Rapport de projet - Mise en place d’une mini-architecture IoT

# L’équipe projet

Nous avons réalisé ce projet à 4, en affectant à chacun un rôle. Cependant, lors de difficultés nous échangions pour savoir comment le corriger. La liste suivante décrit les rôles principaux de chacun des membres de l’équipe, mais nous nous sommes entraidés tout au long du projet.

L'équipe était composée de :

* Alexis THOMAS - Module Android
* Alexis LONCHAMBON - Module Serveur
* Raphaël LARUE - Module Micro:bit Passerelle
* Mathis GIRARDIN - Module Micro:bit Capteurs

## Alexis Thomas - Module Android

Alexis Thomas était responsable du déploiement de l’application Android en Java et de la communication Mobile-Serveur.

Il a développé une communication bidirectionnelle entre le smartphone et le serveur, ainsi qu'une interface graphique permettant à la fois d'afficher les données des capteurs, et de paramétrer l'ordre d'affichage.

On a un premier Thread d'envoi s'assurant exclusivement d'envoyer une éventuelle nouvelle configuration d'ordre d'affichage à travers un canal UDP vers le serveur.

Un second Thread d'envoi gère la demande des données des capteurs à intervalles réguliers au serveur via le canal UDP, qui va répondre par ce même canal UDP en envoyant les données demandées. Ces données seront réceptionnées par le socket UDP au sein d'un dernier Thread de réception, attendant les données envoyées par le serveur, données qui seront ensuite affichées sur l'application mobile.

## Alexis Lonchambon - Module Serveur

Alexis Lonchambon était responsable du déploiement du serveur et de la communication Mobile-Serveur.

Après des difficultés avec le serveur python fourni en énoncé, nous avons décidé de partir sur un serveur .NET. Étant donné qu’Alexis Lonchambon maîtrise mieux le langage C#.

Le serveur communique avec la passerelle sur un port serial pour recevoir les données provenant des capteurs et fournir le format demandé par l’application mobile à la passerelle.

Le serveur C# gère les différentes parties du code sous forme de services :

* Un premier service se charge de la communication UART après un court paramétrage du port et des options. La réception de données se fait par un événement permettant la lecture d’une ligne directement depuis le buffer.  
  L’envoi de données se fait par une simple fonction WriteLine
* Un second service se charge de la communication UDP avec l’application Android. Plusieurs fonctions sont disponibles mais seules les méthodes Receive et Respond sont utilisées.  
  Le service fonctionne avec un système de callback asynchrone ou une fonction est exécutée à la réception d’un message.
* Le service de Logs qui formate l’output des différents services pour l’écrire dans la console. Ce service peut être facilement modifié pour y intégrer du logging sur une Base de Données.   
  La Gateway écrit régulièrement des messages sur le port Serial afin de garantir le bon fonctionnement.

Les données de la gateway sont enregistrées sous un format propriétaire et sont converties en JSON à l'aide d’un parseur Objet ←→ JSON. Ce même parseur est utilisé pour la communication par l’application Android.

Bien que testé sur une variété de systèmes dont Windows, Raspberry Pi, Debian et une VM Ubuntu, il est possible que le serveur ne fonctionne pas sur certains systèmes à cause de problèmes d'accès au port UART

Les deux Alexis ont travaillé main dans la main pour faire communiquer l’application mobile et le serveur.

## Raphaël Larue - Module Micro:bit Passerelle

Raphaël Larue était responsable du déploiement de la passerelle, de la communication entre les modules micro:bits et des rendus.

La passerelle est une micro:bit codée en C++ servant d’interface entre le serveur et les capteurs. Elle transfère les informations reçues du serveur vers la micro:bit servant de capteur à travers un canal de communication RF et envoie le format reçu par le serveur.

Les communications entre les micro:bits sont cryptées en ASCII-shift d’après une clé privée connue uniquement par les 2 micro:bits. Ce cryptage n’est pas le plus sécurisé mais suffit étant donné la non-criticité des données envoyées.

La communication entre le serveur et la gateway, assurée par la liaison serial était l’une des phases les plus complexes du projet. La grande difficulté venait du manque de documentation et d’exemples. Une grande partie de la partie UART / Serial a été gérée par Alexis Lonchambon, venu en support.

Plutôt que d’utiliser la lecture du port Série avec une boucle, nous avons opté pour un fonctionnement par Event : un événement est déclenché à la réception d’une ligne sur le buffer UART.

## Mathis Girardin - Module Micro:bit Capteurs

Mathis Girardin était responsable du micro-controlleur servant de capteur et de l’écran OLED.

La micro:bit capteur reçoit un format depuis la micro:bit passerelle. Ce format est décrypté puis enregistré dans la mémoire de la micro:bit. Le format est utilisé pour savoir quelles données doivent s’afficher sur l’écran OLED.

La micro:bit récupère les différentes données en provenance des capteurs puis les envoie à la passerelle à travers un canal RF. Elle affiche également les données sur l’écran OLED en fonction du format.

Son rôle était aussi de connecter le capteur météo et l’écran OLED à la micro:bit via la platine d’essai et les câbles fournis.

# La réalisation

Pour plus de renseignements sur la réalisation, la mise en place du projet et les problèmes rencontrés et solutions trouvées, un README.md est disponible dans chacun des modules. Il a été rédigé en anglais, tout comme les commentaires au sein du code.