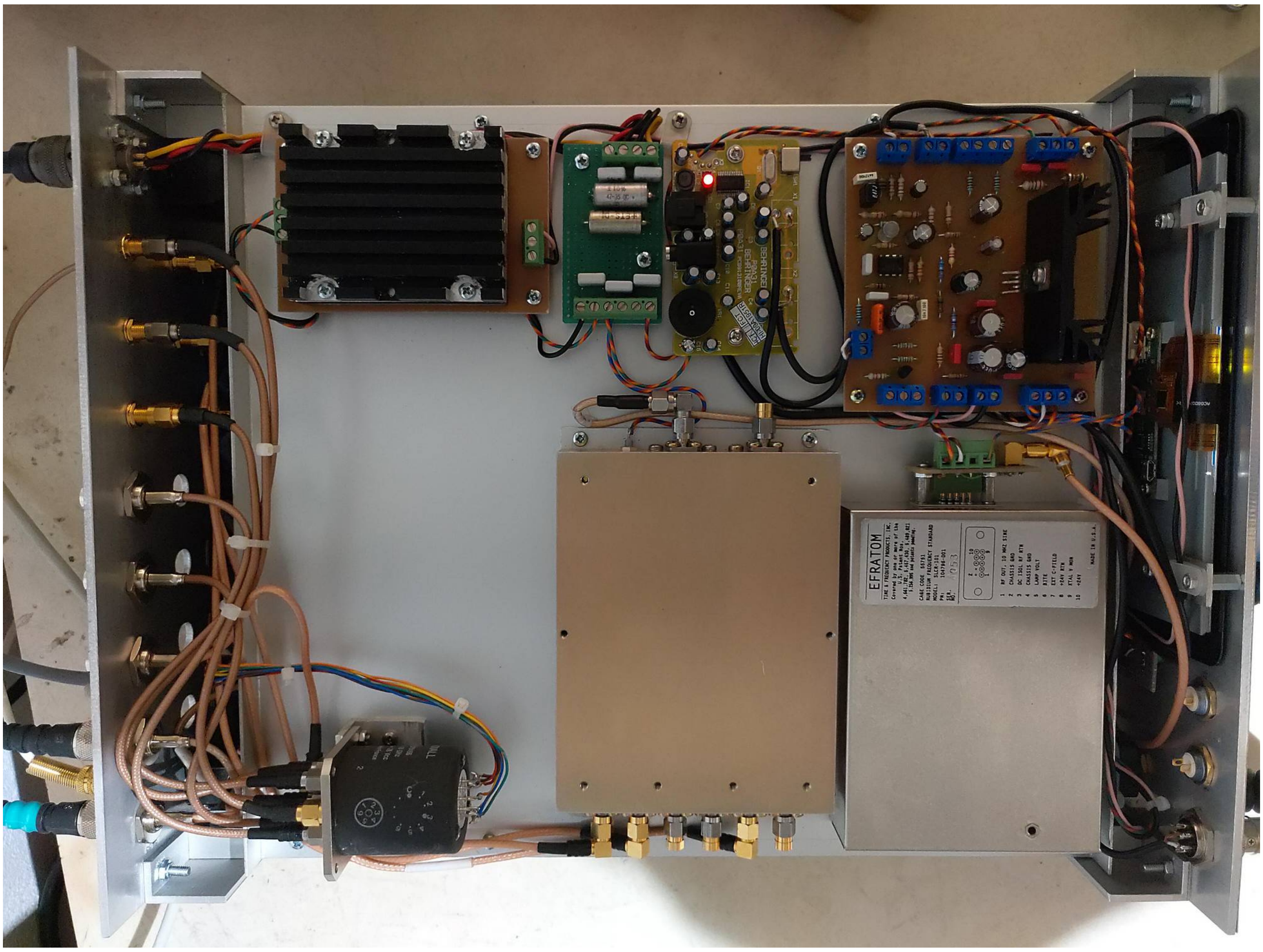

Figure 9. Underside of the MTC. showing clockwise from the top left, low noise high efficiency 12V to 5V switching converter, power conditioning filter, USB audio interface card, microphone pre-amp and speaker power amplifier, Rb. oscillator, 7 channel 10 MHz distribution amplifier and 6-pole co-axial switch for selection of half-duplex transverters. Spare space for adding a small Ethernet switch.

The rear panel is mainly occupied with the RF and control interface connectors to the up to 7 microwave transverters. However there is also one USB interface extended from the R. Pi 4, allowing a mouse or mouse+keyboard to be added. As well as that, one of the HDMI interfaces on the R. Pi 4 is extended to the back panel, allowing a large monitor/TV/projector to be attached if required.

Software environment and functionality

The SBC selected is an R Pi 4 with 4 Mbyte RAM. Earlier comparisons made with various SBC's showed that this was about the minimum configuration that could run the selected applications with a greater than once per second screen update rate (currently uses 7 per second) without causing problems with the audio stream breaking up. It would be expected the newly released R Pi 5 would also be suitable. Other small SBC's could be used.

Having selected to use a "linux" environment, the application environment was selected to be either Quisk or piHPSDR SDR sw's (which are both open sourced and can be readily extended if needed), and the FLDIGI suite for handling digital modes. For basic voice operation, including full-duplex satellite work, standard Quisk or piHPSDR sw can be used with the HL2. They both support full-duplex operation, and split frequency operation if required. The screen design is such that they work quite well in Quisk's "small screen mode" with a 7 inch touch screen, or 7 inch screen and mouse. All controls for normal operation are implemented as either "buttons" or "sliders". Once the configuration of either has been done, there is no need to use any "keyboard", either the touch screen or a mouse can be used for all operations. There are some functions present in Quisk (for example the ability to update the HL2 “gateware” over the Ethernet interface) that are not present today in piHPSDR. Also there is a function present in piHPSDR, not yet present in Quisk, which enables error conditions to lock out the SDR sw from Transmit mode should an error condition exist (e.g. during warmup when the frequency references are not stable, or when a transmit error condition occurs).

The MTC includes an optical encoder connected to the Arduino on the control board. The SW in the Arduino looks after operational functions of the up to 7 transverters, however it also includes a function that allows the optical encoder to operate as a USB bus mouse device, emulating the scroll wheel of a mouse. This allows the tuning function of the MTC to be done by either the scroll wheel on the mouse or the tuning knob on the front panel.

Some notes about SW configuration implementation details.

In the Quisk configuration options for the HL2, "Bands section", there is a freely configurable table. In the normal HL2 usage on HF, with the N2ADR filter board, this must be configured in an appropriate way for the correct HF filters to be selected. In the MTC usage, without the filterboard, it is used to provide a band number which is transferred to the SW on the Control board and used by the Arduino. In piHPSDR the “OC outputs” configuration table provides the equivalent function. The table below shows part of this "Bands" table:

Band	Transverter	Rx IO
28	00000000000	0000101
144	00116000000	0000110
1296	01268000000	0000000
2400	02372000000	0000111
3400	03372000000	0000001
5760	05732000000	0000010
10368	10340000000	0000011

The table above shows the way part of the band table is populated in the current equipping of my transverters.

The "Band" column shows the band displayed in the band selection screen of Quisk, however using metres or cm rather than frequency (e.g. 2400 is displayed as 13cm on Quisk control panel).

The "Transverter" column shows the "LO" frequency of the transverter. As there is also the 116 MHz of the inbuilt converters, this must be added to the actual transverter LO to calculate the correct offset for this column.

The Rx. IO column is the bit pattern/band number that is transferred to the Arduino.

The following fragment of Arduino code shows the table driven approach used to control the hardware interfaces on the Control board. The first column defines which interface to use for the band ("Transverter Select Bit", the second column defines which bit to check for TX OK, the third column which input to use for VSWR monitoring, the fourth for half or full duplex mode, and the fifth for which of the relays to select for the RF input/output in the half duplex mode cases.

<start of code fragment>

```
// Control array is used to define how each band from the HL2 is treated. It contains the values
// that are output to various ports or used as masks or used to determine which analog channel or
// if full duplex mode is used.
// Order: Transverter Select bit, Transverter TX OK bit, VSWR monitor, Duplex, Relay select bit

#define rows 8
#define columns 5

const byte Control_Array[rows][columns]={
  {B00000001,B00000001,B00000000,B00000000,B00000001},  // 1296
  {B00000010,B00000010,B00000001,B00000000,B00000010},  // 2400
  {B00000100,B00000100,B00000010,B00000000,B00000100},  // 5760
  {B00001000,B00001000,B00000011,B00000000,B00001000},  //10368
  {B00010000,B00010000,B00000100,B00000000,B00010000},  //24048
  {B00100000,B00100000,B00000101,B00000000,B00100000},  //   28
  {B01000000,B01000000,B00000110,B11111111,B00000000},  //  144
  {B01000000,B01000000,B00000110,B11111111,B00000000}   // 2400
```

<end of code fragment>

```
                        };
```

Operation with piHPSDR SW

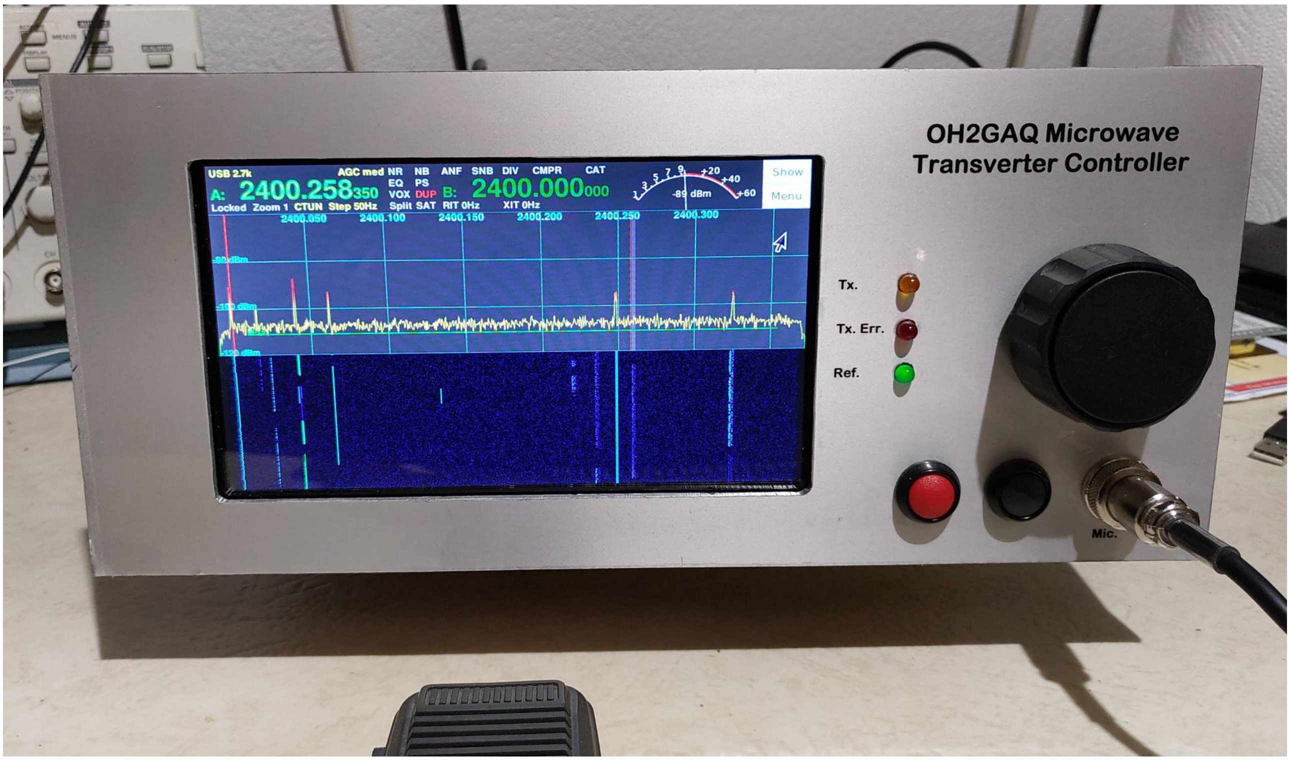

Figure 10. Screen when the piHPSDR SW application is used.

The piHPSDR application provides similar features to Quisk with a different screen layout. The version of piHPSDR used here supports lockout of transmit commands when there is a Tx. Disable signal sent to the HL2 Tx Inhibit input. The Tx. Disable signal is generated by combining signals inside the MTC (PLL Unlock, Rb. Reference not ready), plus the Tx. Ok signal from the selected transverter, taking into account the timeouts needed when activating the Tx. function in the selected transverter. Errors which indicate malfunction in the transverter subsequent to the initial transition to transmit mode are not subject to timeout, but acted on immediately. These include PLL inside the transverter unlock, PA overtemperature, PA overcurrent, high VSWR. This Tx. Disable signal is generated by SW in the Arduino Micro on the Controller board, and does not need any special function in the the piHPSDR SW except acting on the receipt of the HL2 Tx. Inhibit input.

Human Interface to Quisk and piHPSDR

One goal of this project was to eliminate having a large number of “real” switches and knobs on the front panel of the MTC, and also to allow control of the MTC in “normal” operation without necessarily needing a mouse and keyboard. There is only one knob on the front panel, plus two pushbuttons. During configuration operations using either Quisk or piHPSDR, it may be necessary to add a keyboard or keyboard + mouse to ease input of parameters to the SW application. But after the configuration has been done, the touchscreen or touchscreen plus knob can be used for all operations. The knob emulates the function of a mouse scroll wheel. By placing the cursor in the approximate location of any of the slider controls of Quisk or piHPSDR, the slider value may be easily adjusted using the knob. This includes the main tuning function. So when working from a portable location it is not necessary to use even a mouse. The Red button is used as either a Tx. On toggle, or as a reset function if there has been a Tx. Error condition notified. The Black button is a Tx. Off toggle. Alternatively the Mic. PTT button is used to control Tx. / Rx.

When used as a home station, it has been found most convenient to add a mouse with scroll wheel, as all functions including accurate tuning can be addressed without any large hand movements. As a large monitor can be attached to the rear HDMI connection on the MTC, a larger display can be used if desired. But the 7 inch internal display has been found to be adequate for all operations by the author.

Acknowledgements and References

There are many hams and others who have, generally unknowingly, contributed to the design and implementation of this system. The many excellent websites maintained by hams who wish to share their ideas and/or kits with others are too numerous to mention. More details of the implementation of some parts of the MTC, such as the PLL xtal oscillator, are contained on the authors web site: https://www.qsl.net/oh2gaq/.

For those who are interested in more exact constructional details of some parts of the system, for Eagle design files or SW source code, the author may be contacted directly via the address on the website.

References:

1. Hermes-Lite Designer and principal author Steve Haynal, KF7O
website: http://www.hermeslite.com/

2. Quisk Designer and principal author Jim Ahlsthom, N2ADR
Website: http://james.ahlstrom.name/quisk/

3. John Stephensen, KD6OZH, “A Stable, Low-Noise Crystal Oscillator for Microwave and Millimeter-Wave Transverters”, QEX, Nov/Dec 1999.

4. John Hazel, G8ACE, “Constructional Notes for G8ACE MKII OCXO Sept2010 V2”, Available from G8ACE website.

5. Analog Devices Data Sheet for ADF411x RF PLL Frequency Synthesizers.

6. Analog Devices ADIsimPLL version 3.30

7. W6PQL website: http://www.w6pql.com/
Several excellent articles covering microwave transverters and useful sub-systems as well as actual kits for many items.

8. KO4BB website: http://www.ko4bb.com/
Time and frequency control, measuring equipment and generally useful microwave related material.

9. KE5FX website: http://www.thegleam.com/ke5fx/
Time and frequency control, measuring equipment and generally useful microwave related material.
Article covering "Stability and Noise Performance of Various Rubidium Standards"

10. Hamish Kellock, OH2GAQ, "A Microwave Transverter Controller", QEX July/August 2013

11. piHPSDR current maintainer Christoph v. Wullen, DL1YCF.