

GERVAIL Antoine
ZIAN-CHERIF Amine

M2P GI

Galileo

[Introduction](#)

[Architecture](#)

[Galileo](#)

[PC](#)

[Logiciel](#)

[Galileo](#)

[PC](#)

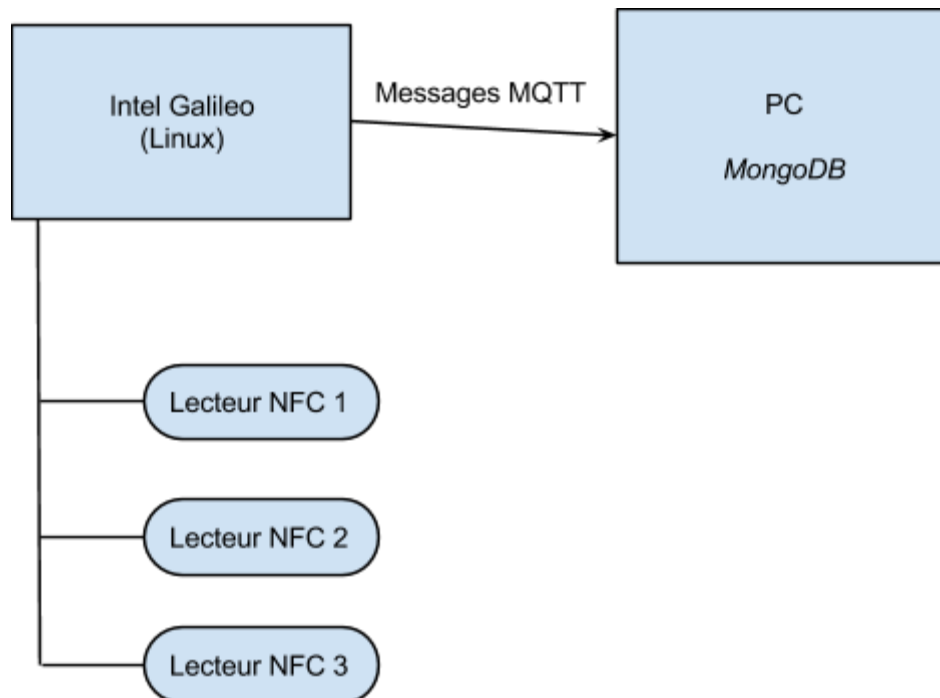
Introduction

Le capteur que l'on nous a confié est un lecteur de Tag NFC. Le but de ce projet était de récupérer les données de ce lecteur et de s'en servir pour effectuer un vote. L'idée était de relier trois lecteurs NFC sur un galileo et d'affecter à chacun une valeur pour le vote actuel (Positif, Négatif, Neutre). Ce système de vote permet une grande facilité d'utilisation puisqu'on peut imaginer l'utiliser dans une conférence où chaque invité a un badge avec un Tag NFC dont il se servira pour voter simplement en l'approchant du lecteur correspondant à son choix. Ce système rend le vote très rapide et permet du surcroît de vérifier que chaque personne ne vote qu'une fois.

Les différents outils utilisés sont Mqtt et MongoDB. Nous les utilisons au travers de scripts NodeJS et Python.

Architecture

Voici l'architecture du projet :



Les trois lecteurs NFC sont connectés en USB sur le port *USB Host* du Galileo. Le Galileo est relié directement au PC par un câble RJ 45.

Galileo

Le Galileo fait tourner un système Linux présent sur sa carte SD. Le fait d'avoir Linux sur le Galileo nous permet d'avoir le support de l'USB ainsi qu'un interprète python et l'ajout des librairies nécessaires pour mqtt est également facilité. Nous avons compilé mosquitto (Mqtt) directement sur la carte pour pouvoir lancer un broker Mqtt et se servir des bindings python pour le Mqtt.

PC

Le PC a lui aussi un broker Mqtt qui tourne et qui lui permet de recevoir les messages envoyé par le Galileo. On lance également une base de données MongoDB qui nous sert à stocker les votes. Pour finir nous utilisons un serveur NodeJS pour recevoir les messages du Galileo et remplir la base de données.

Logiciel

Galileo

On lance trois scripts python, un pour chaque lecteur NFC branché en USB. Chacun gère une valeur pour le vote (Positif=0, Négatif=1, Neutre=2). Pour lancer ces scripts il faut récupérer l'identifiant du device USB à l'aide de la commande lsusb :

```
root@clanton:~# lsusb
Bus 002 Device 003: ID 05e3:0608 Genesys Logic, Inc. USB-2.0 4-Port HUB
Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
Bus 002 Device 001: ID 1d6b:0002 Linux Foundation 2.0 root hub
Bus 002 Device 005: ID 04e6:5591 SCM Microsystems, Inc.
Bus 002 Device 004: ID 04e6:5591 SCM Microsystems, Inc.
```

Par exemple si l'on s'intéresse au premier lecteur SCM Microsystems, les identifiants qui nous intéressent sont 002 et 005.

Admettons qu'on affecte à ce lecteur la valeur 0, voici l'appel à effectuer :

```
python read.py 002:005 1
```

On fera de même pour les autres lecteurs.

Le script read.py se sert de la librairie python nfc. On définit un handler pour chaque lecture d'un tag NFC. Ce handler lit le contenu du tag NFC et transmet dans un message Mqtt l'id du votant et la valeur du vote.

PC

Du côté du PC on a dans un premier temps un serveur NodeJS qui permet d'interfacer avec le Galileo, la base de données MongoDB et le programme python d'affichage des votes.

La base de données MongoDB permet le stockage des votes. Elle nous permet également indirectement de vérifier qu'une personne ne vote pas deux fois grâce à l'exception duplicate key qui est levé si la personne a déjà voté.

Pour le serveur NodeJS on a besoin de deux parties principales.

La première partie est la base de données mongoDB. On se connecte à celle-ci pour pouvoir ajouter des entrées. Pour ajouter des entrées il faut créer un schéma, il est composé d'un id unique et de la valeur du vote.

La seconde partie est le support des messages mqtt. A chaque vote, on reçoit un message mqtt qui nous en informe. Pour cela on souscrit au topic des votes, avec un client qui écoute sur l'ip du galileo.

A chaque réception d'un message on tente d'insérer le vote avec l'id du votant. Si cet id est déjà présent dans la table on ne duplique pas son vote et on affiche un message d'erreur. Si le vote est valide on transmet un second message mqtt sur un autre topic qui servira à l'affichage.

Le programme d'affichage lorsqu'il est lancé récupère les valeurs déjà présente en base de données. Par la suite à chaque réception d'un message mqtt constitué de la valeur du vote l'affichage est actualisé. L'affichage est réalisé à l'aide de la librairie pygame qui permet d'afficher des rectangles et du texte. Les coordonnées des rectangles sont calculés selon la taille du plus grand des trois, de manière à ce que l'échelle soit respectée.

