

Bibliographie

FLIPPER



SOMMAIRE

- 1) Principe d'un flipper
- 2) Etude générale de flippers
 - a) Modèle 1
 - b) Modèle 2
 - c) Modèle 3
- 3) Affichage des points et son
 - a) Ecran à cristaux liquides
 - b) Ecran à matrice de points
 - c) Son
- 4) Détail de conception des éléments d'un flipper
 - a) Le lance bille
 - b) Capteurs
 - c) Les flippers
 - d) Pop bumpers
 - e) Cibles
 - f) Slingshots
 - g) Les composants
- 5) Liaison avec Arduino-plateau de jeu
 - a) MOSFET
 - b) Diviseur de tension
 - c) Capteurs photorésistance
- 6) Conclusion



1) Principe d'un flipper

Un flipper est un jeu électromécanique dont le but est de marquer des points en dirigeant une ou plusieurs bille(s) métallique(s) sur un plateau de jeu. Il faut jouer le plus longtemps possible en évitant qu'elle tombe entre les deux poignets qui nous permettent de la relancer.

Voici la liste des éléments principaux d'un flipper :

- lance bille
- flipper
- bumpers
- cibles
- slingshot
- couloir éjecteur
- trou et couloir de sortie
- lumière



2) Etude générale de flippers

a) Modèle 1

Le flipper

Le flipper est fait de bois avec des dimensions : 58 cm de large et 120 cm de long. Le couvercle est en plexiglass. Il a une pente de 7°.

Il utilise une carte arduino 256 qui permet de déterminer quand les interrupteurs sont activés, d'allumer ou éteindre les lumières en conséquence, de suivre le score et de l'afficher (sur un petit écran LCD).

Lorsque le code détecte que l'entrée est passée de HIGH à LOW, il sait que la cible a été frappée et allume alors les LED appropriées.

Pour le son il a ajouté à la fin une carte Wav Trigger, il peut grâce à elle avoir une musique de fond et par-dessus ajouter les bruits en fonction du jeu qui a lieu sur le flipper.

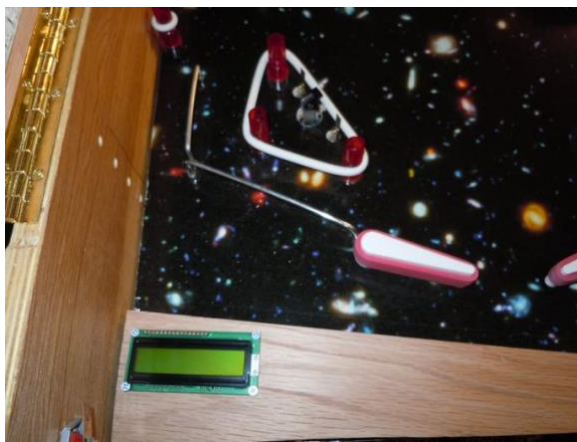
Un petit capteur de force est utilisé pour savoir quand une balle est perdue. Cela permet de suivre le nombre de balles jouées et de déterminer quand le jeu est terminé. Lorsque la partie est terminée, toutes les lumières clignotent en séquence.

Les éléments du flipper

Les composants du flipper ont besoin d'une alimentation de 25V. En particulier les pop bumpers. Étant donné l'écart de tension avec la carte Arduino il a ajouté un diviseur de tension.

Pour faciliter les branchements, il a utilisé des barres omnibus collées sous le jeu (plutôt que de tout brancher sur le secteur). De plus, il a utilisé du fil de calibre 16 pour tous les composants reliés à l'alimentation électrique de 24 volts et du fil de calibre 22 pour les connexions à la carte Arduino.

Pour envoyer la balle, il utilise un tireur de balle ou un poussoir.



Pour les slingshots, il a acheté tous les éléments des frondes sur internet déjà assemblés. Des rails métalliques fabriqués avec du fil d'acier chromé sont utilisés pour guider la balle à côté des batteurs du flipper.

Les pop bumpers ont été achetés sur internet.

b) Modèle 2

Le flipper

Pour ce modèle, le flipper n'a pas été construit : il a acheté un vieux flipper qu'il a rénové. Le flipper est fait de bois avec un couvercle en plexiglass et un angle de 7°.

Il utilise une carte Arduino 256.

Pour les lumières il a utilisé des LED (mais conseille des ampoules). Elles seront les principales sources de sorties de l'Arduino.

Pour le son, il a utilisé une carte de protection MP3 basée sur la VS1053B (qui peut jouer des fichiers audio). Cependant, le temps de traitement est élevé pour certaines actions de flipper notamment pour les pop bumpers qui sont très rapides : le son doit donc être très court. Il est limité aux actions "lentes" comme les cibles et la musique générale.

Enfin, concernant l'alimentation de la machine, 3 voltages différents sont utilisés. Une alimentation de 24 volts pour les bobines, une alimentation de 12V pour l'éclairage ambiant et une alimentation de 5V pour l'Arduino.

Pour le décompte des points, il a mis en place des billes bonus et des points supplémentaires si certaines cibles étaient touchées. De plus, lorsque personne ne joue, un simple "spectacle" lumineux avec les LED contrôlées par l'Arduino a lieu.

Le score est affiché et calculé pendant le jeu, il commence à 0. À la fin de chaque partie, le score actuel est comparé au meilleur score. S'il est supérieur, le score le plus élevé est remplacé. Grâce à la mémoire EEPROM de l'Arduino, on peut conserver le score élevé, même lorsque l'Arduino est éteint.

Les éléments du flipper

Les flippers, pop bumpers et slingshots font partie des éléments principaux de son flipper. Les flippers fonctionnent automatiquement tandis que les autres composants sont contrôlés par l'Arduino. On a deux possibilités : les acheter ou bien les construire. Pour les construire, on utilise un solénoïde comme cœur. C'est une bobine de fil métallique enroulée, serrée qui, une fois alimentée, produit un champ magnétique (*explications partie 3*).

Pour les flippers, il a utilisé des bobines à 3 contacts fonctionnant avec une tension de 24V. Comme ils ne sont pas liés à l'Arduino, ils sont enclenchés mécaniquement par des interrupteurs et un bouton poussoir.

Les slingshots sont généralement constitués de 2 interrupteurs à lames (EOS) et d'une bobine alimentant un "kicker" qui propulse la balle au loin. Sur les anciennes machines, ces bobines étaient directement reliées à l'alimentation électrique, ce qui les rendait très réactives et rapides. Les interrupteurs n'envoient donc pas de signal mais court-circuitent simplement la bobine. Le déclenchement de la bobine active mécaniquement un interrupteur à lames monté en dessous, c'est notre signal pour l'Arduino.

Les pop bumpers ont le même mécanisme que dans la méthode 1. Lorsque la balle rentre dans le cercle, l'arbre descend et l'interrupteur envoie un signal à l'Arduino qui alimente alors la bobine pendant le temps nécessaire. Il y a toujours une tension trop grande pour l'Arduino donc il a utilisé des transistors à effet de champ métal-oxyde-semiconducteur (MOSFET). Pour le code de cette partie il a utilisé la fonction `millis()` pour chronométrer le temps d'activation du pop bumper idéal, pour lui : 40ms.

Détails des éléments du flipper fonctionnant avec solénoïdes

Un solénoïde est une bobine de fil qui est magnétisée lorsqu'un courant est appliqué. L'Arduino ne peut pas interagir directement avec eux car ils ont besoin de beaucoup de courant. Dans un jeu de flipper, un solénoïde est utilisé pour transformer l'électricité en mouvement mécanique. Pour ce faire, on utilise l'effet magnétique pour que la bobine attire une tige métallique (plongeur) qui est reliée à une pièce que l'on veut faire bouger.

Pour protéger l'Arduino et l'alimentation électrique, il a mis des diodes sur chaque solénoïde. Elles empêchent le "rebond" qui se produit lorsque la bobine est éteinte. Lorsque le champ magnétique s'effondre, la diode permet au courant de se dissiper à travers la bobine et non à travers l'Arduino ou l'alimentation électrique, empêchant ainsi leur destruction.

On peut séparer nos solénoïdes en deux groupes, ceux qui seront déclenchés directement et ceux que l'Arduino déclenche.

Si on déclenche directement un solénoïde, il suffit de fermer un interrupteur qui complète le circuit. Le courant circule et le solénoïde est activé. Cela permet une action rapide, l'Arduino n'en a pas conscience. Les points pour l'activation des slingshots sont attribués par un interrupteur qui est déclenché par le mouvement du slingshot.

Pour les déclencher par l'Arduino, il utilise des MOSFET.

L'Arduino ne doit jamais cesser de recevoir des signaux du terrain de jeu. Il utilise alors la fonction millis pour le chronométrage des solénoïdes. Il active la bobine à un moment donné dans notre jeu, déclenche le chronomètre. Lorsque le temps est écoulé, l'Arduino désactive la bobine. Il a réglé millis() à 50ms pour les pop bumpers et un peu plus longtemps pour les autres éléments du flipper.

De plus, il a également ajouté une fonction de secours au cas où une bobine resterait activée trop longtemps. Une autre fonction millis() chronomètre donc le temps où une bobine est allumée pour que si ce temps dépasse un seuil l'Arduino coupe l'alimentation des bobines et que le programme s'arrête.

c) Modèle 3

Pour ce projet il s'agit de la même personne que pour le modèle 2 mais il a modifié certaines choses par rapport à son montage précédent. Il est encore parti d'un flipper déjà existant qu'il a réparé. Ainsi, tous les principaux composants du flipper étaient déjà présents.

Pour ce qui est des lumières, il n'a pas utilisé de LED individuelles mais une chaîne de LED (LED WS2812B), comme ça une seule broche de l'Arduino est utilisée et chaque LED peut être contrôlée individuellement.

Pour le son il a utilisé la même carte que précédemment (VS1053B) mais pour pallier au manque de rapidité du traitement, il a ajouté une autre carte Arduino qui se contrôle depuis son Arduino principal (grâce à la bibliothèque WIRE).

Pour afficher le score il a utilisé deux écrans à matrice de points LED MAX7219, un pour afficher les scores, l'autre pour afficher les meilleurs scores. Comme avant le score est calculé pendant le jeu et commence à 0.

3) Affichage des points et son

La plupart du temps 2 scores sont affichés : le score maximum et le score de celui qui vient de jouer. Il faut donc stocker dans la mémoire de l'Arduino le score maximum pour que même après avoir éteint le flipper, il soit affiché.

a) Ecran à cristaux liquides

L'utilisation des écrans LCD est assez simple. En effet, un décodeur de caractères est déjà intégré à l'écran et permet ainsi d'afficher ce qui est envoyé par l'Arduino et de le garder en mémoire.

b) Ecran à matrice de points

Les écrans à matrice de points (par exemple LED MAX7219) sont contrôlés à l'aide de deux bibliothèques (MD_Parola et MD_MAX72XX). Ils ont un branchement plus simple que pour l'écran LCD, ils n'utilisent que trois broches de l'Arduino mais sont limités à 64 LED.

c) Son

Pour le son il faut un haut-parleur et une carte spéciale qui nous permet de stocker et lancer la musique.

En général, une musique de fond est diffusée en continu. Des bruitages sont émis lorsqu'un élément du flipper est touché. A la fin d'une partie une autre musique est jouée.

Solution 1 : carte de déclenchement WAV (modèle 1)

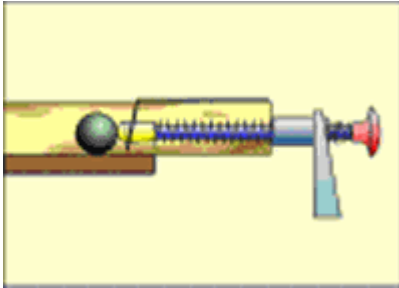
La carte fonctionne avec du 6V les fichiers sont stockés sur une carte mémoire SDHC. Il faut qu'ils soient au bon format c'est-à-dire des fichiers .wav et qui fonctionne à 44,1 kHz. Pour communiquer avec l'Arduino on a besoin de deux bibliothèques supplémentaires : "AltSoftSerial" qui gère la communication entre l'Arduino et la carte et "WAV-Trigger-Arduino-Serial-Library" où sont stockées les différentes commandes.

Solution 2 : carte de protection MP3 (modèle 2-3)

Seule, cette carte à un temps de traitement trop long par rapport aux actions du flipper, pour ne pas avoir de problème de son on peut ajouter une autre carte Arduino qui fait le lien entre la carte MP3 et l'Arduino qui contrôle le jeu. La carte de développement de module MP3 peut lire plusieurs types de fichiers audio (contrairement à la carte WAV), elle utilise la bibliothèque Adafruit_VS1053.

4) Détail de conception des éléments d'un flipper

a) Le lance bille



Fonctionnement

On tire sur le pommeau (rouge), cette action comprime le grand ressort. En lâchant la poignée le ressort se détend d'un coup et propulse la bille. Le petit ressort sert à amortir le choc sinon la poignée se disloquerait rapidement.

b) Capteurs

Pour savoir quand la partie commence et se termine (bille est perdue) deux capteurs sont utilisés. Un est positionné dans la rampe de lancement (quand la bille passe au niveau du capteur la partie commence). L'autre se trouve dans le rail qui récupère la bille après l'avoir perdu (quand la bille passe au niveau du capteur la partie est terminée). Nous avons étudié deux sortes de capteurs.

Capteur de force

Les capteurs de force permettent de détecter quand la bille passe dessus. Un capteur de force est un transmetteur qui transforme une force physique en un signal électrique.

Lorsque la bille passe sur ces capteurs, elle appuie dessus déformant la jauge de déformation à l'intérieur du capteur. Pour traduire cette contrainte sur le capteur, elle diminue la résistance totale du capteur de force. L'Arduino capte ainsi une tension plus élevée et sait lorsque la bille passe sur le capteur.

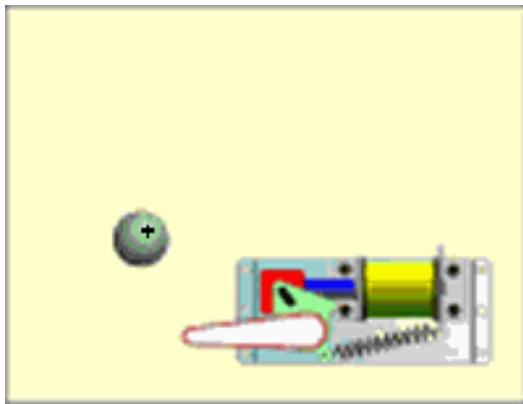
Capteur opto-couleurs

Les capteurs opto-couleurs permettent de détecter quand la bille passe devant.

D'un côté de ces capteurs se trouve une diode émettrice qui crée donc un rayon lumineux. De l'autre côté se trouve un récepteur. S'il est éclairé la tension à ses bornes est très faible. Lorsque la bille passe il n'est plus éclairé car elle coupe le rayon lumineux de l'émetteur, la tension à ses bornes va alors augmenter. L'Arduino sait ainsi quand la bille passe en fonction de la valeur de la tension.

c) Les flippers

Fonctionnement

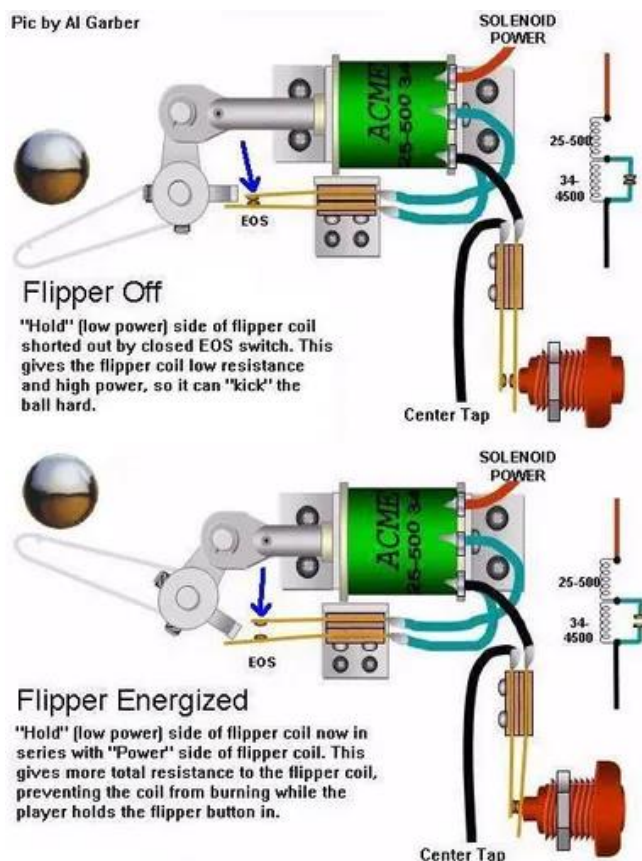


En appuyant sur le bouton du flipper un contact se fait et la bobine (en jaune) reçoit du courant. Elle le transforme en champ magnétique, ce qui attire alors le plongeur (en bleu) (cf explication plongeur). Celui-ci est attiré à l'intérieur de la bobine et entraîne avec lui une billette (en rouge) qui fait tourner la "pince" (en vert) qui soulève le batteur. Le ressort mis en place sert à ramener le batteur à son état initial après l'avoir levé.

Ainsi, les batteurs sont pilotés par une bobine dotée de 2 enroulements en série.

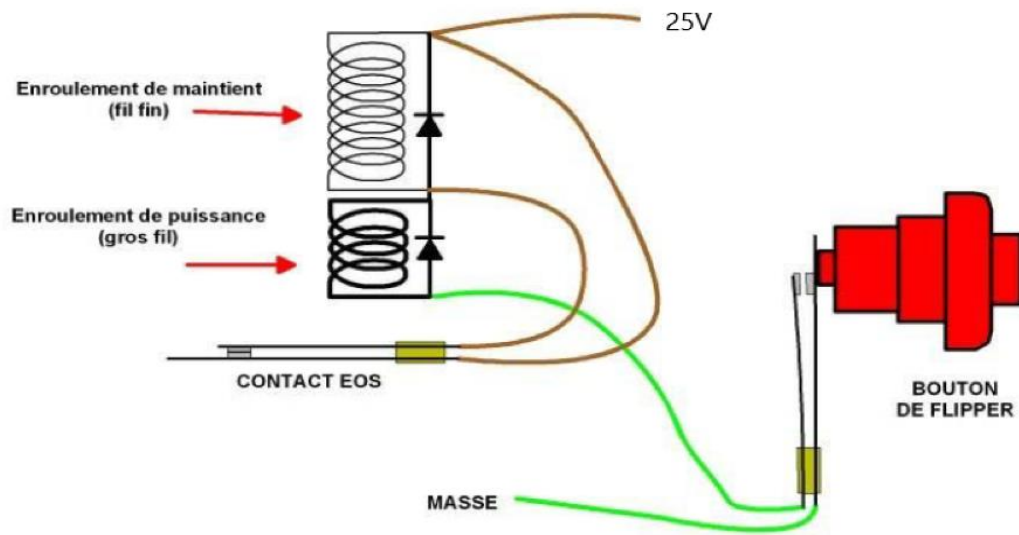
Un enroulement (gros fil) appelé la bobine d'appel (faible résistance) alimenté qu'un court instant donne de la puissance au flipper.

Le deuxième (fil fin) appelé la bobine de maintien (dont la résistance est plus importante) est alimenté tant que le joueur appuie sur le bouton de commande et maintient le batteur du flipper "en l'air".



La bobine de maintien est court-circuitée au repos par le contact fin de course qui est fermé (EOS = "end of stroke"). Quand le joueur appuie sur le bouton de commande du flipper, le courant n'alimente que l'enroulement de puissance, le plongeur s'enfonce violemment (la bobine donne toute sa puissance pendant un court instant), la biellette vient ouvrir le contact permettant d'activer l'enroulement en fil fin et de court-circuiter l'enroulement de puissance. La consommation de la bobine diminue fortement juste ce qu'il faut pour maintenir le batteur "en l'air". Quand le joueur relâche le bouton de commande, le plongeur revient à sa position de départ, le contact EOS se referme prêt pour un nouvel appel.

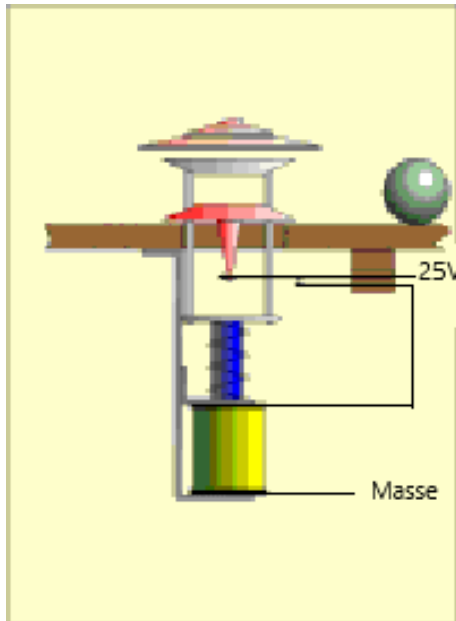
Attention, le réglage du contact EOS est important, si le contact ne s'ouvre pas, la bobine va surchauffer et finir par griller, si le contact n'est pas bien fermé au repos, le flipper va manquer de puissance.



d) Pop bumper

Décomposition d'un pop bumpers





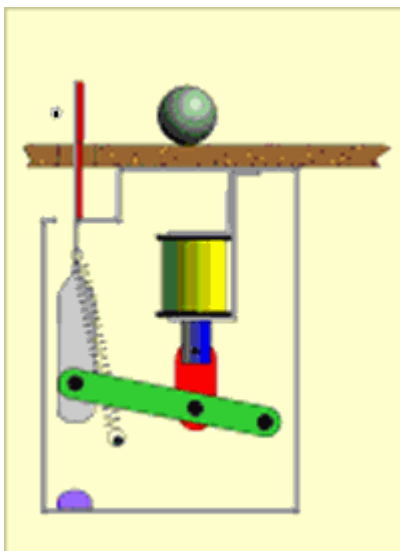
Fonctionnement

La bille en arrivant sur l'assiette (partie rouge) appuie sur la partie "haute" de l'interrupteur à lame (EOS) ce qui a pour effet de fermer le contact. En recevant du courant, la bobine crée un champ magnétique. Le plongeur (en bleu) est alors attiré vers le bas en emportant la coupelle. En descendant, elle repousse la bille à l'extérieur du pop bumpers, la bille qui n'appuie plus sur l'assiette libère alors le contact. La coupelle remonte grâce à un ressort.

e) Cibles

Il s'agit d'une cible verticale, qui enregistre chaque impact de la bille et incrémente le score.

Cible tombante



Fonctionnement

Lorsque la bille heurte la cible, cette dernière (qui jusque-là reposait sur une partie en métal grâce au ressort) tombe. En tombant elle fait pression sur un contact (demi-cercle violet), qui envoie alors une tension à l'Arduino qui ajoute des points. L'Arduino attend la fin de la partie pour envoyer un courant dans la bobine qui comme précédemment permet de faire remonter le plongeur. Il entraîne avec lui le bras (vert) qui remonte la cible, avec l'aide du ressort elle se repositionne à son emplacement de repos. Un bout de caoutchouc est placé derrière la cible (petit rond blanc) pour amortir la cible quand la bille arrive dessus (sans lui les cibles se casseraient).

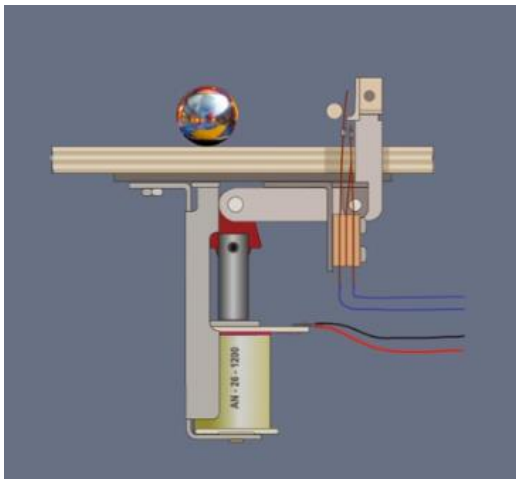
Cible tournante



Fonctionnement

La bille arrive sur le portillon (en damier), la force de la bille contraint le damier gris à se soulever (en tournant). En se soulevant il tire sur la tirette blanche qui tire sur le crochet à son extrémité ce qui ferme le contact à lames. Ainsi le contact est détecté par l'Arduino, il peut compter les points. Mais le portillon continue à tourner après le passage de la bille et donc continue à ouvrir et fermer le contact jusqu'à son arrêt. Il faut donc que l'Arduino ne prenne en compte que le premier contact.

f) Slingshots



Fonctionnement

Lorsque la bille arrive sur le slingshot, elle pousse la barre du contact EOS, et ainsi ferme l'interrupteur. La bobine reçoit alors un courant et se met en marche. Par le même principe qu'avant, le plongeur est attiré vers le bas ce qui tire la bascule en avant. En basculant, le bras gris repousse la bille vers l'extérieur et ré-ouvre le contact EOS. La bobine n'est donc plus alimentée, le plongeur peut (à l'aide d'un ressort) remonter à sa position initiale. Le rond blanc entre la bille et la contact EOS est un élastique qui entoure le slingshot.

g) Les composants

Dans tous les modèles que nous avons vus les éléments de flipper sont construits avec des bobines et d'autres composants décrits ci-dessous.

Le plongeur, un cylindre en "fer doux" est fait en matériau ferromagnétique (s'aimante lorsqu'il se trouve dans un champ magnétique). Cependant, il ne garde pas le magnétisme lorsque le champ est supprimé.



Le plongeur coulisse dans la bobine mais entre les deux se trouve un fourreau en plastique dans lequel le plongeur glisse facilement. Il a un petit épaulement lui permettant de rester en place dans la bobine.

Pour transmettre le mouvement du plongeur on ajoute une bielle, qui relie donc le plongeur avec la coupelle pour les pop bumpers et le batteur pour les flippers.

Pour ramener le plongeur à sa position initiale on ajoute un ressort. Ainsi quand le plongeur n'est plus retenu par la bobine le ressort qui était comprimé se détend et fait remonter le plongeur. Ainsi l'assiette remonte dans le cas des pop bumpers et le flipper se remet dans sa position initiale.

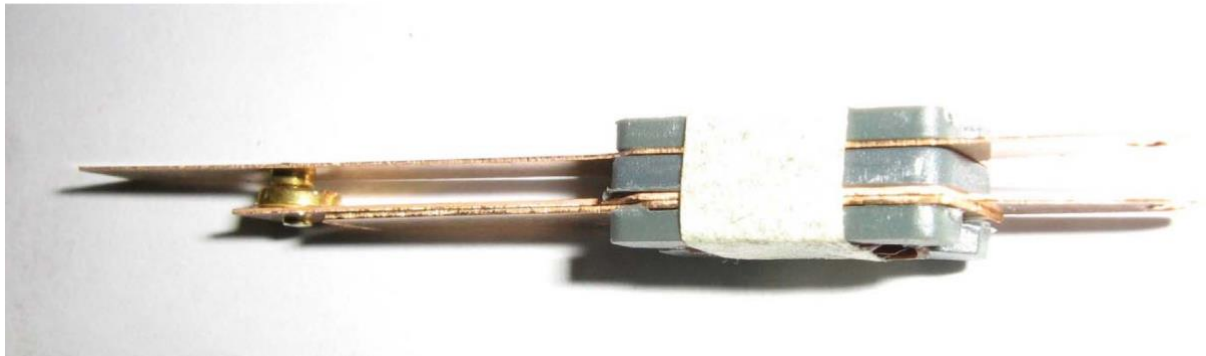
La bobine est fixée par deux "équerres". Une en arrière encaisse les chocs du plongeur. L'autre, en avant, retient le fourreau dans la bobine et laisse passer le plongeur par un trou. Ces 2 équerres sont attachées à une plaque de métal qui est fixée sous le plateau de jeu.



Pour les flippers, un guide axe est ajouté sur la platine c'est là que vient se loger l'axe du batteur.



Sur les flippers à commande électronique l'EOS possède des pastilles de contact de petites dimensions et les fils qui y sont connectés sont reliés à la carte arduino.



5) Liaison Arduino-plateau de jeu

Pour compter les points lorsqu'un bumper est touché par exemple, il faut relier l'Arduino au dispositif avec la bobine. Cependant, la bobine fonctionne avec 25V tandis que l'Arduino ne supporte que 5V. Ainsi, il faut, pour les relier, diminuer la tension que l'Arduino reçoit. Nous avons étudié trois manières pour le faire : grâce à des transistors, par un diviseur de tension ou bien par l'intermédiaire d'un capteur photorésistance.

a) MOSFET

Les MOSFET sont des transistors à effet de champ métal-oxyde-semi conducteur, ils servent, entre autres, à amplifier une tension.

Les MOSFET ont 3 entrées, une entrée SOURCE (1), une entrée GATE (2) et une entrée DRAIN (3). L'entrée 2, GATE est connectée à la bobine, l'entrée 3 (DRAIN) est connectée à l'Arduino et la SOURCE (1) est connectée à la masse.

La tension en sortie du MOSFET a diminué et peut rentrer directement dans l'Arduino.

Les MOSFET peuvent être câblés directement sur les bobines ou bien comme dans le modèle 2, on peut construire une carte de contrôle des MOSFET. De cette façon, nous pouvons facilement connecter et déconnecter nos solénoïdes à l'Arduino, cette solution permet aussi de remplacer plus facilement un mauvais MOSFET.

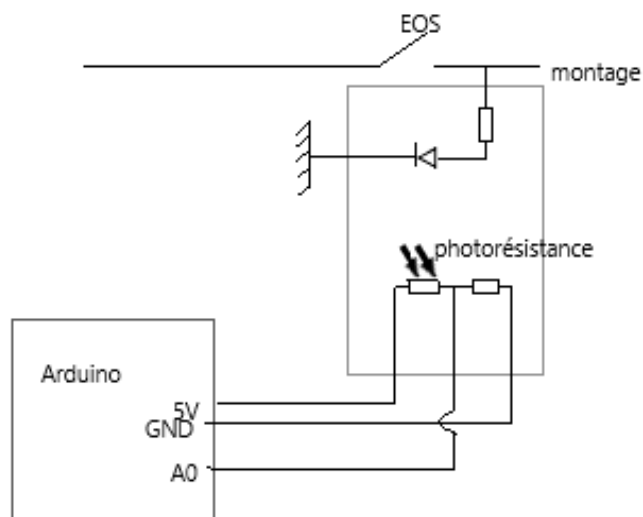
b) Diviseur de tension

La carte Arduino est reliée au montage par deux résistances connectées en série. Une tension U (ici 25 V) est appliquée en entrée sur ces deux résistances et la tension de sortie est mesurée aux bornes de la deuxième résistance. Ainsi, grâce à la loi d'Ohm : $U = R \cdot I$, on peut déterminer la valeur de nos résistances pour avoir la tension de sortie voulue (ici 5V). Il faut ensuite relier notre sortie à l'entrée analogique de notre carte Arduino et la convertir en valeur numérique.

c) Capteurs photorésistance

Afin de protéger entièrement l'Arduino nous pouvons utiliser une photorésistance.

Lorsque le contact EOS est fermé, il alimente une LED, dans une petite boîte, qui s'allume (puisque le courant passe). De l'autre côté se trouve une photorésistance. Elle détecte la lumière de la LED et ainsi sa résistance diminue, ce qui augmente la tension que reçoit l'Arduino. L'Arduino comprend alors que le bumper a été activé et ajoute des points au joueur.



6) Conclusion : notre flipper

Tout d'abord, dans notre flipper nous mettrons deux éléments avec bobine : les flippers (batteurs) et à priori un pop-bumper (au détriment des slingshots). Par ailleurs, nous avons choisi le modèle de cible tournante.

De plus, nous avons décidé de construire un lance bille seulement si le temps nous le permet. La bille sera donc directement déposée en haut du flipper sur une pente. Nous utiliserons les capteurs de forces pour déterminer les débuts et fins de parties.

Par ailleurs, pour les LEDs nous voulons utiliser une chaîne de LED (comme pour le modèle 3). Positionnées sur notre plateau, elles sont directement reliées à notre carte Arduino et changeront de couleur en fonction des situations (début/fin de partie, pop-bumpers en actions...).

Nous ne mettrons pas de son dans notre flipper et l'affichage des points s'effectuera sur un écran LCD.

Enfin, pour relier notre carte Arduino au pop bumper nous avons décidé d'utiliser un capteur photorésistance.

Nos sources

modèle de flipper 1 :

<https://create.arduino.cc/projecthub/BobB/arduino-controlled-pinball-machine-525863>

modèle de flipper 2 :

https://create.arduino.cc/projecthub/Frogger1108/homemade-arduino-pinball-machine-4a3314?ref=search&ref_id=pinball&offset=3

modèle de flipper 3:

https://create.arduino.cc/projecthub/Frogger1108/arduino-conversion-of-an-em-pinball-machine-b2f318?ref=tag&ref_id=pinball&offset=1

Les flippers : <http://the.nerv.free.fr/docpro/flippers.pdf>

Les pop-bumper : <https://www.flipperfrance.com/threads/comment-cela-marche.10/>

Mofest : <https://physiquemangin.pagesperso-orange.fr/BTSSE/cours/transistors.pdf>

Capteur force : <https://learn.adafruit.com/force-sensitive-resistor-fsr/using-an-fsr>

Capteur opto-couleur :

https://sonelec-musique.com/electronique_theorie_optocoupleur.html