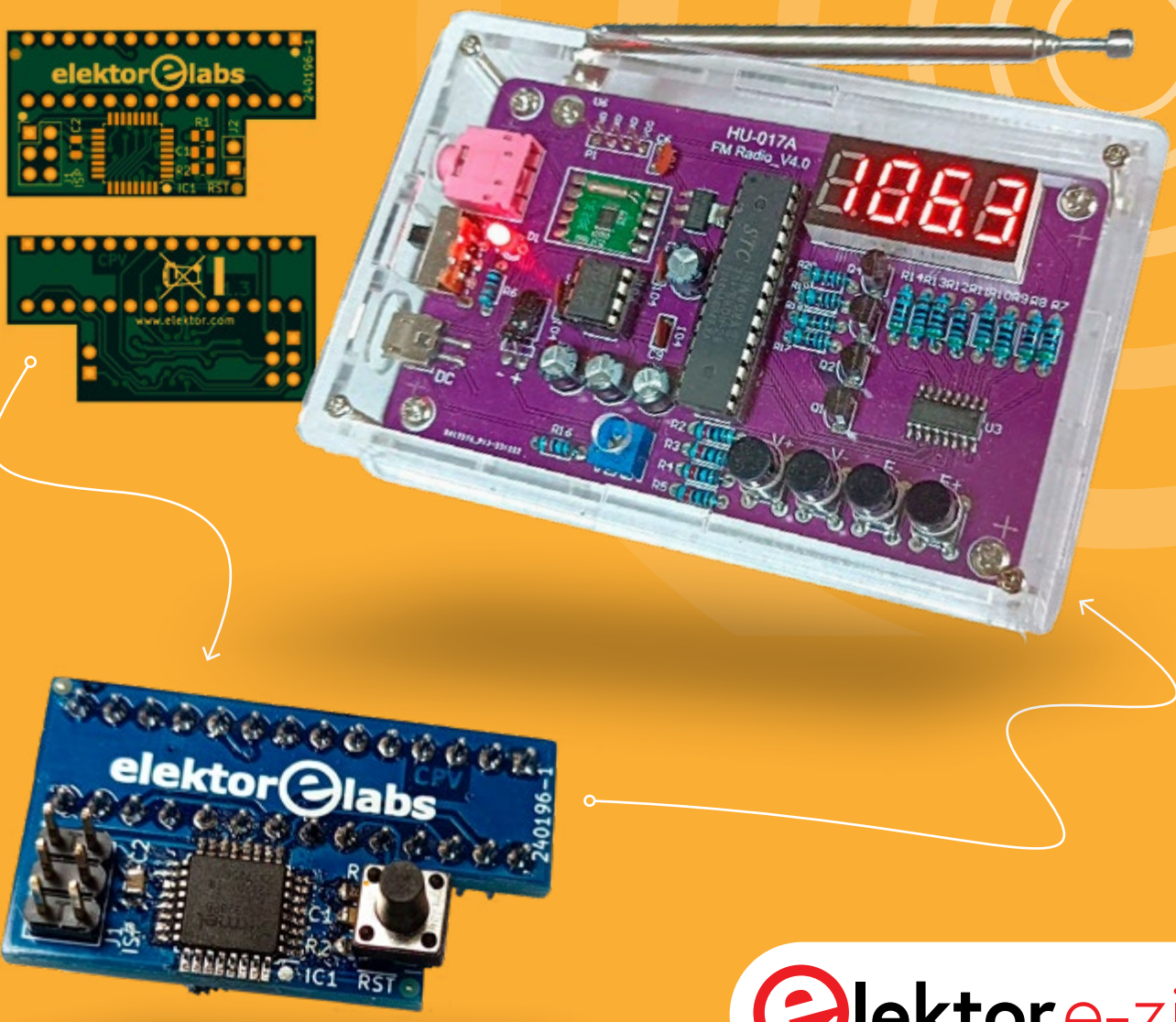
reverse project #01
an e-zine exclusive
June 2024

NEW

MODDING AN RDA5807-BASED FM RADIO KIT

Add RDS, RSSI, Bass Boost, and More



elektor e-zine
design > share > earn

SUBSCRIBE NOW AT
www.elektormagazine.com/elektor-newsletter

Join the Elektor Community

Take out a elektor GOLD membership!

- The Elektor web archive from 1974!
- 8x Elektor Magazine (print)
- 8x Elektor Magazine (digital)
- 10% discount in our web shop and exclusive offers
- Access to more than 5000 Gerber files

Also available

The Digital elektor GREEN membership!

- The Elektor web archive from 1974!
- 8x Elektor Magazine (digital)
- 10% discount in our web shop and exclusive offers
- Access to more than 5000 Gerber files

www.elektormagazine.com/Member

elektor design > share > earn

reverse project #01

an *e-zine* exclusive

June 2024

Dear readers,

We are pleased to announce a new initiative that is exclusively available to our *e-zine* readers: 'reverse projects.' Starting from this issue, we will offer an exclusive article each month that delves into a fun and interesting kit we have sourced and approached as if it were an in-house Elektor project.

For our first project, Elektor engineer Clemens Valens has examined an affordable FM radio kit. Clemens thoroughly analyzes the provided hardware and shares his professional opinion on the build quality and the opportunities the manufacturer missed. He has even taken it a step further by developing a small modification: an additional board that significantly enhances the kit's functionality.

This is a typical Elektor approach: extracting more value from existing products through critical analysis and creative solutions. The Gerber files and all necessary information for this modification are available for free as a small project on the Elektor LABS project platform (*www.elektor-labs.com*), so you can easily make it yourself. Additionally, we offer the option to join an 'Elektor Jumpstarter campaign.' If there are 100 or more interested participants, Elektor will produce a small series of these boards, which will be available for a limited price in our stores. The original kit is, of course, available in the *Elektor Store*.

We wish everyone a lot of enjoyment with this first edition of 'reverse projects.' The next one is already in the works! The criteria for this series are simple: popular, affordable kits that are readily available online and provide a fun project for an evening or a weekend. By analyzing the manufacturers' choices, we collectively learn more about the design and potential of these kits.

Our editorial team and store team are always open to suggestions and feedback. We look forward to hearing from you in the comments or on the online LAB.

Kind regards,

CJ Abate
Content Director, Elektor

Modding an RDA5807-Based FM Radio Kit

Add RDS, RSSI, Bass Boost, and More

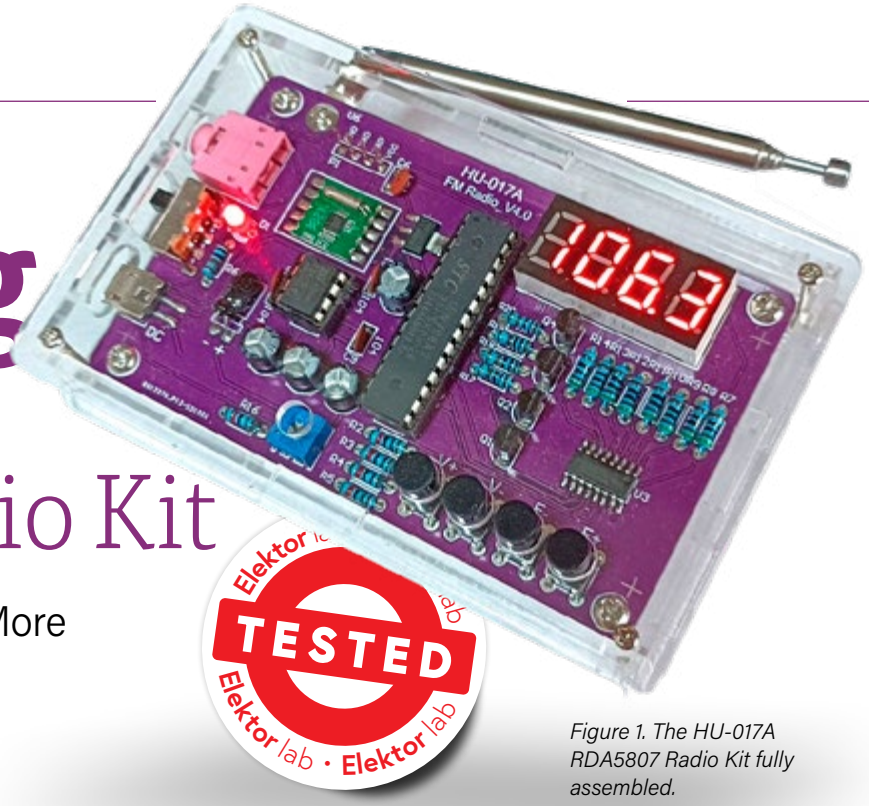


Figure 1. The HU-017A RDA5807 Radio Kit fully assembled.

By Clemens Valens

Building a radio from a kit is fun, but what once it is assembled? For many electronics enthusiasts, the real fun only begins when you can mod it. In this article, we present a cheap FM radio kit and a way to turn it into an Arduino-compatible, user-programmable device.

On the internet, you can find cheap FM radio receiver kits based on the RDA5807 "single-chip FM broadcast stereo radio tuner" by RDA Microelectronics. This IC contains everything you need to build an FM broadcast receiver, including a stereo headphone output. The only thing missing is a user interface that you are supposed to add yourself. And that is precisely what these kits do — they add a microcontroller, display, buttons, antenna, and an audio amplifier to the RDA5807, which itself comes mounted on a small breakout board.

Once assembled, you have a nice little portable battery-powered FM radio (**Figure 1**). The sound quality of the built-in speaker is rather painful, but it's excellent with headphones. Unfortunately, the kits run closed-source software on an STC15W408AS (an enhanced 8051) and expansion ports are not provided.

Limited Functionality

As far as I know, the only functions the radio kit offers are frequency up and down, and volume up and down. This is a pity because the RDA5807 features more functions, notably RDS (Radio Data System [1]) and bass boost. To make these accessible, you must either change the firmware of the provided MCU or connect another controller to the RDA5807's I²C port. A third way is to replace the MCU altogether, which is what I did. But, before we dive into that, let me first describe the kit in some more detail.

The HU-017A RDA5807 Radio Kit

The kit I used is more or less unbranded and answers to the name HU-017A. It consists of a purple PCB of 6 cm by 9.5 cm, a bag of components, a short (25 cm) telescopic antenna and a laser-cut acrylic enclosure.

The components are mainly through-hole parts, but not all. I assume for size reasons, U3, a 74HC595 shift register, is an SMT device. Voltage regulator U10 is also an SMD. The RDA5807 is a tiny chip mounted on a little board with castellated contacts, so, strictly speaking, this is also an SMD. The SMT micro-USB-B power connector only has two pins (no data pins), and it is easy to solder. All other parts are THT devices.

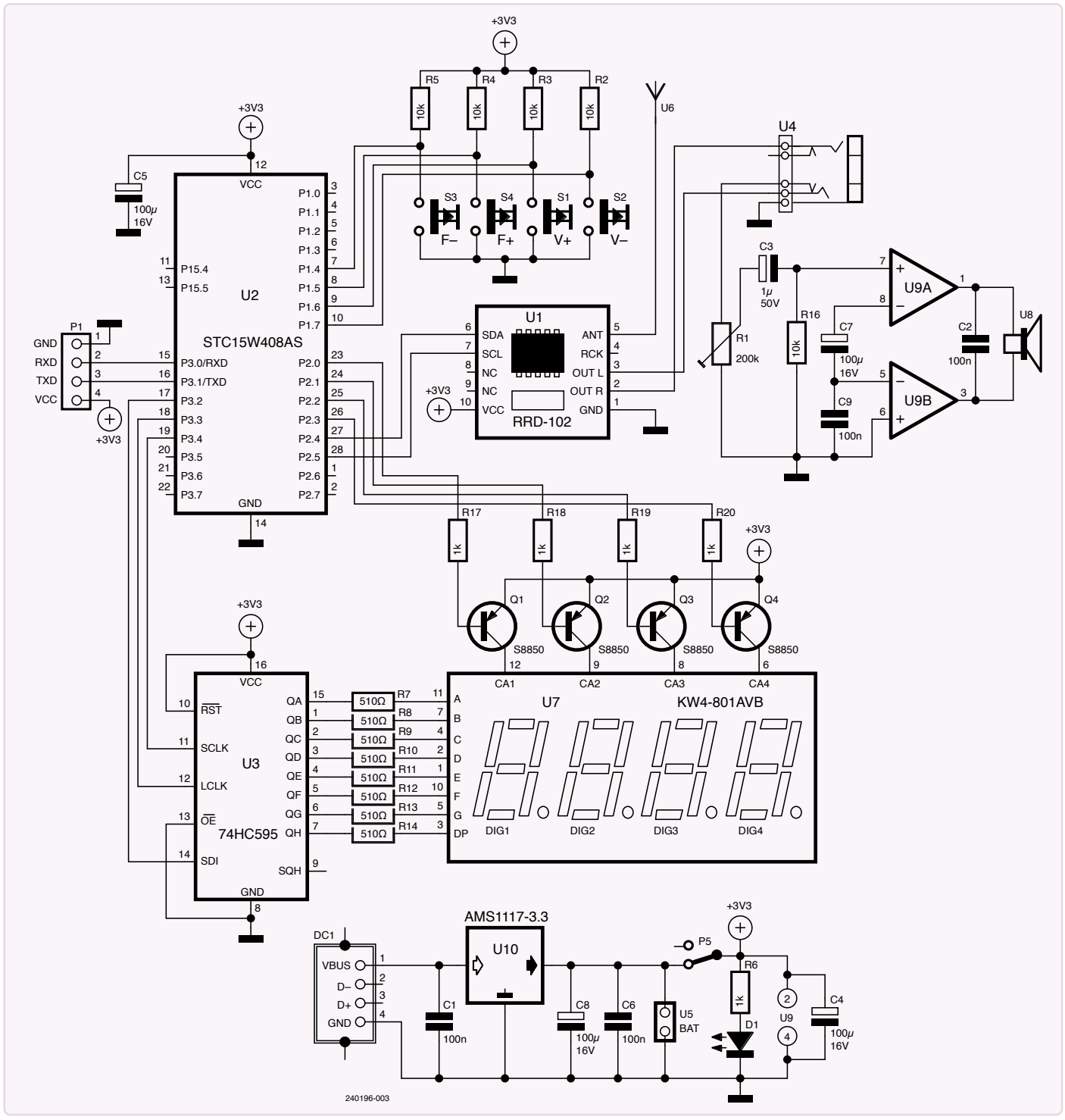
The Circuit

The schematic of the radio is printed in the user manual, and we have redrawn it in **Figure 2**. It shows four push buttons connected to a microcontroller that drives a 4-digit common-anode seven-segment display one digit at a time (i.e., multiplexing) through the aforementioned 74HC595 shift register (U3) with the help of transistors Q1 to Q4.

The MCU communicates with U1, the radio chip, over an I²C bus.

Audio Output

U9 is an audio amplifier in bridge configuration driving an 8 Ω loudspeaker. Its input signal is the right audio



▲

Figure 2. Schematic of the HU-017A RDA5807 Radio Kit.

output channel from the RDA5807. U4, a 3.5 mm socket, is for connecting headphones. When you do that, you will hear the stereo audio signal, which is of much better quality than that of the small speaker. The volume is controlled by two pushbuttons, V– and V+, that control the volume register inside the RDA5807. Trimmer R1 sets the maximum volume level of the amplifier, not for the headphones.

Voltage regulator U10 turns an external input voltage into 3.3 V. It supports up to 15 V at its input, but there is no reverse-polarity protection. Micro USB-B connector DC1 sort of provides this, but only if you use a ready-made phone-charger-like power adapter.

Battery Warning

Make sure to remove the batteries when the radio is powered from an external power supply, as there is no protection for the batteries. If you forget this, the batteries will be damaged and start leaking (or worse).

Pin header P1 exposes a serial port, but I did not observe any activity on it. If you have the right tools, you can use it for in-system programming (ISP) of the STC15W408AS.

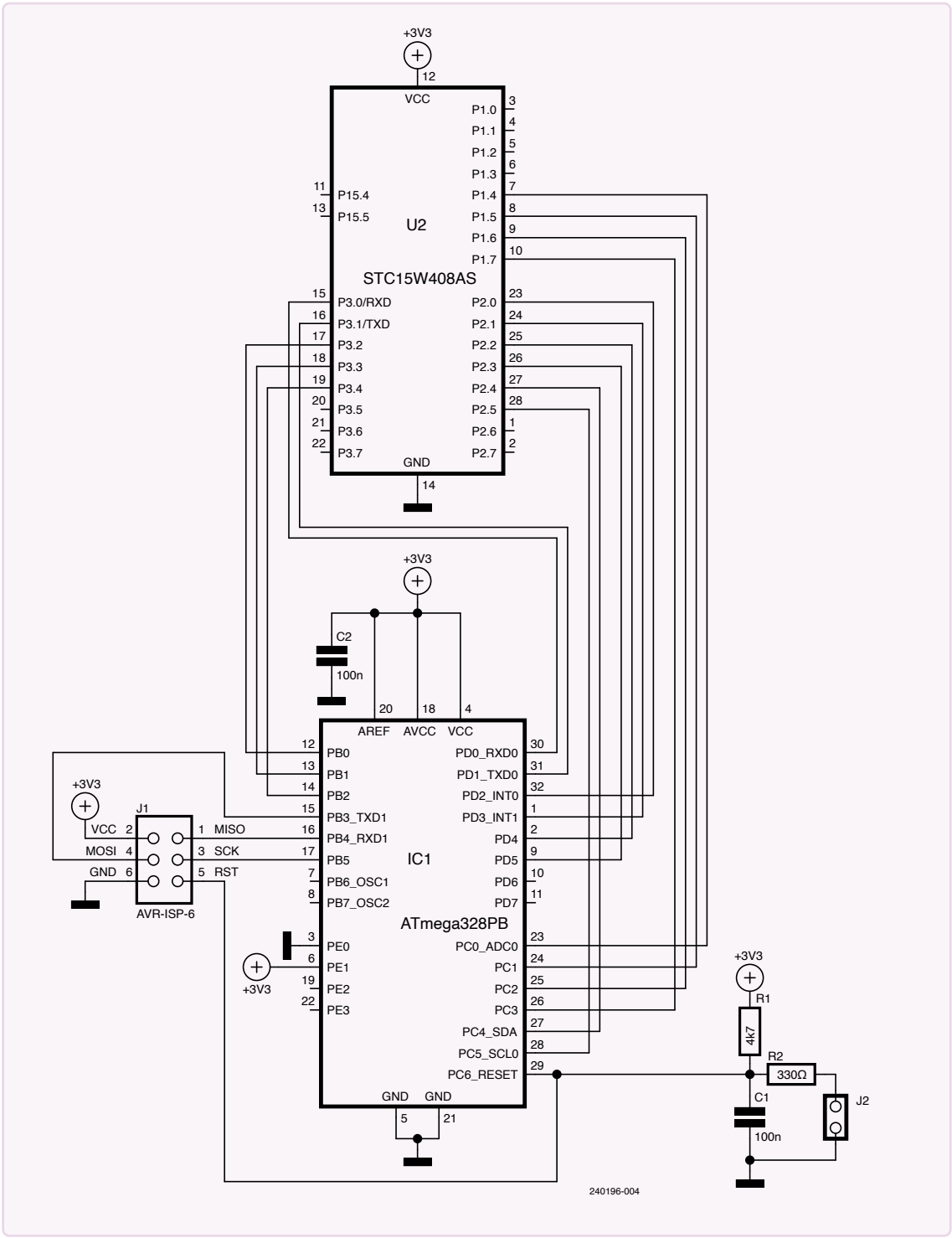


Figure 3. Schematic of the ATmega328PB-to-STC15W408AS adapter board. Note that with a suitable bootloader you can also use an ATmega328P in a 32-pin TQFP package (the "MA" version).

Tune to Arduino

As mentioned, the RDA5807 can do more than the kit allows you to. There exist good libraries to control the chip, notably the Arduino library by Ricardo Lima Caratti [2]. Furthermore, the 74HC595 that drives the 7-segment display is well-known in the Arduino world, so writing new software for the radio using Arduino is relatively straightforward.

Replace the Microcontroller

The STC15W408AS is a 28-pin device, just like the ATMega328 that we know from the Arduino UNO board. Both MCUs are similar, and we can replace one with the other. They are not pin compatible, so a bit of rewiring has to be done to connect everything together.

Note that the ATmega328 is no longer recommended for new designs. Microchip, the manufacturer of the chip, suggests using the ATmega328PB instead (something Elektor already did in 2016, see [3]). The "PB" does not exist in a 28-pin DIP package, but only comes in 32-pin SMT packages. Since rewiring has to be done anyway, this is not really a problem. Actually, it is good, as it keeps the adapter height low.

ATmega328PB Adapter Board

Figure 3 shows how I connected the ATmega328PB to the footprint of the original MCU. There exist many libraries for controlling the 74HC595 over SPI, but I preferred to reserve the SPI bus for MCU ISP programming (J1). J2 is available for connecting a reset pushbutton, which is practical once the new MCU contains an Arduino bootloader, as there is no automatic reset possible (without adding ugly wires) (**Figure 4**).

I designed a small PCB for the adapter that fits in the space delimited by C5 and C8 on the left and Q1 to Q4 on the right — see **Figure 5**.

New Software

The new software should have at least the same functionality as the original software, meaning volume up and down and frequency up and down by pressing the corresponding keys. As there are only four keys, they must also be used to access any other function that might be implemented. Since all possible single-key presses are already used, another mechanism is required. I chose for a long press of the F+ key to access other modes of operation.

Display Driver

The Arduino `shiftOut()` function is used to print characters on the seven-segment display. The digits are multiplexed, meaning that only one can be active at any time. This implies that the display must be refreshed continuously for displaying multi-digit values and strings. This is best done by a timer task running in the background, which means that a timer is required.

TimerOne

Since a timer is needed, I opted for Timer1 as Timer0 is used for the Arduino functions `millis()` and `delay()`, which are commonly used by Arduino libraries, and so it is better not to fool with it. `TimerOne` is a neat little library for using Timer1 on Arduino.

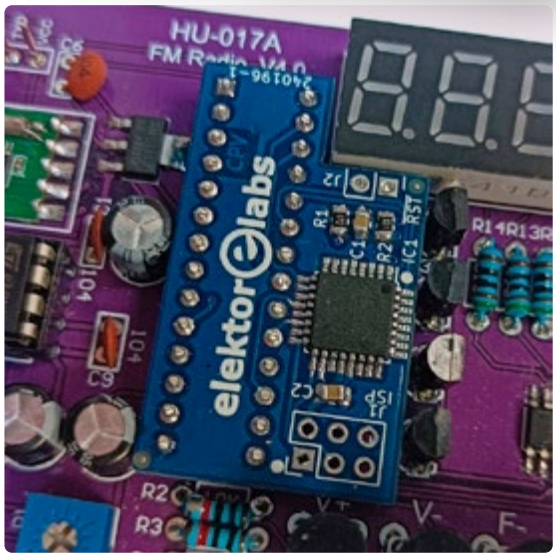
I set up Timer1 for a rate of 1 kHz. Every millisecond, it calls the function `my_millis_counter()`, which, in turn, calls `display_refresh()`. Timer1 also counts milliseconds for the pushbutton scanner. Therefore, display refresh and key scanning are synchronized. This ensures a non-flickering display and responsive, debounced keys.

Key Scanning and Debouncing

Key scanning and debouncing is implemented using a simple algorithm. When a key press is detected (an event), the event start time is recorded. When the key is released, the event end time is recorded. If the period between the start and end time is too short, it is considered a bounce, and the key is reset. If it is longer than a bounce, but shorter than a long press, it is a normal press. Anything else is a long press. Several keys can be pressed simultaneously.

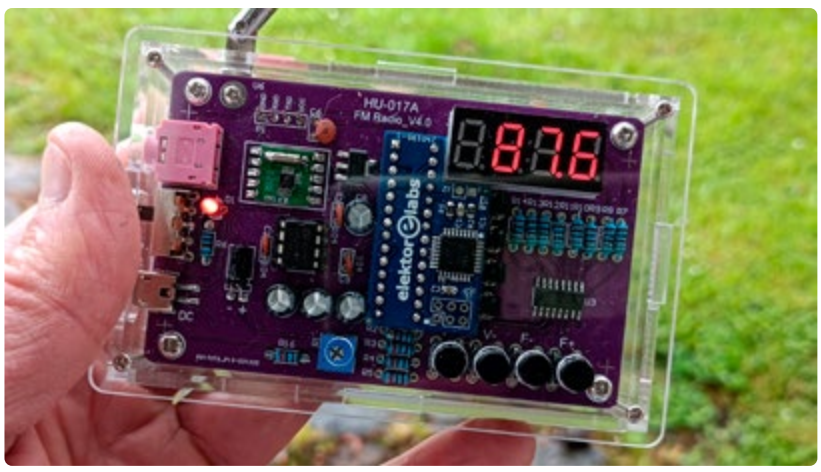
Communication with the RDA5807 radio chip is handled by the library mentioned above. This library has many blocking functions (functions that wait for an action to terminate), yet another reason for a timer-controlled background task to refresh the display continuously.

▲ *Figure 4. The ATmega328PB-to-STC15W408AS adapter board, configured for software development.*

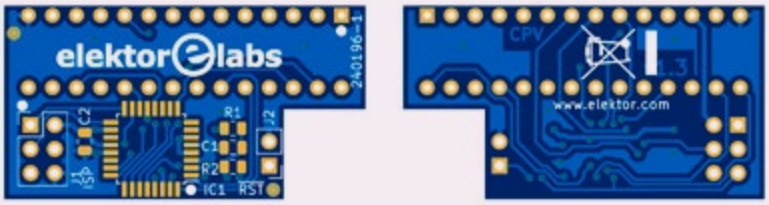


◀ *Figure 5. The adapter board fits between C5 and C8 and transistors Q1 to Q4.*

▼ *Figure 6. The modded FM radio.*



Component List



C1, C2 = 100 nF
IC1 = ATmega328PB-N
J1 = 6-way 2×3 pin header, 0.1″ pitch
J2 = 2-way pin header, 0.1″ pitch
R1 = 4.7 kΩ
R2 = 330 Ω
U2 = 2× 14-way single-row pin header, 0.1″ pitch

Two Modes of Operation

After power-up, the software is in Mode 0, and it works as if you used the original firmware. A long press (one second or more) on F+ activates Mode 1. Now pressing the V+ button shows the signal strength (RSSI), and pressing V– toggles between mono and stereo. Pressing F– toggles bass boost on and off (use headphones to hear it). Repeatedly pressing F+ cycles through RDS information levels (0 means OFF). A long press on F+ returns to Mode 0.

The RDA5807 library has a few more functions to play with, but I leave those to the reader.

RDS

RDS data is sent to the serial port, not to the display (another exercise for the reader). Level 1 is for text group 2A (Radio Text), level 2 is for group 0A (basic tuning information) and level 3 reads RDS time. See [1] for more information about RDS data. Be aware that not every radio station transmits RDS data and if they do, it may be incomplete and/or wrong. If you see a mix of readable and unreadable characters, then you probably need to adjust the antenna to improve reception. The new RSSI function can help you here.

Programming the ATmega328PB

To turn the ATmega328PB in an Arduino-compatible microcontroller, you should first load it with a bootloader. A suitable bootloader can be found at [4] — it is called `optiboot_elektor_uno_r4_8mhz.hex`. Use an ISP adapter to flash the bootloader onto the MCU and to set the fuses. The required fuse values are given in **Table 1**.

Table 1. Fuse values for the Elektor 8 MHz bootloader for the ATmega328PB.

Fuse	Value
Low	0xe2
High	0xde
Extended	0xf5

At [3], you can find detailed instructions on how to use the ATmega328PB with the Arduino IDE.

Connect a push button to J2. This will be the reset button. Connect a serial-to-USB adapter to J1. To upload a compiled sketch to the MCU, press the IDE's Upload button as usual while watching the IDE's output window. When the `Uploading...` message appears in the IDE, press the reset button on the adapter board. The sketch should now upload as usual. Press reset once more to start the new program.

The PCB design files and source docs are available on the Elektor Labs page for this project [4].

240196-01

Questions or Comments?

Do you have technical questions or comments about his article? Email the author at clemens.valens@elektor.com or contact Elektor at editor@elektor.com.

Related Products

- **HU-017A RDA5807 FM Radio Kit**
  www.elektor.com/20866
- **C. Valens, "Mastering Microcontrollers Helped by Arduino" (Elektor, 2017)**
  www.elektor.com/17967

WEB LINKS

[1] Radio Data System (RDS): https://en.wikipedia.org/wiki/Radio_Data_System
[2] RDA5807 library for Arduino: https://github.com/pu2clr/RDA5807
[3] Elektor Uno R4: https://elektormagazine.com/labs/elektorino-uno-r4-150790
[4] This project on Elektor Labs: https://elektormagazine.com/labs/rda5907-fm-radio-kit
[5] FM Radio Kit in Elektor Store: www.elektor.com/20866

LUCKY YOU!

GET FREE DOWNLOAD

An e-zine subscriber never misses the monthly 'reverse project'

Not a subscriber yet? Sign up for our free e-zine newsletter at **elektormagazine.com/ezine-24**

elektor
design > share > earn