

Groupe 2 :
Théo Kunegel – Joseph Liagre – Alice Moron
(TS2)

Rapport ISN

LA MAISON CONNECTÉE

SOMMAIRE :

INTRODUCTION SUR LE PROJET

MATÉRIEL UTILISÉ

DÉMARCHE

CONCLUSION

ANNEXE

[INSÉRER PHOTO MAQUETTE FINALE]

I/ Introduction au Projet

Il y a à peine 20 ans les premiers smartphones furent mis en vente pour la première fois. Ils n'ont cessés de se développer depuis et nous accompagnent de plus en plus dans notre quotidien. Leur utilisation est maintenant plus diversifiée grâce aux différentes applications. On peut aujourd'hui contrôler le conditionneur d'air d'une salle de sa maison ou même se préparer un café à distance.

La technologie continue à se développer et nous nous dirigeons vers une automatisation totale des objets présents tous les jours notre vie. Nous nous sommes donc concentrés sur l'automatisme au sein d'un domicile, en autres termes: la domotique. L'idée pour notre projet nous est venue à partir de ce concept. Nous voulions créer le prototype d'une maison qui serait le plus automatisé et autonome possible, ainsi notre maison effectuerait des actions dépendant de l'environnement et nous évitant certaines tâches pénibles.

L'idée de ce projet nous a particulièrement plu et intéressé car le sujet abordé est au coeur de l'innovation d'aujourd'hui et suscite un certain excitement lorsque l'on imagine qu'un monde entièrement automatisé et robotisé peut bel et bien se mettre en place dans un futur proche. De plus la domotique est un sujet qui nous concerne directement d'une certaine façon puisqu'elle serait présente dans notre vie de tous les jours. Nous nous sentons ainsi concernés par ce domaine qui cherche le développement et l'amélioration de nos conditions de vies.

Nous avons donc décidé de modéliser une maison connectée et intelligente qui exécuterait 3 tâches : celle d'allumer des lumières, celle d'ouvrir ou de fermer des stores et enfin de contrôler un ventilateur. Nous voulions que les différents modules de la maison s'activent en fonction des changements de température, de luminosité ou par l'intermédiaire d'un utilisateur qui le contrôle par son smartphone.

Lorsque la luminosité extérieure (stimulée) baisserait, des lampes s'allumeront et cela fermerait les stores puisque "l'habitant" n'aurait plus besoin de la lumière du soleil. Lorsque la température (stimulée également car la température d'une salle ne varie pas assez vite) atteint une certaine valeur, le ventilateur se mettrait en marche permettant de rafraîchir la pièce avec de l'air naturel. Comme mentionné auparavant, une ampoule s'allumerait à notre guise directement grâce à une interface personnalisée avec un smartphone ou un ordinateur.

II/ Matériel utilisé

Pour concrétiser ce projet, nous avons besoin, pour la domotique :

Un ordinateur Raspberry Pi 3

Des câbles de connection

Des ampoules LED avec des résistances

2 moteurs

2 ventilateurs

Un relai à 4 chaînes

Une plaque de connection

Un capteur de température

Une photorésistance avec un capaciteur (utilisé comme un capteur de lumière)

Un générateur de laboratoire (pour pouvoir tout alimenter en courant)

Un Peltier (pour pouvoir simuler les hausses de températures)

potentiellement une borne d'accès WIFI

Pour la maquette, nous avons utilisé du papier plume

III/ Démarche

Afin de créer une maison connectée fonctionnelle, il nous a fallu entamer une démarche précise. Le but de ce projet est d'abord de pouvoir contrôler à distance des lumières, des volets et des ventilateurs via une interface web et aussi de créer un système qui enclencherait ces éléments en fonction des conditions (température, luminosité).

Ainsi il nous a fallu créer une interface web qui nous permettrait de contrôler à distance ces éléments par le biais du Raspberry Pi. Afin que celui-ci puisse effectuer les actions nécessaires au fonctionnement des LED, des moteurs et des ventilateurs, nous avons codé des scripts en python insérés dans le Raspberry Pi.

En attendant l'arrivée de notre matériel, nous avons tout d'abord créé une interface Web nous permettant de contrôler depuis un ordinateur ou un smartphone une ampoule LED que nous ne possédions pas encore. Nous avons tout d'abord utilisé, pour avoir un modèle de site web, le langage HTML, CSS et des images que nous avons créés nous mêmes. Lorsque cette interface fut terminée, nous avons converti cette interface en PHP pour que l'on puisse exécuter à distance des commandes sur le Raspberry Pi directement depuis le site web. En effet, le langage HTML ne permet pas cela: si un code en HTML de lancement de fichier est lancé, le fichier en question ne peut pas être démarré mais sera seulement téléchargé sur la machine. Avec le langage PHP en revanche, on peut faire lancer le fichier à distance directement sur le serveur. Dans notre projet, le serveur de fichier est le Raspberry Pi.

*1

[Langages de description: Présentation du langage HTML et du principe de séparation du contenu et de la mise en forme.]

[Formats: Les données numériques sont agencées de manière à en faciliter le stockage et le traitement. L'organisation des données numériques respecte des formats qui sont soit des standards de fait, soit des normes.]

Nous avons ensuite pu configurer le Raspberry Pi et nous avons également pu nous y familiariser. Nous avons testé le module Sense HAT, un module contenant de multiples et divers capteurs en plus de plusieurs boutons et d'une matrice de

LED. Nous avons commencé par tester des programmes fournis par Raspberry pour pouvoir contrôler les ampoules LED sur le HAT. *2 Nous avons également testé les différents capteurs du Hat dont ceux pour l'humidité et la température. Nous avons tout de suite remarqué que les valeurs affichées pour le second capteur étaient fausses puisqu'elles captaient la température ambiante autour du Raspberry Pi qui est bien plus chaude que la température de la salle puisque le processeur de l'ordinateur chauffe l'air. Nous en sommes donc arrivés à la conclusion que le capteur de température du HAT ne pouvait pas être utilisé dans ce projet puisque ce capteur est trop proche du centre de contrôle central. *3 Il nous fallait donc un autre capteur de température positionné à une certaine distance du Raspberry Pi.

[Numérisation: L'ordinateur manipule uniquement des valeurs numériques. Une étape de numérisation des objets du monde physique est donc indispensable.]

Nous avons, lorsque nous avons reçu les LED, nous en avons branché une sur le connecteur GPIO numéro 2, et nous avons commencé à tester certains codes que nous avons programmé en Python sur le Raspberry Pi au préalable. Lors de cette étape, nous devons faire attention à choisir le bon connecteur GPIO (connecteur 2, un des 26 connecteurs). *4 Nous avons tout d'abord testé les LED sans le HAT puis, lorsque cela fonctionnait parfaitement, nous avons branché les LED avec le HAT. Cependant, nous avons rencontré un problème car le Raspberry Pi clignotait de manière aléatoire, nous alertant que nous étions surement arrivés à la limite de courant. Nous en avons donc déduit que nous ne pouvions pas utiliser le Sense HAT dû à ce problème. Toutefois, nous avons pu tester des codes directement du terminal python avec les LED directement connectées au raspberry pi.

[Correction d'un programme - test ; - instrumentation ; - situations d'erreur ou bugs.]

Au départ, nous avons prévu de connecter les LED à un relai de 8 chaînes pour que la stabilité de chacune d'elle soit exacte et qu'il n'y ait pas de manque de courant. Étant donné que les relais étaient en rupture de stock à toutes les sources, nous avons modifiés notre projet initial. Nous avons alors décidé de contrôler les LED directement du raspberry Pi à l'aide d'une résistance, ce qui a fonctionné. À partir de ces tests, nous avons pu mettre à jour les codes pour qu'ils puissent marcher depuis notre interface web. Cependant, nous avons rapidement constaté que les codes en python ne démarraient pas à cause d'un manque de permissions sur le raspberry pi. Cela a pu être résolu lorsqu'on a changé le type de démarrage de la lampe: au lieu de démarrer la lampe à l'aide d'un fichier Python, celle-ci démarrait par le biais d'un code de terminal lancé directement par le site web PHP. *5

[Formats: Les données numériques sont agencées de manière à en faciliter le stockage et le traitement. L'organisation des données numériques respecte des formats qui sont soit des standards de fait, soit des normes.]

Nous avons ensuite utilisé un partage de connection d'un téléphone portable afin de simuler un réseau WIFI classique d'un domicile. Suite à cela, tout appareils connectés à la même borne WIFI ont pu avoir accès à notre interface web. Nous avons dû utiliser cette solution puisque nous n'avions pas pu nous procurer un routeur. Un routeur aurait simplifier la tâche de réglage de la connection WIFI sur le Raspberry Pi et sur le téléphone portable. De plus, sans routeur, l'adresse IP du raspberry change puisque l'on ne peut pas régler d'adresses IP fixes sur le partage de connection.

[Adressage sur un réseau: Mécanismes d'adressage pour identifier des machines]

distantes.]

[Transmission point à point: Principes de base d'une transmission d'informations numériques entre un émetteur et un récepteur.]

Lorsque nous avons enfin obtenus la plaque de connection, la peltier, le capteur de température et la photorésistance, nous avons pu testé ces derniers. Nous avons tout d'abord vérifié que le peltier alimenté par un générateur pouvait devenir assez chaud pour pouvoir stimuler un changement de température. Celui-ci a effectivement atteint des températures assez élevées au point de faire un peu fondre le câble de connection. Nous avons ensuite tenté de faire marcher la photorésistance. Ceci fut un succès : nous avons les valeurs qui s'affichaient toutes les secondes. La valeur de la lumière de la salle était d'environ 500Ω (ou vers 1000lux), lorsqu'un flash était envoyé sur la photorésistance, celle-ci descendait à 0Ω, et lorsque que l'on recouvrait la photorésistance, la valeur montait jusqu'à 1500Ω. *6 Nos premières tentatives pour faire fonctionner le capteur de température ont malheureusement étaient vaines. *7

Prochaines étapes:

- codes tableaux de valeurs pour les capteurs
- faire fonctionner le relais
- faire marcher le ventilateur ou le moteur grâce aux variations lumière/ température
- faire marcher toute les pièces par le biais du smartphone
- faire la maquette

IV/ CONCLUSION

V/ ANNEXE

1. Interface web en PHP

```
1 <html>
2 <head>
3 <meta name="viewport" content="width=device-width" />
4 <title>LED Control</title>
5 </head>
6 <body>
7   Controle de la lumiere:
8   <form method="get" action="index.php">
9     <input type="submit" value="I" name="on">
10    <input type="submit" value="O" name="off">
11  </form>
12  <?php
13    $setmode3 = shell_exec("/usr/local/bin/gpio -g mode 3 out");
14    if(isset($_GET['on'])){
15      $gpio_on = shell_exec("/usr/local/bin/gpio -g write 3 1");
16      echo "Lumiere allumee";
17    }
18    else if(isset($_GET['off'])){
19      $gpio_off = shell_exec("/usr/local/bin/gpio -g write 3 0");
20      echo "Lumiere eteinte";
21    }
22  <?>
23 </body>
24 </html>
```

2. Allumer toute les LED du Sense HAT en blanc

```

1 import time
2 from sense_hat import SenseHat
3
4 sense = SenseHat()
5 sense.clear(255, 255, 255)
6 sense.low_light = True

```

Faire apparaître des lettres une par une ("OMG!") sur les LED du Sense HAT

```

1 from sense_hat import SenseHat
2 import time
3 sense = SenseHat()
4 sense.show_letter("O",text_colour=[255,0,0])
5 time.sleep(1)
6 sense.show_letter("M",text_colour=[0,0,255])
7 time.sleep(1)
8 sense.show_letter("G",text_colour=[0,0,255])
9 time.sleep(1)
10 sense.show_letter("!",text_colour=[0,0,0])
11 time.sleep(1)
12 sense.clear

```

Faire apparaître un message ("One small step for Pi!") sur les LED du Sense HAT

```

1 from sense_hat import SenseHat
2 sense = SenseHat()
3 sense.show_message("One small step for Pi!",text_colour=[255,0,0])

```

3. Afficher l'humidité grâce au Sense HAT

```

1 from sense_hat import SenseHat
2 sense = SenseHat()
3 humidity = sense.get_humidity()
4 print("Humidity:%s%%rh"%humidity)

```

Afficher la température grâce au Sense HAT

```

1 from sense_hat import SenseHat
2 sense = SenseHat()
3 temp = sense.get_temperature()
4 print("Temperature:%sC"%temp)

```

4. Allumer une ampoule LED à partir du terminal Python du Raspberry Pi

```

1 import RPi.GPIO as GPIO
2 GPIO.setmode(GPIO.BCM)
3 GPIO.setup(18, GPIO.OUT)
4 GPIO.output(18, GPIO.HIGH)

```

Eteindre une ampoule LED à partir du terminal Python du Raspberry Pi

```

1 import RPi.GPIO as GPIO
2 GPIO.setmode(GPIO.BCM)
3 GPIO.setup(18, GPIO.OUT)
4 GPIO.output(18, GPIO.LOW)

```

5. Ouvrir le fichier code à partir du PHP

6. Afficher les valeurs de luminosité toute les secondes

```
1  #!/usr/bin/python
2  import time
3  import RPi.GPIO as GPIO
4
5  GPIO.setmode(GPIO.BOARD)
6
7  def Rctime (RCpin):
8      reading = 0
9      GPIO.setup(11, GPIO.OUT)
10     GPIO.output(11, GPIO.LOW)
11     time.sleep(0.1)
12
13     GPIO.setup(11, GPIO.IN)
14     while (GPIO.input(11) == GPIO.LOW):
15         reading += 1
16     return reading
17
18 while True:
19     print Rctime(11)
```

7. Installation avec la plaque de connection

