

Développement d'objets intelligents

Évaluation 1

Projet intégrateur

Objectif

L'objectif de ce travail pratique est de :

- Réaliser des applications qui permettent de contrôler et de superviser des objets intelligents.
- Rechercher de l'information technique sur Internet.

Travail à réaliser

Vous devez réaliser deux (2) applications permettant de contrôler et de superviser trois (3) objets intelligents. Deux (2) des trois (3) objets intelligents sont programmés en Python et doivent être modifié pour communiquer avec le protocole MQTT.

Réaliser le projet en équipe de 2 personnes.

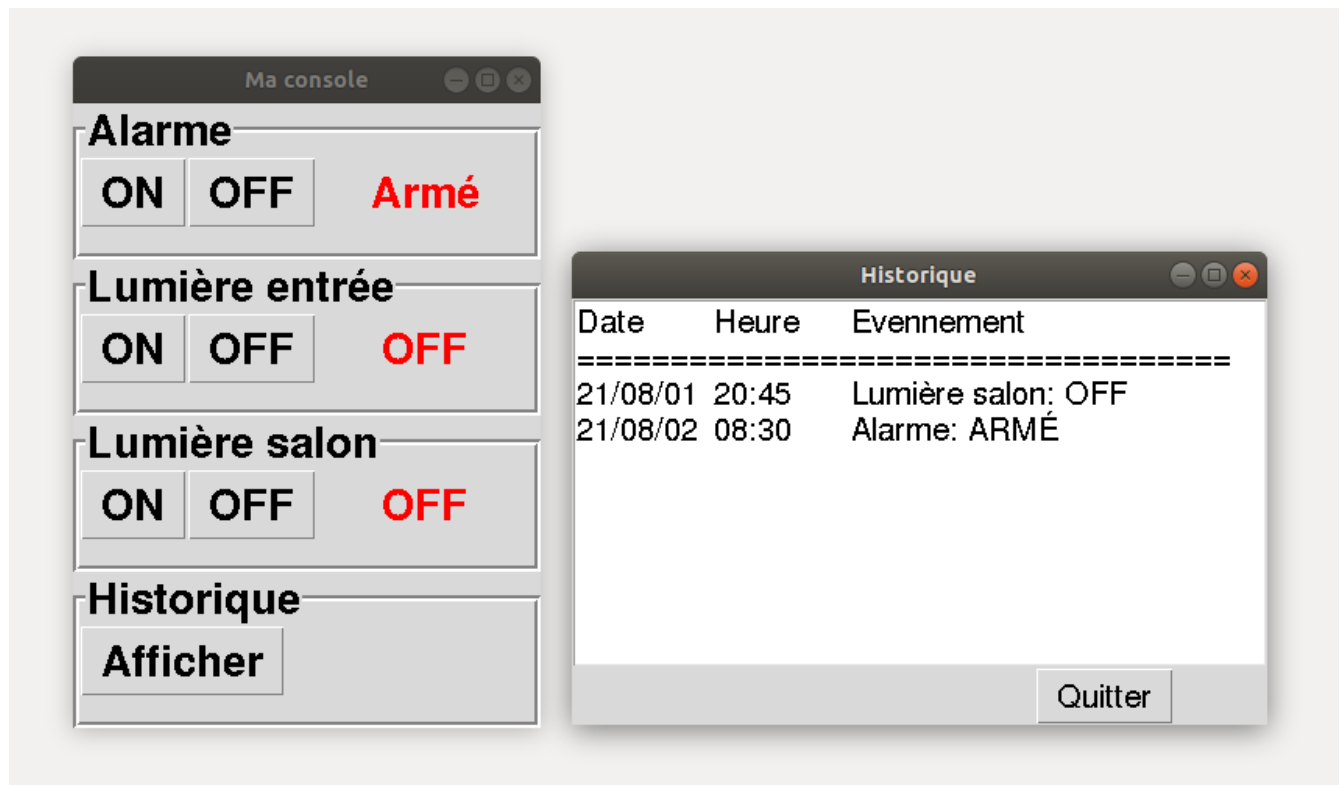
Description des objets intelligents

Nom	Protocole	Description
SmartPlug	MQTT	Module permettant d'allumer ou d'éteindre une lumière. Langage de programmation : Python Plateforme : Raspberry Pi Travail à réaliser : modifié pour communiquer avec le protocole MQTT. Historique des commandes enregistrées sur un serveur MongoDB.
AlarmePi	MQTT	Système d'alarme. Langage de programmation : Python Plateforme : Raspberry Pi Travail à réaliser : modifié pour communiquer avec le protocole MQTT. Historique des commandes enregistrées sur un serveur MongoDB.
SmartDimmer	Zwave	GE Zwave plug-in dimming lamp module.

Application Python

Programmer une application Python utilisant Tkinter qui permet :

- de contrôler les 3 objets intelligents (MQTT et zwave);
- d'afficher l'état des 3 objets intelligents (MQTT et zwave);
- d'afficher l'historique des commandes et des évènements qui sont stockés sur un serveur MongoDB.



Application MQTT-Dash

Programmer une application avec MQTT-Dash :

- de contrôler les 2 objets intelligents compatibles MQTT;
- d'afficher l'état des 2 objets intelligents compatibles MQTT.

Carnet du produit (*product backlog*)

SmartPlug

1. Recevoir des commandes via le protocole MQTT.
2. Envoyer son état via le protocole MQTT;

3. Stocker l'historique des commandes (changement d'état) sur un serveur MongoDB.

AlarmePi

1. Recevoir des commandes via le protocole MQTT.
2. Envoyer son état via le protocole MQTT;
3. Stocker l'historique des commandes (changement d'état) sur un serveur MongoDB.

Application Python

1. Concevoir et programmer l'interface Tkinter.
2. Recevoir les modifications d'état des objets via le protocole MQTT.
3. Envoyer des commandes aux objets via le protocole MQTT.
4. Recevoir les modifications d'état des objets via le protocole zwave.
5. Envoyer des commandes aux objets via le protocole zwave.
6. Afficher l'historique des commandes qui est stocker sur un serveur MongoDB.
7. Conserver seulement les 20 dernières entrées dans l'historique sur le serveur MongoDB.

Application MQTT-Dash

1. Recevoir les modifications d'état des objets via le protocole MQTT.
2. Envoyer des commandes aux objets via le protocole MQTT.

Remise du projet

- Date de remise : mercredi le 8 septembre 2021 (cours 7).
- Remettre via **Omnivox (LÉA)** un fichier ZIP contenant les fichiers sources du projet.
- Le 8 septembre vous devez présenter votre projet à votre enseignant.

Pondération

Présentation orale (10 min)	/1
Fonctionnement	/8
Lisibilité des codes sources	/1