

Ayoub Khazzar

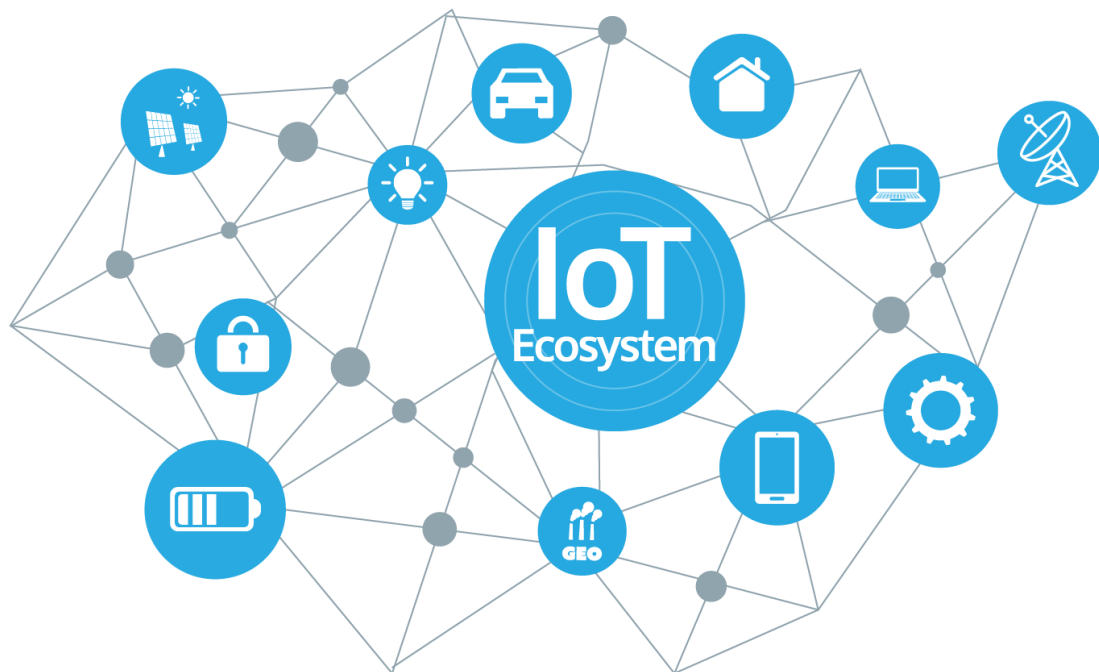
Lucien Burdet

Marouane Azzouz

Nassim Khirredine

Redwan Kara

DECEMBRE 2020



PROJET IOT

4IRC  
CPE LYON

# Détail de la documentation

*Lisez le README.md à la racine du projet en premier.*

*Pour une documentation plus précise, n'hésitez pas à consulter les README.md dans les sous-dossiers et les commentaires de code.*

## Introduction

Le projet d'IOT consiste à mettre en place une application permettant de récupérer les données des capteurs de luminosité et de température. Ces données sont transmises à plusieurs dispositifs formant une chaîne de communication. Pour cela, nous disposons du matériel suivant :

- Smartphone ou machine virtuelle android (application android)
- Ordinateur (passerelle)
- 2 Micro-controlleurs micro:bit
- 1 Écran OLED

## Présentation

Sur les différents composants électroniques, des scripts différents ont été écrits et sont lancés pour qu'ils puissent communiquer entre eux (Fig. 1). Nous avons donc une application Android lancée sur un smartphone Android qui va communiquer avec un serveur qui sert de passerelle (un ordinateur en l'occurrence) grâce à un script en python. Cette communication est sans fil et s'effectue en UDP. La passerelle va communiquer grâce à une connexion câblée avec un des micro:bit grâce à l'UART. Ce micro:bit qui joue le rôle de transfert de données envers et en provenance de la passerelle communique avec un autre micro-controlleur par ondes radio. C'est ce micro-controlleur qui va mesurer des données de température et de luminosité afin de les afficher sur l'écran auquel il est branché selon un certain ordre remonté depuis l'application android. Ce micro-controlleur et écran OLED échangent grâce à un Edge Connector Breakout Board.

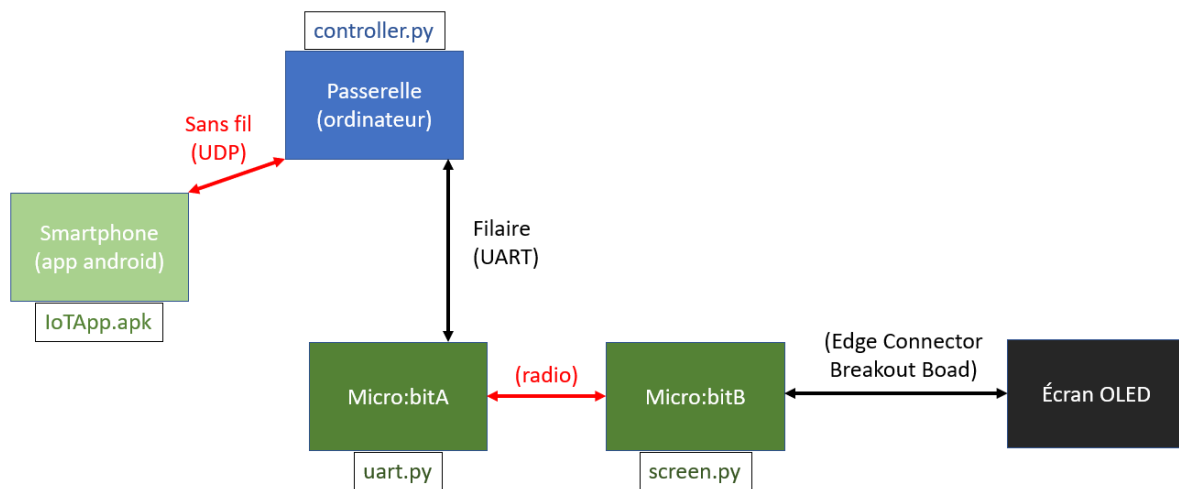


Figure 1 – Schéma global des dispositifs, de leurs liens entre eux et des scripts qui sont utilisés par chacun d'eux

## Choix techniques et activités

Nous avons décidé de travailler uniquement avec le langage python pour la connexion entre les micro-contrôleurs afin de rester cohérent et car nous avons tous plus de facilité dans ce langage.

Pour ce qui est du chiffrement, nous avons utilisé un algorithme de chiffrement de Vigenere (vign.py). Tous les messages envoyés et reçus sont respectivement chiffrés à l'envoi ou réception. Ainsi, un message intercepté n'est pas en clair et ne peut être lu tel quel.

Pour la base de données, nous avons réussi à remplacer le fichier texte par une base de données MongoDB qui est gérée par le serveur (passerelle).

Redwan s'est principalement occupé de la partie serveur du projet. Le code que l'on nous proposait (controller.py) dans le sujet ne fonctionnait pas sur notre environnement et n'effectuait pas exactement ce que l'on voulait. Nous sommes donc parti de ce même fichier et l'avons modifié pour qu'il convienne à notre utilisation. De plus, nous avons mis en place une vraie base de données en MongoDB.

Lucien et Marouane ont travaillé sur l'application Android et la connexion entre les micro:bit. Pour cela nous avons confectionné une interface homme machine simple avec des données à remplir tel que l'adresse du serveur par exemple. Ensuite à l'aide d'un bouton on envoie en UDP l'ordre dans lequel on souhaite récupérer les informations.

Redwan et Marouane se sont occupés de la connexion en UART entre le micro:bit passerelle et le serveur.

Ayoub, Nassim et Lucien se sont occupés de la communication entre les micro-contrôleurs et l'affichage sur l'écran OLED. D'une part nous avons élaboré un protocole de communication simple et fonctionnel. D'autre part nous avons donc dû faire les bons branchements (Fig. 2) et utiliser une librairie externe pour pouvoir afficher des informations à l'écran.

Pour les branchements entre le micro:bit et l'OLED (Fig. 2), nous avons suivi l'exemple donné dans le sujet. C'est ensuite par le biais du script « screen.py » lancé sur le micro:bit relié à l'écran que l'on peut afficher des informations à l'écran.



*Figure 2 – Connexion entre le micro:bit et le Edge Connector Breakout Board*

L'échange de données est bien bidirectionnel. Depuis le smartphone on envoie l'ordre d'affichage au serveur, qui renvoie ces données au micro:bit. Ce micro:bit renvoie ces données à l'autre micro:bit qui s'occupe de mesurer les données, de les afficher sur l'écran OLED et renvoie les mesures dans l'ordre demandé à l'autre micro:bit qui retransmet ces informations à la passerelle puis elles parviennent enfin à l'application.