Boîte inutile

Le projet représente une implémentation simple d’une boîte inutile, un jouet simple où lors de la commutation d’un interrupteur, un mécanisme est déclenché et un bras sort de la boîte pour changer à nouveau l’interrupteur. Ils sont commercialisés et sont devenus populaires ces derniers temps. De nombreuses variantes de ceux-ci sont disponibles, où les appareils font d’autres choses en plus d’appuyer sur le bouton ou sont embellis pour ressembler davantage à un jouet.

Descriere generală

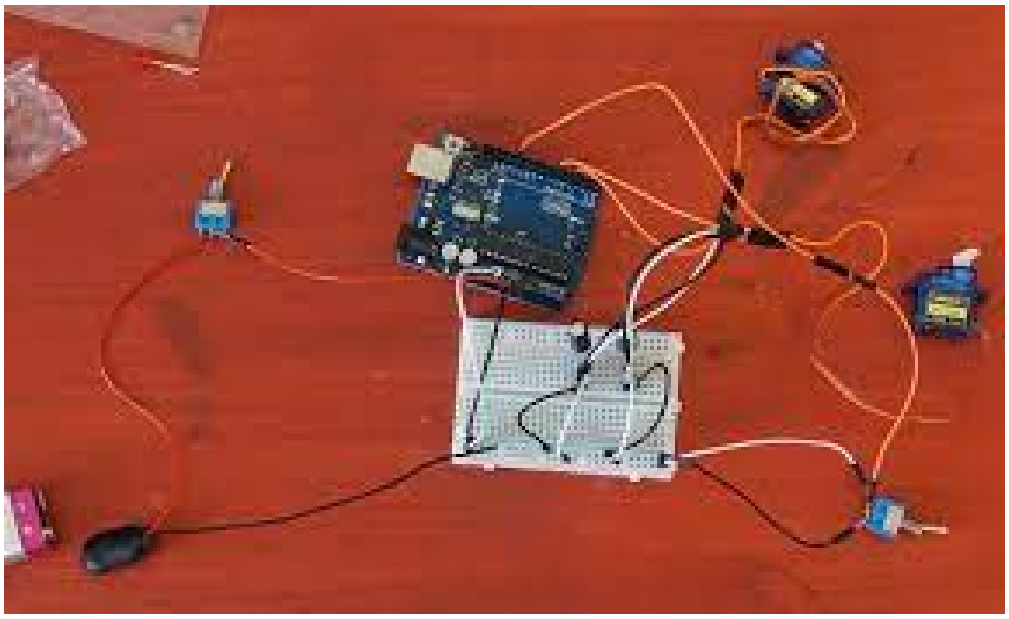
L’appareil se compose d’un interrupteur avec 2 rangées et 3 positions et 2 actionneurs. Lorsque vous appuyez sur le bouton, l’appareil qui agit sur les actionneurs est rétrogradé à son tour. La première fois qu’un actionneur est actionné, ce qui a pour rôle de soulever le couvercle de la boîte. Ensuite, un autre actionneur est actionné qui a pour rôle de mettre l’interrupteur en position d’arrêt.

Les 2 actionneurs ont fixé un bras en aluminium pour avoir assez de puissance pour les déplacer. Les actionneurs sont petits (utilisés davantage dans les jouets) et n’ont pas le pouvoir de déplacer des choses trop lourdes. Une fois que les servos ont terminé la rotation définie par le programme, ils reviennent à leur état d’origine jusqu’à ce que le bouton soit enfoncé à nouveau.

Hardware Design

Les composant de mon projet ce sont ici:

- Arduino UNO R3 - Commutateur - 2 rangées 3 pos - Servomoteur x2 - Feu - Batterie 9V - A daptateur de batterie - Planche à pain



Software Design

Pour la conception des logiciels pour mon projet:

- Environnement de développement : Arduino IDE. - Bibliothèques utilisées : Servo, Stdlib (pour générer un nombre aléatoire).

Dans la mise en œuvre, nous avons utilisé un commutateur externe pour détecter le commutateur à bouton et les fonctions de bibliothèque pour le fonctionnement des actionneurs.

J'ai commencé par emmêler les servomoteurs. J'ai attaché les broches 8 et 9 respectivement et je les ai réglées sur 0 au début du programme pour réinitialiser dans n'importe quel état qui serait resté en cours de route. J'ai défini la broche 4 comme entrée pour détecter les changements de bouton. J'ai activé les interruptions externes et je ne les ai activées que pour la broche 4.

Dans la fonction de gestion des interruptions ne changent qu'une variable globale, les fonctions du mécanisme sont appelées dans la boucle(). Nous n'avons choisi que le changement de la variable globale dans le traitement de l'interruption parce que le mécanisme a une très longue durée de vie et aurait grandement interrompu le cours du programme si nous gardions tout en interruption.

Nous avons créé plusieurs fonctions, toutes similaires les unes aux autres, pour les mouvements du mécanisme. La première fois que la rotation de l'actionneur 1 est exécutée, ce qui a pour rôle de soulever le couvercle.

Ensuite, la rotation de l'actionneur 2 est exécutée pour activer le bouton.

Ensuite, dans l'ordre inverse, il revient à son état d'origine. J'ai utilisé différents délais entre les fonctions pour changer la vitesse des actionneurs ou le temps d'attente entre les exécutions (entre les 2 actionneurs).

Dans la boucle, j'utilise un interrupteur pour détecter quelle fonction appeler et un index i qui est choisi au hasard dans une plage pour rendre le jouet un peu plus indéterminé et drôle.

En tant que fonctions utilisées :

- servo.atach pour fixer une broche à l'actionneur. - servo.write pour définir l'angle de l'actionneur. - retarder. - bord. - sei et cli pour définir les interruptions.

Rezultate Obţinute

J'ai obtenu une copie fonctionnelle d'un jouet populaire dans sa version de base. Dans la démo, vous pouvez voir comment cela fonctionne et comment le bouton est d'abord commuté par moi, puis commuté à nouveau par le jouet. Vous pouvez voir 4 vitesses et lecteurs différents.

https://youtube.com/shorts/f99pmCTBSms?feature=share

Concluzii

J'ai pu comprendre les notions présentées au laboratoire et les appliquer. C'était très intéressant, même si j'aurais préféré choisir quelque chose d'un peu plus complexe mon projet étant assez simple à réaliser. J'ai également rencontré quelques difficultés dues à la puissance des actionneurs et à la résistance du bouton initial, mais j'ai réussi à remplacer les composants et à surmonter les problèmes.

ici le fichier du code source:

codesource.txt

Jurnal

- Page de réalisation - Achat de composants - Document - Schéma de développement à Tinkercat - Création de périphériques (boîte, lames en aluminium) - Assemblage de composants - Programmation agréable - Rédaction de documentation - Finir la page wiki

Bibliografie/Resurse

https://fr.wikipedia.org/wiki/Machine_inutile

https://forums.futura-sciences.com/electronique/790636-une-machine-inutile.html

http://www.wiki-rennes.fr/Useless_Box

Export to PDF

From:
http://ocw.cs.pub.ro/courses/ - **CS Open CourseWare**

Permanent link:
http://ocw.cs.pub.ro/courses/pm/prj2022/avaduva/magic_music_box

Last update: **2022/06/03 12:14**