



Image generated with DALL-E

07 une boîte à meuh avec un microcontrôleur

produire des sons amusants à l'aide d'un microcontrôleur

Bruno Clerc (France)

Les boîtes à meuh ont résisté à l'épreuve du temps. Fabriquons-en une ! Cet article présente un projet de fabrication artisanale utilisant un microcontrôleur ATtiny85 et très peu de matériel. Il permet de reproduire le son d'une vache lorsque la boîte est retournée, en mettant l'accent sur le processus de recherche, de modification et de programmation du meuglement « parfait » dans la mémoire de l'ATtiny.

Ceux qui connaissent mon travail commenceront à voir un début de solution. J'ai promis à ma petite-fille Paolina de faire autre chose qu'un bracelet avec le reste du tube en carton utilisé pour la boîte à musique à manivelle [1]. Mais que faire avec ? Un jeu tactile « *Simon Says* » ?



Figure 1. La boîte à meuh complète.

Un jeu de morpion ? Fabriquons plutôt une boîte à meuh, ces petites boîtes que l'on trouve dans les boutiques de souvenirs et qui imitent le meuglement d'une vache. Le résultat final est illustré à la **figure 1**. Le petit pot en verre à côté de la boîte sera utilisé pour une deuxième version miniaturisée.

Les composants nécessaires

- Un ATtiny85
- Un interrupteur à bascule, comme un interrupteur Tilt
- Un petit haut-parleur de 8 Ω
- Une batterie rechargeable, dans ce cas une cellule type 18650 provenant d'un ordinateur portable
- Un chargeur lithium basé sur le TP4056, ici un HW373

L'idée

Dans le même esprit que mon mini-piano ou ma boîte à musique à manivelle, je vais utiliser un ATtiny85, qui devrait être capable de lire un fichier au format hexadécimal et de restituer le son à travers un haut-parleur pour générer le meuglement. En cherchant sur internet différentes stratégies, pour faire en sorte qu'un microcontrôleur génère un son en utilisant du matériel minimaliste, je suis tombé sur l'excellent site Technoblogy [2]. Cette page présente un projet intéressant dans lequel un simple ATtiny85 est programmé pour émettre un court échantillon de son numérique par l'intermédiaire d'un haut-parleur. Il est possible de jouer un échantillon d'une seconde de n'importe quel son numérisé en utilisant uniquement un haut-parleur et sans aucun autre composant additionnel. Parfait ! Exactement ce dont j'avais besoin.

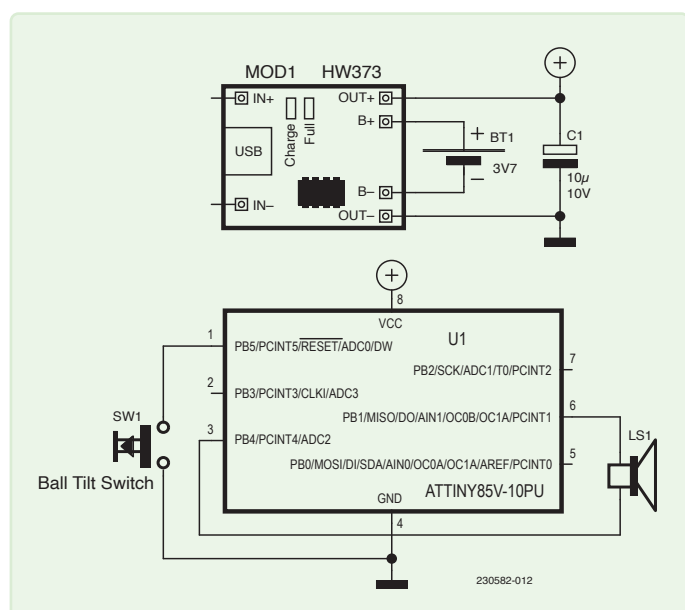


Figure 2. Le schéma de principe.

Cette conception exploite l'option spéciale de la fréquence d'horloge cadencée à 64 MHz de l'ATtiny85, que vous pouvez utiliser pour piloter `Timer/Counter1` pour une conversion numérique-analogique rapide. Pour commencer, j'ai programmé le microcontrôleur avec le programme d'exemple de Technoblogy. Le programme fonctionne bien et un bruit de canard a été généré par le microcontrôleur. Le diagramme schématique est montré dans la **figure 2**.

Trouver le « meuh » parfait

Je me suis mis à la recherche d'un son approprié pour le projet, à savoir un meuglement de vache. Heureusement, grâce à la magie d'Internet, j'ai trouvé un son « Meuh » sur le site de La Sonothèque [3]. Vous y trouverez également un certain nombre de bruits différents qui vous permettront de reproduire le projet avec des variations créatives. Pourtant, comme le mentionne Technoblogy, il est nécessaire de convertir le fichier audio.

Dans l'enregistrement que j'ai trouvé, il y avait une succession de plusieurs « meuh », ce qui est plus que nécessaire. Le fichier a été ouvert avec Audacity [4], puis ramené à un seul meuglement. Le fichier a ensuite été sauvegardé au format WAV en 8 bits. Pour convertir le fichier WAV en un format texte compatible avec le langage C, vous pouvez par exemple utiliser la commande `xxd` sur un Mac ou un système Linux. Pour ma part, j'ai utilisé l'utilitaire `HxD utility` [5] pour ouvrir le fichier WAV et extraire les valeurs. C'est une méthode moins pratique que l'utilisation de la commande `xxd`, car le format des valeurs provenant de HxD n'est pas directement utilisable dans le code C/Arduino, et il faut insérer des `0x` et des virgules dans le tableau ainsi obtenu. Une fois cette tâche fastidieuse accomplie, j'ai dû ajuster la taille de l'enregistrement, en supprimant autant que possible les parties muettes, en réduisant au maximum la fin de la piste, en supprimant manuellement quelques dernières données du tableau, etc.

Pour obtenir un effet sonore acceptable tout en exploitant au mieux les capacités de l'ATtiny85, j'ai dû faire de multiples essais. J'ai effectué plusieurs manipulations de fichiers WAV et des téléchargements de code de test, facilités par EasyTinyProg [6]. Le code final utilise 8104 octets (98%) de l'espace de stockage du programme. Le maximum est de 8192 octets. Il ne reste pratiquement plus de place !

Le principe

Vous pouvez trouver le code du projet sur *Elektor Labs* [7]. Le code utilise la boucle à verrouillage de phase (PLL) à 64 MHz et utilise cette référence de fréquence pour la fonction `Timer1`. Avec cette configuration, l'interruption ISR (`TIMER0_COMPA_vect`) se produit à une fréquence d'environ 512 820 Hz. Le programme utilise `OCR1A` et `OCR1B`, qui sont les registres de comparaison des canaux A et B du `Timer1`. De cette façon, deux signaux PWM complémentaires peuvent être générés à partir de la même valeur d'échantillon audio, ce qui est parfait pour alimenter directement un haut-parleur, car les signaux `OC1A` et `OC1B` sont de polarité opposée, comme dans un circuit amplificateur push-pull.

La lecture du tableau

Le code lit le tableau, une cellule à la fois, à travers une boucle. Pour savoir où vous vous trouvez dans le tableau, deux variables sont utilisées, avec `readWavLen` pour stocker l'index actuel et `mheuWavLen` pour stocker la taille totale de l'échantillon. Chaque fois que la routine d'interruption (ISR) est exécutée, l'instruction `if(readWavLen == mheuWavLen)` est utilisée. Si cette instruction renvoie la valeur `True` (c'est-à-dire si nous avons atteint la fin du tableau), le code désactive les interruptions et met le microcontrôleur en mode Veille pour économiser de l'énergie.

Dans notre exemple, après plusieurs essais et erreurs pour remplir au mieux la mémoire disponible, le tableau comporte 7677 cellules. Comme j'ai commencé avec un échantillon de longueur inconnue et que je l'ai modifié manuellement, je n'avais aucune connaissance précise de sa longueur. Il existe plusieurs stratégies possibles pour déterminer ce nombre. Par exemple, vous pouvez demander à un programme comme Notepad++ de compter les occurrences de « `0x` ». Pour ma part, je me suis amusé à utiliser un Arduino Uno supplémentaire, ainsi que le code `Taille_tableau_mheu.ino`, et le port série. Vous trouverez une explication complète du code Moo Box dans le fichier `ATTINY_MHEU_WAV.ino`.

La construction de la boîte à meuh

La construction de la boîte à meuh est simple. Les composants qui sont assemblés selon le schéma, sortis de leur emballage, sont présentés

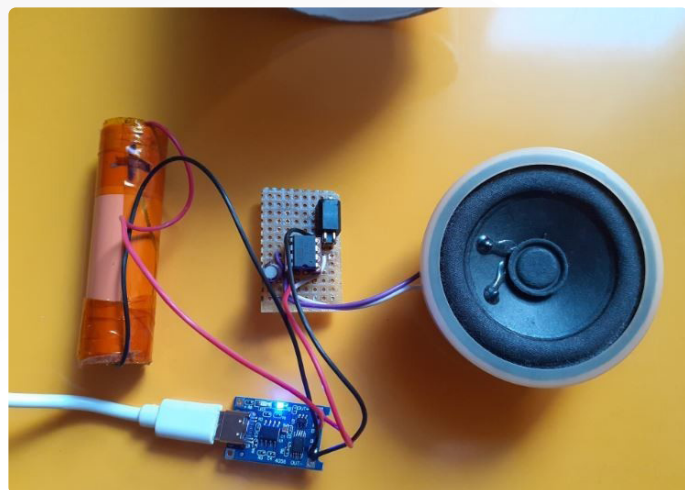


Figure 3. Tout est câblé, à l'extérieur du bocal.

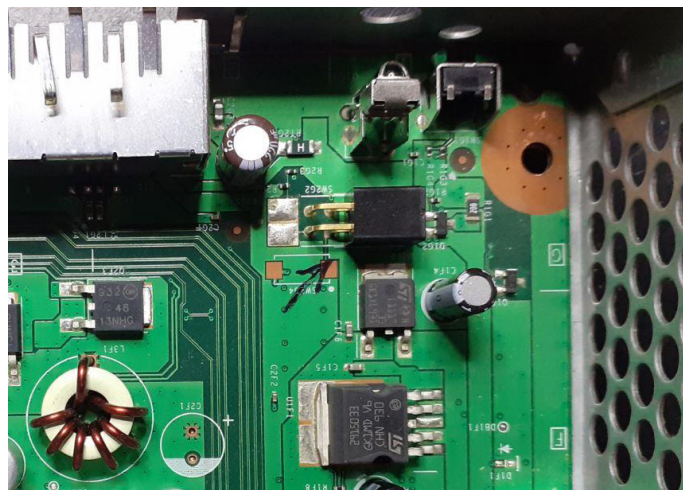


Figure 4. Réutilisation d'un interrupteur à bascule Tilt.

à la **figure 3**. La batterie et le chargeur sont identiques à ceux utilisés dans la boîte à musique à manivelle. L'ensemble est alimenté avec une tension de 3,7 V, mais peut accepter une tension de 5 V. En guise de boîtier, j'ai récupéré le morceau de tube en carton qui restait du prototype de boîte à musique à manivelle de l'article précédent. J'ai utilisé un interrupteur Tilt pour détecter lorsque l'ensemble est retourné à l'envers. Dans ce cas, il provient d'une vieille console Xbox qui allait être recyclée (**figure 4**). Le haut-parleur provient de ma collection personnelle, qui est le résultat de nombreux objets démontés par le passé, et la cellule au lithium 18650 provient d'une vieille batterie d'ordinateur portable. Le circuit imprimé est monté verticalement avec de la colle thermofusible. En position verticale, la bille du capteur est en bas et ne touche pas les contacts, qui se trouvent en haut. Lorsque la boîte est retournée, la bille ferme le circuit. Les fils de ce capteur sont connectés directement à la broche Reset de l'ATtiny et à la masse, ce qui maintient le microcontrôleur en état de réinitialisation. Ainsi, lorsque vous retournez la boîte, la bille redescend et ouvre le contact de réinitialisation. Le microcontrôleur démarre, la boîte émet un « meuh » comme prévu et se remet en veille pour préserver la batterie. Sauf, si l'on recommence ? Une vidéo de démonstration est disponible [8].

Amélioration : une version miniaturisée

Je me suis lancé le défi d'en faire une deuxième version, plus petite. Et c'est chose faite, les dimensions finales sont un diamètre de 50 mm, une hauteur 44 mm et un poids de 38 g. Le boîtier est fait du petit bocal visible sur la première photo. Le haut-parleur provient d'un vieux casque Bluetooth, tandis que la grille métallique provient d'un vieux PC. Pour le microcontrôleur, j'ai encore quelques circuits imprimés que j'ai réalisés pour le projet *Button-Free Door Control* publié il y a deux ans [9]. L'ATtiny85 et son circuit imprimé tiennent dans un support de pile pour CR2032. Un second support, équipé cette fois d'une vraie pile CR2032, alimente l'ensemble (**figure 5**). Le résultat final est montré à la **figure 6**, comme dans la vidéo [10]. ◀

VF : Laurent Rauber — 230582-04

Questions ou commentaires ?

Envoyez un courriel à l'auteur (b.clerc31@laposte.net) ou contactez elektor (redaction@elektor.fr).

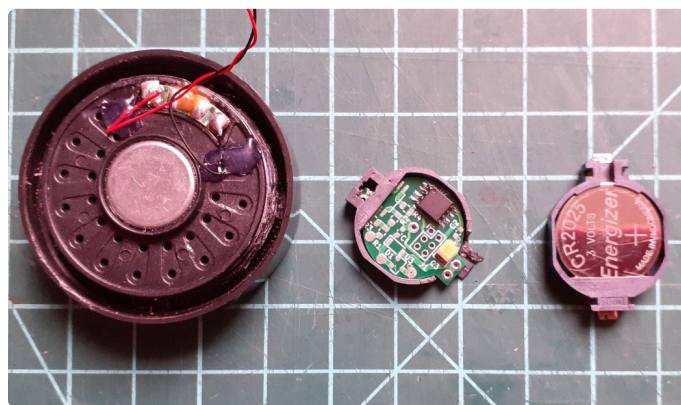


Figure 5. Réduisons la taille.



Figure 6. Version miniature terminée.



Produits

- **Elektor circuits de vacances 2022**
www.elektor.fr/products/elektor-summer-circuits-2022-fr
- **M. Cina, MSP430 Microcontroller Essentials (Elektor 2022)**
www.elektor.fr/20112

LIENS

- [1] Bruno Clerc, "A Tiny Music Box," Elektor Circuit Special 2024: <https://elektormagazine.fr/230505-04>
- [2] Technoblogy : lecteur d'échantillons audio avec un ATtiny : <http://www.technoblogy.com/show?QBB>
- [3] Certains sons de meuglement, et bien d'autres : <https://lasonotheque.org/search?q=meuh>
- [4] Audacity: <https://audacityteam.org>
- [5] HxD hex editor : <https://mh-nexus.de/en/hxd/>
- [6] EasyTinyProg : <https://www.elektormagazine.fr/labs/make-programming-the-attiny-dip8-easier-with-easytinyprog>
- [7] Ce projet sur Elektor Labs : <https://www.elektormagazine.fr/labs/moo-box-or-boite-a-mheu>
- [8] Vidéo de démo 1 : https://youtu.be/DO0_t3U06aI
- [9] Télécommande comodo : <https://www.elektormagazine.fr/magazine/elektor-264/60844>
- [10] Vidéo de démo 2 : <https://www.youtube.com/watch?v=TkjpJaAW-wY>