

Capteur de sons KY-037 et Arduino

“Paf” c’est allumé “Paf” c’est éteint.

Un petit gadget simple et ludique, à la portée de tous dans le monde des microcontrôleurs. Un claquement de main et la lumière fût ! Bien sûr vous trouverez dans le commerce ce genre d’appareil, donc rien de révolutionnaire, mais là, c’est du “fait maison”.

Quoi ! Un microcontrôleur pour allumer ou éteindre une lampe !

C’est un peu exagéré, c’est vrai, mais avec ce petit montage vous pouvez tester vos idées et vos essais sur les microcontrôleurs. Attention tout de fois, car c’est un montage avec le courant du secteur, donc potentiellement mortel ! Ne l’oubliez jamais, prenez toutes les précautions qui s’imposent.

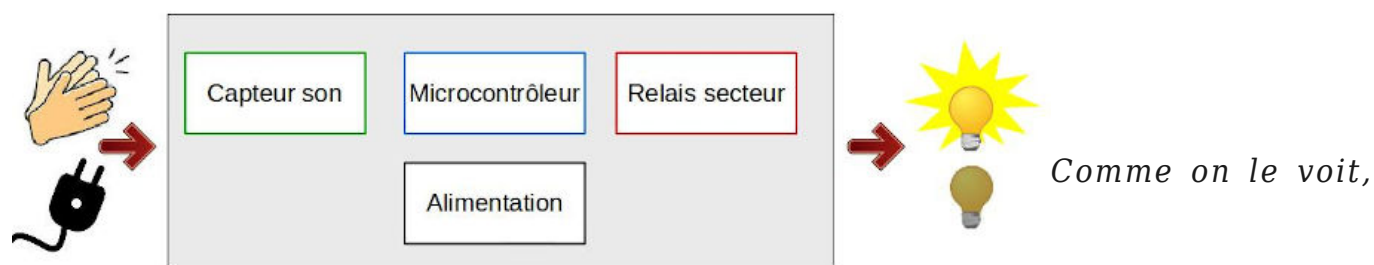
Présentation

En même temps que la présentation de ma réalisation (qui n’est de loin pas optimale), vous trouverez des pistes d’amélioration pour votre propre réalisation. Pour commencer voici la présentation finale du prototype du gadget, comme cela vous savez de quoi on parle :





Un capteur de sons KY-037 avec un Arduino Nano et un relais, voilà les éléments de base. La logique de fonctionnement est relativement simple et pour être le plus clair possible, voici le schéma de principe du système :



deux entrées : le son et l'énergie secteur et en sortie énergie ON/OFF.

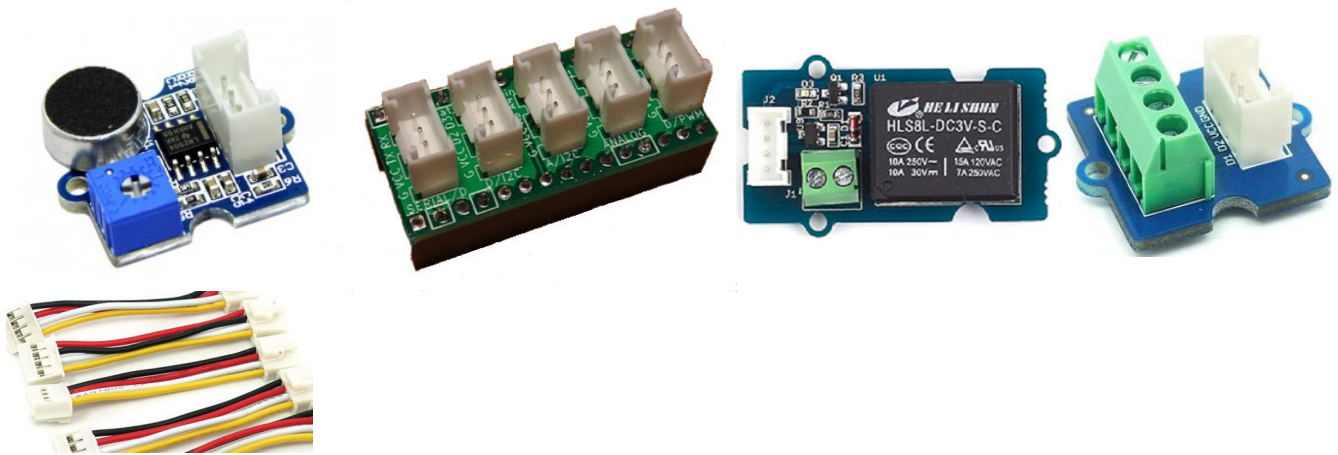
Remarque : Ne vous inquiétez pas pour la sécurité de l'installation électrique, ici c'est aux normes, mais je ne dis pas que c'est bien. ☐

Soudure ou pas soudure ?

Souvent lorsque l'on veut construire un petit système électronique, il est nécessaire de faire de la soudure pour relier les divers composants. Si vous ne voulez pas ou vous ne pouvez pas effectuer ce type de travail, il reste les solutions "sans soudure". Arbitrairement choisi, je vous en présente trois systèmes.

GROVE

Le système [Grove](#) a été mis au point par la société Chinoise Seeed, il y a énormément de modules disponibles à des prix très compétitifs, voici quelques modules que vous pourriez utiliser dans ce projet. Les connecteurs sont standardisés et les modules sont facilement fixables par petites vis. Exemples : capteur son, support Arduino Nano, relais secteur, connecteur à vis et câbles avec connecteurs.



TINKERKIT

Les modules TinkerKit (qui fait partie du monde Arduino, tout au moins à la société Arduino) sont aussi très répandus et pratique d'utilisation, exemple [ici](#), [ici](#), ou encore [ici](#)

DUPONT

C'est le connecteur que vous trouvez pratiquement partout pour les petits modules "électronique Arduino". Vous trouverez partout des fils tout fait avec ce type de connecteur, mais généralement ces fils (surtout s'ils sont bon marché) sont en fait de piètre qualité. Vous pouvez assez facilement acheter le matériel nécessaire pour réaliser vous-même vos fils avec connecteurs. L'inconvénient majeur, à mon sens, c'est la longueur du connecteur environ >15 mm. Mais c'est pratique car pas définitif et donc, en contrepartie, les contacts ne sont pas 100%.

Il vous suffit d'avoir une pince (c'est le plus gros investissement, environ 15€, des connecteurs "dupont" mâle s et femelles, éventuellement les boîtiers qui peuvent grouper les connecteurs et les câbles.

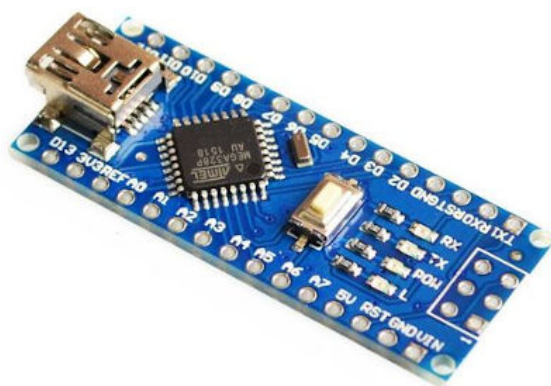


Liste du matériel

Maintenant la liste du matériel utilisé dans le prototype avec le coût approximatif des éléments, en prix chinois □ , mais c'est à vous de choisir votre fournisseur ! Je vous mets cette liste à titre indicatif, il est naturellement possible d'utiliser d'autres modules similaires. Le montant total, sans compter les fils et autres "petits trucs" : de 8 € à 12 €

ARDUINO NANO

C'est notre microcontrôleur, j'ai choisi celui-ci pour les raisons suivantes : matériel répandu et il y a plein internet de tutoriels et de présentations d'Arduino que ce soit sur YouTube ou le web. Prix relativement faible (2 €) et encombrement pas trop important.



Reste que le pour programmer (c'est un microcontrôleur), il est nécessaire d'avoir un ordinateur avec le logiciel adéquat. Pour la famille des Arduino, vous trouverez ce [logiciel](#)

[ici](#). Un PC sous Windows, Mac ou Linux sont utilisables. Il faut aussi un câble USB (USB type A -Mini USB Type B) pour relier le Arduino Nano à l'ordinateur.

Si vous avez des problèmes pour l'installation: site [Arduino France](#).

Attention, dans la gamme Arduino Nano, il y a le nouveau Arduino Nano 33 qui fonctionne en 3,3V. Ne le prenez pas pour ce gadget, car le module son KY-037 et l'Arduino Nano "standard" sont en 5V. Ou alors, changez aussi de module son et de relais, l'alimentation sera aussi à adapter.

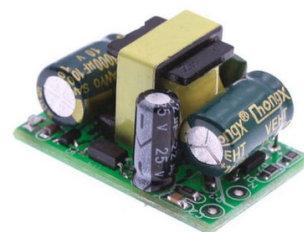
ALIMENTATION

Comme pour tout système électronique pour que les éléments fonctionnent, une puissance électrique est nécessaire. Comme l'ensemble doit piloter l'enclenchement/déclenchement d'une lumière ou un autre appareil en 230V (tension secteur domestique), il m'a semblé logique d'utiliser cette puissance puisqu'elle sera forcément disponible dans le boîtier. Mais comme l'Arduino, le capteur de son ainsi que le relais de commande fonctionne en 5 V, il est nécessaire d'avoir un module transformateur de 230V en 5V, je vous propose d'utiliser ce

[type de module](#), coût environ 2,2€



Le prototype présenté fonctionne avec une autre module, mais l'alimentation ci-dessus me semble plus adaptée pour un prix équivalent. Premier avantage ; tout est englobé dans une matière synthétique donc sécurité augmentée. Deuxième avantage ; pas de soudure pour le branchement au niveau du module, c'est des fils à connecter uniquement. Ci-joint la photo du module que j'ai utilisé dans mon prototype, ça fonctionne malgré tout.

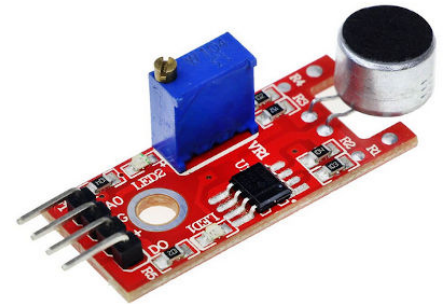


Sécurité

Attention, ce montage est réalisé avec le courant du secteur, c'est dangereux. Ne laissez pas les enfants tenter ce type de réalisation, et de préférence éloignez les pendant vos tests. Soyez méticuleux et soignez particulièrement avec la partie travaillant avec la tension secteur.

MODULE MICROPHONE

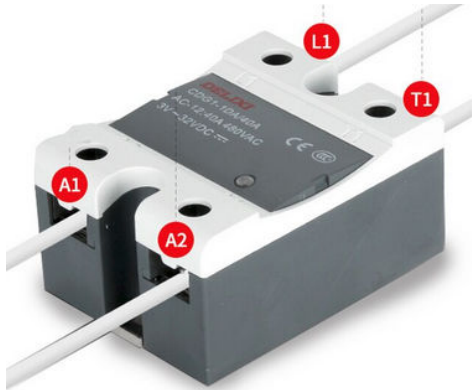
Circuit capteur son, module microphone [KY-037](#), coût inférieur à 1€. Ce genre de capteur capte une intensité sonore, ce qui veut dire que vous pourrez pas en faire un instrument de mesure de décibel. De plus son spectre sonore est très réduit. Il y a deux possibilités d'utilisation :

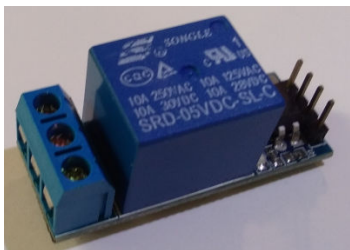


- soit on utilise la sortie analogique (la tension varie de 0 à 5V suivant l'intensité du volume sonore).*
- soit on utilise la sortie digitale (0V ou 5V) avec un seuil de niveau sonore de basculement. C'est cette solution qui est appliquée dans le système présenté. Le niveau du basculement peut-être réglé par le trimmer (le bloc bleu avec une vis).*

Ce KY-037 et Arduino (Uno, Nano ou autre) sont souvent vendus dans les kits pour débuter avec l'Arduino, c'est aussi une des raisons du choix de ce module.

RELAIS DE COMMANDE

Relais 5V, je vous conseille d'utiliser le [module suivant](#) à environ 2,5€ , il est vrai que *je ne l'ai pas testé directement sur une sortie Arduino* (selon les spécifications pour le 10DA-SSR cela doit passer). Dans mon prototype, mon stock a choisi pour moi  , voir la photo du module relais que j'ai utilisé ! Encore une fois la sécurité vous recommande un relais certes un peu plus onéreux et volumineux, mais beaucoup plus tranquilisant et sans soudure.



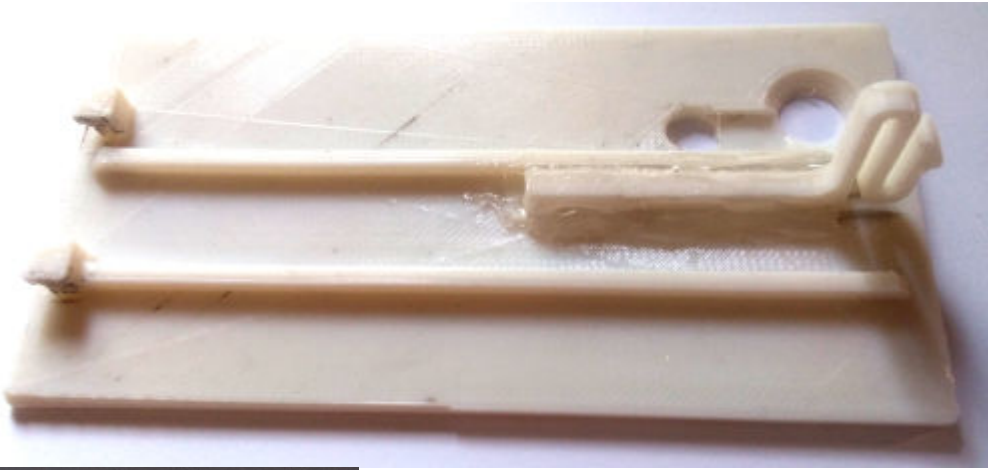
BOÎTIER

Pour le boîtier , il y a deux solutions soit vous avez accès à une imprimante 3D et dans ce cas vous pouvez fabriquer votre boîtier avec le positionnement des modules. Soit vous utilisez un boîtier plastique standard du commerce et vous collerez les divers modules dedans, une colle chaude fera l'affaire. Exemple de boîtier en ABS 1,30€



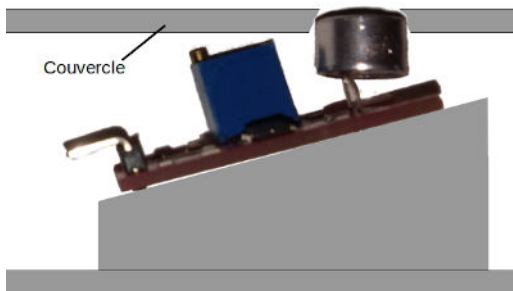
[Téléchargez](#) les dessins du boîtier de mon prototype, cela vous inspirera peut-être, en trois formats stl, step et fcstd.

Détail du couvercle et du crochet. J'avais fait un autre crochet, mais il n'a tenu que quelques utilisations. Je vous propose (mis à jour sur les dessins) un autre crochet. Sur le prototype c'est ajouté à la colle chaude !



Quelques conseils pour le choix du boîtier :

- 1. Il n'y a jamais assez de place !*
- 2. Placez un joint sur l'entrée des câbles secteurs (deux câbles)*
- 3. Inclinez le capteur de sons KY-037 afin de n'avoir qu'un trou dans le couvercle :*



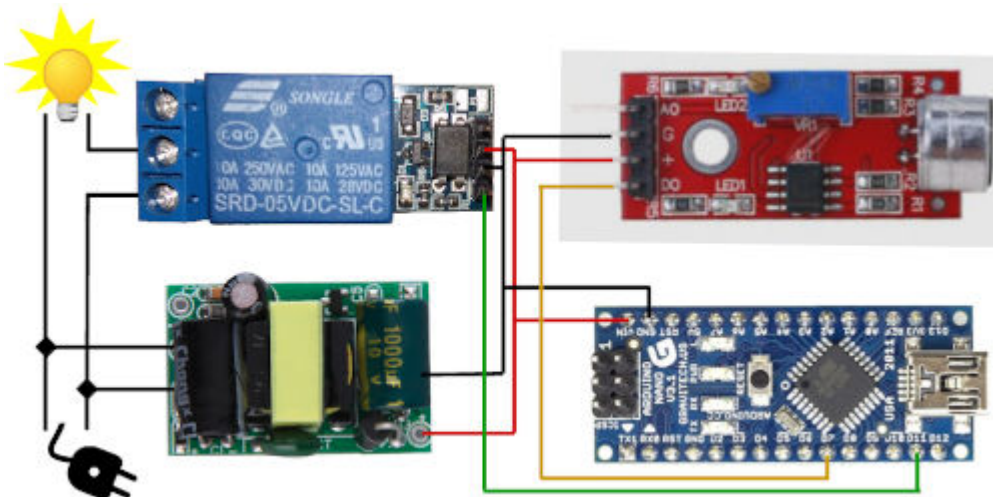
CONNECTEUR



Cela prend de la place, mais c'est simple et sûr. Ce sont des connecteurs secteurs utilisés dans le bâtiment, il vous faut 2 blocs de 3 places, prix environ inférieur à 1€/pièce.

Le projet et sa réalisation

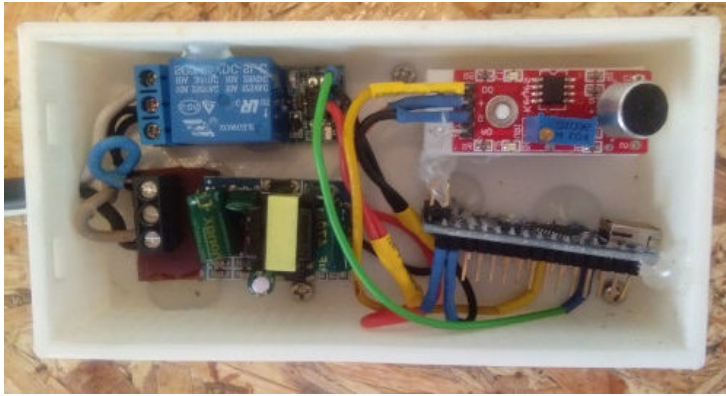
LE SCHÉMA DE PRINCIPE ET DE CÂBLAGE



Comme vous le voyez cela est extrêmement simple, le gros du travail étant d'amener l'alimentation électrique à chaque module : le KY-037, l'Arduino et au relais. En fait il n'y a que deux fils qui effectuent les transferts d'informations.

- 1. le fil (jaune), qui transmet l'état sonore vu par le module KY-037 à l'Arduino.*
- 2. le fil (vert) qui transmet l'ordre d'action au relais.*

Vue réelle du prototype :



LE PROGRAMME

Le principe de fonctionnement a été choisi qu'à l'alimentation du prototype le relais n'est pas activé et que dans ce cas le courant (pour moi la lumière) est ON, le relais est branché en "normalement fermé". Ce qui signifie pas de tension sur le relais, le courant passe. Pour mon installation cela m'a semblé le plus logique, et ma femme était d'accord avec ce fonctionnement, ALORS

Le choix des numéros d'entrée/sortie sur l'Arduino est complètement arbitraire. Vous trouverez donc ce principe appliqué dans le programme, qui est très simple et qui est suffisant annoté pour que chacun puisse retrouver ses petits ☐

```
//*****  
// Pilotage d'un relais par le son  
// MC juin 2019  
// libre de droits  
//*****  
  
//*****  
// branchement Arduino NANO  
int P_relais = 11 ;  
int P_son = 7;  
  
// variables internes  
int val=0; // valeur du son HIG ou LOW  
boolean valRelais=false; // memorisation de l'état du relais  
  
// Pour le délai entre deux prises en compte de commande, j'ai  
préfééré
```

```

// un délai sous la forme d'un compteur, sans bloquer le
microcontrôleur
// dans un délai d'attente avec l'instruction Delay(1000)
long attente=1000; //temps attente entre deux prises en compte
du son
long compteur; //compteur de temps

//*****
void setup()
{
  // initialisation des entrées/sorties
  pinMode (P_relais,OUTPUT);
  pinMode (P_son, INPUT);
  digitalWrite (P_relais,LOW); //initialise l'état du relais
  compteur=millis(); //initialisation du compteur
}

//*****
void loop ()
{
  val = digitalRead(P_son); //lecture du capteur son
  if (val== HIGH)
  {
    // si le son est grand, intense
    if(millis()-compteur > attente){
      //si le temps d'attente est écoulé
      if (valRelais == true)
      {
        // si le relais est actif
        digitalWrite (P_relais,LOW);
        valRelais=false;
      }else{
        //si le relais est inactif
        digitalWrite (P_relais,HIGH);
        valRelais=true;
      }
      compteur=millis(); //re-initialisation du compteur
    }
  }
}
}

```

Le téléchargement du programme : [c'est ici](#).

Conclusion

On n'a bien joué avec ce montage, à force, on avait mal aux mains ☹

Maintenant il fait le travail sans trop de soucis, juste le niveau sonore de basculement est assez fin à trouver, trop haut : on se fait mal aux mains, trop bas : des sons intempestifs font réagir le prototype.

Suggestion

Avec le KY-037 et l'Arduino, la détection analogique (pin A0 du KY037) et analyse par programme serait en fin de compte plus souple et plus précise ?