

18 générateur de **code morse**

utilisez-le comme balise ou dispositif d'apprentissage !

Rob van Hest (Pays-Bas)

Construisez un projet d'émetteur de code morse personnalisable à l'aide d'un minuscule microcontrôleur. Envoyez des messages préprogrammés tout en ajustant la vitesse et la fréquence. Parfait pour les émetteurs de balises ou les chasses au renard.

La plupart des lecteurs connaissent probablement le code Morse [1] grâce aux bips que l'on peut entendre sur les bandes d'ondes courtes ou dans les films. Certains ont peut-être même eu « la chance » d'apprendre le morse pour transmettre et recevoir des messages afin d'obtenir une licence radio. Mais pourquoi en aurait-on encore besoin de nos jours ? Il existe de nombreuses alternatives.

En effet, il y en a. Pourtant, il y a des raisons d'envisager l'utilisation du morse, notamment la simplicité du matériel nécessaire. Un émetteur se compose essentiellement d'un oscillateur RF qui est activé et désactivé en appuyant sur la touche. Cela se fait facilement avec un émetteur à onde entretenue. L'appareil présenté ici peut piloter un tel émetteur. Vous pouvez l'utiliser pour un émetteur de balise ou pour une chasse au renard. Cependant, mon objectif initial était de faire déchiffrer un message codé en morse.

Matériel

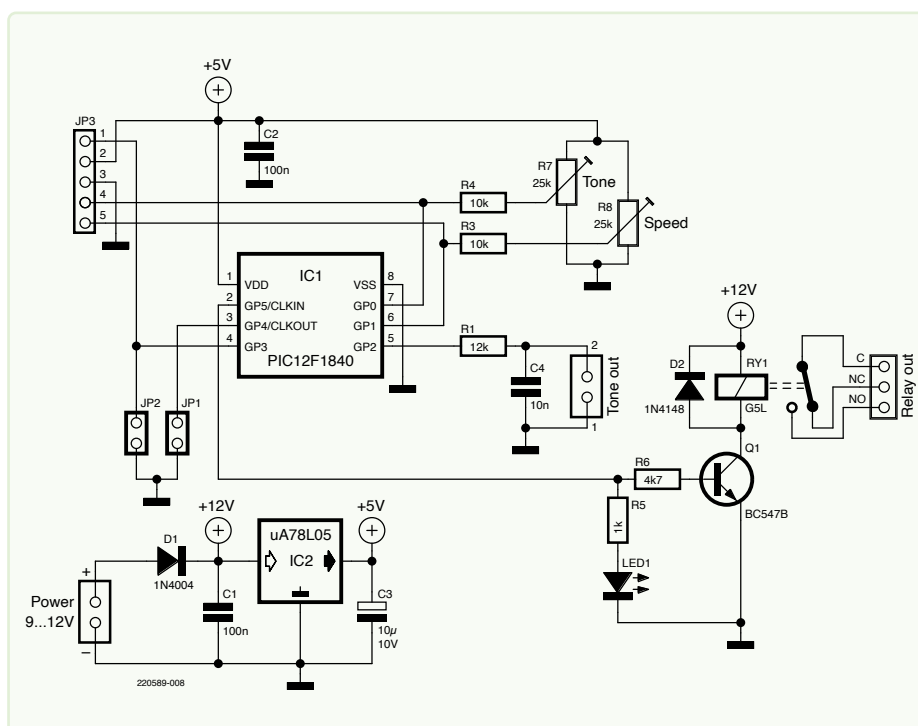
Le schéma de l'appareil est présenté dans la **figure 1**. En son cœur se trouve l'IC1, un petit microcontrôleur PIC12F1840. La broche 2 commande un relais qui peut contrôler un émetteur RF, par exemple. La broche 5 est la sortie du signal audio du code morse. Le filtre passe-bas R1/C4 transforme cette onde carrée modulée en quelque chose de plus sinusoïdal pour un son plus agréable. Il est également

possible de connecter un haut-parleur ou un casque directement à la broche 5, mais vous pouvez avoir besoin d'une résistance en série pour limiter le courant. Le courant de sortie maximum est de 25 mA ; l'impédance doit donc être d'au moins 200 Ω .

Le potentiomètre R8 permet de régler la vitesse de signalisation, tandis que R7 contrôle la fréquence.

L'appareil peut envoyer un message préprogrammé de 32 caractères maximum à intervalles fixes. L'EEPROM du microcontrôleur peut contenir jusqu'à trois de ces messages que vous pouvez adapter à vos propres besoins (voir ci-dessous). Les cavaliers JP1 et JP2 déterminent lequel des trois messages sera envoyé.

Sans JP1 et JP2, le générateur écoute son entrée série sur la broche 6, où vous pouvez connecter l'appareil à un ordinateur pour qu'il convertisse (ou stocke) les messages que vous tapez dans un programme de terminal série en code Morse.



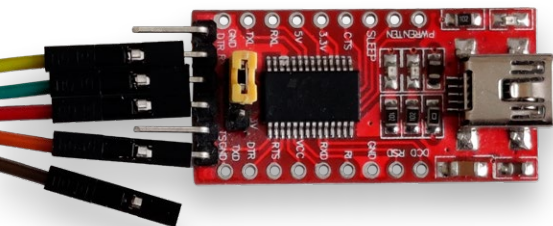


Figure 2. Avec un adaptateur USB-série, vous pouvez taper des messages sur un ordinateur et utiliser le générateur de code morse pour les transformer en morse.

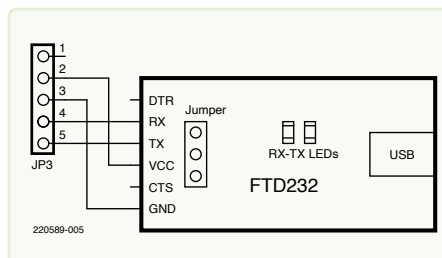


Figure 3. Schéma montrant comment connecter un convertisseur USB-série à JP3.



Figure 4. Le générateur de code morse construit sur la carte universelle pour projet PIC à huit broches de l'auteur.

Logiciel

Le programme associé peut être téléchargé à partir de la page du projet sur Elektor Labs [2]. La source et le fichier HEX compilé y sont disponibles. Si vous souhaitez modifier le programme vous-même, vous aurez également besoin du compilateur C CC5X de B. Knudsen [3].

Alors qu'une interface de programmation PICKit de Microchip peut être connectée à JP3, vous pouvez également connecter un PC ou un terminal via un convertisseur approprié. J'ai utilisé à cet effet un convertisseur USB-série de type FTD232 (figure 2). Pour le schéma de câblage, voir la figure 3. Un programme de terminal doit être exécuté sur le PC. J'utilise PuTTY [5] pour cela, mais il existe d'autres programmes adaptés. Choisir le port série virtuel que le logiciel pilote crée (pour moi, c'était COM6) et régler la vitesse sur 9600 baud.

Des éléments tels que la fréquence, la vitesse et les messages prédéfinis peuvent également être configurés via le port série. Tapez 'H' ou '?' dans le terminal série pour obtenir un aperçu des commandes disponibles. Une description plus détaillée est disponible sur la page du projet [2].

Remarques complémentaires

Bien que la vitesse de signalisation et la fréquence soient contrôlables à l'aide des deux potentiomètres, les paramètres configurables par l'utilisateur stockés dans l'EEPROM peuvent aussi être utilisés à cette fin. Pour ce faire, tournez R8 au maximum et réinitialisez l'appareil. Notez que les potentiomètres ne fonctionnent pas lorsqu'un dispositif USB-série est connecté à JP3.

Le générateur de code morse (figure 4) a été construit sur un circuit imprimé universel (PCB) [4]. J'utilise ce circuit imprimé pour toutes sortes de projets avec le même type de microcontrôleur PIC à 8 broches.

R4, R7 et R8 peuvent être omis s'ils ne sont pas nécessaires. Dans ce cas, connectez R3

à +5 V. Si le circuit imprimé de [4] est utilisé, R3 peut également être omis et SJ4 du circuit imprimé (pas dans le schéma) est ponté. Lorsqu'un convertisseur USB-série est utilisé, une alimentation séparée n'est plus nécessaire. Dans ce cas, le relais doit être remplacé par une version 5 V. Pontez aussi l'entrée et la sortie de IC2. Bien entendu, si seule la sortie audio est nécessaire, le relais peut être omis. ◀

VF : Maxime Valens — 220589-04

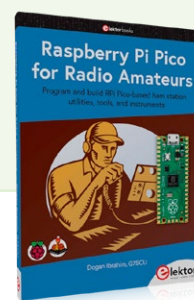
Des questions ?

Envoyez un courriel à l'auteur (trainer99@ziggo.nl) ou contactez Elektor (redaction@elektor.fr). team at editor@elektor.com.



Produits

- ▶ Bert van Dam, 50 PIC Microcontroller Projects (E-book) <https://elektor.fr/18091>
- ▶ Dogan Ibrahim, Raspberry Pi Pico for Radio Amateurs <https://elektor.fr/20041>



LIENS

- [1] Code Morse [Wikipedia] : https://fr.wikipedia.org/wiki/Code_Morse_international
- [2] Ce projet sur Elektor Labs : <https://www.elektormagazine.fr/labs/morse-code-generator>
- [3] B. Knudsen Data, CC5X compiler : <https://bknd.com/cc5x>
- [4] PuTTY : <https://putty.org>
- [5] Circuit imprimé universel pour PIC à 8 broches : <https://www.elektormagazine.fr/labs/board-for-simple-microcontroller-project>