

Table de matière

Table de matière	1
Listes des figure	3
Principales notations	4
Introduction	5
Objectif de projet.....	6
Partie 1 : Les outils utilisées	7
1. Définition de Python.....	8
2. Définition de C++	8
3. Définition d'Arduino	8
4. Définition de Raspberry Pi	8
5. Définition de Capteur de mouvement (PIR).....	8
6. Définition de LED.....	8
7. Définition de Caméra	8
8. Définition de Traitement d'image.....	9
9. Définition de Buzzer	9
10. Définition de GPIO.....	9
11. Définition d'Interface graphique GUI.....	9
Partie II : Système d'éclairage automatique par detection de mouvement	10
1. Introduction	11
2. Rôle d'Arduino	12
3. Composants utilisés	13
4. Principe de fonctionnement.....	14
5. Logique de commande	15
6. Programme Arduino	16
7. Schéma du montage et son résultats	17
• Schéma du montage	17
• Résultat.....	17
8. Conclusion.....	18
Partie III : Détection de goal et signalisation	19
1. Introduction	20
2. Rôle de Raspberry Pi	21
3. Les composantes utilisées.....	22
• Caméra.....	22
• Buzzer.....	24
4. Conclusion.....	25
Partie IV : L'interface graphique	26
1. Introduction.....	27
2. Technologies utilisées.....	28
3. Fonctionnement de l'interface graphique.....	31
4. Avantage de l'interface graphique	32
5. Conclusion	33

Conclusion.....	34
Bibliographie.....	35
Abstract.....	36
Résumé	36

Liste de figure

Figure 1 : Code Arduino.....	16
Figure 2 : Schéma du montage dans WOKWI.....	17
Figure 3 : Résultat du code.....	17
Figure 4 : Historique des vidéos de goals.....	22
Figure 5 : Caméra – Détection zone de goal.....	23
Figure 6 : Détection ballon dans goal.....	23
Figure 7 :Schéma circuit Raspberry et Buzzer.....	24
Figure 8: Interface graphique tkinter.....	28
Figure 9 :Configuration et initialisation.....	29
Figure 10 : Sauvegarde du score et nettoyage système.....	30

Principales notations

ABREVIATION	SIGNIFICATION
EST	Ecole Supérieur de Technologie
GIE	Génie Informatique Embarqué
PIR	Passive Infrared Sensor
GPIO	General Purpose Input/Output
GUI	Graphical User Interface
CPU	Central Processing Unit
I/O	Input/Output
LED	Light Emitting Diode

Introduction

Dans le cadre de ce projet, nous avons réalisé un système de match intelligent reposant sur l'intégration de deux plateformes largement utilisées dans le domaine des systèmes embarqués : Arduino et Raspberry Pi.

L'idée principale est de concevoir un terrain de football miniature capable de réagir automatiquement aux événements du jeu, sans intervention humaine directe.

Le système développé permet notamment l'éclairage automatique du terrain, la détection intelligente des buts à l'aide d'une caméra, ainsi que la gestion et l'affichage du score en temps réel à travers une interface graphique conviviale. Lorsqu'un but est détecté, le système déclenche également un signal sonore afin de signaler l'événement aux joueurs ou aux spectateurs.

Ce projet illustre la complémentarité entre Arduino, dédié aux tâches de contrôle matériel et d'actionneurs, et Raspberry Pi, utilisé pour le traitement d'images, la gestion logique du match et l'affichage graphique.

Il s'inscrit pleinement dans le domaine des systèmes embarqués intelligents et permet de mettre en pratique des compétences en électronique, programmation, automatisation et interaction homme-machine.

Objectif de projet

L'objectif principal de ce projet est de concevoir un système automatisé et intelligent permettant de gérer un match de football miniature de manière autonome. Plus précisément, les objectifs sont les suivants :

- ❖ Mettre en place un système d'éclairage automatique du terrain basé sur la détection de mouvement ou de présence.
- ❖ Assurer la détection automatique des buts en utilisant une caméra connectée au Raspberry Pi.
- ❖ Signaler les événements importants du match, notamment les buts, à travers un affichage visuel et un signal sonore (buzzer).
- ❖ Développer une interface graphique permettant l'affichage du score et des informations du match en temps réel.
- ❖ Concevoir un système fiable, interactif et intelligent, combinant matériel et logiciel pour améliorer l'expérience de jeu.

Partie I :

Les outils utilisées

Dans cette partie, nous présentons les principales notions et termes techniques utilisés dans la réalisation de ce projet afin de faciliter la compréhension du système développé.

❖ **Python :**

Python est un langage de programmation interprété, polyvalent et multi-plateforme, créé en 1991 par Guido van Rossum. Il est reconnu pour sa syntaxe claire, lisible et proche du langage naturel, ce qui facilite son apprentissage et son utilisation.

❖ **C++ :**

Le C++ est un langage de programmation généraliste, orienté objet, créé par Bjarne Stroustrup au début des années 1980 comme extension du langage C. Il combine la programmation procédurale héritée du C avec des concepts de programmation orientée objet, tels que les classes, l'héritage, le polymorphisme et l'encapsulation.

❖ **Arduino :**

Carte électronique programmable utilisée pour le contrôle des capteurs et des actionneurs. Dans ce projet, Arduino assure la gestion de l'éclairage automatique du terrain à partir des informations fournies par le capteur de mouvement.

❖ **Raspberry Pi :**

Nano-ordinateur capable d'exécuter un système d'exploitation. Il est utilisé dans ce projet pour le traitement d'images, la détection des goals, la gestion du buzzer et l'exécution de l'interface graphique.

❖ **Capteur de mouvement (PIR) :**

Capteur infrarouge passif permettant de détecter la présence ou le mouvement. Il est utilisé pour déclencher automatiquement l'éclairage du terrain.

❖ **LED :**

Composant électronique émettant de la lumière lorsqu'il est alimenté. Les LEDs sont utilisées pour simuler l'éclairage du terrain.

❖ **Caméra :**

Dispositif de capture d'images utilisé pour surveiller la zone du but. Les images sont analysées par le Raspberry Pi pour détecter le passage du ballon.

❖ **Traitement d'image :**

Ensemble de techniques informatiques permettant d'analyser et d'interpréter des images numériques. Il est utilisé pour détecter automatiquement les buts.

❖ **Buzzer :**

Dispositif sonore permettant d'émettre un signal audio. Il est activé lorsqu'un goal est détecté afin de signaler l'événement.

❖ **GPIO :**

Broches d'entrée/sortie du Raspberry Pi utilisées pour connecter et contrôler les composants électroniques tels que le buzzer.

❖ **Interface graphique (GUI) :**

Environnement visuel permettant l'interaction entre l'utilisateur et le système. Dans ce projet, elle affiche le score du match en temps réel.

Partie II :

Système d'éclairage utilisées

1. Introduction :

Cette partie du projet est dédiée à l'utilisation de la carte Arduino pour la mise en œuvre d'un système d'éclairage automatique du terrain. L'objectif principal est d'assurer un éclairage intelligent et économe en énergie, en adaptant le fonctionnement des LEDs en fonction de la présence ou de l'activité sur le terrain.

Le système repose sur la détection de mouvement, réalisée à l'aide d'un capteur approprié connecté à la carte Arduino. Lorsqu'un mouvement est détecté sur le terrain, Arduino traite l'information reçue du capteur et commande automatiquement l'allumage des LEDs afin d'éclairer la zone de jeu. En revanche, en l'absence de mouvement pendant une période donnée, le système éteint automatiquement les LEDs, ce qui permet de réduire la consommation énergétique et d'optimiser l'utilisation de l'éclairage.

Cette approche permet non seulement d'améliorer l'autonomie et l'efficacité du système, mais aussi d'illustrer le rôle d'Arduino dans le contrôle des actionneurs et la gestion des entrées/sorties dans un système embarqué. Elle constitue ainsi une base essentielle pour la réalisation d'un système intelligent et automatisé au sein du projet global.

2. Rôle de Arduino :

Dans ce projet, la carte Arduino joue le rôle de contrôleur principal pour la gestion du système d'éclairage automatique du terrain. Elle assure le traitement des informations provenant des capteurs et le pilotage des actionneurs de manière autonome et en temps réel.

Plus précisément, l'Arduino est chargé de lire en continu les données fournies par le capteur de mouvement (PIR) afin de détecter toute présence ou activité sur le terrain. Dès qu'un mouvement est détecté, l'Arduino analyse le signal reçu et envoie une commande appropriée pour allumer automatiquement les LEDs, assurant ainsi un éclairage immédiat de la zone de jeu.

En l'absence de mouvement pendant un certain temps, la carte Arduino commande l'extinction des LEDs, ce qui permet de réduire la consommation d'énergie et d'optimiser l'utilisation du système d'éclairage.

Cette partie du projet fonctionne de manière totalement autonome, sans intervention humaine, ce qui illustre parfaitement le rôle d'Arduino dans les systèmes embarqués dédiés au contrôle et à l'automatisation.

3. Composants Utilisés :

La réalisation du système d'éclairage automatique repose sur l'utilisation des composants suivants :

❖ **Carte Arduino :**

Elle constitue le cœur du système. Elle permet de traiter les signaux reçus du capteur de mouvement et de commander l'allumage ou l'extinction des LEDs selon les conditions détectées.

❖ **Capteur de mouvement (PIR) :**

Ce capteur est utilisé pour détecter la présence ou le mouvement sur le terrain. Il envoie un signal électrique à l'Arduino lorsqu'un mouvement est détecté, déclenchant ainsi le processus d'éclairage.

❖ **LEDs d'éclairage (6 LEDs) :**

Les LEDs sont utilisées pour simuler l'éclairage du terrain. Elles sont commandées par l'Arduino et s'allument uniquement lorsque le capteur détecte un mouvement.

❖ **Fils de connexion :**

Ils assurent les liaisons électriques entre la carte Arduino, le capteur de mouvement et les LEDs, garantissant ainsi le bon fonctionnement du montage.

4. Principe de fonctionnement :

Le système d'éclairage automatique repose sur la détection de mouvement à l'aide d'un capteur infrarouge passif (PIR) connecté à la carte Arduino. Le fonctionnement global se déroule selon une séquence simple et efficace.

Tout d'abord, le capteur PIR surveille en permanence la zone du terrain afin de détecter toute présence ou mouvement. Lorsqu'un mouvement est détecté, le capteur génère un signal logique HIGH, indiquant à l'Arduino qu'une activité est présente sur le terrain.

À la réception de ce signal, l'Arduino traite l'information et active immédiatement les sorties numériques reliées aux LEDs. Les LEDs passent alors à l'état HIGH, ce qui correspond à leur allumage et permet d'éclairer automatiquement le terrain.

En revanche, lorsqu'aucun mouvement n'est détecté pendant une certaine durée, le capteur PIR envoie un signal logique LOW. L'Arduino interprète ce signal comme une absence d'activité et commande l'extinction des LEDs en les plaçant à l'état LOW. Ce fonctionnement permet d'assurer un éclairage intelligent, automatique et économe en énergie.

5. Logique des commandes :

La logique de commande du système est basée sur une condition simple de type si/alors, ce qui rend le fonctionnement clair et fiable.

❖ **Si un mouvement est détecté par le capteur PIR**

→ **le signal devient HIGH**

→ **l'Arduino allume les LEDs (état HIGH).**

❖ **Si aucun mouvement n'est détecté**

→ **le signal devient LOW**

→ **l'Arduino éteint les LEDs (état LOW).**

Cette logique garantit une réponse rapide du système aux changements d'état du terrain. Elle illustre l'utilisation des entrées numériques (capteur PIR) et des sorties numériques (LEDs) dans un système embarqué basé sur Arduino.

6. Programme Arduino :

Le programme Arduino lit l'état du capteur PIR et commande les sorties reliées aux 6 LEDs .

Le code a été simulé et validé à l'aide de la plateforme WOKWI .

```
sketch.ino • diagram.json • Library Manager ▼
1  #define PIR_PIN 2
2  #define LED1 3
3  #define LED2 4
4  #define LED3 5
5  #define LED4 6
6  #define LED5 7
7  #define LED6 8
8  #define LED7 9
9  #define LED8 10
10 void setup() {
11     pinMode(PIR_PIN, INPUT);
12     pinMode(LED1, OUTPUT); pinMode(LED2, OUTPUT);
13     pinMode(LED3, OUTPUT); pinMode(LED4, OUTPUT);
14     pinMode(LED5, OUTPUT); pinMode(LED6, OUTPUT);
15     pinMode(LED7, OUTPUT); pinMode(LED8, OUTPUT);
16     Serial.begin(9600);}
17 void loop() {
18     int motion = digitalRead(PIR_PIN);
19     digitalWrite(LED1, motion);
20     digitalWrite(LED2, motion);
21     digitalWrite(LED3, motion);
22     digitalWrite(LED4, motion);
23     digitalWrite(LED5, motion);
24     digitalWrite(LED6, motion);
25     digitalWrite(LED7, motion);
26     digitalWrite(LED8, motion);}
```

Fig 1 : Code Arduino

7. Schéma du montage WOKWI et son résultat :

➤ Schéma du montage :

Le schéma de montage a été réalisé sur WOKWI afin de vérifier le bon fonctionnement du système.

- Le capteur PIR est connecté à une entrée digitale de l'Arduino.
- Les 6 LEDs sont connectées aux sorties digitales de l'Arduino.
- Aucune résistance n'a été utilisée dans la simulation, car Wokwi gère automatiquement les contraintes électriques nécessaires.

Le schéma permet de visualiser clairement les connexions et de valider le montage avant une implémentation réelle.

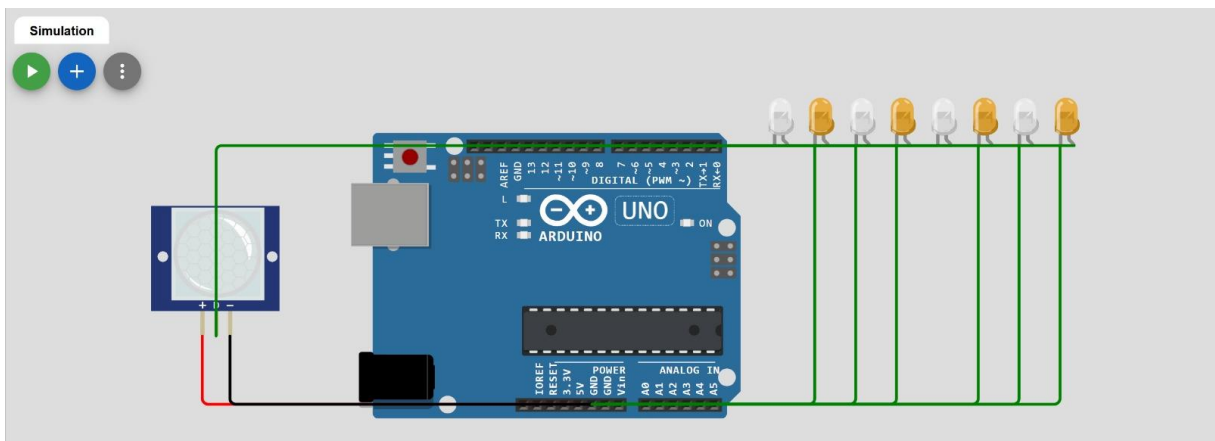


Fig 2 : Schéma du montage dans WOKWI

➤ Résultat :

- Les LEDs s'allument automatiquement dès qu'un mouvement est détecté.
- Les LEDs s'éteignent lorsque le terrain est vide.
- Le système fonctionne de manière stable et automatique.

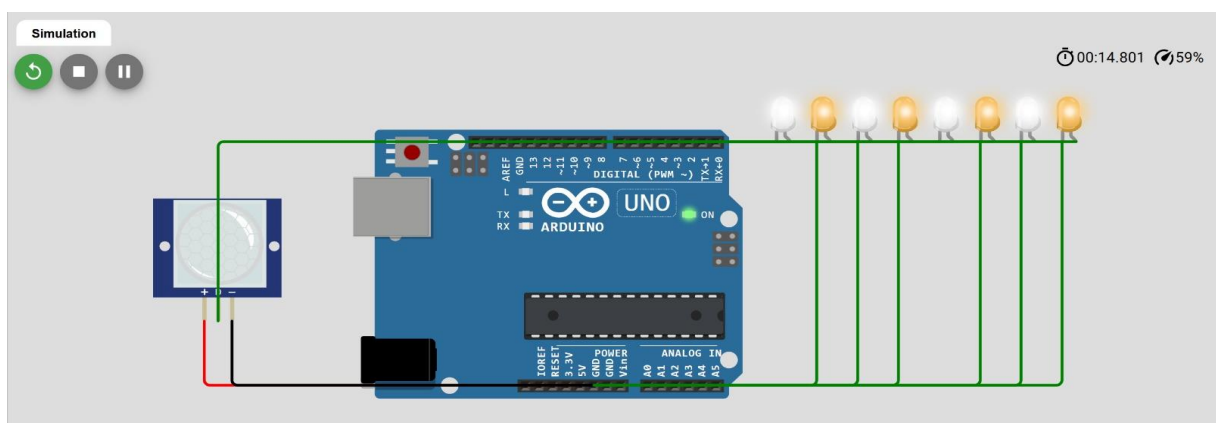


Fig 3 : Résultat du code

8. Conclusion :

La partie Arduino a permis de mettre en place un système d'éclairage automatique efficace et intelligent basé sur la détection de mouvement.

Grâce à l'utilisation d'un capteur PIR et d'une logique simple HIGH/LOW, l'éclairage du terrain est activé uniquement en cas de besoin, ce qui permet une économie d'énergie et une meilleure automatisation du système.

Cette partie constitue un module autonome qui peut être facilement intégré avec d'autres systèmes pour former une solution complète.

Partie III :

Détection de goal et signalisation

1. Introduction :

La partie Raspberry Pi constitue le cœur intelligent du système de détection automatique de goal. Elle repose sur un système embarqué capable de traiter des données en temps réel et de gérer les différents composants du projet. Grâce à sa puissance de calcul et à sa flexibilité, le Raspberry Pi permet l'intégration de la vision par ordinateur pour analyser les images capturées par une caméra.

Dans ce projet, le Raspberry Pi est chargé de détecter automatiquement le passage du ballon dans la zone du but à l'aide d'algorithmes de traitement d'image. Une fois le goal détecté, il déclenche un signal sonore via un buzzer afin d'indiquer immédiatement l'événement.

Ainsi, la partie Raspberry Pi assure la capture, le traitement et la signalisation du goal, offrant un système autonome, fiable et réactif, sans recours à un écran LCD.

2. Rôle du Raspberry Pi

Le Raspberry Pi joue le rôle d'une unité centrale de traitement et de décision au sein du système. Il exécute le programme principal chargé de coordonner la capture des images, leur analyse et la détection du goal.

À partir des images fournies par la caméra, le Raspberry Pi applique des algorithmes de traitement d'image afin d'identifier le passage du ballon dans la zone du but. Lorsque les conditions de détection sont validées, il prend la décision finale.

Suite à cette détection, le Raspberry Pi active le buzzer via les broches GPIO afin de fournir une signalisation sonore immédiate. Il assure ainsi le traitement des données, la prise de décision et le contrôle des dispositifs de sortie, garantissant un fonctionnement autonome et efficace du système.

3-1 Caméra et Détection de Goal :

La caméra constitue le principal capteur du système. Elle est connectée au Raspberry Pi et positionnée de manière à couvrir en permanence la zone du but. Les images ou le flux vidéo sont capturés en continu et transmis au Raspberry Pi pour être analysés. L'algorithme de détection repose sur des techniques de traitement d'image telles que la détection de mouvement, la segmentation de la zone du but et l'analyse de la trajectoire du ballon. Ces étapes permettent d'identifier le moment où le ballon franchit la ligne du but.

Une fois cet événement détecté et validé, un signal interne est généré par le système afin de déclencher l'activation du buzzer.

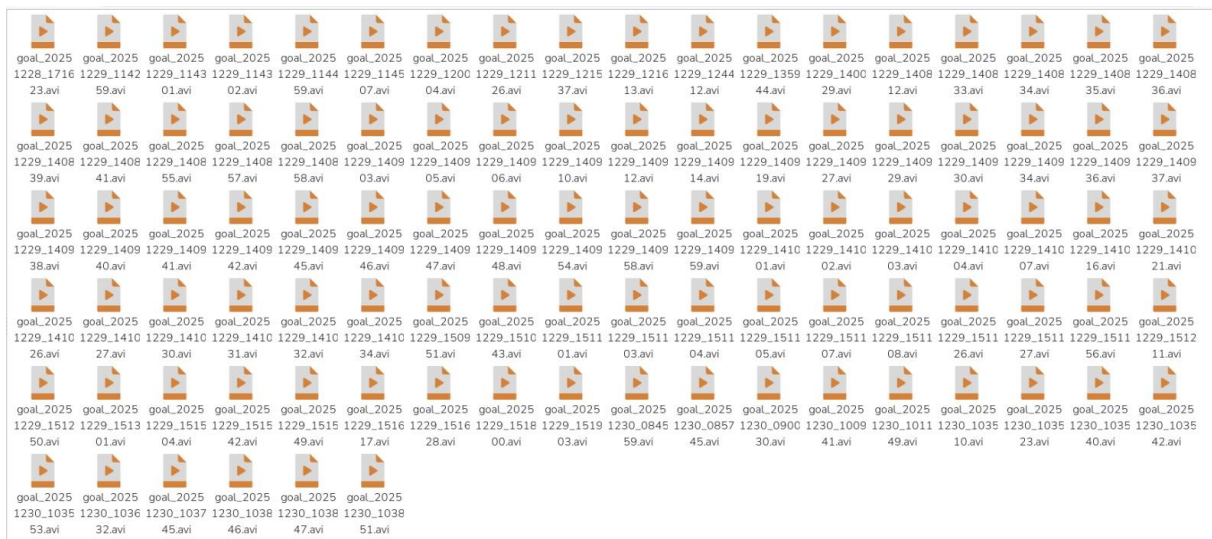


Fig 4 : Historique des vidéos de goals

Les vidéos enregistrées montrent l'historique complet des buts détectés par le système. Chaque fichier contient un enregistrement vidéo du moment exact où un but a été marqué, permettant ainsi une analyse ultérieure.

Cette approche permet une détection automatique et fiable sans intervention humaine.



Fig 5 : Caméra – Détection zone de goal

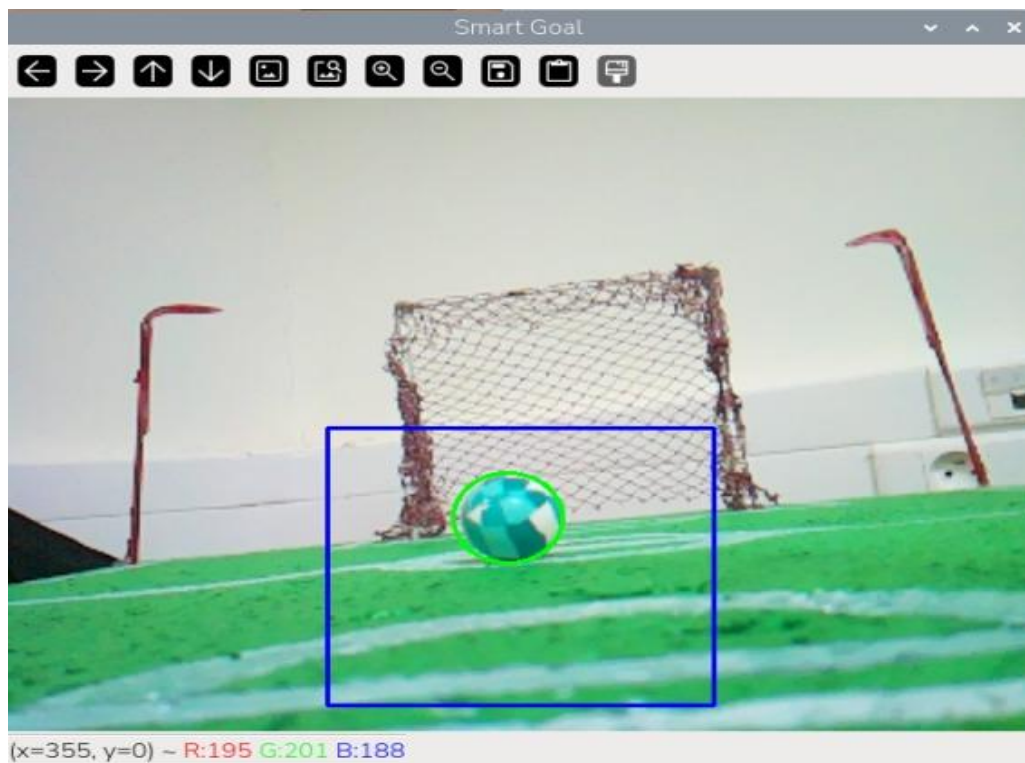


Fig 6 : Détection ballon dans goal

3-2 Buzzer :

Le buzzer est un dispositif de sortie sonore connecté directement au Raspberry Pi à travers les broches GPIO. Son rôle est d'indiquer de manière claire et immédiate qu'un goal a été marqué. Dès que le Raspberry Pi confirme la détection d'un goal, il envoie un signal électrique au buzzer afin de l'activer pendant une durée prédéfinie. La durée et le mode de fonctionnement du buzzer peuvent être contrôlés par logiciel, ce qui permet d'adapter le signal sonore aux besoins du système.

L'utilisation du buzzer permet d'assurer une interaction simple et efficace avec l'utilisateur, en fournissant une notification sonore instantanée.

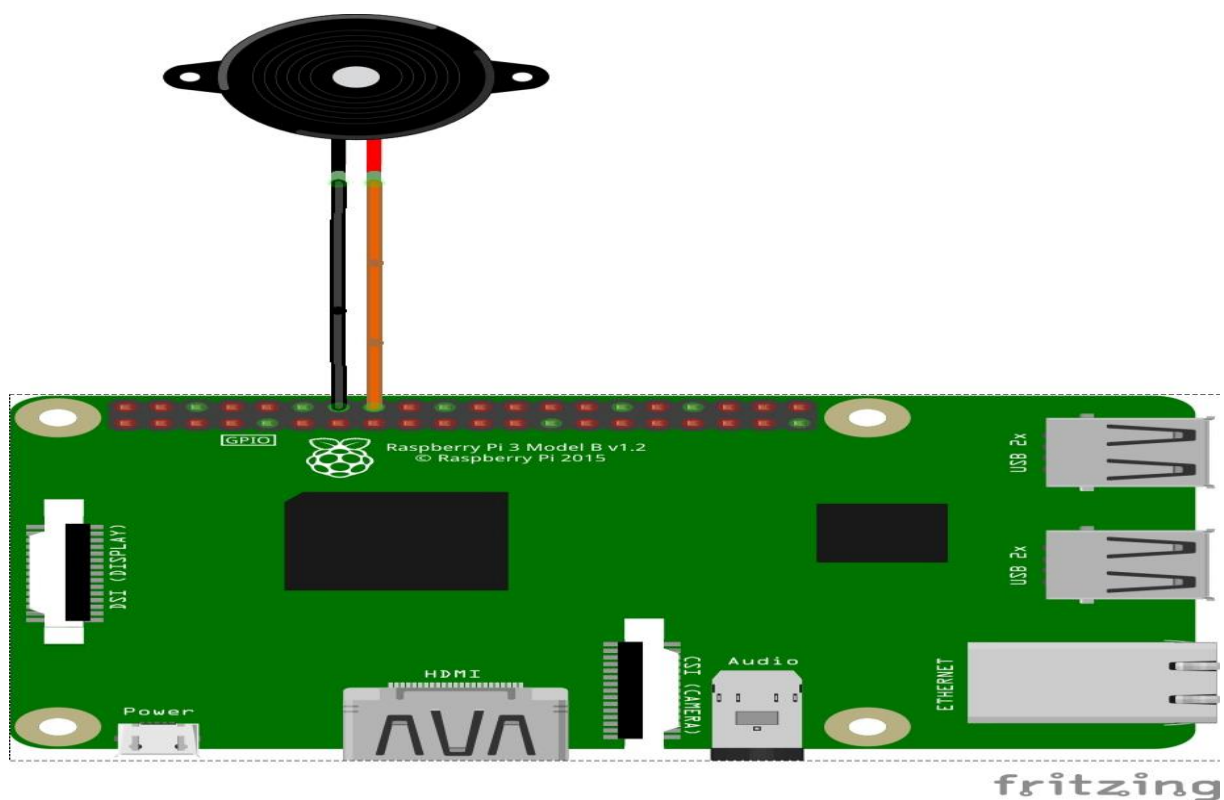


Fig7 :Schéma circuit Raspberry et Buzzer

4. Conclusion :

La partie Raspberry Pi apporte une intelligence significative au projet grâce à l'intégration de la vision par ordinateur et au contrôle des dispositifs électroniques. Elle permet une détection automatique et fiable des goals à partir de l'analyse des images capturées par la caméra.

La signalisation sonore via le buzzer garantit une notification rapide et efficace de l'événement détecté. Cette architecture, volontairement simplifiée par l'absence d'un écran LCD, met en évidence l'efficacité du système embarqué et sa capacité à fonctionner de manière autonome et en temps réel.

Partie IV :

L'interface graphique

1. Introduction :

Dans le but d'améliorer l'interactivité et la convivialité du système de détection automatique de buts, une interface graphique a été mise en place.

Cette interface permet d'afficher le score du match en temps réel, offrant ainsi à l'utilisateur une visualisation claire et instantanée des événements détectés par le système.

Grâce à cette interface, le projet ne se limite plus à une simple détection matérielle (caméra, buzzer), mais devient une solution complète combinant traitement d'image, électronique et affichage visuel.

2. Technologies Utilisées :

L'interface graphique a été développée en utilisant les technologies suivantes :

➤ **Langage de programmation : Python**

Python a été choisi pour sa simplicité, sa compatibilité avec le Raspberry Pi et son intégration facile avec les bibliothèques de traitement d'image et de contrôle matériel.

➤ **Bibliothèque graphique**

Une bibliothèque graphique adaptée à l'affichage en temps réel a été utilisée (telle que Tkinter, PyQt ou Pygame)

```
1 import tkinter as tk
2 from tkinter import Listbox
3 import os, subprocess
4
5 root = tk.Tk()
6 root.title("Smart Goal")
7 root.geometry("400x400")
8
9 score_label = tk.Label(root, text="Score: 0", font=("Arial", 16))
10 score_label.pack(pady=10)
11
12 tk.Label(root, text="Goal videos").pack()
13 lst = Listbox(root, width=45)
14 lst.pack(pady=10)
15
16 score_file = "SmartGoal/score.txt"
17 goals_dir = "SmartGoal/goals"
18
19 def refresh():
20     # Read score
21     if os.path.exists(score_file):
22         with open(score_file, "r") as f:
23             score = f.read().strip()
24             score_label.config(text=f"Score: {score}")
25
26     # Update video list
27     lst.delete(0, tk.END)
28     if os.path.exists(goals_dir):
29         for f in sorted(os.listdir(goals_dir)):
30             if f.endswith(".avi"):
31                 lst.insert(tk.END, f)
32
33     root.after(1000, refresh)
34
35 def play():
36     sel = lst.curselection()
37     if sel:
38         subprocess.Popen(["vlc", f"{goals_dir}/{lst.get(sel)}"])
39
40 tk.Button(root, text="Play video", command=play).pack(pady=10)
41 refresh()
42 root.mainloop()
```

Fig 8: Interface graphique tkinter

```
51         80,
52         param1=100,
53         param2=30,
54         minRadius=15,
55         maxRadius=80
56     )
57
58     # Draw goal zone
59     cv2.rectangle(
60         frame,
61         (GOAL_X1, GOAL_Y1),
62         (GOAL_X2, GOAL_Y2),
63         (255, 0, 0),
64         2
65     )
66
67     if circles is not None:
68         circles = np.uint16(np.around(circles))
69         for x, y, r in circles[0]:
70             cv2.circle(frame, (x, y), r, (0, 255, 0), 2)
71
72         if GOAL_X1 < x < GOAL_X2 and GOAL_Y1 < y < GOAL_Y2 and not goal_detected:
73             score += 1
74             goal_detected = True
75
76             print("GOAL | Score =", score)
77
78             # GPIO alert
79             GPIO.output(LED, 1)
80             GPIO.output(BUZZER, 1)
81
82             # Save video
83             filename = (
84                 "SmartGoal/goals/goal "
85                 + datetime.now().strftime("%Y%m%d_%H%M%S")
86                 + ".avi"
87             )
88
89             video_writer = cv2.VideoWriter(
90                 filename,
91                 cv2.VideoWriter_fourcc("XVID"),
92                 20,
93                 (640, 480)
94             )
95
96             recording = True
97             start = time.time()
98
```

Fig 9 : Configuration et initialisation

Cette bibliothèque permet :

- **La création de fenêtres graphiques**
- **L'affichage dynamique des scores**
- **La mise à jour instantanée de l'interface sans interruption du système**

Rapport de projet match intelligent

```
99         # Save score
100         with open(score_file, "w") as f:
101             f.write(str(score))
102
103     if recording:
104         video_writer.write(frame)
105         if time.time() - start > 3:
106             video_writer.release()
107             recording = False
108             GPIO.output(LED, 0)
109             GPIO.output(BUZZER, 0)
110
111     if circles is None:
112         goal_detected = False
113
114     cv2.imshow("Smart Goal", frame)
115
116     if cv2.waitKey(1) & 0xFF == 27:
117         break
118
119     cv2.destroyAllWindows()
120     GPIO.cleanup()
121     picam2.stop()
```

Fig10 : Sauvegarde du score et nettoyage système

3. Fonctionnement de l'Interface Graphique :

L'interface graphique est exécutée directement sur le Raspberry Pi, en parallèle avec le programme de détection des goals.

Son fonctionnement peut être résumé comme suit :

- ❖ Le système de vision détecte le passage du ballon dans la zone du but.
- ❖ Un signal est envoyé au programme principal indiquant qu'un goal a été marqué.
- ❖ L'interface graphique reçoit cette information.
- ❖ Le score de l'équipe concernée est automatiquement incrémenté.
- ❖ L'affichage est mis à jour en temps réel sur l'écran.

Le score des deux équipes est présenté de manière claire, avec une taille de police lisible et une organisation simple afin de garantir une bonne compréhension par l'utilisateur.

4. Avantages de l'Interface Graphique :

L'intégration d'une interface graphique apporte plusieurs avantages importants au système :

- **Affichage du score en temps réel, sans intervention manuelle.**
- **Interaction améliorée avec l'utilisateur, grâce à une visualisation claire et intuitive.**
- **Meilleure compréhension des événements du match, notamment lors des goals successifs.**
- **Valorisation du projet, en ajoutant une dimension visuelle professionnelle.**
- **Amélioration de l'expérience utilisateur, rendant le système plus attractif et pratique.**

5. Conclusion :

L'interface graphique constitue un élément clé du système de détection automatique des buts. Elle assure le lien entre le Raspberry Pi et l'utilisateur en transformant les données techniques en informations visuelles claires et compréhensibles.

L'affichage du score en temps réel permet un suivi instantané du match sans intervention manuelle, ce qui améliore la fiabilité du système et l'expérience utilisateur.

De plus, cette interface illustre la bonne intégration entre la vision artificielle, l'électronique et le développement logiciel, rendant le projet plus complet et interactif.

En conclusion, l'interface graphique contribue efficacement à la performance globale du système et valorise l'utilisation du Raspberry Pi comme plateforme centrale du projet.

Conclusion

Ce projet de match intelligent basé sur Arduino et Raspberry Pi a permis de concevoir un système automatisé et interactif intégrant plusieurs aspects des systèmes embarqués.

La partie Arduino assure la gestion de l'éclairage automatique du terrain grâce à la détection de mouvement, garantissant ainsi une consommation d'énergie optimisée et un fonctionnement autonome. Le Raspberry Pi, quant à lui, apporte une dimension intelligente au projet : il capture et analyse les images fournies par la caméra pour détecter automatiquement le passage du ballon dans la zone du but. Lorsqu'un goal est détecté, il déclenche une signalisation sonore via le buzzer, permettant à l'utilisateur d'être alerté immédiatement.

L'interface graphique développée en Python complète le système en affichant le score du match en temps réel, offrant ainsi une visualisation claire et instantanée des événements.

En résumé, le projet illustre l'intégration réussie de l'électronique, de la programmation et de la vision par ordinateur pour créer un système fiable, autonome et interactif. Il constitue une application concrète des notions étudiées en systèmes embarqués et peut être amélioré ultérieurement par l'ajout de fonctionnalités avancées telles que la connectivité réseau ou des algorithmes de vision artificielle plus sophistiqués.

Bibliographie

Arduino Documentation officielle : <https://www.arduino.cc>

Raspberry Pi Foundation, Documentation : <https://www.raspberrypi.com>

OpenCV Documentation : <https://opencv.org>

Python Software Fondation, Python Documentation :
<https://www.python.org>

Tkinter Documentation :
<https://docs.python.org/3/library/tkinter.html>

WOKWI Simulator : <https://wokwi.com>

Abstract

This project presents an intelligent football field system based on Arduino and Raspberry Pi. It provides automatic lighting, goal detection using a camera, audio signaling via a buzzer, and real-time score display through a graphical interface.

The system operates autonomously and illustrates a practical application of embedded systems and computer vision technologies.

Résumé

Ce projet présente un smart match system basé sur Arduino et Raspberry Pi. Il permet automatic lighting, goal detection using camera, sound notification via buzzer et real-time score display à travers une graphical interface en Python.

Le système est autonomous, intelligent et améliore l'expérience utilisateur.