

Dossier technique – Partie Personnelle

**Serre Automatique**

Système d’acquisition

Ce document regroupe les informations concernant les capteurs de températures et l’interface web « Temps Réel ».

Valentin Chevallier

21/03/2019



Table des matières

**1.** [Présentation du projet 3](#_Toc4140203)

1.1 [Diagramme d’exigences 3](#_Toc4140204)

1.1.1 Synoptique de la réalisation [3](#_Toc4140205)

1.2 Rappel de la tâche de l'étudiant  [3](#_Toc4140206)

[1.2.1 Ensemble des tâches réalisées par l’etudiant 3](#_Toc4140207)

1.3 [Réalisation de la base de données 4](#_Toc4140208)

[2. Mise en Place des Capteurs de Température 5](#_Toc4140210)

2.1 Recherches et Documentation [5](#_Toc4140211)

2.1.1 Schéma de [Cablage 5](#_Toc4140212)

2.1.2 [Solution de développement 6](#_Toc4140213)

2.2 [Etalonnage du Capteur de Témpérature 6](#_Toc4140214)

2.2.1 Graphique de conversion [6](#_Toc4140215)

[3. Réalisation de l’interface web 7](#_Toc4140217)

[3.1 Analyse du besoin 7](#_Toc4140218)

3.1.1 Solution choisie [7](#_Toc4140219)

3.1.2 [Shéma de l’apercu 8](#_Toc4140220)

3.2 Dynamisation de l'interface [9](#_Toc4140221)

3.2.1 [Connexion et interactions avec la base de données 10](#_Toc4140222)

3.2.2 [Affichage Responsive 12](#_Toc4140223)

3.2.3 Aperçu final  [13](#_Toc4140224)

4. [Tests Unitaires 13](#_Toc4140225)

4.1 [Fonction de conversion(courant en température) 13](#_Toc4140226)

4.2 [Problèmes rencontrés 14](#_Toc4140227)

5. Bilan Personnel………………………………………………………………………....................................15

## 1.Présentation du projet

### Diagramme d’exigences

Le diagramme d’exigences ci-dessous représente les exigences fixées par le cahier des charges au tout départ du projet :

Parties concernant mes tâches dans le projet



Dans ce projet composé de 4 étudiants, 4 grand axes de développement se dégagent de ce diagramme :

-Les Mesures des différents capteurs et leur mémorisation

-La Supervision du système via une application Android

-L’interface Web qui doit à la foi posséder une partie temps réel (pour l’affichage en direct des mesures) et une partie périodisation

-La Gestion des capteurs et de leur mémorisation ainsi que la journalisation des évènements du système via une carte Raspberry

Le schéma suivant schématise ces différentes parties :

### Synoptique de la réalisation

Parties concernant mes tâches dans le projet

### 

**Killian Labattut**, l’étudiant n°1 est chargé de configurer et d’écrire le programme de gestion et de journalisation de la carte Raspberry, et également réaliser l’acquisition des mesures du capteur de vent, l’anémomètre.

**Lucas Minaud**, l’étudiant n°2 est chargé de réaliser la partie périodisation de l’interface web, donc l’affichage des mesures selon une période définie par l’utilisateur. Ainsi que l’acquisition de la luminosité et de l’humidité.

**Audran Raynal**, l’étudiant n°4 est chargé de réaliser l’application Android qui servira à la supervision du système pour permettre à l’utilisateur de savoir l’état de ce dernier, et l’acquisition de la pluviométrie.

## 1.2 Rappel de la tâche de l'étudiant

Pour ma part ma tâche au sein du projet, je devais réaliser la partie temps réel de l’interface.

Donc créer une interface épurée, simple de compréhension et d’utilisation qui permet à l’utilisateur de consulter les mesures acquises par les capteurs en temps réel, le tout en créant un site « responsive » et dynamique pour pouvoir récupérer les mesures contenues dans la base de données.

Je devais également acquérir la température ambiante dans la serre ainsi que la température des tuyaux d’eau de chauffage, avec des capteurs fournis au départ du projet.

Notre projet SFL5 étant en parallèle avec le projet SFL6 la base de données des deux projets est commune, j’ai donc été amené à collaborer avec un étudiant du groupe SFL6 pour réaliser cette dernière.

### 1.2.1 Ensemble des tâches réalisées par l’étudiant

A l’approche de la fin du temps imparti voici les taches réalisées :

L’Acquisition des mesures de température, le câblage et le programme d’acquisition sont fonctionnels et cohérent, l’installation sur la serre est également effectuée.

La base de données est correctement configurée et fonctionnelle.

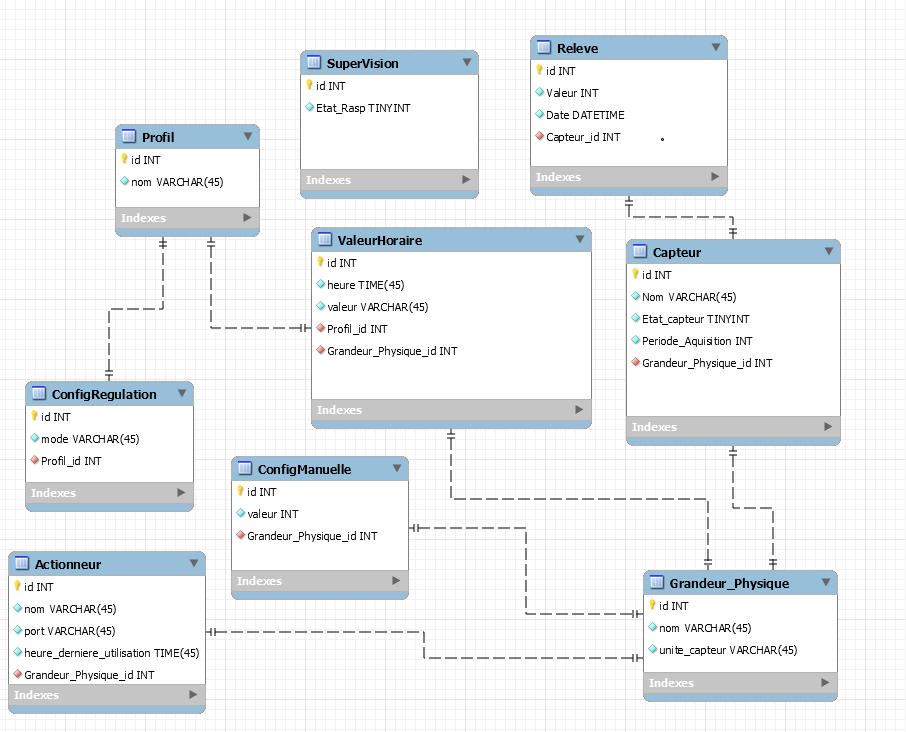
La partie temps réel de l’interface web est fonctionnelle, elle communique avec la base de données et récupère son contenu, elle reste cependant améliorable car elle ne possède par la fonctionnalité d’adaptation automatique.

L’étalonnage du capteur de température est effectué et cohérent.

## 1.3 Réalisation de la base de données

Comme cité précédemment je devais réaliser la base de données en association avec le projet SFL6 plus précisément l’étudiant **Goulven Perrin.**

J’ai donc réalisé sur « MySQL Workbench » un modèle relationnel pour schématiser la base de données :



Les seules tables utilisées par notre projet sont les tables Relevé, Capteur, et Grandeur Physique.

La table que j’utilise en majorité est la table relevé car elle contient les valeurs mesurées selon le capteur id renseigné.

## 2. Mise en Place des Capteurs de Température

### 2.1 Recherches et Documentation

Les deux capteurs de Température ont fait l’objet d’un choix limité, en effet les deux sondes était imposées par le cahier des charges.

Logiquement la plus grosse partie du travail fut de trouver la documentation concernant ces sondes :

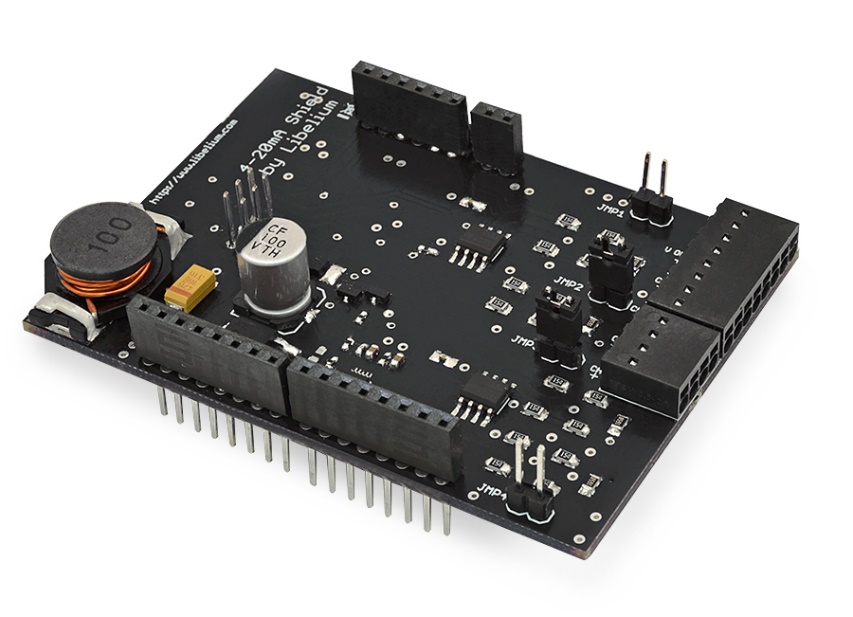


Après plusieurs recherches j’ai trouvé les informations ci-dessus, sachant que les deux sondes sont différentes sur un point : la plage de mesure, le premier va de 0°C à 45 °C (comme sur le document) et l’autre va de 0°C à 100°C.

Grace a la documentation on sait donc que les sondes sont des thermo-résistances, donc c’est la résistance qui évolue en fonction de la température. Elles sont donc analogiques, et la conversion numérique se fait avec une boucle 4-20Ma directement intégrée dans le capteur.

Il a donc fallu trouver un moyen de récupérer l’information numérique sur une carte de traitement : ici une carte Arduino.

La solution trouvée a donc été un Shield Arduino qui embarque une boucle 4-20Ma qui permet de récupérer la valeur de courant et de tension en sortie du capteur :

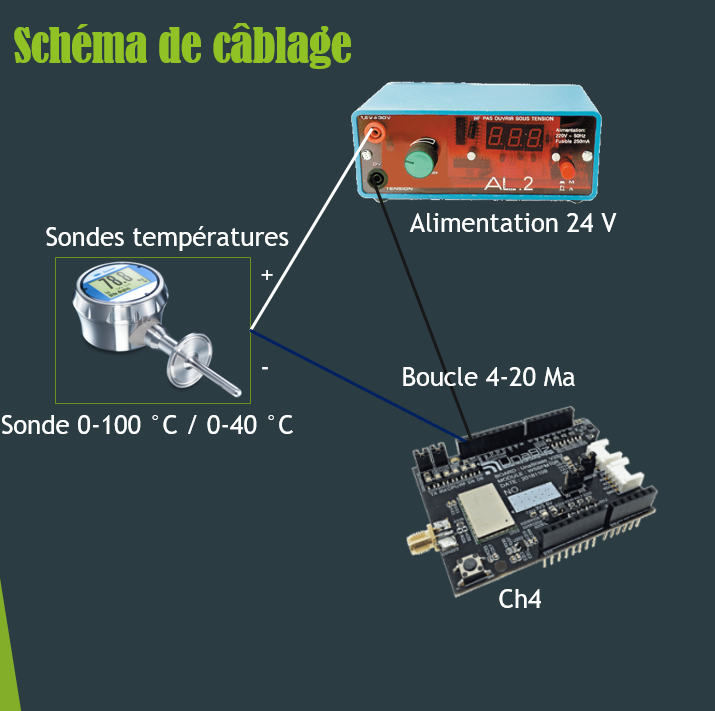


C’est un Shield de la marque « Libellium » qui était déjà présent dans l’établissement, de nouveau il a fallu trouver de la documentation pour trouver comment câbler le capteur.

En sachant que ce dernier a besoin d’une alimentation de 20 à 30 volts pour le fonctionnement nominal.

### 2.1.1 Schéma de câblage

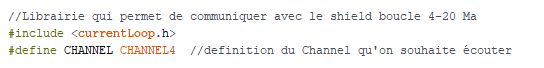
Avec l’aide de la documentation j’ai réalisé un schéma indiquant comment câbler les capteurs avec le Shield et une alimentation externe de 24 V :



Il faut donc créer une boucle d’alimentation avec la masse du capteur relié a la borne plus de l’entrée du Shield et mettre en commun la masse de l’alimentation avec celle du Shield.

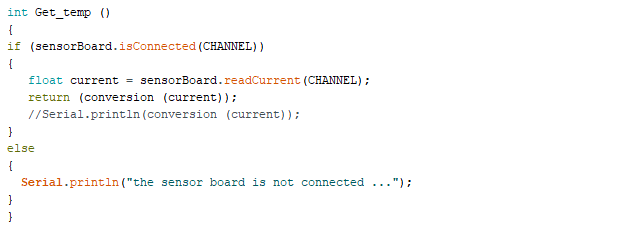
### 2.1.2 Solution de développement

Sur le schéma ci-dessus le « Ch4 » signifie Channel 4 car les entrées du Shield sont en fait des « Channel » numéroté de 1 à 4, cette information est utile pour le programme qui permettra d’interpréter les résultats :



La communication avec le Shield se fait grâce a la librairie montrée ci-dessus nommé « currentLoop.h ».

Cette librairie contient une fonction qui répond à l’objectif souhaité :



Cette fonction nommé « sensorBoard.readCurrent() » retourne la valeur du courant en Ma du Channel renseigné en paramètre.

J’ai également rajouté une fonction pour vérifier si le capteur est bien connecté, cette fonction se nomme « sensorBoard.isConnected() » cette fonction renvoie « True » si un appareil est branché sur le Channel renseigné.

Ma fonction « Get\_temp () » renvoie donc directement la valeur convertie de tension en température grâce à la fonction « conversion » que je vais expliquer ci-dessous.

### 2.2 Etalonnage des Capteurs de Température

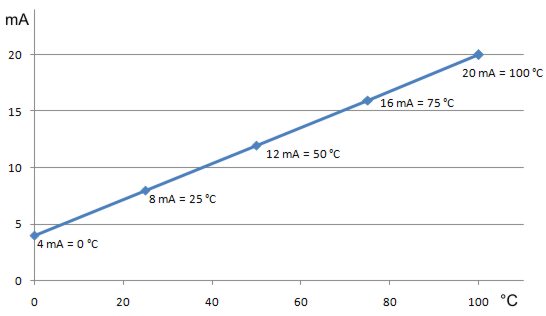
Les sondes fournies sont des sondes pt-100 cela veut dire qu’elles appartiennent à une certaine classe de capteur et souvent les sondes de ce genre utilisent des boucles 4-20 Ma.

Et dans ce cas le 4-20 Ma devient la plage de mesure du capteur, donc quand on a 4 Ma en sortie on est au plus bas mesurable et inversement à 20 Ma c’est la plus haute valeur.

### 2.2.1 Graphique de conversion

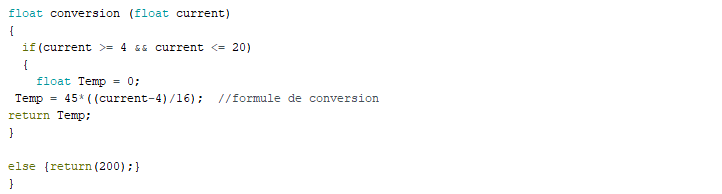
Pour créer une fonction de conversion il a donc fallu que je trouve une formule de conversion.

J’ai donc effectué une série de mesures pour obtenir une courbe de conversion pour trouver l’équation de cette courbe :



Les mesures ici étaient faite avec la sonde 0-100°C et on constate que la courbe est linéaire ce qui donne la formule suivante : **T = Tmax \* (Courant - 4 /16)**

Cette formule ayant été vérifié avec les deux sondes il ne restait plus qu’a la programmer dans une fonction :



La seule valeur qui peut être amenée à changer est la valeur Max (45 ou 100)

La présence d’une boucle « if () » est justifié pour éviter les erreurs, si on sort de la plage 4 – 20Ma la fonction renvoie une valeur excessive et le programme renverra donc une erreur s’il rencontre cette valeur.

Enfin il ne reste plus qu’a installer les capteurs dans la maquette de la serre et tester en conditions réelles et les résultats était cohérent la partie mesure de la température était terminée.

## 3. Réalisation de l’interface web

Au départ du projet la seule contrainte imposée pour l’interface web était le choix de l’outil de développement, les choix étais :

-**Standard**, une page web écrite a la main en HTML, PHP, CSS, Javascript. Une page dynamique stylisée avec les méthodes vues en cours.

-**Framework**, avec un nouvel outil, un framework nommé « Symfony » qui simplifie la réalisation de site web , mais qui demande une connaissance de l’outil et donc dans ce cas présent d’un apprentissage.

### Diagramme de Base de données

VC

Etant donné que les deux groupe (SFL5 et 6) partagent la même base de données

Le projet SFL5 utilise uniquement la table réservée à ce dernier :

Capteur :

Contient le nom du capteur ainsi que la période d’acquisition, c’est-à-dire une valeur de temps ou l’application saura quand effectuer chaque mesure (ex : il ne sert à rien de relevé la température chaque seconde). Ainsi que son état (en fonctionnement ou non).

Relevé :

Contient la grandeur physique mesurée et un format DATETIME pour indiquer la date et l’heure a laquelle a été mesuré cette valeur.

### Diagramme de Déploiement

### 

**Mesure de la température :**

Les deux capteurs étant des sondes RTD c’est-à-dire des capteurs qui mesure la température de manière analogique avec la résistance, L’utilisation d’une boucle 4-20 Ma est obligatoire pour numériser la valeur captée.

Grace au Shield Libellium pour Arduino on peut récupérer la valeur de tension lié à la résistance donc ici pour le capteur de température de l’eau on obtient 4 Ma pour 0 ° C et 20 Ma pour 100 °C.

Une alimentation extérieure est donc nécessaire et après des recherches sur les capteurs ont connais l’alimentation qui est de 24 v.