

# *Projet Magic Mirror*

Paul CONTREMOULIN

Marion RUL

IG3 – Polytech – Année 2016-2017

## I) Introduction

Notre projet Magic mirror est un miroir qui ne vous dit pas seulement que vous êtes le plus beau de tous, il vous affiche aussi l'heure, la date, le temps qu'il fait à l'extérieur, les prochaines dates importantes de votre agenda et, grande nouveauté, il vous dit combien de personnes sont présentes chez vous grâce à ses deux capteurs intégrés. Le principe est de coller un film miroir sans tain sur un écran relié à un Raspberry pi, pour que l'affichage de l'ordinateur se fasse mais que l'on puisse aussi se voir. Vous trouverez dans ce rapport de synthèse tout le manuel de construction de notre projet.

## II) Le projet

### 1) Mode d'emploi

#### **Au niveau des branchements**

Tout d'abord il vous faut un écran. Pour cela nous avons utilisé un écran d'ordinateur portable préalablement démonté. Le problème c'est que l'écran portable utilise une connectique peu courante, le LVDS et nous avons donc dû trouver et acheter un adaptateur HDMI vers LVDS. Il suffit alors de brancher l'adaptateur à l'écran puis à un câble hdmi et le câble hdmi au Raspberry Pi pour faire le lien entre les deux.

Il vous faut ensuite deux capteurs ultrason, que vous relierez au Raspberry Pi grâce à un Shield, une carte que vous branchez sur votre Raspberry pour étendre ses capacités.

#### **Au niveau de l'installation de l'application**

Michael Teuw's a été le premier à construire ce projet. Il a documenté tout le processus de création de cette application et a rendu son document open source et modulable. Nous avons donc utilisé son code puis nous l'avons amélioré de façon à afficher seulement ce qui nous intéressait et à y rajouter notre code.

La première étape consiste donc à installer l'application de Michael Teuw's. Pour cela il vous faut :

- Installer la dernière version de Node.js
- Cloner le répertoire « git clone <https://github.com/MichMich/MagicMirror> »
- Se diriger vers le répertoire « cd MagicMirror »
- Installer et lancer l'application « npm install && npm start »

Une fois l'application bien installée sur votre Raspberry Pi et avant de la lancer, vous pouvez modifier l'affichage qui est sur votre écran ou alors rajouter des modules complémentaires, nous avons donc rajouté le module qui affiche le nombre de personnes présentes. Pour cela il faut créer un nouveau dossier dans « MagicMirror/modules/default » ayant le nom du module et qui contiendra le module développé.

Ce dossier est constitué d'un fichier JavaScript, utilisé par l'application pour construire et afficher le module à l'écran, d'un fichier CSS pour le style du module, et d'un fichier text contenant le nombre

de personnes dans la pièce. Ce fichier text sera modifié en fonction de notre code, nous l'expliquerons mieux dans la cinquième partie de ce rapport.

Afin de référencer votre module sur le miroir, il vous faut aller dans le répertoire MagicMirror puis le répertoire config. Dans ce répertoire vous allez modifier le fichier config.js pour y rajouter le module que vous désirez, ici nbpersonnes, ainsi que la position que vous voulez qu'il ait sur l'écran, nous avons choisi bottom\_left. Vous devez ensuite vous rendre dans le répertoire modules puis le répertoire default où vous avez créé votre nouveau dossier et modifier le fichier defaultmodules.js et ajouter votre module dans la tableau des modules.

## 2) Moyens matériels

Pour effectuer notre projet nous avons eu besoin :

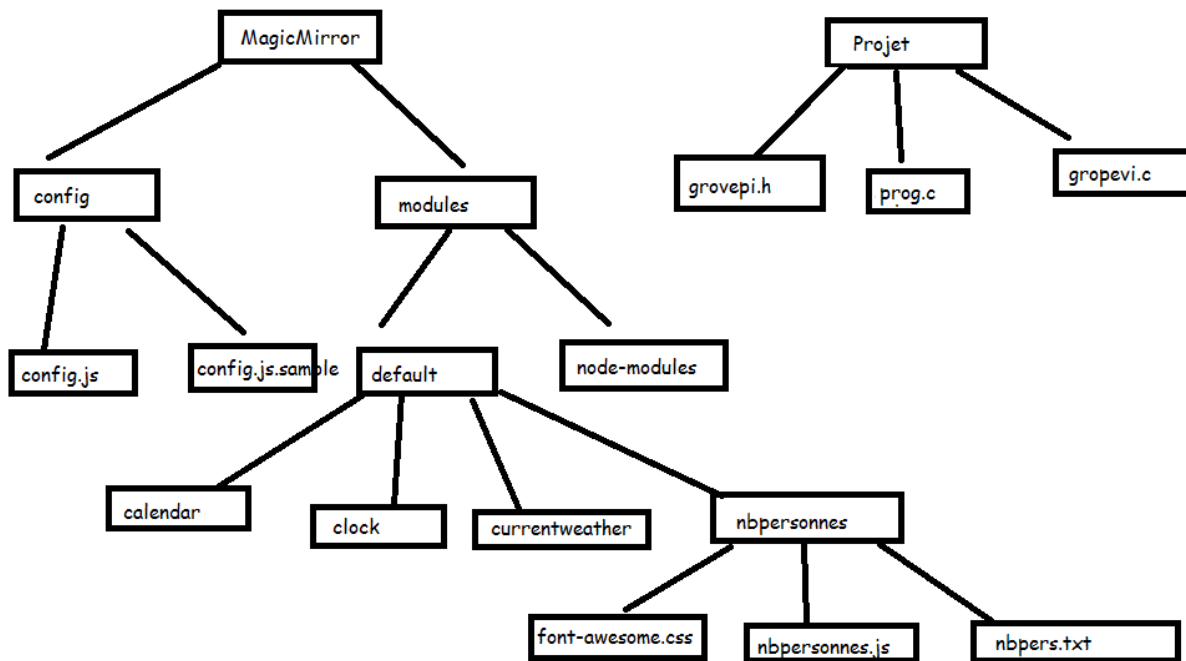
- Un Raspberry Pi, nous devons faire le projet sur Raspberry car nous avons besoin de connecter un écran et nous avons besoin d'avoir le système d'exploitation Linux pour pouvoir installer l'application.
- Un adaptateur HDMI via LVDS pour pouvoir connecter notre écran au Raspberry.
- Un câble HDMI pour relier le Raspberry et l'adaptateur.
- Un film miroir sans tain, pour pouvoir le coller sur l'écran et avoir l'effet miroir en pouvant en même temps voir les informations renvoyées par l'écran.
- Deux capteurs ultrason. Nous avons choisi ces capteurs car nous avons besoin de détecter les passages des personnes se faisant devant le miroir pour ensuite pouvoir comptabiliser le nombre de personnes présentes dans la maison.

## 3) Moyens humains

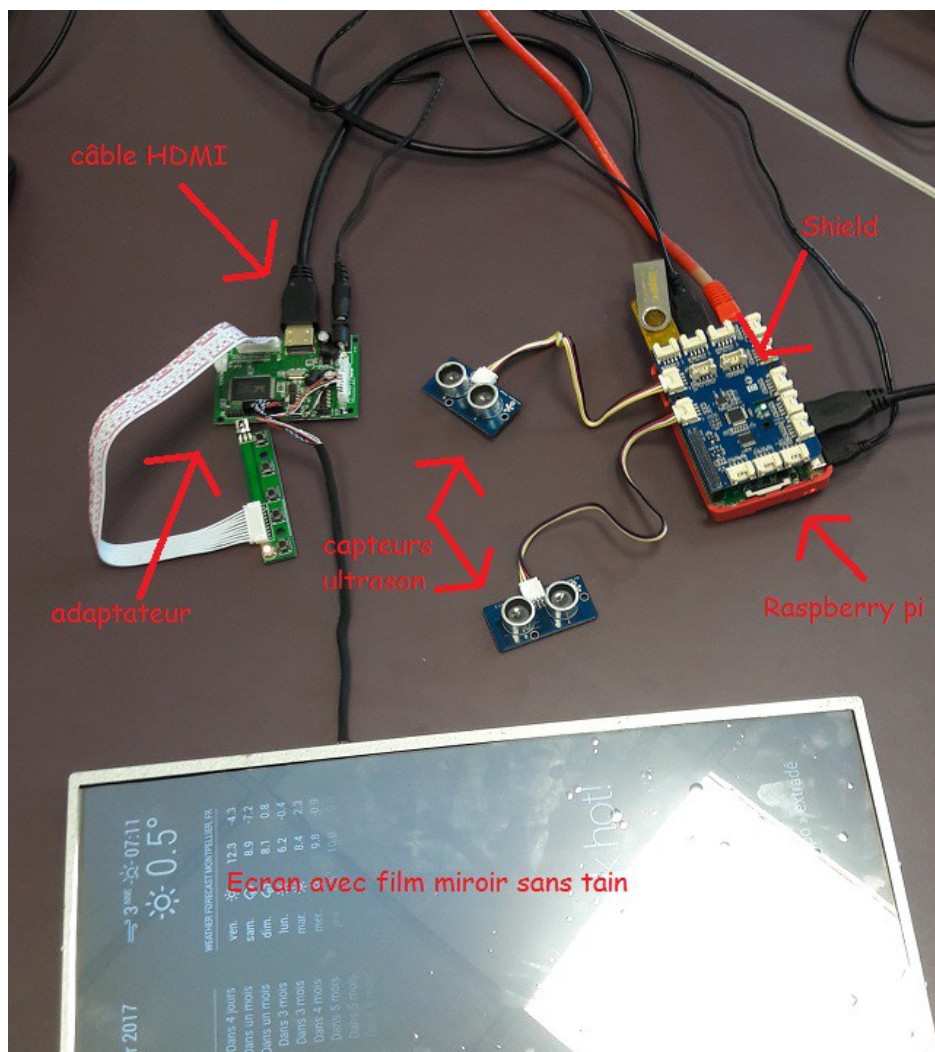
Concernant l'organisation du groupe nous avons effectué le travail ensemble pour la plus grande partie. Nous nous réunissions à Polytech car nous avions tout le matériel nécessaire. Nous avons alors effectué ensemble l'installation de l'application et la création et le test du code concernant le nombre de personnes présentes dans la maison. Ensuite, vers la fin du projet, Paul s'est occupé de rajouter le module concernant le nombre de personnes dans l'application et Marion a rédigé le rapport de synthèse.

## 4) Architecture du projet

Au niveau de l'architecture des fichiers nous avons deux dossiers, le dossier MagicMirror contenant l'application et le dossier Projet contenant les codes de notre programme sur les capteurs. Voici le schéma représentant l'arborescence, pour l'application nous avons seulement présenté les dossiers les plus importants, ceux que nous avons utilisés et modifiés.



Au niveau de l'architecture matérielle voici les différents composants et leurs branchements :



## 5) Code

Concernant le code de notre programme. Nous avons d'abord utilisé les fichiers **grovepi.c** et **grovepi.h** disponibles sur l'ENT. Ces fichiers sont des bibliothèques qui permettent d'avoir les fonctions de base pour utiliser GrovePi en langage C, c'est ce qui connecte le Raspberry aux capteurs. Nous avons ensuite créé notre propre fichier qui permet de connaître le nombre de personnes présentes dans la maison grâce à nos deux capteurs ultrason. Pour cela, nous avons utilisé le fichier grovepi-ultrasonic dans notre fichier principal **prog.c**. La fonction **int capteur(int PIN)** contient alors le code du fichier grovepi-ultrasonic et permet d'identifier le capteur ultrason et de récupérer la distance qu'il détecte. Ensuite, il y a une fonction **int estModifie(int pin)** qui, avec la fonction capteur, renvoie 1 si la distance du capteur est modifiée, 0 sinon. On a également une fonction **int modifValeur(int val)**, c'est cette fonction qui va afficher le nombre de personnes modifié. Elle regarde dans le fichier nbpers du module nbpersonnes de l'application MagicMirror la valeur qui y est présente et y ajoute une valeur, qui sera égale à -1 (si quelqu'un sort) ou 1 (si quelqu'un rentre). Cette valeur est estimée dans notre dernière fonction **main**. Cette fonction regarde si les deux capteurs sont modifiés et les compare, si le capteur 1 est modifié puis le capteur 2 ensuite alors on appelle la fonction modifValeur avec la valeur 1. Sinon, on l'appelle avec la valeur -1. Ce fichier prog.c se compile avec le fichier grovepi.c.

## 6) Perspectives

Si nous avions plus de temps, nous aurions aimé ajouter une caméra au miroir qui, à chaque fois que quelqu'un entre dans la maison, prenne une photo et l'envoie au propriétaire sur son téléphone. Cela aurait permis de bien voir à chaque fois qui entre pour vérifier que la personne n'est pas un imposteur. Cependant nous avons réussi à aller jusqu'au bout de nos objectifs principaux. Nous avons rencontré des problèmes au niveau de l'installation de l'application, comme expliqué dans la partie mode d'emploi, nous avons alors pris beaucoup plus de temps que prévu. Au niveau de la commercialisation du projet, ce projet n'a jamais encore été commercialisé. En effet, il doit être effectué, mis à jour et lancé sur un Raspberry, il concerne plutôt les personnes qui en possèdent un et qui veulent s'amuser à créer leur miroir magique.