

Projet IOT

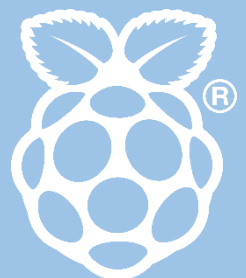
Accéder à la température depuis une page internet

Nathan **Hervouet**

Charles **Huet**

Thomas **Mahe**

Alexis **Vivier**



1

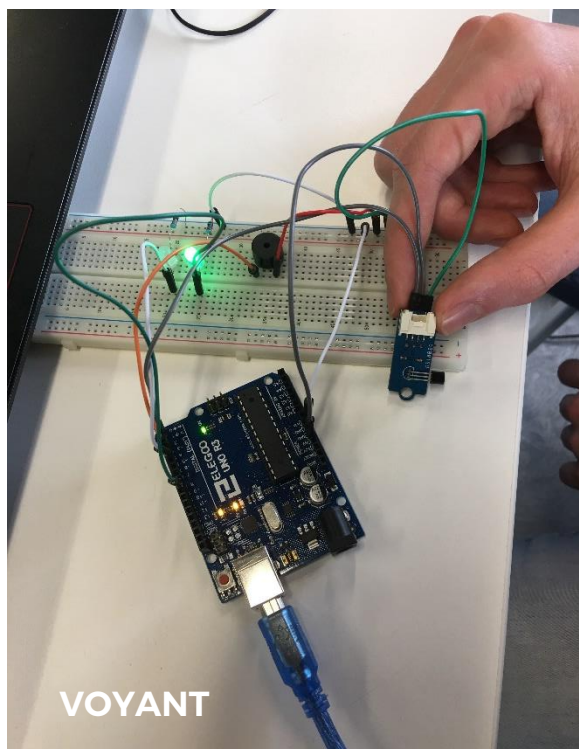
Présentation du projet

Durant ce cours portant sur les objets connectés (IOT) nous avons parlé de tous les composants d'un objet connecté. Le projet qui nous à été proposé était de faire une preuve de concept d'une solution IOT pour récupérer la température d'une salle serveur et de l'afficher sur une interface user « friendly » sur un téléphone portable.

2

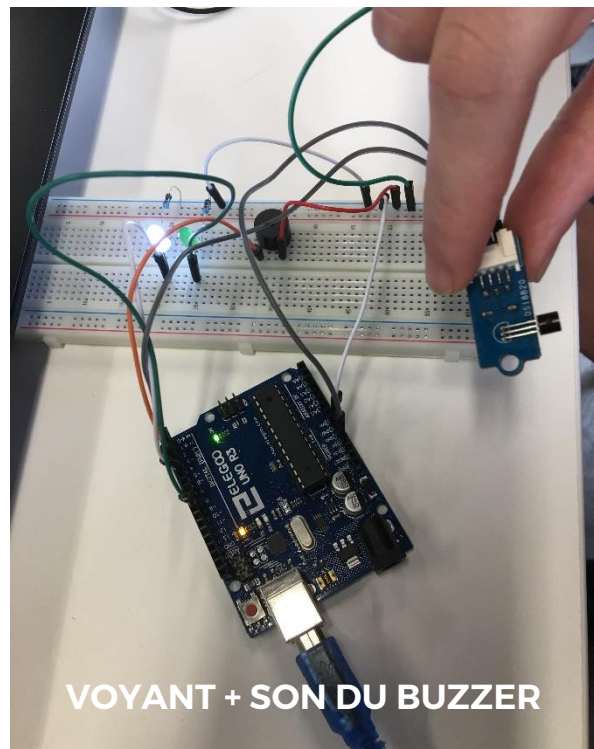
Solution proposée

Notre solution de preuve de concept pour ce projet contient donc plusieurs parties. Tout d'abord nous devons capter la température de la pièce. Pour se faire nous avons utilisé une carte Arduino UNO. Cette carte est un micro Controller qui va nous permettre, en ajoutant un module de mesure de température. Après ce capteur nous avons décidé d'ajouter deux leds, une verte et une blanche qui, avec un simple coup d'œil nous permet de prendre connaissance de la température, voir si elle est supérieure à une valeur que nous pouvons modifier dans le code. L'idée était de commencer à prendre en main Arduino et de s'entraîner sur le côté électronique. Et ce qui va de pair avec les leds nous avons aussi ajouté un buzzer qui, lorsque la température dépasse un certain seuil, le buzzer s'active et donc nous sommes informés que la température commence à être critique.



VOYANT

Température inférieure à 30°C



VOYANT + SON DU BUZZER

Température supérieure à 30°C

Maintenant que la donnée est récupérée il faut donc la traiter et, in fine, afficher la donnée traitée sur un téléphone portable. Donc après l'avoir récupéré il faut l'envoyer sur notre service de base de données qui est, MongoDBLab. Nous avons choisi d'utiliser ce service de base de données car c'est un service qui sert au big data et nous voulons enregistrer beaucoup de données et les traiter pour obtenir des statistiques comme la température moyenne sur une journée. Notre but est, au-delà d'afficher la température, est de faire de la statistique sur les données récupérées.

Nous avons dû choisir entre plusieurs technologies pour faire transiter l'information de l'Arduino à la Raspberry.

- Utiliser un module WIFI
- Utiliser un module Bluetooth
- Utiliser un module RF433 MHz

La première idée n'était pas mauvaise mais la portée du wifi est assez faible et le coup d'équiper toutes nos Arduino en Shield wifi était trop coûteux donc nous ne l'avons pas gardé pour notre solution finale.

La seconde idée n'a pas non plus été retenue car la portée n'était pas assez bonne et, comme la solution utilisant le wifi, équiper toutes les Arduino en Shield Bluetooth serait trop coûteux.

Donc la troisième idée est celle que nous avons décidé de garder car elle est peu coûteuse et plus simple à mettre en place.

Pour envoyer les données sur notre service de base de données nous voulions utiliser un module RF433 Mhz qui permet de lier la Arduino qui récupère la température et l'envoie à la Raspberry PI qui elle, sert de serveur et va envoyer la donnée dans notre base MongoDBLab. Par manque de connaissances sur le matériel RF433 alors nous n'avons pas pu passer par cette technologie sans fil. Pour contourner ce manque de connaissances nous avons décidé d'utiliser une bibliothèque qui nous permet de « discuter » avec la Arduino UNO mais en filaire.

Une fois les données récupérées et traitées nous devons maintenant les envoyer sur la page web qui va permettre d'afficher la température issue du capteur.

3

Estimation du coût

Voici la liste des composants pour la réalisation de notre projet :

Nom du composant	Unité	Prix	Lien
Arduino Uno R3	x1	2.64 €	https://bit.ly/2K0JnjZ

<i>Raspberry Pi 3</i>	x1	30.31 €	https://bit.ly/2HAGgAN
<i>Module Arduino RF 433 Mhz</i>	x1	0.62 €	https://bit.ly/2vrVKIX
<i>Module Buzzer Arduino</i>	x1	0.29 €	https://bit.ly/2HjMykQ
<i>Capteur de température numérique Arduino</i>	x1	0.81 €	https://bit.ly/2vAKdR8
<i>Kit Arduino: Led, resistance, breadboard, cable</i>	x1	1.86 €	https://bit.ly/2w8PuS
Total :		36.53 €	
<i>Abonnement mensuel MongoDB Atlas M2</i>		7.6 €	

(Les composants viennent de chez Aliexpress pour un meilleur tarif)

Seconde estimation pour un cas réel utilisant plusieurs capteurs dédiés à 6 salles différentes :

<i>Nom du composant</i>	Unité	Prix	Lien
<i>Arduino Uno R3</i>	x6	15.84 €	https://bit.ly/2K0JnjZ
<i>Raspberry Pi 3</i>	x1	30.31 €	https://bit.ly/2HAGgAN
<i>Module Arduino RF 433 Mhz</i>	x6	22.32 €	https://bit.ly/2vrVKIX
<i>Module Buzzer Arduino</i>	x6	1.74 €	https://bit.ly/2HjMykQ
<i>Capteur de température numérique Arduino</i>	x6	4.86 €	https://bit.ly/2vAKdR8
<i>Kit Arduino: Led, resistance, breadboard, cable</i>	x1	1.86 €	https://bit.ly/2w8PuS
Total :		76.93 €	
<i>Abonnement mensuel MongoDB Atlas M2</i>		7.6 €	

Estimation de la consommation

Calcul de la puissance annuelle des différents composants :

- Arduino Uno R3 - 0.4 Watt - $0.4 \times 24 \times 365 / 1000 = 3.5$ kWh (x6 cartes)
- Raspberry Pi 3 - 3.5 Watt - $3.5 \times 24 \times 365 / 1000 = 30.66$ kWh

Total : 51.66 kWh (comprends 6 Arduino et 1 Raspberry)

Prix du kWh actuelle (2018) TTC :

- EDF Tarif Bleu (réglementé) - 0.14 830 €

$51.66 \times 0.14830 = \mathbf{7.66 \text{ € à l'année}}$

Estimation du prix du développement et du déploiement de la solution

Site web, liaison MongoDB et mise en place de l'infrastructure :

- Ergo/UX
 - 30€/h avec 16h de travail
- Intégration/Responsive
 - 30€/h avec 20h de travail
- Mise en place MongoDB
 - 30€/h avec 8h de travail
- Mise en place de l'infrastructure
 - 30€/h avec 8h de travail

Coût de maintenance :

- Forfait à l'année pour 30h de travail
 - 900€/an
 - 40€/h pour chaque heure passée en plus

Conclusion :

- Mise en place sans forfait maintenance : 1560€
- Mise en place avec forfait maintenance : 2460€

4

Contraintes rencontrées

Concernant les contraintes rencontrées lors de la réalisation de ce projet elles ont été assez impactante sur la direction de notre projet mais nous avons su rebondir et trouver des solutions qui restaient dans la ligne directrice de notre projet.

La première problématique rencontrée est le manque de connaissances sur l'utilisation du module RF433 MHz. Tant sur le module en lui-même que sur la façon

de l'utiliser avec du Python. Ce manque de connaissances combiné à un manque de temps nous avons dû trouver une solution pour avoir une solution au sujet proposé.

La seconde problématique rencontrée était le choix de la technologie à utiliser. Nous devons utiliser MongoDB pour l'enregistrement des données. Mais lorsque nous devons choisir ce que nous devons utiliser comme langage pour envoyer les données de l'Arduino à la Raspberry nous nous sommes perdus et donc lancé dans un langage sans avoir assez veillé sur toutes les possibilités qui s'ouvraient à nous.

5

Conclusion

Pour conclure sur ce projet nous pouvons dire que notre solution est viable car la chaine qui lie tout de la récolte de l'information à son traitement est fiable. L'utilisation de la technologie RF433Mz est une très bonne façon de gérer le sans fil entre l'Arduino et la Raspberry. Notre choix d'interface qui avait pour but d'être bien lisible sur téléphone s'est donc porté vers un site web responsive pour pouvoir, en plus, toucher un maximum de personnes et de « devices ». Pour finir, avec les compétences requises nous aurions pu finir et faire la preuve de concept de notre solution.