



PROJET : MISE EN PLACE D'UNE MINI-ARCHITECTURE IoT

ARENSMA Clément
BROSSARD Maxime
MARIN Thomas
PICHARD David

SOMMAIRE

I. Présentation et explication du projet.....	3
II. Mise en Place du Réseau IoT.....	4
A. Mise en place du canal de communication	4
B. Configuration des Capteurs.....	5
C. Affichage sur l'écran OLED	5
III. Configuration de la Passerelle et du Serveur.....	5
IV. Création de l'Application Android.....	6
A. Choix d'Affichage.....	6
B. Définir le Serveur de Destination.....	6
C. Communication Bidirectionnelle avec le Smartphone.....	6
V. Conclusion.....	7

I. Présentation et explication du projet

Le projet s'articule autour de la mise en place d'une architecture IoT, couvrant plusieurs aspects essentiels :

1. Développement d'un Protocole Réseau : Établissement d'une communication bidirectionnelle entre les objets connectés et la passerelle
2. Configuration des Capteurs : Programmation des objets pour collecter des données (température et luminosité) et les afficher sur un écran OLED
3. Configuration de la Passerelle et du Serveur : Mise en place d'un serveur stockant, transmettant les données reçues du microcontrôleur, transmission et réponses des requêtes du client Android
4. Création de l'Application Android : Développement d'une application permettant de contrôler l'ordre d'affichage des données

L'entreprise fictive envisage le déploiement d'objets connectés munis de capteurs de température dans chaque bureau. L'objectif est de collecter ces données et de les afficher de manière configurable via une application Android. Actuellement, les composants nécessaires ne sont pas configurés, laissant à l'équipe la responsabilité de mettre en place l'architecture objet-passerelle-serveur.

La suite du rapport abordera plus en détail les différentes étapes du projet et les choix technologiques effectués.

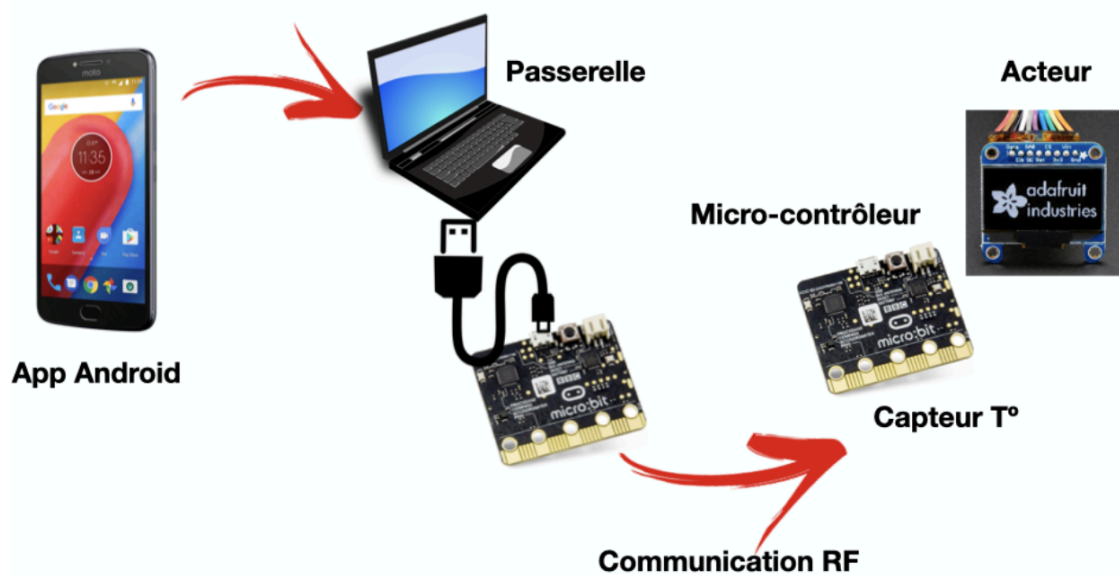


FIGURE 1 – Maquette de l'architecture du mini-projet

Pour mieux concevoir la suite du projet nous avons fait un autre schéma avec les différents protocoles de communication.

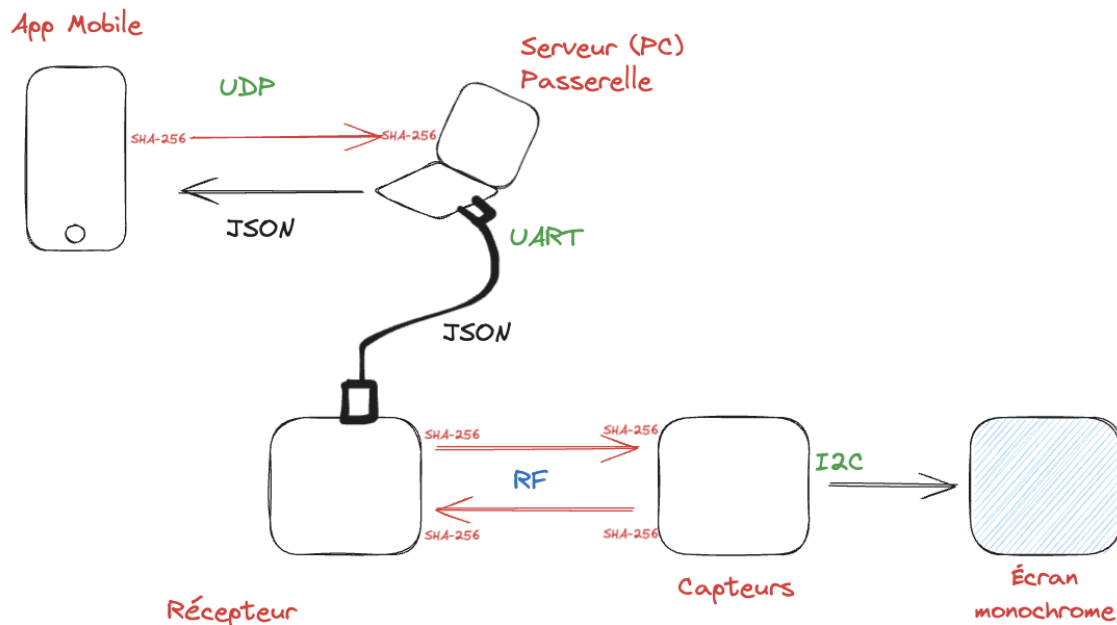


Figure 2 : Schéma de l'architecture avec les différents protocoles de communication.

II. Mise en place du réseau IOT

A. Mise en place du canal de communication

Le but était de pouvoir faire communiquer deux microcontrôleurs de manière bidirectionnelle. Il a très vite été décidé de passer par l'interface radio embarqué du microBit. Le seul inconvénient était qu'il était possible d'avoir des interférences si plusieurs microcontrôleurs émettaient en même temps. Pour remédier à cela, nous avons défini un groupe fréquence partagé par les deux microcontrôleurs.

Pour la communication entre la passerelle et le microcontrôleur, une liaison série USB directe a été imposée.

La donnée reçue par les capteurs est au format JSON de bout en bout. Il est généré dynamiquement et chiffré avant envoi.

Pour le chiffrement, nous avons au départ opté pour un chiffrement avec clé asymétrique, un chiffrement des plus robustes, mais les contraintes de temps et la difficulté à implémenter cette solution nous a forcé à utiliser un chiffrement classique mais suffisant, le chiffrement CAESAR (ou chiffrement par décalage).

B. Configuration des Capteurs

Le capteur sur le microcontrôleur est équipé pour capter la lumière et la température. Toutes les secondes, on actualise notre écran OLED avec les données collectées. Une fois cela fait, on formate la donnée en JSON avant de la chiffrer et de la transmettre à l'autre microcontrôleur par onde radio.

Côté serveur, les données sont récupérées par liaison série directement depuis le microcontrôleur. Il n'a pas été jugé nécessaire de chiffrer cette partie. Suite à cela, la donnée est enregistrée en base de données. Les données côté serveurs s'affichent donc dans l'ordre d'arrivée. Si plusieurs microcontrôleurs envoient des requêtes en même temps, c'est la première requête qui est traitée en premier. On pourrait par la suite imaginer un système de priorité basé sur un champ supplémentaire dans le JSON

C. Affichage sur l'écran OLED

L'affichage sur écran OLED se fait directement par coordonnée de départ (ligne, caractère). Il a fallu quelque temps pour réussir à convertir les données captées en format affichable, mais cela a pu être résolu. L'ordre d'affichage des valeurs sur l'écran OLED est de base le suivant: Luminosité → Température, mais une émission d'une requête précisant l'ordre est possible, ce qui permet à l'utilisateur mobile de configurer l'ordre d'affichage de l'écran à distance. Cette requête est elle aussi en JSON et est elle aussi cryptée de bout en bout.

III. Configuration de la Passerelle et du Serveur

Nous avons choisi d'utiliser un de nos PC personnels en tant que passerelle.

- Configuration du serveur pour stocker les données reçues

Après avoir récupéré le code fourni du serveur sur e-campus. Nous avons cherché à améliorer l'existant qui était déjà fonctionnel pour ajouter des nouvelles features et améliorer l'expérience utilisateur. Nous avons donc choisis de stocker les données reçues par le serveur dans une base de données influxDB démarrée en conteneur sur la même machine.

- Écoute des requêtes du client Android par UDP

Le serveur écoute en continu les requêtes par le client Android, décode le message sous format JSON pour le renvoyer en JSON sur l'UART si c'est un ordre d'affichage. Sinon le serveur va alors lui renvoyer le dernier message si le contenu de la demande est "getValues()".

- Évolutions du serveur

Comme dit précédemment, le stockage des données a été fait avec InfluxDB et une interface web(Grafana) a aussi été mise en place tout deux lancées sous conteneur. Ainsi nous avons la possibilité de voir les données envoyées au serveur sous différents formats(graphique) selon l'heure de soumission et la requête envoyée.

IV. Création de l'Application Android

A. Choix d'Affichage

L'application Android offre une interface complète pour explorer et interagir avec les données provenant de la Micro:bit. Elle se caractérise par un écran unique, simplifiant la connexion au serveur et la visualisation des informations. L'application permet également d'envoyer des instructions à la Micro:bit, déterminant l'ordre d'affichage des données à l'écran.

- Développement de l'interface pour définir l'ordre d'affichage des données

Affichage des données en liste afin d'avoir une historisation des données au fur et à mesure de l'utilisation de l'application

- Interaction avec les données à travers l'écran tactile

Permettre le changement d'ordre d'affichage des valeurs de température et de luminosité.

Figure 3 : WIREFRAME APPLI MOBILE

Température	Luminosité

B. Définir le Serveur de Destination

- Configuration de l'adresse IP et du port dans l'application Android

Permettre le choix à l'utilisateur sur le serveur sur lequel il va se connecter avec un champ IP à renseigner pour s'y connecter

- Communication via le protocole UDP sans ACK

Création de Thread de communication avec le serveur passerelle. Communication via des paquets UDP avec le serveur.

C. Communication Bidirectionnelle avec le Smartphone

- Réception des messages du serveur sur le Smartphone

Création de Thread de réception avec le serveur passerelle. Communication via des paquets UDP avec le serveur.

- Affichage des données dans l'application Android

Récupération des données et affichage de celle-ci. Données arrivent en JSON et sont affichées en texte dans l'application

V. Conclusion

En conclusion, ce projet de mise en place d'une infrastructure IoT a été une expérience enrichissante, mettant en pratique nos compétences acquises tout au long du semestre. Nous avons réussi à relever les défis liés à la communication bidirectionnelle entre les objets, les passerelles, et le serveur, en développant un protocole de communication efficace. La configuration des capteurs et l'affichage des données sur l'écran OLED ont été mis en œuvre avec succès, contribuant à la collecte et à l'affichage précis des informations demandées par le serveur.

La passerelle et le serveur ont été configurés de manière à stocker les données reçues, et des évolutions potentielles, telles que l'utilisation de bases de données avancées ou la création d'une interface web, ont été identifiées pour une gestion plus sophistiquée des données. L'application Android développée a permis de contrôler l'ordre d'affichage des données et de choisir le serveur de destination, facilitant ainsi la personnalisation et le contrôle de l'affichage des informations.

En présentant nos choix technologiques et en effectuant une démonstration de notre implémentation, nous sommes convaincus d'avoir répondu de manière efficace aux besoins de l'entreprise. Ce projet a consolidé nos connaissances pratiques dans le domaine de l'Internet des Objets et a ouvert des perspectives pour des améliorations futures.