



---

# MIROIR MAGIQUE

---



NDIAGNA SECK

Encadrants : Aomar OUSMANI, Massinissa HAMIBI  
20 Janvier 2019

I. Table des matières

II. Introduction.....	2
III. Cahier de charges .....	3
IV. Liste des composants .....	4
V. Aspect logiciels .....	5
A. Installation et configuration du raspberryPi .....	5
➤ Incliner l'écran .....	5
➤ Full screen.....	6
➤ Timezone .....	6
➤ Désactiver l'écran de veille.....	7
B. Installation et configuration de Magic Mirror.....	8
VI. Conclusion .....	12
VII. Problèmes rencontrés .....	13

## II. Introduction

Les miroirs sont passifs et ennuyants. On les regarde tous les jours et on y voit toujours la même chose : une version de soi quasiment identique mais aussi toujours plus vieille. Ce temps passé à se regarder est du temps perdu mais nous sommes pourtant bien obligés de passer par la salle de bain et le miroir. Et si vous pouviez, juste à la sortie de votre douche, savoir quoi porter aujourd'hui grâce à une prévision météo ? Et si vous pouviez, tout en vous coiffant, être au courant de ce qui se passe dans le monde ? Et si vous pouviez, en vous brossant les dents, planifier votre journée de rendez-vous ? C'est ce que je vous propose avec mon prototype de miroir intelligent !

Le sujet de mon projet de Miroir Intelligent m'a immédiatement plu pour son aspect high tech et pourtant bien ancré dans le présent. Mon projet s'est articulé en deux parties. Premièrement je vais réaliser l'interface web qui sert d'interface utilisateur. Deuxièmement je vais construire le cadre et configurer la partie matérielle.

### III. Cahier de charges

L'idée est donc d'avoir un miroir qui fournit tout un tas de services, le plus intelligemment possible

- Dispose d'une interface utilisateur
- Reconnaissance de l'utilisateur
- Commande vocal
- Affichage de l'heure, la date, le météo
- Détection de présence via la caméra du Raspberry Pi et du visage

Après recherche et documentation, la solution la plus simple est de tourner le serveur par Raspberry Pi et le logiciel MagicMirror.

## IV. Liste des composants

Composants	Illustration	Description	Prix
<b>raspberryPi</b>		Cœur du projet : «petit ordinateur»	40 euros
<b>Ecran dalle</b>		Ecran dalle pour afficher les configuration de raspberryPi	35 euros avec sa carte mère
<b>Carte mère pour la dalle</b>		Pour faire marcher l'écran	
<b>Clavier et souris</b>		Pour écrire des scripts	Un clavier fournit par la fac et ma souris sans fil
<b>Alimentation pour écran</b>		Pour alimenter l'écran	15 euros
<b>Miroir sans tain</b>		Le miroir lui même	16 euros
<b>Carte micro SD</b>		Pout télécharger les logiciels neccesaires	9 euros
<b>Caméra web</b>		Pour recuperer les photos	Une vielle caméra d'un ami
<b>Capteur PIR</b>		Capte un mouvement à proximité	6 euros
<b>Câble HDMI vres HDMI</b>		Pour lier le raspberrypi et la carte mère	
<b>Micro USB</b>		Pour la reconnaissance vocal	9 euros
<b>Haut parleur</b>		Pour la sortie vocal	9 euros

## V. Aspect logiciels

### A. Installation et configuration du raspberryPi

Voici le lien que j'ai téléchargé la dernière version de raspbian

<https://www.raspberrypi.org/downloads/raspbian/>

L'archive est décompressée par unzip

J'ai téléchargé Etcher :

Etcher est un logiciel qui va permettre d'installer Raspbian (ou n'importe quel système d'exploitation) sur la carte SD et de le rendre directement bootable.

Il existe d'autres logiciels mais Etcher a l'avantage d'être compatible avec Mac, Windows, Linux et est recommandé par la fondation Raspberry Pi.

Je l'ai téléchargé [ici](#)

Grace a Etcher j'ai flashé la carte SD avec Raspbian

#### ➤ Désactiver le curseur de la souris

J'ai ajouté un petit programme qui va se charger de cacher le curseur de la souris automatiquement :

```
sudo apt-get install unclutter
```

#### ➤ Incliner l'écran

Pour incliner l'écran, éditez le fichier `/boot/config.txt` j'ai ajouté le flag `display_rotate` selon la syntaxe suivante :

```
display_rotate=1 pour une rotation 90 de degrés
```

### ➤ Full screen

Pour être en plein écran, j'ai édité le fichier `/boot/config.txt` :

```
disable_overscan=1
```

### ➤ Timezone

Pour basculer sur l'heure Française

J'ai lancé le gestionnaire de configuration Raspbian par:

```
sudo raspi-config
```

Dans l'onglet "Timezone", j'ai sélectionné "Europe", puis "Paris" et le tour est joué

Par ailleurs, être à la bonne heure est aussi plutôt appréciable.

Pour cela j'utilise le protocole **NTP** (Network Time Protocol). C'est un protocole qui permet à un ordinateur de synchroniser son horloge sur un autre ordinateur avec une précision plus élevée via internet.

Installation du paquet qui va bien :

```
sudo apt-get install ntp
```

Lancement du daemon :

```
sudo systemctl start ntp
```

```
sudo systemctl status ntp
```

J'ai édité le fichier `/etc/ntp.conf` et on modifie les lignes « server » pour se synchroniser sûr les serveurs français :

```
server 0.fr.pool.ntp.org iburst dynamic
server 1.fr.pool.ntp.org iburst dynamic
server 2.fr.pool.ntp.org iburst dynamic
```

## Miroir magique

`server 3.fr.pool.ntp.org iburst dynamic`

Je redémarre le service pour prendre en compte les changements :

`sudo systemctl restart ntp`

J'ai vérifié que l'heure est correcte avec la commande suivante :

`Date`

### ➤ Désactiver l'écran de veille

Je souhaite voir uniquement l'affichage du Magic Mirror sans interruption. Par défaut Raspbian possède un écran de veille qui se lance automatiquement après une période d'inactivité.

J'ai édité `/etc/xdg/lxsession/LXDE/autostart` et `.config/lxsession/LXDE/autostart`, en commentant la ligne suivante :

`@xscreensaver -no-splash`

Et en ajoutant la ligne suivante en fin de fichier

`@xset s noblank @xset s off @xset -dpms`

Enfin, j'ai édité le fichier `/etc/lightdm/lightdm.conf` et ajouter la ligne suivante dans la section **[Seat:\*]** :

`xserver-command=X -s 0 -dpms`

Il faut redémarrer le RaspberryPi pour prendre en compte ces changements.



## B. Installation et configuration de Magic Mirror

Avec cette commande :

```
bash -c "$(curl -sL https://raw.githubusercontent.com/MichMich/MagicMirror/master/installers/raspberry.sh)"
```

J'ai à présent un dossier MagicMirror dans le "home" de l'utilisateur "pi". Je me Place dans ce dossier et lancer l'installation des modules javascript.

```
cd ~/MagicMirror  
npm install
```

### ➤ Installation des modules

J'active la configuration par défaut, pour cela je copie juste le fichier :

```
cp config/config.js.sample config/config.js
```

Et puis j'ai édité le fichier **config/config.js**. Le dernier est écrit en javascript.

Dans le fichier **config.js**, j'ai commencé par mettre la langue du miroir en français :

```
language: "fr",  
timeFormat: 24,  
units: "metric",
```

j'ai jouée principalement avec la variable "modules" qui contient la liste des modules chargés sur le miroir et leurs configurations.

Un module c'est quoi ? C'est un plugin qui va effectuer une tache et dans la plupart des cas afficher une information sur le miroir.

Une déclaration de module s'effectue avec la syntaxe suivante :

```
{  
  module: "nom_du_module",  
  position: "nom_de_la_position",  
  header: "nom du texte précédent le module",  
  config: {  
    variable_config1: "valeur 1",  
    variable_config2: "valeur 2",  
  }  
}
```

```
}
```

Voici la liste des principaux éléments que compose un module :

Option	Description
Module	Nom du module. Par exemple <code>clock</code> , <code>calendar</code> , <code>mymodule</code> .
Position	La position du module sur le miroir. Les valeurs possibles sont : <ul style="list-style-type: none"><li>• <code>top_bar</code>, <code>top_left</code>, <code>top_center</code>, <code>top_right</code></li><li>• <code>upper_third</code></li><li>• <code>middle_center</code>,</li><li>• <code>lower_third</code></li><li>• <code>bottom_left</code>, <code>bottom_center</code>, <code>bottom_right</code>, <code>bottom_bar</code></li><li>• <code>fullscreen_above</code>, <code>fullscreen_below</code>.</li></ul>
Header	Optionnel: Affiche un texte au dessus du module
Disabled	Optionnel: Mettre à <code>true</code> pour désactiver le module
Config	Optionnel: Objet contenant les propriétés de configuration du module. Il faut se référer à la documentation de chaque module pour connaître les propriétés.

On peut trouver la liste complète sur [la documentation officielle ici](#).

Un exemple avec le module “newsfeed”, qui n’est autre q’un lecteur de flux RSS. Ici je le configure pour lire le flux du site “Le Monde” :

```
{
  module: "newsfeed",
  position: "bottom_bar",
  config: {
    feeds: [
      {
        title: "Le Monde",
        url: "https://www.lemonde.fr/rss/une.xml"
      }
    ],
    showSourceTitle: true,
    showPublishDate: true
  }
},
```

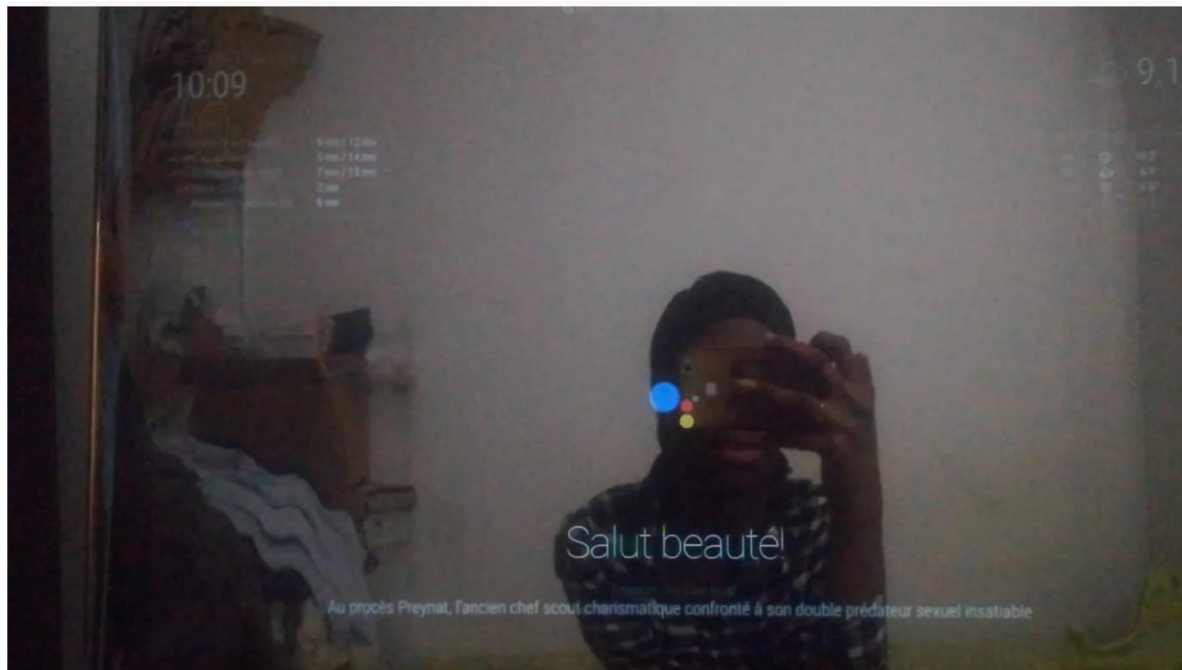
## Miroir magique

Pour les modules communautaires on y apprend qu'il faut :

- Une clé d'api, que l'on obtient en allant sur ce lien: <https://trello.com/app-key>
- Un token via ce lien  
: [https://trello.com/1/authorize?expiration=never&scope=read&response\\_type=token&name=Server%20Token&key=API\\_KEY](https://trello.com/1/authorize?expiration=never&scope=read&response_type=token&name=Server%20Token&key=API_KEY).
- Un identifiant de liste (que l'on obtient en ajoutant **.json** sur l'url de du board Trello.

Je vais avoir une configuration qui ressemble à ceci :

```
{  
  module: 'MMM-module',  
  position: 'top_left',  
  config: {  
    api_key: "ma_cle_api",  
    token: "mon_token",  
    list: "id_de_ma_liste"  
  }  
},
```



Et voilà le résultat:



## VI. Conclusion

Ce projet m'a permis de connaître les différentes étapes pour fabriquer un objet connecté, d'enrichir mes connaissances dans certains langages de programmation (JavaScript, Shell, ...) et d'acquérir des compétences à travers la création de mon miroir magique.

La fascination du miroir magique vient du fait qu'un simple objet du quotient pourrait assurer de nombreuses fonctionnalités en même temps. Il est un objet incontournable pour assurer le confort dans une maison moderne.

## VII. Problèmes rencontrés :

Pour lancer le Magic Mirror j'ai rencontré des difficultés

Le Magic Mirror ne se démarre pas automatiquement au démarrage, pour cela j'essayer de le faire démarre avec npm :

```
npm start
```

Mais toujours rien

Après quelques tâtonnements et des essais sur des commandes trouvées sur l'internet, j'ai essayé aussi pm2 et de créer un fichier mm.sh où j'écris un scripte pour le démarre le Magic Mirror en exécutant le fichier voici mon script :

```
cd ~/MagicMirror  
DISPLAY=:0 npm start
```

Pour le rendre exécutable `chmod + x mm.sh`

J'ai essayé à plusieurs reprises de formater la carte SD et de recommencer en zéro mais le problème persisté.

J'ai réussi à contourner le problème grâce aux différents forums que j'ai suivi sur l'Internet.