

Conception logicielle

Mini-projet : Gestion automatique d'une maison

Ce mini-projet consistera à mettre en place une architecture basée sur des webservices pour la gestion automatique d'une maison. Vous allez mettre en place des scénarios qui vous permettent de gérer automatiquement la température, l'humidité et la luminosité dans une pièce. Vous pouvez imaginer d'autres scénarios.

Pour ce faire, vous allez utiliser un module ESP8266, un capteur de température et d'humidité, un capteur de luminosité, une LED, un bouton poussoir et un afficheur LCD.

Etape 1 : Application nodeMCU

1/ Configuration de l'IDE Arduino pour le développement de NodeMCU (ESP8266)

S'il n'est pas déjà installé, téléchargez et installez l'IDE Arduino à partir du lien suivant : <https://www.arduino.cc/en/Main/Software>

Afin qu'Arduino IDE prenne en charge la carte NodeMCU, suivez les étapes suivantes :

- Accédez à "Fichiers" > "Préférences".
- Copiez l'URL ci-dessous dans la zone de texte "Additional boards Manager URLs" : http://arduino.esp8266.com/stable/package_esp8266com_index.json
- Cliquez sur "OK" pour fermer la boîte de dialogue "Préférences".
- Allez dans "Outils", puis sélectionnez "Gestionnaire de cartes".
- Tapez dans la zone de texte "esp8266", puis accédez à "esp8266 by esp8266 community" et installez le logiciel pour Arduino.
- Allez dans "Outils" > "Carte...", puis sélectionnez la carte "NodeMCU 1.0".
- Une fois ces étapes validées, vous êtes prêt à programmer la carte NodeMCU avec Arduino IDE.

2/ Premiers tests

Dans cette partie, vous allez récupérer les données de température et d'humidité et les afficher sur le moniteur série de la carte et agir sur la lampe et l'afficheur LCD.

Vous pouvez créer un système intelligent pour votre maison avec ce comportement :

- Changement de la couleur de la LED : rouge pour une valeur d'humidité élevée et vert pour une valeur normale. (Pensez à définir un seuil)
- Affichage « Température Elevée / Normale / Basse » en fonction de la valeur de température mesurée.
- Allumer/ Éteindre la LED : changement de l'état de la LED lorsque la luminosité est supérieure/ inférieure à un seuil.
- Allumer/ Éteindre la LED manuellement avec le bouton.

L'accès en lecture et en écriture à ces équipements nécessite l'ajout de bibliothèques spécifiques. Vous pouvez vous référer aux ressources suivantes pour les intégrer correctement :

- Capteur de température et d'humidité : https://wiki.seeedstudio.com/Grove-TemperatureAndHumidity_Sensor
- LED : https://wiki.seeedstudio.com/Grove-Chainable_RGB_LED/
- LCD : https://wiki.seeedstudio.com/Grove-16x2_LCD_Series
- Capteur de luminosité : veuillez télécharger la bibliothèque disponible sur moodle.

Le code développé devra être inclus parmi les livrables.

Etape 2 : Web service

Vous allez mettre en place un web service qui vous permettra de gérer un capteur/ actionneur. Ce web expose l'API suivante :

- Affichage de la liste de tous les capteurs
- Affichage un capteur par son Id.
- Affichage la dernière valeur mesurée.
- Affichage de la plus ancienne valeur mesurée.
- Ajout d'un capteur.
- Ajout d'une valeur pour un capteur en précisant son Id.

Vous avez la possibilité d'implémenter un webservice distinct pour chaque équipement, ou bien de concevoir un webservice unique centralisé assurant la gestion de l'ensemble des dispositifs matériels disponibles.

Veuillez justifier votre choix dans le rapport.

Spring Boot est le framework recommandé pour la mise en œuvre du webservice.

Etape 3 : Interconnexion webservice et NodeMCU

Dans cette partie, vous allez rendre votre client NodeMCU capable de communiquer avec le(s) webservice(s) que vous avez implémenté(s) dans l'étape précédente.

Pour ce faire, vous allez utiliser les bibliothèques : `<ESP8266WiFi>` et `<ESP8266HTTPClient>`.

Dans un premier temps, le client NodeMCU doit être capable d'ajouter (initialiser) un capteur dans le webservice et d'ajouter périodiquement une valeur.

Ensuite, vous allez mettre en place un serveur HTTP sur le NodeMCU dans le but de le rendre capable d'intercepter des commandes depuis l'extérieur (webservice) et de changer l'état d'un actionneur (ici la LED) en fonction de la valeur transmise.

Pour tester votre application, modifiez les caractéristiques du réseau pour se connecter au réseau wifi local (votre propre réseau wifi avec votre smartphone).

Le code développé devra être inclus parmi les livrables.

Etape 4 : Automatisation Domotique par Orchestration de capteurs et d'actionneurs

Vous allez implémenter un web service orchestrateur mettant en place les scénarios suivants :

- Pour une valeur de luminosité faible => action : allumer la LED
- Pour une valeur de luminosité forte => action : éteindre la LED
- Pour une valeur d'humidité faible=> action : écrire un message sur l'afficheur LCD « valeur normale, VMC : faible vitesse »
- Pour une valeur d'humidité élevée => action : écrire un message sur l'afficheur LCD « valeur élevée, VMC : forte vitesse »

Vous pouvez imaginer d'autres scénarios.

Vous devriez rendre vos webservices découvrables et utilisables par les clients indépendamment de leur adresse de déploiement. Vous allez utiliser l'API développée par la société de streaming vidéo Netflix OSS (Open Source Software). Parmi ce que cette API offre, vous allez utiliser Eureka qui offre un service d'enregistrement (service Registry). Ce dernier est un annuaire de services, qui permet de mettre en place un serveur de découverte. Ainsi, les microservices qu'on souhaiterait rendre « découvrables », s'inscrivent sur le serveur, pour être vus par les autres.

Etape 5 : Interface web

L'objectif de cette partie est de créer une application en utilisant le travail que vous avez réalisé dans les précédentes parties. Pour ce faire, vous allez utiliser le framework Node-RED.

Node-RED est un outil puissant et intuitif basé sur un navigateur qui permet aux développeurs de créer facilement des applications. Il offre une approche visuelle pour connecter des nœuds et créer des flux de données, éliminant ainsi la nécessité d'une programmation complexe.

Node-RED est basé sur Node.js, ce qui lui confère une grande flexibilité et une vaste communauté de développeurs. Il peut être utilisé pour développer des applications IoT, des tableaux de bord de données en temps réel et bien d'autres applications.

1/ Installation

Avant d'installer Node-RED, vous devez avoir une installation fonctionnelle de Node.js. Je vous recommande d'utiliser la dernière version de Node.js. (version > 18)

Une fois l'installation de Node.js terminée, vous allez procéder à l'installation de Node-RED. La méthode la plus simple consiste à utiliser le gestionnaire de paquets de Node, npm, qui est inclus dans Node.js.

Pour l'installer en tant que module global, ajoutez la commande node-red à votre chemin système :

```
npm install -g --unsafe-perm node-red
```

Pour lancer Node-RED, ouvrez un terminal et exécutez la commande suivante : node-red.
Pour accéder à l'interface graphique, saisissez cette url dans votre navigateur :
<http://localhost:1880> (1880 est le port par défaut de node-red).

Vous pouvez consulter la documentation pour créer vos premiers flux.
(<https://nodered.org/docs/tutorials/>)

2/ Application

Créez un tableau de bord qui affiche les valeurs des capteurs avec différents graphiques et des boutons pour allumer/éteindre la LED.

Pour ce faire vous allez utiliser des nœuds de type : « node-red-dashboard ». Vous devez les installer manuellement à partir de « manage palette »

Bonus : Vous pouvez configurer les seuils d'humidité et de luminosité à partir de l'interface graphique.