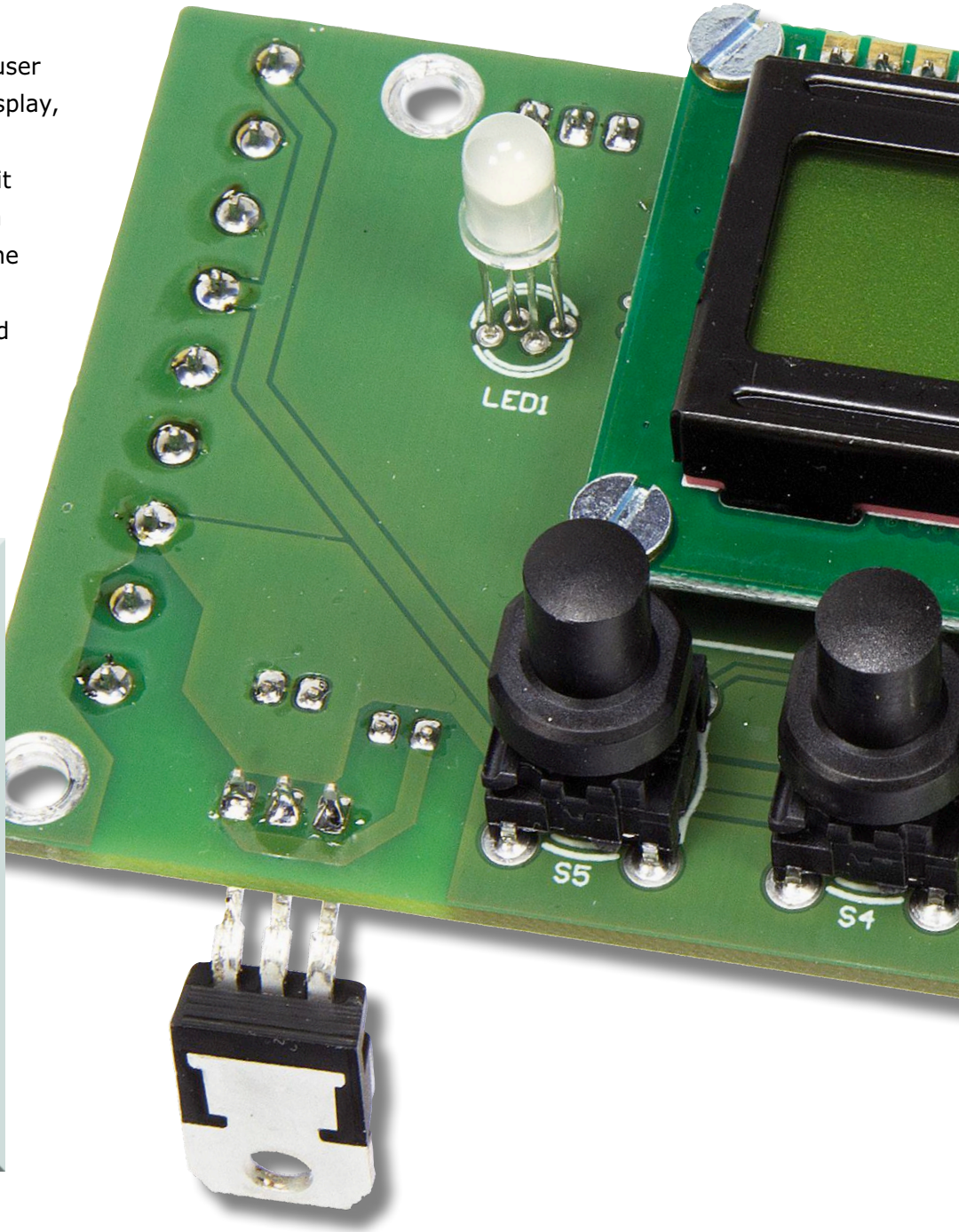
eLABS PROJECT

FPGA-DSP Board for Narrowband SDR

Part 3: microcontroller board

By **Daniel Uppström** (SM6VFZ, Sweden)
and **Ton Giesberts** (Elektor Labs)

Presented in this installment is a user interface (UI) or front end with display, knobs and buttons to control the FPGA-DSP radio. Based on an 8-bit ATmega128A microcontroller from Microchip the board can replace the Raspberry Pi that we used up to now. An audio amplifier is included too, making it possible to listen to the radio without ear-warming headphones.



PROJECT INFORMATION

FPGA DSP SDR
Radio Ham Radio RF
Microcontroller

entry level
intermediate level
➡ expert level

4 hours approx.

SMD soldering tools,
drill press,
metal work tools

£90 / €100 / $110 approx.

Features

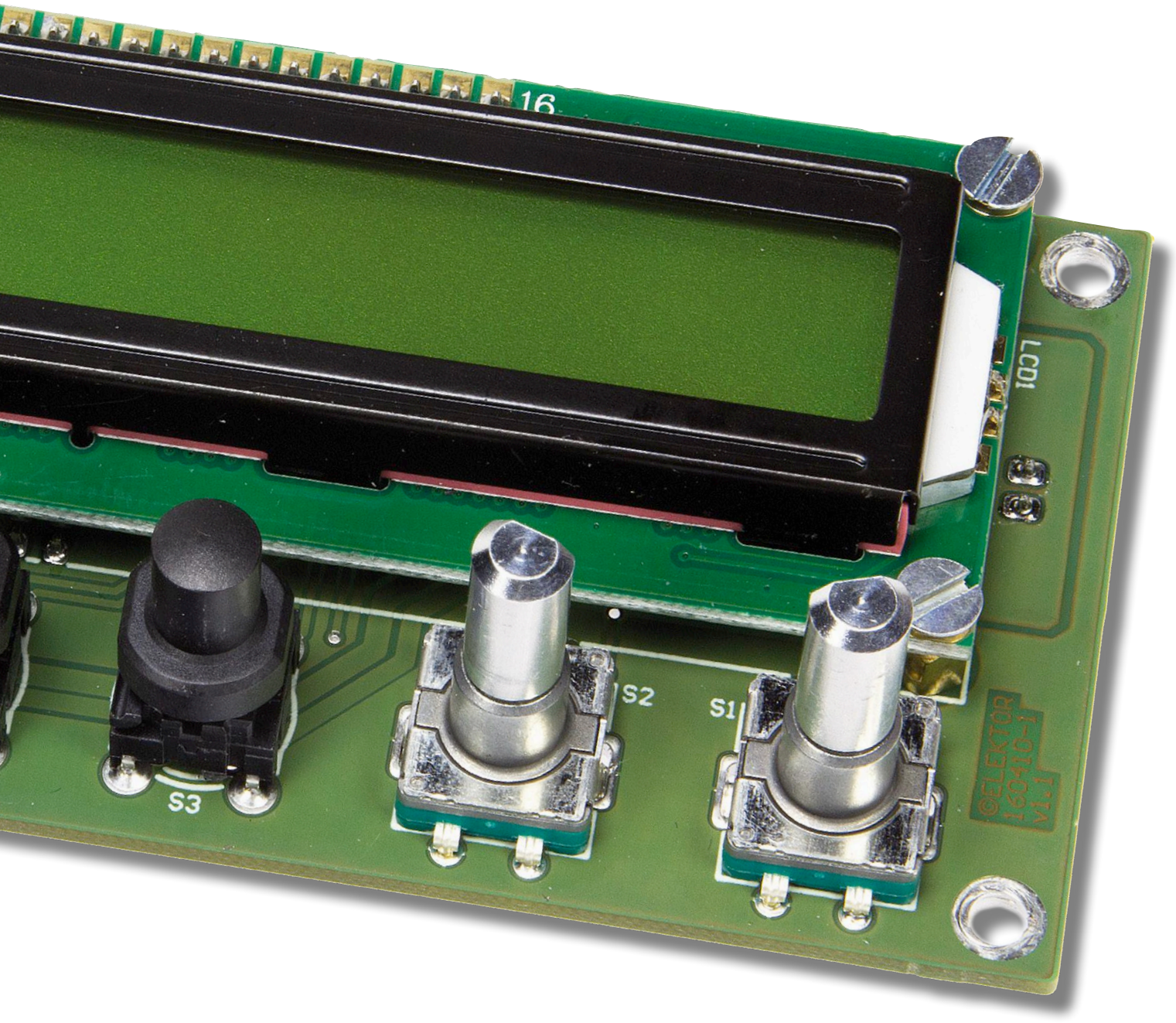
- 2×16 LCD + RGB LED
- 2 rotary encoders, 5 pushbuttons
- Clarity control
- Squelch
- Multifunctional
- On-board audio amplifier

The microcontroller board described in this article was specifically designed as a controller for the FPGA-DSP radio, as well as an alternative to the Raspberry-Pi-based controller. Where an RPi requires additional hardware like a monitor, keyboard and mouse, this MCU board comes with a 2×16 LCD, three pushbuttons, two rotary encoders (with pushbutton), a 2-watt audio power amplifier, a 5-volt voltage regulator and everything else needed to create a stand-alone radio. An (incomplete) overview of the board is shown in **Figure 1**. The versatility of this MCU board allows it to double for many other applications.
Figure 2 shows the complete circuit diagram of the MCU board.

I²C

The MCU board and the FPGA-DSP board communicate over an I²C bus, exposed through K3. When the controller starts up, a few settings are sent to the FPGA to define its basic operation. Further settings like frequency, audio volume and mode of operation (AM, SSB, CW, etc.) can then be controlled by the user. The controller continuously reads status information from the FPGA. This data incudes things like received signal strength (RSSI) and whether the radio is in receive or transmit mode.

I²C is simple, requires few wires, and allows adding more nodes to the same bus. The SDA and SCL ports are of the open-collector/drain type and resistors R2 and R3 pull them up to 5 V. This might seem incompatible with the 3.3 V used for the FPGA I/O's, but the resistors will allow only a few milliampères to flow and the body diodes in the FPGA will ensure non-destructive voltages.
Resistors R4 and R5 and the two ferrite beads L4 & L5 attenuate any high-frequency noise at the interface.
Solder jumper JP1 on the FPGA-DSP board must be left open to select I²C mode (instead of UART mode).



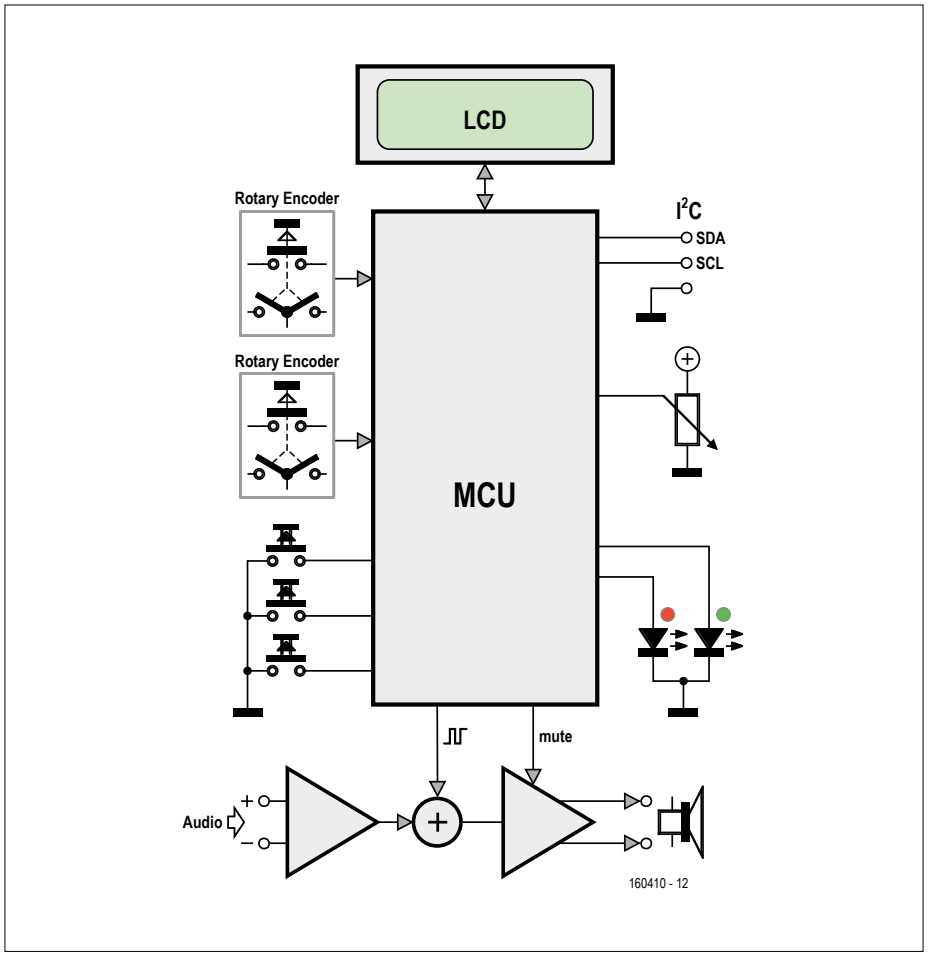


Figure 1. Even though this drawing only shows the main controls and peripherals of the MCU board, it is clear that this board is at ease in many other applications.

Knobs & buttons

Pushbuttons S3, S4 and S5, together with encoders S1 and S2 and the LCD form the main user interface or UI of the radio.

Rotary encoder S1 is intended for tuning the radio's operating frequency. To go faster, press it while turning.

The second rotary encoder, S2, sets the volume. Push and rotate it to set the squelch level. This will mute the radio's audio output when the signal strength drops below a certain level.

Pushbutton S5 allows to quickly step through frequency bands, whereas S4 toggles between the modes AM, LSB, USB, CW and CWN (narrow CW filter).

Pushbutton S3 is intended for accessing a configuration menu, but at the time of writing this function wasn't implemented in the firmware.

The display, LCD1, lets the user see what the current settings are. It is a standard HD44780-compatible alphanumerical LCD with two lines of 16 characters; its contrast is set by trimmer P2. The LCD is used in full 8-bit mode to get the most out of it.

Big-ass tuning knob

When doing serious radio amateur work with lots of tuning up and down the band one quickly gets fed up with a cheap mechanical encoder like S1. To make life more comfortable it is therefore possible to connect a better — but more expensive — optical encoder, with a nice, big knob to get that smooth feeling associated with expensive high-end shortwave transceivers. Not only do these encoders spin better, we also implemented an accelerator for it that makes fast rotations result in larger frequency steps. The Bourns ENA1J-B28-L00128L is a good example of a reasonably priced optical encoder compatible with the MCU board's firmware. Connect such an encoder to K1.

Clarifier

When using the radio for transmitting, it is often desirable to fine-tune the frequency of reception without altering the transmission frequency. This is known as *clarifying* and is achieved with potentiometer P1. When it deviates from its middle position an offset is added to the reception frequency, indicated on the display with a plus or minus sign.

RGB LED

Today no device is complete without an RGB LED, and so there is one on the MCU board too: LED1. However, here it is not used to create a colorful indicator, but simply to show the main operating modes:

1. Transmit (red)
2. Receive (green)
3. Off (blue)
4. Mute (black)

It may seem strange that Off is considered an operating mode, but in this case it is. In fact, LED1 will light blue when the radio is turned off using S6, while the 5 V supply remains available — a kind of stand-by mode. When the radio is on, the LED will be green in receive mode and red otherwise. If the squelch is set and the audio signal is muted, the LED will not light up at all.

Power supply

If a voltage source capable of providing at least 7 V is available, it is possible to mount voltage regulator IC5 to obtain the precious 5 V needed for the rest of the board. If this is done, care should be taken to allow IC5 to dissipate its heat. The source connected to K7 can be turned on and off by S6 mentioned before provided it is of the double-pole variety. K5 and K6 can be used to either supply the board with 5 V or to distribute the 5 V to other boards. Note that K5 is ahead of S6, K6 is behind it, meaning that the latter is directly connected to the rest of the circuit.

Sound & music

The FPGA-DSP board contains an audio CODEC with a class-AB speaker output capable of delivering a maximum output power of 250 mW into an 8-Ω load. Because small speakers usually have low efficiency and 250 mW is often only enough for headphones, a small amplifier was added to the MCU board to provide more power. A class-AB amplifier was used instead of class-D to avoid additional RF noise.

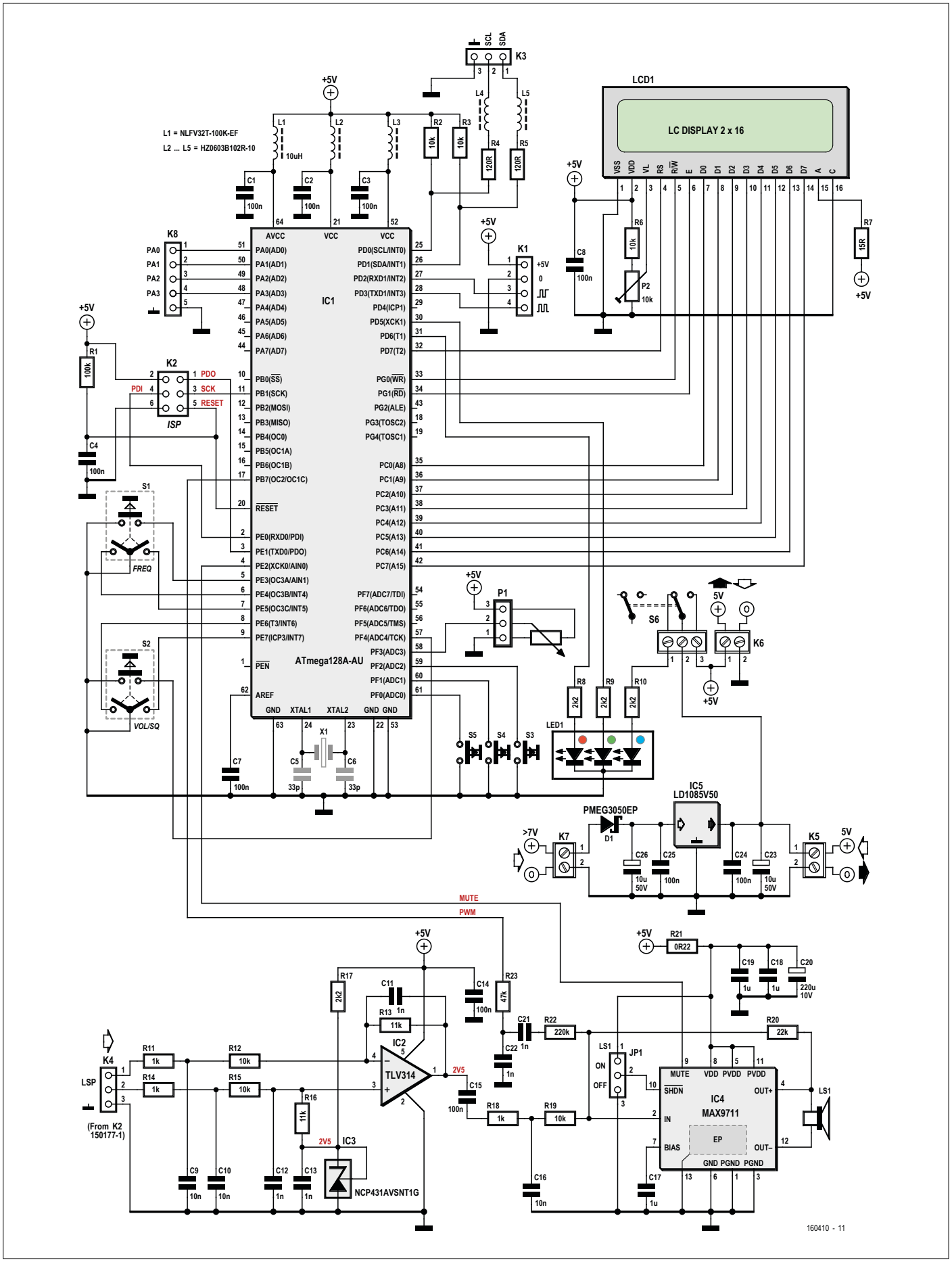


Figure 2. The lower part of this schematic shows the single-channel audio amplifier of the MCU board. IC2 converts the symmetric input signal to asymmetric, whereas IC4 does the inverse. The MCU's PWM output is mixed with the input signal.

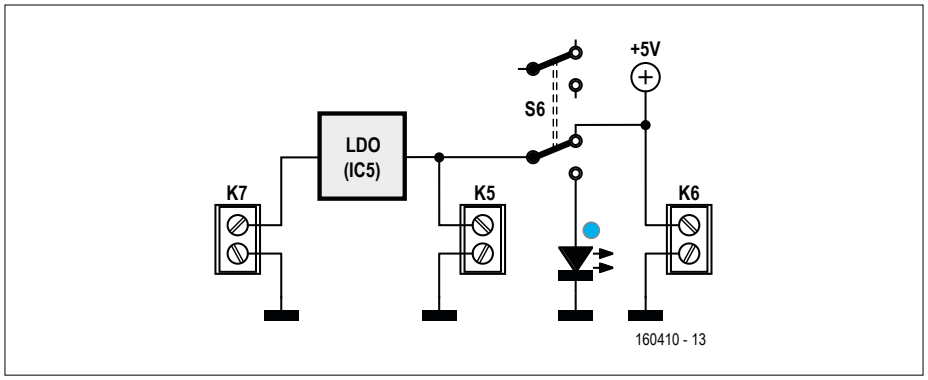


Figure 3. Close-up of the power connections. K7 is an input, K5 is either a 5 V input or output while K6 is a (switched) 5 V output. The second pole of S6 is available for switching some other device or signal, like the source connected to K7.

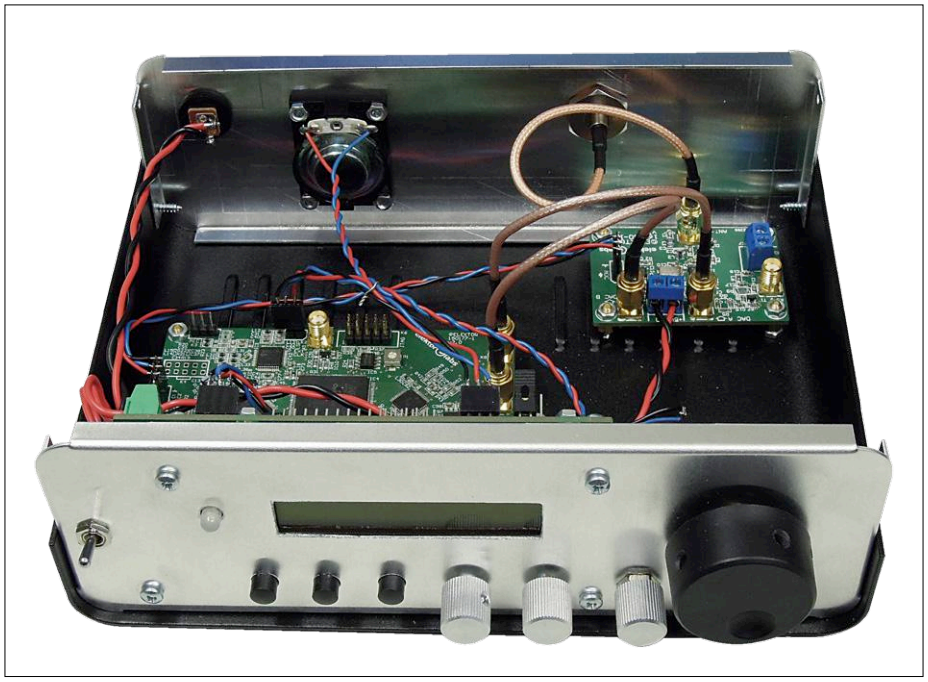


Figure 4. A look inside our finished prototype. Notice how the antenna input cable loops to accommodate its slightly too great length.

For our AF amplifier the MAX9711 was chosen (IC4). It has a differential output stage and its measured output power is close to 2 watts into a 4-Ω load with only a single 5-V power supply. Its input is single ended whereas the FPGA-DSP board has a differential audio output. For this reason IC2 makes the audio signal single-ended before sending it to IC4. It also acts as a low-pass filter with a cutoff frequency of 8.6 kHz, defined by C9 to C12 together with C16. This bandwidth is more than enough for voice; the bandwidth can be extended by decreasing the values of these capacitors. The resistors and capacitors at IC2's input (R11 to R16 and C9 to C12) should have 1% tolerance or better to get the highest common mode suppression. IC3, a 2.5-V reference device, lifts the non-inverting input of IC2 to half the supply voltage.
The feedback of the MAX9711 is inverting allowing the power amplifier to be easily used as an adder for different input signals. This gives the processor a means to generate blips and beeps — as feedback to the user — through a simple low- and high-pass filter added for this purpose, R23/C22 and R22/C21 respectively.
The processor can also mute the audio amplifier, because its port PE2 is connected to the mute input of IC4.
Be careful when connecting a speaker as both output pins carry a signal. Accidentally connecting one of them to ground equals a short circuit and will probably damage the MAX9711 (we didn't try to find out).
To connect K4 to output K2 of the FPGA-DSP board twist/braid the three wires; at the FPGA-DSP board side the ground terminal is not connected.
A very detailed description of the audio amplifier can be found at [4].

And finally, the processor...

Since most of the circuitry has been described by now, there is not much left to say about the microcontroller, the brains and the heart of this MCU board. Quartz crystal X1, together with its load capacitors C5 and C6, can optionally be mounted if an extra-stable clock is required, but for the current firmware it is sufficient to use the RC oscillator you get for free with your ATmega128A. This may also produce less radiation since no clock signals leave the processor (through a pin).
The MCU is programed by a standard AVR ISP adapter connected at K2, while connector K8 exposes some leftover GPIO ports for experimentation and extension purposes.

Some words about the software

The firmware for the MCU board is open source, and can be downloaded from the project web page [1][4], check [6] for the most-recent version. By default the firmware is configured for shortwave amateur radio use with the simple radio board presented in a previous installment, but it can easily be adapted for other frequency ranges or for something else altogether.
Except for the LCD driver, everything is contained in the file main.c while some configuration parameters are available in the file build_settings.h. The code is straightforward and easy to read.
Two timers assure a smooth user experience. Every 100 ms the Band (S5) and Mode (S4) buttons are read together with the clarity control P1, and the RSSI value is updated.
The rotary encoders generate interrupts when they are turned and they are serviced when this happens. A 10-ms timer controls their maximum speed and provides contact debouncing. The tune encoder S1 (ROT1 in the software) functions in parallel with the optional optical

encoder connected to K1 (ROT2 in the software). Since the optical encoder does not have mechanical contacts, it does not need to be debounced, which makes it much more responsive. Fast spinning can now be detected too, and will result in larger frequency steps.

Putting it all together

Since the purpose of the MCU board is to provide a means of building a stand-alone radio, let's have a look at the mechanical side of how this can be done. A suitable and fairly cheap enclosure that fits the FPGA-DSP radio was found at Conrad Electronics, reference GSS03. Its dimensions are 200×150×70 mm. It's a very simple design with identical top and bottom shells made of robust 1.5 mm steel with a few ventilation slots. The top and bottom are screwed to two identical 1 mm thick aluminum front and back plates with four self-tapping screws. There are no additional board support rails. All the boards fit easily in this enclosure. A bigger version of the enclosure might be interesting if future extensions are a possibility. Dimensions of the GSS04 enclosure are 250×200×70 mm, making it possible to mount the speaker on the front.

Cabling

Although a bit too long, two commercially available 15 cm SMA male-to-male RG405 cables can be used to easily connect the RF board to the DAC outputs of the FPGA-DSP board without requiring special tools to attach SMA connectors to shielded cables. Such cables can be found online, for instance on eBay.

For an antenna input for the RF board we used a 15 cm long RG316 BNC female-to-male SMA cable. Since this cable is also longer than strictly needed, we allowed it to loop to the back of the enclosure (**Figure 4**). An external antenna can now be connected through a BNC cable.

The MCU board is mounted on the front panel and placed just far enough to the right to leave sufficient space for the power switch. Potentiometer P1 was placed next to the optical rotary encoder at a distance similar to the one separating S1 from S2.

Use pinheader sockets and thin stranded wires to connect the potentiometer and optical encoder to the corresponding pinheaders on the MCU board (P1 and K1 respectively). Be careful when wiring the optical encoder as connecting its power supply the wrong way around will probably permanently damage it. If the encoder seems to turn in the wrong direction, swap the two data lines.

The connections of the I²C-bus and speaker signals (K3 and K4 of the MCU board to K7 and K2 of the FPGA-DSP board respectively) and the output of the RF board (K4) to the input of the FPGA-DSP board (K1) is also done using sockets and thin stranded wires. Twist or braid all wires. Using sockets for the connections will make potential future changes and/or additions much easier to execute, but feel free to solder all wires directly to the pin headers.

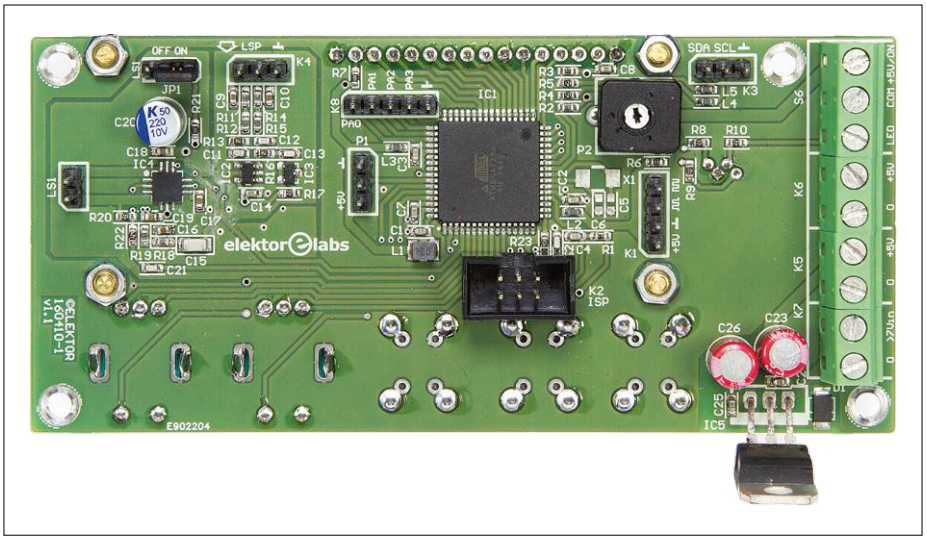


Figure 5. The pins of voltage regulator IC5 are folded so that the IC can be easily fixed to the bottom of the enclosure.

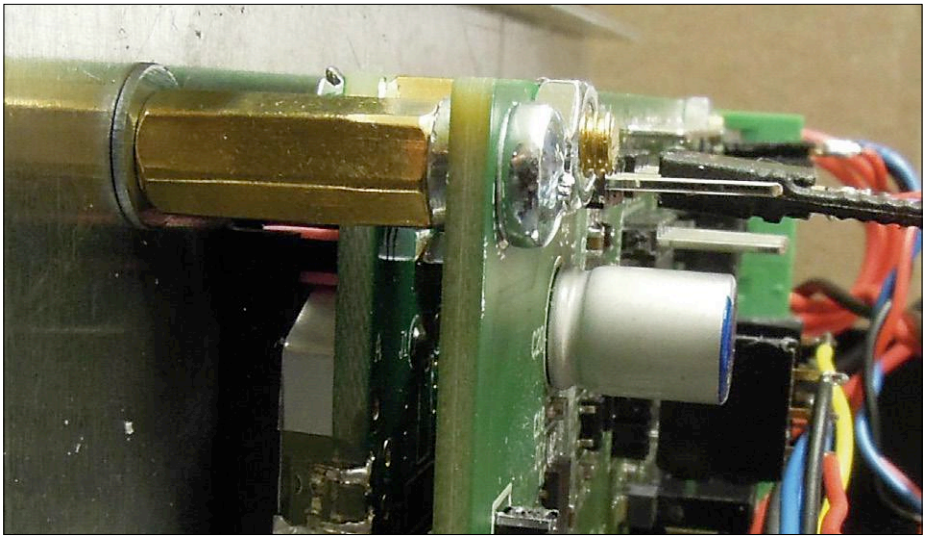


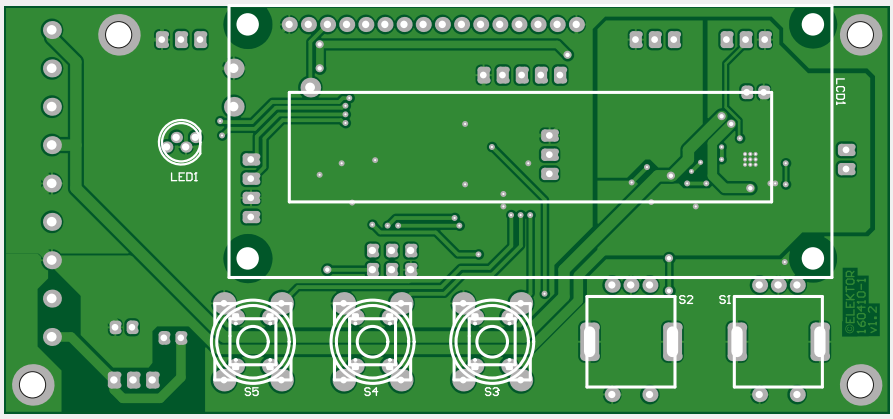
Figure 6. Washers and stand-offs were used to precisely position the MCU board behind the enclosure's front and so create a tight fit for the LCD.

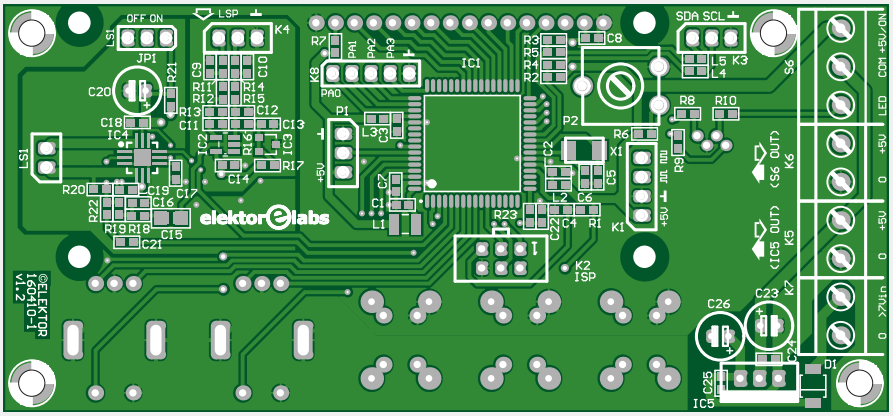
Power supply, part two

In our prototype we used the 5-volt regulator IC5 on the MCU board (**Figure 5**). It is fixed to the bottom of the enclosure with an insulating bush and washer. The bottom of the enclosure is made of steel and conducts heat less efficiently than aluminum. The higher the input voltage, the higher the power loss in IC5, and you might consider adding a real heat sink for it if the input voltage will be more than 8 V.

The DC power input jack on the back is connected with thick (0.75 mm² or more) stranded wires to K7 of the MCU board. Use the same wire for all other power supply connections too. The swit-

COMPONENT LIST





Resistors

Default: 1%, 0.1W, 0603
R1 = 100kΩ
R2,R3,R6,R12,R15,R19 = 10kΩ
R4,R5 = 120Ω
R7 = 15Ω
R8,R9,R10,R17 = 2.2kΩ
R11,R14,R18 = 1kΩ
R13,R16 = 11kΩ
R20 = 22kΩ
R21 = 0.22Ω
R22 = 220kΩ
R23 = 47kΩ
P1 = 10kΩ linear potentiometer with solder lugs
P2 = 10kΩ horizontal trimpot

Capacitors

Default: 0603
C1,C2,C3,C4,C7,C8,C14,C24,C25 = 100nF
C5,C6 = 33pF (optional)
C9,C10 = 10nF, 1%, U2J
C11,C12 = 1nF, 1%
C13,C21,C22 = 1nF
C15 = 100nF, 1206
C16 = 10nF
C17,C18,C19 = 1µF
C20 = 220µF, 10V, 6.3mm diameter, 2 or 2.5mm pitch
C23,C26 = 10µF, 35V, D 6.3 mm diameter, 2 or 2.5 mm pitch

Inductors

L1 = 10µH, 250mA, 0.2Ω, 1210
L2,L3,L4,L5 = ferrite bead, 1kΩ @ 100 MHz, 200mA, 0.6Ω, 0603

Semiconductors

D1 = PMEG3050EP
LED1 = RGB, 5mm, common cathode
IC1 = ATmega128A-AU, programmed
IC2 = TLV314IDBVT
IC3 = NCP431AVSNT1G
IC4 = MAX9711ETC+
IC5 = LD1085V50, TO-220

Miscellaneous

K1 = 4-pin SIL pinheader, vertical, 0.1" pitch
K2 = 6-pin (2x3) pinheader, vertical, 0.1" pitch
K3,K4,JP1 = 3-pin SIL pinheader, vertical, 0.1" pitch
K5,K6,K7 = 2-way terminal block 630V, 0.2" pitch
K8 = 5-pin SIL pinheader, vertical, 0.1" pitch
LCD1 = HD44780-compatible, 2x16 character LCD
LS1 = mini speaker, 4W, 4Ω
S1,S2 = rotary encoder + push-on switch, Alps EC11E183440C
S3,S4,S5 = pushbutton, Multimec RA3FTH9
S6 = toggle switch, DPDT, 28VDC, 5A
X1 = 8MHz quartz crystal, 18pF, 5x3.2 mm (optional)

For IC5: Insulating kit, thermal pad TO-220 + bush TO-220
For IC5: M3 screw, nut, 2 pcs. washer
For JP1: shunt jumper, 0.1" pitch
For K1: 4-way SIL pinheader socket, vertical, 0.1" pitch
For K3,K4,P1: 3-way pinheader socket, vertical, 0.1" pitch
For K8: 5-way SIL pinheader socket, vertical, 0.1" pitch
For LCD1: 16-pin SIL pinheader, vertical, 0.1" pitch
For LS1: 2-pin pinheader, vertical, 0.1" pitch
For LS1: 2-way pinheader socket, vertical, 0.1" pitch
For P1: 3-way pinheader, vertical, 0.1" pitch
For S1,S2,P1: Knob, round, 12mm, shaft 6mm
For S3,S4,S5: Cap, black 19mm, Multimec 1S09-19.0
For S6: 3-way PCB screw terminal block 630V, 0.2" pitch
PCB # 160410-1 from Elektor Store

ched 5-volt power supply output K6 of the MCU board is connected to K10 of the FPGA-DSP board and from there connected to K6 of the RF Board. Screw terminal blocks have been used for power supply connections on all boards for a sturdy and secure connection.

If an external 5-volt power supply is used, the onboard voltage regulator can be omitted along with D1 and C23 to C26, and K5 is the input (also see Figure 3). Be careful, there is no polarity protection here! The 5 V on K5 is switched by S6. If this is unwanted, then K6 can act as the 5 V power supply input, but this is less practical as K6 is also supposed to be the distribution point of the supply voltage to the other PCBs.

Audio

The miniature speaker is placed on the rear panel and connected by two wires to connector LS1 on the MCU Board. The speaker can also be mounted on either the bottom or top shell, whatever you prefer. If nothing is going to be placed on top of the enclosure, putting the speaker there would give a more direct sound.

The small speaker we used (ABS-230-RC) is small and easy to mount. However, its low-frequency response could be better; at 300 Hz the loss of sound pressure is already noticeable. Feel free to use a bigger 4 Ω speaker (2 W nominal minimum). Enough space is left on the front to add a 3.5-mm (maybe even a 6.3-mm) headphone jack. Connect left and right together if you have a stereo connector. If it has a switch, use it to disable the speaker. Put a resistor in series (start with 100 Ω or so) with the headphone output to limit the maximum output power and decouple any capacitive load of the cable.

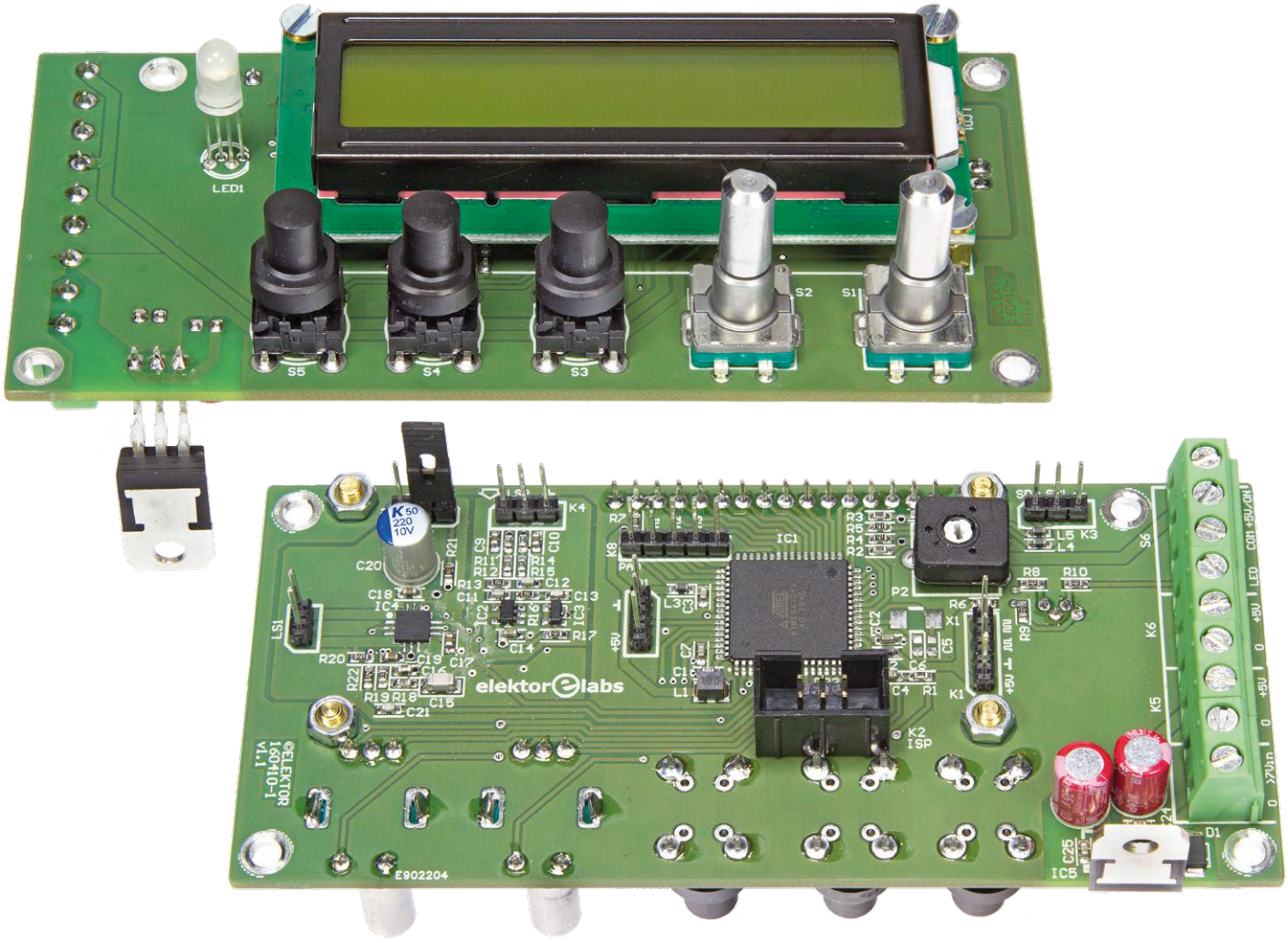
Mounting the LCD

Use a 1:1 printout of the bottom side component print of the MCU board as a drill template. (Be careful, a 1:1 laser print isn't always exactly 1:1.) The viewing area of the LCD is also printed on the bottom overlay, but it is for the (standard) module we have in our Store (120061-74/SKU 16414). LCD modules from other manufacturers may be slightly different. Always look at the datasheet for the exact location and size of the viewing area, and check that the pinout is the same.

The MCU board is mounted on four 12 mm M3 female-female standoffs extended with an M3 washer and locking ring (**Figure 6**). Together they place the board at exactly 13.4 mm behind the front plate, leaving just enough room to mount our LCD module 5 mm above the board. Again, if a module from another manufacturer is used this distance may have to be adjusted.

The LCD itself is fixed to the MCU board with four 5 mm M3 male-female standoffs and four short 4 mm screws. ⏮

(160410)



Web Links

[1] www.elektormagazine.com/160410

[2] www.elektormagazine.com/150177

[3] www.elektormagazine.com/160160

[4] www.elektormagazine.com/labs/microcontroller-board-for-fpga-dsp-radio-160410

[5] Author's blog: sm6vfz.wordpress.com

[6] For the latest firmware: https://github.com/danupp/radiocontrol-mega128