



MONITORING : MISE EN PLACE D'UNE SOLUTION IoT AVEC
AZURE

Dossier d'installation

Auteurs :
Florian GRANTE

Superviseur :
David LAROSE

Table des matières

Introduction	1
1 Présentation du matériel	2
1.1 La <i>RaspberryPi</i>	2
1.2 Le shield <i>GrovePi+</i>	2
1.3 Les composants	3
1.3.1 Le capteur de température et d'humidité	3
1.3.2 Le capteur de luminosité	3
1.3.3 L'écran	3
1.3.4 L'encodeur rotatif	3
2 Configuration de la station	4
2.1 Création et installation de la distribution <i>Raspbian</i>	4
2.2 Première mise en route de la <i>RaspberryPi</i>	5

Introduction

L'objectif de ce dossier est de vous expliquer de A à Z la mise en place d'un système avec capteurs permettant l'affichage en temps réel de différentes données comme la température, l'humidité ou encore la luminosité d'une pièce, d'un couloir, etc....

En premier lieu, nous allons nous attarder sur la partie matériel : *La Raspberry Pi* (je vous laisserais le choix du genre, c'est une question qui divise la France comme bien d'autre)



FIGURE 1 – *Une RaspberryPi à nue*

Nous aborderons l'installation de capteur sur cette carte puis nous verrons l'installation logiciel.

Dans un second temps, nous verrons ensemble comment envoyer ces données dans le *Cloud* mis en place par **Microsoft : Azure**

A la fin de ce dossier, vous serez en mesure de regarder n'importe où dans le monde la température ambiante de votre installation mais cessons de tergiverser et mettons nous au travail !

Chapitre 1

Présentation du matériel

1.1 La *RaspberryPi*

Si vous avez été intrigué par la bête que je vous ai mis en photo dans l'introduction, pas de panique, elle va devenir votre nouvelle meilleure amie. Il s'agit ni plus ni moins que d'un ordinateur en modèle réduit. Cette puce est capable de faire tourner une distribution de *Linux* connue sous le nom de *Raspbian*. La *RaspberryPi* est un véritable bijoux de miniaturisation pour les *Makers* qui désire embarquer des ordinateurs dans leur projets les plus fou. Vous l'aurez compris, cela va être notre base de notre station.

Pourquoi ce choix ?

La *RaspberryPi* peut être repoussante au début car vendue tel que vous avez pu la voir plus tôt mais une fois prise en main, elle est simple d'utilisation. Elle permet d'avoir un accès internet de façon simple avec son port ethernet, on peut y brancher un écran ainsi que des périphériques si nécessaire pour réaliser de la maintenance (même si il existe d'autre moyen d'y accéder mais nous verrons ça plus tard). Une raison non négligeable pour notre projet est son prix. Compté une quarantaine d'euros pour le dernier modèle en date. Elle possède également des broches pour réaliser de l'électronique embarquée comme nous allons le faire. Ainsi notre ordinateur est modulaire et on peut y ajouter des composants par ces broches. Son autre atout et non des moindre, c'est sa communauté qui s'est étendue de façon exponentielle ces dernières années car accessible à tout le monde et permet alors une aide conséquente via les forums.

Vous voilà plus familier avec l'engin, si vous voulez plus d'information, je vous laisse visiter [le site officiel](#) où vous pouvez notamment télécharger la version de base de son OS, *Raspbian* mais qui possède nombre de version modifiée (un OS pour réaliser une station de jeux rétro est un exemple)

1.2 Le shield *GrovePi+*

Je vous parlait juste avant que la *RaspeberryPi* possédait des broches. Ces dernières vont nous servir à y installer les capteurs. Mais nous n'allons pas brancher les capteurs directement sur la *RaspberryPi*. En effet nous allons ajouter ce que l'on appelle un "*Shield*". Il s'agit ni plus ni moins que d'une autre carte électronique avec ses fonctions propre qui se branche sur la *RaspberryPi* pour communiquer avec elle. Ainsi, nous pouvons utiliser les fonctions du "*Shield*" en envoyant des instruction de puis la "*RaspberryPi*".

Celui que nous utilisons porte le doux nom *GrovePi+*. Sans m'attarder sur ses détails techniques, sachez (pour les connaisseurs) qu'elle embarque un microcontrôleur ATMega pour gérer les capteurs. Il s'agit ni plus ni moins d'une version "*User Friendly*" d'un *Arduino* que l'on branche sur notre *RaspberryPi*.

Comme vous voyez sur la photo, nous avons des connecteurs 4-pins, en blanc. C'est là dessus que l'on va brancher nos capteurs.

ATTENTION : Ces connecteurs sont reliés à des broches particulière du microcontrôleur. Ils sont donc chacun d'entre eux prévu pour un type de composant particulier.

On distingue trois types de connecteurs parmi ceux-là :

- port analogique : Ils sont disponible uniquement en *INPUT*, c'est à dire qu'il s'agit d'une entrée pour une tension comprise entre 0 et 5V. Le *Shield* possède un convertisseur analogique numérique qui échantillonne cette valeur sur 10bits ($0V = 0$ et $5V = 1023$). Exemple si on envoie 2.5V, alors on aura la valeur 512)
- port digital : ceux ci peuvent être régler en *INPUT* ou en *OUTPUT*. Comme ce sont des entrée/sortie numérique, elle répond à la loi du tout ou rien. Soit nous avons 0V, soit nous avons 5V (0 ou 1). Typiquement, pour une sortie cela peut être utilisé comme interrupteur électronique.

- port I2C : Il s'agit d'un protocole de communication développé par Philips pour minimiser le nombre de fil nécessaire pour faire communiquer deux périphériques entre eux. Il y a en effet uniquement 3 fils : un signal de données (SDA), un signal d'horloge (SCL), et un signal de référence électrique (masse).
Nous allons maintenant parler rapidement des différents capteurs et composants que nous allons brancher sur notre *Shield*

1.3 Les composants

1.3.1 Le capteur de température et d'humidité

Les capteurs d'humidité répondent à une loi qui dépend de la température. C'est pour cette raison que ces capteurs fournissent bien souvent les deux à la fois. De notre côté, il s'agit du capteur *DHT11*. Ce composant utilise un connecteur digital

Datasheet : [%urlici](#)

1.3.2 Le capteur de luminosité

Le capteur de luminosité mesure la lumière ambiante. Exprimé en Luxmètre, les capteurs de luminosité "basique" ne permettent pas d'obtenir une valeur précise de la luminosité. Néanmoins, cette valeur est très abstraite pour l'utilisateur. C'est pour cette raison que nous allons l'utiliser pour déterminer si la lumière de la pièce où il est placé est allumé ou non. Nous détaillerons un protocole à réaliser pour essayer de faire marcher au mieux ce capteur. Ce composant utilise un connecteur analogique

Datasheet : [%urlici](#)

1.3.3 L'écran

Pour pouvoir accéder localement aux valeurs des capteurs, nous allons placer un écran. Nous pouvons afficher 2 lignes de 16 caractère sur ce dernier et choisir la couleur de fond qui est affiché. Ce composant utilise un connecteur I2C. Datasheet : [%urlici](#)

1.3.4 L'encodeur rotatif

Nous n'allons pas l'utiliser comme un capteur à proprement parler ici mais d'un bouton tournant pour naviguer dans différents affichage sur l'écran. On pourra ainsi passer de l'affichage de la température à l'humidité,etc...

Maintenant que nous avons connaissance du matériel que nous avons à disposition pour notre station, nous allons pouvoir passer à son installation.

Chapitre 2

Configuration de la station

2.1 Creation et installation de la distribution *Raspbian*

Pour une *RaspberryPi*, son disque dur est nulle autre qu'une carte Micro SD. C'est donc sur ce support que nous allons faire l'installation.

Pour realiser cette tape, vous aurez besoin de :

- d'une *RaspberryPi*
- d'une carte micro SD de minimum 4Go
- du logiciel *Etcher*
- de *Raspbian*. En realite il s'agit d'une version modifie par la socit Dexter Industries qui est spcialis dans l'utilisation de *RaspberryPi* pour la robotique.
- d'un adaptateur pour relier la micro SD  votre ordinateur.

Nous pouvons dsormais commencer l'installation de l'OS sur notre *RaspberryPi*

1. Connecter la micro SD sur votre ordinateur.

2. Lancer le logiciel *Etcher*

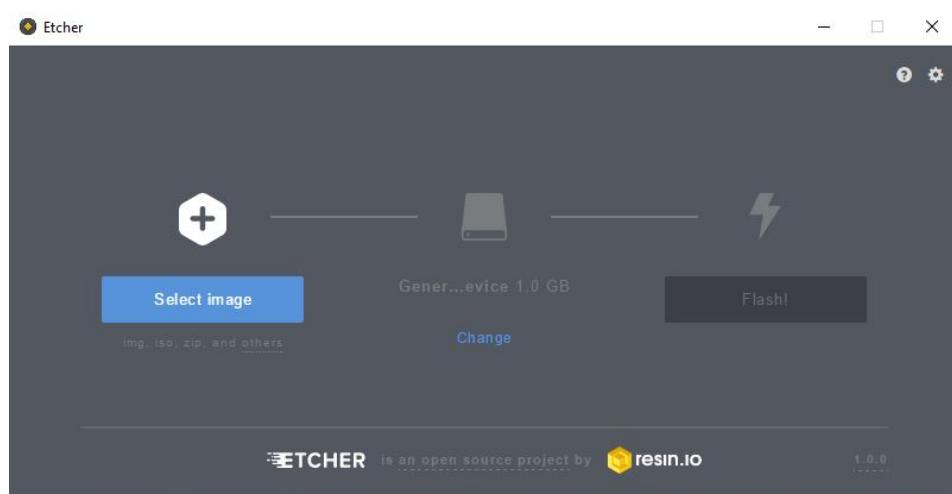


FIGURE 2.1 – Logiciel *Etcher* au lancement

3. Cliquer sur "Select image"  gauche puis selectionner le fichier .zip recupr depuis le site *SourceForce* dans les pr-requis.
4. Verifier que le periphrique qui est renseign au centre est bien la micro SD. Dans le cas contraire cliquer sur "Change".
5. Si les deux tapes prcdentes sont OK, cliquer sur "Flash".

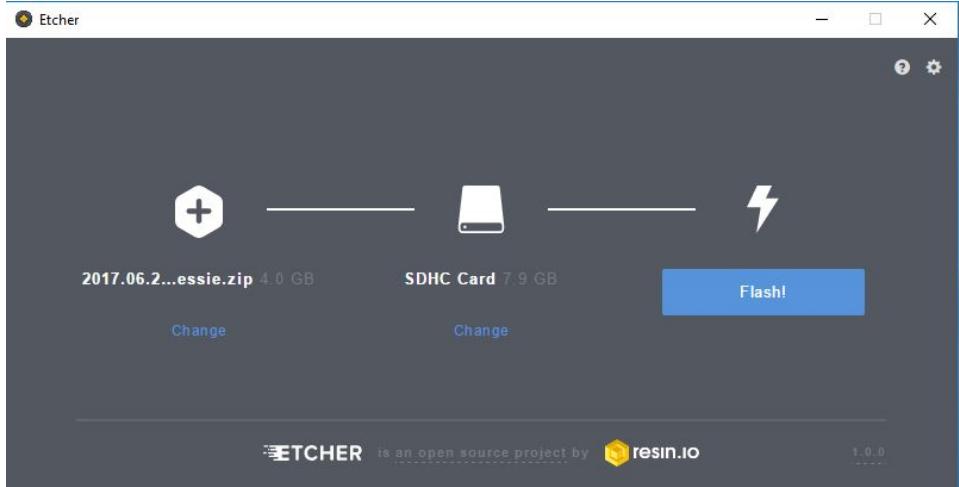


FIGURE 2.2 – *Etcher* prêt à flasher

Et voilà, l'opération peut prendre une dizaine de minutes. C'était simple non ?

2.2 Première mise en route de la *RaspberryPi*

- Maintenant que vous avez *Raspbian*, nous pouvons démarrer notre nouvel ordinateur. Pour cela, il vous suffit d'insérer la microSD au dos de la *RaspberryPi*.



FIGURE 2.3 – La *RaspberryPi* de dos

- Ensuite, nous allons pouvoir lancer la *RaspberryPi*. Avant de l'alimenter, nous allons distinguer deux cas.. Si vous avez la possibilité, de façon extérieur, d'accéder à l'adresse IP de votre *RaspberryPi* alors vous pouvez sauter l'étape N°3. Si cela n'est pas possible, brancher un écran et un clavier.
Vous pouvez maintenant l'alimenter en utilisant le port micro USB qui est à côté du port HDMI.
- Si vous suivez cette étape, vous devriez voir apparaître des lignes de commandes qui défilent. Quelques instants plus tard, vous arrivez sur l'environnement de bureau de votre *RaspberryPi*.
Une fois le terminal ouvert, entrer la commande suivante :

```
1 sudo ifconfig
```

un mot de passe devrait vous être demandé, par défaut, le mot de passe est "robots1234"

4. Très bien, désormais que vous avez l'adresse IP à disposition, nous allons pouvoir installer le nécessaire pour utiliser nos capteurs. Nous aurions très bien pu continuer cette installation directement sur la *RaspberryPi* mais si vous n'avez jamais fait ce qui va suivre, cela vous fera un bon entraînement.

- (a) Télécharger le logiciel [Git Bash](#)
- (b) Lancer *Git Bash*
- (c) taper la commande en remplaçant "xxx.xxx.xxx.xxx" par l'adresse IP de la *RaspberryPi*

```
1 ssh pi@xxx.xxx.xxx.xxx
```

la première fois, il vous sera demander si vous faites confiance, taper alors "yes" puis sur *Entrée*

- (d) Le mot de passe est "robots1234". Si tout c'est bien passé, vous devriez avoir cet aperçu :
- (e) vous naviguez maintenant dans la *RaspberryPi*. Dans un premier temps, nous allons changer le mot de passe car celui ci est un mot de passe par défaut. Enter la commande :

```
1 sudo raspi-config
```

un écran bleu devrait apparaître.

- (f) Quitter le menu pour revenir au terminal.
- (g) Maintenant, nous allons installer GrovePi+ pour pouvoir utiliser le *Shield*. Entrer alors les deux commandes suivantes :

```
1 sudo curl https://raw.githubusercontent.com/DexterInd/Raspbian_Forum_Robots/master/upd_script/fetch_grovepi.sh | bash
2
3
4 sudo reboot
```

Votre *RaspberryPi* va redémarrer.

- (h) Connectez-vous à nouveau en "ssh" comme pour l'étape 4 mais cette fois ci avec votre nouveau mot de passe.
- (i) Réaliser alors cette suite de commande une à une. Appuyer sur la touche *Entrée* lorsque l'on vous demande de continuer :

```
1 cd /home/pi/Desktop
2 sudo git clone https://github.com/DexterInd/GrovePi
3 cd /home/pi/Desktop/GrovePi/Script
4 sudo chmod +x install.sh
5 sudo ./install.sh
```

- (j) Appuyez à nouveau sur *Entrée* une fois arrivé sur cette configuration :
- (k) Au moment où la *RaspberryPi* va redémarrer (vous verrez un "Restart" écrit dans le terminal. Appuyez sur *Ctrl + C* pour empêcher le redémarrage.
- (l) Effectuer alors la commande :

```
sudo shutdown now
```

Elle va alors s'arrêter. Vous pouvez alors la débrancher une fois que vous avez un écran noir.

5. Vous pouvez désormais ajouter le *Shield* sur la *RaspberryPi* Comme ci-dessous **ATTENTION AU BROCHES UTILISÉES SUR LA PHOTO !**
6. Brancher à nouveau la *RaspberryPi*. Vous devriez avoir une LED qui s'allume sur votre *Shield*

7. nous allons tester s'il a bien été reconnue, pour cela, connecter vous en "ssh" sur votre *RaspberryPi* (vous devriez savoir le faire maintenant !)
8. lancer la commande : [style=MyBashStyle] sudo i2cdetect -y 1 vous devriez obtenir ce résultat, avec le 04 en première ligne.

Voilà, la première mise en route de la *RaspberryPi* est terminé, nous allons maintenant pouvoir nous occuper du code pour la station final.

2.3 Installation, configuration et test du code de la station avec ses capteurs.