

Φ -X/IT

PHYsically eXtended Interactive Technologies

Le sujet du projet

Nous allons concevoir un système permettant de découvrir, collecter et monitorer un ensemble de capteurs disséminés dans un environnement. Ces capteurs sont de deux types : soit directement accessibles via TCP/IP (par un broker MQTT, API REST ou directement via des sockets).

Les capteurs sont soit des **capteurs de terrain** qui émettent régulièrement (fréquence propre à chaque capteur) vers un **agrégat**, soit des **capteurs « virtuels »** (accessibles via des API REST). Les capteurs de terrain peuvent être reliés via différents protocoles matériels et/ou logiciels (liaison série, USB, radio RF 433, Zigbee, BLE, etc.).

L'objectif de ce projet est de permettre de *gérer le plus finement possible* un réseau des capteurs d'une part et de proposer dans un dashboard une visualisation et gestion des données afin de prévenir de potentiels problèmes.

Etape 1 : une représentation formelle

La première tâche a pour objectif de définir une représentation numérique des capteurs.

Vous choisirez un format assez générique permettant de détailler les caractéristiques d'un capteur : son emplacement physique et numérique (url éventuellement, port de communication), son format de données, sa fréquence maximale, etc.).

Proposez cette représentation en discutant des avantages/inconvénients

Etape 2 : une architecture générale

Ces différents capteurs seront regroupés dans des **ensembles** de capteurs ; un capteur pouvant se retrouver dans différents ensembles.

Un **ensemble** consiste pour ce projet en un « *certain nombre* » (au moins 2) de capteurs (de **température, présence, consommation électrique, ...**) qui renvoient périodiquement leur mesure (pas forcément à la même fréquence).

Les ensembles stockent la configuration générale (nombre de capteurs, type, localisation, fréquence de remontée des données, ...) qui s'appuie sur la configuration de chaque capteur et traitent les informations reçues (par exemple, un calcul de moyenne lissée des différentes valeurs des capteurs, détection de données aberrantes, ...).

Chaque **ensemble** renvoie ses données vers un **superviseur** exposé sur internet via une API REST toutes les 10 mn (6 fois par heure). N'importe quel client web peut alors récupérer à tout moment ces données auprès du slot grâce à une requête web (le type de requête est à définir).

Proposez une architecture réseau supportant ce cahier des charges.

Etape 3 : une preuve de concept

Cette étape permet de mettre en œuvre une version initiale du réseau de capteurs supporté par les **ensembles**.

Vous devrez utiliser au moins dans ce projet un capteur physique (à base d'arduino, nodeMCU, ESP32, Raspberry Pi, etc.) Les autres peuvent être simulés logiquement.

Développez vos capteurs et n **ensemble** de capteurs comme décrit précédemment.

Il s'agira notamment de pouvoir ajouter/supprimer facilement un capteur dans un ensemble.

Nota : le matériel pourra être prêté pour le projet (micro-contrôleur ou Raspberry + capteurs) sur demande pour le développement du projet.

Étape 4 : Un service opérationnel

Le **superviseur** souhaite pouvoir récupérer la configuration courante des différents **ensembles** (son état au cours du temps, combien de capteurs sont connectés sur chaque ensemble, leurs types, leur localisation, etc.).

Dans cette dernière étape, on souhaite **proposer un dashboard** permettant de récupérer et d'afficher facilement toutes les données d'un ensemble ou d'un capteur en particulier avec une cartographie permettant une localisation des capteurs (vous pouvez utiliser par exemple l'API OpenStreetMap ou tout autre API de votre choix).

Proposez enfin sur le dashboard une détection de toute valeur anormale dans les capteurs, codez les alertes correspondantes et proposez une possible « prédiction » de l'évolution des valeurs. (Le dashboard peut être codé sur le web ou via une interface graphique « classique »).

Le projet s'effectue en binôme. Vous devrez remettre à l'issue du projet :

- **Un rapport** contenant la liste de vos choix de conception illustrés par des schémas, copies d'écran ou vidéo explicative.
Faites un bilan des avantages/inconvénients/limites de votre solution. Tâchez d'être le plus objectif possible !
- Le **code source** de vos développements (repository **git** ou archive **zip**) ainsi que l'**exécutable** éventuel.

L'ensemble des documents et liens devra être envoyé par **mail** (archive en pièce jointe ou lien de téléchargement) à Philippe Truillet (Philippe.Truillet@univ-tlse3.fr) **avant le dimanche 15 janvier 2023 23h55 UTC.**

0,25 pt de pénalité par jour de retard sera appliqué au-delà de cette date.

Vous pourrez aussi convenir d'un rendez-vous pour exposer oralement votre travail si vous le désirez.