

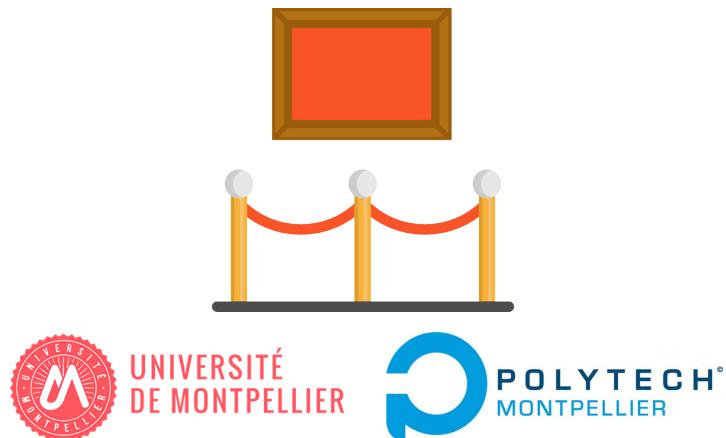
Rapport de projet FAS

**Keep away : un système de sécurité
pour le musée**

FAIZA Mohamed Iheb
BENCHRAA Soufiane

sous la direction de
M. David Delahaye
M. Vincent Berry

December 2016



Sommaire

1	Introduction	3
2	Installation et Fonctionnement	4
2.1	Mise en place du matériel	4
2.2	Installation du logiciel	4
2.3	Fonctionnement	4
3	Choix et Conception	5
3.1	Choix du dispositif	5
3.2	Choix des capteurs	5
3.3	Architecture	6
4	Partie Programmation	6
5	Organisation et Outils de travail	6
5.1	Répartitions des tâches	6
5.2	Outils utilisées	6
6	Perspectives	7

Remerciements :

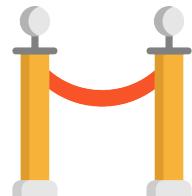
Nous adressons nos remerciements aux personnes qui nous ont aidé dans la réalisation de ce projet. En premier lieu, nous remercions M. David Delahaye et M. Vincent Berry en tant que tuteurs de projet, ils nous ont guidé dans notre travail et nous ont aidé à trouver des solutions pour avancer. Nous remercions aussi l'université de Montpellier et Polytech Montpellier pour l'environnement de travail qu'elles proposent et qui nous a permis de mener à bien ce projet.

1 Introduction

Dans le cadre d'un appel à projets lancé par le musée Fabre à Montpellier pour une soirée sous le nom "François-Xavier n'est pas couché", nous avons eu à réaliser un projet qui a pour but de "Révolutionnez le musée" et qui doit être liée à notre domaine d'études.

Pour cela, nos enseignants M. Berry et M. Delahaye nous ont proposé de concevoir un dispositif basé sur le nano-ordinateur Raspberry Pi ou sur le microcontrôleur Arduino tout en nous laissant le choix et l'imagination des fonctionnalités et des objectifs de ce dispositif.

Dans un musée, la distance minimale entre les œuvres et les visiteurs est souvent déterminée par une barrière, de sorte que les visiteurs du musée ne peuvent pas la dépasser. Malgré cela, les œuvres risquent d'être abîmées par des personnes qui ne respectent pas cette distance. Pour cela, nous avons eu l'idée de bien protéger et sécuriser les œuvres en créant un système de sécurité, capable de détecter les personnes se trouvant dans une zone qui met l'œuvre en danger, et d'avertir les agents de sécurité du musée.



2 Installation et Fonctionnement

2.1 Mise en place du matériel

Tout d'abord, il faut vérifier le branchement des capteurs au Raspberry : le capteur ultrason doit être branché sur le pin D4 du grove shield, la camera sur le port CSI et les hauts parleurs sur le port jack, puis, brancher le Raspberry au réseau via un câble Ethernet et le brancher à l'électricité avec l'adaptateur. Il est possible aussi de connecter le Raspberry pi au réseau via WiFi au lieu de Ethernet. Ensuite, il faut positionner le dispositif devant l'oeuvre et l'orienter de sorte que le capteur ultrason et la caméra puissent couvrir la zone en face de l'oeuvre.

2.2 Installation du logiciel

Pour l'installation, il faut installer le package i2c-tools sur le système, ensuite télécharger les fichiers du projet sur le Raspberry, la manière la plus rapide de faire ça c'est avec la commande :

```
git clone https://github.com/medihebfaiza/museum_security.git
```

Enfin, il faut exécuter le programme C qui est déjà compilé. Par ailleurs, ce programme dépend de la bibliothèque grovepi et d'un script python inclus dans le projet donc il ne faut pas les effacer ou même les déplacer.

2.3 Fonctionnement

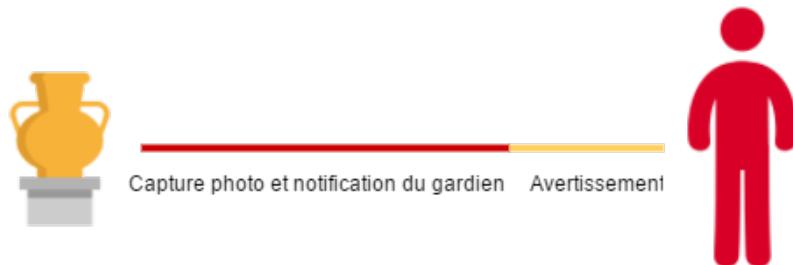


Figure 1: Distance minimale respectée



Figure 2: Distance minimale dépassé

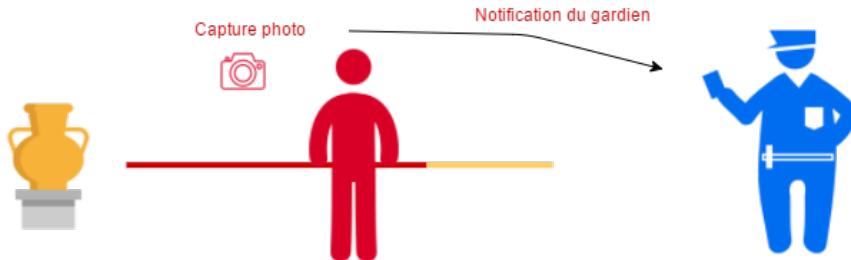


Figure 3: Notification du gardien

Nous avons désigner une distance minimale à respecter par rapport à l'oeuvre, une fois cette distance n'est pas respectée, le matériel lance un avertissement, si l'individu continue à s'approcher de l'oeuvre, une alarme se déclenche, avec une prise de photo de l'individu et une envoie de notification à l'agent de sécurité (ceci se passe en quelques secondes).

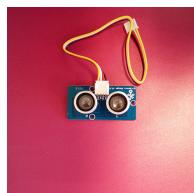
3 Choix et Conception

3.1 Choix du dispositif

Pour notre projet, nous avons choisi Raspberry pi, pour la facilité qu'il nous offre, car le raspberry pi est un véritable ordinateur sur lequel on peut installer un système d'exploitation, écrire des programmes dans plusieurs langages et même installer un serveur web. On peut aussi brancher des périphériques et un câble réseau.

3.2 Choix des capteurs

Nous avons choisi nos capteurs en se basant sur plusieurs critères : compatibilité avec le Raspberry, les besoins de notre projet, prix, disponibilité, etc.



Capteur Ultrason Groove



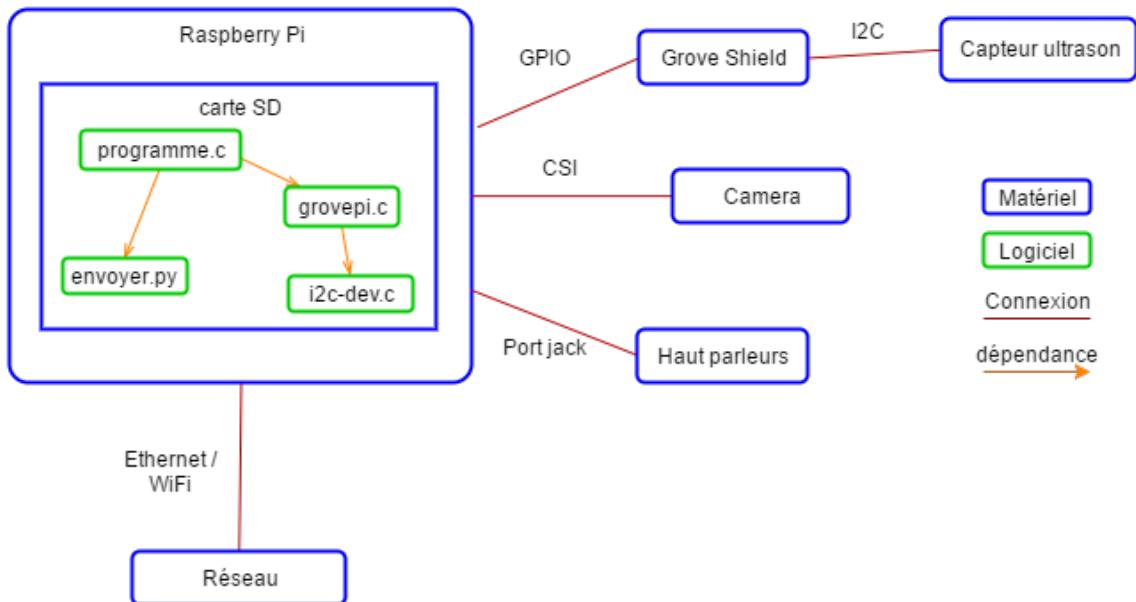
Camera nocturne PI-NoIR



Haut Parleurs

- **Capteur Ultrason Groove :** permet de mesurer la distance sans contact à l'aide de transducteurs à ultrasons. Ce capteur a une portée de 4 mètres ce qui est idéal pour notre projet.
- **PI-NoIR :** Une caméra sans filtre infrarouge, idéal pour la vision nocturne avec un grand angle de couverture de 105, une connexion CSI, et un prix très raisonnable (30 euros).
- **Hauts parleurs :** simples, faciles à monter et à un prix raisonnable. Alimenté avec le cable jack et (pas besoin de batteries). Le seul inconvénient de ces hauts parleurs est que leurs volume est bas.

3.3 Architecture



4 Partie Programmation

D'abord, notre programme doit être capable de communiquer avec le capteur ultrason afin de récupérer la distance et la vérifier. Pour cela, nous utilisons la bibliothèque grovepi que nous avons réussi à implémenter pendant les séances de TP et qui contient toutes les fonctions de haut niveau nécessaire pour lire et écrire des données via le bus I2C. Donc, nous avons inclus la bibliothèque dans le programme principale.

Après les initialisations, le programme entre dans une boucle infinie dans laquelle à chaque 0.7 secondes, il récupère la distance calculé par le capteur ultrason et la vérifie. Si la distance est supérieur à 1.5 mètres il continue à boucler, si c'est entre 1 mètre et 1.5 mètres il lance un appel système pour jouer le message d'avertissement. Finalement, si la distance est inférieure à 1 mètre, le programme lance un premier appel système pour faire la capture photo et un second appel pour exécuter le script python qui s'occupe de l'envoi de la photo prise par e-mail.

5 Organisation et Outils de travail

5.1 Répartitions des tâches

- **Tâches communes :** conception du dispositif, choix des capteurs, tests et optimisations .
- **Mohammed :** programmation du programme principal et des appels système.
- **Soufiane :** programmation du script d'envoi.

5.2 Outils utilisées

- **Github** pour la gestion des versions du code.

- **Google Drive** pour l'écriture du rapport et le partage des différentes documentations.
- **ShareLatex** pour l'écriture de la version finale du rapport.
- **Atom** un éditeur de texte avec l'intégration du git.

6 Perspectives

Notre collaboration régulière nous a permis de tous nous intéresser aux différents aspects de ce projet et de le mener à bien. Ce fut donc l'occasion pour nous d'approfondir nos connaissances dans le domaine technologique et de les mettre en pratique.

Notre projet est modeste, mais il est basé sur un concept qui préoccupe le monde entier, c'est la sécurité. Dans le marché de la sécurité, il y a une très grande concurrence par des entreprises spécialisées. Pour commercialiser le nôtre, il faut rajouter quelques améliorations (plus de capteurs, caméras plus développée), ensuite viser les clients à low budget, et essayer de les convaincre de posséder un matériel de sécurité (simple, pas cher, et surtout sécurisant).

Nous gardons un ressenti très positif à l'égard de cette expérience et avons pleinement de conscience de l'apport qu'elle représente pour nous, tant au niveau académique (connaissances, mise en oeuvre) que professionnelle (première expérience très valorisante). Nous tenons également à souligné que l'aspect ludique d'un tel projet aide largement les étudiants à en tirer profit.