

Rapport :

Mise en place d'une mini-architecture IoT

Membres du projet : Linda Mohamed, Julien Gazeilles, Julien Andreoli, Gustave Malachane.

Introduction au projet :

Dans notre projet, on a pour objectif de mettre en place un système de communication sans fil entre deux microcontrôleurs afin de permettre l'affichage de données issues d'un capteur météo. On affichera ces données sur une application Android qui permettra également de contrôler l'ordre de cet affichage. On disposera en plus d'un écran OLED afin de suivre les données affichées, les étapes de la communication entre les deux MicroBit.

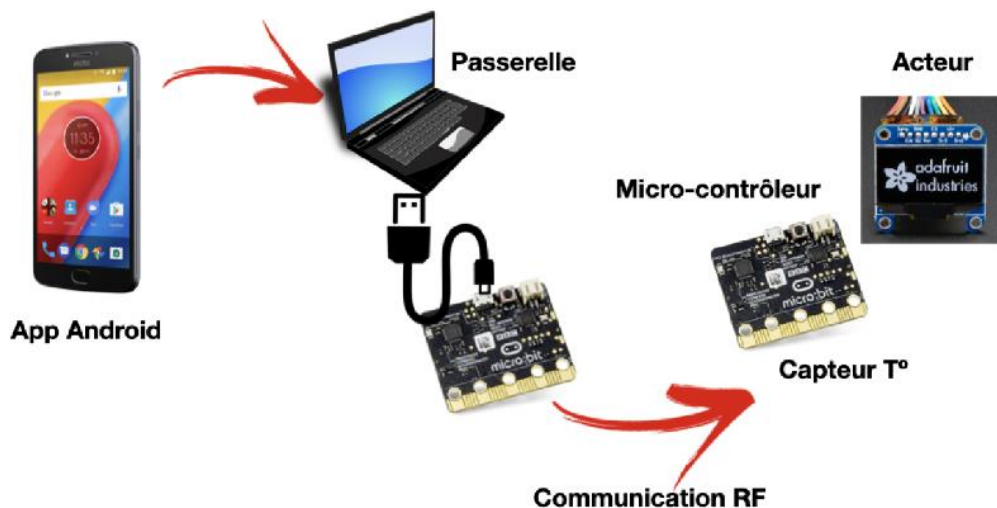


Figure 1 : Schéma du fonctionnement du projet

Fonctionnalités mises en place :

Configuration des capteurs et de l'écran :

Dans notre projet, nous utilisons plusieurs capteurs connectés au micro pour mesurer des données environnementales :

- BME280 : Capteur pour la température, la pression atmosphérique et l'humidité.
- TSL2561 : Capteur pour la luminosité et la lumière infrarouge.
- VEML6070 : Capteur pour mesurer le rayonnement UV.

Ces capteurs sont configurés via le protocole I2C et les données collectées sont affichées sur un écran SSD1306 OLED avant d'être sécurisées pour la transmission via radio.

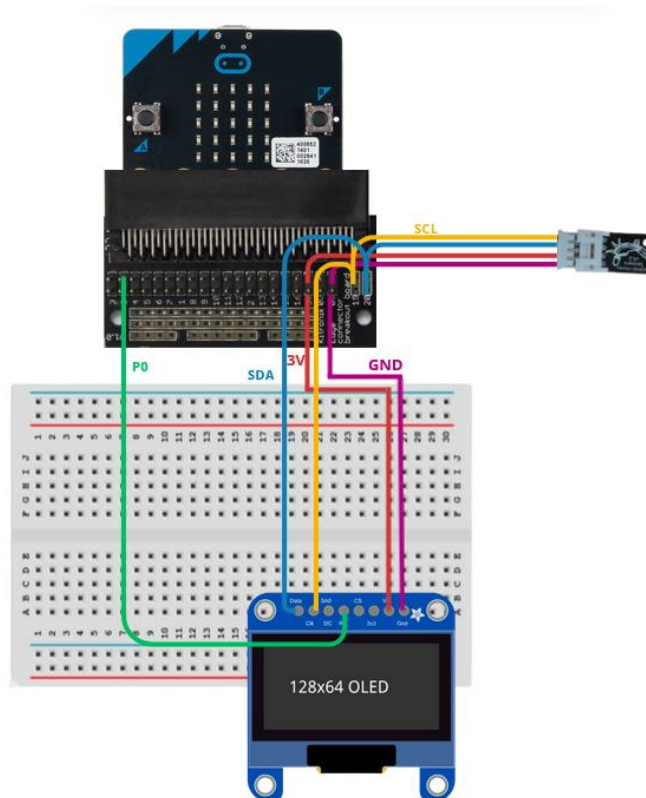


Figure 2 : Schéma du câblage du capteur avec le MicroBit

Communication avec le MicroBit Serveur et le MicroBit Capteur :

On a défini un protocole pour la communication qui transmet une trame composée de :

- OPCODE (1 octet) : Code d'opération pour indiquer le type de commande ou de donnée.
- ID Source (2 octets) : Identifiant de l'émetteur.
- ID Destination (2 octets) : Identifiant du récepteur.
- Data Size (2 octets) : Taille des données contenues.
- Données (X octets) : Informations spécifiques transmises (taille variable).
- Checksum (2 octets) : Code de contrôle pour vérifier l'intégrité des données.

L'initialisation de la communication est réalisée de la manière suivante :

- Le MicroBit capteur (MC) envoie son SerialNumber au MicroBit serveur (MS) pour initialiser la communication.
- Le Microbit Serveur le transmet au serveur qui attribue un ID device à ce capteur.
- Le serveur renvoie ensuite le SN du MC suivi de l'ID et un ordre d'affichage des données du capteur.

Après cette initialisation terminée, le MC envoie des données des capteurs en continu au MS suivant l'ordre qui a été défini.

OPCODE	ID Source	ID Destination	Data Size	Données	Checksum
1 Octet	2 Octets	2 Octets	2 Octets	X Octets	2 Octets

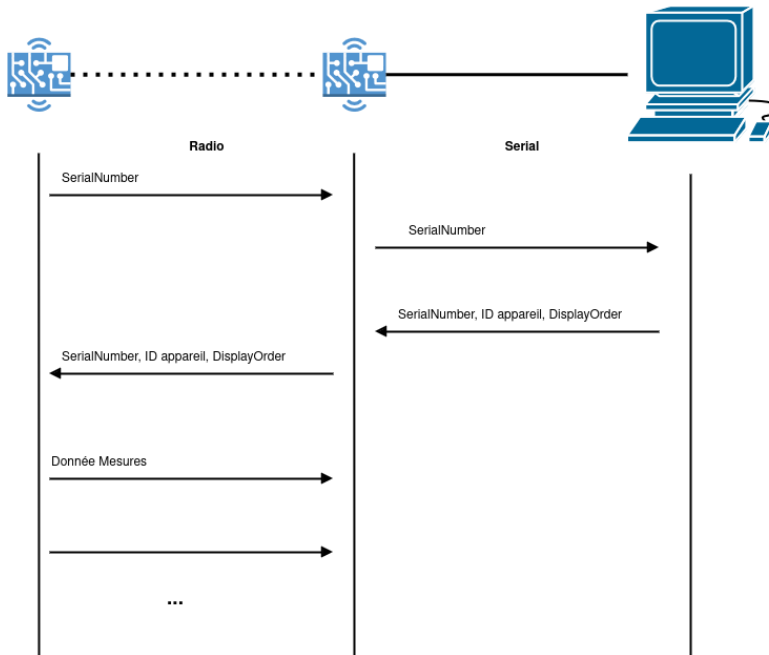


Figure 3 : Schéma du fonctionnement du protocole

Application Android :

Interface Utilisateur

1. Écran de connexion :
 - L'utilisateur entre l'IP et le port du serveur pour se connecter via UDP.
 - Un bouton "Connexion" initie le handshake avec le serveur.
2. Activité principale :
 - Trois onglets disponibles :
 - Accueil : Sélection d'un appareil de mesure et affichage des données (température, luminosité, humidité, pression).
 - Ordre d'affichage : Personnalisation de l'ordre des mesures pour un appareil spécifique avec un bouton "Appliquer".
 - Historique : (Non implémenté) Prévu pour afficher l'historique des mesures d'un appareil.

Architecture de l'Application

1. Vues :
 - Deux activités, dont une principale avec trois fragments pour naviguer entre les onglets.
2. Contrôleurs :

- CommandHandler : Formate les commandes avant l'envoi au serveur.
- ResponseHandler : Traite les réponses du serveur et stocke les données.

3. Services :

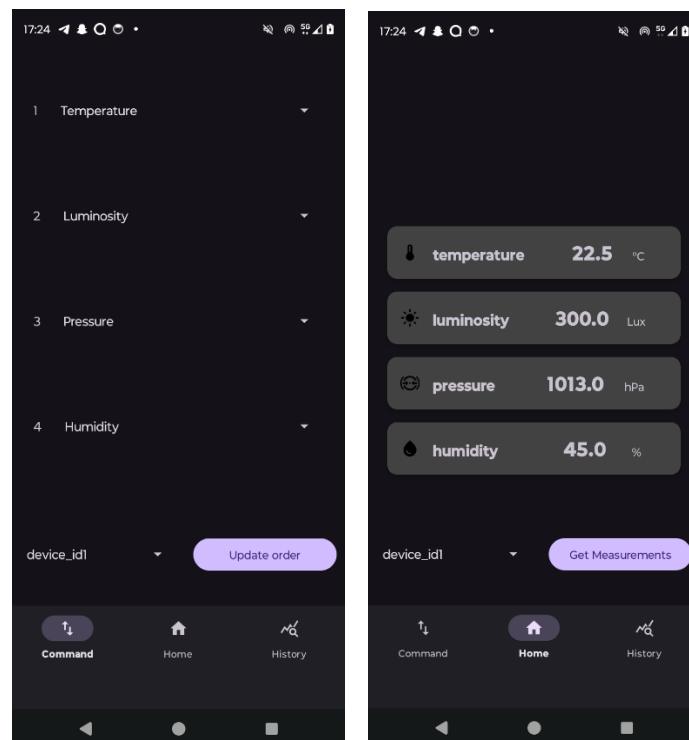
- UdpService : Gère les communications UDP via des threads pour envoyer/recevoir des données de manière asynchrone.

4. Repositories :

- MeasurementRepository : Stocke temporairement les mesures pour un affichage rapide.

Communication

- Toutes les interactions avec le serveur sont effectuées via UDP en utilisant un format JSON pour la sérialisation des données.



Base de données :

On a mis en place une base de données en SQLite pour stocker les données reçues par les capteurs. L'application Android envoie une commande au serveur qui lui renvoie les données depuis la base de données qui est mise à jour continuellement par le serveur via le MicroBit serveur.