QRP CW Transceiver Manual

Board rev. 2, Firmware rev. 2.5

Hannes Matuschek – DM3MAT

`<dm3mat [at] darc [dot] de>`

`https://dm3mat.darc.de/cw2019`

December 26, 2021

This document describes the assembly, alignment and usage of the second revision of my small, portable CW TRX. If you cannot find any *rev 2* markings on your boards, this is the wrong document!

The main difference to the first revision is the mixer. Instead of a `74HC4053`, which has some significant damping on higher frequencies, the second revision relies on the famous `FST3253` switch. This allows for using the TRX well into the 10m band. Additionally, the board layout has been changed to allow for board-to-board interconnects and proper connectors to ease the assembly of the final TRX. To this end there are only two boards (instead of three) with a $10 \times 8\,cm$ footprint that can be squeezed into a $10 \times 10 \times 5\,cm$ Fischer-chassis (`KOH-2100` + `KOH-4100` + `DPL 2-4`). Using this chassis, the receiver will weight about $580\,g$ (without batteries & key).

Contents

1 Receiver & Controller Board

The receiver and controller board houses the majority of the TRX circuitry. The receiver is a direct conversion receiver using the `FST3253` CMOS switch as a sample-hold mixer (in conjunction with `C4-C7`). The latter takes two VFO input signals with 90° phase difference and produces four mixing products with 0°, 90°, 180° and 270° phase shifts. These signal pairs (0° + 180°) and (90° + 270°) are fed into two independent difference-amplifiers (`U6A`, `U6B`) producing two 90° phase-shifted audio signals (for USB, −90° for LSB). The following phase-shift-network `U7` & `U8` then shifts back the audio phase by −90° at $700\,Hz$. Finally the resulting signals are added at `RV3`. Here the USB signals will interfere constructively while the LSB signals will cancel out. This phasing method allows for an unwanted side-band suppression in a direct conversion receiver. The next stages (`U3` & `U5`) will filter and amplify audio signals near $700\,Hz$, which are then fed into the audio PA `U4`.

The controller and VFO section of the board consists of a `ATMega 328P` MCU controlling the `SI5351` PLL oscillator. The `CLK2` signal of the PLL drives the two flip-flops `U9`, which divide the `CLK2` frequency by four and produce a proper 90° phase-shifted signal for the mixer.

1.1 Controller Assembly

In a first step, the controller section of the receiver board is assembled. Install `C2`, `C26`, `J7` connector, `L3`, `C38`, `C40`, `C41`, `C42`, `U10`, `J10`, `C14`, `J3`, `R3`, `C46`, `R4`, `U11` + Socket, `LCD1` connector, `R43`, `R5`, `RV4`, `J9` connector, `R41`, `R40`, `C39` & `U9` first.

> **ⓘ Remark**
>
> Due to the compact layout of the PCB, the component references might be hard to read. I therefore used a very nice plugin for KiCad to export an interactive BOM for this board. This certainly helps to find the spots where all components are placed, even if the silk-screen is hard to read.

> **⚠ Warning**
>
> The metal tab of the TO-220 package of `U10` is usually connected directly to the center pin (GND). Please check beforehand if this is the case for your variant. Some rare packages connect this pad to *+5V* leading to a nice & solid short-circuit that will blow either `U10` or `L3` or both.

If you got one of the rare TO-220 variant of U10 that connects the metal tab to +5V, use a mica-insulator between `U10` and the PCB. If you have enough head-room, you could also install `U10` vertically, heat-sinking to the PCB is not necessary.

`R43` limits the current to the LCD back-light. By default a $220\,\Omega$ resistor is installed. This limits the current to about $23\,mA$. This results in a very dim back-light but reduces the current consumption from max. $120\,mA$ (`R43` shorted). I personally consider this dim setting the most useful: During daylight a back light makes no sense, as the LCD is perfectly readable. During night without any external light-sources, a very dim back light is still sufficient to read the LCD. There is simply no need to drain the battery for unused or too-bright back lights. If you use the TRX only at home and current consumption is not an issue, you may choose a smaller value for `R43` or even bridge it for maximum brightness.

> **ⓘ Remark**
>
> You may also replace the resistor `R43` with a push-button at the front panel to switch the back-light on when needed.

Then, assemble the switch plug and (top right corner labeled SW). Connect a $12\,V$ power-supply (ideally current-limited to not more than $100\,mA$) to the pads of the supply connector (also top-right corner) labeled $+$ G, where the $+$ pad is $+12\,V$ and the G pad is ground.

> **✏ Remember**
>
> The receiver board itself has no reverse-polarity protection. So be careful when connecting the RX board to the power-supply!

If nothing blows up, switch the TRX off and continue assembling the rotary-encoder and LCD plugs.

The rotary encoder plug has the following connections (read left-to-right): *Button*, *Encoder B*, *Encoder A*, *Ground*.

The LCD-plug has the following connections (read left-to-right): *5V* pin 2 on LCD, *contrast* pin 3 on LCD, *LED backlight* pin 15 on LCD, *GND* pin 1 on LCD, *RS (register selection)* pin 4 on LCD, *D4-D7* pins 11-14 on LCD, *chip-enable (EN)* pin 6 on LCD. You also need to connect pins 5 and 16 on the LCD-board directly to ground (pin 1 on LCD).

Plug the LCD & rotary encoder into the board and install the `ATMega328` MCU. Then connect the board to the power-supply. If you got a pre-programmed MCU you should see a brief greeting on the LCD. If you do not see anything at all or a series of back blocks on the screen, you may have to set the LCD supply/contrast setting using RV4. If you still don't see anything on the LCD, double-check your connects to the LCD!

1.2 Testing PLL

In a next step, verify the proper operation of the `SI5351` PLL. First, install the `SI5351` break-out board.

> ⚠️ **Warning**
>
> Be careful when inserting the `SI5351` beakout-board, there is no protection in the socket to plug it in the wrong way: The board should overlap with the LCD connector not U9!

Then switch on the RX and verify that there is a $3.5MHz$ signal at pins 8, 9, 11 and 12 of `U9`. If not, verify that there is a $14.0Mhz$ signal at the `CLK2` output of the `SI5351` break-out board.

You may also verify the $90°$ phase-shift between pins 9 and 12 of `U9` using a dual-channel oscilloscope.

> ℹ️ **Remark**
>
> By default, the RX is wired to receive on the upper side-band (USB). If you prefer LSB-reception, you may break the jumper labeled *USB* and put a solder-bride on jumper *LSB*. You also need to set the side-band in the firmware.

If everything works fine for you so far, continue with assembling the rest of the receiver board.

1.3 Receiver Assembly

Before starting to assemble the receiver, disconnect everything from the board. That is, remove the `SI5351`, LCD and encoder wires and also remove the power-supply wires.

In a first step, install the SMD SOIC-16 `FST3253` mixer.

> ⚠️ **Warning**
>
> Be careful! The `SI5351` as well as the `FST3253` are **very** sensitive to electrostatic discharge (ESD). Wear an antistatic wrist wrap and use an antistatic mat (if available) and avoid wearing wool or synthetic fibers. I personally destroyed two `FST3253` that way.

If you never soldered SMD components before   It is surprisingly simple. Get a sharp soldering-iron tip and thin solder (e.g., $0.75mm$) as well as plenty of flux. Put a lot of flux on the pads. Then put a small amount of solder on one of the pins, for example pin 1 in the bottom-right corner. Pick up the IC with pointy tweezers and place it on the board. Verify the orientation and alignment of the IC. Solder only pin 1 to the board by heating the pad and pushing the IC on the board. Then take a magnifying glass and verify the alignment of all pins with their pads. If needed, reheat pad 1 and improve the alignment. Once the alignment is right, you can start to solder the remaining pads. You do not need much solder! And remember: The flux will do all the work for you. If you got a solder-bridge between two pins, grab some desoldering-wig and remove the solder.

Once the mixer is installed, it does not matter in which order the next components are installed. Complete the assembly of the board but leave the board-to-board interconnects (labeled *TX, RX, +G* and the unlabeled 10-pin connector at the bottom) as well as the RCA plug labeled *TX-OC* unpopulated for now. They will added in the final assembly step later.

> 🛈 **Remark**
>
> The audio PA (`LM386`, `U4`) has plenty of amplification. If you want to use head-phones, you may leave `R13` and `C16` off the board. You may add them later if more audio amplification is needed.

When winding the input transformer `T1`, cut 3 about 15-16cm long pieces of magnet wire and start the winding by pushing the three wires tough the core **from the back**. Hold them. Then make the next winding counter-clock wise by pushing the other end of the wires though the core from the front and continue to wind 8 turns on the core counter-clock wise. Every time the wires pass through the core, counts as a turn.
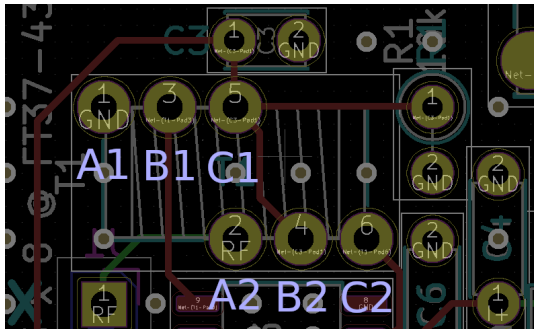
(a) Footprint of T1
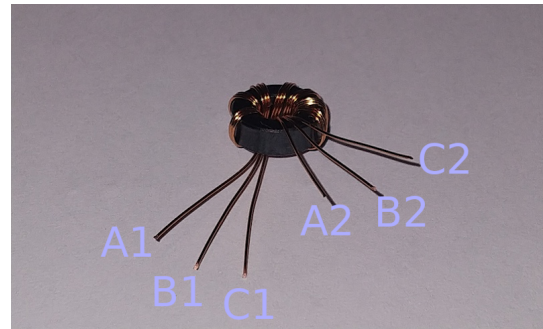
(b) T1 wound and labeled.

Figure 1: Before soldering the core, make sure the wires are aligned correctly using a multi-meter. That is, top-left connects to bottom left (i.e., A1 to A2), top-center to bottom-center (i.e., B1 to B2) and top right to bottom right pads (i.e., C1 to C2) on the board.

> ⚠ **Warning**
>
> The audio PA being used has a tendency to oscillate, in particular if a small speaker is used. If it does so, it may drive the complete audio-chain into saturation. In this case, the audio-chain may feed-back $12\,V$ to the FST3253 mixer. This may destroy the hard-to-get mixer! To prevent this problem, you **should** install four 5.1V Zener-diodes in parallel to C4-C7. This limits the voltage that can be fed back to the mixer and saves it.

1.4 Receiver Test and Alignment

Once the receiver is completed, re-connect the top-right pads *+ G* to the power-supply, prepare the volume and head-phone plug. Also connect a piece of coaxial cable to the *RX/GN* pads in the top-left. This is the antenna input for the test. Depending on the selected band, you may need to increase the current-limit of the power supply to about $120\,mA$.

1.4.1 Frequency Alignment

If a accurate frequency-counter is at hand, you may start your alignment by fixing the PLL offset. Select the 20m band and tune to $14.0007MHz$. Then measure the frequency at the USB pad. You should read exactly $14.000MHz$ (if the CW side-tone frequency is set to $700\,Hz$). If not, enter the Setup menu and select the *PLL correction* option. Now adjust the correction until you read $14.000MHz$ in the frequency counter.

1.4.2 CW Filter Alignment

Although the CW filter cannot be *tuned*, variations in component values may lead to small variations in the center frequency of the narrow CW audio filter. For a proper alignment of the RX frequency-shift,

the CW-tone must be set to the center frequency of the CW filter. Hence, measure the spectrum (e.g., using your PC sound card and an application that can display the audio spectrum) of the audio signal. Determine the center frequency of the CW filter and set the CW side-tone frequency (under `Menu →` `Setup → CW Tone`) to that value. This will also adjust the RX frequency-shift.

> ✏️ **Remember**
>
> The displayed frequency is the frequency, the TRX will transmit on. It is not the beat-frequency the TRX is receiving on.

1.4.3 Side-band Suppression

If an signal generator is at hand, tune the generator to $14.000MHz$. Then select the 20m Band and tune to $14.0012MHz$. You may hear a low-frequency tone (about $500Hz$). Tune `RV2` & `RV3` to minimize the unwanted side-band tone. Then tune to $14.0016MHz$. You may hear a high-frequency tone (about $900Hz$). Tune `RV1` & `RV3` to minimize the unwanted side-band tone. You may repeat the previous steps several times to obtain the optimal result. In my experience, the unwanted side-band suppression will work satisfactorily on all bands once aligned properly on 20m.

With these steps, the alignment of the RX is completed and you may take the RX on a short ride through the bands.

1.5 RX Component List

#	Component	Value	Remarks
1	R15	10R	
4	R20, R21, R22, R23	100R	
1	R43	220R	
3	R38, R39, R42	1k	
1	R13	2.2k	
9	R7, R27, R29-R31, R33, R35-R37	3.3k	
1	R32	4.3k	
1	R4	4.7k	
1	R26	5.1k	
1	R34	7.5k	
10	R1, R2, R6, R11, R18, R24, R25, R28, R40, R41	10k	
2	R9, R10	30k	
2	R12, R14	33k	
2	R16, R17	36k	
2	R3, R5	47k	
1	R8	120k	
1	R19	470k	
2	RV1, RV2	50k	
1	RV3	500R	
1	RV4	10k	
3	C13, C20, C23	1n	
2	C25, C27	3.3n	
3	C30, C31, C35	10n	
5	C12, C17, C18, C21, C33	47n	
23	C3, C8, C11, C22, C24, C26, C29, C32, C34, C37-C41, C46-C54	100n	
5	C4, C5, C6, C7, C36	470n	
2	C1, C28	1u	
3	C15, C16, C42	10u	
3	C2, C14, C19	100u	
8	L1, L3, L4, L5, L6, L7, L8, L9	100u	
1	T1	3 x 8 @ FT37-43	
1	D1	1N4148	
4	in parallel to C1-C4	5.1V Zener	
1	Q1	BS170	
1	Q2	2N3904	
1	U1	FST3253	
5	U3, U5, U6, U7, U8	LM833	
1	U4	LM386	
6	–	DIP-8 Sockets	
1	U9	74AC74	
1	U10	L7805	
1	U11	ATmega328-PU	
1	–	DIP-28 Socket	
1	-	SI5351 break-out	
2	J1, J3, J4, J11	1x10 pin-header long	
1	J6	2x3 pin-header	
1	J9	1x7 pin-socket	
2	J2, J7	1x2 90deg	
1	–	Switch	
1	J8	1x3 90deg	
1	–	10k log	potentiometer
1	J10	1x4 90deg	
1	–	Rotary encoder	
1	LCD1	1x8 90deg	

1	–	LCD	2 x 8 symbols
1	J13	TX-OC	RCA-jack (optional)
1	J5	Key	3.5mm stereo jack

2 PA & Low-Pass Board

The majority of the PA/LP board consists of four 7-pole Chebyshev low-pass filters. The Chebyshev-type filters are needed to get away with only four filters to cover all RF bands from 80m to 10m. The right-most filter covers the 80m and 60m band and should have a cutoff frequency near $5.6\,MHz$. The next filter covers 40m and 30m and should have a cutoff frequency near $10.5\,MHz$. The second-to-last filter covers 20m and 17m and should have a cutoff frequency near $20\,MHz$. Finally the leftmost filter will cover 15m, 12m and 10m with a cutoff frequency near $30\,MHz$. The low-pass filters are switched by 4 sub-miniature relays controlled by the RX-board via the board-to-board (b2b) connector `J2`.

The `Q4` mosfet acts as the TX/RX switch passing the LP-filtered RF from the input to the RX via b2b `J6`.

The PA section consist of a `74HCT00` as driver, four `BS170` mosfets as PA transistors and a `BD140-16` for key-shaping. The TX oscillator signal arrives from the RX-board via b2b `J7`.

The complete TRX is powered from the barrel-jack `J3` and the supply voltage is passed to the RX-board via b2b `J11`.

> **ⓘ Remark**
>
> Before assembling the low-pass filers on the board. Please assemble them ugly-style on a piece of PCB material and measure their properties using a network analyzer or spectrum analyzer. Being Chebyshev low-pass filter, the exact component values are critical!

If no measurement equipment is present. Please subtract one winding from the given values below. The number of turns for each core was determined using AL-values provided by the manufacturer. In my experience, this leads to an slightly overestimation of the number of turns by about one. Moreover, you are probably more willing to risk a lower stop-band suppression rather than an increased damping in the pass band.

> **ⓘ Remark**
>
> Like for the RX board, the PA board has its own interactive BOM.

The diode `D4` is connected in parallel to the barrel-jack and acts together with a fuse as a reverse-polarity protection. If the supply is connected to the TRX in reverse-polarity, the diode will short the input which will cause a huge current that blows the fuse. A 2A *flink* fuse will do.

Finally install the b2b **sockets** on the PA/LP board, that is `J2`, `J6`, `J7` and `J11`.

2.1 PA/LP Component List

#	Component	Value	Remarks
2	R12, R13	100	
1	R9	470R	
1	R10	1k	
1	R14	4.7k	
7	R1, R3, R5, R7, R11, R15, R16	10k	
4	R2, R4, R6, R8	100k	
2	C2, C15	100p	NP0
4	C3, C6, C11, C16	220p	NP0
4	C4, C7, C12, C17	470p	NP0
2	C5, C18	820p	NP0
2	C8, C13	1n	NP0
2	C9, C14	1.5n	NP0
11	C1, C10, C19, C20, C22, C23, C24, C27, C28, C29, C30	100n	
1	C25	1u	
1	C26	220u	
1	L14	22u	07HCP
1	L13	100u	MICC
3	L1, L5, L9	330n	11T on T37-6
3	L2, L6, L10	480n	13T on T37-6
3	L3, L7, L11	750n	14T on T37-2
3	L4, L8, L12	1700n	21T on T37-2
1	T1	–	8T + 8T on FT37-43
4	K1, K2, K3, K4	G6S-2	
6	D1, D2, D3, D5, D6, D7	1N4148	
1	D4	1N4001	
4	Q1, Q2, Q3, Q10	2N3904	
6	Q4, Q6, Q7, Q8, Q9, Q11	BS170	
1	Q5	2SB 772	
1	U1	74HCT00	
1	J1	BNC	
1	J2	1 x 8 pin-socket	
1	J3	barrel-jack	
3	J6, J7, J11	1 x 2 pin-socket	

3 Final Assembly

Finally, the board-to-board interconnects are installed. The way these connectors are installed, depends on the chassis and distance between the boards you chose. The example I give here applies to the Fischer `KOH-2100` + `KOH-4100` + `DPL 2-4` kit. This is a nice and compact $10 \times 10 \times 5\,cm$ chassis. The boards slide into the chassis with $\approx 1.5\,cm$ distance. Unfortunately, I was not able to find stackers or long pin-headers with a matching length here in Germany. Have a look at Ebay or at Mouser. Pin headers with insertion lengths of about $12\,mm$ should do the trick. I, however, went with stackers of $38\,mm$ total length which is is way too long and therefore needed to be shortened.
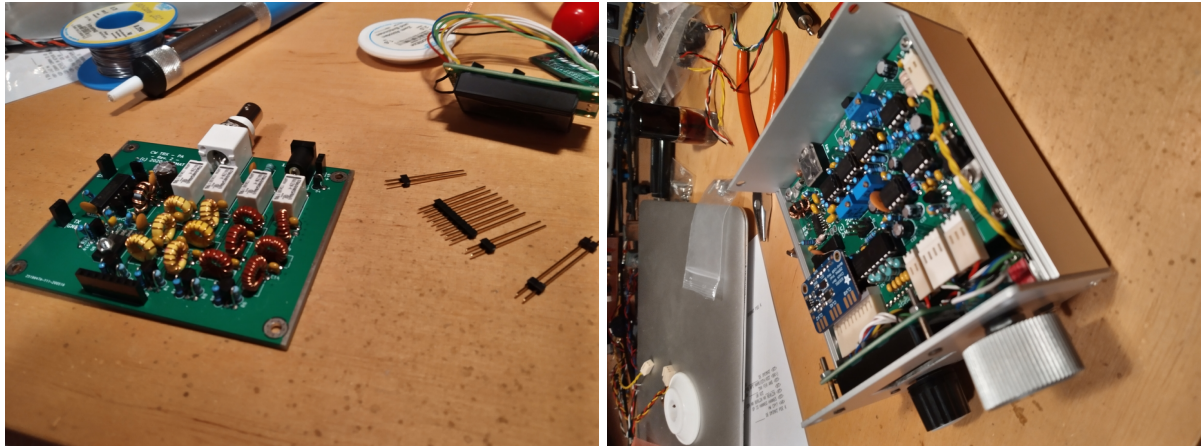
(a) Cut stackers to get long pin-headers, before inserting them into the socket of the PA board.

(b) Put RX board on top, insert into chassis, solder pins to RX board and cut pins to length.

Figure 2: Final assembly of the board-to-board interconnects.

The easiest way to do that, is to cut the top-part away from these stackers (see figure 2a) and stick them into the sockets on the PA board. Than put the RX board on top. Now carefully insert both boards in the chassis. Make sure that you can close the chassis. Finally solder the stackers/long pin-headers into the RX board.

> ⚠️ **Warning**
>
> Do not overheat the stackers/pin-headers as the socket may melt. Just tag them in-place and reflow the header connections once the PA-board is removed.

When the TRX is installed into the Fischer chassis mentioned above, the RCA connector at the back of the RX board to key an external PA will not fit. The boards are way to close together. If a different chassis is used, the connector can be installed right now.

3.1 Final Testing

With these final steps completed, it is now time to test the PA and RX/TX switching. Reconnect everything and connect a dummy-load to the output. Measure the RF voltage-drop across the dummy-load (oscilloscope). Alternatively, connect a power/SWR meter between the TRX and dummy-load. You should get about $22\,Vp$ across the dummy-load or about $5\,W$ on transmit, depending on the supply voltage (assuming about $13.8\,V$). On the lower band (e.g., up to $30\,m$) it might even be a little bit more.

If your output is way less, measure the peak voltages at the PA-output and LP-input and double-check the low-pass filters. If the output matches, the assembly is complete.

3.2 Hot fixes

The second revision was a redesign to some extend. Hence not everything is perfect yet. For example, the `74AC74` produces a lot of high-frequency ripple on the 5V rail. To suppress them stronger, a $470\,pF$ capacitor should be soldered directly from pin 16 (5V) to pin 8 (GND).

The audio PA has a tendency to oscillate, in particular if a small speaker is used. If it does so, it may drive the complete audio-chain into saturation. In this case, the audio-chain may feed-back $12\,V$ to the FST3253 mixer. This destroys the hard-to-get mixer! To prevent this problem, you **should** install four 5.1V Zener-diodes in parallel to C4-C7 on the receiver board. This limits the voltage that can be fed back to the mixer and saves it.

3.3 Mechanical Parts

#	Component	Remarks
1	chassis top	
1	chassis bottom	
1	chassis front/back	
1	10k log pot	
1	rotary encoder	
1	LCD display	
1	head-phone socket	
1	switch	
1	barrel plug	
2	fuse holder	optional
2	2A flink	optional
1	speaker	optional

4 Build & Load the Firmware

To program the firmware onto the device in the first place or update the firmware to a newer version, a so-called in-system programmer ISP is needed. This is a small USB device that provides the interface from the computer to the controller (MCU). There are many cheap AVR ISPs out there (usually between 5€ and 10€ on ebay). It is also possible to build a makeshift ISP for the good old serial-port or using a Arduino as ISP.

Once you got your hands on an ISP, you still need some software to upload the firmware to the device. I used avrdude under Linux for this but others should work as well. If you got a working setup, you can simply download the pre-compiled firmware and EEPROM image from my github page and program the device immediately. You only need to set the fuses of the AtMega first to low=`0xE2` and high=`0xD9`. These fuse settings tells the MCU to use the internal 8Mhz RC-oscillator (no pins left for a crystal) and not to divide the clock by 8. All other settings are left as default.

```
1 MISO   ■  ■   2 VCC
3 SCK    ■  ■   4 MOSI
5 RST    ■  ■   6 GND
```

You can find the 6-pin ISP header (J3) on the RX board. Pin 1 is marked by a small circle on the corner. Consequently the small tab on the ISP plug must face away from the MCU.
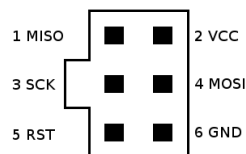
When programming the MCU, you need to first set the fuses. This step must only be done once as these settings are conserved across programming cycles. In a second step the firmware (`firmware-atmega328p.hex`) is written to the MCU. This step usually erases the EEPROM content. The EEPROM is used to store settings, last frequency etc. The firmware may not properly work with some uninitialized EEPROM. Hence, after the firmware was written, the default EEPROM image (`firmware-atmega328p-eeprom.hex`) must be written to the MCU.

4.1 Building Firmware from Scratch

In this section, I like to describe building and loading the firmware from scratch on a Ubuntu/Debian system (e.g., on a Raspberry Pi). Before you can start a bunch of software needs to be installed. That is the cross compiler, build tools and programming software. So run on a command line

```
sudo apt-get install git cmake gcc-avr avrdude
```

`git` is the version control system I use. It is used to download the latest source code. `cmake` is the build system. There is a nice extension to cmake for the avr-gcc and avrdude that makes building and writing the firmware to the MCU easy. `gcc-avr` is the cross-compiler for the AVR devices. Finally, `avrdude` is the programming tool to write the firmware to the device.

Once all the software is installed, you need to download the firmware source from my github page. So call

```
git clone https://github.com/hmatuschek/cwtrx.git
```

You will now find a new directory named `cwtrx`. Before the firmware can be programmed to the device, you may need to adjust some settings to reflect your programmer. The first lines in the `cwtrx/firmware/-CMakeLists.txt` specify which programmer you use and where to find it.

By default you should find something like this

```
1  # Specify programmer
2  set(AVR_UPLOADTOOL avrdude)
3  set(AVR_PROGRAMMER stk500v2)
4  set(AVR_UPLOADTOOL_PORT /dev/ttyACM0)
```

In the second line, you can set the programming tool. Here avrdude is used. The third line specifies the ISP programmer being used. Here I used a STK500V2 compatible one (actually called DIAMEX-AVR). Finally, line 4 specifies the port where the programmer can be found. This programmer appears as a serial port to the computer (as device `/dev/ttyACM0`). You may need to change lines 3 and 4 to match your local setup.

Once everything is set up, you can start to build the firmware. Enter the `cwtrx` directory, create a so-called build directory and enter it as well.

cd cwtrx
mkdir build
cd build

In the next step, the build will be configured with

cmake ..

This step will search for the AVR GCC and prepare the build. Once completed, build the firmware and EEPROM image with

make

After this step, you need to set the fuses of the MCU. Simply call

make set_fuses

This step may fail if the MCU is brand new. Just retry several times. Once the fuses are set, the communication with the MCU should be stable. Then upload the firmware to the MCU with

make upload_firmware

and finally write the EEPROM image with

make upload_eeprom

Your TRX is now ready.

4.2 Updating the Firmware

Assuming you already performed the steps above, updating the firmware is quiet straight forward. Enter the `cwtrx` directory and run

git pull
cd build
make
make upload_firmware
make upload_eeprom

And you are ready to go.

5 Usage

Handling a feature-rich TRX using a single rotary encoder with a single push-button is kind of challenging. I've designed a two-level menu navigation that puts everything that is needed frequently in the first level (blue in figure 3) and everything else in a second level. This keeps the navigation fast for frequently used options.

The second menu level (red in figure 3) contains all settings that are usually not touched during the operation. They concern the basic setup and alignment of the TRX.
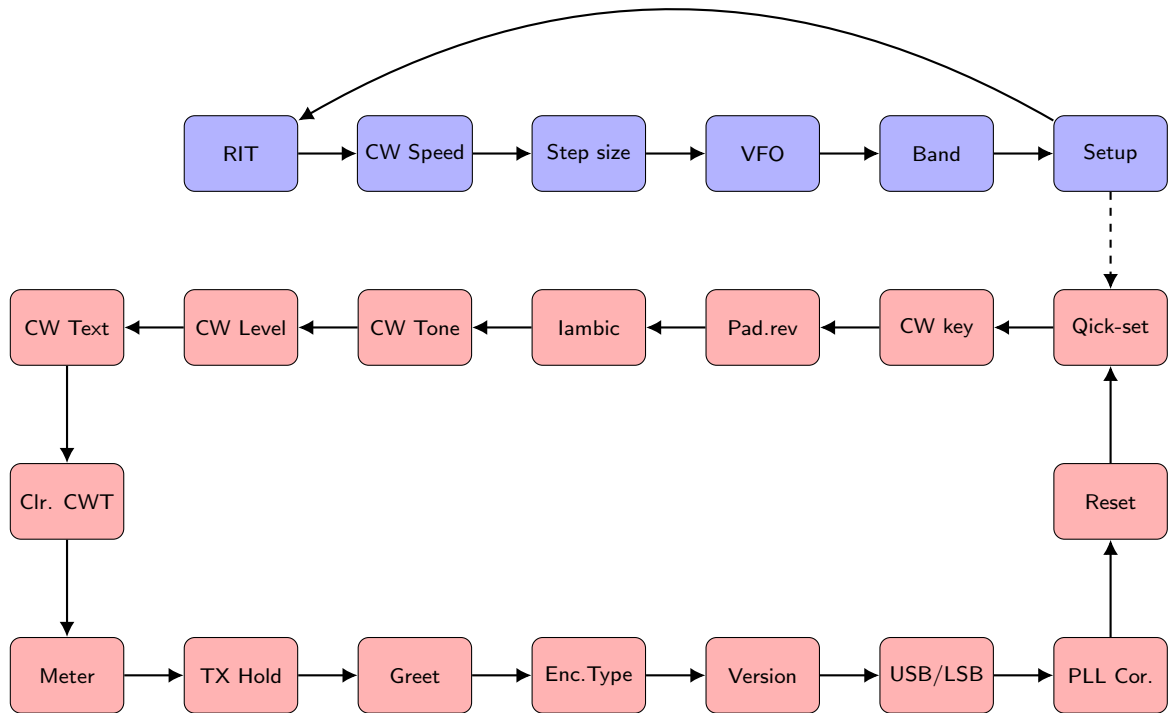
Figure 3: Menu navigation

To enter the menu, just click the button on the rotary encoder. You will land on either the RIT setting or on the last modified setting. Using the rotary-encoder you can navigate through the first menu level. To change a settings, click the center button again. You can now change the selected setting using the rotary encoder. To finish the setting and leave the menu, click the center button again.

To enter the second menu-level, enter the first level by clicking on the center button and select the menu entry `Setup`. Click on the center button again to enter the second-level menu. You will land on the `Quick-set` setting. You may now navigate the second menu-level using the rotary encoder. Changing and leaving the second-level menu works in the same way as the first-level menu.

5.1 Gestures

The firmware distinguishes four different *gestures* of the rotary encoder:

1. *tune* - just rotate the encoder.

2. *click* - press and release the center button of the encoder.

3. *long-press* - press and hold the center button for at least 2 seconds.

4. *press-rotate* - press and rotate the encoder.

During RX, a *tune* gesture will simply tune the frequency (of cause), a *click* gesture will enter the menu, a *long-press* gesture will send the keyer memory (if set) and a *press-rotate* gesture will quick-set a property (if specified).

5.2 Settings

The *settings* are all settings that might be changed during normal operation of the TRX. Hence they are located at the first level of the menu for the sake of quick access.

Receive/Transmit offset (RIT) Allows to fine-tune the RX frequency while keeping the TX frequency constant. A non-zero RIT will be indicated by a `+` or `-` symbol in the bottom-right corner of the display during RX.

CW speed Specifies the speed in WPM for the iambic and automatic keyer.

Tuning step-size Specifies the step-size when tuning the TRX.

VFO Specifies the current VFO and VFO-mode. That is `VFO A`, `VFO B` or `Split`.

Band Specifies the current band.

5.3 Quick-Set

The *quick-set* feature allows you to modify a selected setting without entering the menu at all. Just press & rotate the encoder.

To use this feature, you have to specify the settings you want to *quick-set* under `Menu → Setup → QuickSet` (see below).

5.4 Setup

Setup hides all settings that are usually not touched during regular operation and are therefore hidden in the second menu-level. You can access these settings by selecting `Menu → Setup`.

Quick-set The quick-set feature allows to set a specific property without entering the menu. By holding and turning the rotary-encoder at the same time. This menu item allows to select one of four settings to be manipulated by the quick-set feature: RIT, keyer speed, tuning step-size, band or *none*. The latter disables the quick-set feature (default).

Key type The key type specifies the type of keyer you use. Possible options are *straight key*, *paddle* and *iambic*. The TRX switches automatically to *straight key* if the TRX detects a mono-plug at startup.

Paddle reverse If enabled, the meaning of the two paddles are reversed. That is, the left paddle will produce a series of *dahs* and the right one produces a series of *dits*.

Iambic mode Specifies either *iambic A* or *iambic B* modes for the automatic keyer if the type is set to *iambic*.

CW side-tone level Specifies the volume of the CW side-tone. This is a value between 1 and 255.

Keyer memory `CW Text` Specifies the text to be send automatically (e.g., a CQ call) when the rotary encoder is long-pressed. If no text is set, nothing will happen.

To edit the CW text head to `Menu → Setup → CW Text` and *click* the encoder to enter the editor. You should see a cursor beneath the first character. You can use the encoder to select the character you want to edit. *Click* again and the character should start to blink. You can now change the selected character. Once the desired character is found, *click* again to set it. You are now back at the editor level and the next character is automatically selected. Once the text is edited, leave the editor by *long-pressing* the encoder.

Clear keyer memory Simply clears the keyer memory.

Meter selection Selects a meter to display during RX. This can be `Volt` to display the supply voltage or `Temp` to display the core-temperature of the MCU.

TX-hold time Specifies the delay between TX and RX in *ms*. By default a value of $50\,ms$ is selected. This small value allows for quasi-QSK but is sufficiently long to avoid ugly clicks in the audio path when switching from TX to RX.

> **ⓘ Remark**
>
> When operating a PA or in split mode, a longer delay should be chosen to allow the relays to switch properly.

Greet text An editor that allows you to change the *greeting* at startup. The editing is similar to the *keyer memory* editor above.

5.5 Alignment

These settings can be found under the *Setup* sub-menu and are usually only changed during the alignment of the TRX.

CW side-tone frequency Specifies the CW side-tone frequency as well as the RX frequency shift. This frequency should be identical to the center frequency of the narrow CW audio filter.

Encoder type Specifies the encoder type used. There are two different generic encoder-types `A` & `B`. If you ordered exactly the encoder listed in the BOM, you should have a type-A encoder (default). If you encounter problems with the encoder (e.g., double-steps) you may change this setting to type B. If you wired the encoder in the wrong way (wrong tuning direction) you may simply change this setting to `A rev` or `B rev`.

PLL correction This setting specifies the PLL clock frequency correction in PPM. See `RX Alignment` above.

Factory reset The previous version of the TRX encountered some issues with the EEPROM memory, leading to corrupted settings which turn the TRX unusable. This option allows to reset the entire EEPROM memory to *factory settings* fixing the corrupted memory issue.
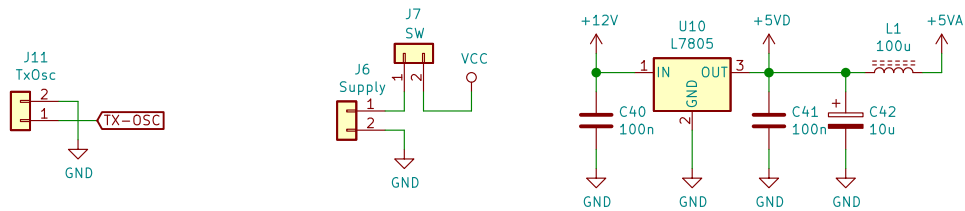
6 Bill-of-Material (BOM)

#		Value	Reichelt	Cost	Sum
1	☐	10R	METALL 10,0	0,05 €	0,05 €
6	☐	100R	METALL 100	0,05 €	0,29 €
1	☐	220R	METALL 220	0,05 €	0,05 €
1	☐	470R	METALL 470	0,05 €	0,05 €
4	☐	1k	METALL 1,00K	0,05 €	0,20 €
1	☐	2.2k	METALL 2,20K	0,05 €	0,05 €
9	☐	3.3k	METALL 3,30K	0,05 €	0,44 €
1	☐	4.3k	METALL 4,30K	0,05 €	0,05 €
2	☐	4.7k	METALL 4,70K	0,05 €	0,10 €
1	☐	5.1k	METALL 5,10K	0,05 €	0,05 €
1	☐	7.5k	METALL 7,50K	0,05 €	0,05 €
17	☐	10k	METALL 10,0K	0,05 €	0,83 €
2	☐	30k	METALL 30,0K	0,05 €	0,10 €
2	☐	33k	METALL 33,0K	0,05 €	0,10 €
2	☐	36k	METALL 36,0K	0,05 €	0,10 €
2	☐	47k	METALL 47,0K	0,05 €	0,10 €
4	☐	100k	METALL 100K	0,05 €	0,20 €
1	☐	120k	METALL 120K	0,05 €	0,05 €
1	☐	470k	METALL 470K	0,05 €	0,05 €
2	☐	50k	64W-50K	0,31 €	0,62 €
1	☐	500R	64W-500	0,27 €	0,27 €
1	☐	10k	ACP 6-L 10K	0,21 €	0,21 €
2	☐	100p	NPO-2,5 100P	0,07 €	0,14 €
4	☐	220p	NPO-2,5 220P	0,08 €	0,32 €
4	☐	470p	NPO-2,5 470P	0,09 €	0,36 €
2	☐	820p	Conrad: 1578706	0,16 €	0,32 €
3	☐	1n	X7R-2,5 1,0N	0,07 €	0,21 €
2	☐	1n	NPO-2,5 1,0N	0,09 €	0,18 €
2	☐	1.5n	X7R-2,5 1,5N	0,07 €	0,14 €
2	☐	3.3n	X7R-2,5 3,3N	0,07 €	0,14 €
3	☐	10n	X7R-2,5 10N	0,07 €	0,21 €
5	☐	47n	X7R-2,5 47N	0,11 €	0,55 €
35	☐	100n	Z5U-2,5 100N	0,05 €	1,75 €
5	☐	470n	Z5U-5 470N	0,19 €	0,95 €
3	☐	1u	Z5U-5 1,0µ	0,26 €	0,78 €
3	☐	10u	RAD 10/35	0,02 €	0,06 €
3	☐	100u	RAD 100/16	0,03 €	0,09 €
1	☐	220u	RND 150ECR AY	0,06 €	0,06 €
1	☐	22u	L-07HCP 22µ	0,38 €	0,38 €
9	☐	100u	L-MICC 100µ	0,27 €	2,43 €
2	☐	FT37-43	FT 37-43	0,58 €	1,16 €
6	☐	T 37-6	T 37-6	0,78 €	4,68 €
6	☐	T 37-2	T 37-2	0,54 €	3,24 €
7	☐	1N4148	1N 4148	0,02 €	0,14 €
1	☐	1N4001	1N 4001	0,02 €	0,02 €
4	☐	5.1V Zener	RND 1N751A	0,02 €	0,08 €
7	☐	BS170	BS 170	0,10 €	0,70 €
5	☐	2N3904	2N 3904	0,04 €	0,20 €
1	☐	2SB772	SB 772	0,24 €	0,24 €
1	☐	FST3253	SOIC-16, Ebay	2,00 €	2,00 €
5	☐	LM833	LM 833	0,90 €	4,50 €
1	☐	LM386	LM 386 DIP	0,22 €	0,22 €
6	☐	DIP-8 socket (optional)	GS 8	0,04 €	0,24 €
1	☐	74AC74	Kessler: 522421	0,36 €	0,36 €
1	☐	74HCT00	74HCT 00	0,33 €	0,33 €

1	☐	L7805	L 7805 CV	0,25 €	0,25 €
1	☐	ATmega328-PU	ATMEGA 328P-PU	2,50 €	2,50 €
1	☐	DIP-28 socket (optional)	GS 28-S	0,10 €	0,10 €
1	☐	SI5351 breakout	Adafruit	7,35 €	7,35 €
4	☐	G6S-2	NA 12W K	0,96 €	3,84 €
2	☐	1x10 pin-header long	STAPELLEISTE 10	0,38 €	0,76 €
1	☐	2x3 pin-header	MPE 087-2-006	0,14 €	0,14 €
1	☐	1x7 pin-socket	MPE 156-1-007	0,37 €	0,27 €
2	☐	1x2 90deg	PSS 254/2W	0,02 €	0,04 €
2	☐	1x2 plug	PSK 254/2W	0,02 €	0,04 €
1	☐	1x3 90deg	PSS 254/3W	0,04 €	0,04 €
1	☐	1x3 plug	PSK 254/3W	0,02 €	0,02 €
1	☐	1x4 90deg	PSS 254/4W	0,09 €	0,09 €
1	☐	1x4 plug	PSK 254/4W	0,02 €	0,02 €
1	☐	1x10 90deg	PSS 254/10W	0,10 €	0,10 €
1	☐	1x10 plug	PSK 254/10W	0,08 €	0,08 €
2	☐	contacts	PSK-KONTAKTE	0,29 €	0,58 €
2	☐	head-phone socket	EBS 35	0,30 €	0,60 €
1	☐	TX-OC	CBP	0,21 €	0,21 €
1	☐	BNC	UG 1094W1	0,99 €	0,99 €
1	☐	1 x 8 socket	MPE 094-1-008	0,24 €	0,14 €
1	☐	barrel jack	DC BU21 90	0,46 €	0,46 €
3	☐	1 x 2 socket	RND 205-00642	0,02 €	0,06 €
1	☐	chassis top	KOH-2100	5,40 €	5,40 €
1	☐	chassis bottom	KOH-4100	5,80 €	5,80 €
1	☐	chassis front/back	DPL 2-4	7,99 €	7,99 €
1	☐	10k log pot	RK11K112-LOG10K	1,35 €	1,35 €
1	☐	rotary encoder	STEC11B03	3,70 €	3,70 €
1	☐	LCD display	LCD-PM 2X8-5 A	7,30 €	7,30 €
1	☐	switch	KIPP 1A11	1,20 €	1,20 €
1	☐	barrel plug	HS 21-13	0,20 €	0,20 €
2	☐	fuse holder (optional)	RND 170-00170	0,19 €	0,38 €
2	☐	2A flink (optional)	ESKA 520.020	0,09 €	0,18 €
1	☐	speaker (optional)	LSM-S20K	2,70 €	2,70 €
Total[1]					**89,26 €**

6.1 Shave off some € from the BOM

The single most expensive item on the BOM is the chasis. You can certainly find some extruded aluminium chassis for less than € 5 on Ebay. You can also get cheap clones of the Adafruit SI5351 break-out board for around € 3, cheap rotary encoders (€ 1) and 2x8 LCD displays (€ 3) there. I shoud announce a competition for the cheapest BOM for this TRX.

---

[1]Prices may change over time.
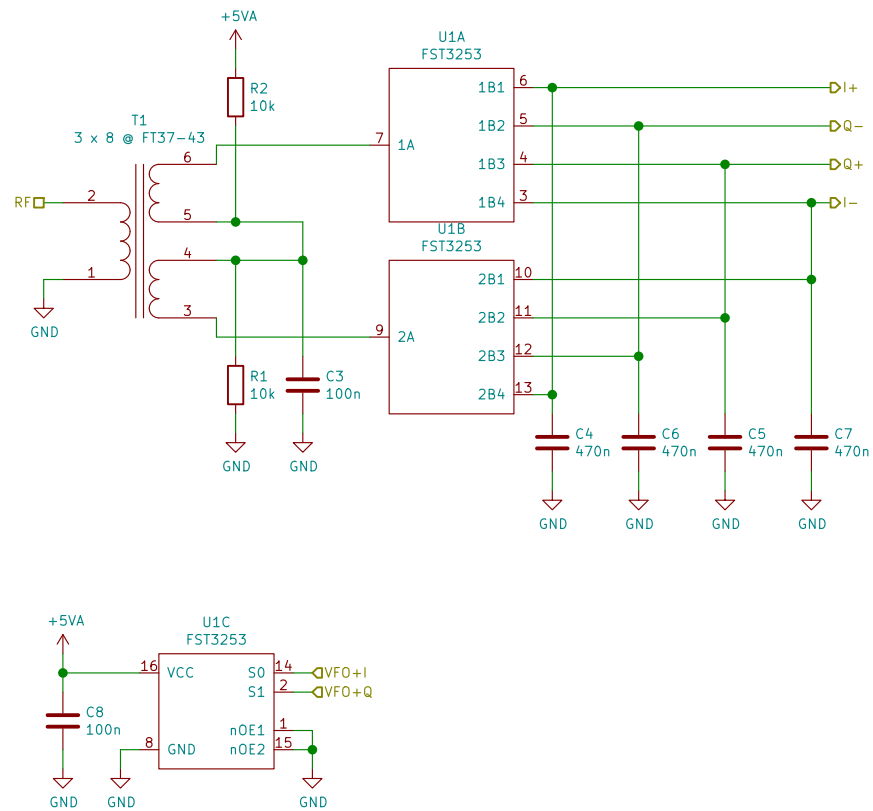
Sheet: Mixer
File: rx_mixer.sch
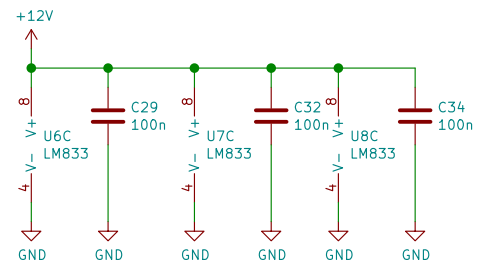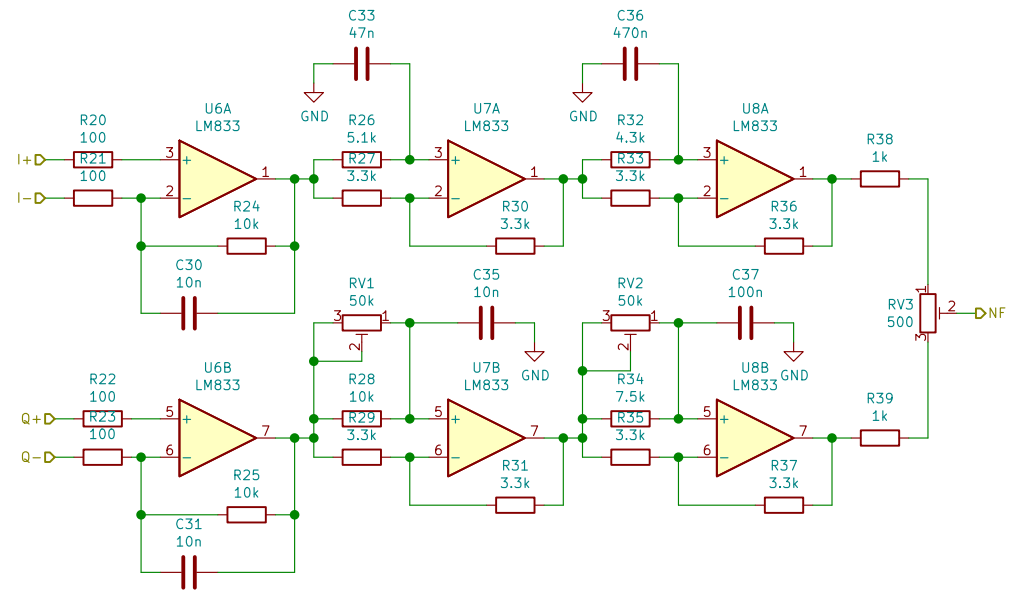
Sheet: Demod
File: rx_demod.sch
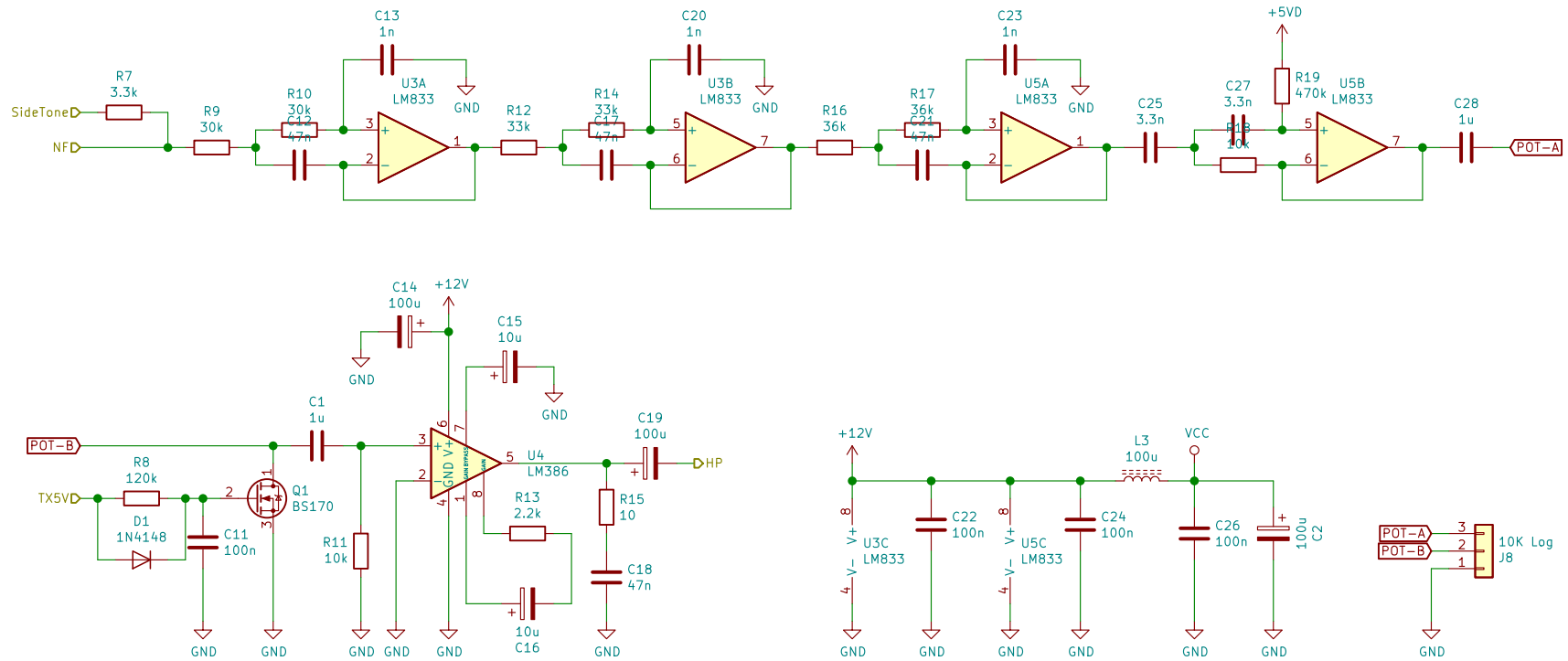
Sheet: Audio
File: rx_audio.sch

Sheet: VFO
File: rx_vfo.sch

Sheet: Ctrl
File: rx_ctrl.sch

J1 RX−RF

J2 HP

J4 ToPA

J11 TxOsc

J7 SW

J6 Supply

U10 L7805

based on QCX
Receiver
DM3MAT
Sheet: /
File: rx.sch
Title: QRP CW TRX
Size: A4   Date: 2020−06−24   Rev: 2
KiCad E.D.A. kicad 5.1.5+dfsg1−2build2   Id: 1/6

T1
3 x 8 @ FT37-43

R2 10k

R1 10k

C3 100n

U1A FST3253

U1B FST3253

U1C FST3253

+5VA

RF

GND

I+
Q-
Q+
I-

1B1 6
1B2 5
1B3 4
1B4 3
1A 7

2B1 10
2B2 11
2B3 12
2B4 13
2A 9

C4 470n
C6 470n
C5 470n
C7 470n

GND GND GND GND

+5VA

C8 100n

VCC 16
S0 14
S1 2
GND 8
nOE1 1
nOE2 15

VFO+I
VFO+Q

GND GND GND GND GND

C13 1n — U3A LM833 — GND
C20 1n — U3B LM833 — GND
C23 1n — U5A LM833 — GND
+5VD — R19 470k — U5B LM833
C27 3.3n

R7 3.3k
SideTone
NF
R9 30k
R10 30k
C12 47n
3 +
2 −
1
R12 33k
R14 33k
C17 47n
5 +
6 −
7
R16 36k
R17 36k
C21 47n
3 +
2 −
1
C25 3.3n
R18 10k
5 +
6 −
7
C28 1u
POT-A

C14 100u — +12V
GND
C15 10u
GND

C1 1u
POT-B
R8 120k
TX5V
D1 1N4148
C11 100n
Q1 BS170
R11 10k
R13 2.2k
3 +
2 −
6
7
5
4
1
8
GND BYPASS GAIN GAIN
V+
U4 LM386
C19 100u
HP
R15 10
C18 47n
10u C16

+12V
V+ 8
U3C LM833
V- 4
C22 100n
V+ 8
U5C LM833
V- 4
C24 100n
L3 100u
VCC
C26 100n
100u C2
GND

POT-A 3
POT-B 2
1
10K Log
J8
GND

GND GND GND GND GND GND GND GND GND GND GND GND GND

+5VD
+5VD

R40
10k

C38
100n

CLK2

U9A
74HC74

2 D
3 C
S 4
R 1
Q 5
Q̄ 6

VFO+I
VFO−I

J9
Si5351
+5VD

1
2
3 SDA
4 SCL
5 CLK2
6 CLK1
7 CLK0

GND

R41
10k

GND

U9B
74HC74

12 D
11 C
S 10
R 13
Q 9
Q̄ 8

VFO+Q
VFO−Q

+5VD

U9C
74HC74

14 VCC
7 GND

C39
100n

GND

GND

based on QCX
PLL + I/Q gen
DM3MAT

Sheet: /VFO/
File: rx_vfo.sch

Title: QRP CW TRX

| Size: A4 | Date: 2020−06−24 | **Rev: 2** |
| KiCad E.D.A. kicad 5.1.5+dfsg1−2build2 | | Id: 5/6 |

Title: QRP CW TRX

based on QCX
Controller
DM3MAT

Sheet: /Ctrl/
File: rx_ctrl.sch

Size: A4 | Date: 2020-06-24 | Rev: 2
KiCad E.D.A. kicad 5.1.5+dfsg1-2build2 | Id: 6/6

based on QCX
DM3MAT

Sheet:
File: rx.kicad_pcb

Title: CW TRX Receiver

| Size: A4 | Date: 2020-06-24 | | Rev: 2 |
| KiCad E.D.A. kicad 5.1.6-c6e7f7d86ubuntu18.04.1 | | | Id: 1/1 |

CW-TRX
— RX —
Rev. 2

Dit GND Key
Dah
J5
J13
TX-OC
Q2
R6
RV3
C2
G +
C26
J7
SW
C3 R1
C30 C33 R26 R27 R32 R33 R9 R10 C12 L3
T1 R24 R30 C36 R36 R38 C43 R7
C4 R20 U6 C29 U7 C32 U8 C34 C20 J3 C22
RX GN C6 R21 R23 R34 R39
U1 R2 R22 C5 R28 R31 R35 R37 C16 R14 R12 R17 R16 C42
C8 L1 C7 R25 R29 C37 C23 C41 C40
USB LSB R11 R1 C21
(c) 2020 C31 RV1 RV2
DM3MAT C35 C14 C18 U4 R13 R49 U5
C39 U9 ISP-6 C52 C27 C25 R16 C28
R40 R41 C1
TX GN R3 C2 R18
J9 C38 R4 C46 R42 R15 C24 C11 D1
Contrast RV4 R5 U11 C19 C15 A D1 U10
R43 C47
LCD1 J2 J10 J8
5V Cont LED GND RS D4 D5 D6 D7 EN HP GND Btn RotB RotA GND GND VOL I NP

based on QCX
DM3MAT
Sheet:
File: rx.kicad_pcb
Title: CW TRX Receiver
Size: A4   Date: 2020-06-24   Rev: 2
KiCad E.D.A. kicad 5.1.6-c6e7f7d86ubuntu18.04.1   Id: 1/1

H1

GND

100n
C3

3 x 8 @ FT37-43
T1

R1 10k

C6

USB 470n

C4

R21 470n

100
R22

R20 R23

U6 LM833

10k R2

100n R2

C8

L1
100u

C7
470n

C5

LSB

J5

Key

Conn_Coaxial

2N3904

Q2

R6 10k

500
RV3

R7 3.3k

C2
100u

C26

100n SW

10n
C30

47n
C33

5.1k 3.3k
R26 R27

C29

4.3k 3.3k
R32 R33

30k
R9

30k
R10

C12
47n

L3
100u

R24 10k

3.3k
R30

C36

100 C32

3.3k
R36

R38 1k

C13

1n

LM833 U3

100n

C22

74HC74
U9

C39
100n

R40 10k

R41 10k

C38 100n

R4
4.7k

AVR-ISP-6

J3

R3 47k

C46
100n

J9 Si5351

RV4
10k

R5
47k

U11 ATmega328-PU

220
R43

LCD

H4

GND

47n

100n

C31

10n

R25 10k
R28

3.3k
R29

50k
RV1

C35

10n

R31 3.3k
R34

3.3k
R35

50k
RV2

C37

R37 3.3k

1k
R39

C20

R14 33k

R12 33k

C23

1n

100n
10k
R11

C16
10u

47n

C18

LM386
U4

R13

R19 470k

LM833 U5

C17

C21

C42 10u

R16 36k

R17 36k

100n 100n

C41

C40

47n

R18 10k

C52

R15 10

1k
R42

C19 100u

C24 100n

C15
10u

3.3n
C27

C25

C28 1u

HP

C14
100u

100u

100n

1u

C1

R8 120k

R11

100n
C11

Q1 BS170

D1 1N4148

A

ROT

10K Log

U10
L7805

C47

100n

L4 L6 L9 L8 L7 L5

C48 C50 C54 C53 C51 C49

100u

100n

H3

H2

FST3253 U1

100
100
100

10k

2.2k

H2

GND

+12V

K1 G6S-2 — RL10M
LP 10m
11T@T37-6 11T@T37-6 11T@T37-6
L1 330n L5 330n L9 330n
C2 100p C6 220p C11 220p C15 100p
GND

J1 Conn_Coaxial
LPF
GND

D1 1n4148 C1 100n
RL10M
Q1 2N3904
R1 10k 10m
R2 100k
GND

K2 G6S-2 — RL20M
LP 17m
13T@T37-6 13T@T37-6 13T@T37-6
L2 480n L6 480n L10 480n
C3 220p C7 470p C12 470p C16 220p
GND

D2 1n4148 C10 100n
RL20M
Q2 2N3904
R3 10k 20m
R4 100k
GND

K3 G6S-2 — RL40M
LP 30m
14T@T37-2 14T@T37-2 14T@T37-2
L3 750n L7 750n L11 750n
C4 470p C8 1n C13 1n C17 470p
GND

D5 1n4148 C27 100n
RL40M
Q10 2N3904
R11 10k 40m
R8 100k
GND

K4 G6S-2 — RL80M
LP 60m
21T@T37-2 21T@T37-2 21T@T37-2
L4 1700n L8 1700n L12 1700n
C5 820p C9 1.5n C14 1.5n C18 820p
GND

D3 1n4148 C19 100n
RL80M
Q3 2N3904
R5 10k 80m
R6 100k
GND

+12V
C29 100n
D6 1n4148 D7 1n4148
RX-HF
GND

R7 10k
R12 100Ω
C30 100n
R16 10k
GND

Q4 BS170
C21 1n
C20 100n
LPF
GND

RX5V
C26 220u C31 100n
GND GND

T1 8+8@FT37-43
Q5 BD140-16
C25 1u
C23 100n
GND

L14 22u VCC
R9 470
R10 1k
Q8 BS170
R15 100Ω R13
KEY5V
R13 10k
GND GND

+5V
4.7k R14

U1A 74HCT00
U1B 74HCT00
U1D 74HCT00
TX5V
TX-HF

Q11 BS170 Q6 BS170 Q7 BS170 Q9 BS170
GND GND GND GND

U1C 74HCT00
TX5V RX5V

+5V
J2 FromRX
TX5V
KEY5V
10m
20m
40m
80m
GND

RX-HF J6 RX-HF
TX-HF J7 TX-HF
GND

+5V
U1E 74HCT00
VCC
C28 100n
GND GND

H1 GND
H2 GND
H3 GND
H4 GND

+12V
L13 100u VCC
J11 RXSupply
C22 100n C24 100n
D4 1N4001
J3 Barrel_Jack_Switch
GND GND GND

CW TRX – PA
– Rev. 2 –
(c) 2020 DM3MAT

12V

RX GND

TX GND
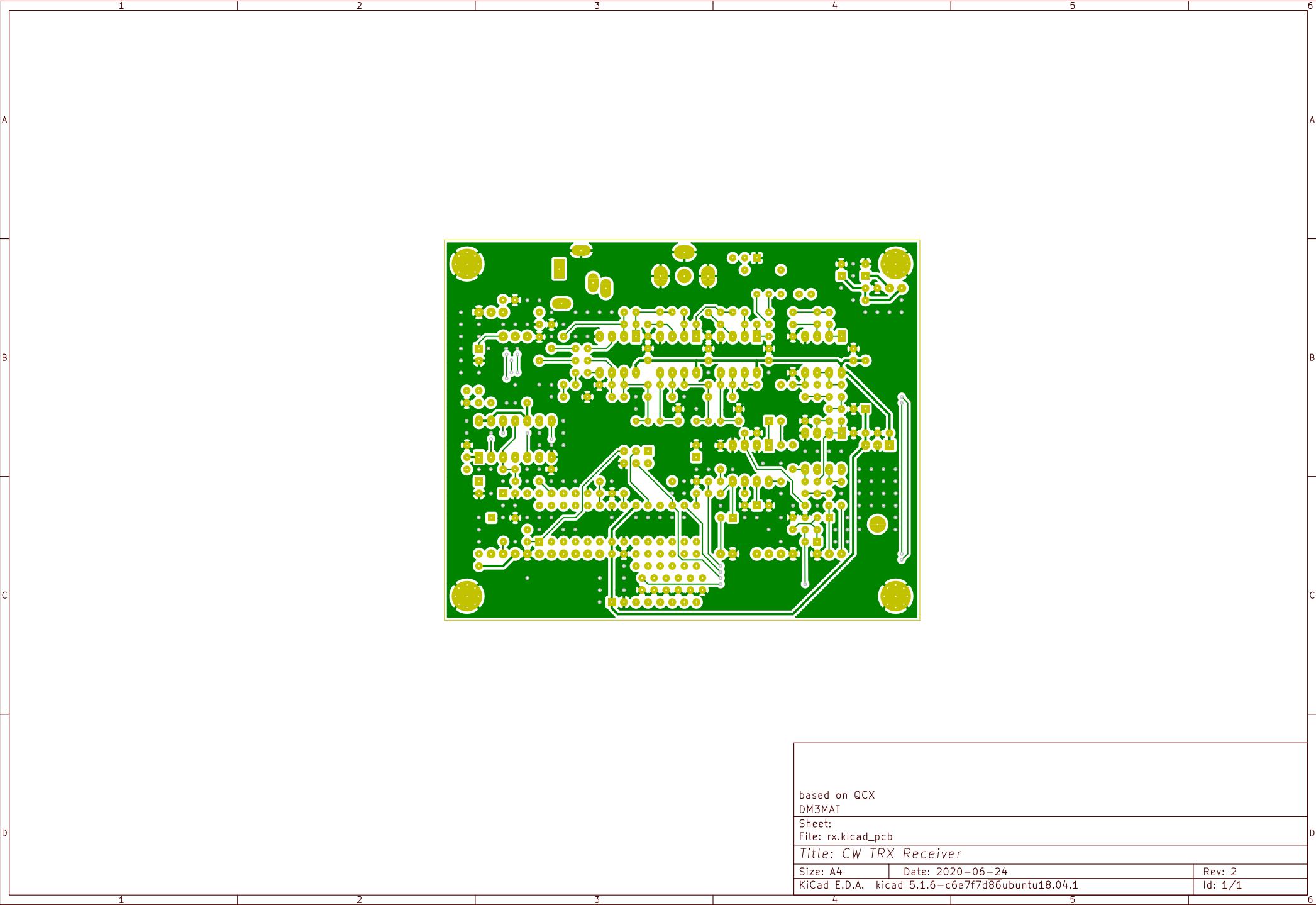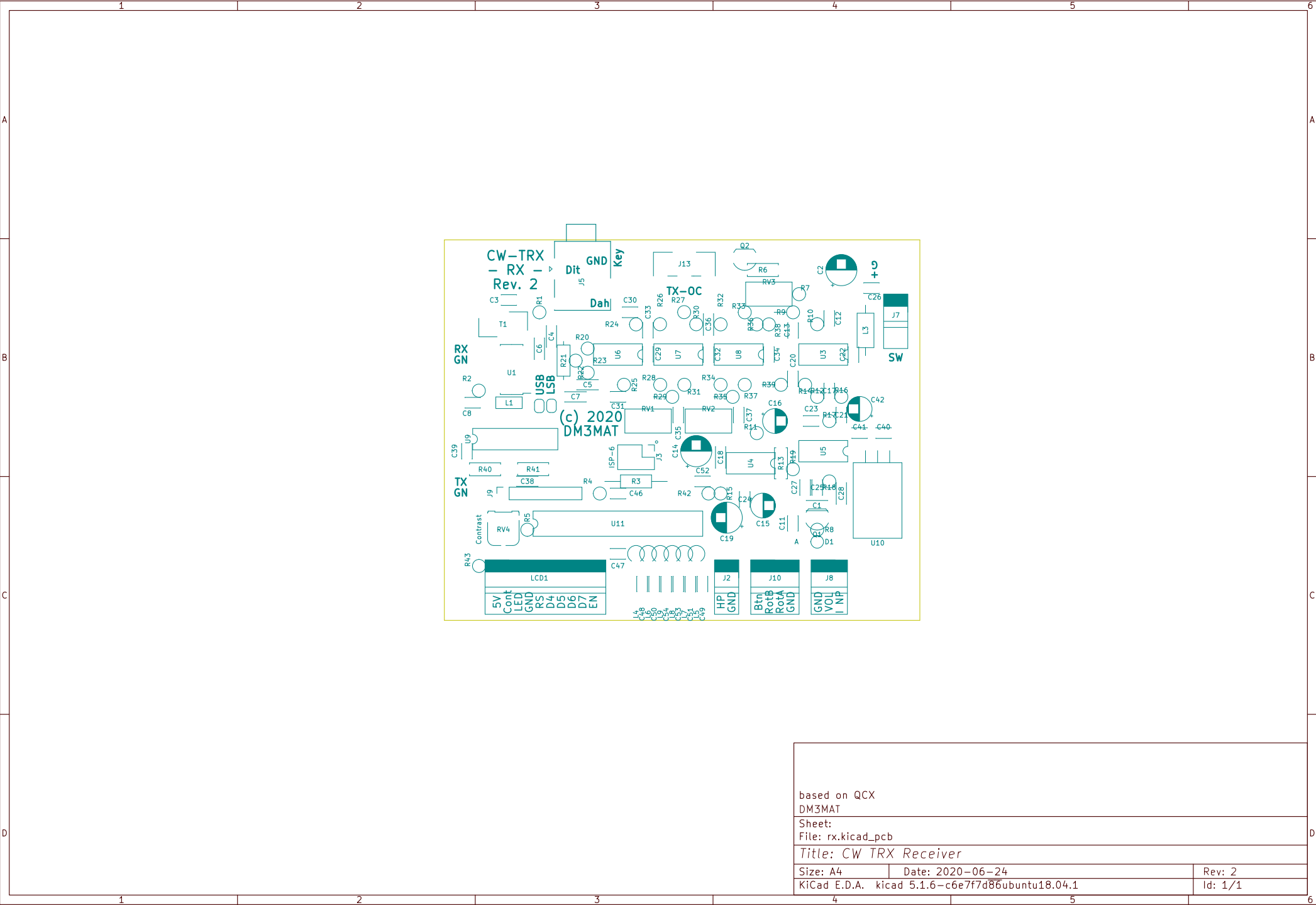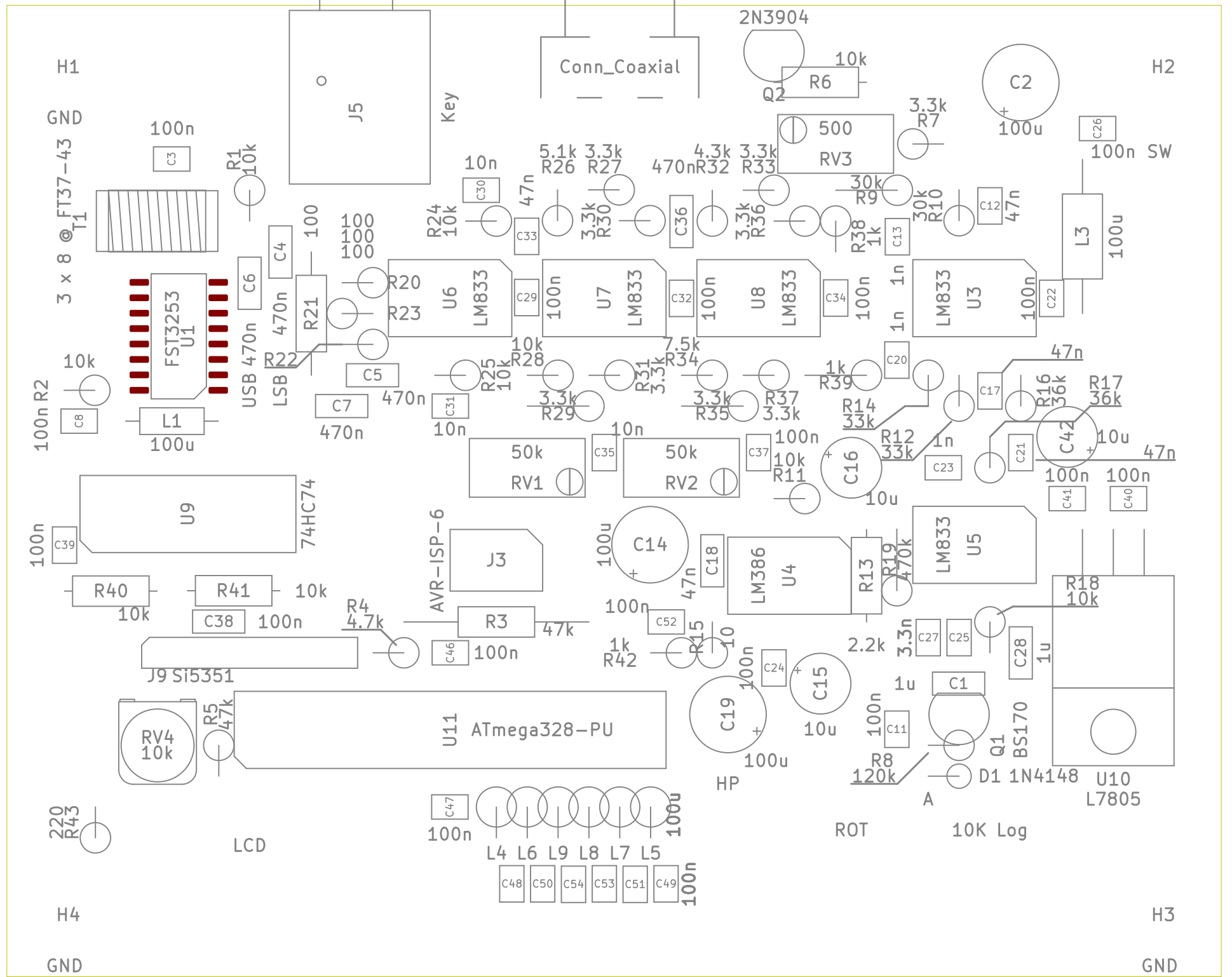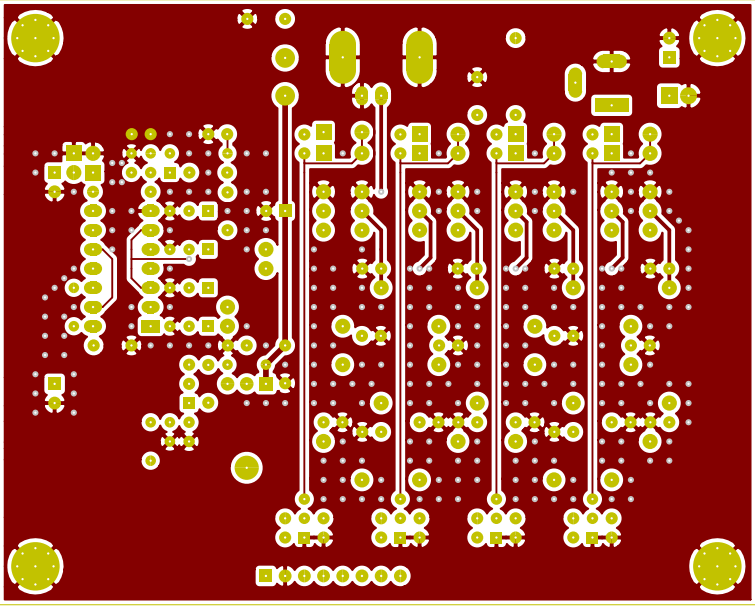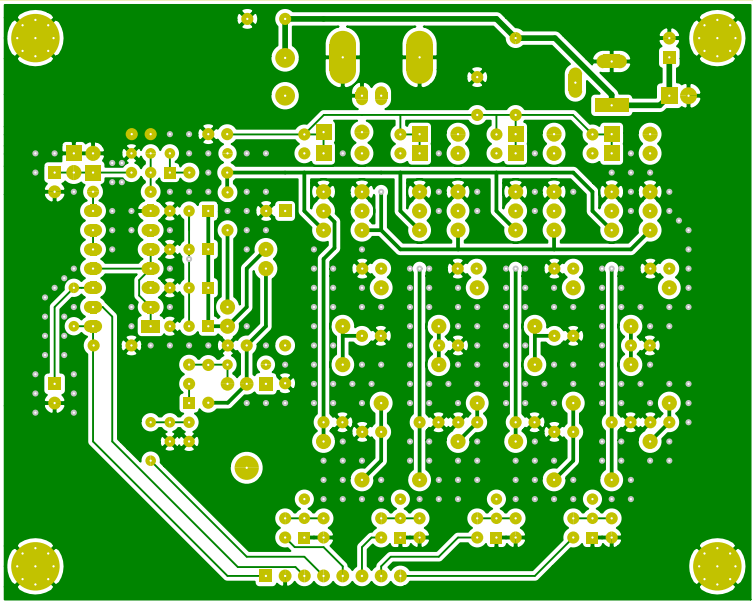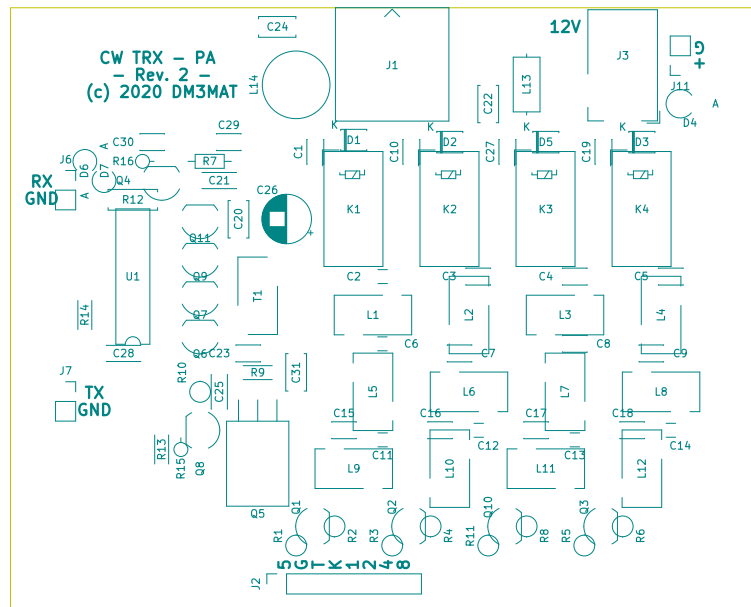
based on QCX
DM3MAT

Sheet:
File: pa.kicad_pcb

Title: CW TRX PA

| Size: A4 | Date: 2020-06-24 | Rev: 2 |
| KiCad E.D.A. kicad 5.1.6-c6e7f7d86ubuntu18.04.1 | | Id: 1/1 |

Barrel_Jack_Switch

H3

GND

C24
100n

J1

RXSupply
J3

J11
D4

H2

GND

A

1N4001

L14
22u

L13
100u

100n C22

D7 1n4148
AD6 1n4148

BS170
Q4

100n

100n C30

R16
10k

10k

R7
10k

C29
100n

C21 1n

K
D1

1n4148
100n

K
D2

1n4148
100n

K
D5

1n4148
100n

K
D3

1n4148

100n C1

C10

C27

C19 100n

J6

A

RX-HF

R12
100

BS170
Q11

C20
100n

220u
C26

K1

G6S-2

K2

G6S-2

K3

G6S-2

K4

G6S-2

74HCT00

U1

Q9

T1

8+8@FT37-43

100p C2

330n L1

C3

C4

220p

470p

750n L3

C5

820p

L4 1700n

4.7k R14

Q7

Q6

100n

470

C23

R9

C31
100n

220p C6

C7

470p

L2 480n

1n C8

L7 750n

C9

1.5n

C28
100n

1k R10

C25

1u

BD140-16
Q5

330n L5

C15

100p

220p

C16

C12

480n L6

220p

470p

C17

470p

L8 1700n

820p

C13

1n

C18

C14

1.5n

J7

TX-HF

100 R13

10k R15

BS170
Q8

C11

220p

L9 330n

L10 480n

470p

L11 750n

L12 1700n

R1 10k

Q1 2N3904

R2 100k

R3 10k

Q2 2N3904

R4 100k

R11
Q10
100k

2N3904
R8

R5 10k

Q3 2N3904

R6 100k

H4

GND

J2 FromRX

H1

GND

Drillplan for front panel using Fisher chassis.
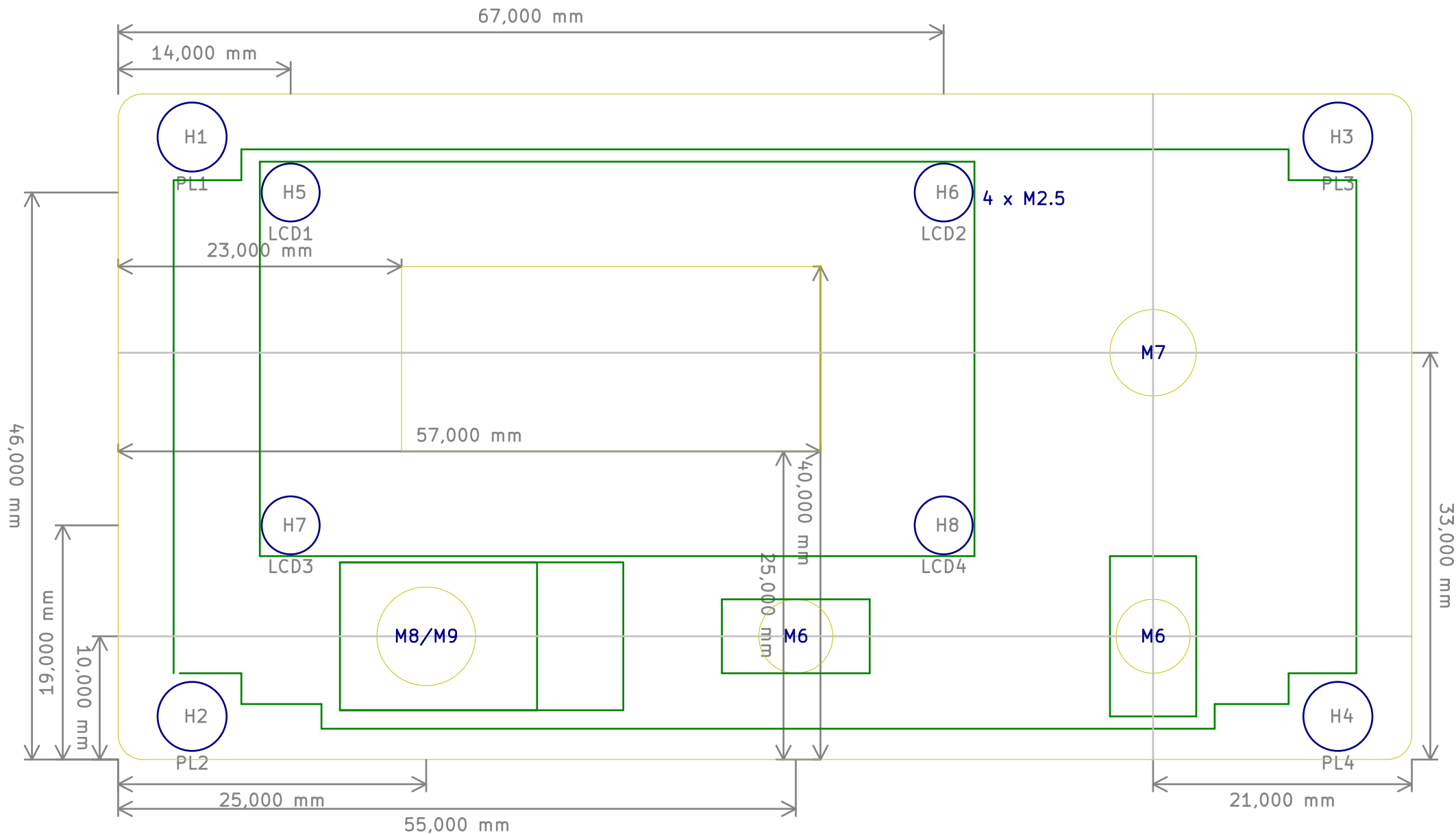
Drillplan for back panel using Fisher chassis.

32,000 mm

KEY

12V ANT GND

6,000 mm

12,000 mm

13,000 mm

6,000 mm

4,000 mm

31,000 mm

18,000 mm

19,000 mm

21,000 mm

12,000 mm

51,000 mm