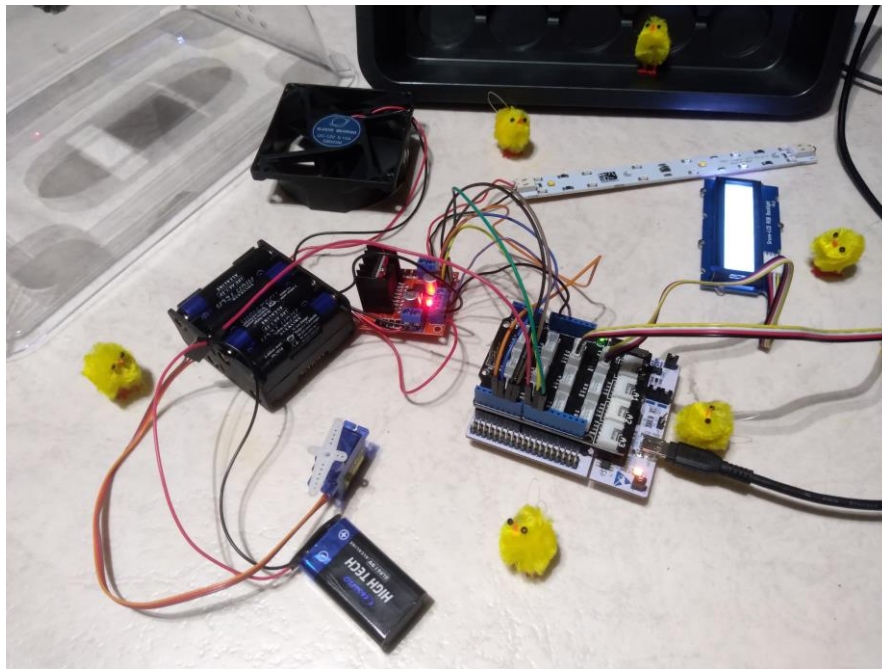


Rapport TP et BE Réalisation Système STM32



Réalisé par :

Nouria KACEMI
Blanche LEGUIN

Numéro Binôme: STM32_G28

Github: STM32_APP_KACEMI_LEGUIN_28

Encadré par: Thierry PERISSE

Année Universitaire 2020_2021

M1 SME, Département EEA, FSI, Université Toulouse III, 118 Route de Narbonne, 31062
Toulouse Cedex 9, France.

SOMMAIRE

I. Introduction

II. Projets de base :

1. Partie Hardware (capteur DHT22)
2. Partie Software
 - 2.1. Configuration carte STM32
 - 2.2. Test et validation
3. Partie Hardware (capteur STH31)
4. Partie Software
 - 4.1. Configuration carte STM32
 - 4.2. Test et validation

III. Bureau d'étude Serre Automatisée

1. Présentation projet
2. Diagramme de contexte
3. Réalisation du système
 - 3.1. Alimentation en 12V
 - 3.2. Module d'ouverture
 - 3.3. Température
 - 3.4. Diagramme de classe
4. Câblage
5. Remarques

IV. Conclusion

I. Introduction :

Ce projet a été réalisé dans le cadre de l'unité d'enseignement Réalisation systèmes par les étudiants du Master SME de l'Université Toulouse III - Paul Sabatier sous la supervision de M. PERISSE. La famille STM32 est une série de microcontrôleurs 32-bits en circuits intégrés réalisés par la société Franco-Italienne STMicroelectronics. Chaque microcontrôleur est constitué d'un cœur de calcul, de mémoire vive (RAM) statique, de mémoire flash (pour le stockage), d'une interface de débogage et de différents périphériques. Le *STM32 F4* qui comporte un Cortex-M4F, un DSP et un FPU est géré par le système libre temps-réel ChibiOS/RT.

Nous devons prendre en main quelques cartes STM32, leurs environnements et logiciels pour la programmation et la création des bibliothèques dans le but de faire fonctionner tous types de capteurs (I2C, One wire ...) ou d'actionneurs.

Pour mener à bien notre projet, nous avons utilisé quelques outils :



II. Projets de base :

Les projets de base sont sous forme de travaux pratiques encadrés par monsieur PERISSE, ils ont pour but de créer des bibliothèques sur une carte STM32 en utilisant CubeIDE pour faire fonctionner 2 types de capteurs de température et d'humidité one wire (DHT22), I2C (STH31) et afficher les mesures sur un afficheur LCD .

Nous allons détailler plus dans la suite en commençant par les matériels utilisés et leurs références.

Capteur DHT22

1. Partie Hardware :

Savoir mesurer la température et l'humidité d'une zone est une information très utile, en particulier pour prédire la météo ou pour stocker du matériel et des aliments dans de bonnes conditions et c'est là qu'on se pose donc la question suivante : comment mesurer (simplement) à la fois la température et l'humidité d'une pièce ou d'un contenant avec une carte STM32 ?

Pour cela, nous avons des matériels suivants :

| Matériels utilisés | Référence |
|--------------------|------------------|
| Carte STM32 | NUCLEO-L152RE |
| Capteur DHT22 | 101020019 |
| Ecran LCD | RGB |
| Bouclier I2C | base shield v2.1 |

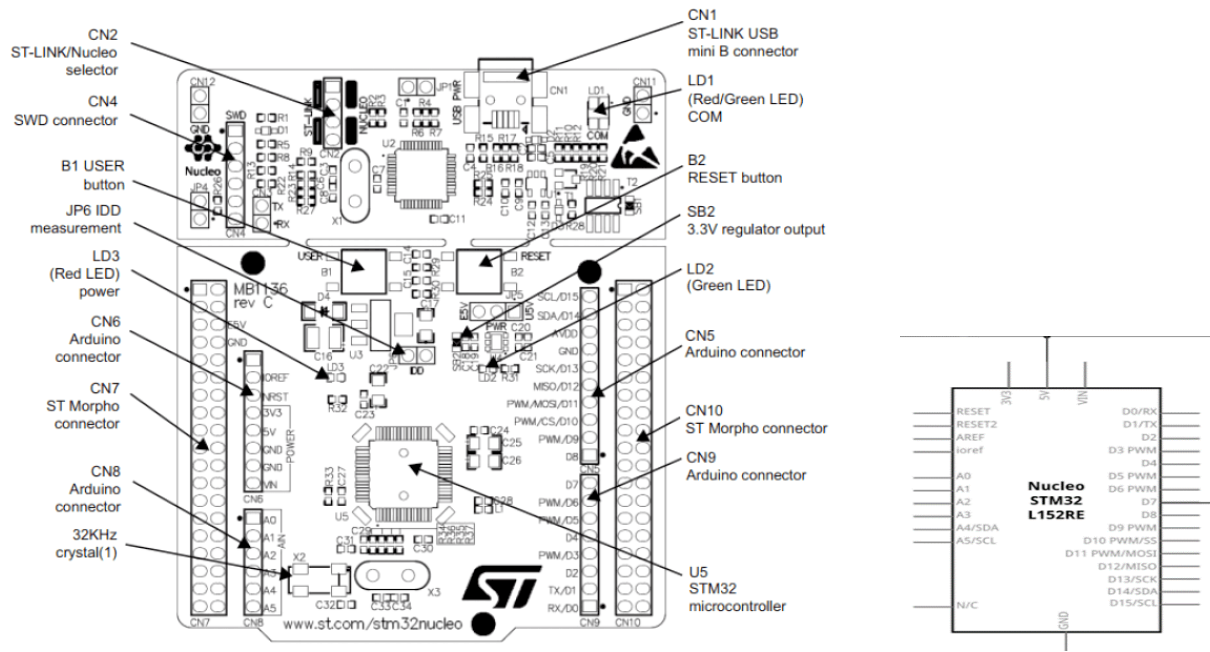
Carte STM32 (NUCLEO-L152RE):

([Fiche technique](#))

Communication: **I2C**

Entrée de tension: **5V**

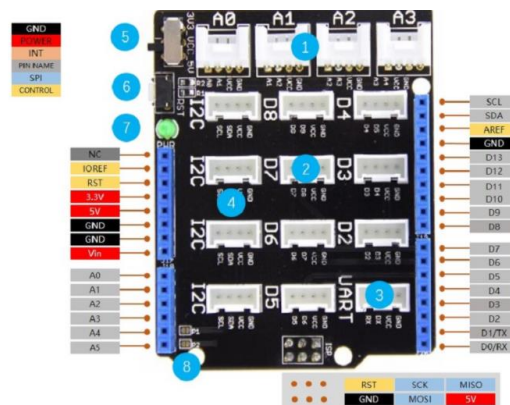
Température: **-20 à 45 °C**



Bouclier I2C :

(*Fiche technique*)

| Paramètre | Valeur/Rang |
|---------------------------|-------------|
| Tension de fonctionnement | 3 à 5 V |
| Température | -25 à 85 °C |
| Ports analogique | 4 |
| Ports digital | 7 |
| Ports UART | 1 |
| Port I2C | 4 |
| Dimension | 69mm x53mm |



Ecran LCD :



| Article | Valeur |
|---------------------------|------------|
| Tension d'entrée | 5V |
| Courant de fonctionnement | <60 m A |
| CGROM | 10880 bits |
| CGRAM | 64x8 bits |
| Adresse LCD I2C | 0x3E |
| Adresse RGV I2C | 0x62 |

Capteur DHT22:

([Fiche technique](#))

| Paramètre | Valeur/rang |
|---------------------------|-------------|
| Tension de fonctionnement | 3.3 à 5 V |
| Température | -40 à 80 °C |
| Résolution (Température) | 0.1 °C |
| Humidité | 5 à 99 % |
| Résolution (Humidité) | 0.1 % |
| Temps de réponse | 6 à 20 s |
| Mode de broche de signal | Numérique |

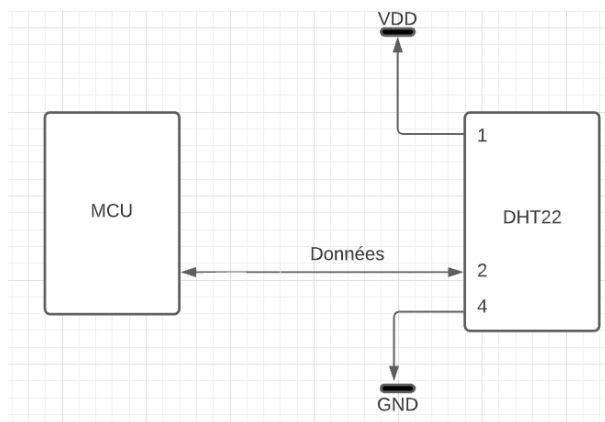


Figure 1 Diagramme de câblage

La broche n°1 est la broche d'alimentation (5 volts ou 3.3 volts).

La broche n°2 est la broche de communication. Celle-ci doit impérativement être reliée à l'alimentation via une résistance de tirage de 4.7K ohms (il s'agit d'une sortie à collecteur ouvert).

La broche n°4 est la masse du capteur (GND).

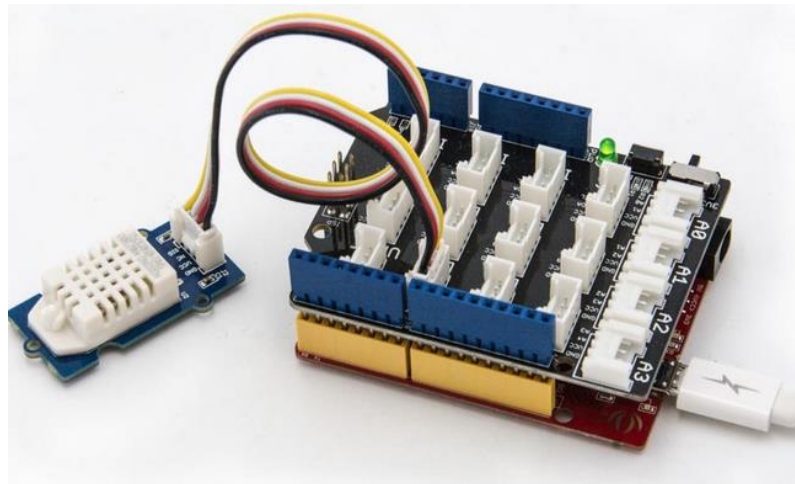


Figure 2 Câblage

Pour le protocole de communication, il se déroule comme le montre la trame suivante :

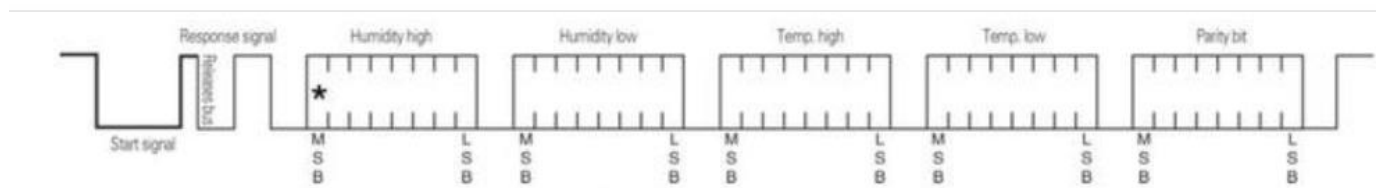


Figure 3 format d'une trame de communication

La communication avec un capteur DHT22 se fait en 3 étapes :

Le microcontrôleur (STM32) réveille le capteur en plaçant la ligne de données à LOW pendant au moins 800µs. Le capteur va se réveiller et préparer une mesure de température et d'humidité. Une fois le temps écoulé, le microcontrôleur va libérer la ligne de données et passer en écoute.

Une fois la ligne de données libérée, le capteur répond en maintenant la ligne de données à LOW pendant 80µs puis à HIGH pendant 80µs.

Le capteur va ensuite transmettre une série de 40 bits (5 octets). Les deux premiers octets contiennent la mesure de l'humidité. Les deux octets suivants contiennent la mesure de la température et le cinquième octet contient une somme de contrôle qui permet de vérifier que les données lues sont correctes.

2. Partie Software :

Après l'étude effectuée sur les caractéristiques des matériels, nous passons à la programmation et la création des bibliothèques pour faire fonctionner le capteur DHT22.

2.1. Configuration carte STM32

Tout d'abord, nous avons configuré la carte STM32 avec CubeIDE qui nous permet de générer une structure du code que nous devons compléter selon nos besoins. Nous avons choisis et initialisé a carte NUCLEO-L152RE, activé le timer TIM6 et le I2C1. Ensuite, nous avons utilisé 2 pins pour la communication I2C1 (PB9 → I2C1_SDA, PB8 → I2C1_SCL) avec la carte, l'écran LCD et le capteur DHT22. Pour le GPIO_Output, nous l'avons mis sur PA1 comme l'illustre la figure ci-dessous :

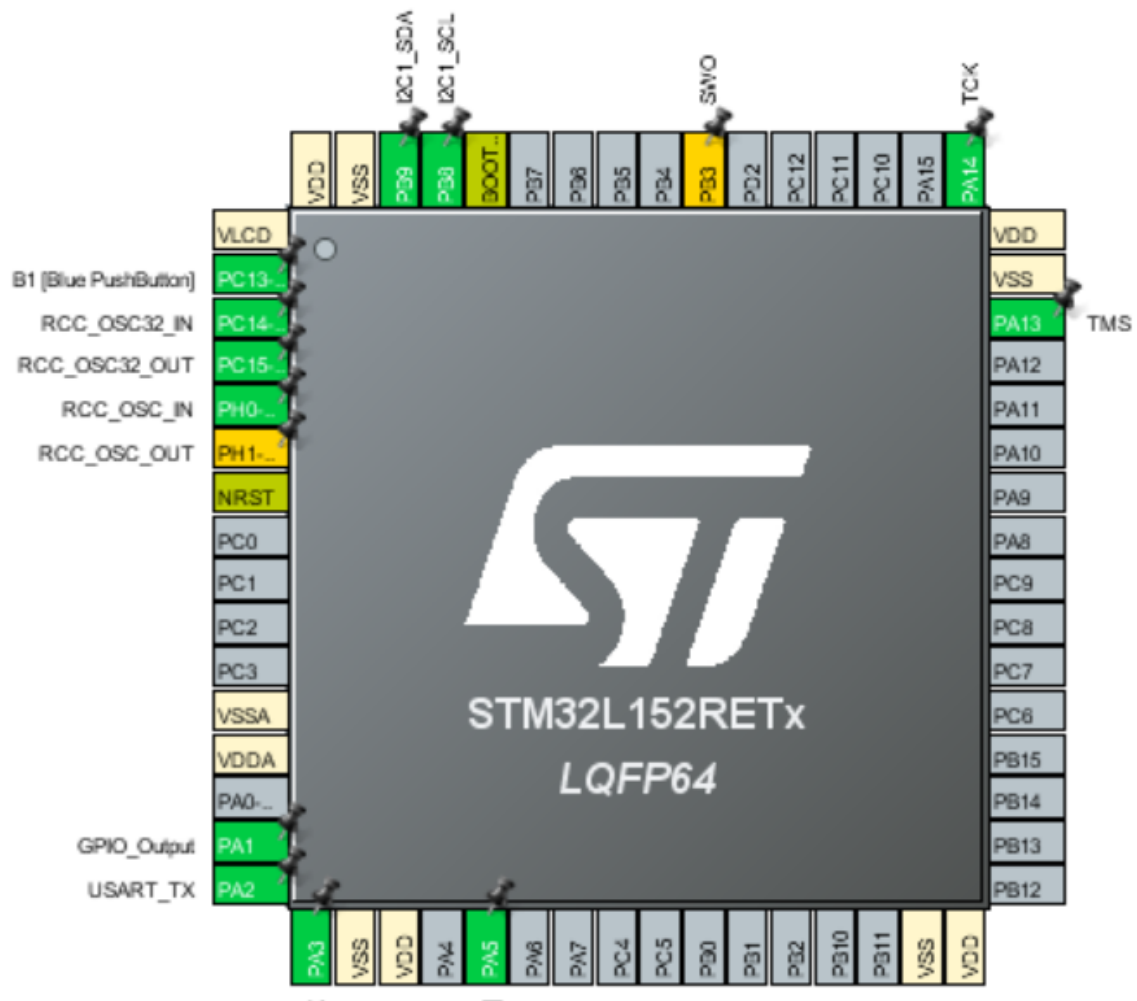


Figure 4 configuration carte STM32

2.2. Test et validation :

Après avoir tout câblé là où il faut, nous passons aux tests.

Grâce aux 16 premiers bits, nous pouvons mesurer l'humidité et aux 16 bit qui se suivent nous mesurons la température en convertissant le nombre binaire obtenu lors de la visualisation sur l'oscilloscope.

D'après l'image ci-dessous, on obtient une humidité de 56.4 %

Et une température de 23,1 C

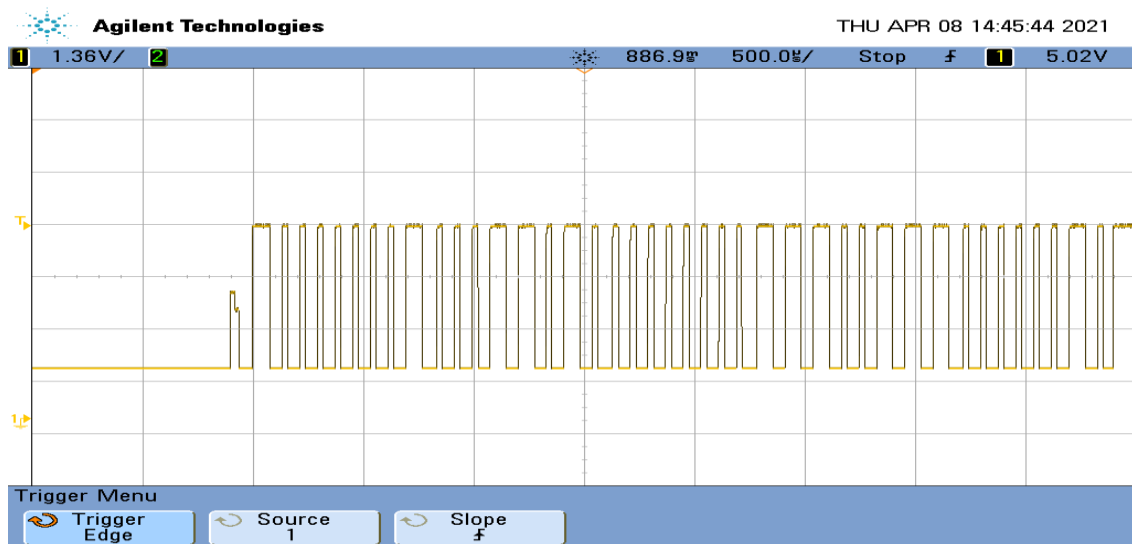
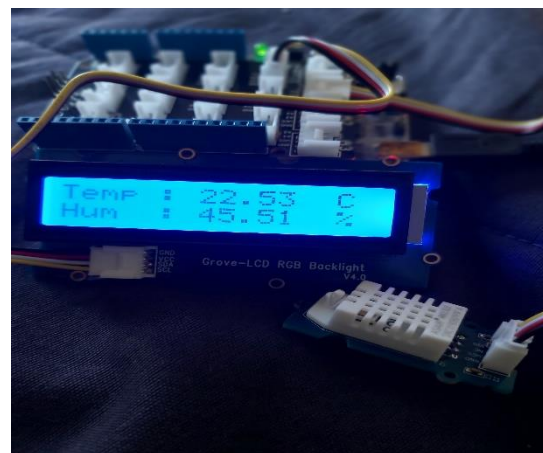
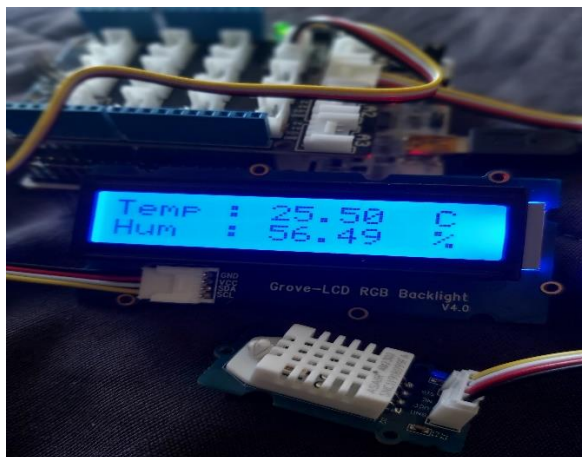


Figure 5 trame visualisée sur l'oscilloscope

Nous arrivons à avoir un résultat pour la mesure de la température en degré et de l'humidité en pourcentage qui augmentent en essayant de tenir le capteur dans la main pour une certaine durée de temps.



Capteur SHT31

3. Partie Hardware:

Capteur STH31:

https://wiki.seeedstudio.com/Grove-TempAndHumi_Sensor-SHT31/

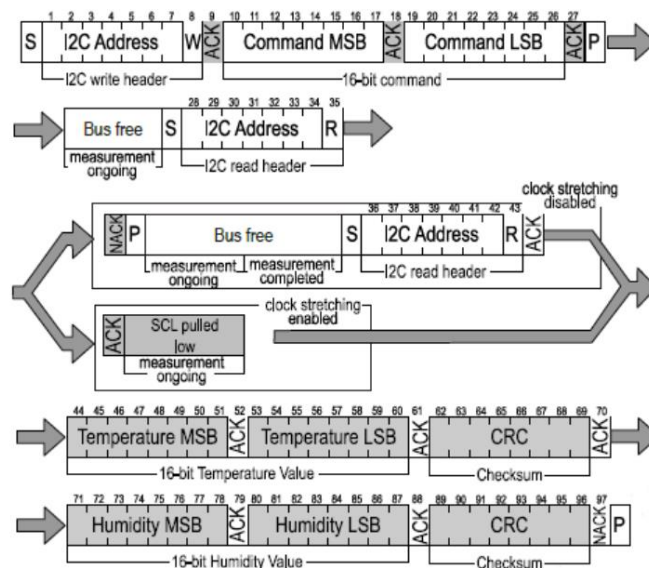
| Paramètre | Valeur/rang |
|---------------------------|--------------|
| Tension de fonctionnement | 3.3 à 5 V |
| Température | -40 à 125 °C |
| Résolution (Température) | ± 0.3 °C |
| Humidité | 5 à 99 % |
| Résolution (Humidité) | ± 2 % |
| Temps de réponse | 8 s |
| Mode de broche de signal | Numérique |

| Condition | | Hex. code | |
|---------------|------------------|-----------|-----|
| Repeatability | Clock stretching | MSB | LSB |
| High | enabled | 0x2C | 06 |
| Medium | | | 0D |
| Low | | | 10 |
| High | disabled | 0x24 | 00 |
| Medium | | | 0B |
| Low | | | 16 |

e.g. 0x2C06: high repeatability measurement with clock stretching enabled



Commandes de mesure pour le mode d'acquisition de données Single Shot, ce mode permet l'acquisition d'une paire de données chaque fois qu'une commande est émise. Chaque paire de données se compose d'une température de 16 bits et de 16 bits valeur d'humidité (dans cet ordre) suivie d'une somme de contrôle CRC comme c'est représenté dans le schéma ci-dessous.



Les données de mesure sont toujours transférées en tant que valeurs 16 bits (entier non signé). Ces valeurs sont déjà linéarisées et compensé par rapport à la température et la tension d'alimentation. La conversion de ces valeurs brutes en des valeurs physiques peut être obtenue en utilisant les formules suivantes :

<https://datasheetspdf.com/pdf/1116083/Sensirion/SHT31-DIS/1>

$$T[^\circ\text{C}] = -45 + 175 * \frac{\text{St}}{2^{16} - 1}$$

$$\text{RH}[\%] = 100 * \frac{\text{SRH}}{2^{16} - 1}$$

4. Partie Software :

Nous passons à la programmation et la création des bibliothèques pour faire fonctionner le capteur SHT31 I2C.

Nous avons une nouvelle fois configuré la carte STM32 avec CubeIDE qui nous permet de générer une structure du code. On choisi NUCLEO-L152RE, on active le timer TIM2 et le I2C1. Nous allons utiliser les mêmes pins que précédemment pour la communication I2C1 (PB9 → I2C1_SDA, PB8 → I2C1_SCL) avec la carte, l'écran LCD et le capteur SHT31 I2C. Pour le GPIO_Output, nous l'avons mis sur PA8 comme ci-dessous :

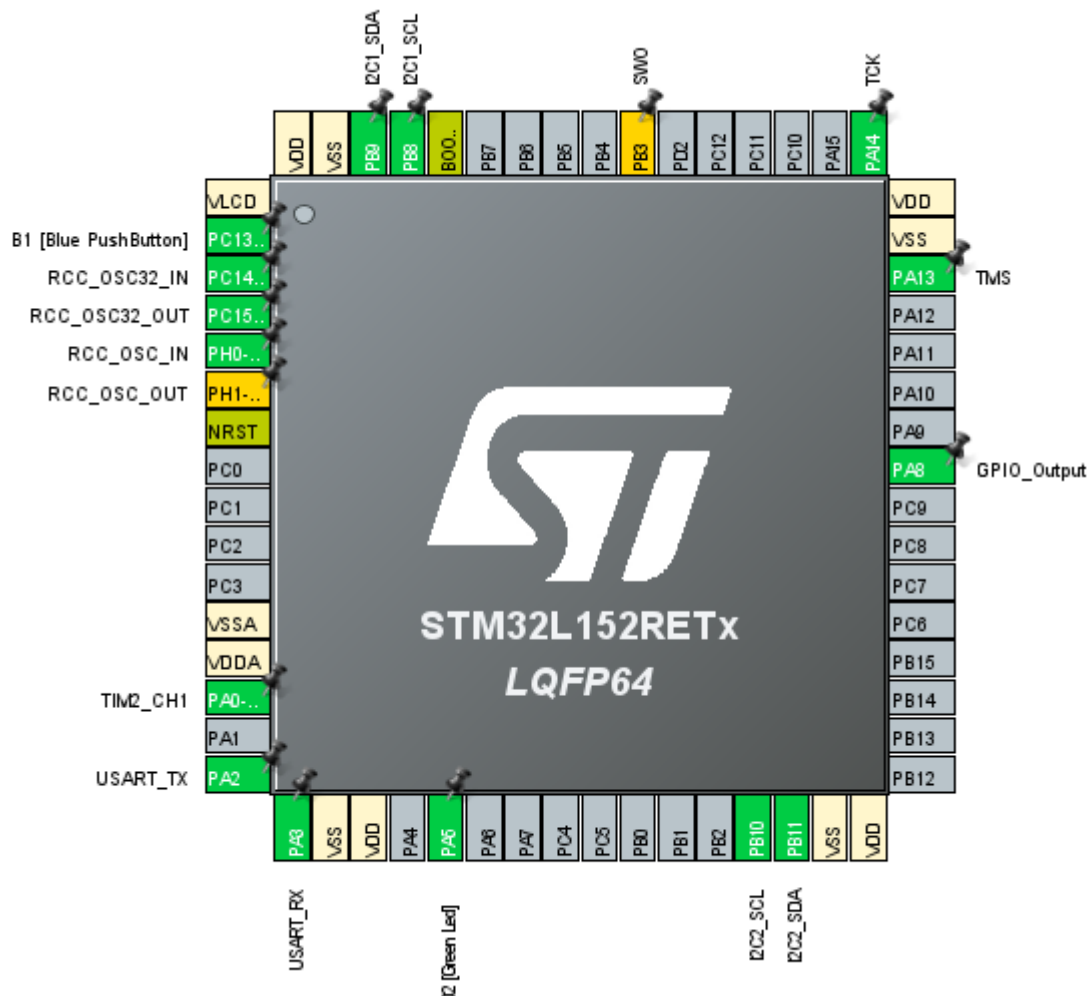
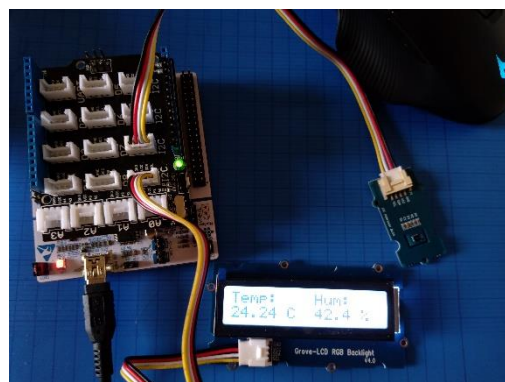
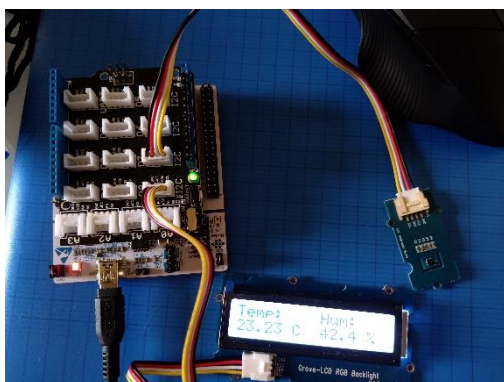


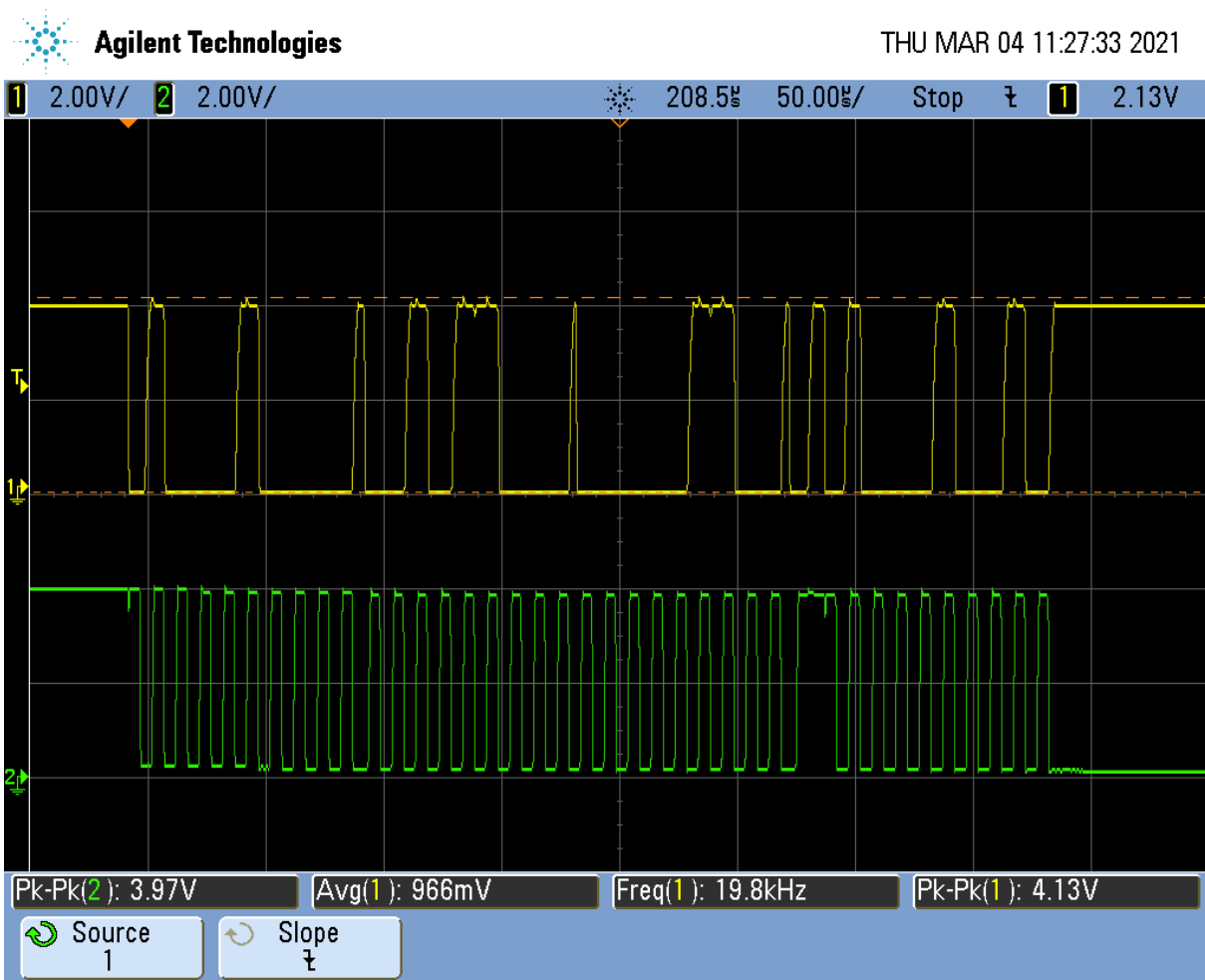
Figure 6 Configuration carte stm32

Test et validation :

Après avoir tout câblé, nous passons aux tests. Nous arrivons à avoir un résultat pour la mesure de la température en degré et de l'humidité en pourcentage qui augmentent en essayant de tenir le capteur dans la main pour une certaine durée de temps.



Quand on regarde les sorties plus attentivement on obtient le résultat suivant



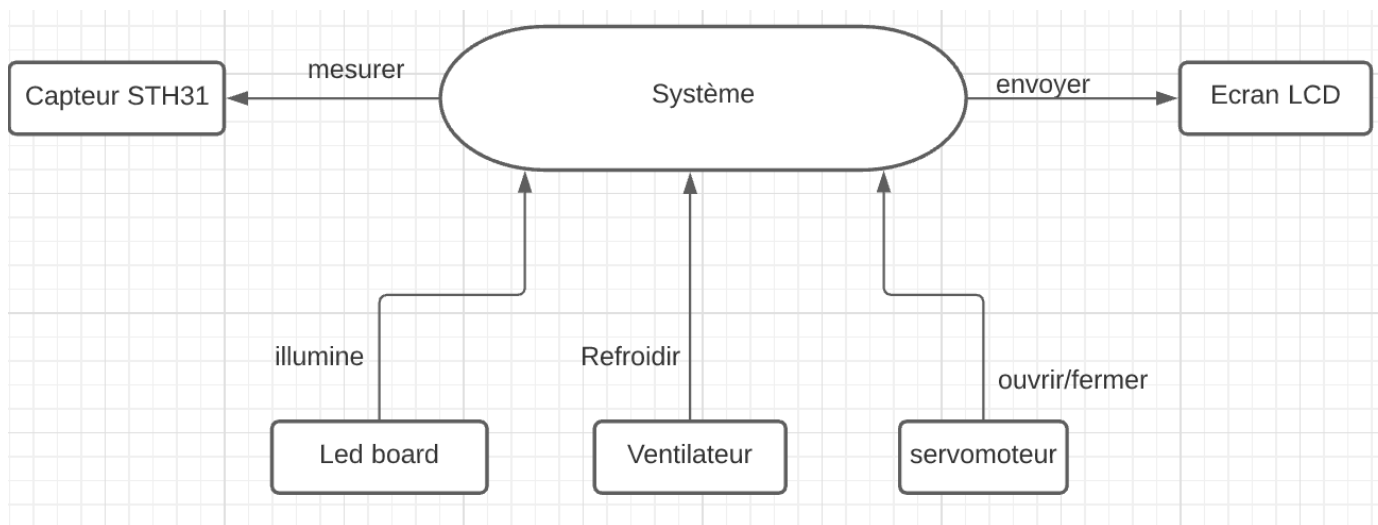
III. Bureau d'étude Serre Automatisée

1. Présentation projet :

Notre projet est la réalisation d'une serre connecté. Elle aura pour but de maintenir une température souhaité et de permettre de surveiller le taux d'humidité, cela en ventilant et en allumant une led board.

2. Diagramme de contexte

Afin de bien mener notre projet, nous l'avons met en œuvre et modélisé. Donc nous avons réalisé le diagramme suivant :



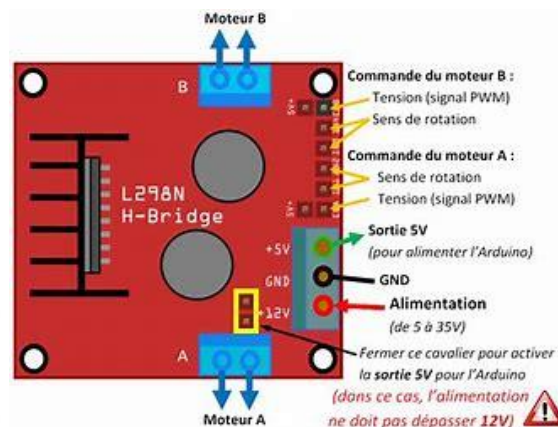
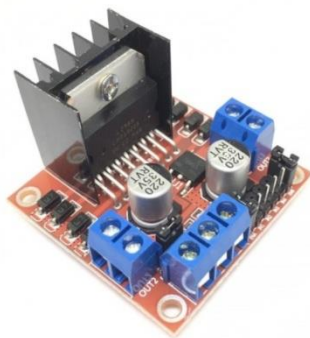
3. Réalisation du système :

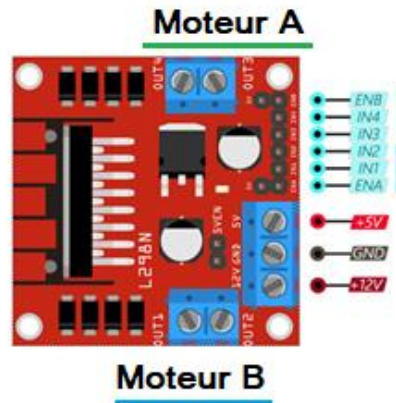
On peut séparer notre projet en 3 parties :

3.1. Alimentation en 12V

Module L298N

Il nous a été nécessaire d'utiliser une alimentation externe pour alimenter le *board led* et le ventilateur. En effet le voltage maximum du STM32 étant de 5V et cela n'était pas suffisant pour ces deux éléments car ils nécessitent une alimentation de 12V. Pour cela nous avons dû utiliser un *Module L298N - Contrôler Moteur avec Arduino*, illustré comme ci-dessous.








Les éléments en charge du bandeau de leds est représenté par la parite A, et le ventilateur par la partie B. La masse est reliée à celle de l'arduino et la sortie 5V à la carte STM32. Cet élément permet par ailleurs de rendre le système indépendant en énergie (il n'a plus besoin d'être alimenté par le port USB).

Dans le cas du ventilateur le sens de rotation et donc influencé par les contrôle et devront être précisé dans le code. Quant à la tension de signal, elle va permettre d'influencer la vitesse de rotation du ventilateur

Dans le cas du *led board*, le sens de rotation n'a pas d'importance donc sera géré par un seul port. Mais la sortie PWM va quant à elle influencer la luminosité du système. Ce qui par ailleurs révèle une sous-alimentation de notre système car la brillance idéale de notre *board led* est comme montré par le tableau suivant.

| intensité lumineuse faible | intensité lumineuse maximum | intensité lumineuse idéale |
|---|---|---|
|  |  |  |

Nous pouvons ensuite détailler plus profondément les autres éléments de cette partie du système, nous avons donc :

Alimentation

Alimentation 12V



Ventilateur



Sleeve Bearing S8025M

Tension de fonctionnement DC-12V

Courant 0.15A

Dimension 40x40x10 mm

Il s'agit de l'élément que nous utiliserons pour refroidir notre système. Il sera par défaut éteint et ne sera sollicité quand cas de trop forte chaleur.

Bandeau Led



TRU Light LED-Board V1.2 35/2017 (marque Tru component)

- *Angle de diffusion : 120°*
- *Tension de fonctionnement : 10.6V*
- *Courant d'alimentation : 700mA*
- *Dimension : (L x l x H) 18 cm x 16 mm x 7 mm*
- *Courant de service par unité ; 700mA*
- *Puissance consommée/consommée par unité/Puissance: 7.5W*
- *LED: 16 Nombre de LEDs par unité*
- *Température min : v-20°C*
- *Température max : +65°C*
- *Eclairage : Lumière mixte rouge/blanc + UV*

Il s'agit de Led conçu spécialement pour l'horticulture qui favorise la croissance et la floraison des plantes avec les UV. Elle émet également une légère chaleur et permet de participer au maintien de la température de la serre.

3.2. Module d'ouverture

Pour permettre la bonne circulation de l'air en fonction de la température il faut être capable d'ouvrir et de fermer l'ouverture présente sur la serre utilisé. Elle à la forme ci-dessous :



Cette ouverture à néanmoins le problème d'être concave. Pour pallier à ça, nous avons dû réaliser une impression 3D afin de pouvoir l'ouvrir à partir d'un servomoteur.

Il nous a falli designer sur impression 3D une pièce que l'on a monté par la suite sur un servomoteur. Il est composé de 4 pièces qui ont été assemblé manuellement :

| Mécanisme vu du dessus | Mécanisme vu du dessous | Mécanisme vu de profil |
|------------------------|-------------------------|------------------------|
| | | |



Le seul élément électronique qui sera utilisé ici est un servomoteur :

Servomoteur:

TowerPro SG90

- Dimensions : 22 x 11,5 x 27 mm
- Vitesse: 0.12 sec/60° sous 4.8V
- Couple: 1.2Kg/cm sous 4.8V
- Tension: 4.8V - 6V
- Prise type Graupner UNI



Nous allons utiliser ce servomoteur dans le cadre de l'ouverture et la fermeture de la trappe situé au-dessus de la serre. Il devait être placé à côté de l'ouverture mais à cause de problème de longueur de câble il n'a pas pu être fixé à sa place. Nous avons sélectionné cet élément spécifiquement car nous devons ouvrir la trappe selon un angle précis.

Il est en adéquation avec le ventilateur pour permettre l'aération. Il sera par défaut éteints et ne sera sollicité quand cas de trop forte chaleur.

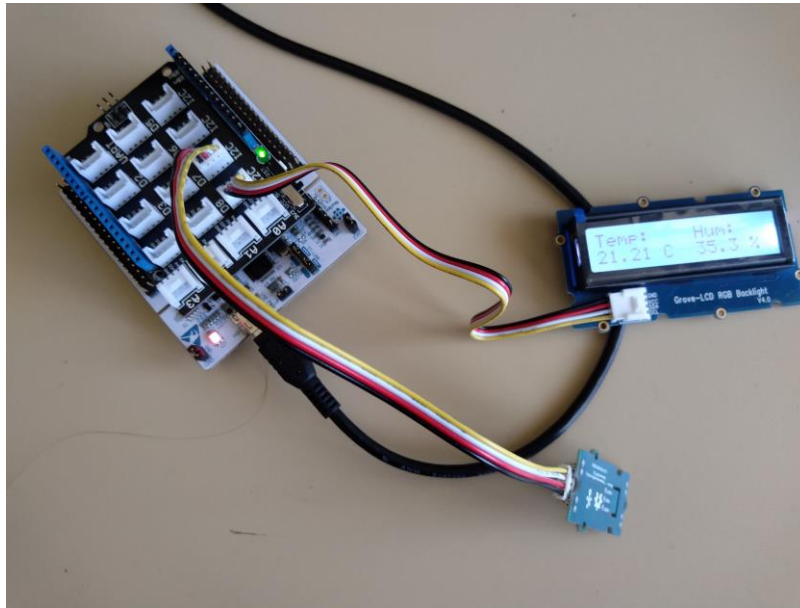
Cependant il a nécessité une alimentation extérieure sous la forme d'une pile 9Vcédement on utilise le shield pour le lcd et le capteur, nous permet d'avoir la température et l'humidité dans la serre.

3.3. Module Température

Ce module a été directement repris de la première partie du travail effectué dans le cadre de notre projet. Il consiste donc à récupérer les mesures du capteur et les afficher sur l'écran LCD. Nous réutilisons le Shield le SHT31 et l'écran LCD. C'est le capteur qui va impacter tous les autres composants du système et est l'élément central de notre projet.

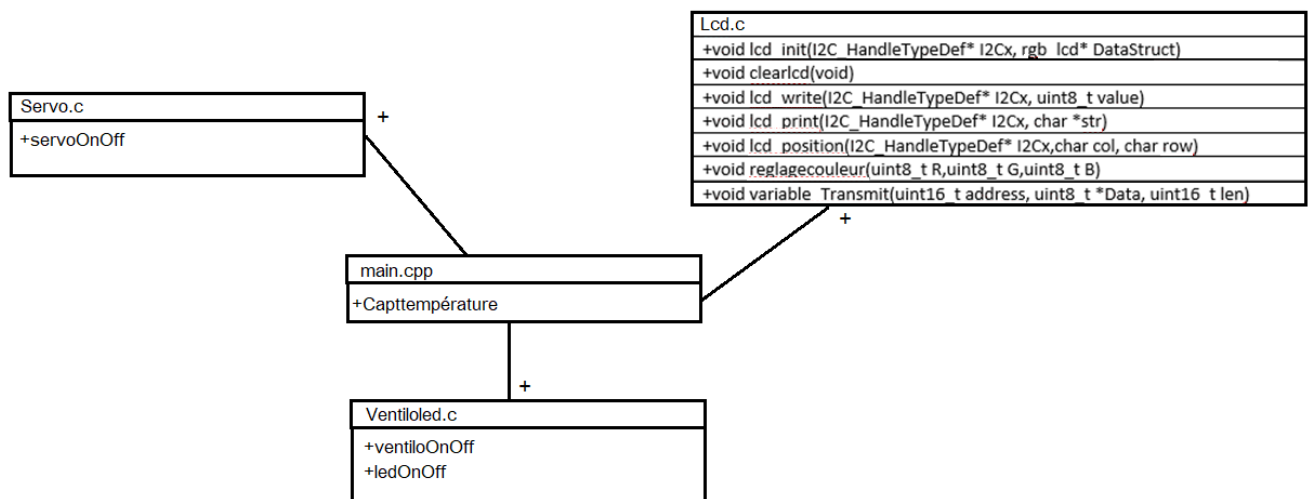
Les détails de ce capteur ne seront pas répétés ici et les détails sont disponibles en première partie de ce rapport.





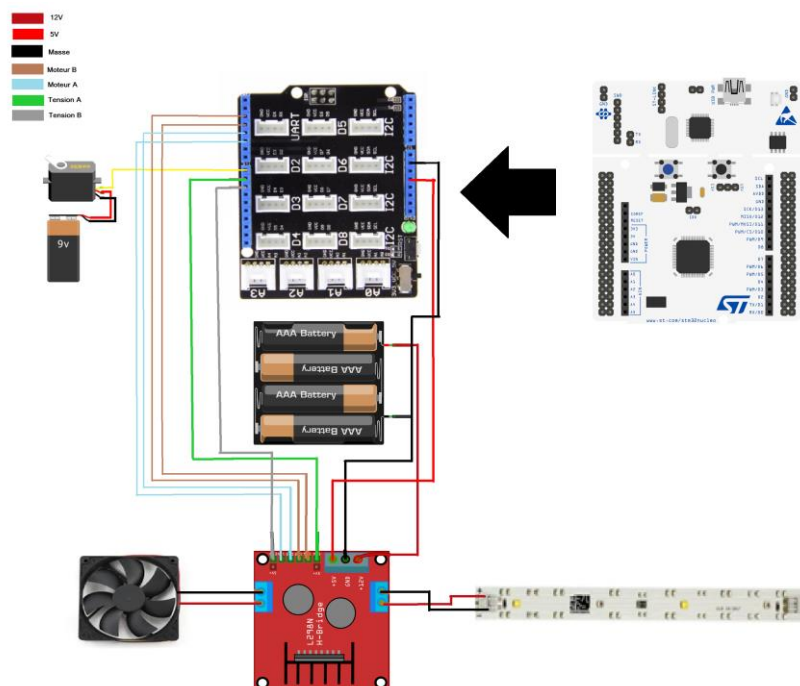
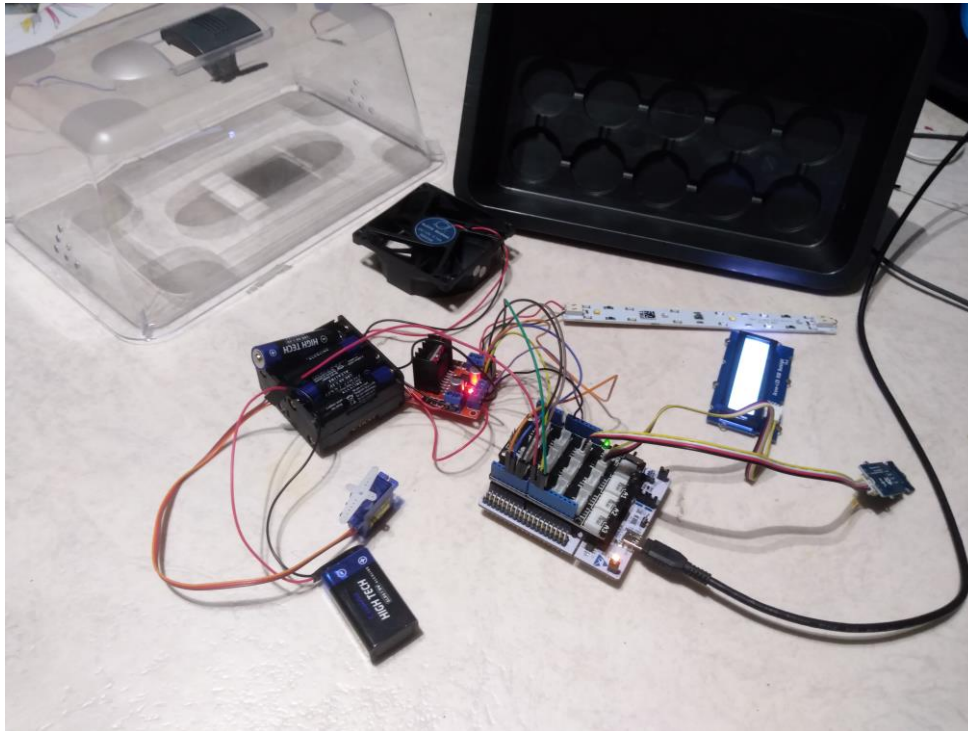
3.4. Diagramme de classe :

Nous allons présenter les classes et les interfaces du système ainsi que leurs relations, chaque classe représente un ensemble de fonctions et de données qui sont liées, elle permet de modéliser un programme et ainsi de découper une tâche complexe en plusieurs petits travaux simples.



La totalité du code est disponible sur [gitHub](#)

4. Câblage :



5. Remarques

Nous sommes arrivées à une serre autonome et fonctionnelle qui respecte la température ambiante précisé en code, néanmoins ce projet reste améliorable. Tout d'abord en ce qui concerne l'alimentation. Il existe 2 point d'alimentation (9V et 12V), ce qui n'est pas optimale et l'a taille du système STM32 est trop imposante pour le volume de serre choisie. Une solution serait de trouver une autre source d'alimentation plus élevé, pour également palier au problème de sous-alimentation, et plus petite, afin de réunir les alimentations en une seule. Pour la taille il faudrait chercher des composant plus petit, mais afin qu'il ne soit pas visible, un sur-socle en impression 3D serait envisageable. Les composant serait ainsi plus visible et ne risquerait plus d'être débranché. Le rajout envisageable serait également l'addition d'u tapis chauffant 12V, mais cela impliquerait un changement du *Module L298N* pour un module avec plus d'entrée comme une Carte de contrôle RAMPS.

IV. Conclusion

Bien que fonctionnel, ce projet permet encore beaucoup d'amélioration varié. Il nous a permis de découvrir beaucoup de subtilité du microcontrôleur STM32 et aussi de nous faire prendre.