

Détecteur automatique de but

Au babyfoot

Introduction

Ce projet explore l'intégration de capteurs et de composants électroniques avec Arduino Uno pour créer un système interactif de détection de mouvement. En utilisant un capteur de mouvement PIR, une LED, un buzzer et un afficheur 7 segments, il démontre comment Arduino peut détecter les mouvements, afficher des informations visuelles et sonores, tout en suivant le nombre de détections.

Notre réalisation se concentre spécifiquement sur un détecteur automatique de but pour un jeu de baby-foot. Le système utilise un capteur de mouvement PIR pour surveiller le passage de la balle en continu. Chaque fois qu'une balle passe, le capteur détecte le mouvement et incrémente un compteur LED correspondant. À chaque but marqué, un son de félicitation retentit pour signaler le succès.

Cette solution combine l'utilisation du capteur de mouvement pour détecter les événements, la gestion d'un compteur LED pour suivre les buts marqués, et l'émission d'un signal sonore pour célébrer les réalisations des joueurs. En résumé, ce système offre une approche automatisée et amusante pour suivre les scores lors d'une partie de baby-foot, tout en illustrant les possibilités créatives et pratiques d'Arduino dans le domaine des applications interactives et ludiques.

Matériels nécessaires

Pour mener à bien cette expérience, nous avons utilisé les composants suivants :

- 1x Carte Arduino Uno
- 1x Câble USB d'alimentation
- 1x Afficheur 7 segments
- 4x Résistance 1 kΩ
- 1x Buzzer
- 1x Capteur de présence infra-rouge
- Fils de connexion (fils au total)

Difficultés rencontrées

- Nombre de pin limité du a l'importante utilisation de câble
- L'afficheur 7 Segments affichait les nombres dans la loupe et clignotait

Circuit

- Capteur IF ne capte pas la balle car pas de rayonnement infrarouge émis par celle ci

Code

Pour réaliser notre projet nous avons utilisé la librairie "SevSeg.h"

- `begin(hardwareConfig, numDigits, digitPins, segmentPins, resistorsOnSegments)` : Configure l'affichage avec les paramètres spécifiés, y compris le type de matériel, le nombre de chiffres et les broches utilisées.
- `setBrightness(brightness)` : Régle la luminosité de l'affichage. La luminosité peut être réglée de 0 (éteint) à 100 (luminosité maximale).
- `setNumber(num, decimalPlaces)` : Définit le nombre à afficher. Vous pouvez également spécifier le nombre de places décimales pour les nombres à virgule flottante.

On initialise un certain nombre de variables :

```
#define PIR A5
#define LED 13
#define BUZZER 14 // Définir la broche A0 comme sortie numérique
int score = 0;
//const int sensitivity = 50;

enum DetectionState { IDLE, DETECTED, DEBOUNCING };
DetectionState state = IDLE;
```

Configuration des broches

```
pinMode(PIR, INPUT);  
pinMode(LED, OUTPUT);  
pinMode(BUZZER, OUTPUT);
```

PIR : Configure la broche du capteur PIR (A5) comme une entrée, permettant à l'Arduino de lire l'état du capteur.

LED : Configure la broche de la LED (13) comme une sortie, permettant à l'Arduino d'allumer ou d'éteindre la LED.

BUZZER : Configure la broche du buzzer (14) comme une sortie, permettant à l'Arduino de contrôler le buzzer.

- numDigits : Définit le nombre de chiffres de l'affichage à 7 segments. Ici, il y a 4 chiffres et fonctionne par balayage.
- digitPins : Déclare un tableau des broches Arduino connectées aux chiffres de l'affichage à 7 segments.
- ²segmentPins : Déclare un tableau des broches Arduino connectées aux segments de l'affichage à 7 segments.

Améliorations

- Avoir un son plus personnalisé (utiliser la librairie pitches.h)
- Utiliser moins de câble pour Rendre cela plus compact pour une installation plus facile sur un babyfoot (utiliser une matrice)
- Avoir une source d'énergie indépendante (utiliser la pile)
- Utiliser un capteur plus adapté

Conclusion

En conclusion, cette expérience nous a permis de comprendre les bases de la programmation et de l'intégration matérielle avec Arduino. Nous avons appris comment utiliser différents capteurs pour collecter des données environnementales, et comment utiliser ces données pour créer des systèmes interactifs et réactifs

Sources

<https://ledisrupteurdimensionnel.com/arduino/fonctionnement-de-lafficheur-de-7-segments-a-4-digits/#code>

<https://blog.manuel-esteban.com/tuto-lire-un-capteur-infrarouge-avec-arduino/>