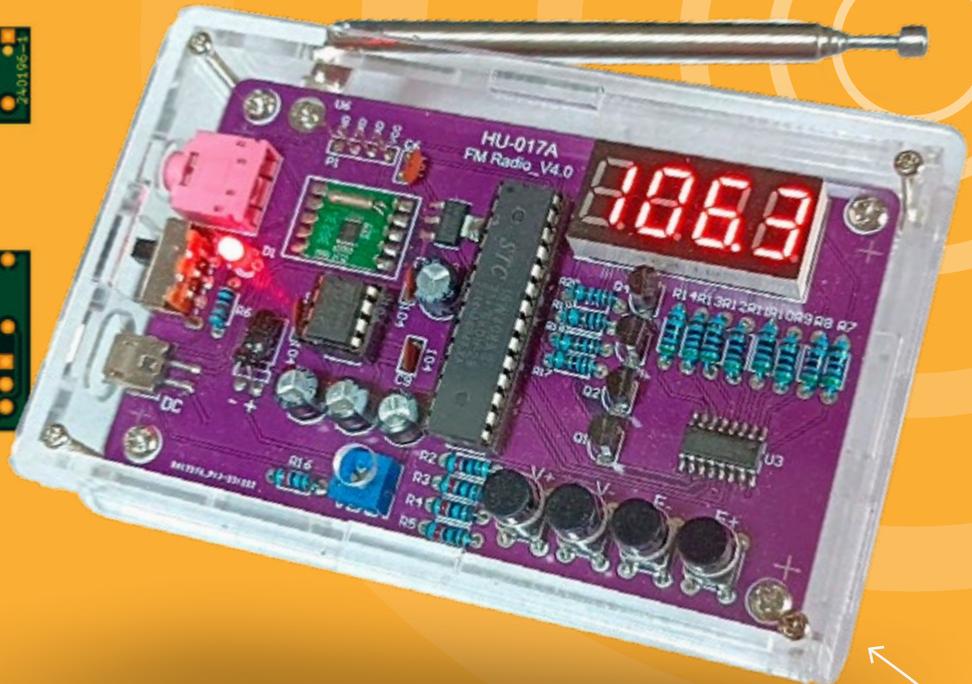
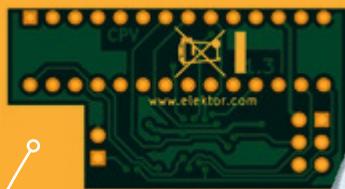


MODIFIZIERUNG EINES RDA5807-BASIERTEN FM-RADIO-KITS

NEU

RDS, RSSI, Bass Boost und mehr hinzufügen



Treten Sie jetzt der Elektor Community bei!



Jetzt



Mitglied werden!



- ✓ Zugang zum kompletten Online-Archiv (1970-heute)!
- ✓ 8x Elektor Magazin (gedruckt)
- ✓ 8x Elektor Magazin (PDF)
- ✓ 10% Rabatt im Elektor Store und exklusive Angebote
- ✓ Zugriff auf über 5.000 Gerber-Dateien
u.v.m. aus der Projektplattform Elektor Labs



Auch erhältlich

Die digitale
Mitgliedschaft!



- ✓ Zugang zum kompletten Online-Archiv
- ✓ 8x Elektor Magazin (PDF)
- ✓ 10% Rabatt im Elektor Store und exklusive Angebote
- ✓ Zugriff auf über 5.000 Gerber-Dateien
u.v.m. aus der Projektplattform Elektor Labs



www.elektormagazine.de/Abonnement



Liebe Leserinnen und Leser,

Wir freuen uns, eine neue Initiative anzukündigen, die exklusiv für unsere e-zine-Leser verfügbar ist: die „reverse projects“. Ab dieser Ausgabe werden wir jeden Monat einen exklusiven Artikel anbieten, der sich mit einem interessanten und unterhaltsamen Kit beschäftigt, das wir beschafft haben und als eigenes Elektor-Projekt betrachten. Für unser erstes Projekt hat der Elektor-Ingenieur Clemens Valens ein erschwingliches FM-Radio-Kit unter die Lupe genommen. Clemens analysiert die gelieferte Hardware gründlich und teilt seine professionelle Meinung zur Bauqualität und den vom Hersteller verpassten Möglichkeiten. Er hat sogar den nächsten Schritt gemacht und eine kleine Modifikation entwickelt: eine zusätzliche Platine, die die Funktionalität des Kits erheblich erweitert.

Dies ist ein typischer Elektor-Ansatz: Mehr aus bestehenden Produkten herausholen durch kritische Analyse und kreative Lösungen. Die Gerber-Dateien und alle notwendigen Informationen für diese Modifikation sind kostenlos als kleines Projekt auf unserer [Elektor Labs Plattform](#) verfügbar, sodass Sie es einfach selbst herstellen können. Darüber hinaus bieten wir die Möglichkeit, an einer „Elektor Jumpstarter-Kampagne“ teilzunehmen. Wenn sich 100 oder mehr Interessenten finden, wird Elektor eine kleine Serie dieser Platinen produzieren, die zu einem begrenzten Preis in unseren Geschäften erhältlich sein wird. Das Original-Kit ist natürlich im [Elektor Store](#) erhältlich.

Wir wünschen allen viel Spaß mit dieser ersten Ausgabe der „reverse projects“. Die nächste ist bereits in Arbeit! Die Kriterien für diese Serie sind einfach: beliebte, erschwingliche Kits, die online gut verfügbar sind und ein unterhaltsames Projekt für einen Abend oder ein Wochenende bieten. Durch die Analyse der Entscheidungen der Hersteller lernen wir gemeinsam mehr über das Design und die Möglichkeiten dieser Kits.

Unser Redaktionsteam und das Storeteam sind immer offen für Vorschläge und Feedback. Wir freuen uns darauf, von Ihnen in den Kommentaren oder im Online-LAB zu hören.

Mit freundlichen Grüßen,

CJ Abate
Content Director, Elektor

Modifizierung eines RDA5807- basierten FM-Radio-Kits

RDS, RSSI, Bass Boost und mehr hinzufügen

Von Clemens Valens

Ein Radio aus einem Bausatz zu bauen macht Spaß. Was ist aber, wenn es fertig ist? Für viele Elektronikbegeisterte beginnt der eigentliche Spaß erst, wenn sie hier modifizieren können. In diesem Artikel stellen wir Ihnen einen günstigen FM-Radio-Bausatz vor und zeigen, wie Sie ihn in ein Arduino-kompatibles, frei programmierbares Gerät verwandeln.

Im Internet finden Sie günstige UKW-Radioempfänger-Bausätze, die auf dem RDA5807 „single-chip FM broadcast stereo radio tuner“ von RDA Microelectronics basieren. Dieser IC enthält alles, was Sie zum Bau eines FM-Rundfunkempfängers benötigen, einschließlich eines Stereo-Kopfhörerausgangs. Es fehlt lediglich eine Benutzeroberfläche, die Sie selbst hinzufügen sollen. Genau das tun diese Kits – sie fügen dem RDA5807, der selbst auf einem kleinen Breakout-Board montiert ist, einen Mikrocontroller, ein Display, Tasten, eine Antenne und einen Audioverstärker hinzu.

Sobald sie zusammengebaut sind, haben Sie ein nettes kleines tragbares, batteriebetriebenes FM-Radio (**Bild 1**). Die Klangqualität des eingebauten Lautsprechers ist eher schlecht, aber mit Kopfhörern ist sie ausgezeichnet. Leider laufen die Kits mit Closed-Source-Software auf einem STC15W408AS (einem verbesserten 8051) und es gibt keine Erweiterungs-Ports.

Eingeschränkte Funktionalität

Soweit ich weiß, sind die einzigen Funktionen, die das Radio-Kit bietet, Frequenz hoch und runter sowie Lautstärke lauter und leiser. Das ist schade, denn der RDA5807 bietet mehr Funktionen, insbesondere RDS (Radio Data System [1]) und Bass Boost. Um diese

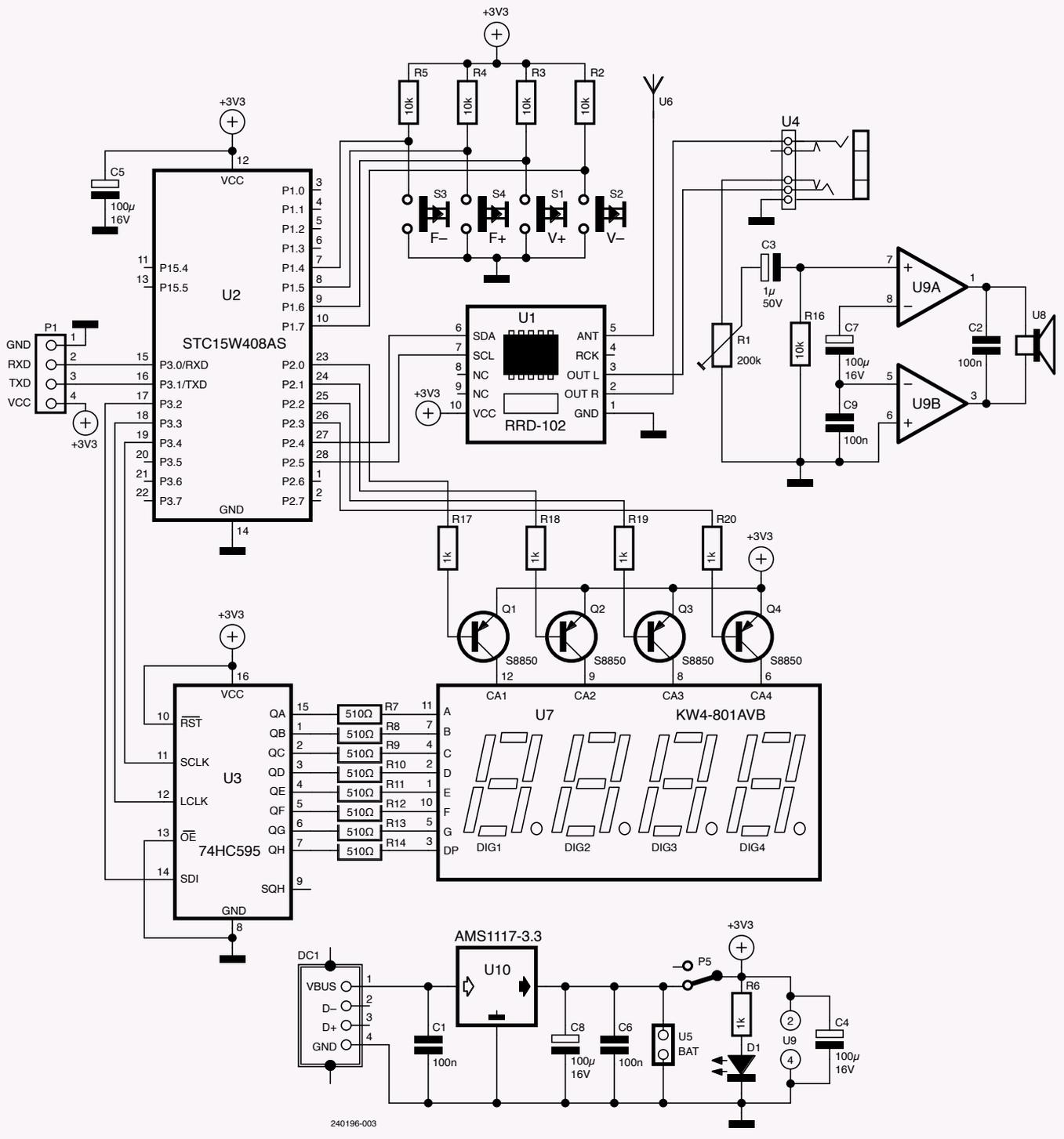
zugänglich zu machen, müssen Sie entweder die Firmware der mitgelieferten MCU ändern oder einen anderen Controller an den I²C-Port des RDA5807 anschließen. Eine dritte Möglichkeit besteht darin, die MCU ganz zu ersetzen. Das habe ich getan. Aber bevor wir uns damit befassen, möchte ich Ihnen zunächst das Kit etwas genauer beschreiben.

Das HU-017A RDA5807 Radio Kit

Der von mir verwendete Bausatz ist mehr oder weniger markenlos und hört auf den Namen HU-017A. Er besteht aus einer lilafarbenen Platine von 6 cm x 9,5 cm, einem Beutel mit Komponenten, einer kurzen (25 cm) Teleskopantenne sowie einem lasergeschnittenen Acrylgehäuse. Bei den Komponenten handelt es sich hauptsächlich um durchkontaktierte Bauteile. Das ist aber nicht bei allen der Fall. Ich nehme an, dass U3, ein 74HC595 Schieberegister, aus Gründen der Größe ein SMD-Bauteil ist. Der Spannungsregler U10 ist ebenfalls ein SMD-Bauteil. Der RDA5807 ist ein winziger Chip, der auf einer kleinen Platine mit Wabenkontakten montiert ist. Er ist also streng genommen auch ein SMD-Bauteil. Der SMT-Micro-USB-B-Stromanschluss hat nur zwei Pins (keine Datenpins), sodass man ihn leicht löten kann. Alle anderen Teile sind THT-Bauteile.



Bild 1.
Das HU-017A RDA5807
Radio Kit komplett montiert.



240196-003

Der Schaltkreis

Der Schaltplan des Funkgeräts ist im Benutzerhandbuch abgedruckt. Zudem haben wir ihn in **Bild 2** nachgezeichnet. Sie sehen vier Drucktasten, die mit einem Mikrocontroller verbunden sind. Er steuert Ziffer für Ziffer eine 4-stellige Sieben-Segment-Anzeige mit gemeinsamer Anode über das bereits erwähnte Schieberegister 74HC595 (U3) mithilfe der Transistoren Q1 bis Q4 an (d.h. Multiplexing).

Die MCU kommuniziert mit dem Funkchip U1 über einen I²C-Bus.

Audio Output

U9 ist ein Audioverstärker in Brückenkonfiguration, der einen 8-Ω-Lautsprecher ansteuert. Sein Eingangssig-

nal ist der rechte Audioausgangskanal des RDA5807. U4, eine 3,5-mm-Buchse, ist für den Anschluss von Kopfhörern gedacht. Tun sie das, hören Sie das Stereo-Audiosignal, das von viel besserer Qualität als das des kleinen Lautsprechers ist. Die Lautstärke wird über zwei Drucktasten gesteuert V- und V+. Sie kontrollieren das Lautstärkeregister im RDA5807. Der Trimmer R1 stellt die maximale Lautstärke des Verstärkers ein, nicht die des Kopfhörers.

Der Spannungsregler U10 wandelt eine externe Eingangsspannung in 3,3 V um. Er unterstützt bis zu 15 V am Eingang. Allerdings gibt es keinen Umkehrschutz für die Verpolung. Der Micro-USB-B-Anschluss DC1 bietet diesen Schutz, aber nur, wenn Sie ein fertiges Ladegerät für Telefone verwenden.

▲
Bild 2. Schaltplan des HU-017A RDA5807 Radio Kits.

siehe [3]). Das „PB“ gibt es nicht in einem 28-poligen DIP-Gehäuse, sondern nur in 32-poligen SMD-Gehäusen. Da eine Neuverkabelung ohnehin erforderlich ist, ist das nicht wirklich ein Problem. Eigentlich ist es sogar gut, denn es hält die Höhe des Adapters gering.

ATmega328PB Adapter-Board

Bild 3 zeigt, wie ich den ATmega328PB mit dem Gehäuse der ursprünglichen MCU verbunden habe. Es gibt viele Bibliotheken zur Steuerung des 74HC595 via SPI. Ich zog es aber vor, den SPI-Bus für die ISP-Programmierung der MCU zu reservieren (J1). J2 steht für den Anschluss eines Reset-Tasters zur Verfügung. Das ist praktisch, sobald die neue MCU einen Arduino-Bootloader enthält, da kein automatischer Reset möglich ist (ohne hässliche Drähte hinzuzufügen) (**Bild 4**).

Ich habe eine kleine Platine für den Adapter entworfen, die in den von C5 und C8 auf der linken Seite und Q1 bis Q4 auf der rechten Seite begrenzten Raum passt – siehe **Bild 5**.

Neue Software

Die neue Software sollte mindestens die gleiche Funktionalität haben wie die ursprüngliche Software, d.h. Lautstärke laut und leise sowie Frequenz auf und ab – durch Drücken der entsprechenden Tasten. Da es nur vier Tasten gibt, müssen sie auch für den Zugriff auf alle anderen Funktionen verwendet werden, die möglicherweise implementiert werden. Da alle möglichen Einzeltastendrücke bereits verwendet werden, ist ein weiterer Mechanismus erforderlich. Ich habe mich für einen langen Druck auf die Taste F+ entschieden, um auf andere Betriebsmodi zuzugreifen.

Bildschirmtreiber

Die Arduino-Funktion `shiftOut()` wird verwendet, um Zeichen auf der Sieben-Segment-Anzeige auszugeben. Die Ziffern sind gemultiplext. Das bedeutet, dass immer nur eine aktiv sein kann. Die Anzeige muss also ständig aktualisiert werden, um mehrstellige Werte und Zeichenketten anzuzeigen. Das geschieht am besten durch einen Timer-Task, der im Hintergrund läuft. Deswegen ist natürlich ein Timer erforderlich.

TimerOne

Da ein Timer notwendig wird, habe ich mich für Timer1 entschieden, da Timer0 für die Arduino-Funktionen `millis()` sowie `delay()` verwendet wird. Sie werden häufig von Arduino-Bibliotheken verwendet und daher ist es besser, nicht damit zu experimentieren. **TimerOne** ist eine schicke kleine Bibliothek, um Timer1 beim Arduino zu nutzen.

Ich habe Timer1 für eine Frequenz von 1 kHz eingerichtet. Jede Millisekunde ruft er die Funktion `my_millis_counter()` auf, die wiederum `display_refresh()` aufruft. Timer1 zählt auch die Millisekunden für den Tastenscanner. Daher sind die Aktualisierung des Displays und die Tastenabfrage synchronisiert. Das gewährleistet ein flackerfreies Display und reaktionsschnelle, entprellte Tasten.



Bild 4. Das ATmega328PB-zu-STC15W408AS-Adapter-Board, konfiguriert für die Software-Entwicklung.

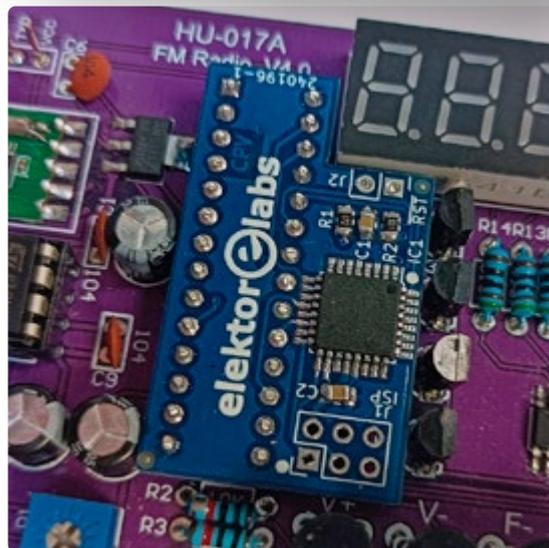


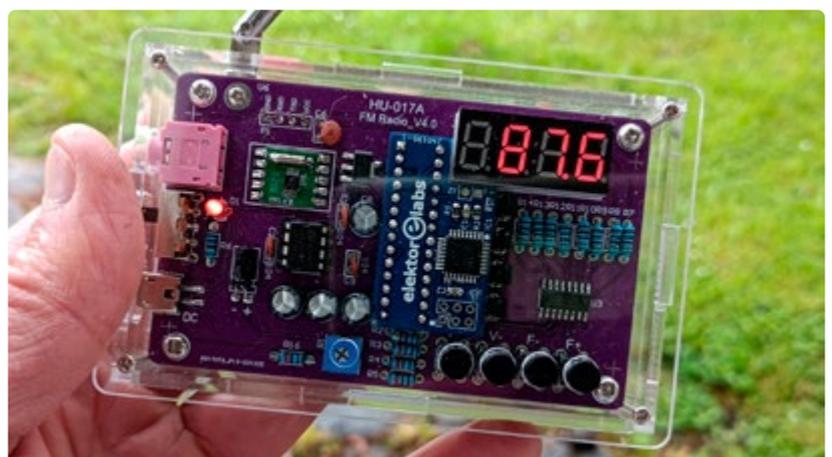
Bild 5. Die Adapter-Board passt zwischen C5 und C8 und die Transistoren Q1 bis Q4.

Tastenabfrage und Entprellung

Die Tastenabtastung und Entprellung wird durch einen einfachen Algorithmus umgesetzt. Wird ein Tastendruck erkannt (ein Ereignis), wird die Startzeit des Ereignisses aufgezeichnet. Wird die Taste losgelassen, dann wird die Endzeit des Ereignisses aufgezeichnet. Ist der Zeitraum zwischen der Start- und der Endzeit zu kurz, wird er als Prellen betrachtet und die Taste zurückgesetzt. Ist die Zeitspanne länger als ein Prellen, aber kürzer als ein langer Druck, alles andere ist ein langer Tastendruck. Es können mehrere Tasten gleichzeitig gedrückt werden.

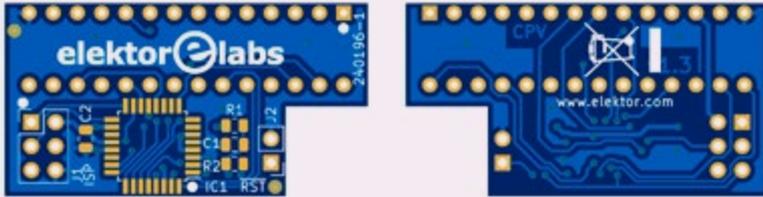
Die Kommunikation mit dem RDA5807-Funkchip wird von der oben erwähnten Bibliothek abgewickelt. Diese Bibliothek verfügt über viele Sperrfunktionen (Funktionen,

Bild 6. Das modifizierte FM-Radio.





Liste der Komponenten



C1, C2 = 100 nF
 IC1 = ATmega328PB-N
 J1 = 2x3 Stiftleiste, Rastermaß 0,1"
 J2 = 2x1 Stiftleiste, Rastermaß 0,1"
 R1 = 4.7 kΩ
 R2 = 330 Ω
 U2 = 2x 1x14 Stiftleiste, Rastermaß 0,1"

die auf die Beendigung einer Aktion warten). Das ist ein weiterer Grund für eine Timer-gesteuerte Hintergrundaufgabe zur kontinuierlichen Aktualisierung der Anzeige.

Zwei Modi für den Betrieb

Nach dem Einschalten befindet sich die Software im Modus 0 und funktioniert so, als ob Sie die Original-Firmware verwenden würden. Ein langer Druck (eine Sekunde oder länger) auf F+ aktiviert den Modus 1. Wenn Sie nun die Taste V+ drücken, wird die Signalstärke (RSSI) angezeigt. Beim Drücken von V- schalten Sie zwischen Mono und Stereo um. Durch Drücken von F- schalten Sie die Bassverstärkung ein und aus (verwenden Sie Kopfhörer, um sie zu hören). Durch wiederholtes Drücken von F+ wechseln Sie zwischen den RDS-Informationstufen (0 bedeutet AUS). Ein langer Druck auf F+ kehrt zum Modus 0 zurück.

Die RDA5807-Bibliothek bietet noch ein paar weitere Funktionen, aber die überlasse ich dem Leser.

RDS

Die RDS-Daten werden an die serielle Schnittstelle gesendet, nicht an das Display (eine weitere Übung für den Leser). Ebene 1 ist für die Textgruppe 2A (Radio-Text), Ebene 2 ist für die Gruppe 0A (grundlegende Abstimmungsinformationen) und Ebene 3 liest die RDS-Zeit. Siehe [1] für weitere Informationen über RDS-Daten. Beachten Sie, dass nicht jeder Radiosender RDS-Daten sendet. Sollte das dennoch der Fall sein, können diese unvollständig und/oder falsch sein. Sehen Sie eine Mischung aus lesbaren und unlesbaren Zeichen, müssen Sie wahrscheinlich die Antenne anpassen, um den Empfang zu verbessern. Die neue RSSI-Funktion kann Ihnen dabei helfen.

Den ATmega328PB programmieren

Um den ATmega328PB in einen Arduino-kompatiblen Mikrocontroller zu verwandeln, sollten Sie ihn zunächst

mit einem Bootloader laden. Einen geeigneten Bootloader finden Sie unter [4] – er heißt `optiboot_elektor_uno_r4_8mhz.hex`. Verwenden Sie einen ISP, um den Bootloader auf die MCU zu flashen und die Sicherungen zu setzen. Die erforderlichen Sicherungswerte finden Sie in **Tabelle 1**.

Tabelle 1. Sicherungswerte für den Elektor 8 MHz Bootloader für den ATmega328PB.

Sicherung	Wert
Niedrig	0xe2
Hoch	0xde
Erweitert	0xf5

Unter [3] finden Sie eine ausführliche Anleitung, wie Sie den ATmega328PB mit der Arduino IDE verwenden können.

Schließen Sie einen Druckknopf an J2 an. Das wird die Reset-Taste sein. Schließen Sie einen Seriell-zu-USB-Adapter an J1 an. Um einen kompilierten Code auf die MCU hochzuladen, drücken Sie wie gewohnt die Taste Upload der IDE, während Sie das Ausgabefenster des IDE beobachten. Erscheint die Meldung `Uploading...` im IDE, drücken Sie den Reset-Knopf auf dem Adapter-Board. Der Code sollte sich nun wie normal hochladen. Drücken Sie den Reset-Knopf abermals, um das neue Programm zu starten.

Die PCB-Designdateien und die Quelldokumente sind auf der Elektor-Labs-Seite für dieses Projekt verfügbar [4].

240196-02

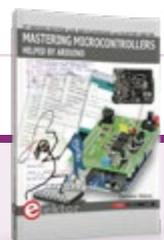
Fragen oder Kommentare?

Haben Sie technische Fragen oder Kommentare zu diesem Artikel? Schicken Sie eine E-Mail an den Autor unter clemens.valens@elektor.com oder kontaktieren Sie Elektor unter editor@elektor.com.



Verwandte Produkte

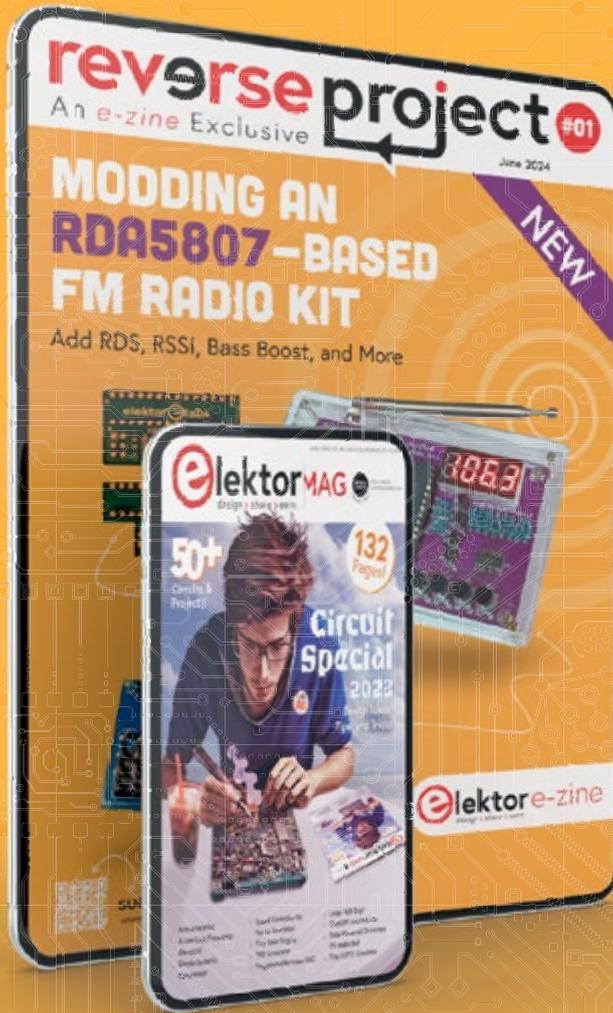
- > **HU-017A RDA5807 FM Radio Kit**
www.elektor.de/20866
- > **C. Valens, "Mastering Microcontrollers Helped by Arduino" (Elektor, 2017)**
www.elektor.de/17967



WEBLINKS

- [1] Radio Data System (RDS): https://en.wikipedia.org/wiki/Radio_Data_System
- [2] RDA5807-Bibliothek für Arduino: <https://github.com/pu2clr/RDA5807>
- [3] Elektor Uno R4: <https://elektormagazine.com/labs/elektorino-uno-r4-150790>
- [4] Dieses Projekt bei Elektor Labs: <https://elektormagazine.com/labs/rda5907-fm-radio-kit>

DU GLÜCKSPILZ!



KOSTENLOSER
DOWNLOAD

Ein Leser des e-zine Newsletters verpasst nie das monatliche 'reverse project'

Du bist noch kein Leser? Melde dich für unseren Newsletter an, elektormagazine.de/ezine-24

