

réduction de la consommation d'énergie de votre repousse-taupes

Grâce au remplacement du 555 par un ATtiny13



Gerhard Dürr (Allemagne) et Luc Lemmens (Elektor)

Les taupinières peuvent ruiner une pelouse entretenue avec amour en un rien de temps. Si vous protégez votre précieux jardin avec un repousse-taupes électronique, cette mise à niveau matérielle peut multiplier par dix la durée de vie de ses piles !

Si vous avez un jardin avec une pelouse, surtout si vous habitez à la campagne, vous courez le risque d'être désagréablement surpris par la partie visible de l'activité des taupes : les taupinières. La bonne nouvelle c'est que la terre de votre jardin est sans doute propre et fertile. Vous ne verrez qu'exceptionnellement les taupes elles-mêmes, qui ne quittent que rarement les galeries qu'elles creusent en quête de vers et d'insectes, leur menu favori. La terre qu'elles déplacent pendant qu'elles creusent doit bien sûr aller quelque part, et elles la poussent vers le haut, formant ainsi les monticules qui signent leur présence. C'est tellement peu apprécié que de nombreuses méthodes ont été mises au point pour leur barrer la route ou pour les déloger une fois installées. L'une des solutions est ce que l'on appelle le repousse-taupes : un appareil qui produit des vibrations (sonores) pour empêcher ou mettre

fin à ces activités souterraines. Il s'agit d'un tube pointu que l'on insère dans le sol, pourvu à son extrémité d'une source sonore (généralement un vibreur électronique à courant continu) censée chasser les taupes et autres vermines comme les campagnols et les rats.

Certains fabricants prétendent que les répulsifs antiparasites qu'ils fabriquent produisent des vibrations *ultrasoniques*. Peut-être en existe-t-il, mais les spécifications de la plupart des appareils que nous avons évalués précisent qu'ils produisent des sons dont les fréquences se situent dans la zone inférieure des kilohertz ou même plus bas, bien en dessous de la gamme des ultrasons. Dans la plupart des cas, le terme repousse-taupes sonore (sans ultra) sera plus approprié. Le son est émis pendant environ une demi-seconde, à des intervalles de quelque 30 s. Ces impulsions seraient si désagréables pour une

taupe qu'elles la feraient décamper et/ou rester à distance. Certaines personnes sont absolument convaincues de l'efficacité de ces appareils, d'autres prétendent qu'il s'agit d'une arnaque qui ne profite qu'aux seuls fabricants et revendeurs ; nous ne nous prononcerons pas sur ce point. Quelques réflexions sur l'éloignement des taupes en général sont présentées dans l'encadré. Dans cet article, nous n'aborderons qu'un moyen de réduire la consommation d'énergie d'une variante produisant du son et fonctionnant sur piles.

Comment ça marche

Ce type de répulsif contient un circuit minuteur et une sorte de sonnette constituée le plus souvent d'un vibreur piézoélectrique, mais nous avons également trouvé des dispositifs avec des moteurs vibrants, comme ceux utilisés dans les téléphones portables. En électronique, « circuit minuteur » est synonyme de « 555 », peut-être le plus célèbre des circuits intégrés, qui vient tout juste de fêter son 50^e anniversaire. Il y a toujours des exceptions, mais le concepteur du circuit proposé ici a également découvert un appareil dans lequel le 555 est utilisé comme multivibrateur bistable. La majeure partie de l'énergie sert à produire le son, mais il en faut un peu pour maintenir le 555 actif lorsque le vibreur est silencieux. Bien qu'une modification de ce circuit puisse procurer une économie d'énergie et donc prolonger la durée de vie des piles (par exemple, en remplaçant un 555 standard par une version CMOS ou en modifiant les valeurs de ses composants externes), le concepteur a choisi une approche différente. Le circuit complet de la minuterie est remplacé par un petit microcontrôleur qui reste en mode veille la plupart du temps ; sa minuterie de chien de garde le réveille environ toutes les 30 s et fait retentir le vibreur pendant une seconde, puis le repasse en mode veille. La consommation de courant est réduite de près d'un facteur dix : de 4 mA avec le circuit original de la minuterie à 555 à une moyenne de 430 μ A avec le microcontrôleur, soit une durée de vie des piles portée de moins de 200 jours à environ 1450 jours ! En pratique, cette durée sera plus courte, en raison de facteurs tels que l'autodécharge des piles, mais il s'agit tout de même d'une grande amélioration par rapport à la consommation de courant du circuit original.

Le nouveau matériel

Le schéma du circuit à microcontrôleur est représenté sur la **figure 1**. IC1, un ATtiny13 de Microchip (anciennement Atmel), est le cœur du circuit de la minuterie. Sa mémoire de programme interne peut être flashée avec une interface de programmation ISP connectée à K2. Si le microcontrôleur est programmé avant d'être inséré dans son support, on peut omettre ce port à six broches.

Le circuit est alimenté par des piles, dans ce cas quatre piles de type D (LR20) en série – c'est-à-dire 6 V au total – connectées entre K1 et K3 (les bornes positive et négative, respectivement). Avec quatre piles de 1,5 V neuves, à vide ou avec une charge très faible, la tension, de l'ordre de 6,5 V, dépasse la tension d'alimentation maximale absolue de 5,5 V de l'ATtiny13. D1 est une LED rouge standard (de préférence pas un type à haut rendement) qui a une tension directe d'environ 1,3 V, même à un courant direct aussi bas que 1 μ A ; la tension d'alimentation du microcontrôleur s'en trouve réduite à une valeur sûre, bien en dessous du maximum de 5,5 V. D1 peut aussi être constituée de deux ou trois diodes standard 1N4148 en série. Un régulateur de tension linéaire n'est pas une bonne solution, car il ajouterait sa propre consommation. C1 et C2 sont des condensateurs de découplage pour l'alimentation du microcontrôleur.

Le vibreur est récupéré du matériel d'origine et connecté aux broches

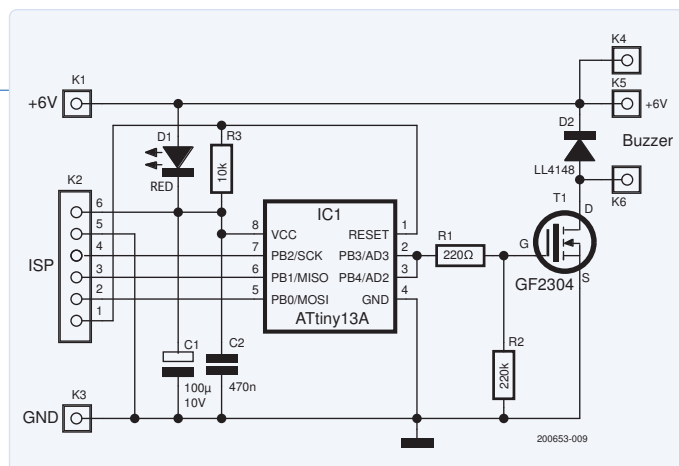


Figure 1. Le schéma du circuit de mise à niveau du minuteur.

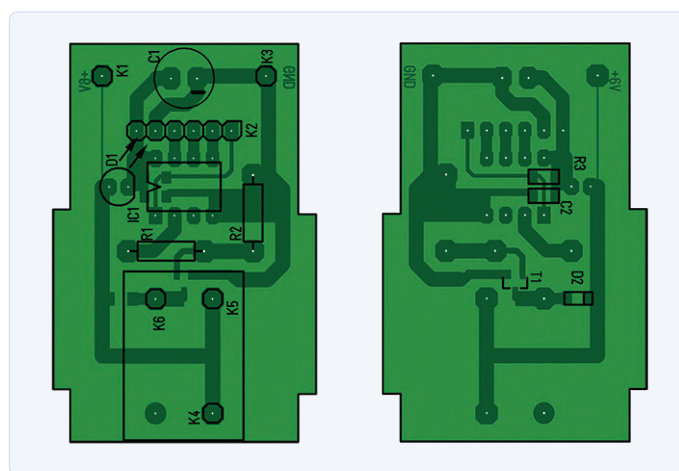


Figure 2. Implantation sur le circuit imprimé.

K4/K5 (positif) et K6 (négatif). Certains vibreurs sont inductifs et susceptibles de produire des pics, notamment lors de la mise hors tension. D2 est une diode de roue libre qui protège le transistor T1 contre ces pics. Le choix de ce transistor de commutation n'est pas du tout critique. Le courant traversant le vibreur étant inférieur à 10 mA, un BC547 standard peut convenir avec 4,7 k Ω pour R1 et 10 k Ω pour R2. N'importe quel NMOS-FET à niveau logique fera également l'affaire, en remplaçant R1 par un fil de liaison et en supprimant R2.

Il y a même un circuit imprimé

Le schéma a été conçu avec une version gratuite de Target3001! et l'auteur a même réalisé un petit circuit imprimé pour le passage au microcontrôleur (**fig. 2**). Les fichiers de conception sont disponibles en téléchargement sur la page Elektor Labs de ce projet [1]. Compte tenu du grand nombre de marques différentes de repousse-taupes disponibles sur le marché, ce serait une pure coïncidence si, comme sur les photos des auteurs (**fig. 3**), le circuit imprimé s'adaptait exactement au vôtre. Si vous désirez modifier le circuit imprimé, voilà une belle occasion de vous faire la main sur Target3001! avec ce projet relativement simple, mais refaire le schéma et le circuit imprimé avec un autre programme de CAO ne serait pas plus compliqué. Il y a quelques composants CMS soudés sur la face inférieure en cuivre du circuit imprimé qui peuvent facilement être remplacés par des composants traversants, même sans changer le circuit imprimé.



Figure 3a. Vue d'ensemble.

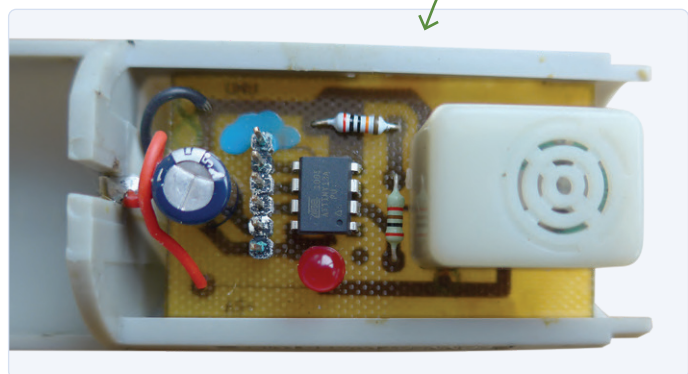


Figure 3b. Le nouveau circuit imprimé installé dans le repousse-taupes initial.

Rien ne fonctionne sans micrologiciel

Le code source (**listage 1**) pour l'ATtiny13 est développé sous AVR Studio 4.19 avec le compilateur AVR-GCC installé. Bien que cet EDI soit dépassé, il est toujours disponible en téléchargement, mais il ne sera pas difficile d'adapter le code source à des outils de développement plus récents. Le code source, le projet AVR Studio et le fichier HEX peuvent être téléchargés depuis la page Elektor Labs du projet [1]. Ce dernier vous permet de flasher la mémoire de programme de l'ATtiny13 sans avoir à compiler. Les fusibles de configuration de ce

microcontrôleur resteront à leur valeur d'usine (c'est-à-dire qu'ils n'ont pas besoin d'être programmés).

Le programme est essentiellement constitué d'instructions en assembleur insérées dans un module écrit en C. Après la mise sous tension, les broches GPIO sont configurées ; les minuteries, le convertisseur A/N, les résistances de rappel et le comparateur analogique sont désactivés et le mode veille est réglé sur *Powerdown*. Le code entre ensuite dans une boucle sans fin (`while(1)`), qui effectue les opérations suivantes :

- Le vibreur est activé.
- Le chien de garde est réglé pour produire une interruption après 128k cycles, avec une horloge de 128 kHz, cela donne un délai d'interruption d'une seconde (environ).
- L'interruption est activée et l'ATtiny passe en mode *Powerdown* avec la commande assembleur `sleep`.
- Après une seconde, l'interruption du chien de garde réveille le microcontrôleur de son état *Powerdown*
- Le vibreur est éteint et l'interruption du chien de garde est portée à son intervalle maximum de 1024k cycles, c'est-à-dire 8 s. Comme nous voulons que le repousse-taupes soit silencieux pendant environ une demi-minute, le chien de garde doit produire trois interruptions avant que le `while(1)` ne recommence avec l'activation du vibreur.

La fin des taupes ?

Cette mise à niveau du repousse-taupes prolonge la durée de vie des quatre piles de type D, mais ne modifie pas l'efficacité de l'appareil lui-même. Comme nous l'avons mentionné, nombreux sont les débats entre ceux qui croient en l'efficacité des repousse-taupes et ceux qui pensent ou même sont convaincus que leur effet est nul. La profusion de marques et de revendeurs suggère que beaucoup de gens y croient ou s'interrogent sur leur efficacité. En tout cas, c'est un remède qui ne nuit pas à l'environnement. La chose la plus négative que nous ayons trouvée, ce sont des témoignages de voisins gênés par les sons doux et répétitifs de ces répulsifs. ❏

200653-04

Des taupes dans votre jardin ? Que faire ou ne pas faire ?

Si vous cherchez un remède sur l'internet contre les taupes dans votre jardin ou pour leur barrer l'accès à votre propriété, vous serez surpris de voir tout ce qui a été inventé au fil des ans pour leur pourrir la vie. Les experts ne donnent pas cher de l'efficacité de nombre de ces solutions, mais vous pourrez en être certain si votre magnifique pelouse est soudain défigurée par des taupinières.

Nous ne sommes pas du tout des experts en taupes, mais, d'après les informations dont nous disposons, le mieux est de ne rien faire du tout. S'il y a des taupinières, c'est qu'une taupe a déjà creusé son terrier et celui-ci ne s'agrandit généralement que lorsqu'il n'y a plus assez de nourriture. Si ces mini-montagnes vous dérangent, la meilleure chose à faire est d'enlever la terre avec précaution à l'aide d'une pelle. Si vous les cognez ou les aplatissez, il y a de fortes chances que l'animal répare l'effondrement de ses galeries en produisant de nouvelles taupinières.



LISTE DES COMPOSANTS

Résistances

R1 = 220 Ω
R2 = 220 kΩ
R3 = 10 kΩ

Condensateurs

C1 = 220 µF/25V
C2 = 470 nF

Semi-conducteurs

D1 = LED rouge
D2 = LL4148
T1 = GF2304 (voir texte)
IC1 = ATtiny13A (style DIL-8)

Divers

K2 = barrette à 6 broches

Contributeurs

Conception : Gerhard Dürr
Texte : Gerhard Dürr,
Luc Lemmens
Illustrations : Gerhard Dürr,
Patrick Wielders, Luc Lemmens
Rédaction : Jens Nickel, C. J. Abate
Mise en page : Giel Dols
Traduction : Helmut Müller

Des questions, des commentaires ?

Contactez Elektor
(redaction@elektor.fr).



PRODUITS

➤ **Programmeur avec accessoires EXA-Prog de Diamex**
www.elektor.fr/19912



Listage 1. Code source complet.

```
// Controlling the molechaser-buzzer with the watchdog-timer
// * MC ATtiny13A, all fuse bytes default
// * PB3 and PB4 HIGH: buzzer ON
// * Configurable with watchdog-timer-prescaler
// * Selected: 128K-cycles(~1s) ON, 3*1024K-cycles(~24s) OFF
// * Current consumption: powerdown ~6µA, buzzing ~7.2mA, average 0.43mA
#include <avr/io.h>
#ifndef F_CPU
#warning "Defining F_CPU 1.2MHz"
#define F_CPU 1200000UL
#endif
#include <avr/interrupt.h>
int main(void) {
    uint8_t i=0;
    DDRB = 1<<PB0|1<<PB1|1<<PB2|1<<PB3|1<<PB4;
    PRR = 1<<PRTIM0|1<<PRADC; // Power Reduction Timer0, ADC
    MCUCR = 1<<PUD|1<<SE|1<<SM1|0<<SM0; // Pullup Disable, Sleepmode: Powerdown
    ACSR = 1<<ACD; // Analog Comparator Disable
    while(1) {
        i = 0;
        PORTB |= 1<<PB3|1<<PB4; // buzzer ON
        // change Watchdog-Interrupt to 1s (measured: 1.14s @ 4.5V, 22°C)
        MCUSR = 0; // clear Reset-Flags
        WDTCSR = 1<<WDCE |0<<WDE; // Watchdog Change Enable
        // Watchdog Timer Prescaler: 128K-cycles(~1s), Watchdog Timer Interrupt Enable
        WDTCSR = 1<<WDTIE|0<<WDE|1<<WDP2|1<<WDP1;
        sei();
        asm volatile("sleep");
        asm volatile("wdr"); // Watchdog Reset, wait for WDT-ISR(~1s)
        PORTB &= ~(1<<PB3|1<<PB4); // buzzer OFF
        // change Watchdog-Interrupt to 3*8s: (measured: 27.9s @ 4.5V, 22°C)
        asm volatile("wdr"); // Watchdog Reset
        MCUSR = 0; // clear Reset-Flags
        WDTCSR = 1<<WDCE|0<<WDE; // Watchdog Change Enable
        // Watchdog Timer Prescaler: 1024K-cycles(~8s), Watchdog Timer Interrupt Enable
        WDTCSR = 1<<WDTIE|0<<WDE|1<<WDP3|1<<WDP0;
        while(i < 3) { // wait 24s
            i++;
            asm volatile("sleep");
            asm volatile("wdr"); // Watchdog Reset, wait for WDT-ISR(~8s)
        }
    }
}
EMPTY_INTERRUPT(WDT_vect); // awaking from powerdown-mode
```

LIEN

[1] Ce projet sur Elektor Labs : <https://bit.ly/3JWCXBk>