

# Projet SI3 – PS6

## Introduction à l'IoT

### 1 Plate-forme matérielle/logicielle IoT

Pour mettre en œuvre un prototype rapidement, rien de tel qu'une plateforme du type Raspberry Pi ou Arduino. Pour vous faciliter la vie, nous n'avons pas choisi un Arduino, mais Raspberry Pi Zéro, plus proche de vos connaissances actuelles.

#### 1.1 Raspberry Pi Zéro WH

##### 1.1.1 Présentation

La Raspberry Pi que nous vous fournissons est une version Zéro (avec le plus petit encombrement) et WH (qui possède une interface de communication Wifi, Bluetooth et un port GPIO soudé à la carte). Mais avant de pouvoir y faire quoi que ce soit, il faut d'abord installer le système d'exploitation qui lui permettra de démarrer.



Voici un résumé rapide des étapes nécessaires pour l'installation du système d'exploitation sur la Raspberry Pi. Il y a de nombreux tutoriels, mais nous vous conseillons d'aller à celui de référence :

<https://www.raspberrypi.org/documentation/installation/installing-images/>

Vous veillerez à télécharger l'image « Raspbian Stretch Lite » qui est la plus compacte et vous laissera le plus de place possible. Puis, suivant le système d'exploitation de votre machine de travail, vous choisirez en bas de la page le tutoriel adapté à votre cas.

##### 1.1.2 Configuration pour se connecter au réseau

Une fois l'image enregistrée sur la carte SD qui vous a été fournie, il va vous falloir activer la possibilité de vous connecter à distance ainsi que lui donner les informations pour se connecter à un réseau WiFi. Pour ces deux étapes cruciales, il vous suffit de créer deux fichiers dans la partition /boot de la carte SD.

- Tout d'abord, vous allez créer le fichier `ssh` (sans extension). Ceci permettra d'autoriser la connexion via le service `ssh` qui a été installé sur le système d'exploitation mais pas activé.
- Ensuite, vous créez un fichier `wpa_supplicant.conf` avec le contenu suivant (que vous modifierez en fonction du nom de votre Wifi et du mot de passe pour s'y connecter) :

```
country=fr
update_config=1
ctrl_interface=/var/run/wpa_supplicant

network={
scan_ssid=1
ssid="WIFI_Name"
psk="cle_de_securite"
}
```

Nous vous invitons à vous connecter sur le Wifi de votre téléphone par exemple pour éviter tout problème de conflit avec plusieurs plateformes connectées au même Wifi (conflit de nom tant que l'on ne l'a pas changé).

Une fois ces deux fichiers créés, vous pouvez insérer la carte SD dans la Raspberry Pi Zéro et la brancher. Le premier démarrage est un peu plus long car il y a plusieurs opérations à effectuer avant de vous autoriser à vous connecter.

Pour vous connecter à la plateforme, il vous suffira de faire :

```
ssh pi@raspberrypi.local
```

# Projet SI3 – PS6

## Introduction à l'IoT

Si vous n'arrivez pas à vous connecter de votre poste de travail à la Raspberry Pi, c'est soit que votre machine n'est pas connectée sur le même réseau Wifi, soit que votre poste de travail n'a pas le protocole Bonjour. Dans ce cas, installez le pour votre machine et retentez de vous connecter.

### 1.1.3 Configuration du Systèmes d'Exploitation

Une fois connecté à la Raspberry par ssh, il va vous falloir mettre à jour le système d'exploitation et installer quelques paquetages complémentaires à l'aide des commandes suivantes :

```

sudo apt-get update && sudo apt-get upgrade
sudo apt-get install i2c-tools git
sudo raspi-config
  
```

Avec la dernière commande, vous pouvez configurer la timezone ainsi que le nom de votre machine. Si vous changez celui-ci (ce qui est conseillé), vous utiliserez le nom choisi (`mon_nom.local`) à la place de `raspberrypi.local` dans la suite de ce tutoriel. Vous devriez aussi changer le mot de passe par défaut pour se connecter avec l'utilisateur `pi`.

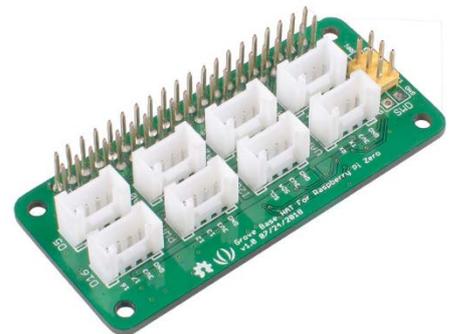
Vous êtes maintenant prêt à passer à la suite.

## 1.2 Grove Base Hat

### 1.2.1 Présentation

La Raspberry Pi possède un port GPIO (*General Purpose Input/Output*) pour communiquer avec différents capteurs et actionneurs. Pour faciliter la connexion de ces capteurs et actionneurs, nous allons utiliser une carte d'extension à la Raspberry Pi Zéro WH : la Grove Base Hat pour Raspberry Pi Zéro. Cette plateforme vous donne la possibilité de connecter 5 types de capteurs ou actionneurs :

- Digitaux
- Analogiques
- I2C
- PWM
- Série



La connectique fournie pour la carte est compatible Grove. Cette gamme de produit possède un nombre important de capteurs et d'actionneurs prêts à être utilisés à la fois sur des plateformes à base de microcontrôleur et de microprocesseur (même si tous ne sont pas compatibles avec cette dernière plateforme).

### 1.2.2 Configurations logicielles pour mettre en œuvre cette carte d'extension

Pour utiliser cette carte d'extension, il est nécessaire d'installer plusieurs nouveaux paquetages et un logiciel pour disposer d'exemples d'utilisation (en Python, malheureusement...).

Voici les commandes à lancer pour installer ce qui est nécessaire (attention au « - » si vous faites un copier-coller).

```

sudo apt-get install python-pip python3-pip
curl -sL https://github.com/Seeed-Studio/grove.py/raw/master/install.sh | sudo bash
-s -
git clone https://github.com/Seeed-Studio/grove.py
cd grove.py
# Python2
sudo pip install .
# Python3
sudo pip3 install .
  
```

# Projet SI3 – PS6

## Introduction à l'IoT

Vous trouverez les instructions détaillées et les mises à jour possibles de cette procédure à la page suivante :

<https://github.com/Seeed-Studio/grove.py#installation>

Vous voilà prêts à tester un capteur et un actionneur.

**Attention : L'électronique de cette plateforme est un peu limitée. Il n'est donc pas possible de connecter ou de déconnecter un capteur ou un actionneur alors que la plateforme fonctionne. Il est donc impératif d'éteindre la Raspberry Pi pour faire toute modification de connexion des capteurs et actionneurs. Sinon, votre système plante et vous n'avez plus accès à la carte SD. Vous serez donc obligé de lancer la commande**

```
sudo poweroff
```

**avant de modifier ce qui est connecté à la plateforme.**

### 1.2.3 Capteur et Actionneur

Nous vous avons fourni avec la plateforme un capteur (le bouton) et deux actionneurs : la LED qui se trouve dans le bouton et un afficheur 7 segments. Vous pouvez déjà tester l'afficheur 7 segments avec le script Python fourni dans les sources que vous avez téléchargé par git (`grove_4_digit_display.py`). Ceci vous permet de voir si vous avez bien configuré la plateforme. Vous connecterez ce Grove\_4\_Digit\_Display sur le port Digital D5.

[http://wiki.seeedstudio.com/Grove-4-Digit\\_Display/](http://wiki.seeedstudio.com/Grove-4-Digit_Display/)

De la même manière, vous pouvez tester la bonne réception de l'appui sur le bouton. Vous connecterez le bouton sur le port Digital D16.

[http://wiki.seeedstudio.com/Grove-LED\\_Button/](http://wiki.seeedstudio.com/Grove-LED_Button/)



## 2 Orchestration de flots de l'Internet des Objets

### 2.1 Installation de l'environnement Node-RED

Maintenant que vous disposez de capteurs et actionneurs, vous allez vous intéresser à l'environnement Node-RED qui nous permettra d'exploiter ces informations.

<https://nodered.org/>

Node-RED est basé sur Node.js (que vous devez déjà utiliser ou que vous utiliserez pour votre projet). Pour installer Node-RED sur la Raspberry Pi, voici les instructions nécessaires :

```
bash <(curl -sL https://raw.githubusercontent.com/node-red/raspbian-deb-package/master/resources/update-nodejs-and-nodered)
cd ~/.node-red
npm install node-red-dashboard
```

Pour lancer Node-RED, il vous suffira une fois l'installation terminée de lancer la commande `node-red`. L'interface de contrôle de l'environnement est alors accessible sur votre machine à l'adresse suivante :

<http://raspberrypi.local:1880/>

## Projet SI3 – PS6

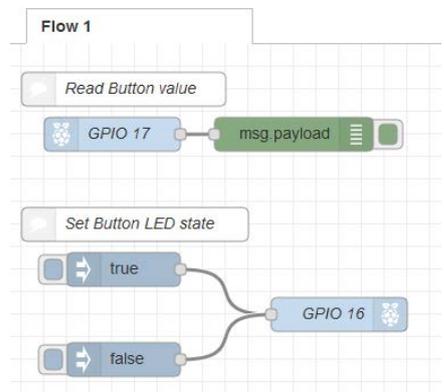
### Introduction à l'IoT

#### 2.2 Un premier flot de test

Commencez par tester le bon fonctionnement depuis Node-RED du bouton (à la fois un capteur et un actionneur). Voici un exemple que vous pouvez importer dans Node-Red pour tester à la fois que vous pouvez contrôler la LED et récupérer l'information de clic sur le bouton.

Pour réaliser vous-même cet assemblage, vous utiliserez les nœuds « *rpi gpio* » de la section Raspberry Pi qui permettent d'accéder aux ports GPIO. Il n'y a pas vraiment de logique mise en place dans ce cas très simple, mais cela permet de vérifier que vous avez bien accès aux informations.

Pour faciliter le test, voici le ce flot que vous pouvez importer directement sous Node-RED (Menu à droite, Import / Clipboard).



```
[{"id":"b53447d5.07ee2","type":"tab","label":"Flow 1","disabled":false,"info":"","z":"b53447d5.07ee2","name":"GPIO 16","pin":"36","set":false,"level":"0","freq":"","out":"out","x":300,"y":240,"wires":[]}, {"id":"ed29d8be.0b871","type":"rpi-gpio out","z":"b53447d5.07ee2","name":"GPIO 16","pin":"36","set":false,"level":"0","freq":"","out":"out","x":300,"y":240,"wires":[]}, {"id":"fa699296.f7ced","type":"inject","z":"b53447d5.07ee2","name":"","topic":"","payload":"true","payloadType":"bool","repeat":"","crontab":"","once":false,"onceDelay":0.1,"x":110,"y":200,"wires":[["ed29d8be.0b871"]]}, {"id":"48159fe6.2f378","type":"rpi-gpio in","z":"b53447d5.07ee2","name":"GPIO 17","pin":"11","intype":"tri","debounce":"25","read":true,"x":100,"y":80,"wires":[["fffb62e34.dfc428"]]}, {"id":"fffb62e34.dfc428","type":"debug","z":"b53447d5.07ee2","name":"","active":true,"tosidebar":true,"console":false,"tostatus":false,"complete":false,"x":260,"y":80,"wires":[]}, {"id":"1d23bb8.4e0d145","type":"inject","z":"b53447d5.07ee2","name":"","topic":"","payload":"false","payloadType":"bool","repeat":"","crontab":"","once":false,"onceDelay":0.1,"x":110,"y":280,"wires":[["ed29d8be.0b871"]]}, {"id":"bbfef175.66075","type":"comment","z":"b53447d5.07ee2","name":"Read Button value","info":"","x":110,"y":40,"wires":[]}, {"id":"44bd743a.03dee4","type":"comment","z":"b53447d5.07ee2","name":"Set Button LED state","info":"","x":120,"y":160,"wires":[]}]
```

#### 2.3 L'afficheur Grove 4 digits à 7 segments

Pour pouvoir utiliser l'afficheur 7 segments, il est nécessaire d'avoir un code dédié pour envoyer les différentes informations pour l'allumage des segments. Un code Python d'exemple est fourni dans les sources de grove.py que vous avez clonées précédemment. Mais bien entendu ce code en Python n'est pas utilisable sous Node.js. Je vous ai donc écrit le code JavaScript et vous fournis le nœud utilisant ce code pour vous permettre de faire un affichage avec l'afficheur 7 segments. Pour ajouter un paquetage à Node-RED (et rendre de nouveaux nœuds disponibles), vous pouvez le faire depuis l'interface web de Node-RED (Menu / Manage Palette / Install) ou bien via le terminal avec les commandes suivantes :

```
cd ~/.node-red
npm install nom_du_paquetage
```

Dans notre cas particulier, comme le paquetage n'est pas publié sur npmjs, vous devez l'installer avec les commandes suivantes :

```
cd ~/.node-red
npm install https://github.com/lavirott/node-red-contrib-grovehat
```

Juste en passant, vous jetterez un coup d'œil à la console pendant l'installation. Mais que peuvent bien signifier les lignes `node-gyp rebuild` puis `CXX(target)` et `SOLINK_MODULE(target)` ??? Cela vous fera peut-être penser à des choses que vous avez vu dans l'excellent cours « Programmation Systèmes » ! 😊

## Projet SI3 – PS6

### Introduction à l'IoT

Une fois le nœud installé, relancez Node-RED pour en bénéficier. Vous pourrez faire un petit test en affichant l'heure sur l'afficheur par exemple.



```
[{"id":"cb01310a.0aa448","type":"tab","label":"Flow 2","disabled":false,"info":""},{id":"3c44253f.955fba","type":"grove-4-digit-display","z":"cb01310a.0aa448","name":"","pin":"5","x":460,"y":40,"wires":[]},{id":"f6f51c77.a63a08","type":"inject","z":"cb01310a.0aa448","name":"","topic":"","payload":"","payloadType":"date","repeat":"1","crontab":"","once":false,"onceDelay":"0.1","x":110,"y":40,"wires":[["5897622e.bac58c"]]}, {"id":"5897622e.bac58c","type":"function","z":"cb01310a.0aa448","name":"timeConvert","func":"if ( !msg.timestamp ) msg.timestamp = Math.round(+new Date());\n\nvar dt = new Date(msg.timestamp);\n\nvar msg = {\n\t'month':\tdt.getMonth() + 1,\n\t'day':\tdt.getDate(),\n\t'year':\tdt.getFullYear(),\n\t'hours':\tdt.getHours(),\n\t'mins':\tdt.getMinutes(),\n\t'secs':\tdt.getSeconds()\n}\n\nmsg.payload = msg.hours + \"\" + msg.mins;\n\nreturn msg;","outputs":1,"noerr":0,"x":270,"y":40,"wires":[["3c44253f.955fba"]]}]
```

### 3 Conclusion et perspectives pour la suite de projet

Vous venez de mettre en place et de tester des fonctionnalités que vous allez pouvoir utiliser dans votre projet. Il vous reste maintenant deux étapes essentielles :

- Ajouter la logique de traitement des données issues des capteurs ou envoyés aux actionneurs,
- Ajouter les protocoles de communication qui vous permettront d'intégrer votre objet avec vos développements précédents (backend de gestion de liste d'attente).

L'avantage de Node-RED est de vous fournir des nœuds prêts à l'emploi pour toutes sortes de protocoles de communication et de fonctionnalités. Vous veillerez à choisir le protocole adapté pour interconnecter votre objet connecté avec votre backend de gestion de liste d'attente.

A vous de jouer maintenant !