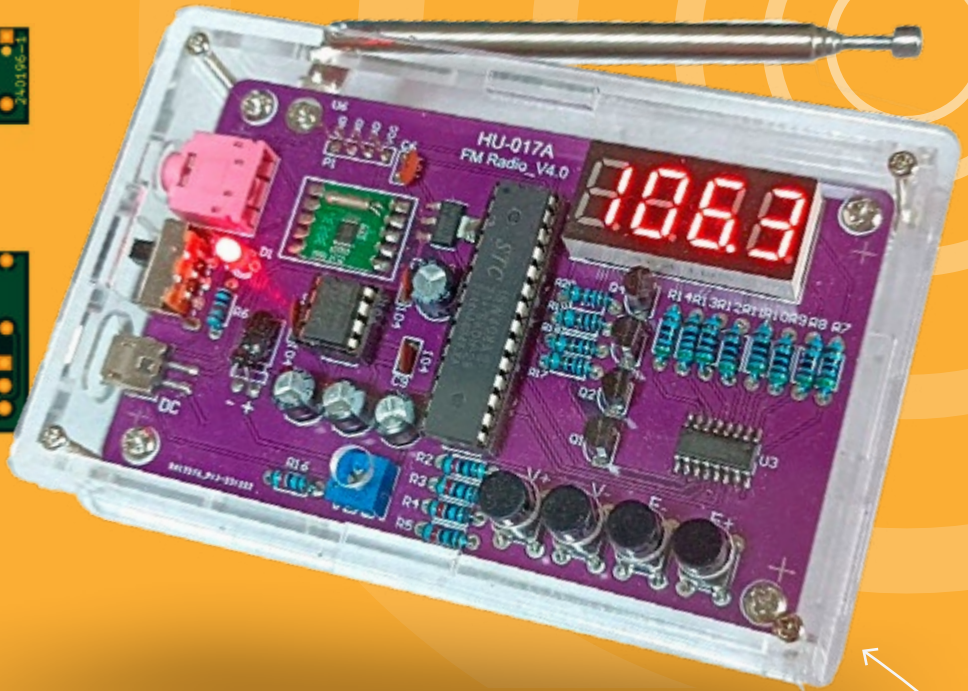
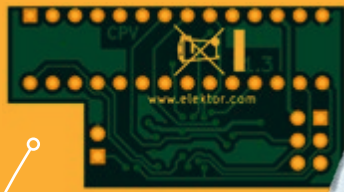


MODIFICATION D'UN KIT RADIO FM UTILISANT UN RDA5807

NOUVEAU

Ajout de RDS, RSSI, renforcement des graves et plus encore



Rejoignez la communauté Elektor



Devenez membre maintenant !



- ✓ accès à l'archive numérique depuis 1978 !
- ✓ 8x magazine imprimé Elektor
- ✓ 8x magazine numérique (PDF)
- ✓ 10 % de remise dans l'e-choppe et des offres exclusives pour les membres
- ✓ accès à plus de 5000 fichiers Gerber



Également disponible

abonnement



sans papier !

- ✓ accès à l'archive numérique d'Elektor
- ✓ 10 % de remise dans l'e-choppe
- ✓ 8x magazine Elektor (PDF)
- ✓ accès à plus de 5000 fichiers Gerber



www.elektormagazine.fr/membres



Chers lecteurs,

Nous sommes heureux d'annoncer une nouvelle initiative exclusivement réservée à nos lecteurs de l'e-zine : les « reverse project ». À partir de ce numéro, nous proposerons chaque mois un article exclusif qui se penche sur un kit amusant et intéressant que nous avons sourcé et abordé comme s'il s'agissait d'un projet Elektor en interne. Pour notre premier projet, un ingénieur d'Elektor a examiné un kit radio FM abordable. Il analyse en profondeur le matériel fourni et partage son avis professionnel sur la qualité de construction et les opportunités manquées par le fabricant. Il a même franchi une étape supplémentaire en développant une petite modification : une carte additionnelle qui améliore considérablement les fonctionnalités du kit. C'est une approche typique d'Elektor : extraire plus de valeur des produits existants grâce à une analyse critique et à des solutions créatives. Les fichiers Gerber et toutes les informations nécessaires pour cette modification sont disponibles gratuitement en tant que petit projet sur le labo d'[Elektor en ligne](#), afin que vous puissiez facilement le réaliser vous-même. De plus, nous offrons la possibilité de rejoindre une "campagne Elektor Jumpstarter". Si 100 participants ou plus sont intéressés, Elektor produira une petite série de ces cartes, qui seront disponibles à un prix limité dans nos magasins. Le kit original est bien sûr disponible dans la [boutique Elektor](#).

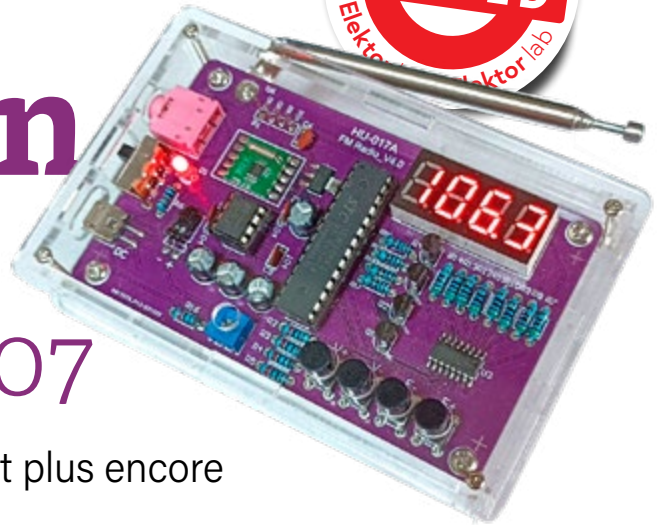
Nous vous souhaitons beaucoup de plaisir avec cette première édition des « reverse project ». Le prochain est déjà en préparation ! Les critères pour cette série sont simples : des kits populaires et abordables, facilement disponibles en ligne, et offrant un projet amusant pour une soirée ou un week-end. En analysant les choix des fabricants, nous apprenons collectivement davantage sur la conception et le potentiel de ces kits.

Notre équipe éditoriale et l'équipe de la boutique sont toujours ouvertes aux suggestions et aux retours. Nous sommes impatients de vous entendre dans les commentaires ou sur le labo en ligne.

Cordialement,

CJ Abate
Content Director, Elektor

Modification d'un kit radio FM utilisant un RDA5807



Ajout de RDS, RSSI, renforcement des graves et plus encore

Clemens Valens

Construire une radio à partir d'un kit est amusant, mais qu'en est-il une fois qu'elle est assemblée ? Pour de nombreux passionnés d'électronique, le véritable plaisir ne commence que lorsque vous pouvez la modifier. Dans cet article, nous présentons un kit de radio FM bon marché et une façon de le transformer en un appareil compatible avec Arduino et programmable par l'utilisateur.

Sur Internet, il est possible de trouver des récepteurs radio FM peu coûteux basés sur le circuit-intégré RDA5807 "single-chip FM broadcast stereo radio tuner" (tuner radio stéréo FM sur un seul chip) de RDA Microelectronics. Ce circuit intégré contient tout ce qui est nécessaire pour réaliser un récepteur pour la gamme de radiodiffusion FM, incluant une sortie pour écouteur stéréophonique. Il ne manque qu'une interface utilisateur que vous devez ajouter vous-même. C'est précisément l'objet de ces kits – ils ajoutent un microcontrôleur, un affichage, des boutons, une antenne et un amplificateur audio au RDA5807 qui, lui, est monté sur une petite carte. Après avoir terminé l'assemblage, vous disposez d'une petite radio FM portable, alimentée par une pile (**Figure 1**). La qualité sonore du haut-parleur intégré est assez désagréable, mais le son est excellent en utilisant des écouteurs. Malheureusement, le kit utilise un logiciel fermé fonctionnant sur un STC15W408AS (version améliorée du 8051) et aucun port d'extension n'est disponible.

Fonctionnalités limitées

Selon mes constatations, les seules fonctions qu'offrent les kits permettent le réglage de la fréquence (boutons augmenter et diminuer) et du volume (plus fort, moins fort). C'est regrettable, car le RDA5807 permet d'autres fonctions, en particulier le RDS (Radio Data System) et le renforcement des graves (Bass boost). Pour les mettre

Figure 1. Le kit radio HU-017A RDA5807 assemblé.

à disposition, il faut soit changer le microcode du microcontrôleur utilisé, ou associer au RDA5807 un autre circuit de contrôle, connecté au bus I²C. Une troisième possibilité consiste à remplacer le microcontrôleur, c'est ce que j'ai fait. Mais avant d'étudier en détail cette modification, permettez-moi de décrire le kit de façon plus complète.

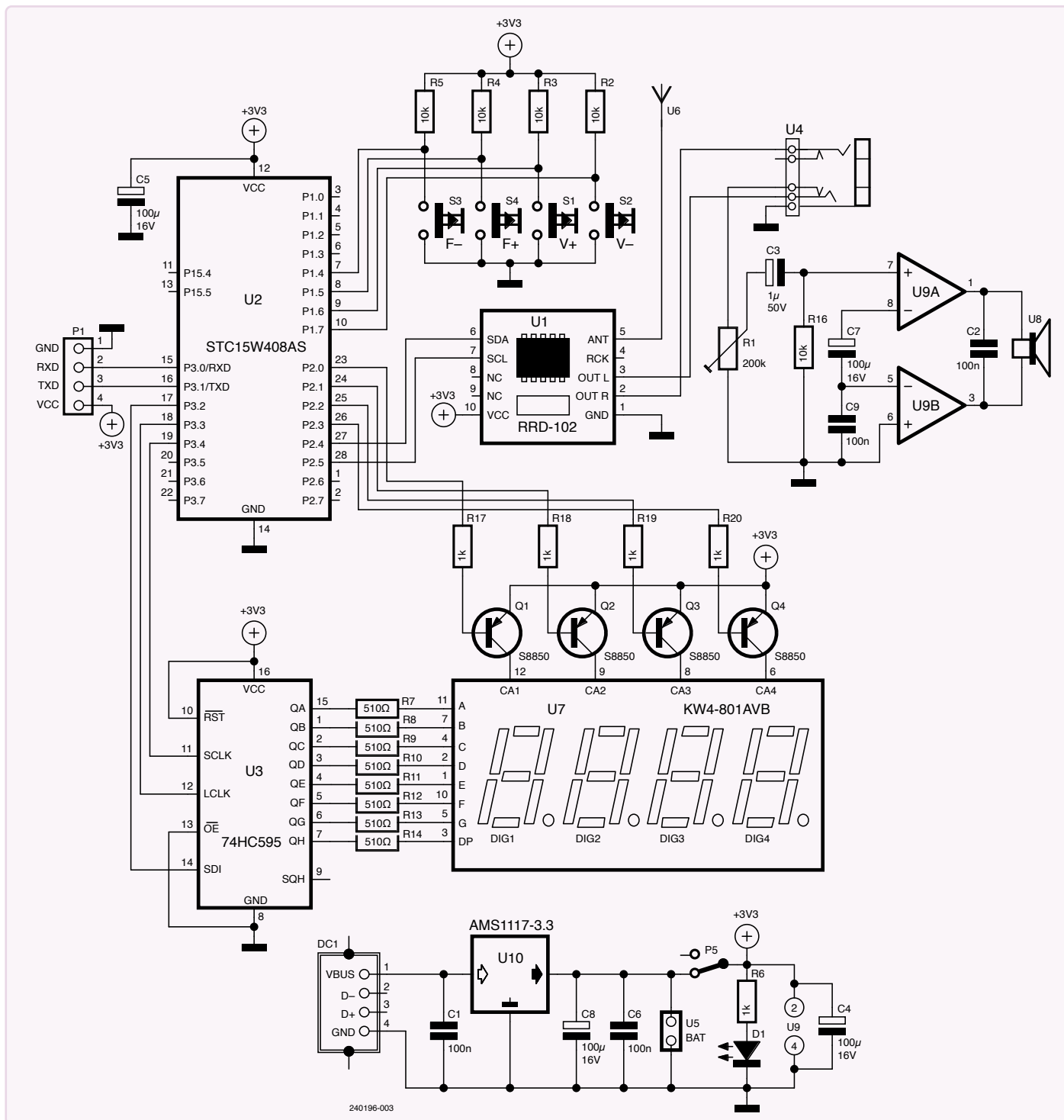
Le kit radio HU-017A RDA5807

Le kit que j'ai utilisé est plus ou moins générique, sans marque, il est identifié HU-107A. Il est constitué d'un circuit-imprimé de couleur violette de 6 cm par 9,5 cm, d'un sachet de composants, d'une petite antenne télescopique de 25 cm et d'un boîtier en acrylique, découpé au laser.

Les composants sont pour la plupart de type traversant, mais pas tous. Sans doute pour des raisons d'encombrement, U3, un registre à décalage 74HC595 est un composant CMS (monté en surface). Le régulateur de tension U10 est également de type CMS. Le RDA5807 est un petit circuit intégré monté sur une petite plaque comportant des contacts crénelés, ainsi, en vérité, c'est également un composant CMS. Le connecteur USB-B également monté en surface ne comporte que deux contacts (pas de broches de données), il est ainsi facile à souder. Tous les autres composants sont de type traversant.

Le circuit

Le schéma de ce récepteur figure dans le manuel utilisateur, nous avons dû le retracer sur la **Figure 2**. Il comporte quatre boutons-poussoirs reliés à un microcontrôleur qui pilote un afficheur 7-segments de 4 chiffres à anode commune, affichant un seul chiffre à la fois (par multiplexage), par l'intermédiaire du registre à décalage 74HC595 (U3) associé aux transistors Q1 à Q4. Le microcontrôleur communique avec le circuit intégré récepteur radio U1, par l'intermédiaire d'un bus I²C.



Sortie audio

U9 est un amplificateur audio à configuration en pont, pilotant un haut-parleur de 8 Ω. Le signal à son entrée provient de la sortie audio du canal de droite du RDA5807. U4 est un connecteur jack de 3,5 mm permettant de connecter les écouteurs. Avec cette connexion, le son stéréophonique sera diffusé, avec une qualité supérieure au son délivré par le petit haut-parleur. Le volume est contrôlé par deux boutons-poussoirs V- et V+ qui contrôlent le registre du volume interne au RDA5807. Le potentiomètre ajustable R1 définit le volume maximum de l'amplificateur, mais pas celui des écouteurs. Le régulateur de tension U10 convertit la tension d'alimentation externe en 3,3 V pour l'alimentation interne. Il supporte jusqu'à 15 V en entrée, mais il n'y a pas de

protection contre l'inversion de polarité. Le connecteur micro USB-B DC1 offre un peu de protection, mais uniquement si vous utilisez un chargeur pour téléphone portable.

Attention à la batterie ou aux piles

Prenez la précaution de retirer les piles lorsque le récepteur est alimenté par une source externe, car il n'y a pas de protection pour les piles ou la batterie. Si vous oubliez de le faire, les piles seront endommagées et pourraient fuir, ou pire éclater.

Le connecteur linéaire P1 permet l'accès au port série, mais je n'ai constaté aucune activité sur ses broches. Si vous disposez des outils adéquats, vous pourrez l'utiliser pour programmer le STC05W408AS in-situ.

▲
Figure 2. Schéma du kit radio HU-017A RDA5807.

Carte adaptatrice ATmega328PB du projet

La **Figure 3** montre la façon dont j'ai connecté l'ATmega328PB sur l'emplacement du microcontrôleur d'origine. Il existe de multiples bibliothèques de logiciel permettant de contrôler le 74HC595 à partir du bus SPI, mais j'ai préféré réserver le bus SPI à la programmation du microcontrôleur (J1). J2 est disponible pour la connexion d'un bouton-poussoir, ce qui est pratique dans le cas où le nouveau microcontrôleur contient un programme d'amorçage Arduino, car il n'y a pas de possibilité de réaliser une réinitialisation automatique (sans ajouter des connexions affreuses) voir **Figure 4**.

Pour cet adaptateur, j'ai conçu un petit circuit-imprimé qui se loge dans l'espace délimité par C5 et C8 à gauche de Q1 à Q4 à droite. Voir la **Figure 5**.

Nouveau logiciel

Le nouveau logiciel doit, au minimum, offrir les mêmes fonctionnalités que la version originale, c'est-à-dire le réglage du volume (V+ et V-) et de la fréquence de réception (F+ et F-), en appuyant sur les boutons-poussoirs correspondants. Du fait de la présence limitée à quatre boutons-poussoirs, ceux-ci doivent également permettre l'accès à toute autre fonction qui serait ajoutée. Les appuis brefs étant déjà utilisés, un autre mode de fonctionnement est nécessaire. J'ai choisi un appui long sur la touche F+ pour accéder aux autres modes d'opération.

Contrôleur de l'affichage

La fonction Arduino `shiftOut()` est utilisée pour afficher les caractères par l'afficheur 7-segments. Les chiffres sont multiplexés, c'est-à-dire qu'un seul peut être actif à un instant donné. Cela implique que l'affichage soit rafraîchi continuellement, afin d'afficher les valeurs à plusieurs chiffres et des chaînes de caractères. Cela se fait par un temporisateur fonctionnant en arrière-plan, ce qui nécessite la présence d'un temporisateur.

Premier temporisateur

Étant donné qu'un temporisateur est nécessaire, j'ai choisi Timer1, puisque Timer0 est utilisé par les fonctions Arduino `millis()` et `delay()`, qui sont habituellement mises en œuvre dans les bibliothèques Arduino, et il est préférable de ne pas s'en écarter. `TimerOne` est une petite bibliothèque logicielle prévue pour l'utilisation du Timer1 Arduino.

J'ai paramétré Timer1 à la fréquence de 1kHz. Il appelle la fonction `my_millis_counter()` à chaque milliseconde qui provoque un appel consécutif à `display_refresh()`. Timer1 décompte également les millisecondes pour la lecture de l'état du bouton-poussoir. De ce fait, le rafraîchissement de l'affichage et la scrutation des boutons-poussoirs sont synchronisées. Cela évite le clignotement de l'affichage et permet une bonne réactivité des boutons-poussoirs, sans effet de rebond.



Figure 4. Carte d'adaptation de l'ATmega328PB-au-STC15W408AS, configurée pour le développement de logiciel.

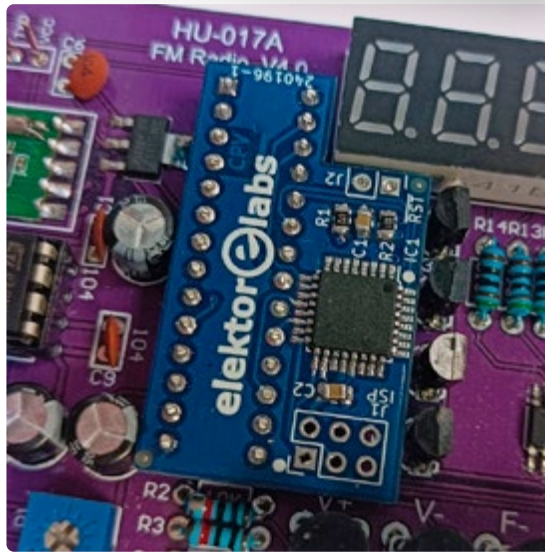
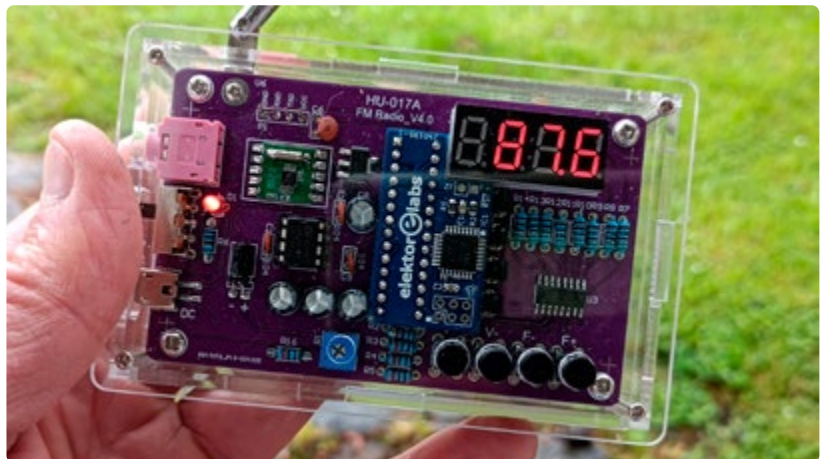


Figure 5. La carte d'adaptation est logée entre C5 et C8 et les transistors Q1 à Q4.

Scrutation et antirebond

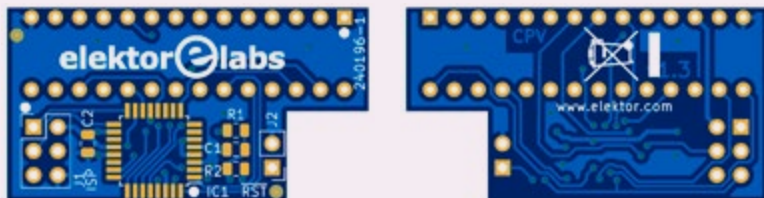
La scrutation et l'antirebond des boutons-poussoirs utilisent un algorithme simple. Quand un bouton-poussoir est enfoncé (détection d'une action), l'heure de début de l'action est enregistrée. Idem pour le relâchement du bouton. Si la période qui s'écoule entre le début et la fin de l'appui est trop courte, il s'agit d'un rebond, et l'état du bouton-poussoir est réinitialisé. Si ce temps est plus long que celui d'un rebond, mais plus court que celui d'un appui long, c'est un appui normal. Toute autre durée est un appui long. Plusieurs boutons-poussoirs peuvent être actionnés simultanément.

Figure 6. La radio FM modifiée.





Liste des composants



C1, C2 = 100 nF

IC1 = ATmega328PB-N

J1 = connecteur linéaire 2 x 3 broches mâles, pas de 2,54 mm.

J2 = connecteur linéaire de 2 broches mâles, pas de 2,54 mm.

R1 = 4,7 kΩ

R2 = 330 Ω

U2 = 2 connecteurs linéaires de 14 broches mâles, pas de 2,54 mm.

La communication avec la puce radio RDA5807 est assurée par la bibliothèque précédemment citée. Cette bibliothèque comporte plusieurs fonctions bloquantes (fonctions qui attendent qu'une action en cours se termine), c'est une raison supplémentaire de la nécessité de disposer d'un temporisateur contrôlé par une tâche en arrière-plan pour le rafraîchissement continu de l'affichage.

Deux modes d'opération

Après mise sous tension, le logiciel est en Mode 0, et il fonctionne comme si vous utilisiez le logiciel d'origine. Un appui long (une seconde au minimum) sur le bouton F+ active le Mode 1. Un appui sur le bouton V+ permet alors de visualiser la force du signal (RSSI), un appui sur le bouton V- permet de passer du mode Mono au mode Stéréo et inversement. Un appui sur F- active ou désactive le renforcement des sons graves (les écouteurs doivent être utilisés pour en profiter). Une succession d'appuis sur la touche F+ permet de parcourir les différents niveaux des informations RDS (le niveau 0 signifie arrêt). Un appui long sur F+ permet de revenir au mode 0.

La bibliothèque du RDA5807 comporte quelques fonctions supplémentaires dont je confie la mise en œuvre aux lecteurs

RDS

Les données RDS sont envoyées au port série, mais pas à l'afficheur (autre modification réalisable par les lecteurs). Le niveau 1 correspond au groupe texte 2A (Radio text), le niveau 2 correspond au groupe 0A (informations de base), et le niveau 3 transmet l'heure RDS. Voir [1] pour plus d'informations sur les données RDS. Il faut se rappeler que toutes les stations de radiodiffusion ne transmettent pas de données RDS. Quand elles le font, ces données peuvent être fausses ou incomplètes. Si vous constatez qu'une partie de ces données comporte des caractères illisibles, vous devez probablement ajuster l'antenne afin d'améliorer la réception du signal. La nouvelle fonction RSSI peut vous y aider.

LIENS

[1] Radio Data System (RDS): https://en.wikipedia.org/wiki/Radio_Data_System

[2] RDA5807 bibliothèque pour Arduino: <https://github.com/pu2clr/RDA5807>

[3] Elektor Uno R4: <https://elektormagazine.com/labs/elektorino-uno-r4-150790>

[4] Ce projet sur Elektor Labs: <https://elektormagazine.com/labs/rda5907-fm-radio-kit>

Programmation de l'ATmega328PB

Pour rendre le microcontrôleur ATmega328PB compatible Arduino, vous devez d'abord le charger avec un programme de chargement initial (bootloader). Un tel programme peut être téléchargé en [4] – il est nommé `optiboot_elektor_uno_r4_8mhz.hex`.

Un logiciel ISP (Programming in-situ) permettant de flasher le programme de chargement initial sur le microcontrôleur, et de paramétrer les "fusibles" de configuration (fuses) est nécessaire. Les valeurs des fusibles sont indiquées sur la **Table 1**.

Table 1. Valeur des fusibles pour le programme de chargement (bootloader) Elektor pour l'ATmega à 8 MHz.

Fusible	Valeur
Low	0xe2
High	0xde
Extended	0xf5

En [3], vous trouverez les instructions détaillées permettant d'utiliser l'ATmega328PB avec l'environnement de programmation intégré (IDE) Arduino.

Connectez un bouton-poussoir en J2. Il servira de bouton de réinitialisation. Connectez un adaptateur série-USB en J1. Pour téléverser un sketch (croquis) dans le microcontrôleur, cliquez sur le bouton Upload de l'IDE comme d'habitude, tout en vérifiant la sortie sur la fenêtre de transfert. Lorsque le message "Uploading..." apparaît dans l'IDE, appuyez sur le poussoir de réinitialisation de la carte adaptatrice. Le sketch devrait alors être téléversé comme à l'accoutumée. Appuyez une fois de plus pour démarrer l'exécution du nouveau programme.

Les fichiers de conception du circuit-imprimé et la documentation du code source sont disponibles à la page de ce projet sur le site Elektor Labs [4] ◀

240196-04



Produits

➤ **HU-017A RDA5807 FM Radio Kit**
www.elektor.fr/20866

➤ **C. Valens, "Mastering Microcontrollers Helped by Arduino" (Elektor, 2017)**
www.elektor.fr/17967



VOUS AVEZ DE LA CHANCE !



TÉLÉCHARGEZ
GRATUITEMENT

Un abonné à l'e-zine ne manque jamais
le mensuel «reverse project»

Vous n'êtes pas encore abonné ? Inscrivez-vous à
notre e-zine gratuit à elektormagazine.fr/ezine-24

