**Lycée Raymond Queneau**

18

**Projet final BTS SN**

**La maison du futur - Domotique**

**Chauveau Aurélien**

SOMMAIRE

# 1 - Présentation................................................................................p3

# 2 - Les tests unitaires......................................................................p5

## Le thermomètre..............................................................................p5

## L'hygromètre...................................................................................p6

## Capteur de qualité d'air................................................................p8

## Carte SD............................................................................................p10

## Radiateur..........................................................................................p12

## Module RTC.....................................................................................p14

## Compteur Electrique.....................................................................p15

# 3 - Regroupement des composants..............................................p17

## Le compteur électrique, la carte SD et le module RTC.........p17

## Le thermomètre,l'hygromètre, capteur de qualité d'air et le radiateur.....................................................................................................p20

## Regroupement final des composants.........................................p21

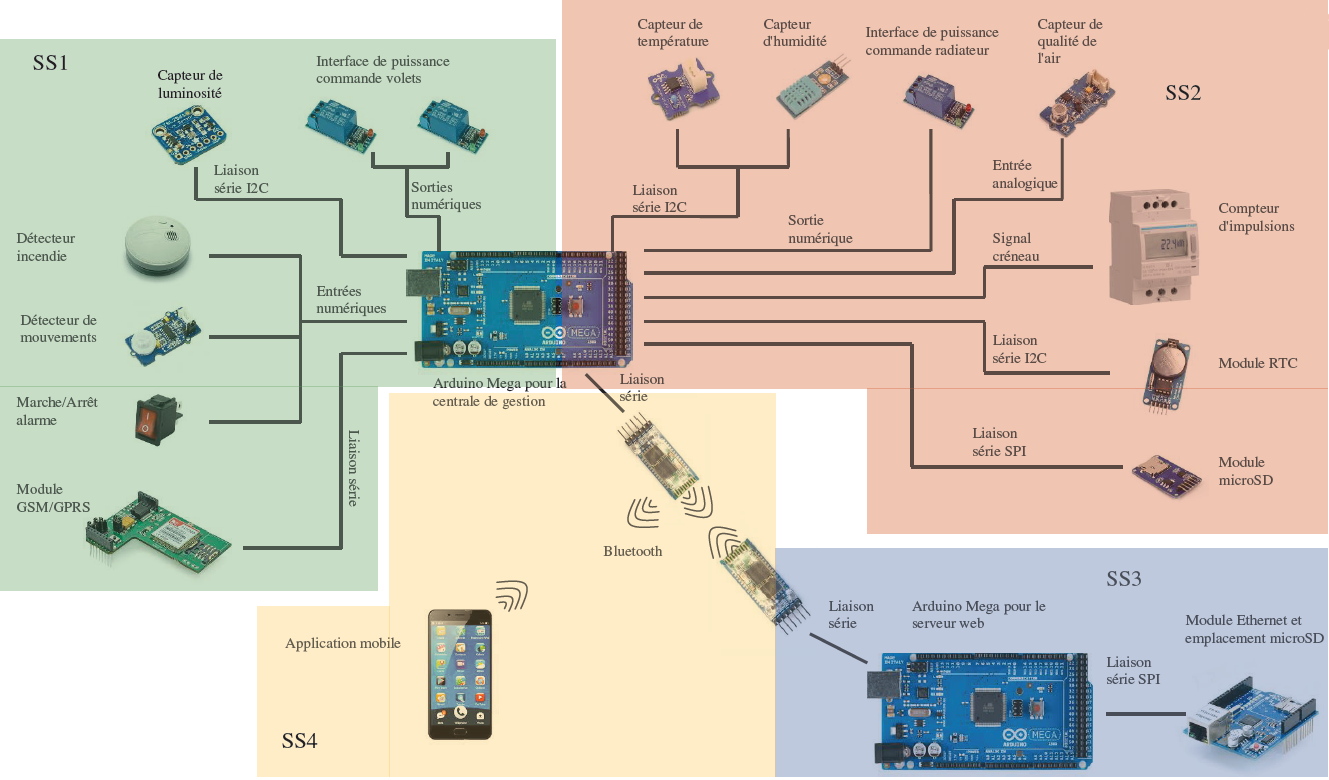
# 4 - Regroupement des différentes parties..................................p27

## La bibliothèque capteur.h............................................................p29

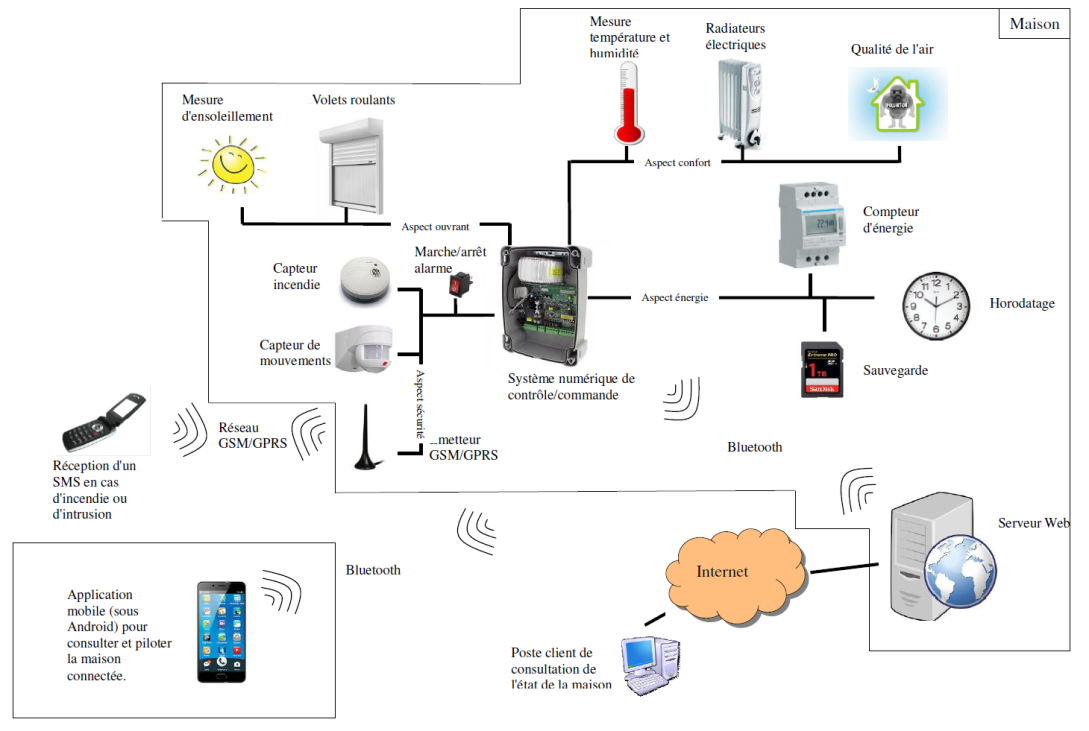
## La bibliothèque gestionMaison.h...............................................p30

## Fonction d'envoi des trames........................................................p31

1 - PRÉSENTATION



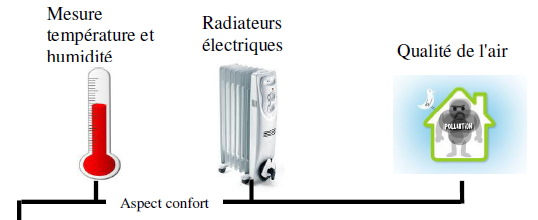
Le projet domotique (la maison du futur), consiste à automatiser une maison. Pour réaliser cela, nous sommes 4 étudiants, répartis sur différents aspects du projet.

 L'étudiant 1 se chargera du contrôle des volets, de la lumière et de la partie sécurité qui comprend un capteur de mouvement et de fumée.

L'étudiant 3 sera chargé de créer un serveur web qui affichera les valeurs envoyées par lesétudiants 1 et 2.

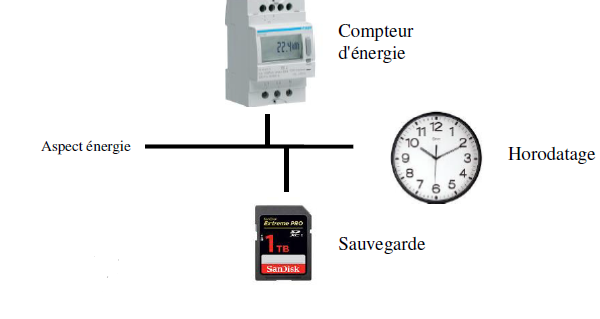
L'étudiant 4 devra créerune application mobile qui permettra de contrôler les actionneurs des étudiants 1 et 2 et d'afficher les valeurs envoyées par les étudiants 1 et 2.

Les étudiants 3 et 4 auront une partie commune sur la communication Bluetooth qui servira à la communication entre la centrale de gestion (carteArduino), l'application et le serveur web.

 En tant qu'étudiant 2, je suis chargé de l'aspect confort et consommation.

La partie confort du projet consiste à récupérer plusieurs valeurs, utiles au propriétaire, telles que : la température, le taux d'humidité ou encore la qualité de l'air. Nous aurons pour cela différents capteurs à choisir et à mettre en marche afin de réaliser cette collecte des valeurs. Nous devrons également piloterun relais via une carte Arduino. Ce dernier nous permettra par la suite de contrôler et d'automatiser un radiateur. Pour automatiser le radiateur nous devrons récupérer une valeur (latempérature), saisie par l'utilisateur, puis nous devrons égaliser la température de la maison à la valeur récupérée. Une plage d'hystérésis est donc nécessaire, elle sera de 1°C.

Chaque valeur récupérée devra être envoyéà intervalle régulier au serveur web et à l'application mobile. La température et l'humidité devront être envoyées toutes les 5secondes et la qualité de l'air toutes les 10 secondes.



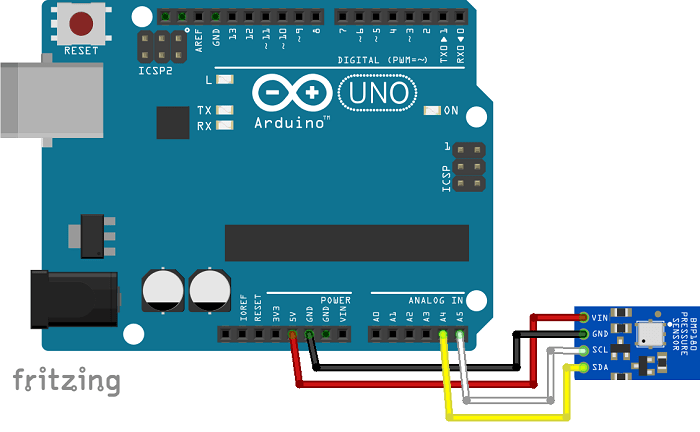
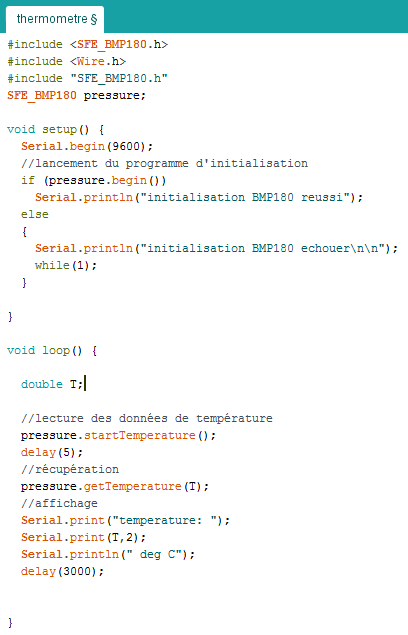
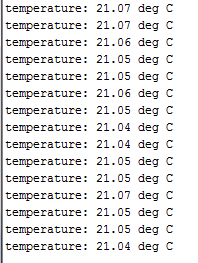
L'aspect consommation d'énergie est aussi à ma charge. Pour se faire nous avons un compteur d'énergie, une horloge et une carte SD. Tout cela devra nous permettre d'horodater[[1]](#footnote-2) et de sauvegarder sur une carte SD la consommation en watt par heure.

Une partie commune entre l'étudiant 1 et 2nous charge de regrouper nos codes dans une seule carte Arduino et de créer un objet pour stocker le maximum de valeur utile au bon fonctionnement du projet. Le code final devra être lisible, intuitif et donc surtout compréhensible.

Nous devrons aussi réunir tous les étudiants du groupe afin de nous mettre d'accord sur le protocole de la trame de données à envoyer au serveur web et à l'application.

2 - LES TESTS UNITAIRES

# LE THERMOMÈTRE :

 Le capteur de pression BMP180 étant équipé d'un capteur de température, nous l'avons utilisé pour notre projet. J'ai par la suite récupéré une bibliothèque sur GITHUB[[2]](#footnote-3) pour commander le capteur. J'ai, ensuite, relié le BMP180 à une carte Arduino, créé un code simple et inclus la bibliothèque à mon code pour réaliser des tests. La communication entre la carte Arduino et notre capteur se fait par I2C.

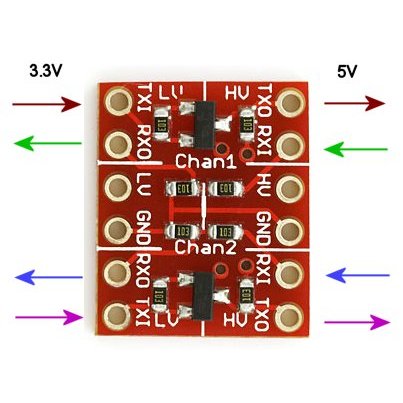
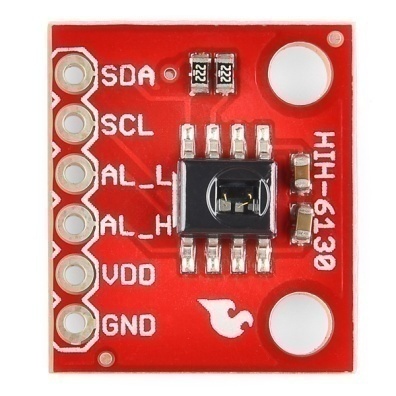
## Valeur obtenue

## Montage

## Code de test unitaire

Le capteur fonctionne bien, il est très sensible à la chaleur qu'on lui transmet lorsqu'on le touche mais a du mal à la dissiper. Il prend très vite 1degrés en 1seconde jusqu'à 6-7 degrés en 10 secondes.

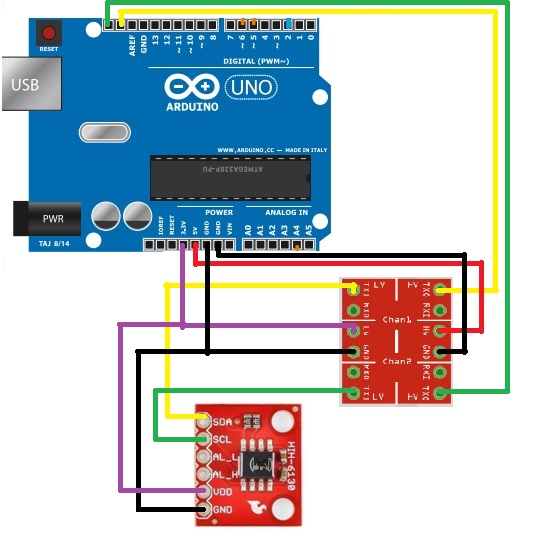
# L'HYGROMÈTRE :

 Un hygromètre est un capteur qui sert à mesurer le taux d''humidité dans l'air. Pour se faire nous allons utiliser le capteur HIH6130 qui sert aussi de thermomètre et une bibliothèque trouvée sur GITHUB pour gagner du temps. Le HIH6130 mesure le taux d'humidité a plus ou moins 5% entre 10% et 90% ainsi que la température entre 5°C et 50°C. Nous allons aussi avoir besoin d'un convertisseur de tension, qui nous servira à ne pas cramer notre capteur d'humidité/température avec le bus I2C qui fonctionne en 5V alors que notre HIH6130 reçoit du 3,3V.

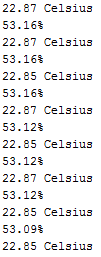
## Montage du système

## HIH6130

## Convertisseur



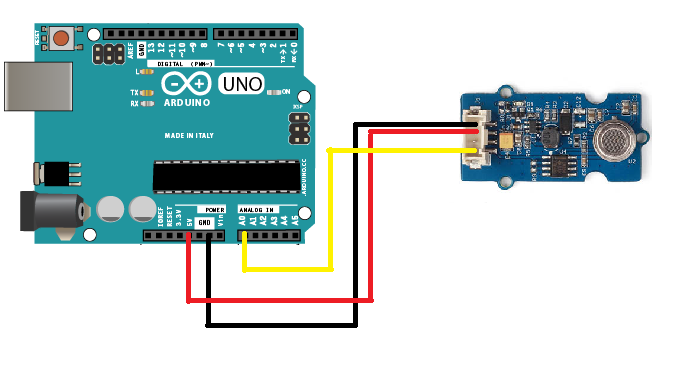
Le convertisseur nous permet de passer de 5V à 3,3V pour passer de la carte Arduino au capteur et vice-versa.

 Le code de test unitaire m’a permis de vérifier les valeurs de température du capteur, qui sont correctes. Nous allons donc plutôt utiliser le HIH6130 pour pouvoir à la fois récupérer la température et le taux d'humidité, au lieu d'avoir deux capteurs distincts.

## Valeur obtenue

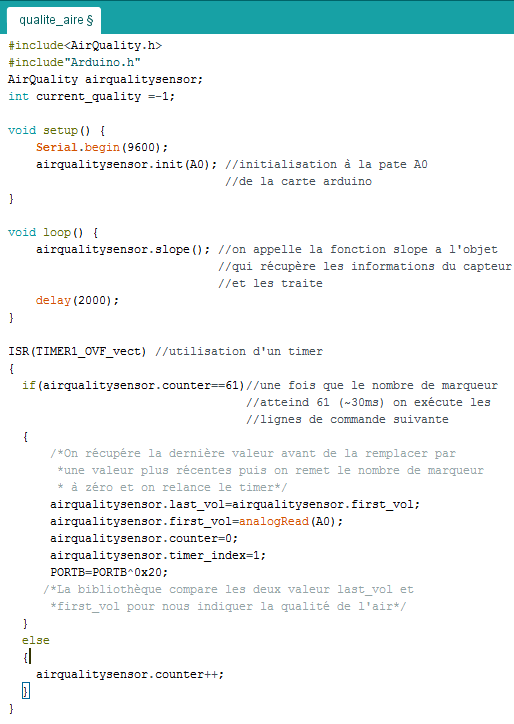
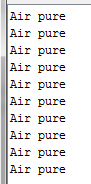
## Code de test unitaire

# CAPTEUR DE QUALITE D'AIR :

 Le capteur de qualité d'air est branché en analogique à la carte Arduino. On récupère les informations du "Air QualitySensor" avec l'aide d'une bibliothèque ajoutée à notre programme. Ce dernier nous renvoie 0 (haute pollution),1,2 ou 3(air pur).

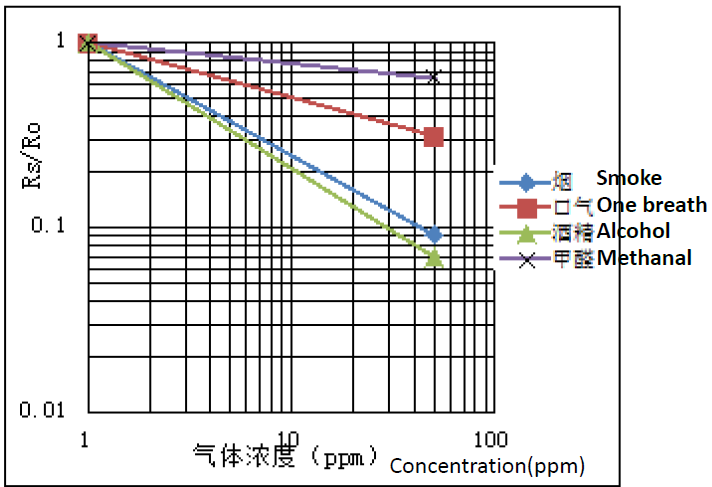
## Capteur de qualité d'air

## Montage du système



## Code de test unitaire

## Valeur obtenue

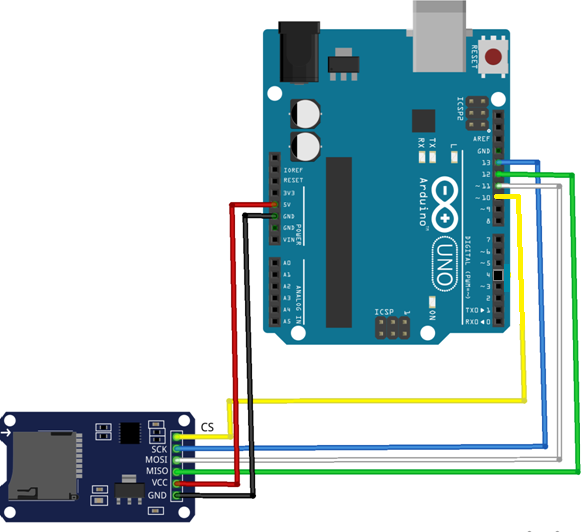


Rs est une résistance, dans un gaz ciblé avec une concentration différente, R0 est la résistance du capteur à l'air propre. Toutes les valeurssont déterminées dans des conditions de test standard.

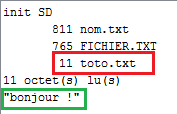
# 

# CARTE SD :

La carte SD nous servira à sauvegarder les données du compteur d'énergie. Pour cela il m’a fallu apprendre à lire et à écrire dans une carte SD via une carte Arduino. Un article de Hackable magazinea fait un tutoriel à ce sujet, j'ai donc suivile tutoriel afin de réaliser mon objectif. Il fallait aussi relier la carte SD à la carte Arduino, nous avons donc utilisé un "MicroSDcard Adapter".



## Montage du système

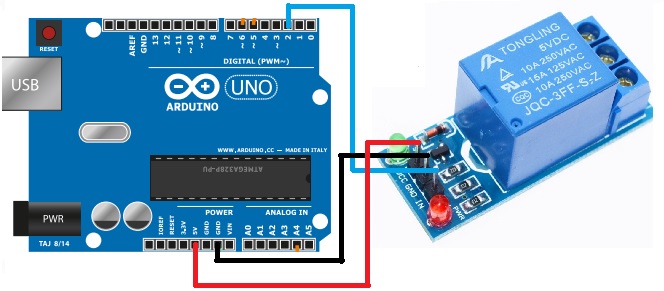


## Code de test unitaire

## Valeur obtenue

**11 toto.txt** : 11 correspond au nombre d'octet dans le fichier nommé **toto.txt**

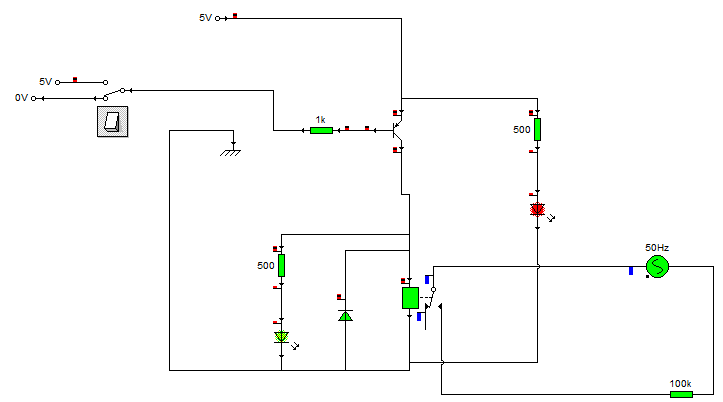
# LE RADIATEUR :

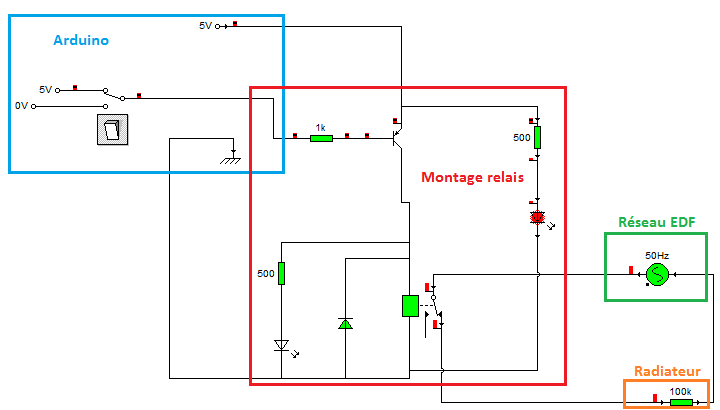
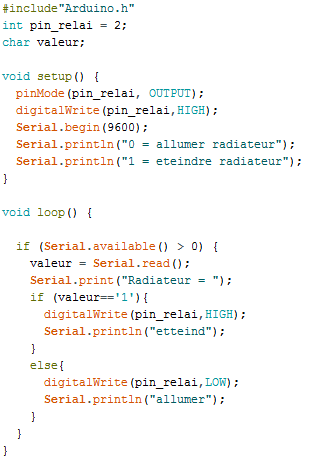
 Pour contrôler le radiateur avec une carte Arduino, il nous faut un relais qui nous permettre de contrôler avec le 5V de la carte, le radiateur qui est alimenté en 230V sur le réseau EDF.

Le schéma ci-dessus nous montre comment est reliée la carte aux CMS, carte qui permet de contrôler le relais.

## Montage du système

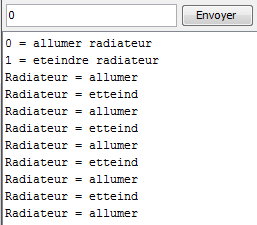
Le radiateur est éteint





## Montage du système

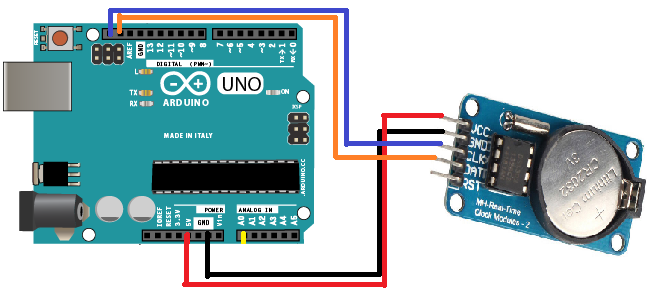
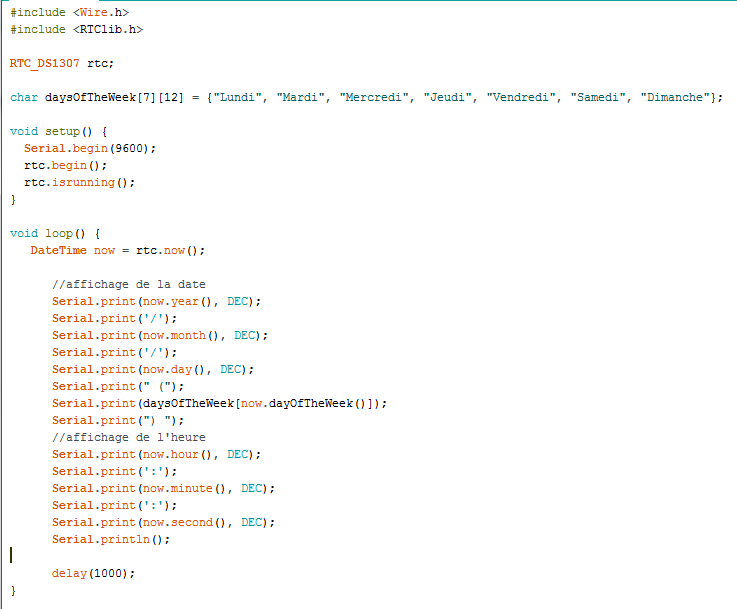
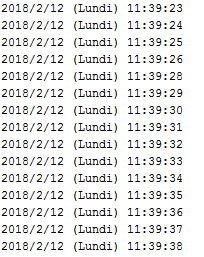
Le radiateur est allumé



## Valeur sur l'interface

## Code de test unitaire

# LE MODULE RTC :

 Le module RTC est un module qui permet de gérer l'heure, même quand le système n'est pas sous tension grâce a une petite pile. Pour le faire fonctionner j'ai fait appel à une bibliothèque sur GitHUB. Avec les méthodes j'ai donc créé un mini programme donnant l’heure qui nous servira pour l'horodatage de données de consommation.

## Valeur obtenue

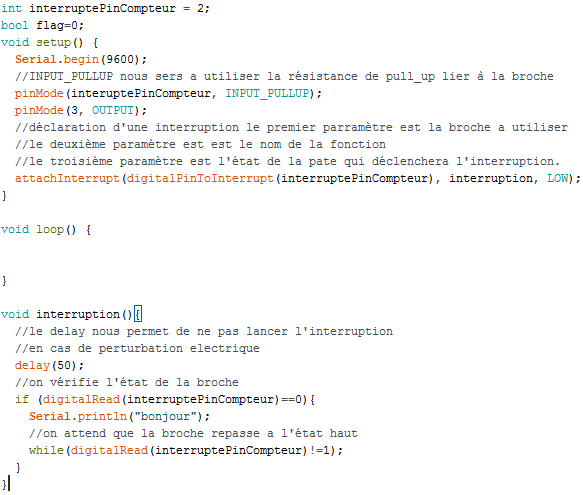
## Code de test unitaire

## Montage du système

# COMPTEUR ELECTRIQUE :

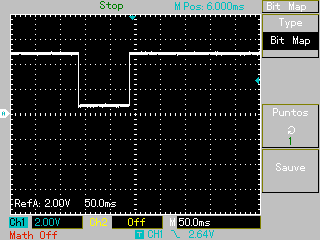
Avec le compteur électrique nous allons pouvoir récupérer le nombre de KWh que le radiateur va consommer. Il délivrera une impulsion tout les 100Wh que nous allons récupérer avec la carte Arduino.

J'ai pour cela utilisé la pin 2 de la carte car elle est la seule pin avec la numéro 3 à posséder une interruption externe. Ensuite nous avons utilisé la résistance de pull-up interne de la pin 2 afin de ne pas avoir à monter le composant sur une carte.

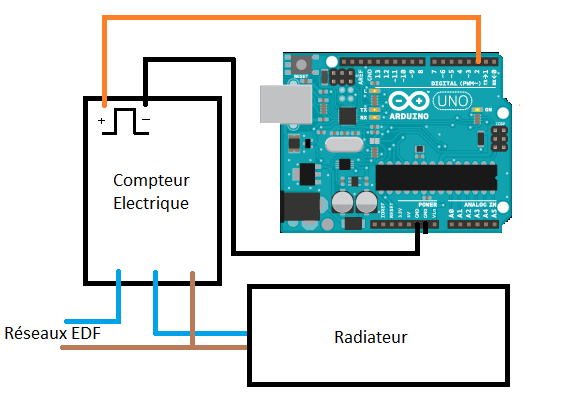


## Code de test unitaire

Le délai est de 50 ms,ce qui correspond à la moitié de la durée de l'impulsion générée par le compteur. Ensuite, on revérifie l'état de la broche. Cette opération nous permet d'éliminerles parasites liés aux perturbations électriques.

Voici à l’oscilloscope, l’impulsion générée par le compteur électrique :

## Montage du système

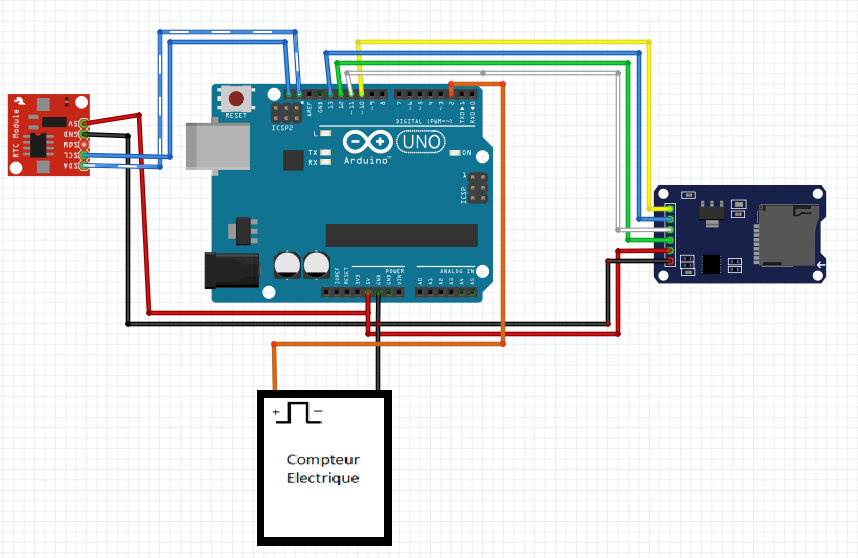


3 - REGROUPEMENT DES COMPOSANTS

Maintenant que tous les tests unitaires sont faits, nous pouvons regrouper les composants sur une seule carte et les codes dans un seul fichier.

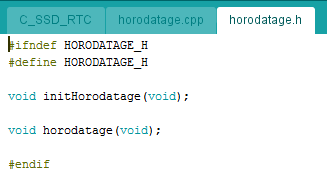
# LA CARTE SSD, LE MODULE RTC ET LE COMPTEUR ÉLECTRIQUE :

## Montage du système



## RTC

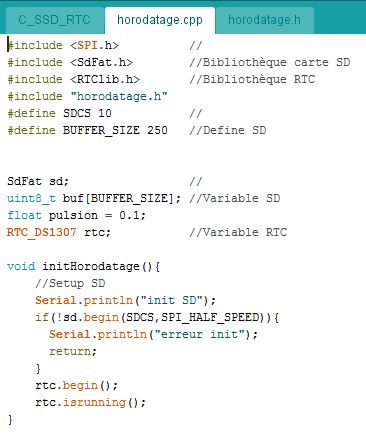
## SSD



## .h du regroupement



## Code principale du regroupement.



L’un des avantages d’exécuter le programme d’interruption dans le loop est que nous évitons les conflits d’interruption, car la bibliothèque qui nous sert à horodater utilise aussi une interruption, donc si nous avions exécuté la fonction horodatage dans le programme d’interruption il y aurait eu conflit d’interruption (une interruption dans une interruption).

## .cpp partie 1 du regroupement

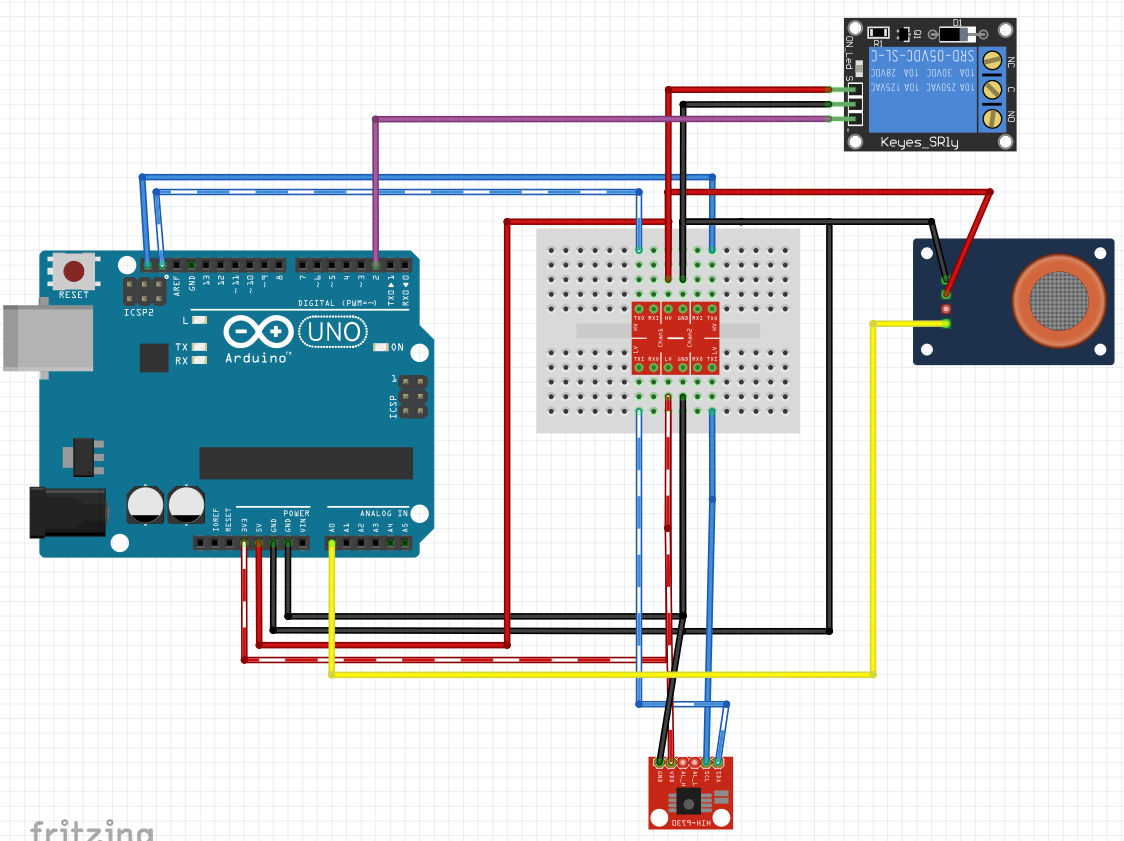
# 

## .cpp partie 2 du regroupement

Dans cette partie nous avons regroupé tout ce qui concerne l'horodatage de la consommation électrique. Le compteur crée une impulsion à laquelle on ajoute une date, une heure et une valeur en Watt/h, puis cette trame est sauvegardée sur une carte SD dans un fichier texte (.txt).

# LE THERMOMETRE, l'HYGROMETRE, LE CAPTEUR DE QUALITE D'AIR ET LE RELAIS :

## Montage du système



## RELAIS

LE RELAIS

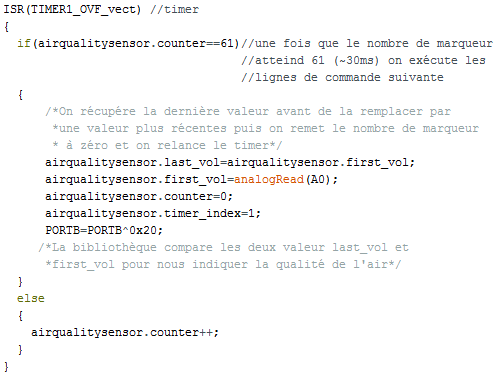
## HYGROMÉTRE

## TERMOMÉTRE

## QUALITE D'AIR

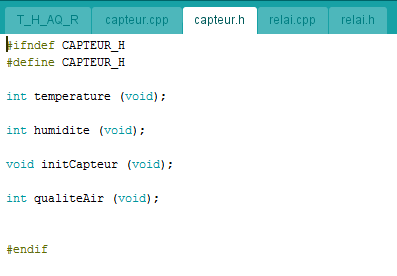


## Code principale du regroupement.



## capteur.cpp partie 2 du regroupement

## capteur.cpp partie 1 du regroupement



## Capteur.h du regroupement



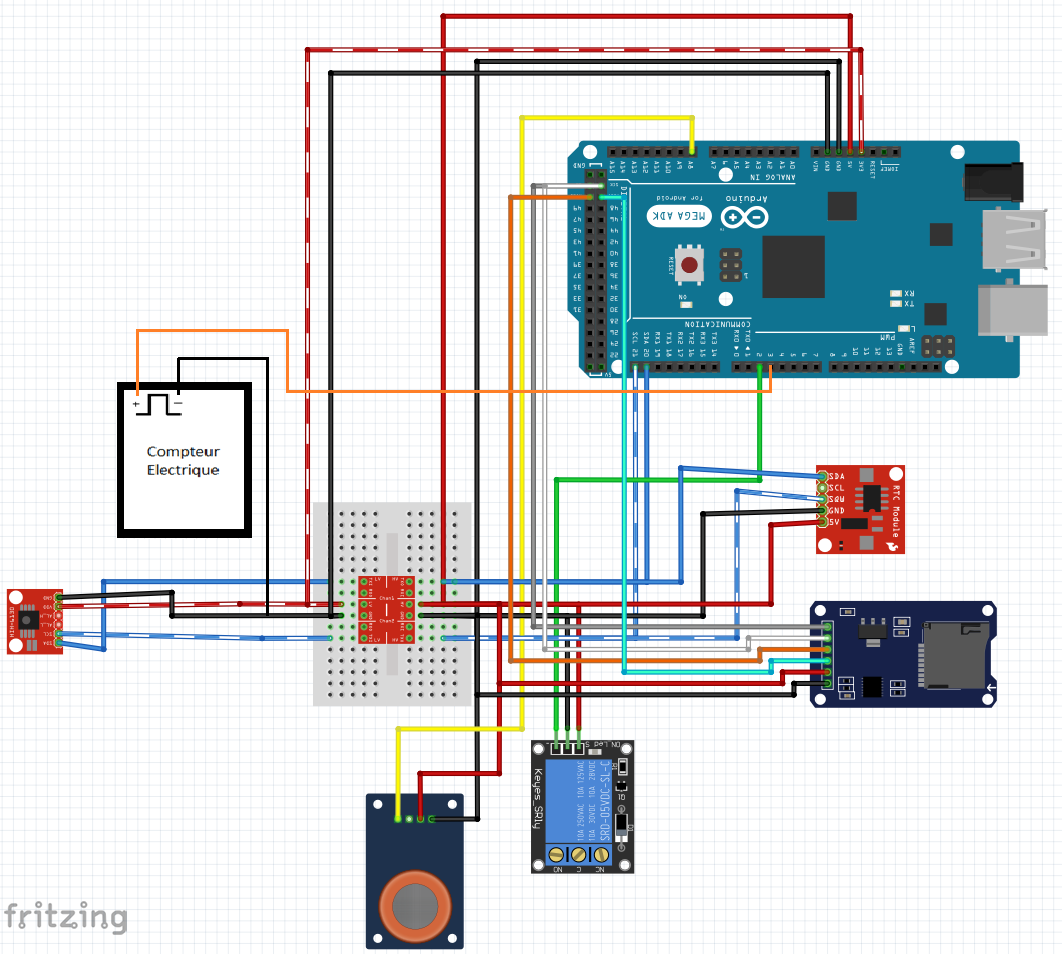
## capteur.cpp du regroupement



## Capteur.h du regroupement

Ici, nous avons regroupé les capteurs et actionneurs de la partie confort. On y récupère la température(°C), le taux d’humidité (%), la qualité de l'air et on contrôle le relais qui agira sur le radiateur.

# REGROUPEMENT FINAL DES COMPOSANTS :

 Une fois la première partie de regroupement réalisée, nous pouvons à présent regrouper la totalité de cette partie en un seul montage et un seul programme. Nous pourrons ensuite regrouper les codes de la partie de l'étudiant 1 avec notre partie.

## Montage du système

## QUALITE D'AIR

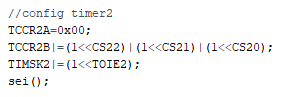
## RELAIS

## RTC

Lors du branchement du module SD nous avons changé de **pin** pour la communication SPI car elle varie en fonction de la carte que nous utilisons.

**Arduino UNO** : MOSI = 11 / MISO = 12 / SCK = 13 / SS = 10

**Arduino MEGA** : MOSI = 51 / MISO = 50 / SCK = 52 / SS = 53



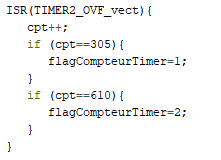
Ici nous paramétrons un timer qui nous permettra de récupérer les valeurs et états de nos capteurs et actionneurs.

**TCCR2A** nous indique que le timer est en mode normal.

**TCCR2B** est le coefficient diviseur de la fréquence du CPU.

**TIMSK2** nous indique pour que le timer entre en interruption à chaque débordement (Overflow).

**sei() :** Active les interruptions.

****

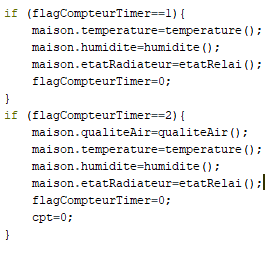
Ici notre timer entre en interruption 61 fois

par seconde, ce qui correspond à la levée des drapeaux.

Le premier drapeau se lève tous les 305 débordements

du timer, donc toutes les 5 secondes. Quant au deuxième

drapeau, il se lève toutes les 10 secondes.

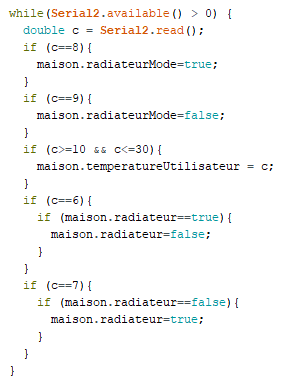


Nous voyons maintenant que les

données sont stockées dans l’objet maison

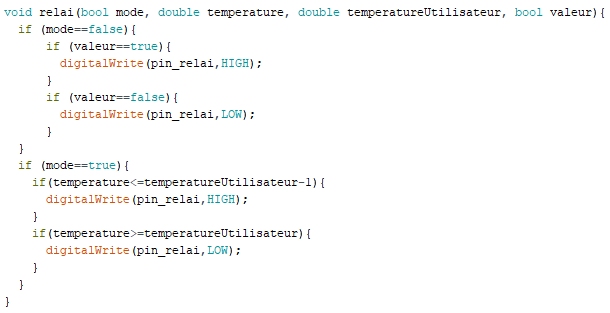
toutes les 5 et 10 secondes.

Les drapeaux sont remis a zéro ainsi que le compteur, à la fin du programme, lancé par le deuxième drapeau.

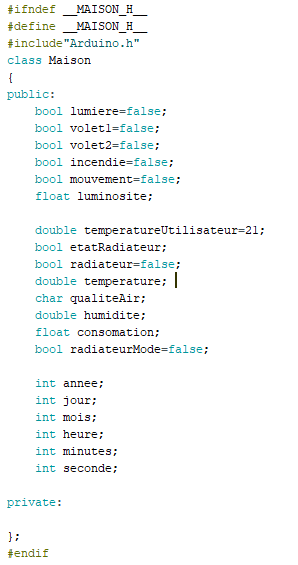


Ce morceau du code nous permet de récupérer les données envoyées par la tablette. Ces données sont des valeurs qui correspondent à une requête soumise par l’utilisateur. Par exemple si la valeur envoyée est 8, nous passons en mode manuel, si la valeur est 9 nous passons en mode automatique. De cette manière nous pouvons contrôler le mode de fonctionnement du radiateur.

Maison.temperatureUtilisateur prend la valeur choisie par l’utilisateur pour sont chauffage. Ainsi, lorsque le chauffage sera en mode automatique, le chauffage restera à latempératuresouhaitée.



Ici nous avons le code qui permet de contrôler le chauffage en automatique et en manuel. On peut remarquer qu’il y a une plage d’hystérésis de 1 degré qui nous permet de ne pas casser notre relais.

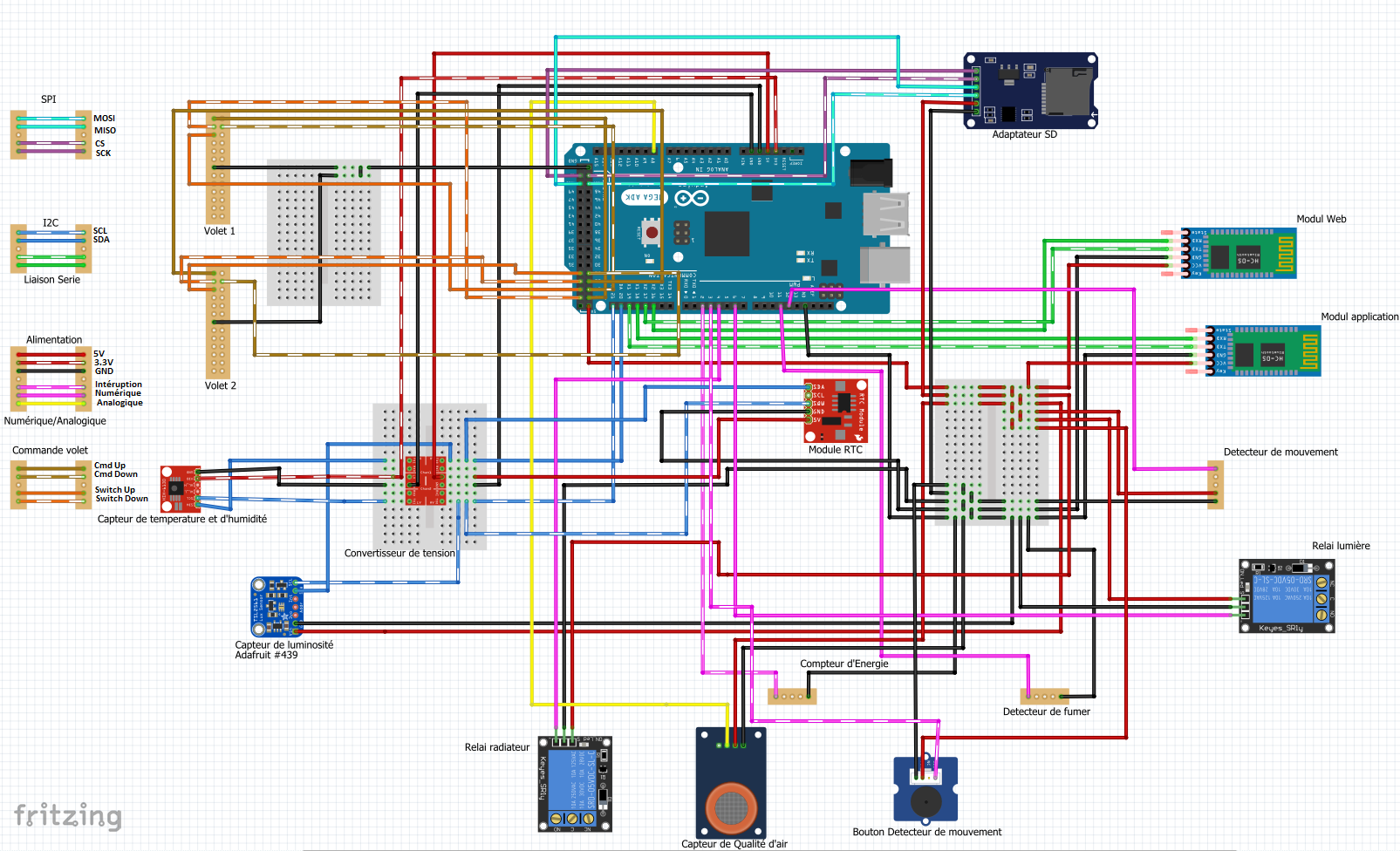


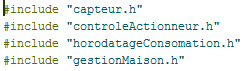
Voici la première version de l’objetmaison. Il sera amélioré par la suite avec l’arrivée du code de l’étudiant 1.

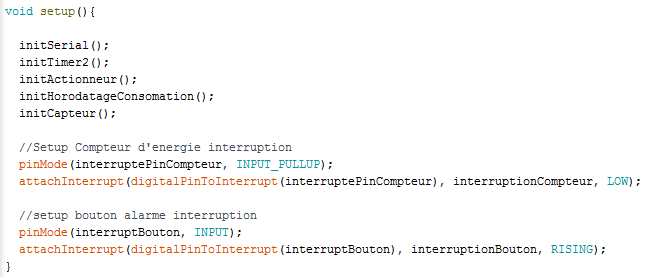
Dans cet objet nous avons toutes les variables liées à un capteur ou à un actionneur (sonétat, les valeurs qu’il renvoie, son mode de fonctionnement, etc.).

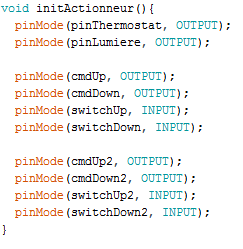
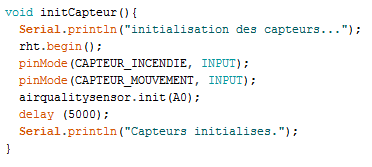
L’heure et la date sont aussi sauvegardées séparément pour pouvoir par la suite les envoyer un par un à la tablette ou au serveur Web.

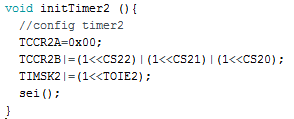
4 - REGROUPEMENT DES DIFFÉRENTES PARTIES.

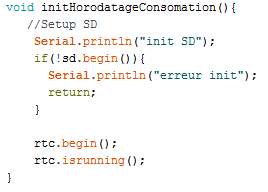


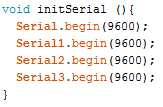
 Dans cette partie,nous verrons le travail que nous avons réalisé lors du regroupement de toutesnos parties. Dans un premier temps, nous avons réorganisé le code, puis nous avons créé de nouvelles bibliothèques plus intuitives et plus claires.

Ces nouvelles bibliothèques ont leur propre fonction initialisation, ce qui rend le voidsetup() beaucoup plus intuitif.







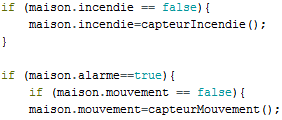
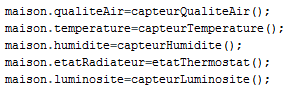


# LA BIBLIOTHÈQUE CAPTEUR.H :

Dans cette bibliothèque, nous gérons tous les capteurs (humidité, luminosité, température, qualité d'air, mouvement et incendie), ainsi que l'envoi de SMS créé par l'étudiant 1.



Chaque capteur renvoie une valeur qui est récupérée dans l'objet maison.



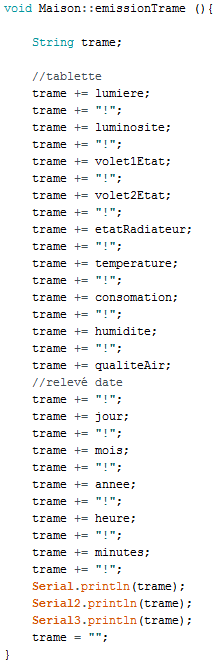
# LA BIBLIOTHÈQUE GESTIONMAISON.H :



Voici l'objet maison finalisé, on y stocke toute valeur utilisable ainsi que deux méthodes dont nous allons nous servir pour communiquer avec la tablette et pour envoyer nos données au serveur web.

Ceci est une première ébauche de notre objet, il est nettement améliorable, surtout au niveau des règles d'encapsulation.

# FONCTION D'ENVOI DE LA TRAME :



Dans cette fonction nous ajoutons les valeurs récupéréesdes différents capteurs les unes après les autres, en les séparant par un point d'exclamation.

Cette opération nous génère une trame que nous allons envoyer.

trameC.png

État de la lumière 0 = éteint et 1 = allumé

Luminosité en LUX

État du volet n°1 0 = fermé et 1 = ouvert

État du volet n°1 0 = fermé et 1 = ouvert

État du radiateur 0 = éteint et 1 = allumé

Température en °C

Consommation en KwH

Taux d'humidité en %

Qualité de l'air de 0 à 3

Heure et date : jour ! mois ! année ! heure ! minutes !

# 

# CONCLUSION :

## Mon ressenti :

Ce projet m'a beaucoup plu. Le travail d'équipe est quelque chose de très appréciable, et le faitd'être le chef de projet m'a beaucoup apporté, de la confiance en soi, des techniques d’organisation, et j'ai pris connaissance de mes capacités de travail.

## Les évolutions possibles :

Pour les radiateurs il faudrait envisager de les piloter automatiquement avec une plage d'hystérésis plus fine, car 1°C de différence dans une maison se ressent considérablement.

Il faudrait aussi faire en sorte que les actionneurs puissent changer d'état en même temps et pas les uns après les autres.

Une amélioration de l'objet maison est aussi possible et recommandé, surtout au niveau des règles d'encapsulation.



1. **Horodater :**Ajouter de manière automatique la date et l'heure à un document. [↑](#footnote-ref-2)
2. **GITHUB :** Service web d'hébergement et de gestion de développement de logiciels. [↑](#footnote-ref-3)