

PROJET FAS



22/01/2017

Mehdi Delvaux – Yves Alain Agbodjogbe

Compte rendu du projet FAS semestre 1
IG3 Polytech Montpellier

Projet FAS

MEHDI DELVAUX – YVES ALAIN AGBODJOGBE

Table des matières

I) INTRODUCTION.....	2
II) MODE D'EMPLOI.....	2
a) Partie Matérielle	2
b) Partie Logicielle	3
III) MOYENS MATERIELS	3
IV) MOYENS HUMAINS.....	4
V) ARCHITECTURE DU PROJET.....	5
VI) CODES	5
VII) PERSPECTIVES	6
VIII) CONCLUSION.....	6

I) INTRODUCTION

Pour garantir un niveau sonore adéquat au Musée, nous avons décidé de nous pencher sur la mise en place d'une ambiance lumineuse qui s'adapte au niveau sonore de la pièce et au nombre de personnes dans cette même pièce. Notre mission fût de récupérer ces différentes informations (niveau sonore et nombre de personnes) pour créer une ambiance lumineuse qui s'accorderait parfaitement à la situation dans laquelle nous nous trouvons. Cela à terme permettrait au musée de vérifier et contrôler le bruit dans les pièces du musée pour assurer un confort "auditifs" aux amateurs de peinture et signaler les pièces bondées. Nous avons décidé d'appeler ce projet : "Le Musée en Lumière".

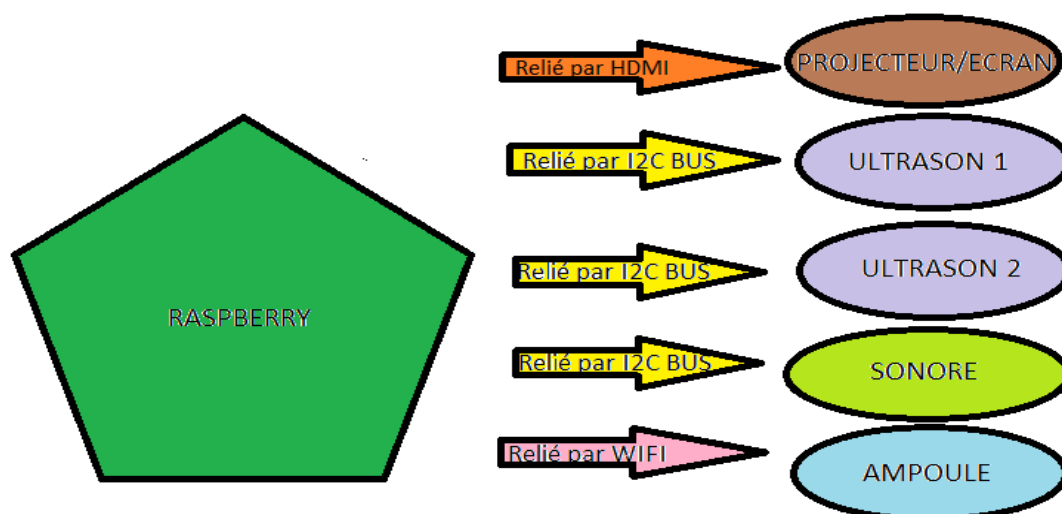
Dans ce rapport nous allons vous donner un mode d'emploi pour que tout le monde puisse le mettre en œuvre, les moyens matériels, les moyens humains, l'architecture de notre projet, les différents codes écrits et enfin les perspectives de notre projet.

II) MODE D'EMPLOI

Dans cette partie nous allons décrire avec précision les différents branchements matériels à effectuer et les packages ou plutôt bibliothèques à installer pour réaliser ce projet.

Nous allons commencer par la partie matérielle qui explicitera les différents branchements ensuite nous passerons à la partie logicielle.

a) Partie Matérielle



Les capteurs ultrasons et le capteur sonore sont reliés au raspberry grâce au shield grove via les bus i2c. Vérifiez que les ultrasons sont reliés aux ports digitaux D2 et D3, le capteur sonore quant à lui doit être branché sur un port analogique A1. Quant à l'ampoule, elle est reliée au raspberry via le wifi. Il est donc nécessaire de connaître son adresse IP et qu'elle soit mise au préalable en "mode développeur" ce qui permet à n'importe quel outil informatique de l'utiliser via la connexion. Le dispositif de lancement du programme (pour nous le Raspberry) et l'ampoule doivent être connectés au même réseau Wifi. L'ampoule se connecte au réseau souhaité grâce à une application mobile appelé "Yeelight" disponible sur tout OS de smartphone (sauf windows phone). C'est aussi grâce à cette application que l'ampoule peut être mise en "mode développeur".

Le projecteur ou l'écran communique grâce à l'entrée HDMI du Raspberry. Si vous êtes connecté de manière distante au Raspberry (c-à-d via ssh) il faut absolument exécuter cette commande dans le shell : `export DISPLAY=:0` car le Raspberry ne sait pas sur quel écran l'afficher. Si vous êtes directement sur le Raspberry, ce n'est pas nécessaire de faire l'export.

b) Partie Logicielle

Nous avons utilisé différents packages pour arriver à réaliser ce projet. Il faut absolument les installer avant de commencer toutes mise en œuvre.

- ☐ **Eye of Gnome (eog --version 3.14.1)** : Nous a servi pour l'affichage des images. C'est une visionneuse d'image qui n'était pas installé sur le Raspberry par défaut.
- ☐ **Yeelight** : C'est la bibliothèque python qui nous permet d'envoyer des instructions à l'ampoule RGB. En installant l'utilitaire "Yeecli" dont le lien est donné plus bas, la bibliothèque s'installe automatiquement.
- ☐ **Yeecli (yee --version 0.0.16)** : C'est un utilitaire qui permet de faire appel aux fonctions définies dans la bibliothèque Yeelight. Il est trouvable à cette adresse : <https://github.com/skorokithakis/yeecli>

III) MOYENS MATERIELS

Nous allons maintenant lister tous les capteurs nécessaires à l'élaboration du projet. Nous donnerons par la même leurs fonctions.

- ☐ **Raspberry Pi 3** : Nous avons choisis le raspberry pi pour mettre en œuvre notre projet. Il offre beaucoup plus de possibilités qu'un arduino notamment en termes d'affichage d'image et de communication avec d'autres appareils.
- ☐ **Shield Grove** : Nous a permis d'utiliser les différents capteurs avec le Raspberry

- ☐ **Capteurs ultrasons** : Utilisé pour le comptage des personnes entrantes et sortantes de la pièce. Nous en avons utilisé 2. Le sens dans lequel les personnes passent est déterminé par les valeurs que prennent les capteurs à un instant T.
- ☐ **Capteur sonore** : Utilisé pour recueillir le niveau sonore de la pièce.
- ☐ **Projecteur** : Utilisé pour la projection des images relatives au niveau sonore.
- ☐ **Ampoule Xiaomi Yeelight RGB** : Transcrit les informations relatives au comptage de personnes en “lumière” selon notre propre algorithme.

IV) MOYENS HUMAINS

Qui a fait quoi ? Notre équipe se compose de Yves-Alain Agbodjogbe et Mehdi Delvaux. Nous nous sommes répartis le travail afin d’être le plus efficace et le plus rapide possible.

Agbodjogbe Yves-alain :

Gestion de la partie sonore et affichage par projection d’images :

- ☐ Installation des packages de visualisation d’images sur écran / projecteur
- ☐ Mise en place d’un programme de contrôle sonore et évolution de la couleur en fonction du niveau capté.

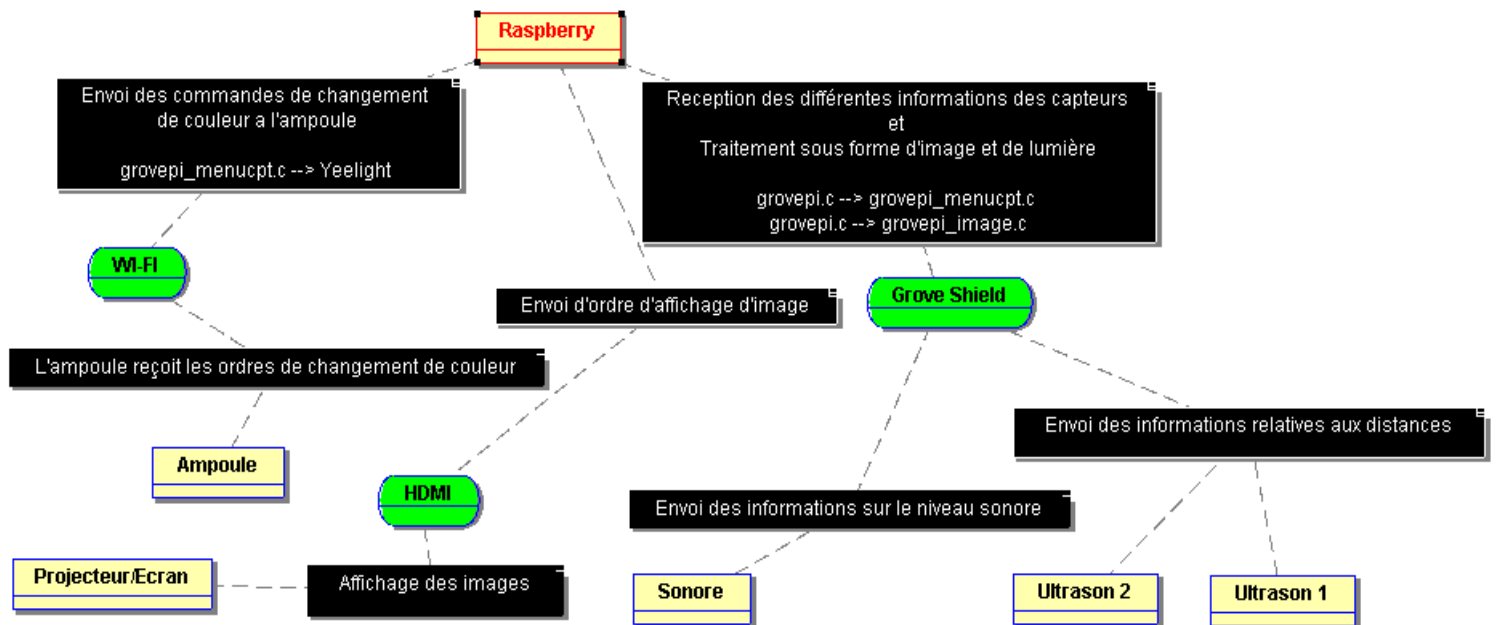
Delvaux Mehdi :

Gestion de la partie comptage par Ultrason et ampoule RGB :

- ☐ Installation des packages de contrôles d’ampoules via raspberry Pi
- ☐ Mise en place d’un programme de comptage de personne via ultrason et adaptation progressive de la couleur des ampoules en fonction du nombre de personne dans la salle.

Nous avons aussi pu profiter de l’expérience de nos professeurs et également de différents travaux réalisés sur internet. Notamment celui de Stavros Korokithakis qui a créé l’utilitaire “Yeecli” ainsi que la bibliothèque Yeelight.

V) ARCHITECTURE DU PROJET



VI) CODES

Gestion sonore et affichage via projecteur :

void `change_image` : int => void ; Affichage via projecteur de l'image correspondante à l'entier entré en paramètre (le nom des images est un entier).

int `definirCouleur` : int => int ;Défini la bonne couleur à afficher en fonction de la valeur lu par le capteur sonore. C'est l'entier renvoyé par cette fonction qui va être utilisé dans la fonction `change_image`(int commande).

Gestion du comptage via capteur Ultrason et allumage de l'ampoule RGB :

void `change_color` : int x int x char => void; Fonction qui fait appel à l'utilitaire Yeecli et qui permet de faire changer de couleur notre ampoule RGB. Nécessite l'ip de l'ampoule en paramètre, l'entier correspondant à la couleur actuelle de l'ampoule et la commande qui est

- 1 : On incrémente la couleur c'est à dire évolution du rouge >>>>> au vert
- 0 : On décrément la couleur c'est à dire évolution du vert >>>>> au rouge



De plus, nous utilisons les fonctions fournies dans le fichier GROVEPI.C pour effectuer une lecture des capteurs ultrasons ou du capteur sonore. Voir code sur GitHub

VII) PERSPECTIVES

Nous avons réussi à atteindre les objectifs de départ de notre projet c'est à dire adapter l'ambiance lumineuse en fonction du son et du nombre de personnes. Avec plus de moyens matériels (capteurs) et plus de temps, nous voulions mettre en place une application web qui aurait permis de connaître les informations relatives à la salle équipée par nos capteurs, c'est à dire que la personne en charge de la surveillance du musée puisse connaître en temps réel le nombre de personnes dans la pièce sans pour autant être présent dans celle-ci. L'application pourrait ainsi recevoir toute les données de nos capteurs et indiquer à la personne concernée d'intervenir, par exemple dans le cas où une pièce est surchargée ou trop bruyante.

De plus le couplage de ce projet avec celui qui effectue le comptage de personnes grâce à une caméra rendrait ce calcul plus précis (plusieurs personnes peuvent entrer simultanément avec la caméra, tandis qu'il est impossible de détecter ce genre de chose avec des ultrasons).

Commercialisation : Notre projet pourrait intéresser entre autres : les musées, les bibliothèques etc... les endroits où le niveau sonore doit être maîtrisé. Notre projet peut donc être commercialisé à faible coût puisqu'il nécessite seulement quelques capteurs, un Raspberry, une lampe RGB et un projecteur pour un total de 250€.

VIII) CONCLUSION

Pour conclure, nous avons réussi à respecter le cahier des charges fixés au début du projet. Nous avons appris à contrôler des objets connectés via le Raspberry.

Également ce projet a développé nos capacités d'auto-formation et nos connaissances sur le Raspberry. Nous avons ainsi pu constater que le Raspberry a une incroyable force d'adaptativité et est très pratique peu importe le projet que l'on veut accomplir notamment dans le domaine de la domotique. Nous remercions nos professeurs pour ce programme et les connaissances techniques qu'ils nous ont apporté durant toute la durée de ce projet.