

PIRATEZ LES BOUTONS

Prenez le contrôle d'autres systèmes autour de vous en utilisant quelques composants supplémentaires, vous pourrez "presser" les boutons avec votre Arduino

Découverte : Optocoupleur, connexion avec d'autres composants

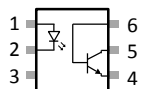
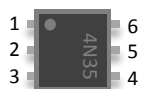
Durée : 45 minutes

Difficulté: ■■■■■

Basé sur les projets : 1,2,9

Avertissement : *Vous ne devez plus être un débutant si vous décidez de réaliser ce projet. Vous allez ouvrir un appareil électronique et le modifier. Vous allez annuler la garantie de cet appareil, et si vous n'êtes pas soigneux, vous pourriez endommager ce dispositif. Assurez-vous d'être familiarisé avec tous les concepts électroniques des projets précédents avant de tenter celui-ci. Pour vos premières tentatives, nous vous recommandons d'utiliser des éléments peu onéreux que vous ne craignez pas d'endommager, jusqu'à ce que vous ayez acquis suffisamment d'expérience et de confiance en cette technique.*


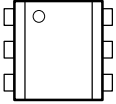
Même si Arduino permet de contrôler beaucoup de choses, il est quelquefois plus facile d'utiliser des outils qui ont été créés pour répondre à des besoins spécifiques. Peut-être voulez-vous contrôler une télévision ou un lecteur de musique, ou même piloter une voiture radio commandée. La plupart des appareils électroniques ont un panneau de contrôle avec des boutons, et beaucoup de ces boutons peuvent être modifiés pour que vous puissiez les « presser » avec un Arduino. Contrôler des sons enregistrés est un bon exemple. Si vous voulez enregistrer et rejouer des sons, il vous faudra énormément de temps et d'efforts pour permettre à un Arduino de réaliser ces tâches. Il est beaucoup plus simple de récupérer un petit appareil qui peut effectuer un enregistrement et relire les sons (lecteur/enregistreur MP3, dictaphone, ...), puis de modifier ses boutons pour en prendre le contrôle via les sorties pilotées par votre Arduino.



Les **Optocoupleurs** sont des circuits intégrés qui vous permettent de contrôler un circuit depuis un autre sans aucune connexion électrique entre les deux. A l'intérieur d'un optocoupleur il y a une LED et un détecteur de lumière. Lorsque la LED dans l'optocoupleur est allumée par votre Arduino, le détecteur de lumière ferme un interrupteur interne. L'interrupteur est connecté à 2 broches de sortie (4 et 5) de l'optocoupleur. Lorsque l'interrupteur interne est fermé, les broches de sortie sont connectées ensemble. Lorsque l'interrupteur est ouvert, elles ne sont plus connectées. De cette manière, il est possible d'activer des interrupteurs sur différents appareils sans avoir besoin de les connecter directement à l'Arduino.

Dans cet exemple, les schémas et le programme permettent de contrôler un module d'enregistrement numérique capable d'enregistrer et de rejouer 20 secondes de sons, mais les notions exposés sont transposables à tout appareil possédant des interrupteurs qui vous sont accessibles. Bien qu'il soit possible de mettre en œuvre cet exemple sans souder aucun fil, quelques soudures vous permettront certainement de simplifier sa réalisation. Pour plus d'information à propos des soudures, reportez-vous au projet 'Serrure sonore' (#12).

INGREDIENTS

| | |
|---|---|
|  |  |
| RESISTANCE 220 Ω | OPTOCOUPLEUR |
| x1 | x1 |

CONSTRUIRE LE CIRCUIT

Fig.1 - [Circuit]

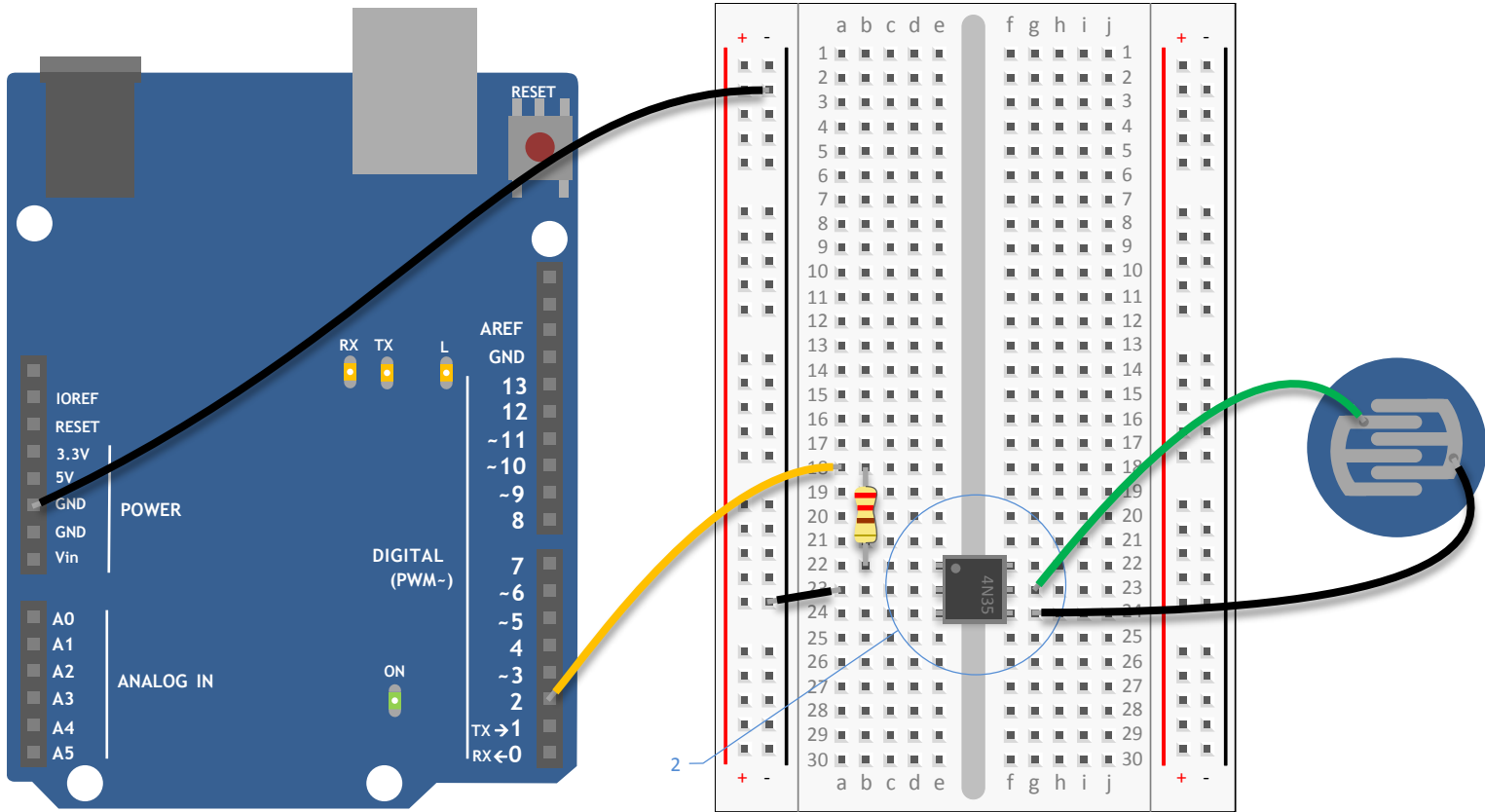
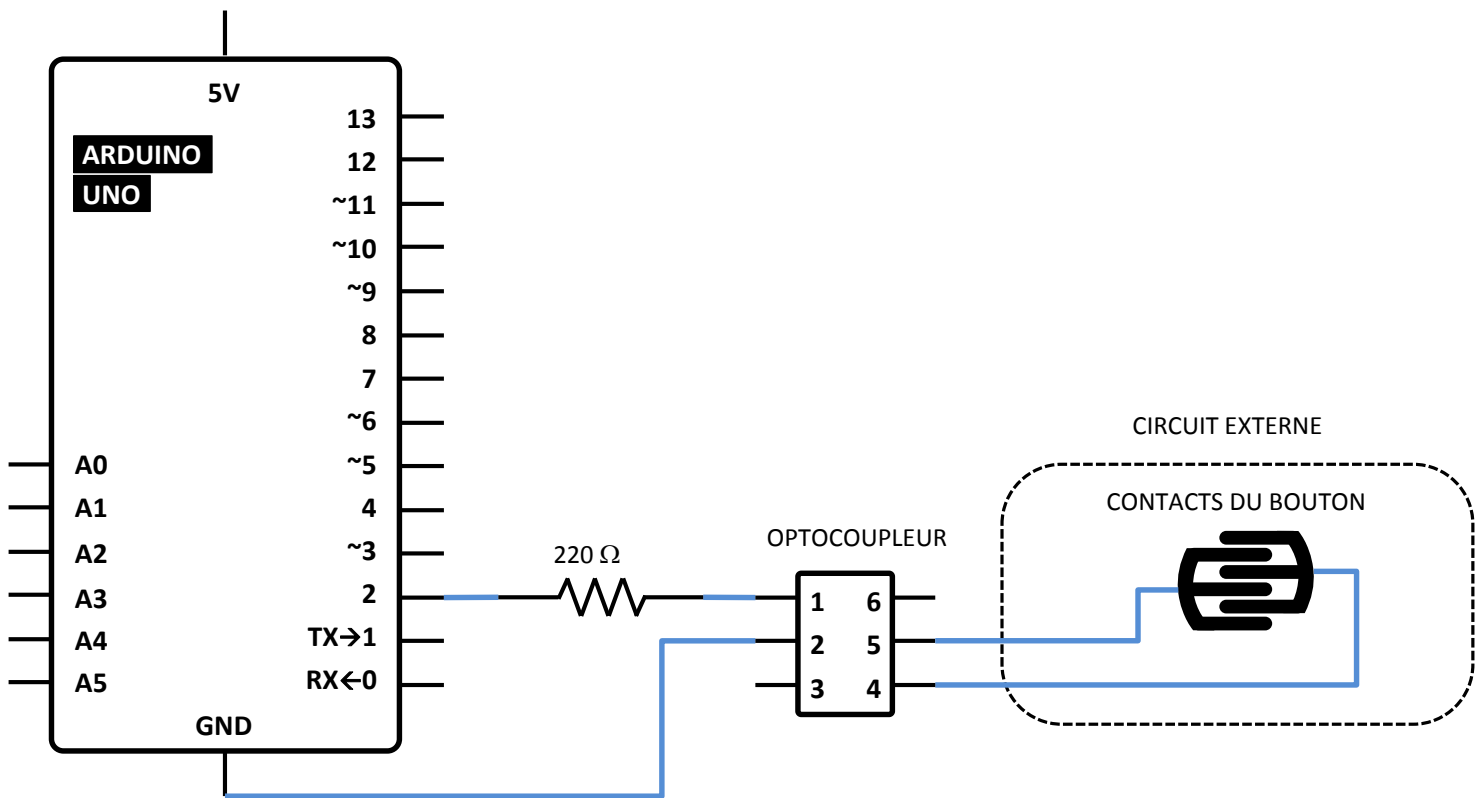


Fig.2 - [Schéma]



- 1) Connecter la masse à votre platine de test depuis votre Arduino.
- 2) Placer l'optocoupleur sur la platine de test de telle manière qu'il chevauche le milieu de la plaque (voir le diagramme du circuit sur la *Fig. 1*).
- 3) Connecter la broche 1 de l'optocoupleur à la borne 2 de l'Arduino, en série avec une résistance de $220\ \Omega$ (rappelez-vous, vous alimenter une LED à l'intérieur de l'optocoupleur, vous ne voulez pas la griller). Connecter la broche 2 de l'optocoupleur à la masse.
- 4) Sur la carte principale du module d'enregistrement numérique, il y a un grand nombre de composants électroniques, dont un bouton de 'relecture'.
Pour contrôler cet interrupteur, vous allez avoir besoin de retirer ce bouton. Retourner la carte électronique et trouvez les attaches qui tiennent le bouton en place. Pliez doucement ces attaches et retirez le bouton de la carte.
- 5) Sous ce bouton se trouvent 2 petites plaques en métal. Cette disposition est typique de nombreux appareils électroniques avec des boutons poussoirs. Les deux "fourchettes" visibles sont les deux côtés de l'interrupteur. Un petit disque métallique à l'intérieur du bouton poussoir connecte ces deux fourchettes lorsque vous appuyez sur ce bouton.
- 6) Lorsque les fourchettes sont connectées, l'interrupteur est fermé sur la carte électronique. Vous allez fermer cet interrupteur grâce à l'optocoupleur.
Cette méthode, la fermeture d'un interrupteur avec un optocoupleur, fonctionne seulement si un des deux côtés de l'interrupteur du bouton poussoir est connecté à la masse sur l'appareil. Si vous n'en êtes pas sûr, prenez un multimètre et mesurez la tension entre une des fourchettes et la masse sur l'appareil. Vous devez faire cette mesure lorsque l'appareil est allumé, il faut donc être vigilant afin de ne rien toucher d'autre sur la carte électronique. Dès que vous avez détecté quelle fourchette est connectée à la masse, déconnectez l'alimentation de l'appareil.
- 7) Ensuite, connecter un fil à chacune des petites plaques en métal. Si vous soudez ces fils, assurez-vous de ne pas relier les deux côtés de l'interrupteur ensemble. Si vous préférez ne pas souder, et utiliser du ruban adhésif, assurez-vous que votre connexion soit franche et solide, sinon l'interrupteur ne pourra pas se fermer. Enfin, assurez-vous qu'aucun fil connecté à l'une des fourchettes ne soit connecté aussi à l'autre fourchette, sinon votre interrupteur sera perpétuellement en position 'fermé'.
- 8) Connecter les deux fils aux broches 4 et 5 de l'optocoupleur : Relier le coté de l'interrupteur connecté à la masse dans l'appareil à la broche 4 de l'optocoupleur et reliez l'autre fourchette à la broche 5 de l'optocoupleur.

LE PROGRAMME

Déclarer une constante

La partie la plus amusante de ce projet réside dans le montage du circuit et de l'optocoupleur. Le code est similaire à celui du 1^{er} projet que vous avez créé avec votre Arduino. Vous allez jouer le son toutes les 20 secondes en positionnant la broche 2 à l'état HAUT (**HIGH**).

Créez une constante pour indiquer le n° de la broche de contrôle de l'optocoupleur.

```
const int optoBroche = 2;
```

Configurer le mode de la broche

Dans la fonction d'initialisation **setup()**, définissez la broche de l'optocoupleur comme étant une sortie.

```
void setup() {  
    pinMode(optoBroche, OUTPUT);  
}
```

Positionner la broche à l'état haut et bas

La boucle **loop()** met `optoBroche` à l'état haut (**HIGH**) pendant quelques millisecondes, assez longtemps pour que l'optocoupleur ferme l'interrupteur sur l'appareil. Puis, la broche `optoBroche` est mise à l'état bas (**LOW**).

```
void loop() {  
    digitalWrite(optoBroche, HIGH);  
    delay(15);  
    digitalWrite(optoBroche, LOW);  
}
```

Attendre quelques instants

Attendre pendant 21 secondes afin de laisser passer suffisamment de temps pour que le message soit lu entièrement avant de démarrer une nouvelle boucle **loop()**.

```
    delay(21000);  
}
```

UTILISEZ LE MONTAGE

Mettez une pile dans votre enregistreur de sons. Pressez et tenez appuyé le bouton d'enregistrement de votre appareil. Pendant que vous maintenez ce bouton, vous pouvez enregistrer des sons au travers du microphone. Essayer d'enregistrer un peu moins de 20 secondes, votre voix, le chat, ou jouez des percussions avec les pots et les cuillères dans la cuisine pour faire un peu de bruit (mais faites quand même attention avec le chat !).

Une fois que vous êtes satisfait de ce que vous avez réussi à enregistrer, alimentez votre Arduino avec le câble USB. Votre enregistrement devrait commencer à être joué. Si vous avez effectué un enregistrement d'une durée proche de 20 secondes, la lecture de votre enregistrement devrait recommencer très peu de temps après la fin.



Expérimentez différents sons, différentes longueurs d'enregistrement, et différentes durées de bascule de l'interrupteur avec la fonction **delay()** de votre programme.

Si vous basculez l'interrupteur pendant qu'un son est joué, la lecture sera stoppée. Comment pouvez-vous tirer avantage de ce comportement pour créer une séquence unique de sons?



Les circuits intégrés (CI = **IC** en anglais) sont virtuellement dans tous les appareils électroniques. La grosse puce de 28 broches de votre Arduino est un CI qui héberge les cerveaux de la carte. Il y a d'autres CIs sur la carte qui l'assistent pour la communication et l'alimentation. L'optocoupleur et la puce principale de l'Arduino sont des puces dont le format est DIL ou DIP (**Dual In-line Package** en anglais), qui peut se traduire par « boîtier à double rangée ». Ces puces DIP sont un type de puces que la plupart des amateurs d'électronique utilisent parce qu'elles s'enfichent aisément dans les platines de test et n'ont pas besoin d'être soudées de manière permanente pour être utilisées.



L'exemple dans ce projet rejoue simplement le son à intervalle régulier. Comment pourriez-vous intégrer les entrées des projets précédents pour déclencher la lecture de ces sons ? Quels sont les autres objets à pile ou à batterie, autour de vous, dans votre maison, qui auraient besoin d'être contrôlés par un Arduino ? Cette technique de contrôle d'un appareil électronique grâce à un optocoupleur, en connectant les deux extrémités d'un interrupteur, peut être utilisée dans bien d'autres appareils. Quels autres appareils souhaitez-vous contrôler ?

Les optocoupleurs peuvent contrôler des appareils qui sont sur un circuit différent. Les deux circuits sont électriquement séparés l'un de l'autre à l'intérieur du composant.

EXTRAIT DU LIVRE DE PROJETS ARDUINO

EDITEURS

Projets et texte par Scott Fitzgerald et Michael Shiloh
Revue de texte complémentaire par Tom Igoe

DESIGN ET DIRECTION ARTISTIQUE

TODO

Giorgio Olivero, Mario Ciardulli, Vanessa Poli, Michelle Nebiolo
todo.to.it

EDITION NUMERIQUE ET GESTION DE PROJET

Officine Arduino Torino
Katia De Coi, Enrico Bassi

CONSEILLERS ET SUPPORTERS

Massimo Banzi, Gianluca Martino, Smart Projects

TESTEURS DES PROJETS ET RELECTEURS

Michael Shiloh, Michelle Nebiolo, Katia De Coi, Alessandro Buat, Frederico Vanzati, David Mellis

REMERCIEMENTS

Un grand merci à toute la communauté des utilisateurs Arduino pour leurs contributions continues, leur soutien, et leurs retours.

Nous remercions particulièrement l'équipe Fritzing: Quelques-unes des illustrations de composants électroniques utilisés dans le livre sont issues ou inspirées par le projet open-source de Fritzing (www.fritzing.org).

Un grand merci à Paul Badger pour la bibliothèque *CapacitiveSensor* utilisée dans le projet 13.

Le texte du Livre de Projets Arduino est distribué sous licence Creative Commons Attribution-NonCommercial-ShareAlike 3.0 License 2012 par Arduino LLC. Cela signifie que vous pouvez copier, réutiliser, adapter et vous appuyer sur le texte de ce livre en citant l'œuvre originale (mais pas d'une manière qui suggérerait que nous souscrivons à vous ou votre utilisation de l'œuvre) et seulement si le résultat de votre travail est transmis sous la même licence Creative Commons.

Les termes complets de la licence sont disponibles à : creativecommons.org/licenses/by-nc-sa/3.0/

© 2012 Arduino LLC. Le nom *Arduino* et le logo sont des marques de Arduino, déposées aux États-Unis et dans le reste du monde. Les autres noms de produits et de sociétés mentionnés dans ce document sont des marques commerciales de leurs sociétés respectives.

Les informations contenues dans ce livre sont distribuées «telles quelles» sans aucune garantie supplémentaire. Bien que toutes les précautions aient été prises dans la conception de ce livre, ni les auteurs ni Arduino LLC ne pourraient endosser une quelconque responsabilité envers toute personne ou entité à l'égard de toutes pertes ou dommages causés ou déclarés causés directement ou indirectement par les instructions contenues dans ce livre ou par le logiciel et le matériel qu'il décrit.

Ce livre ne peut être vendu séparément du 'Kit de Démarrage Arduino'.

Conçu, imprimé et relié à Turin, Italie
Septembre 2012

Première réimpression, Décembre 2012

Traduit de l'anglais par Nicolas PONCET