|  |
| --- |
| Groupe Olivier |
| Dossier technique du projet - Partie personnelle |
| Etudiant 2 : gestion de l’interface web et acquisition de mesure par capteur |

|  |
| --- |
|  |



Table des matières

[I. Situation dans le projet 2](#_Toc9418165)

[1) Synoptique de la réalisation 2](#_Toc9418166)

[2) Rappel des taches de l’étudiant 3](#_Toc9418167)

[3) Contraintes liées au développement 3](#_Toc9418168)

[II. Conception et mise en œuvre 4](#_Toc9418169)

[1) Recherche et achat de capteur 4](#_Toc9418170)

[1.1) Capteur d’intensité lumineuse 4](#_Toc9418171)

[1.2) Capteur d’hygrométrie 6](#_Toc9418172)

[Conclusion recherche et achat de capteur : 6](#_Toc9418173)

[2. Connexion des capteurs à la Raspberry 6](#_Toc9418174)

[**2.1)** Réalisation du diagramme de classes 7](#_Toc9418175)

[III. Développement Arduino 8](#_Toc9418176)

[3. Test Unitaire 9](#_Toc9418177)

[3.1) Test Unitaire de la récupération de valeurs dans la base de données pour l’application Web 9](#_Toc9418178)

[4. 10](#_Toc9418179)

# Situation dans le projet

## Synoptique de la réalisation



Partie du système concernant les tâches auxquelles j’étais assigné.

*Figure 1*

Au sein du projet, j’ai eu pour tâche d’introduire deux capteurs (capteur de luminosité et capteur d’hygrométrie) afin qu’ils puissent transmettre des valeurs à la carte de gestion et dans un second temps j’ai confectionné, avec l’étudiant 3, un interface web permettant visualiser les données récupérées dans la base de données.

## Rappel des taches de l’étudiant

Dans ce projet de serre automatique, J’avais comme objectif de programmer 2 capteurs (capteur de luminosité et capteur d’hygrométrie) afin qu’ils puissent transmettre des valeurs correctes a la carte de gestion, via une carte d’interface et dans un second temps j’ai confectionné, avec l’étudiant 3, un interface web permettant visualiser les données récupérées dans la base de données. Ma partie du site web va être consacré à la visualisation des données en fonction du temps, sous forme de graphique et a échelle variable (jour, semaine mois ou année).

Dans un premier temps, je me suis concentré sur les capteurs que je devais, commander (la maquette nous étant fournie pour le projet, ne comprenait pas les deux capteurs donc je devais m’occuper), programmer et connecter afin qu’ils puissent correctement communiquer avec les équipements liés à la communication.

Pour visualiser toutes les données récoltées, j’ai dû produire un site web avec, pour ma part, une page permettant de visualiser l’évolution des données récupérées par les captures, et ce, avec une période variable (que ce sois une visualisation, en jour, semaine, mois, année).

## Contraintes liées au développement

Les pages Web seront développées en HTLM/PHP/CSS sous NetBeans avec la possibilité d’utiliser symfony.

L’interface Web et la base de données seront hébergées sur un serveur local.

# Conception et mise en œuvre

## Recherche et achat de capteur

### Capteur d’intensité lumineuse

Le capteur d’intensité lumineuse doit être choisi de façon à ce que toutes les attentes soient respectées et pour cela, une recherche approfondie est nécessaire afin de pouvoir acquérir un capteur efficace.

Pour donner suite aux recherches effectuées, un document comparatif est élaboré et rendu au professeur afin qu’il puisse valider la demande et commander le capteur. (Document comparatif ci-dessous)

Choix de capteur d'intensité lumineuse :

Attente Désiré :

Le projet que nous voulons accomplir nous demande de récupérer le niveau d’intensité lumineuse dans la serre.

Pour choisir ce capteur, il a fallu que je vérifie certaines caractéristiques déterminants le capteur que l’on choisira.

-Intensité lumineuse nécessaire au bon développement de la serre :

**1 000 à 10 000 Lux.** Donc le capteur doit avoir une plage de mesure de cette envergure, même encore plus grande pour pouvoir ajuster les volets le plus efficacement possibles.

**-** En France, l’intensité lumineuse, en période de plein ensoleillement, plafonne les **100 000 Lux**. Nous voulons tendre vers cette intensité.

-un moindre coût.

-une marge d’erreur minimale.

-une facilitée de connexion avec une Raspberry/Arduino.

Capteur choisi :

Je me suis donc penché sur 2 capteurs d’intensité lumineuse différents :

Capteur de lumière TSL2591



Caractéristiques :

* + Alimentation : 3,3 à 5 Vcc
  + Signaux : 3,3/5 V
  + Régulateur intégré
  + Interface I2C (adresse 0x29 fixe)
  + Spectre de réponse proche de l'œil humain
  + Plage de mesure : 188 µLux à 88000 Lux
  + Dimensions : 19 x 17 x 3 mm
  + Prix : 8.70€ TTC

Détecteur de lumière Grove 101020030 (Capteur TSL2561)

Caractéristiques :

* + Interface : compatible Grove
  + Sortie : I2C 16 bits 400 kHz
  + Plage de mesure : 0,1 à 40000 lux
  + 3 spectres sélectionnables
  + Dimensions : 20 x 20 mm
  + Température de service : -40°C à +85°C
  + Compatible avec les supports Wrapper 1 x 1
  + Prix : 10.40€ TTC

Conclusion capteur d’intensité lumineuse :

Le capteur TSL2561 à une plage de mesure allant que jusqu’à 40 000 lux (nous voulons tendre vers 100 000 lux), ce qui ne correspond pas à nos attentes. Le prix, lui aussi, reste élevé en comparaison des autres capteurs sur le marché.

Le capteur de lumière TSL2591 est, lui, plus adapté à la situation, il peut atteindre 88 000lux ce qui est bien supérieur à la luminosité voulu pour l’intérieur de la serre. Il admet bien la plage de mesure 1 000 – 10 000 lux, avec des ports adaptés pour une carte Raspberry ou Arduino (compatible via le bus I2C). Ce capteur à un prix bon marché.

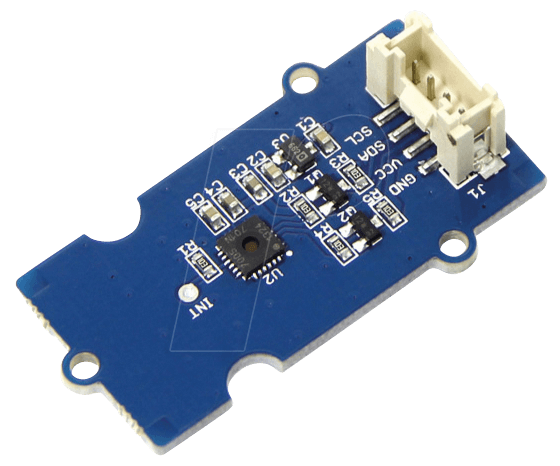
Donc le capteur le plus propice à ce projet serait le capteur de lumière TSL2591.

### Capteur d’hygrométrie

Pour le capteur d’hygrométrie, suite à la livraison de la mini-serre pour notre projet, nous y avons trouvé un capteur correspondant aux caractéristiques demandé par le cahier de charges :

**Caractéristiques attendues :**

• Plage de fonctionnement d’humidité atmosphérique : de 30% à 90% (minimal et maximal en France)  
•Connexion I2C  
•Solidité-Durabilité

**Caractéristiques techniques du capteur reçus :**  
• Large plage de tension de fonctionnement : 3,3 ... 5 V  
• Faible consommation d'énergie : 350 µA pour l'humidité  
• Plage de fonctionnement d'humidité atmosphérique : 0 ... 100 %   
• Humidité : ± 4,5 % (humidité relative)   
• Interface hôte I²C  
• Excellente stabilité sur le long terme

Conclusion recherche et achat de capteur :

Les capteurs idéalement choisir pour ce projet sont tout les deux programmables en python, un langage de programmation très utilisé par les programmeurs et aussi le plus commun pour le codage de capteurs.

## Connexion des capteurs à la Raspberry

Il faut savoir qu’il y a plusieurs façons de connecter des capteurs a une Raspberry et surtout, ce qui est déterminant, il y a aussi plusieurs façons de les programmer.

Suite à beaucoup de recherche, il s’avère que la façon la plus simple que programmer le capteur d’intensité lumineuse est de passer par une carte Arduino et via l’application « Arduino » implémenter un programme dans la carte qui permet la conversion des données envoyé par le capteur pour les envoyer, par la suite à la carte de gestion.

Pour le capteur d’hygrométrie, après quelques recherches, une bibliothèque existe et celle-ci simplifie énormément la tâche. (Description du programme : Partie III. Développement Arduino)

## Réalisation du diagramme de classes

Une image contenant texte, carte

Description générée automatiquement

*Figure 2*

A partir de la manière dont sont transmises les données des capteurs, j’ai pu réaliser le diagramme de classe ci-dessus.

**Les classes CCapteurD’intensiteeLumieuse et CCapteur D’hygrométrie :**

Elles héritent de la classe CCapteur, chaqu’une de ces classes gène un capteur dans la récupération des données des capteurs et la transmission à la CCarteD’interface.

La classe CCapteur

# Développement Arduino

En langage de programmation Python, une multitude de bibliothèque existe pour nous faciliter la programmation d’éléments, telle que des capteurs ou logiciels etc…

Dans un premier temps, la connexion entre le capteur et la carte de gestion (carte Rasberry) devras, par choix, passer par une carte d’interface (carte arduino), pour cela, utiliser l’application Arduino était indispensable pour téléverser le programme développé au sein de la carte exécutant le programme.

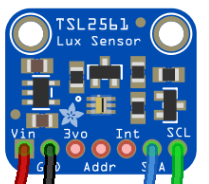
Nous avons à disposition une seule carte Arduino Méga 2560 avec un shield, qui devra accueillir cinq capteurs, donc le programme qui doit être développé pour transmettre les données devra regrouper tous les codes de chaque capteur et faire en sorte que cela soit le plus optimisé possible.

Pour commencer, je me suis occupé du capteur d’intensité lumineuse :

## Capteur d’intensité lumineuse

Pour Connecter ce capteur à la carte d’interface :

Une documentation en ligne est disponible pour faire en sorte que le capteur puisse se connecter en I2C (voir Annexe pour tout connaitre sur la connexion I2C) à la carte Arduino :



*Figure 3*

Suite au câblage du capteur, il faut pouvoir récupérer les données du capteur et faire en sorte qu’elles soient compréhensibles.

Via la bibliothèque de l’application Arduino, un code est déjà proposé pour ce capteur, il nous permet de récupérer les fonctions écrites spécialement a cet effet et de nous proposer un code claire, structurer et fonctionnel.

Une image contenant capture d’écran

Description générée automatiquement

*Figure 4*

Une image contenant capture d’écran

Description générée automatiquementCes deux bibliothèques ci-dessus suffisent pour générer un programme efficace.

*Figure 5*

Ce fichier tsl2591 est la structure du programme voulu et on obtient un code comme celui ci-dessous.

Une image contenant capture d’écran

Description générée automatiquement

Ce code permet d’obtenir les valeurs qu’on souhaite en fonction des réglages qu’on renseigne.



Cette ligne permet de transmettre un numéro pour l’identifiant du capteur et ainsi, le démarquer et le rendre facile d’accès.



La fonction « tsl.getLuminosity(TSL2591\_VISIBLE) » permet de récupérer la valeur de l’intensité lumineuse en volt et de la transformer automatiquement de lux afin de pouvoir la visualiser plus simplement sur le monitor pour les tests comme ci-dessous :

Une image contenant capture d’écran

Description générée automatiquement

# Test Unitaire

## Test Unitaire de la récupération de valeurs dans la base de données pour l’application Web

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Elément testé :** | Test de récupération de valeurs au sein de la base de données afin de les afficher sur le site | | | | | |
| **Objectif du test :** | « Vérifier que le site récupère bien les valeurs de la base de données » | | | | | |
| **Nom du testeur :** | Minaud Lucas | | **Date :** | | Mardi 14 Mai 2019 | |
| **Moyen mis en œuvre :** | Logiciel :  Google Chrome, WinSCP | | Matériel : Ordinateur, serveur | | Outils de développement :  NetBeans IDE | |
| **Procédure de test** | | | | | | |
| **Description du vecteur de test** | | **Résultat attendu** | | **Résultat obtenu** | | **Validation**  **(O/N)** |
| Après connexion au serveur via WinSCP, on lance le fichier : « test\_unitaire.php » se trouvant dans le fichier : /var/www/sfl5-site3 | | Lancement de NetBeans IDE. | | Une image contenant capture d’écran  Description générée automatiquement | | O |
| On tape dans la barre de recherche de chrome :  « [http://10.16.37.161](http://10.16.37.161/phpmyadmin/)  /phpmyadmin/ »  On rentre les identifiants nécessaires :  Utilisateur : root  MDP : serreautomatique  Puis appuyer sur le bouton « Exécuter » | | La page, avec toute les bases de données, s’ouvre | | Une image contenant capture d’écran  Description générée automatiquement | | O |

|  |  |  |  |
| --- | --- | --- | --- |
| On va dans :  ->BDD\_Serre\_Automatique  -> Releve | Une page avec tout les relevés apparait |  | O |
| On tape dans la barre de recherche d’une nouvelle page de chrome : « <http://10.16.37.161/sfl5-site3/test_unitaire.php> » | La page  «test\_unitaire» se lance.  Une page décrivant tout le tableau où sont stocker les relevés apparait |  | O |

# Fiches recette

# Annexe