HAUM Sweet OHM Documentation

The HAUM

janv. 19, 2019

Contents:

1 Histoire **1**

2 Ressources **3**
2.1 Réseau . . . 3
2.2 Broker MQTT . . . 3
2.3 Serveur MPD . . . 3
2.4 Laumios . . . 3
2.5 MQTT MPD . . . 4

3 Les capteurs **5**
3.1 Capteur Bouttons poussoir . . . 5
3.2 Télécommande infrarouge . . . 6
3.3 Capteur détecteur de présence . . . 6
3.4 Capteur de distance . . . 6
3.5 Capteur atmosphérique . . . 7

4 Contrôler la musique : pont MQTT MPD **9**
4.1 Structure . . . 9
4.2 Commandes . . . 9

5 Les Objectifs **11**
5.1 La base . . . 11
5.2 Attentes spécifiques . . . 11

6 Déroulement **13**

7 Évaluation **15**

8 FAQ **17**

CHAPITRE 1

Histoire

Yuang-Lu aime beaucoup les Laumios [1] du Haum et en a fabriqué une collection pour décorer son hackerspace local. En bons bidouilleurs, les occupants du lieu souhaitent aller plus loin avec ce système et façonner l'ambiance en fonction des évènements du laboratoire et du monde extérieur.

Tsai-Hen s'occupera d'agrémenter le lieu de capteurs physiques tandis que votre équipe chevronnée a choisi de rivaliser d' ingéniosité pour concevoir la partie logicielle.

Bien motivés, Yuang-Lu et Tsai-Hen ont déjà mis en réseau les laumios et quelques capteurs. De votre côté, vous avez choisi de faire tourner votre logiciel sur une de vos machines personnelles qu' il suffira de relier à ce réseau pour le premier prototype présenté dimanche. Vous avez donc tout pour commencer les expérimentations.

Lampes multicolores pouvant être animées et pilotées en réseau https://haum.org/pages/laumios.html

CHAPITRE 2

Ressources

2.1 Réseau

ESSID HAUMSweetOHM
Pass Bienvenue aux 24 Heures du code.

2.2 Broker MQTT

Hostname mpd.lan
Port 1883
User/Pass (aucun)

Websockets
Port 1884
User/Pass (aucun)

2.3 Serveur MPD

Hostname mpd.lan
Port 6600
User/Pass (aucun)

2.4 Laumios

Code github.com/laumio
Documentation laumiomqtt.rtfd.io

2.5 MQTT MPD

Code github.com/mqtt_mpd

Documentation mqtt-mpd

CHAPITRE 3

Les capteurs

Les capteurs dialoguent exclusivement avec le protocole MQTT. Ils publient leur état et réagissent à quelques commandes par le biais de topics MQTT détaillés par la suite.

3.1 Capteur Bouttons poussoir

Il s'agit d'un ensemble de quatre boutons poussoir associés à quatre LEDs. Un appui sur le boutton change l'état de la LED associée et transmet l'état de ces deux entités sur le topic concerné. L'état des LEDs peut être piloté indépendamment également.

Le capteur publie les informations suivantes :

`capteur_bp/status :` Il s'agit du statut de connexion du capteur qui peut valoir « online » ou « offline ». Cette donnée est émise par le broker à la connexion au topic.

`capteur_bp/switch/ledX/state :` Il s'agit de l'état de la led n°X (X de 1 à 4 inclus). La valeur peut être « ON » ou « OFF ». Cette donnée est émise par le broker à la connexion au topic.

`capteur_bp/binary_sensor/bpX/state :` Il s'agit de l'état du bouton poussoir n°X (X de 1 à 4 inclus). La valeur peut être « ON » ou « OFF ». Cette donnée est émise lors du changement d'état du bouton.

`capteur_bp/sensor/bp_rssi/state :` Il s'agit du niveau du signal wifi du capteur. Cette donnée est transmise lors du changement de sa valeur. Elle est publiée comme une chaîne de caractères de la valeur en décibels l'indicateur de réception du signal. Elle est mise à jour toutes les 15 secondes.

`capteur_bp/sensor/uptime_sensor/state` Il s'agit du temps de fonctionnement (uptime) du capteur. La valeur est exprimée en minutes. Cette donnée est transmise lors du changement de sa valeur.

Le capteur reçoit les commandes suivantes :

`capteur_bp/switch/ledX/command :` Il s'agit de la commande de la led n°X (X de 1 à 4 inclus). Le contenu du message attendu est « ON » ou « OFF ».

`capteur_bp/status/advertise` Il s'agit d'une commande de découverte. Quel que soit le contenu du message, la réponse est publiée sur le topic `capteur_bp/status` avec la valeur « online » si le capteur est présent.

3.2 Télécommande infrarouge

Il s'agit d'un récepteur de télécommande infrarouge. La télécommande dispose de 21 touches dont les états sont publiés sur les topics :

`remote/NOM_DE_LA_TOUCHE/state` avec NOM_DE_LA_TOUCHE la touche correspondante (voir la liste qui suit). Chaque appui publie deux messages, le premier avec un payload « ON » à la pression et le second avec le message « OFF » au relâchement.

les touches disponible sur la télécommande sont les suivantes :

power	mode	mute
playp	prev	next
eq	minus	plus
0	chg	u_sd
1	2	3
4	5	6
7	8	9

3.3 Capteur détecteur de présence

Ce capteur indique la présence d'une personne dans son champ de détection.

Le capteur publie les informations suivantes :

`presence/state` : Il s'agit de l'état de la détection par le capteur. Les valeurs peuvent être « OFF » lorsqu'il n'y a pas de détection ou « ON » pour une présence devant le capteur. Il publie un nouvel état a chaque changement de celui-ci.

`presence/status` : Il s'agit du statut de connexion du capteur qui peut valoir « online » ou « offline ». Cette donnée est émise par le broker à la connexion au topic.

Le capteur reçoit la commande suivante :

`presence/status/advertise` : Il s'agit d'une commande de découverte. Quel que soit le contenu du message, la réponse est publiée sur le topic `presence/status` avec la valeur « online » si le capteur est présent.

3.4 Capteur de distance

Il s'agit d'un capteur qui donne une indication de la distance le séparant du plus proche objet dans son champ de détection.

Le capteur publie les informations suivantes :

`distance/value` : Il s'agit de la distance de l'objet lui faisant face exprimée en mètres. La précision est de quelques centimètres. Elle est publiée comme une chaine de caractères de la valeur en mètres sous la forme X.XX . Elle est mise à jour toutes les 5 secondes quand un objet est dans le périmètre de détection.

`distance/status` : Il s'agit du statut de connexion du capteur qui peut valoir « online » ou « offline ». Cette donnée est émise par le broker à la connexion au topic.

Le capteur reçoit la commandes suivante :

`distance/status/advertise` : Il s'agit d'une commande de découverte. Quel que soit le contenu du message, la réponse est publiée sur le topic `distance/status` avec la valeur « online » si le capteur est présent.

3.5 Capteur atmosphérique

Il s'agit d'un capteur de métrologie atmosphérique : température, pression, humidité.

Le capteur publie les informations suivantes :

`atmosphere/status` : Il s'agit du statut de connexion du capteur qui peut valoir « online » ou « offline ». Cette donnée est émise par le broker à la connexion au topic.

`atmosphere/temperature` : Il s'agit de la température mesurée exprimée en degrés Celsius. La valeur est mise à jour toutes les 15 secondes. Elle est publiée sous forme d'chaine de caractères sans indication d'unité.

`atmosphere/pression` : Il s'agit de la pression atmosphérique mesurée exprimée en hPa. La valeur est mise à jour toutes les 15 secondes. Elle est publiée sous forme d'chaine de caractères sans indication d'unité.

`atmosphere/humidite` : Il s'agit de l'humidité relative mesurée exprimée en pourcent. Elle est publiée sous forme d'une chaine de caractères sans indication d'unité. La valeur est mise à jour toutes les 15 secondes.

`atmosphere/humidite_absolue` : Il s'agit de l'humidité absolue mesurée exprimée en g/m^3. La valeur est mise à jour toutes les 15 secondes. Elle est publiée sous forme d'chaine de caractères sans indication d'unité.

Le capteur reçoit la commandes suivante :

`atmosphere/status/advertise` : Il s'agit d'une commande de découverte. Quel que soit le contenu du message, la réponse est publiée sur le topic `atmosphere/status` avec la valeur « online » si le capteur est présent.

CHAPITRE 4

Contrôler la musique : pont MQTT MPD

Pour diffuser de la musique dans le hackerspace, c'est MPD qui a été retenu. Ce programme expose une interface telnet sur le port 6600 par défaut (les commandes sont présentées dans sa documentation).

Afin de contrôler ce programme, les hackers ont écrit un pont entre l'interface telnet de MPD et MQTT que cette page décrit.

4.1 Structure

Un petit script Python relié au broker MQTT écoute des commandes sur les topics `music/control/<commande>`. La payload associée permet au besoin de passer des arguments à la commande.

Seules les commande de base sont exposées dans MQTT.

4.2 Commandes

Les commandes fonctionnelles sont les suivantes (sauf mention contraire, la payload est ignorée et le script ne renvoie rien) :

`music/control/getstate` retourne l'état de lecture dans `music/status` (*play*, *pause* ou *stop*)

`music/control/getvol` retourne le volume courant (entre 0 et 100) dans `music/status`

`music/control/next` passe au morceau suivant

`music/control/previous` passe au morceau précédent

`music/control/stop` arrête la musique

`music/control/play` (re)démarre la lecture

`music/control/pause` met la lecture en pause

`music/control/toggle` agit comme un bouton Play/Pause (met en pause si en lecture et en lecture si stoppé ou pausé)

`music/control/setvol` permet de règler le volume au moyen d'un paramètre en payload entre 0 et 100 (entier uniquement)

CHAPITRE 5

Les Objectifs

Dans le cadre de ce lieu partagé, ingéniosité et originalité sont de rigueur. Votre projet doit par ailleurs être facile d'utilisation sous peine de le voir remplacer par celui d'une autre équipe qui aura davantage séduit. Plusieurs petits défis s'agrègent pour satisfaire au mieux les futurs utilisateurs.

5.1 La base

Vous devez réaliser ces tâches dans l'ordre indiqué. Elles ont un objectif didactique et nous permettent d'évaluer la vitesse de progression des équipes.

- Être capable de choisir la couleur des laumios
- Être capable d'animer les laumios (fournir plusieurs animations, éventuellement configurables)
- Gérer l'allumage/animation des laumios par groupes ou individuellement
- Récupérer les informations des capteurs et y réagir (callback)

5.2 Attentes spécifiques

Vous pouvez réaliser ces tâches dans l'ordre que vous souhaitez. Vous êtes également libres, une fois tous ces points traités, de perfectionner votre système.

- Réagir à des sources extérieures (ex. Mastodon, Wikipedia, météo, manifestations. . .)
- Avoir une interface quelconque (ex. CLI, application mobile, interface web, application PC. . .) pour interagir facilement avec l' installation
- Permettre de programmer soi-même des réactions aux évènements (note : certains utilisateurs ne connaissent pas les langages de programmation, soyez créatifs)

Les utilisateurs sont pointilleux quant à la simplicité de programmation mais aussi à la complexité des scenarii qu' ils pourront mettre en œuvre avec votre outil

CHAPITRE 6

Déroulement

Vous avez accès à un réseau de test sur lequel des laumios et les capteurs sont installés. Les équipes doivent s'organiser afin qu' il n'y ait pas de monopolisation du terrain (note : selon la nature des tests, il est même possible de partager cet espace en même temps). Il peut être utile de mettre en place des micro-simulateurs sur votre machine pour effectuer les tests en dehors de l' infrastructure physique.

Les sujets proposés par le HAUM n'ont pas l'habitude d'être à visée scolaire. Ce sujet-ci ne déroge pas à la règle et est tout à fait dans l'esprit du hackerspace : ce week-end, nous partageons collectivement une aventure expérimentale. Ce qui nous intéresse n'est pas votre capacité à résoudre un problème fermé mais bien l' ingéniosité que vous mettrez pour imaginer une solution possible à cette problématique atypique.

Parce que les 24h c'est aussi et surtout un temps de partage entre passionnés. Les équipes du sujet HAUM Sweet OHM se verront proposer d'aller boire un verre pour se détendre et faire connaissance au cours de la nuit.

Des séances collectives (obligatoires) avec toutes les équipes sont prévues tout au long de l'évènement pour faire le point et discuter des difficultés et des succès. Nous serons présents au cours de la nuit.

CHAPITRE 7

Évaluation

L'évaluation des solutions est effectuée en partie de manière continue (lors des points collectifs ou lorsque vous évoquez vos choix techniques, vos avancées et vos difficultés aux porteurs du sujet par exemple) et par une présentation que vous ferez à tous (avec démonstration si possible) à la fin de l'épreuve (pensez à dormir). Nous accorderons beaucoup d' importance à :

- Originalité/innovation
- Pertinence des choix techniques
- Appréciation et implication collectives
- Qualité de la prise en compte de tous les objectifs
- Facilité d'utilisation et complexité des scenarii réalisables par l'utilisateur final

CHAPITRE 8

FAQ

Cette page sera complétée tout au long des 24 heures du code.