

PROJET ARCHITECTURE DES ORDINATEURS

Lien du git : <https://github.com/clement-bsrs/archi-mk1>

Fonctionnement global :

Notre projet représente un ensemble d'appareils pouvant servir dans une maison connectée. Il est composé de capteurs (température, pression atmosphérique, humidité de l'air), ainsi que d'un capteur de présence (représenté par un capteur de badge RFID). Ces capteurs sont placés sur des Arduino faisant office d'émetteurs. Leur but est d'envoyer ces données à un serveur. Le projet est donc aussi composé d'un Arduino récepteur, que l'on a appelé serveur. Il récupère les données des différents émetteurs afin de les stocker, les afficher et les traiter.

Emetteur :

Chaque émetteur représenté par : Un Arduino équipé d'un émetteur 433 MHz, d'un récepteur 433 MHz, d'un module RFID (VMA405), d'un capteur (VMA335 permettant de lire la température, la pression atmosphérique et le taux d'Humidité Relative en %) et d'un écran à cristaux liquide équipé d'une liaison parallèle.

L'Arduino émetteur, fonctionne d'une manière assez particulière. Nous avons essayer de développer une manière d'augmenté artificiellement la distance de communication entre un émetteur et un récepteur (Serveur) en faisant en sorte de retransmettre les trames par chaque Arduino émetteur, cette trame n'est en aucun cas changée. Pour se faire nous avons réfléchi à un code qui permettrai de retransmettre la trame reçu uniquement si on ne la pas déjà reçu, ce qui permettrai, en théorie, si les émetteurs sont bien disposés, d'atteindre à coup sûr le serveur sans qu'une trame déjà retransmise le soit a nouveau en créant des boucles infini. Ce système fonctionne de façon non-bloquant de bout en long.

L'envoi de trame est visible sur l'écran par une « * » qui clignote à chaque envoie, les données des capteurs sont affiché en temps réel et l'initialisation est visible au démarrage de l'Arduino. Nous avons deux types de trames, une pour les informations du capteur des constantes et une pour le lecteur RFID.

Ces Arduino sont capable de recevoir des trames du serveur, avec la possibilité d'identifier chaque Arduino.

On peut imaginer tout type de branchement, par exemple, contrôler des volets, des radiateur, une ventilation de salle de bain (en fonction de l'humidité présente dans l'air) mais cette feature du projet n'a pas été développé jusqu'au bout par manque de temps et de matériel sur les actionneurs, nous nous sommes dit que ce n'était pas la chose la plus importante tout en restant un axe à améliorer.

Récepteur/Serveur :

Le serveur ne possède pas beaucoup de technologies. En effet, il est simplement muni d'un récepteur et émetteur RF afin de récupérer, et potentiellement envoyer, des informations aux différents Arduino émetteurs/capteurs.

Le serveur est capable de reconnaître si une donnée provient bien des Arduino émetteurs en fonction de la taille de la trame, ainsi que de l'en-tête (« XX »). Ainsi, il ne stocke et ne traite pas d'informations parasites.

De la même manière, il ignore les doublons (un doublon provient du fait que deux Arduino envoient la même trame avec la fonctionnalité du Peer to Peer, mais que le serveur reçoit les deux). Pour l'instant, nous avons choisi de ne stocker que 50 trames à la fois, puisque nous ne traitons pas les informations (nous n'en faisons rien).

Enfin, le serveur affiche de temps en temps un menu avec les informations de la dernière trame.

Points d'amélioration :

- Faire en sorte que les données de l'Arduino soit réceptionné par un ordinateur afin de les stocker dans une base de donnée (Excel, format JSON, ou autre), nous ne l'avons pas produit car nous avons eu des complication avec les machines de la fac.

- (Optionnel) Faire en sorte que le menu du serveur soit divisé en plusieurs sections, avec une section pour chaque Arduino émetteur (leurs dernières données), et une dernière avec la moyenne des données de toute la maison.