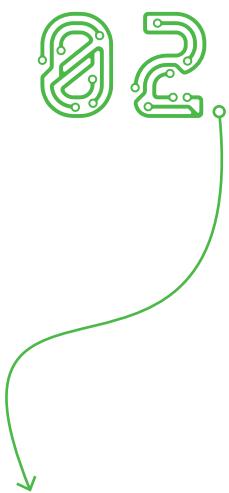


Construire un gradateur numérique 220-V CA avec Arduino



Hesam Moshiri (Iran)

Les charges électroniques CA sont partout autour de nous puisque la plupart des appareils domestiques sont alimentés par la tension du secteur. Par conséquent, nous nous trouvons souvent dans des situations où nous avons besoin d'un contrôle total (gradation) sur une charge CA telles qu'une lampe, un moteur ou un aspirateur.

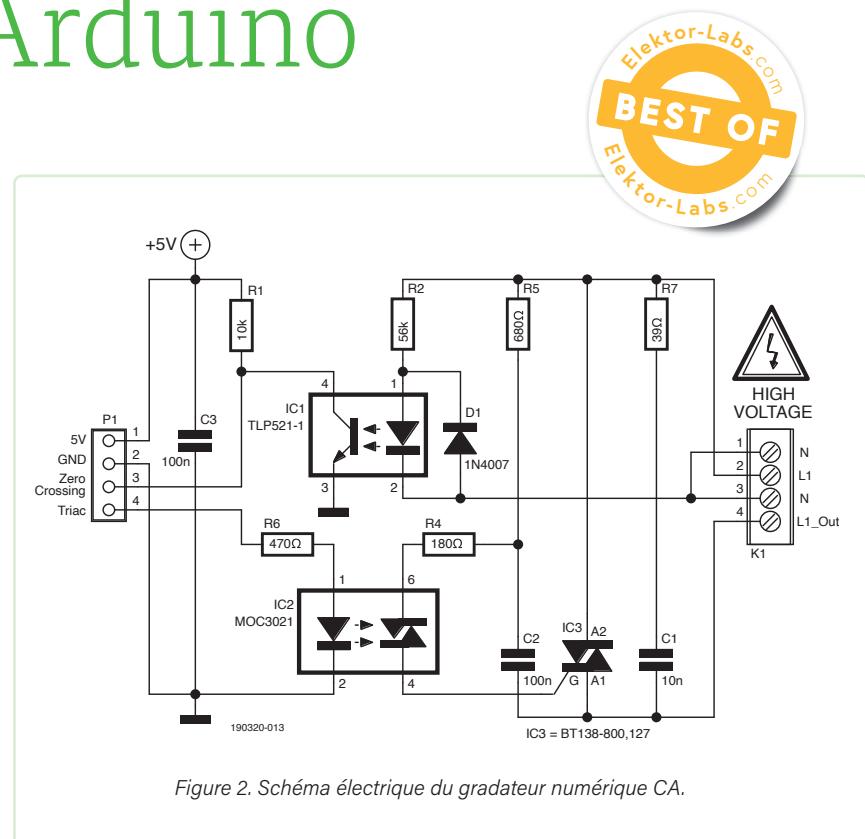


Figure 2. Schéma électrique du gradateur numérique CA.

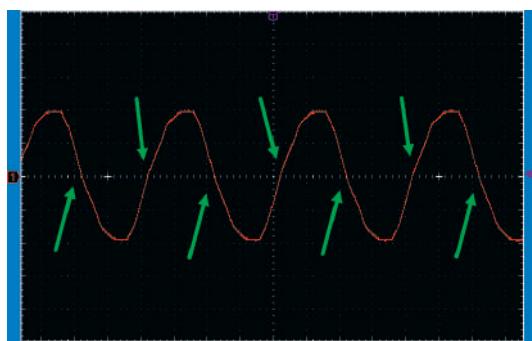


Figure 1. Onde sinusoïdale du secteur. Les flèches vertes indiquent les points de passage à zéro.

Avertissement : Ce projet utilise la tension du secteur. Il peut être mortel ! Faites attention, car ce circuit ne peut pas être utilisé pour contrôler les appareils qui utilisent des alimentations à découpage CA-CC (SMPS) comme unité d'alimentation, notamment les lampes LED modernes.

La **figure 1** montre l'onde sinusoïdale du secteur avec une fréquence de 50 Hz (dans certains cas 60 Hz). Pour construire un gradateur, les points de passage à zéro (les points où l'onde change de polarité) sont importants. Pour identifier ces points, nous devons utiliser un détecteur de passage à zéro.

La **figure 2** représente le schéma du circuit. R1, R2, IC1, D1 et C3 constituent le détecteur de passage à zéro. Il est conçu pour réaliser une isolation correcte (optique) avec la tension du secteur. Nous pouvons donc nous attendre à obtenir un signal sans bruit qui peut être connecté en toute sécurité aux E/S de l'Arduino.

La **figure 3** montre le signal de sortie du détecteur de passage à zéro (broche 4 de IC1). Vous pouvez certainement utiliser d'autres références d'optocoupleurs similaires.

Nous allons utiliser l'impulsion de passage par zéro comme un déclencheur pour le circuit de contrôle principal. Pour simplifier, exactement après un déclenchement (un passage par zéro), nous devons décider de la quantité de puissance à délivrer. Vous comprendrez mieux cela en examinant plus loin le code Arduino et la forme d'onde de sortie.

Le circuit IC3 est un triac BT138. La charge est en série avec le triac

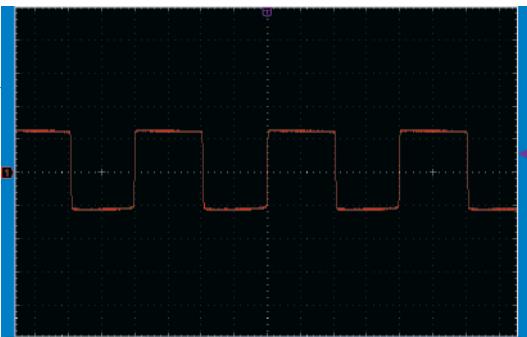


Figure 3. Le signal de sortie du circuit de détection du passage à zéro.

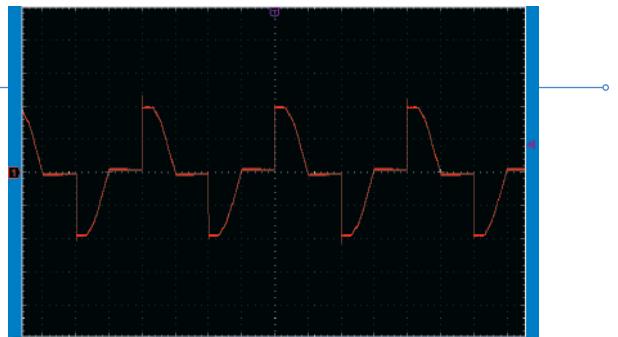


Figure 4. La forme d'onde de sortie à 50 % de la puissance (dim = 5000).

et la ligne CA, donc le triac détermine la quantité de puissance qui doit être fournie à la charge.

Attention : Le dissipateur du triac est connecté à la broche 2. Cela signifie que vous ne devez jamais toucher le dissipateur thermique ou le visser à un boîtier métallique sans isolateur !

R4, R5 et C2 forment un circuit amortisseur pour IC2. C1 et R7 constituent un amortisseur pour IC3. Ces composants permettent au dispositif d'être compatible avec une variété de charges, telles que les charges inductives.

IC2 est un opto-triac qui assure l'isolation galvanique entre la partie numérique et la ligne CA. La référence choisie est MOC3021. Vous pouvez également utiliser des références similaires ; cependant, veillez à ne pas utiliser des composants avec un détecteur de passage à zéro intégré. Ils ne sont utiles que pour commuter les charges CA (ON/OFF), mais pas pour la gradation.

Code Arduino

Il est maintenant temps d'examiner le code Arduino (**listage 1**) téléchargeable sur [1]. Il y a deux méthodes pour capter les impulsions du détecteur de passage à zéro : l'interrogation et l'interruption. Dans ma première tentative, j'ai utilisé une interruption, mais j'ai constaté un scintillement de la charge dans certaines situations. C'est un phénomène ennuyeux qui se produit avec certains gradateurs, et la raison en est une mauvaise temporisation. Comme je l'ai mentionné précédemment, les points de passage à zéro sont assez importants et tout décalage aléatoire du temps provoque une instabilité. L'utilisation d'une interruption introduisait une certaine fluctuation, qui provoquait un scintillement pour certaines valeurs `dim`. C'est pourquoi j'ai changé de méthode et opté pour l'interrogation (lignes 12 à 17).

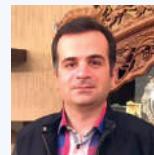
Tout ce que nous devons faire, c'est de changer le temps d'arrêt du triac dans les deux cycles (OFF-time), ainsi la variable `dim` définit la valeur de la puissance transférée.

La **figure 4** montre la forme d'onde de sortie (50 %). Vous pouvez développer le code et ajouter deux boutons pour augmenter et diminuer la puissance de sortie. ↵

190320-04

Des questions, des commentaires ?

Envoyez un courriel à l'auteur (hesam.moshiri@gmail.com) ou contactez Elektor (redaction@elektor.fr).



À propos de l'auteur

Hesam Moshiri est titulaire d'un master en conception de systèmes embarqués. Il a de l'expérience en électronique et en vision artificielle/apprentissage automatique. Il s'intéresse principalement à la rédaction et au marketing numérique dans les domaines précédemment mentionnés. Il a lancé sa chaîne YouTube « MyVanitar », où il publie des vidéos sur des projets électroniques et des conseils de conception et de mesure.



Listage 1. Code Arduino.

```
const byte ZCP = 2;
const unsigned int dim = 5000;

void setup() {
    pinMode(ZCP, INPUT);
    pinMode(10, OUTPUT);
    digitalWrite(10, LOW);
}

void loop() {
    if (digitalRead(ZCP) == HIGH)
        Zero_Cross();
}

void Zero_Cross() {
    digitalWrite(10, LOW);
    delayMicroseconds(dim);
    digitalWrite(10, HIGH);
}
```

LIENS

[1] Logiciel : <https://www.elektormagazine.com/summer-circuits-22>

↓ Télécharger le projet

