

---

CPE Lyon - 4IRC - S7 - Année 2019/20  
Architecture et Protocoles Réseaux pour l'IoT  
**Projet - Mise en place d'une infrastructure IoT**

*Oscar Carrillo - Jonathan Tournier*

---

Ce projet a pour objectif d'utiliser les différentes compétences que vous avez acquises pendant les TPs pour vous permettre de mettre en place une architecture de l'internet des objets, telles que :

1. Développer un protocole réseau pour établir une communication entre un objet et une passerelle.
2. Programmer un micro-contrôleur afin que ce dernier puisse collecter des données et les envoyer sous un format précis, ainsi que de recevoir des données de la passerelle afin de les afficher sur un écran OLED.
3. Programmer une passerelle et serveur afin que ce dernier puisse recevoir les données d'un client, les manipuler, les stocker et les envoyer au client suivant en format précis. (Optionnel)
4. Développer une application Android capable de se connecter à un serveur pour demander des données et les afficher, ainsi que renvoyer des configurations à un objet connecté.

## Organisation

Ce projet est à réaliser par équipes de 4 ou 5 élèves.

## Introduction

Votre entreprise souhaite installer des objets connectés avec capteurs de température, d'humidité et de luminosité au sein de plusieurs bureaux afin de pouvoir relever et afficher chacune des données pour chaque bureau possédant un de ces objets. De plus, l'affichage des données pourra être défini via une application Android communiquant directement avec le serveur. Cependant, les objets qui collectent les informations, et le serveur qui stockent et manipule les données ne sont pas configurés. C'est donc à vous de réaliser l'infrastructure objet-passerelle-serveur entre votre objet et le serveur qui vous sont fournis. De même, aucune application mobile pour l'affichage et configuration des objets n'existe actuellement, et vous devrez donc la créer pour réaliser les besoins de votre entreprise.

La figure 1 montre un schéma de l'architecture que vous allez mettre en place pour la configuration de votre infrastructure. Ici un smartphone Android sera connecté via WiFi avec un serveur-passerelle Raspberry Pi 3, la raspberry fera la fonction de passerelle entre votre application Android et le micro-contrôleur qui enverra et recevra des données via RF Sub 1GHz. Pour finir vous aurez un des objets à déployer dans les bureaux de la société, il est composé d'un micro-contrôleur qui va gérer le capteur *Sensors* et un écran OLED (acteur) pour l'affichage des informations.

Pour valoriser la mise en place de cette infrastructure, à la fin de ce mini-projet le ou les groupes avec les meilleures infrastructures seront invités à présenter leurs travaux lors du Salon de l'Internet des Objets (SIDO), qui se tiendra les 12 et 13 mai 2020 à Lyon, Cité Internationale.

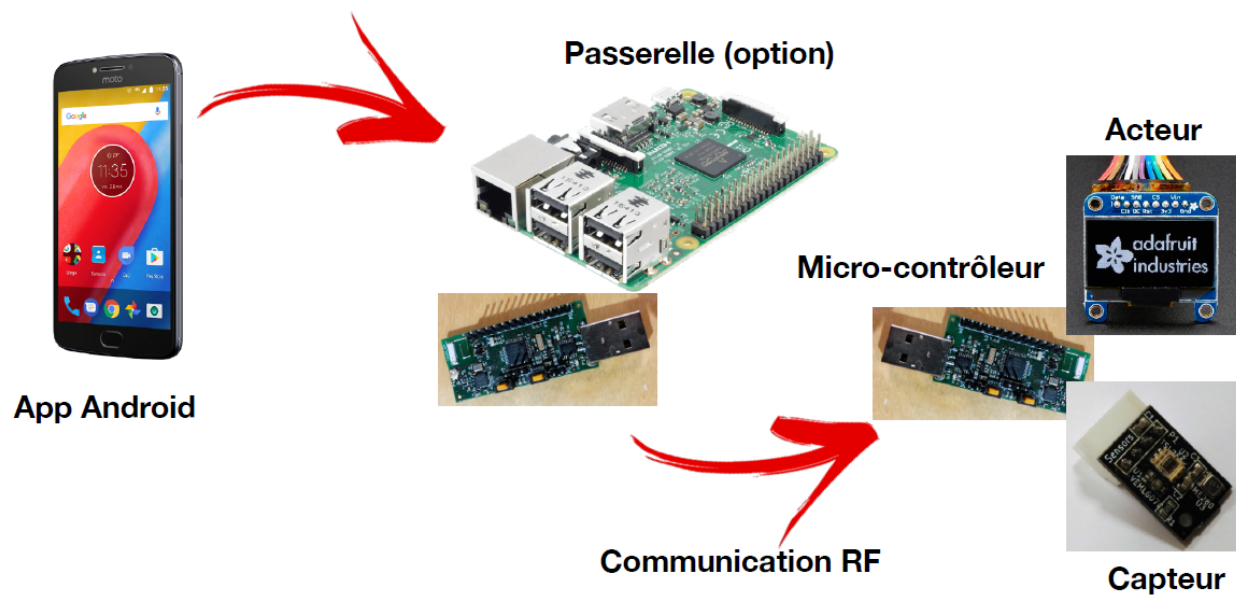


FIGURE 1 – Maquette de l'architecture du mini-projet

## Mise en place d'une infrastructure objet-passerelle

### Exercice 1 : Mise en place du réseau

La première étape de ce projet est le réseau et consiste à mettre en place un canal de communication entre votre objet et votre passerelle afin que ces deux puissent communiquer et échanger des messages entre eux. Les communications sont bidirectionnelles, c'est à dire, que l'objet comme la passerelle sont en mesure de recevoir et d'envoyer des messages de l'un à l'autre.

Vous devez définir un protocole simple pour envoyer les messages entre les deux modules micro-contrôleurs, *pensez aussi à la sécurité des données envoyées*.

### Exercice 2 : configuration des capteurs

Une étape importante de la mise en place de l'infrastructure côté objet consiste à configurer l'objet à déployer dans les bureaux, afin que ce dernier puisse collecter les informations souhaitées (température, humidité et luminosité), les envoyer au serveur (passerelle) et enfin afficher les informations dans l'ordre demandé par le serveur. Pour ce faire, vous disposez d'un micro-contrôleur de type **RF Sub-1GHz de Techno-Innov** équipé du module *Sensors* afin de pouvoir collecter toutes les informations demandées. De plus, l'objet dispose d'un afficheur OLED sur lequel les données pourront être affichées.

### Collecte d'information

Cette étape consiste à vous familiariser avec le module *Sensors* afin de pouvoir collecter les données souhaitées. Pour ce faire, vous disposez du code contrôleur sur le git suivant : <https://github.com/CPELyon/modules-techno-innov>. Le code est dans le sous-dossier `/apps/rf_sub1G/sensors/`.

Le module *sensors* a une interface I<sup>2</sup>C pour pouvoir accéder à ses données, la distribution des pins est présenté dans la figure 2.

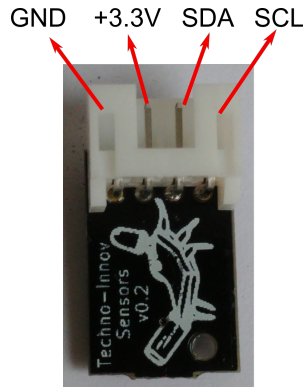


FIGURE 2 – Disposition des pins du module *sensors*

### Affichage sur l'écran OLED

La deuxième partie de la configuration de l'objet est l'affichage des données sur l'écran OLED qui accompagne le module RF. Avant d'afficher directement les informations dans l'ordre défini par le serveur, assurez vous de comprendre comment afficher des informations sur cet écran. Ainsi, pour commencer, vous pouvez programmer votre module RF afin que ce dernier affiche les données récupérées via le module Sensors. Pour ce faire, vous disposez du code contrôleur d'un écran OLED Adafruit sur le git : <https://github.com/CPELyon/modules-techno-innov>. Le code est dans le sous-dossier /apps/rf\_sub1G/oled/ est vous devez l'adapter selon les connecteurs de votre micro-contrôleur.

Dans le code donné, on utilise l'interface I<sup>2</sup>C pour pouvoir envoyer les informations à afficher, on positionne les données en indiquant la ligne à utiliser, la distribution des pins dans la carte est présenté dans la figure 3.

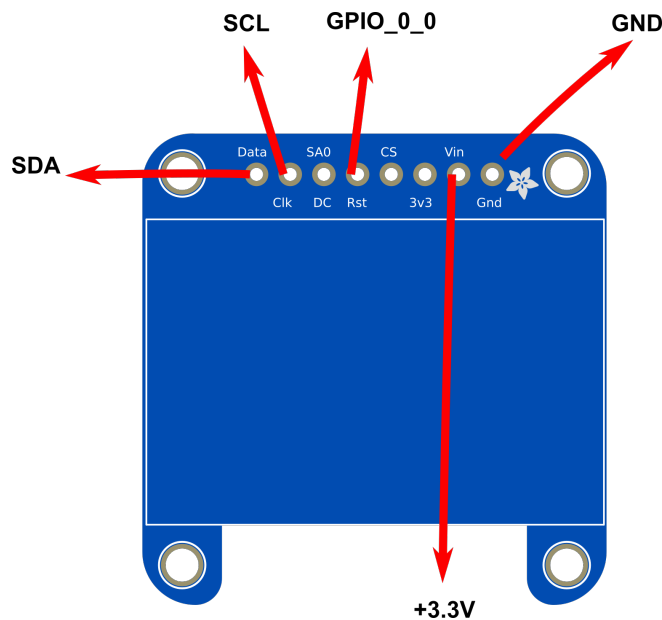


FIGURE 3 – Disposition des pins du module *OLED*

## Communications avec la passerelle

Cette étape englobe deux rôles. En effet, il faut que votre objet soit capable d'envoyer les informations récoltées avec le module Sensors au serveur, mais qu'il soit également capable d'écouter les messages envoyés par le serveur afin de pouvoir afficher le contenu des messages sur l'écran OLED.

Le format des messages à envoyer au serveur est libre, ainsi les données peuvent être envoyées brutes (sans traitement côté objet), ou bien suivant un format précis (type fichier de configuration format JSON par exemple). Dans un premier temps, n'utilisez aucun format et envoyez les données brutes au serveur, le choix d'un format est optionnel.

Le micro-contrôleur attaché à la raspberry fera le rôle de passerelle et recevra les données de la part des capteurs déployées mais sera aussi en mesure d'envoyer la configuration d'affichage à chaque client. Les données de configuration seront reçues à travers son interface USB (UART), indiquant l'ordre d'affichage des données par 3 lettres majuscules, ainsi :

- **TLH** : Indique que la **T**empérature sera affiché en premier, puis la **L**uminosité et enfin l'**H**umidité.

Pour ce faire, sur le git : <https://github.com/CPELyon/modules-techno-innov> vous avez un code d'exemple qui vous permettra de démarrer votre application. Le code est dans le dossier /apps/rf\_sub1G/rf\_isp/ est vous devez l'adapter aux messages que vous allez échanger. Dans ce code, 2 micro-contrôleurs sont liés par ses adresses réseaux (0x01 et 0x02) et à travers le bouton ISP le chenillard de la LED status de l'autre micro-contrôleur est activé ou éteint. Pendant votre implémentation de protocole, *n'oubliez pas de bien identifier chaque module avec une adresse réseau*, mettez-vous d'accord avec vos collègues des autres équipes pour ne pas avoir les mêmes adresses, ça évitera de recevoir des messages indésirables.

### Exercice 3 : configuration du serveur (*optionnel - pour aller plus loin*)

Le serveur est, dans un premier temps, configuré pour stocker les données reçues au format brut dans un fichier texte. Puis ce dernier envoie les données au client Android sous le même format envoyé depuis l'objet. Cette réponse est renvoyée une fois le client Android envoie par UDP le message "getValues()".

L'objectif de cet exercice est de faire évoluer le serveur, plusieurs tâches sont possibles :

- Remplacer le fichier texte par une vraie structure de base de données telle que MongoDB ou SQLite, afin d'avoir une meilleure gestion des données côté serveur
- Définir un format spécifique pour l'échange des données avec le client Android et le micro-contrôleur, comme par exemple un fichier de configuration au format JSON. Cependant, le choix du format reste libre
- Définir une interface web pour la visualisation des données reçues, par exemple Grafana
- Donner la possibilité de gérer plusieurs objets. Dans l'état actuel le serveur reçoit des données brutes sans faire un filtrage et de ce fait, il n'est pas possible de gérer plus d'un objet, vous pouvez donc implémenter un protocole pour pouvoir gérer plusieurs objets au même temps et les gérer depuis l'interface web et/ou l'application Android.
- ...

Pour pouvoir configurer les raspberry vous allez avoir besoin des identifiants suivants :

- login : pi
- password : raspberrycpe

## Création de l'application Android

La dernière partie de la mise en place de l'infrastructure IoT consiste à développer une application Android qui permet de contrôler l'ordre d'affichage des données collectées sur un des objets en particulier. Ainsi, l'application a deux fonctionnalités, le choix de l'affichage des données d'un des objets et le choix du serveur sur lequel cela s'applique. Pour simplifier, on assume qu'il y a seulement un objet associé à chaque serveur et donc on doit choisir seulement le serveur de destination et pas un des objets qui lui sont associés.

Pour faciliter le développement de votre application, les données seront envoyées via le protocole UDP, et votre Smartphone sera connecté au serveur via WiFi. Les identifiants pour se connecter au WiFi mis à disposition pour le développement de ce mini-projet :

- SSID : 4irc-aiot-2
- password : 4irc-aiot-pwd

### Exercice 1 : choix d'affichage

Dans un premier temps votre application devra être en mesure de définir un ordre d'affichage pour les 3 différentes données collectées. Vous pouvez, pour ceci, afficher les trois données à l'écran et par simple pression du doigt définir l'ordre d'affichage des données. *Ceci n'est qu'un exemple, et le choix d'interface visuel de votre application reste libre.*

### Exercice 2 : définir le serveur de destination

En plus de choisir l'ordre d'affichage des données, votre application doit être en mesure de choisir le serveur de destination. En effet, le but est de pouvoir contrôler l'affichage des données pour chaque objet via le serveur avec lequel il communique. Ainsi, le choix du serveur dans l'application est indispensable. La configuration du serveur dans l'application Android se fait via l'adresse IP du serveur et son port d'écoute (par défaut : 10000). Donc dans votre application Android, vous devez disposer de deux champs dans lequel il sera possible de saisir une adresse IP et un port, qui permettront la communication avec le serveur souhaité.

De plus, la communication sera effectuée via le protocole UDP, et aucun ACK n'est demandé. Ainsi, votre application doit seulement faire de l'émission en direction du serveur, sans se préoccuper de devoir réceptionner des paquets. Les données envoyées par votre application seront les 3 lettres majuscules indiquant l'ordre d'affichage dans l'écran OLED.

### Exercice 3 : Communication bidirectionnelle avec le Smartphone

Dans l'objectif d'être capable d'afficher les données également sur votre Smartphone, votre application doit être en mesure de recevoir des messages venant du serveur. Tout comme le serveur envoie des données à l'objet, ce dernier devra pouvoir envoyer les mêmes données avec le même format à votre Smartphone. Ce dernier devra être capable de réceptionner les données et de les afficher dans l'application créée précédemment.

De plus, le Smartphone ne doit réceptionner que les données émises depuis le serveur avec lequel il est connecté (défini dans l'exercice précédent).

## Rendu final

L'évaluation du mini-projet sera faite par une démonstration de votre implémentation d'une durée de 10 minutes maximum. De même, un rendu sur la plateforme e-campus devra être fait au plus tard le 28 novembre à 18h. Le rendu doit contenir :

1. Un rapport très synthétique listant les membres de l'équipe et décrivant les activités réalisées (au moins 2 pages)
2. L'application Android *documenté* créée pour le contrôle et accès à votre infrastructure de l'internet des objets
3. Votre projet C *documenté* créé pour gérer la capture et affichage des données dans l'objet
4. Votre projet C *documenté* créé pour le micro-contrôleur faisant le rôle de passerelle
5. (*Optionnel*) L'application coté serveur *documenté*, si vous avez créé une nouvelle application pour le serveur