



420-410 GR : 00001
Introduction aux plateformes IdO

Épreuve finale

Remise : 24 mai 2024

Prénom & Nom : _____

Matricule : _____

Signature : _____

/ 40

- *Il est interdit d'aller chercher de l'information ailleurs que dans le matériel autorisé.
(Ex. : Écran autre que le sien, échanger du code avec un camarade, site interdit ...)*
- *Tout le matériel non autorisé doit être déposé à l'avant de la classe.*
- *Le téléphone DOIT être déposé dans votre sac, à l'avant de la classe.*

Remise :

- *Détails dans ce document.*

Matériel autorisé :

- *Toute documentation externe est permise à l'exception des outils IA de génération de code
(ChatGPT, Gemini, Perplexity etc.)*

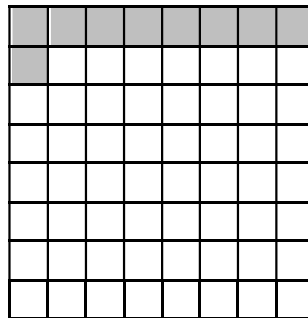
Description générale

Vous devez faire une application IoT utilisant un *RaspberryPi* et un conteneur. Les deux doivent communiquer entre eux à l'aide de *sockets* TCP et le serveur doit aussi communiquer avec un agent MQTT.

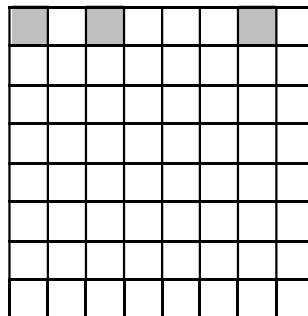
Le *RaspberryPi* est relié à un bouton et à une matrice LED 8x8.

Lorsqu'on appuie ou on relâche le bouton, l'application envoie le signal correspondant à l'agent MQTT. L'application doit également allumer sur la matrice une ou plusieurs LED en fonction des boutons qui sont appuyés par les autres équipes.

Par exemple s'il y a 9 équipes, les 9 premières LED de la matrice seront utilisées :



Chaque LED correspond à une équipe, donc si par exemple les équipes 1, 3 et 7 appuient en même temps sur leur bouton les LED seront allumées comme suit :



Spécifications

Communication TCP

Le *RaspberryPi* envoie « 0 » ou « 1 » au conteneur selon que le bouton est appuyé ou non.

Le conteneur envoie deux nombres au *RaspberryPi* qui correspondent à la LED à allumer ou éteindre. Le premier nombre est le numéro d'une équipe et le deuxième est « 0 » ou « 1 » selon qu'on doit allumer ou éteindre la LED.

Vous avez le choix d'utiliser une même connexion TCP pour les deux types de messages, ou deux connexion distinctes (une pour chaque type de message).

Communication MQTT

Les messages doivent être envoyés à l'agent à l'adresse 10.10.0.197 sur la rubrique MQTT nommée « final ».

Le format des messages MQTT est **équipe:état**, où **équipe** est le numéro d'une équipe et **état** peut être la chaîne de caractères « on » ou « off ».

La connexion doit être authentifiée et sécurisée à l'aide d'un certificat numérique – vous devrez faire la demande de certificat vous-même et la transmettre au professeur pour qu'il vous remette le certificat à déployer sur votre client MQTT.

Répartition du travail

Le travail est fait en équipe de deux. Un membre de l'équipe travaillera exclusivement sur le RaspberryPi, et l'autre travaillera exclusivement sur le conteneur. La note est individuelle.

Chaque membre devra créer un projet privé sur GitHub et mettre le professeur comme « maintenir ». Le coéquipier ne sera pas membre du projet.

Livrables

À la fin de chaque séance de travail, le code devra être « poussé » sur GitHub. Dans le cas d'un oubli, un retard de 5% par jour sera compté.

Lorsque le projet sera terminé, une démonstration devra être faite au professeur.

Grille de correction

| | |
|---|------------|
| Fonctionnalité : le programme fait ce qui est demandé et le fait de manière efficace / rapide | /10 |
| Robustesse : le programme fonctionne dans des conditions réalistes sans erreurs d'exécution | /10 |
| Qualité du code : le code du programme est bien structuré (fonctions, structures de contrôle) et ne contient pas d'éléments inutiles ou inefficaces. | /10 |
| Commentaires : Les commentaires sont clairs et expliquent, pour les différentes sections du code, à quoi elles servent et leur fonctionnement général | /10 |
| TOTAL | /40 |

Attention : les programmes qui ne compilent pas ne seront pas corrigés.