

# Projet IOT

**BIOTTE Nathan**  
**GUINGAND Jules**  
**GUICHARD Jonathan**  
**PAUTRAT Olivier**

## Répartition des rôles

BIOTTE Nathan :	Application Android
GUINGAND Jules :	Application Android / Plateforme
GUICHARD Jonathan :	Plateforme / Micro:bit
PAUTRAT Olivier :	Micro:bit

## Sommaire

I.	<a href="#"><u>Présentation du projet</u></a>	P. 4
II.	<a href="#"><u>L'application Android</u></a>	P. 5
	1. Explication des choix techniques	
	2. Axes d'amélioration	
III.	<a href="#"><u>La plateforme</u></a>	P. 6
	1. Explication des choix techniques	
IV.	<a href="#"><u>La Micro:bit</u></a>	P. 7
	1. Explication des choix techniques	
	2. Définition du protocole	
V.	<a href="#"><u>Conclusion</u></a>	P. 8
VI.	<a href="#"><u>Bibliographie</u></a>	P.9

## I. Présentation du projet

Durant cette première partie de l'année, nous avons mis en place un système permettant de recueillir des informations à l'aide des capteurs d'une carte Micro:bit. Il nous est possible de récupérer la température et la luminosité d'une pièce à distance à l'aide d'un serveur mais aussi d'une application Android.

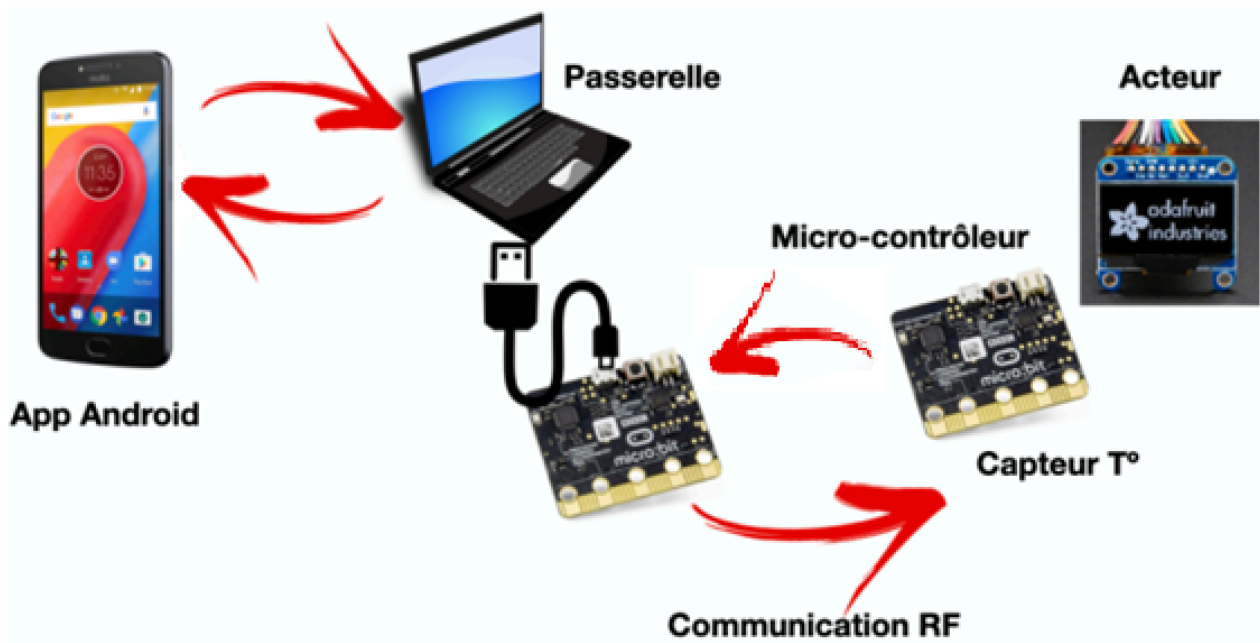


Figure 1: Maquette de l'architecture

### Cahier des charges :

- Créer une application Android spécifique.
- Utiliser un ensemble de capteurs permettant de récupérer des informations sur l'environnement (température, luminosité, etc.)
- Afficher les valeurs des capteurs sur l'écran de la Micro:bit
- Permettre de saisir l'ordre d'affichage des données sur la Micro:bit via l'application Android
- Utiliser un protocole pour la communication Android-serveur
- Définir un protocole sécurisé pour la transmission de données entre la Micro:bit et le serveur

## II. L'application Android

Dans notre système, l'application Android permet de prendre connaissance des différentes données envoyées depuis la Micro:bit mais aussi d'interagir avec celle-ci et la piloter.

Notre application se compose d'un seul et unique écran permettant la connexion au serveur mais aussi l'affichage des données. Elle permet aussi d'envoyer à la Micro:bit l'ordre d'affichage des différentes données qui apparaissent sur l'écran. Il est aussi possible de sélectionner la pièce à afficher.

### 1. Explication des choix techniques

Dans cette partie, nous allons vous présenter les différents choix techniques que nous avons faits en ce qui concerne l'application Android.

Choix	Justifications
Une seule Activity	<ul style="list-style-type: none"><li>• Simplifier le code</li><li>• Pas de transfert de données entre activités</li></ul>
Un seul écran	<ul style="list-style-type: none"><li>• Simplifier le code</li></ul>
Utilisation de Retrofit	<ul style="list-style-type: none"><li>• Simplifier la communication HTTP</li></ul>
Récupération dynamique des pièces	<ul style="list-style-type: none"><li>• Permet d'avoir une solution modulable :<ul style="list-style-type: none"><li>○ L'ajout de pièces n'impactera pas l'application.</li></ul></li></ul>

### 2. Axes d'amélioration

Suite à la fin de notre projet, nous avons quelques axes d'amélioration pour notre application :

Possibilités	Intérêts
Utilisation de Fragment	<ul style="list-style-type: none"><li>• Modularisation de l'application au niveau des supports</li><li>• Permet de réutiliser une partie de l'application</li></ul>
Récupération dynamique de la liste des capteurs	<ul style="list-style-type: none"><li>• Application plus modulable :<ul style="list-style-type: none"><li>○ Rajout de capteur possible</li></ul></li></ul>

### III. La plateforme

Les capteurs / microcontrôleurs que nous utilisons dans le projet ne permettent pas d'être interconnectés facilement (couche TCP/IP non implémentée). Pour répondre à cette problématique, il a été choisi de mettre en place une plateforme assurant la communication entre les capteurs d'une part et les autres entités du projet (base de données, application mobile, outils de visualisation).

#### 1. Explication des choix techniques

Fonction	Solution apportée	Technologies utilisées	Avantages	Inconvénients
<b>Enregistrer les données de manière persistante</b>	Base de données SQLite	<ul style="list-style-type: none"><li>• Python</li><li>• SQL</li></ul>	Bonne scalabilité	N'est pas compatible avec Graphana
<b>Communication avec les capteurs</b>	Une Micro:bit reliée au PC faisant office de plateforme par une connexion série	PC : <ul style="list-style-type: none"><li>• Python</li></ul> Micro:bit : <ul style="list-style-type: none"><li>• JavaScript</li><li>• MakeCodeBlock</li></ul>	Liaison fiable et sécurisée (il faut avoir accès au matériel pour pouvoir interférer)	Impossibilité d'utiliser la liaison série pour d'autres usages (ex : debugger)
<b>Communication avec l'application Android</b>	Exposition d'une API renvoyant des données issues de la base de données au format JSON	Serveur web : <ul style="list-style-type: none"><li>• Flask</li></ul> Connecteur SGBD : <ul style="list-style-type: none"><li>• Peewee</li></ul>	Le JSON est un standard très utilisé : échanges facilités avec d'autres applications	/
<b>Affichage synthétique des relevés des capteurs</b>	Création d'une application web récupérant les données par l'API exposée par la plateforme	Application web : <ul style="list-style-type: none"><li>• ReactJS</li><li>• TypeScript</li></ul>	Personnalisation illimitée de l'application	L'évolution de l'application est à notre initiative (contrairement à une solution faite par un tier)

## IV. La Micro:bit

Le but de cette partie est d'obtenir des données à partir de capteurs embarqués, de les afficher grâce à un écran OLED placé à proximité des capteurs et de les envoyer au serveur qui s'occupe de les stocker et de les transmettre. Cette partie est donc entièrement prise en charge par une Micro:bit, qui possède un capteur de température et un capteur de luminosité faisant office de producteurs de données, ainsi qu'un module RF (Radio-Fréquence) permettant la communication avec la passerelle. Pour ce qui est de l'affichage, un écran OLED SSD1306 est relié à la Micro:bit par le biais d'une connexion I<sup>2</sup>C.

### 1. Explication des choix techniques

Problèmes rencontrés	Solution mise en place
Envoi de données en RF limité à 19 caractères par message	Conception et implémentation d'un protocole de transport permettant la segmentation et l'encapsulation d'une donnée en plusieurs paquets
Manque de sécurité au niveau des transmissions RF	Implémentation d'un algorithme de chiffrement simple (XOR avec clé)
Absence de fonction de transfert d'un caractère ASCII vers sa représentation décimale	Utilisation de la fonction de transfert d'entier décimal vers caractère ASCII pour créer une table de correspondance ; l'index d'un caractère dans cette table est alors sa représentation décimale

### 2. Définition du protocole

Lors de l'envoi, le programme scinde en blocs de 11 octets la donnée à envoyer afin de pouvoir la transférer en plusieurs fois.

Avant l'envoi du premier paquet, un paquet de connexion est envoyé afin de préciser les métadonnées de l'échange (id de connexion, nombre de paquets, checksum).

Lors de la réception d'un paquet de données, un message d'acknowledgement est envoyé à l'émetteur du message. À la fin de l'échange, les paquets pour lesquels il n'y a pas eu message d'acknowledgement seront renvoyés après un bref délai.

Des compléments d'informations sont disponibles dans le code (spécificités des Micro:bit, etc.).

## V. Conclusion

Suite à ce projet, nous avons pu mettre en place toutes les connaissances acquises durant le module d'IOT. Il nous a permis de mettre en relation tous les acteurs sur lesquels nous avons travaillé durant ce début d'année.

Il nous a aussi permis d'approfondir nos connaissances, aussi bien côté Android avec l'intégration d'un nouveau formalisme de communication (JSON), mais aussi côté Micro:bit avec la création d'un nouveau protocole sécurisé afin de communiquer avec la plateforme.

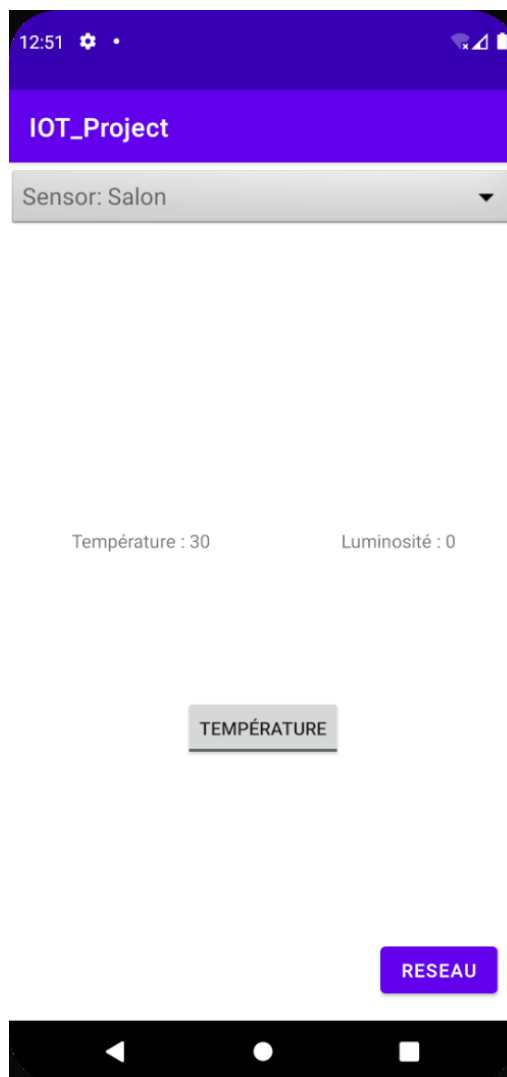


Figure 2: Application Android



## VI. Bibliographie

- Retrofit : <https://square.github.io/retrofit>
- OpenClassroom : <https://openclassrooms.com/fr/courses/4568596-construisez-une-interface-utilisateur-flexible-et-adaptative>
- MakeCode : <https://makecode.microbit.org> IDE web permettant de coder sur les Micro:bit