|  |
| --- |
| Université du québec en outaouais |
| INF4173 - Projet Synthèse |
| Plan de travail |
| Présenté à M. Michal Iglewski  Par |
| **Vincent Crispin Emond & Kossi Ahlin Cornelus Madjri** |
| **1/18/2017** |

# Introduction

Pour se développer, une plante a besoin d’eau, fourni par la pluie, de chaleur fourni par le soleil et d’une terre bonne et fertile. Le rôle d’une mini serre automatisée, est de fournir chacun de ces besoins aux plantes d’une manière absolument autonome et efficace. L’avantage d’une mini serre automatisée est qu’elle peut arroser les plantes et leur donner de la lumière, et de la chaleur même la nuit, grâce à l’arrosage automatique et à l’éclairage de la serre.

La serre de notre projet sera utilisée par Monsieur **Sylvain Delagrange**, professeur au département de sciences naturelles à l’UQO, dans le cadre de ses recherches en plantations à croissance rapide et en régénération forestière.

La serre est munie d’un réseau de capteurs utilisés pour mesurer la température, l’humidité de l’air et du sol ainsi que le taux de CO2 et d’O2. Les capteurs communiquent avec des cartes microcontrôleur appelée « sink » à l’aide d’émetteur/récepteur radio XBEE avec le protocole de routage DigiMesh et le protocole de données MQTT-SN (Sensor Network). Les données des capteurs sont envoyées sur des routeurs XBEE à plus grande portée pour être redirigé sur une Raspberry Pi qui sert en guise de passerelle réseau. La Raspberry Pi fais alors la conversion du protocole MQTT-SN à MQTT et communique avec le serveur web qui sert de MQTT broker, ce serveur communique aussi avec une base de donnée.

La serre compte aussi des activateurs qui servent à contrôler les paramètres environnementaux de celle-ci. Il y a une plaque chauffante, des lumières LED, et un ventilateur.

Le tout est contrôlable à distance grâce à une application web codé avec un produit Java open source appelé OpenHAB.

# Tâches à effectuer

Il s’agirait de travailler sur le module de communication, c’est-à-dire les communications réseau dans la serre. Le protocole MQTT-SN a déjà été implémenté dans la serre mais selon M. Kamel Adi des améliorations peuvent être apportées pour augmenter la qualité de l’implémentation.

En deuxième lieu, un protocole de communication appelé iSerre Sensor Network (iSN) a été défini par les étudiants ayant participé à ce projet auparavant mais faute de temps ils n’ont pas pu compléter son implémentation. Ce protocole est responsable de la communication entre les capteurs et les sinks, il permet par exemple de détecter et de configuré de nouveaux capteurs automatiquement. Le travail consisterait à terminer l’implémentation (codage) de ce protocole sur le matériel ainsi que l’amélioration de sa spécification au besoin.

Finalement, le code de la passerelle réseau qui fait la conversion de MQTT-SN à MQTT pourrait aussi être amélioré.

À noter aussi que l’ajout d’une caméra dans la serre était quelque chose de planifié, si cette caméra est ajouté, il faudra définir et implémenter un protocole pour communiquer avec elle.

# Objectifs du projet

* Améliorer l’efficacité et la stabilité de l’architecture de communication de la serre.
* Implémenter (coder) le protocole iSerre Sensor Network (iSN) sur les cartes microcontrôleurs.
* Pouvoir ajouter ou retirer de nouveaux capteurs dans la serre sans avoir à les configurer manuellement.
* Obtenir de l’expérience de programmation embarquée avec C++ et de programmation de protocoles de communication.

# Méthodologie

* Nous allons commencer par analyser le code qui a été fait et identifié les lacunes ou les points améliorables.
* Nous allons utiliser Kinetis Design Studio (KDS) qui va nous permettre de coder les protocoles réseaux sur les microcontrôleurs sans avoir à perdre du temps à apprendre les moindres détails spécifiques au matériel. Il nous permettra aussi de déboguer directement sur la carte.
* Nous allons nous référer à l’ébauche du protocole iSN écrite dans le rapport des étudiants précédent pour faire son implémentation et/ou amélioration.
* Nous allons tester notre code à l’aide de l’équipement disponible dans les laboratoires afin de confirmer son bon fonctionnement.
* Nous allons vérifier si une implémentation du protocole de communication ZigBee qui est un standard de l’industrie pourrait être envisageable à la place du protocole propriétaire DigiMesh. Cela pourrait améliorer l’interopérabilité des composantes utilisées dans la serre.
* Nous allons nous référé aux documents techniques des standard existants pour comprendre leur fonctionnement. Par exemple : MQTT et DigiMesh
* Nous allons tenter de voir s’il est possible d’utiliser un seul module Xbee par sink au lieu de deux.

# Calendrier

|  |  |  |
| --- | --- | --- |
| Semaine | Date | Travail |
| 1 | 9 janvier | Choix du projet |
| 2 | 16 janvier | Formation des équipes et apprentissage du matériel |
| 3 | 23 janvier | Apprentissage du matériel installation de l’environnement de développement |
| 4 | 30 janvier | Apprentissage des protocoles en place |
| 5 | 6 février | Analyse du code de l’architecture réseau |
| 6 | 13 février | Analyse du code de l’architecture réseau |
| 7 | 20 février | Codage et test (manière itérative) |
| 8 | 27 février | Codage et test (manière itérative)/Rédaction du rapport de progrès |
| 9 | 6 mars | Codage et test (manière itérative) |
| 10 | 13 mars | Codage et test (manière itérative) |
| 11 | 20 mars | Codage et test (manière itérative) |
| 12 | 27 mars | Codage et test (manière itérative) |
| 13 | 3 avril | Finalisation du livrable et préparation de la présentation |
| 14 | 10 avril | Présentation du projet |
| 15 | 17 avril | Finalisation du rapport final |