**Candidat : GIORGIS Esteban, Chef de projet : FAVRE Raphael, Expert 1 : ROY Alain, expert 2 : MASSON Baptiste**

Horloge à LED



Table des matières

[1. Analyse préliminaire 3](#_Toc104837703)

[1.1 Introduction 3](#_Toc104837704)

[1.2 Organisation 3](#_Toc104837705)

[1.3 Objectifs 4](#_Toc104837706)

[1.4 Objectifs personnels 4](#_Toc104837707)

[1.5 Planification initiale 5](#_Toc104837708)

[2. Analyse / Conception 7](#_Toc104837709)

[2.1 Stratégie de test 7](#_Toc104837710)

[2.2 Uses cases scénario 9](#_Toc104837711)

[2.3 Risques techniques 12](#_Toc104837712)

[2.4 Planification finale 12](#_Toc104837713)

[2.5 Dossier de conception 13](#_Toc104837714)

[2.6 Dépôt distant 18](#_Toc104837715)

[2.7 Tableau Trello 19](#_Toc104837716)

[3. Réalisation 20](#_Toc104837717)

[3.1 Dossier de réalisation 20](#_Toc104837718)

[3.2 Description des tests effectués 26](#_Toc104837719)

[3.3 Erreurs restantes 31](#_Toc104837720)

[3.4 Liste des documents fournis 31](#_Toc104837721)

[4. Conclusion 32](#_Toc104837722)

[4.1 Objectifs atteints / non atteints 32](#_Toc104837723)

[4.2 Points positifs / négatifs 33](#_Toc104837724)

[4.3 Difficultés particulières 33](#_Toc104837725)

[4.4 Suites possibles pour le projet 33](#_Toc104837726)

[4.5 Bilan personnel 33](#_Toc104837727)

[5. Annexes 34](#_Toc104837728)

[5.1 Résumé rapport du TPI 34](#_Toc104837729)

[5.2 Sources – Bibliographie 35](#_Toc104837730)

[5.3 Journal de travail 36](#_Toc104837731)

[5.4 Glossaire 40](#_Toc104837732)

[5.5 Table des illustrations 40](#_Toc104837733)

[5.6 Manuel d’installation et de mise en place 41](#_Toc104837734)

[5.7 Manuel d’utilisation 46](#_Toc104837735)

# 1. Analyse préliminaire

## 1.1 Introduction

Le projet consiste à réaliser une horloge à LED fournissant également une indication du taux de CO2 et de la température. Certaines alertes comme des alertes visuelles et/ou sonores devront être faites dans le cas où le taux de CO2 dépassait un certain seuil voulu.

Ce projet est réalisé dans le cadre de mon projet TPI, l’examen final pour ma formation CFC. J’ai choisi ce sujet parce qu’il englobe deux domaines que j’apprécie particulièrement, la programmation et l’électronique. Réaliser ce TPI me permettra non seulement d’approfondir ceux deux domaines, mais aussi de me familiariser dans la réalisation de projets dans le domaine de l’informatique embarqué.

Nous avons décidé, M. Favre et moi-même, d’utiliser le « waterfall » avec quelques touches « d’agile » pour la méthode de gestion de projet.

## 1.2 Organisation

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Candidat | Chef de projet | Expert 1 | Expert 2 |
| Nom | Giorgis | Favre | Roy | Masson |
| Prénom | Esteban | Raphael | Alain | Baptiste |
| Email | esteban.giorgis@cpnv.ch | raphael.favre@cpnv.ch | alain.tpi@bluewin.ch | baptiste.masson@elca.ch |
| Tel. | 079 501 19 20 | 076 427 93 59 | 079 444 01 54 | 079 829 50 85 |

## 1.3 Objectifs

### 1.3.1 Partie Affichage de l’heure

* L’horloge devra être synchronisée avec une RTC
* Affichage de l’heure sur un anneau de 60 LED RGB
* Affichage de l’heure sur un affichage à 7 segments
* Les secondes devront être visible par un clignotement des deux petits points

### 1.3.2 Partie Affichage du taux de CO2 avec alerte

* Une alerte clignotante visuelle doit apparaître en cas d’un taux de CO2 dépassant une valeur prédéfinie
* Une alerte sonore doit être entendu en cas d’un taux de CO2 dépassant une valeur prédéfinie
* Un bouton poussoir permettant de sélectionner une des 4 options d’alertes
* La ou les alertes doivent s’arrêter au bout d’un certain temps même si le taux de CO2 mesuré est trop élevé puis reprendre après un certain temps.

### 1.3.3 Partie Affichage de température

* Affichage de la température sur l’affichage 7 segments
* Un bouton poussoir permettant de sélectionner un des 4 modes d’affichage disponible

## 1.4 Objectifs personnels

* Tenir à jour le journal de travail
* Maintenir une bonne communication avec toutes les personnes liées à mon projet
* Maintenir les rendus de documentations hebdomadaires
* Respecter au maximum le temps accordé pour la réalisation du projet (90 heures)
* Réaliser un dossier de projet le plus complet possible
* Bien commenter le code
* Bien versionner le code avec un cartouche indiquant la version

## 1.5 Planification initiale

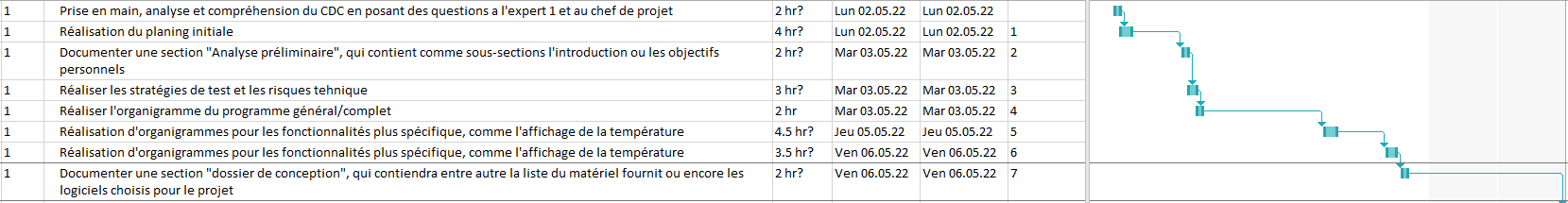


Figure 1 Semaine 1

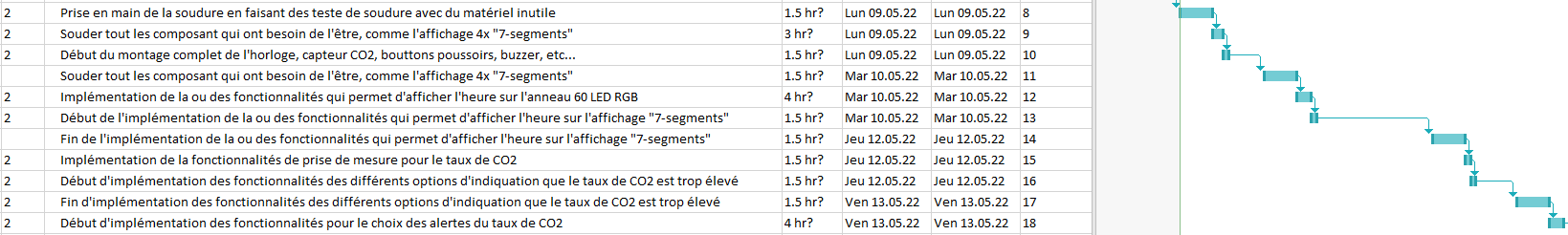


Figure 2 Semaine 2

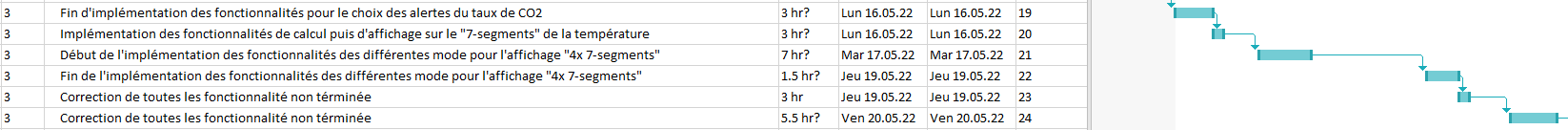


Figure 3 Semaine 3

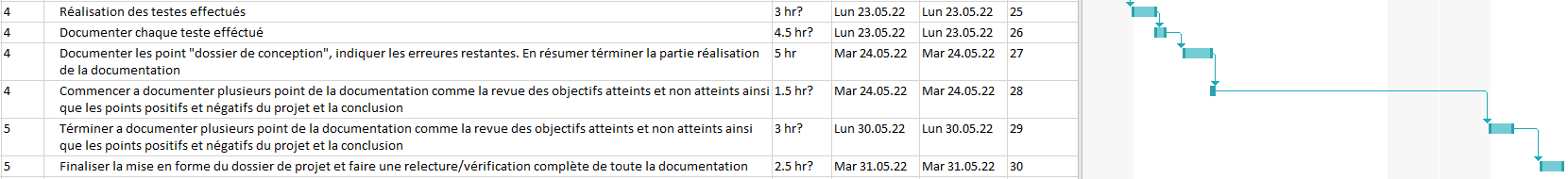


Figure 4 Semaines 4 et 5

# 2. Analyse / Conception

## 2.1 Stratégie de test

**1er test : « Test de l’anneau 60 LED »**

Les 4 anneaux de 60 LED RGB seront soudé entre eux pour former un rond et branché sur un breadboard relié à la carte Arduino. Celle-ci sera reliée au PC à l’aide d’un câble USB 2.0.

* J’utiliserai la valeur de l’heure indiquée par les 4 anneaux de 60 LED RGB pour vérifier si elle correspond à celle indiquée par mon ordinateur CPNV. L’heure, la minute et les secondes doivent être indiqué avec des couleurs distinctes.

Le test sera effectué par mon chef de projet ainsi que moi-même.

**2ème test : « Test de l’affichage 7 segments »**

L’affichage 4x « 7-segments » sera branché sur un breadboard relié à la carte Arduino. Celle-ci sera reliée au PC à l’aide d’un câble USB 2.0.

* J’utiliserai la valeur de l’heure indiquée par l’affichage 4x « 7-segments » pour vérifier si elle correspond à celle indiqué par mon ordinateur CPNV. L’heure et la minute doivent être indiqué.

Le test sera effectué par mon chef de projet ainsi que moi-même.

**3ème test : « Test des secondes pour l’affichage 7 segments »**

L’affichage 4x « 7-segments » sera branché sur un breadboard relié à la carte Arduino.

Celle-ci sera reliée au PC à l’aide d’un câble USB 2.0.

* Les secondes doivent pouvoir être identifiées par le clignotement de deux points situés au centre de l’affichage 4x « 7-segments ».

Le test sera effectué par mon chef de projet ainsi que moi-même.

**4ème test : « Test du capteur CO2 avec alerte visuelle »**

Un capteur de CO2 ainsi que les 4 anneaux de 60 LED RGB sera branché sur un breadboard relié à la carte Arduino. Celle-ci sera reliée au PC à l’aide d’un câble USB 2.0.

* Une fois que le taux de CO2 dépasse un seuil indiqué au préalable, les 4 anneaux de 60 LED RGB clignoteront à intervalle régulier durant 10 secondes.

Le test sera effectué par mon chef de projet ainsi que moi-même.

**5ème test : « Test du capteur CO2 avec alerte sonore »**

Un capteur de CO2 ainsi qu’un buzzer seront branchés sur un breadboard relié à la carte Arduino. Celle-ci sera reliée au PC à l’aide d’un câble USB 2.0.

* Une fois que le taux de CO2 dépasse un seuil indiqué au préalable, le buzzer produira un son.

Le test sera effectué par mon chef de projet ainsi que moi-même.

**6ème test : « Test des options d’alertes »**

Un bouton poussoir, l’affichage 4x « 7-segments » et les 4 anneaux de 60 LED RGB seront branchés sur un breadboard relié à la carte Arduino. Celle-ci sera reliée au PC à l’aide d’un câble USB 2.0.

* Pour chaque clique du bouton poussoir, l’option d’alerte pour indiquer que le taux de CO2 est trop élevé passera à la suivante, l’ordre est le suivant :
  + Alerte visuelle ET alerte sonores activées
  + Alerte visuelle uniquement
  + Alerte sonore uniquement
  + Aucune alerte
* L’option d’alerte de base à chaque démarrage est le premier mode (Alerte visuelle ET alerte sonores activées)
* Une fois la dernière option sélectionnée (Aucune alerte), presser sur le bouton poussoir et le premier mode sera sélectionné (Alerte visuelle ET alerte sonores activées).

Les tests seront effectués par mon chef de projet ainsi que moi-même.

**7ème test : « Test des modes d’affichage »**

Un bouton poussoir et l’affichage 4x « 7-segments » sera branché sur un breadboard qui lui-même sera branché sur la carte Arduino. Celle-ci sera reliée au PC à l’aide d’un câble USB 2.0.

* Presser sur le bouton poussoir change le mode d’affichage du 4x « 7-segments », l’ordre d’affichage est le suivant :
  + Température / Heure/ Taux de CO2 activés en alternance
  + Température uniquement
  + Heure uniquement
  + Taux de CO2 uniquement
* L’affichage de base à chaque démarrage est le premier mode (Température / Heure / Taux de CO2 activités en alternance).
* Une fois le dernier mode atteint (Taux de CO2 uniquement) presser sur le bouton poussoir et le premier mode sera sélectionné (Température / Heure/ Taux de CO2 activés en alternance).

Les tests seront effectués par mon chef de projet ainsi que moi-même.

## 2.2 Uses cases scénario

### 2.2.1 Partie affichage de l’heure

|  |  |  |
| --- | --- | --- |
| Contexte | Condition | Résultat |
| L’horloge entière est alimentée | - | L’horloge 60 LED affiche l’heure avec trois couleurs différentes |
| L’horloge entière est alimentée | - | L'affichage 7 segments affiche en alternance : l'heure (en heures et minutes) durant 3 secondes puis la température durant 3 secondes. |
| L’horloge entière est alimentée | Le module Arduino Uno est débranché de l’ordinateur durant un certain temps | L’heure sera maintenue grâce à la synchronisation de l’horloge à une RTC |

### 2.2.2 Partie Affichage du taux de CO2 avec alerte

|  |  |  |
| --- | --- | --- |
| Contexte | Condition | Résultat |
| L’horloge entière est alimentée. Le mode d’alerte « Alerte visuelle et sonore » est sélectionné | Le taux de CO2 dans l’environnement dépasse le seuil voulu | L’horloge 60 LED clignote durant 3 secondes ainsi que le buzzer sonne par à coup durant 3 secondes |
| L’horloge entière est alimentée. Le mode d’alerte « Alerte visuelle uniquement » est sélectionné | Le taux de CO2 dans l’environnement dépasse le seuil voulu | L’horloge 60 LED clignote durant 3 secondes |
| L’horloge entière est alimentée. Le mode d’alerte « Alerte sonore uniquement » est sélectionné | Le taux de CO2 dans l’environnement dépasse le seuil voulu | Le buzzer sonne par à coup durant 3 secondes |
| L’horloge entière est alimentée. Le mode d’alerte « Aucune alerte » est sélectionné | Le taux de CO2 dans l’environnement dépasse le seuil voulu | Aucune alerte n’est affichée |
| L’horloge entière est alimentée. | - | Le mode d’alerte « Alerte visuelle et sonore » est sélectionné |
| L’horloge entière est alimentée. Le mode d’alerte « Alerte visuelle et sonore » est sélectionné | Le bouton lié aux modes d’alertes de CO2 est pressé puis relâché | Le mode d’alerte « Alerte visuelle uniquement » est sélectionné |
| L’horloge entière est alimentée. Le mode d’alerte « Alerte visuelle uniquement » est sélectionné | Le bouton lié aux modes d’alertes de CO2 est pressé puis relâché | Le mode d’alerte « Alerte sonore uniquement » est sélectionné |
| L’horloge entière est alimentée. Le mode d’alerte « Alerte sonore uniquement » est sélectionné | Le bouton lié aux modes d’alertes de CO2 est pressé puis relâché | Le mode d’alerte « Aucune alerte » est sélectionné |
| L’horloge entière est alimentée. Le mode d’alerte « Aucune alerte » est sélectionné | Le bouton lié aux modes d’alertes de CO2 est pressé puis relâché | Le mode d’alerte « Alerte visuelle et sonore » est sélectionné |

### 2.2.3 Partie Affichage de température

|  |  |  |
| --- | --- | --- |
| Contexte | Condition | Résultat |
| L’horloge entière est alimentée. | - | L'affichage 7 segments affiche en alternance : l'heure (en heures et minutes) durant 3 secondes puis la température durant 3 secondes. |
| L’horloge entière est alimentée. L’option d’affichage « Température / Heure/ Taux de CO2 activés en alternance » est sélectionné | - | L’affichage 7 segments affichage la température / l’heure et le taux de CO2 en alternance |
| L’horloge entière est alimentée. L’option d’affichage « Température uniquement » est sélectionné | - | L’affichage 7 segments affichage uniquement la température |
| L’horloge entière est alimentée. L’option d’affichage « Heure uniquement » est sélectionné | - | L’affichage 7 segments affichage uniquement l’heure |
| L’horloge entière est alimentée. L’option d’affichage « Taux de CO2 uniquement » est sélectionné | - | L’affichage 7 segments affichage uniquement le taux de CO2 |
| L’horloge entière est alimentée. | - | L’option d’affichage « Température / Heure/ Taux de CO2 activés en alternance » est sélectionnée |
| L’horloge entière est alimentée. L’option d’affichage « Température / Heure/ Taux de CO2 activés en alternance » est sélectionné | Le bouton lié aux options d’affichage du 7 segments est pressé puis relâché | L’option d’affichage « Température uniquement » est sélectionnée |
| L’horloge entière est alimentée. L’option d’affichage « Température uniquement » est sélectionné | Le bouton lié aux options d’affichage du 7 segments est pressé puis relâché | L’option d’affichage « Heure uniquement » est sélectionnée |
| L’horloge entière est alimentée. L’option d’affichage « Heure uniquement » est sélectionné | Le bouton lié aux options d’affichage du 7 segments est pressé puis relâché | L’option d’affichage « Taux de CO2 uniquement » est sélectionnée |
| L’horloge entière est alimentée. L’option d’affichage « Taux de CO2 uniquement » est sélectionné | Le bouton lié aux options d’affichage du 7 segments est pressé puis relâché | L’option d’affichage « Température / Heure/ Taux de CO2 activés en alternance » est sélectionnée |

## 2.3 Risques techniques

Ne pas avoir soudé depuis quelques années est un risque, une mauvaise soudure qui créerait des cours circuits peu devenir rapidement un problème. Planifier des entrainements de soudure pendant quelques heures me permettra de reprendre la main sur la façon correcte de souder.

La grande quantité de composants/modules pour réaliser le projet peut représenter un risque au niveau du branchement/montage. Un branchement tardif d’un module qui n’aurait pas de place et donc qui m’empêcherait de réaliser une fonctionnalité serait bien dommage. Pour combler au maximum ce risque, effectuer le montage complet du projet rapidement, m’assurera la possibilité de faire chaque fonctionnalité demandée.

Réaliser un projet Pré-TPI similaire à ce TPI, me permet d’être bien préparé aux éventuels problèmes ou imprévus que les projets dans le domaine de l'informatique embarqué peuvent provoquer. Avoir suivi des cours d’IEL, réaliser un autre petit projet embarqué au milieu de ma formation et surtout avoir réalisé plusieurs projets en programmation me donne une bonne préparation à la réalisation de plus gros projet comme ce TPI.

## 2.4 Planification finale

A voir dans le fichier « Planification\_finale\_EGS.mpp » -> chemin : « Documentation\ Planification\_finale\_EGS.mpp »

## 2.5 Dossier de conception

### 2.5.1 mATéRIEL PHYSIQUE pour réaliser le projet

* 1 ordinateur du CPNV
* 1 Carte Arduino Uno avec câbles USB.
* 4 « Fragments » d’anneau de 15 LED.
* 1 Affichage 4x « 7-segment »
* 1 RTC (Real-Time-Clock)
* 1 Capteur mesurant le taux de CO2
* 1 Multi-Capteur mesurant la température, le taux d’humidité et la pression atmosphérique
* 1 poste à soudure
* Une boîte fournit par le CPNV contenant divers Composants électroniques associés à l’Arduino : (Breadboard, fils de connexions, résistances, condensateurs, boutons poussoirs, Buzzer, etc…)

### 2.5.2 Système d’exploitation utilisé

* Windows 10 Education, Version 21H2

### 2.5.3 Différents logiciels utilisés

**GitHub desktop :** Utilisé pour transférer l’avancement de mon projet sur un dépôt distant

**Arduino IDE :** Comme son nom l’indique, l’IDE permettant de coder toutes les fonctionnalités demandées

**Word :** Utilisé pour la réalisation du dossier de projet

**Excel :** Utilisé pour la réalisation du journal de travail

**MS Project :** Utilisé pour réaliser les différentes planifications

**Google Chrome :** Permet d’effectuer toutes les recherches nécessaires pour réaliser le projet

**Microsoft Teams :** Utilisé pour communiquer avec le chef de projet

**Outlook :** Utilisé pour communiquer avec le chef de projet, l’expert 1 et l’expert 2

### 2.5.4 Diagramme de flux



Figure 5 Diagramme de flux du programme général



Figure 6 Diagramme de flux pour l'option "Génération de l’alerte"



Figure 7 Diagramme de flux pour l'option "Lecture du bouton poussoir option d’alerte"



Figure 8 Diagramme de flux pour l'option "Lecture du bouton poussoir mode d’affichage 7 segments"

## 2.6 Dépôt distant

Un dépôt distant « GitHub » contenant le projet en entier (documentation, code, diagramme de flux, etc…), a été créé pour pouvoir reprendre le projet en entier à n’importe quel moment. Il permet aussi de voir l’évolution complète du projet depuis le début.

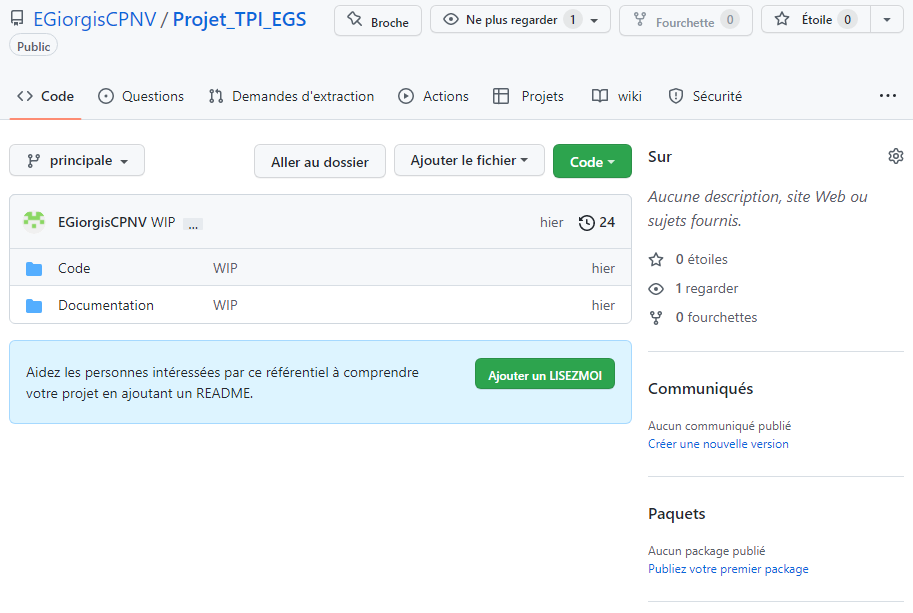


Figure 9 Dépôt distant GitHub du projet

## 2.7 Tableau Trello

Un tableau Trello a été créé pour la méthode de suivi de projet, voici une partie du tableau :

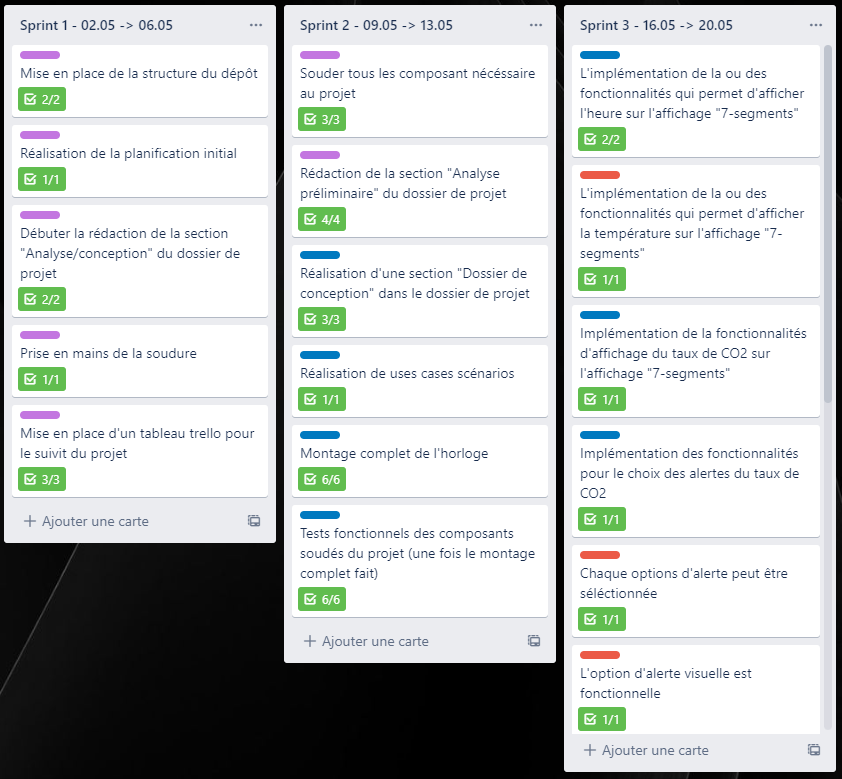


Figure 10 Tableau Trello utilisé pour le suivi du projet

# 3. Réalisation

## 3.1 Dossier de réalisation

### 3.1.1 Version des différents logiciels utilisés

* GitHub desktop : v3.0.0 (x64)
* Arduino IDE : v1.8.12
* Word : v16.0.4266.1001
* Excel : v16.0.4266.1001
* MS Project : v16.0.4266.1001
* Google Chrome : v101.0.4951.54 (x64)
* Outlook : v16.0.4266.1001

### 3.1.2 Description exacte du matériel

**Général :**

* 1 ordinateur type CPNV
* 4 « Fragments » d’anneau de 15 LED
* 1 Affichage 4x « 7-segment »
* 1 RTC (Real-Time-Clock)
* 1 Capteur mesurant le taux de CO2
* 1 poste à soudure
* 1 Arduino Uno R3 avec ATMEGA 328P-PU
* 1 Multi-Capteur mesurant la température, le taux d’humidité et la pression atmosphérique
* 1 BreadBoard ZY-60
* 1 Câble USB 2.0 A-B/1.5m
* 1 Sonde de température et 'humidité DHT11 + câble V
* 30 Câble Jumper M/M
* 40 Pinheader 2.54mm
* 1 Buzzer (active)
* 1 Buzzer (passive)
* 1 Carte résistance code des couleurs 4 et 5 bandes Résistances

**Résistances :**

* 10 100 W
* 10 220 W
* 10 330 W
* 10 -1 kw
* 10 4.7 kW
* 10 10 kW
* 10 47 kW
* 10 100 kW
* 1 10kW Potentiomètre

**Condensateurs :**

* 10 22 pF
* 10 100 pF
* 10 10 nF
* 10 100 nF
* 10 10 uF
* 10 470 uF

**LED :**

* 10 5mm Rouge
* 10 5mm Vert
* 10 5mm Jaune
* 10 5mm Bleu
* 2 RGB

### 3.1.3 Structure de dossier



Figure 11 Contenu du dossier "Code"



Figure 12 Contenu du dossier "Documentation"

### 3.1.4 Branchement final de l’horloge au complet

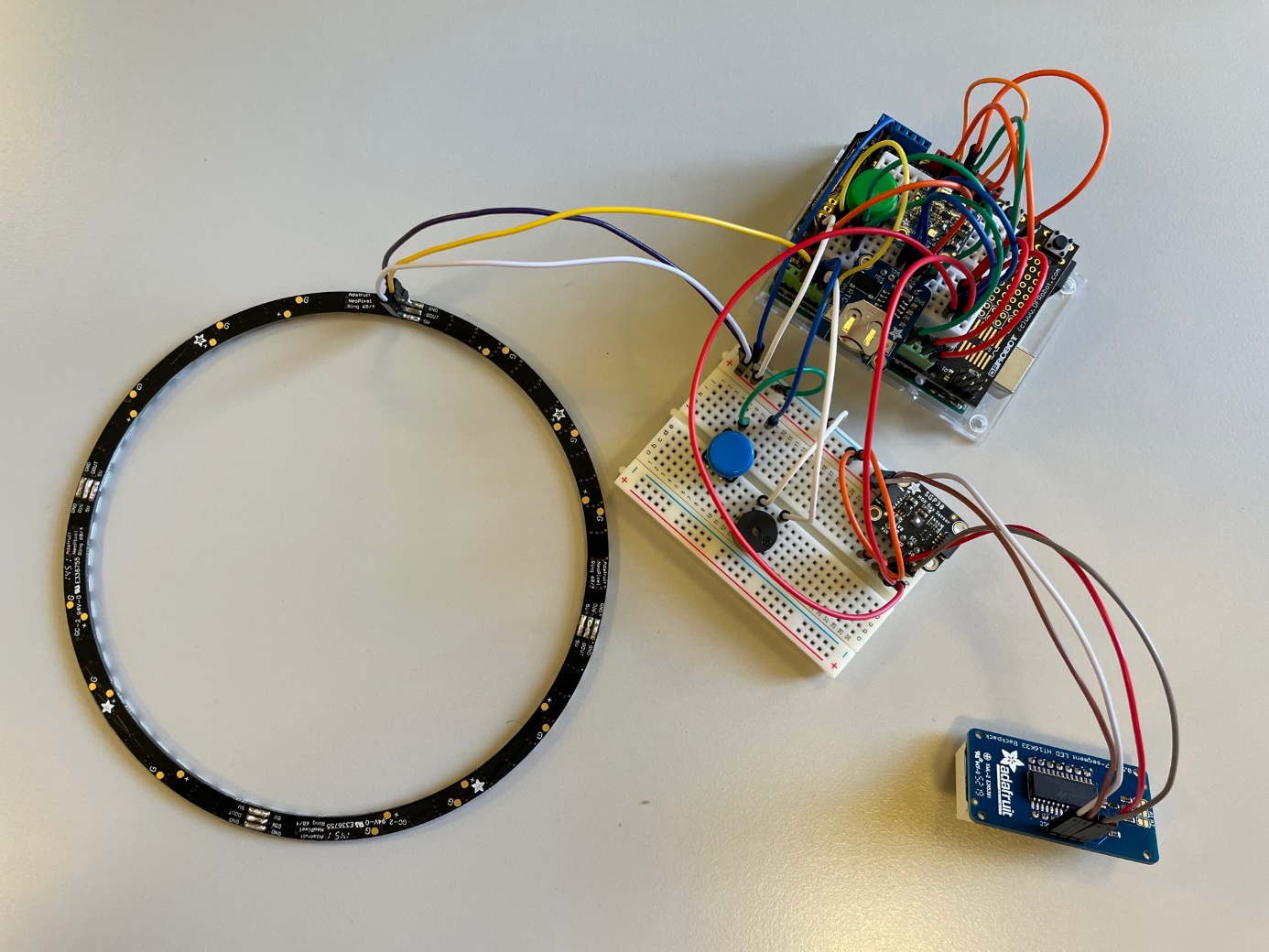


Figure 13 Branchement de l'horloge au complet

1

7

8

4

2

9

6

3

5

1) Horloge 60 LED (Utilisation du protocole I2C pour la communication)

2) Boutons poussoirs

3) Capteur de température BME280 (Utilisation du protocole I2C pour la communication)

4) Arduino Uno R3 avec ATMEGA 328P-PU

5) RTC

6) Capteur du taux de CO2 SGP30 (Utilisation du protocole I2C pour la communication)

7) Affichage 7 segments

8) Breadboard

9) Buzzer

### 3.1.5 Projet final

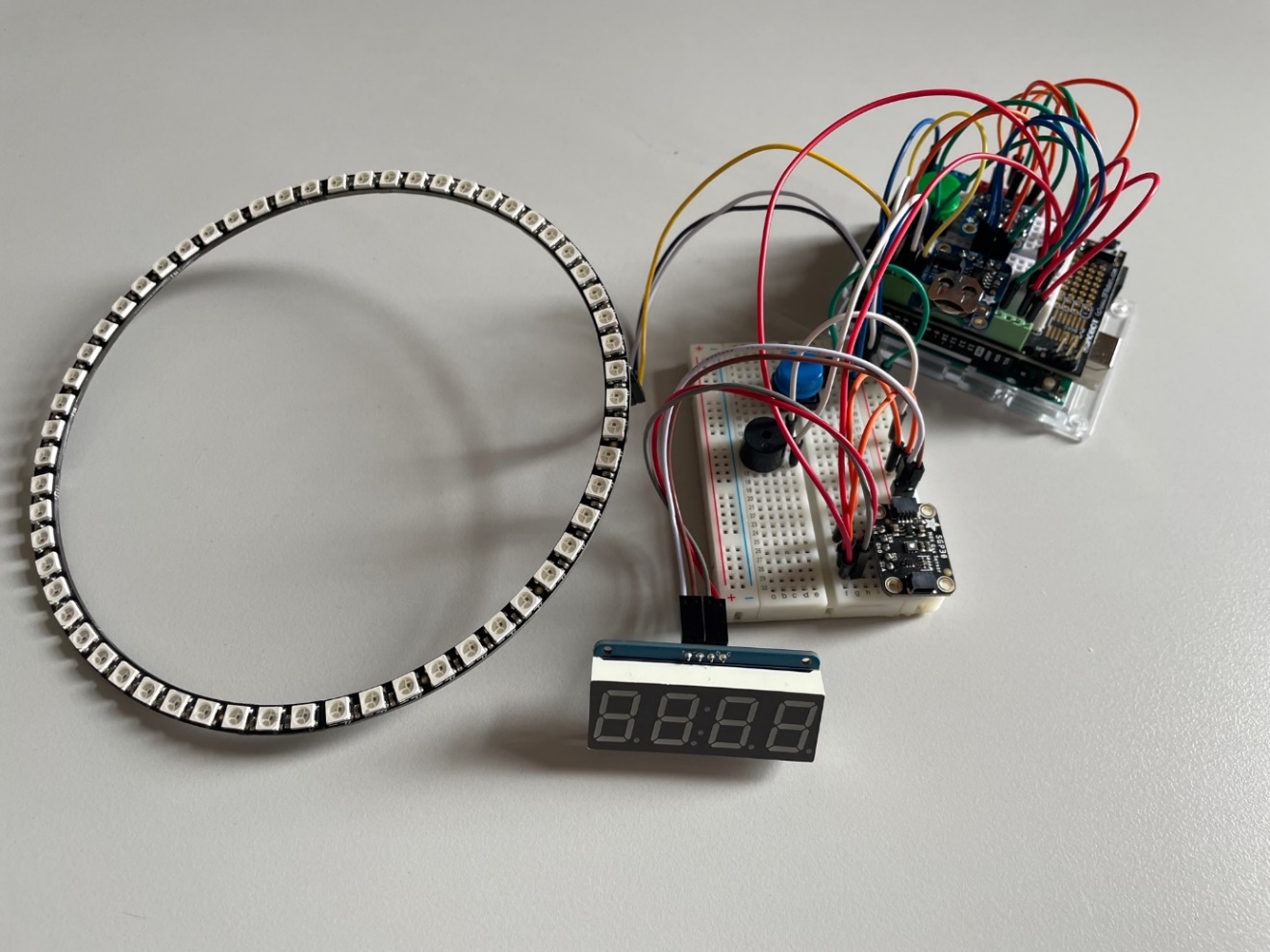


Figure 14 Projet final

### 3.1.6 Librairies utilisées

**« Adafruit\_BME280.h »**, cette librairie permet d’utiliser les fonctions liées au multi-capteur BME280 utilisé pour la partie température du projet.

**« RTClib.h »**, cette librairie permet d’utiliser les fonctions liées à la RTC donc pour tout ce qui est lié aux temps/heures, afficher l’heure actuelle sur l’affichage 7 segments par exemple.

**« Adafruit\_LEDBackpack.h »**, cette librairie permet d’utiliser les fonctions liées à l’affichage 7 segments.

**« Adafruit\_SGP30.h »**, cette librairie permet d’utiliser les fonctions liées au multi-capteur SGP30 utilisé pour la partie mesure du taux de CO2 du projet.

**« Adafruit\_NeoPixel.h »**, cette librairie permet d’utiliser les fonctions liées à l’utilisation des 60 LED elle-même utilisée pour réaliser l’horloge 60 LED.

## 3.2 Description des tests effectués

### 3.2.1 Test de l’anneau 60 LED

**Description :** Le but de ce test est d’afficher l’heure actuelle, synchronisé avec la RTC, sur l’horloge 60 LED (soudure de 4 parties de 15 LED) en utilisant 3 couleurs différentes pour indiquer l’heure, les minutes et les secondes. Le test a été effectué par moi-même.

**Matériel utilisé :** L’horloge 60 LED ainsi qu’une RTC sont branchées à la carte Arduino lui-même alimenté à un ordinateur tournant sous windows 10 avec un câble USB 2.0 A-B.

**Résultat attendu :** Les heures seront indiquées en rouge, les minutes en bleu et les secondes en vert. Les heures sont indiquées en bleu clair quand il s’agit des heures du matin et en bleu foncé pour les heures de l’après-midi et du soir.

**Résultat final :** Aucun affichage de l’heure, la fonctionnalité complète n’a pas été faite par manque de temps.

**Validité du test :** Test pas validé

### 3.2.2 Test du capteur CO2 avec alerte sonore

**Description :** Le but de ce test est de faire biper en alternance le buzzer quand le taux de CO2 mesuré par le capteur SGP30 dépasse le seuil voulu. Le test a été effectué par moi-même.

**Matériel utilisé :** Un buzzer ainsi que le capteur SGP30 sont branchées à la carte Arduino lui-même alimenté à un ordinateur tournant sous windows 10 avec un câble USB 2.0 A-B.

**Résultat attendu :** Une fois que le taux de CO2 dépasse le seuil voulu, le buzzer produit du son en alternance.

**Résultat final :** Une fois que le taux de CO2 dépasse le seuil voulu, le buzzer produit du son en alternance. La vidéo du résultat est disponible [ici](https://github.com/EGiorgisCPNV/Projet_TPI_EGS/blob/main/Documentation/Tests_videos/test_alerte_sonore.mov)

**Validité du test :** Test validé

### 3.2.3 Test de l’affichage 7 segments

**Description :** Le but de ce test est d’afficher l’heure actuelle, synchronisé avec la RTC, sur l’affichage 7 segments. L’heure et les minutes sont indiquées et sont séparées par deux petits points. Le test a été effectué par moi-même.

**Matériel utilisé :** Un affichage 7 segments ainsi qu’une RTC sont branchées à la carte Arduino lui-même alimenté à un ordinateur tournant sous windows 10 avec un câble USB 2.0 A-B.

**Résultat attendu :** Les heures et les minutes sont affichées en chiffre sur le 7 segments séparé par deux petits points.

**Résultat final :** Les heures et les minutes sont affichées en chiffre sur le 7 segments séparé par deux petits points.

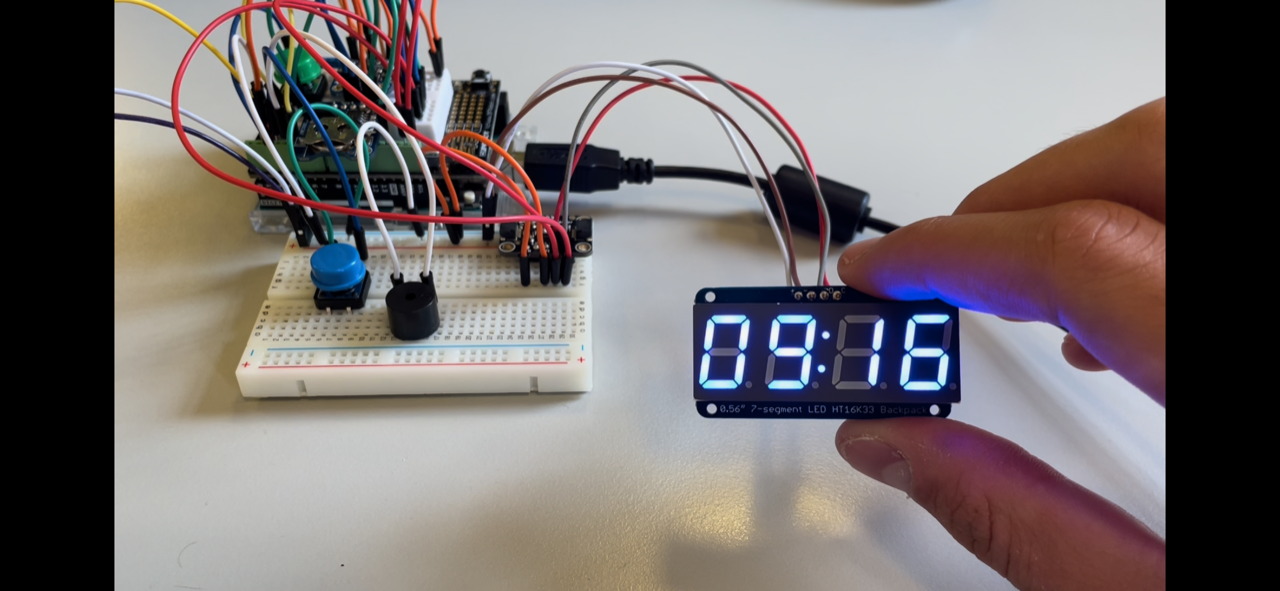


Figure 15 Résultat final du test « Test de l’affichage 7 segments »

**Validité du test :** Test validé

### 3.2.4 Test des secondes pour l’affichage 7 segments

**Description :** Le but de ce test est de faire clignoter les deux petits points pour indiquer les secondes sur l’affichage 7 segments. Le test a été effectué par moi-même.

**Matériel utilisé :** Un affichage 7 segments ainsi qu’une RTC sont branchées à la carte Arduino lui-même alimenté à un ordinateur tournant sous windows 10 avec un câble USB 2.0 A-B.

**Résultat attendu :** Chaque secondes les deux petits points de l’affichage 7 segments clignotent.

**Résultat final :** Chaque secondes les deux petits points de l’affichage 7 segments clignotent. La vidéo du résultat est disponible [ici](https://github.com/EGiorgisCPNV/Projet_TPI_EGS/blob/main/Documentation/Tests_videos/test_affichage_secondes_7segments.mov)

**Validité du test :** Test validé

### 3.2.5 Test du capteur CO2 avec alerte visuelle

**Description :** Le but de ce test est de faire clignoter toutes les LED simultanément de l’horloge 60 LED quand le taux de CO2 mesuré par le capteur SGP30 dépasse le seuil voulu. Le test a été effectué par moi-même.

**Matériel utilisé :** L’horloge 60 LED ainsi que le capteur SGP30 sont branchées à la carte Arduino lui-même alimenté à un ordinateur tournant sous windows 10 avec un câble USB 2.0 A-B.

**Résultat attendu :** Une fois que le taux de CO2 dépasse le seuil voulu, chaque LED de l’horloge 60 LED clignotent simultanément.

**Résultat final :** Une fois que le taux de CO2 dépasse le seuil voulu, chaque LED de l’horloge à LED clignotent simultanément.

**Validité du test :** Test validé

### 3.2.6 Test des options d’alertes

**Description :** Le but de ce test est de pouvoir sélectionner une option d’alerte en pressant sur un bouton poussoir vert. De base, l’option d’alerte visuelle et sonore est sélectionnée. Le test a été effectué par moi-même.

**Matériel utilisé :** Un buzzer, l’horloge 60 LED, un bouton poussoir vert ainsi que le capteur SGP30 sont branchées à la carte Arduino lui-même alimenté à un ordinateur tournant sous windows 10 avec un câble USB 2.0 A-B.

**Résultat attendu :**

Le bouton poussoir vert est pressé :

* Si l’option actuellement sélectionnée est l’option d’alerte visuelle et sonore, alors sélectionne l’option d’alerte uniquement visuelle.
* Si l’option actuellement sélectionnée est l’option d’alerte uniquement visuelle, alors sélectionne l’option d’alerte uniquement sonore.
* Si l’option actuellement sélectionnée est l’option d’alerte uniquement sonore, alors sélectionne l’option aucune alerte.
* Si l’option actuellement sélectionnée est l’option aucune alerte, alors sélectionne l’option d’alerte visuelle et sonore.

**Résultat final :**

Le bouton poussoir vert est pressé :

* Si l’option actuellement sélectionnée est l’option d’alerte visuelle et sonore, alors sélectionne l’option d’alerte uniquement visuelle.
* Si l’option actuellement sélectionnée est l’option d’alerte uniquement visuelle, alors sélectionne l’option d’alerte uniquement sonore.
* Si l’option actuellement sélectionnée est l’option d’alerte uniquement sonore, alors sélectionne l’option aucune alerte.
* Si l’option actuellement sélectionnée est l’option aucune alerte, alors sélectionne l’option d’alerte visuelle et sonore.

**Validité du test :** Test validé

### 3.2.7 Test des modes d’affichage

**Description :** Le but de ce test est de pouvoir sélectionner le contenu de l’affichage 7 segments avec un bouton poussoir bleu. De base, l’affichage 7 segments affiche la température / heure / taux de CO2 en alternance. Le test a été effectué par moi-même.

**Matériel utilisé :**  L’affichage 7 segments, un bouton poussoir bleu, un capteur de température BME280 ainsi que le capteur SGP30 sont branchées à la carte Arduino lui-même alimenté à un ordinateur tournant sous windows 10 avec un câble USB 2.0 A-B.

**Résultat attendu :**

Le bouton poussoir bleu est pressé :

* Si le mode d’affichage actuellement sélectionné est l’affichage de la température / heure / taux de CO2 en alternance, alors sélectionne le mode d’affichage de la température.
* Si le mode d’affichage actuellement sélectionné est l’affichage de la température, alors sélectionne le mode d’affichage de l’heure.
* Si le mode d’affichage actuellement sélectionné est l’affichage de l'heure, alors sélectionne le mode d’affichage du taux de CO2.
* Si le mode d’affichage actuellement sélectionné est l’affichage du taux de CO2 alors sélectionne le mode d’affichage de la température / heure / taux de CO2 en alternance.

**Résultat final :**

Le bouton poussoir bleu est pressé :

* Si le mode d’affichage actuellement sélectionné est l’affichage de la température / heure / taux de CO2 en alternance, alors sélectionne le mode d’affichage de la température.
* Si le mode d’affichage actuellement sélectionné est l’affichage de la température, alors sélectionne le mode d’affichage de l’heure.
* Si le mode d’affichage actuellement sélectionné est l’affichage de l'heure, alors sélectionne le mode d’affichage du taux de CO2.
* Si le mode d’affichage actuellement sélectionné est l’affichage du taux de CO2 alors sélectionne le mode d’affichage de la température / heure / taux de CO2 en alternance.

**Validité du test :** Test validé

## 3.3 Erreurs restantes

Figure 16 Erreurs restantes



## 3.4 Liste des documents fournis

### 3.4.1 Format électronique

- Dossier de projet [Word] [PDF]

- Journal de travail [Excel] [PDF]

- Cahier des charges [PDF]

- Un canevas d’un dossier de projet qui contient les points importants à indiquer dans le dossier de projet [Word]

- Planning initial [MS Project] [PDF]

- Les erreurs restantes [Excel]

- Les diagrammes de flux [Visio]

- Résumé du rapport TPI [Word]

- Des vidéos montrant le résultat final de certains tests [Vidéo]

- Un dossier contenant tout ce qui est en lien avec le code, comme des exemples de prise en mains de certains capteurs ou encore basiquement le code du projet principal.

### 3.4.2 Format papier

- Dossier de projet

- Journal de travail

- Résumé du rapport TPI

# 4. Conclusion

## 4.1 Objectifs atteints / non atteints

### 4.1.1 Objectifs atteints

Pour la partie « Affichage de l’heure » l’affichage des heures ainsi que les minutes sont disponibles sur l’affichage 7 segments de même pour l’affichage des secondes indiqué avec un clignotement de deux petits points. L'heure est synchronisée avec une RTC.

Pour la partie « Affichage du taux de CO2 avec alerte », une alarme clignotante visuelle ainsi qu’une alarme sonore est générée dans le cas où le taux de CO2 dépassé un certain seuil prédéfini. Un bouton poussoir permettant de changer d’options d’alerte est disponible. Les alarmes s’arrêtent au bout d’un certain temps et recommence après un second laps de temps si le taux de CO2 est toujours trop élevé. J’ai décidé de faire durer la ou les alertes au maximum pendant 2 secondes suivit d’un délai de 5 secondes sans que les alertes ne s'enclenchent même si le taux de CO2 est trop élevé. Au bout de ces 5 secondes, si le taux de CO2 est toujours trop élevé ou qu’il dépasse la limite prédéfinie alors les alertes se réenclenchent durant au maximum 2 secondes. J’ai ajouté une fonctionnalité qui permet au moment où le taux de CO2 est trop élevé, donc durant la génération des alarmes, le taux de CO2 est instantanément indiqué au même moment sur l’affichage 7 segments et une fois que le taux de CO2 devient inférieur à la limite ou une fois 2 secondes écoulées, l’affichage 7 segments reprends son mode d’affichage.

Pour la partie « Affichage de température », la température peut être affichée sur l’affichage 7 segments. Un bouton poussoir permettant de changer de mode d’affichage pour l’affichage 7 segments est disponible.

### 4.1.2 Objectifs non atteints

L’heure n’est pas disponible sur l’horloge 60 LED. Ayant besoin plusieurs secondes avant d’afficher la réelle valeur, le capteur du taux de CO2 (SGP30) affichera durant ces 15 secondes « 400 » en permanence, il n’y a aucune indication que le capteur a besoin d’un certain temps avant d’afficher des valeurs correctes.

## 4.2 Points positifs / négatifs

### 4.2.1 Points positifs

J’ai énormément apprécié travailler sur un projet d’informatique embarqué, j’ai acquis de bonnes bases techniques dans plusieurs petits domaines comme le montage des composants, la soudure, j’ai de même continué à m’améliorer et à me familiariser avec le langage C. Là où je me suis le plus amélioré selon moi est pour la partie d’analyse, même si j’ai mis beaucoup de temps pour la réaliser, je me sens maintenant beaucoup plus à l’aise pour sa réalisation et j’ai bien compris à quel point elle avait de l’importance pour toute la suite du projet. J’ai beaucoup apprécié la présence de mon chef de projet (M. Favre) qui m’a aiguillé du début à la fin du projet. Les quelques conseils donnés par les deux experts ont bien aidé à la bonne réalisation du projet.

### 4.2.2 Points négatifs

J’ai pris pas mal de temps pour réaliser la partie analyse, ce qui m’a obligé à réduire le temps pour la partie d’implémentation. J’ai trouvé le temps mis à disposition pour réaliser ce TPI très « juste », dans le sens où on ne peut vraiment pas rester bloqué longtemps, chaque minute compte.

## 4.3 Difficultés particulières

Tout comme mon ancien projet Pré-TPI, j’ai rencontré pas mal de difficultés à m’exprimer à l’écrit, ça m’est arrivé plusieurs fois de bloquer sur comment amener telle information dans mon journal de bord ou écrire telle explication dans le dossier de projet. J’ai aussi pas mal bloqué sur la réalisation du diagramme de flux qui m’a demandé plusieurs retouches.

## 4.4 Suites possibles pour le projet

Implémenter la ou les fonctionnalités permettant d’afficher l’heure sur l’horloge 60 LED compléterait bien le projet. Ajouter une ou plusieurs indications indiquant, au moment où l’on presse sur l’un des deux boutons poussoirs, quel est le mode d’affichage ou l’option d’alerte que l’on vient de sélectionner.

## 4.5 Bilan personnel

Ce projet a été dans sa globalité une bonne expérience et j’ai beaucoup apprécié le réaliser. Je me suis amélioré dans pas mal d’aspect dans le domaine de réalisation de projet, surtout dans la partie analyse/ conception. Aucun problème majeur n’est apparu dans le projet et comme dis plus haut dans les points positifs, mon chef de projet ainsi que mes experts ont été là pour m’aider du début à la fin du projet. J’ai aussi constaté plusieurs points dans lesquels je dois encore m’améliorer comme la rédaction de texte ou encore la gestion de temps de chaque partie.

# 5. Annexes

## 5.1 Résumé rapport du TPI

### 5.1.1 Situation de départ

Le projet a débuté le 2 mai 2022, la première chose que j’ai effectué a été de prendre connaissance de mon CDC pour ensuite le validé avec mon premier expert ainsi que mon chef de projet. J’ai ensuite directement commencé à réaliser la partie mise en place du projet, c’est-à-dire la création d’un dépôt distant GitHub, d’un tableau Trello avec plusieurs sprints pour le suivi du projet complet, puis réaliser une bonne structure de dossiers. J’ai ensuite effectué la partie d’analyse/conception du projet avec la réalisation, entre autres, d’une planification initiale, de la stratégie de test, de uses cases scénario ainsi que plusieurs diagrammes de flux qui ont pris un certain temps à être réalisé correctement.

### 5.1.2 Mise en œuvre

Je me suis ensuite penché sur la partie d’implémentation qui a pas mal été raccourcie à cause du temps de réalisation de la partie d’analyse/conception. J’ai donc entamé cette partie avec un entrainement de soudure de plusieurs petits composants inutiles comme des résistances ou encore des jumpers. J’ai ensuite enchainé avec la soudure de certains composants nécessaire au projet comme l’affichage 7 segments, capteur SGP30 ou encore l’horloge à 60 LED formée de 4 parties de 15 LED. Viens ensuite la partie montage et test de tous les composants que je dispose qui fut relativement rapide et sans problèmes majeurs, à noter qu’une partie du montage complet du projet TPI a été pris de mon ancien projet Pré-TPI. Après avoir réalisé un montage complet fonctionnel et avoir pris en main chaque capteur du projet, je me suis lancé dans l’implémentation des fonctionnalités demandées. J’ai décidé de stopper la partie d’implémentation en fin de semaine 3/ début de semaine 4 pour pouvoir me focaliser sur les dernières étapes du projet comme la réalisation de certains documents et la finalisation de la documentation.

### 5.1.3 Résultats

La majorité des résultats ont été convaincants, que ce soit pour la partie d’analyse qui a certes pris du temps, mais qui s’est avéré très utile pour la suite du projet, de même pour la partie d’implémentation avec une majorité des objectifs qui ont été atteints. Pour la partie réalisation des tests, une grande majorité de ceux-ci ont été validés, certains plus tard que d’autres. Les résultats liés aux objectifs plus « personnel » comme la tenue du journal de travail, la tenue du dossier de projet dans sa globalité, la ponctualité avec les différents rendez-vous organisé tout au long du projet ont été, de manière générale, bons. Quelques petits ajustements liés aux remarques du journal de travail ou encore à la rédaction des différents noms et descriptions des commits sur le dépôt distant ont dû être fait pour obtenir un résultat plus correct.

## 5.2 Sources – Bibliographie

- Utilisation du Livre « LE LIVRE DE PROJET ARDUINO » pour certains points de base avec Arduino

- Outils de versionning : <https://github.com/>

- Vérification des fautes d’orthographes : <https://languagetool.org/fr>

- Explication d’une LED : <https://fr.wikipedia.org/wiki/Diode_%C3%A9lectroluminescente>

- Explication RGB : <https://fr.wikipedia.org/wiki/Rouge_vert_bleu>

- Exemple de soudure pour l’horloge à 60 LED : <https://www.youtube.com/watch?v=EXr2_zSfnFw>

- Logo du CPNV : <https://www.cpnv.ch/>

- Réalisation du diagramme de flux : <https://app.diagrams.net/>

- Branchement du capteur de CO2 (SGP30) : <https://learn.adafruit.com/adafruit-sgp30-gas-tvoc-eco2-mox-sensor/pinouts>

- Compréhension du capteur de CO2 (SGP30) : <https://learn.adafruit.com/adafruit-sgp30-gas-tvoc-eco2-mox-sensor>

- Fonctions liées à la librairie du capteur de CO2 (SGP30) « Adafruit\_SGP30 » : <https://adafruit.github.io/Adafruit_SGP30/html/class_adafruit___s_g_p30.html#a03d6f71c0670a46aeeeb4e050c6585b7>

- Compréhension de l’affichage 7 segments : <https://www.adafruit.com/product/879>

- Fonction liées à la librairie « Adafruit\_LEDBackpack.h » de l’affichage 7 segments : <https://github.com/adafruit/Adafruit_LED_Backpack>

- Fonctions liées à la librairie « Adafruit\_NeoPixel.h » pour l’horloge 60 LED : <https://adafruit.github.io/Adafruit_NeoPixel/html/class_adafruit___neo_pixel.html>

- Compréhension de certains types de variable : <https://www.tutorialspoint.com/cprogramming/c_data_types.htm>

- Installation de l’IDE d’Arduino : <https://www.arduino.cc/en/software/OldSoftwareReleases>

- Installation de « GitHub desktop » : <https://desktop.github.com/>

## 5.3 Journal de travail

Figure 17 Journal de travail



Figure 18 Journal de travail



Figure 19 Journal de travail



Figure 20 Journal de travail

## 5.4 Glossaire

**Affichage 7 segments / affichage 4x « 7-segments » / affichage 7-segments :** Un rassemblement de 4 affichages possédants chacun 7 digits.

**Capteur BME280 :** Le capteur BME280 est une « combinaison » de trois capteurs, il possède un capteur de température, d’humidité et de pression atmosphérique.

**Capteur SGP30 :** Le capteur SGP30 permet de mesurer la « qualité » de l’air comme le taux de CO2.

**Horloge 60 LED :** Soudure entre 4 parties de 15 LED donc 60 LED.

**IDE :** De l’anglais « Integrated Development Environment », est un logiciel qui permet tout simplement de réaliser notre code.

**I2C :** I2C pour « Inter-Integrated Circuit est un protocole qui permet la communication entre des dispositifs ou des puces dans un système embarqué.

**LED :** LED, diode électroluminescente ou encore « light-emitting diode » en anglais, est un dispositif optoélectronique qui émet de la lumière lorsqu'un courant électrique le traverse.

**RGB :** « Red Breen Blue » de l’anglais, est un système, dans le domaine de l’informatique, de codage pour les couleurs.

**RTC :** RTC pour « Real Time Clock » est un module qui permet de retourner l’heure et la date courante.

## 5.5 Table des illustrations

[Figure 1 Semaine 1 5](file:///F:\TPI\Projet_TPI_EGS\Documentation\Dossier_Projet_EGS.docx#_Toc104837778)

[Figure 2 Semaine 2 5](file:///F:\TPI\Projet_TPI_EGS\Documentation\Dossier_Projet_EGS.docx#_Toc104837779)

[Figure 3 Semaine 3 6](file:///F:\TPI\Projet_TPI_EGS\Documentation\Dossier_Projet_EGS.docx#_Toc104837780)

[Figure 4 Semaines 4 et 5 6](file:///F:\TPI\Projet_TPI_EGS\Documentation\Dossier_Projet_EGS.docx#_Toc104837781)

[Figure 5 Diagramme de flux du programme général 14](#_Toc104837782)

[Figure 6 Diagramme de flux pour l'option "Génération de l’alerte" 15](#_Toc104837783)

[Figure 7 Diagramme de flux pour l'option "Lecture du bouton poussoir option d’alerte" 16](#_Toc104837784)

[Figure 8 Diagramme de flux pour l'option "Lecture du bouton poussoir mode d’affichage 7 segments" 17](#_Toc104837785)

[Figure 9 Dépôt distant GitHub du projet 18](file:///F:\TPI\Projet_TPI_EGS\Documentation\Dossier_Projet_EGS.docx#_Toc104837786)

[Figure 10 Tableau Trello utilisé pour le suivi du projet 19](file:///F:\TPI\Projet_TPI_EGS\Documentation\Dossier_Projet_EGS.docx#_Toc104837787)

[Figure 11 Contenu du dossier "Code" 22](#_Toc104837788)

[Figure 12 Contenu du dossier "Documentation" 23](#_Toc104837789)

[Figure 13 Branchement de l'horloge au complet 24](file:///F:\TPI\Projet_TPI_EGS\Documentation\Dossier_Projet_EGS.docx#_Toc104837790)

[Figure 14 Projet final 25](file:///F:\TPI\Projet_TPI_EGS\Documentation\Dossier_Projet_EGS.docx#_Toc104837791)

[Figure 15 Résultat final du test « Test de l’affichage 7 segments » 27](file:///F:\TPI\Projet_TPI_EGS\Documentation\Dossier_Projet_EGS.docx#_Toc104837792)

[Figure 16 Erreurs restantes 31](file:///F:\TPI\Projet_TPI_EGS\Documentation\Dossier_Projet_EGS.docx#_Toc104837793)

[Figure 29 Journal de travail 36](#_Toc104837794)

[Figure 30 Journal de travail 37](#_Toc104837795)

[Figure 31 Journal de travail 38](#_Toc104837796)

[Figure 32 Journal de travail 39](#_Toc104837797)

[Figure 17 Projet final 41](file:///F:\TPI\Projet_TPI_EGS\Documentation\Dossier_Projet_EGS.docx#_Toc104837798)

[Figure 18 Installation de la version 1.8.12 de l'IDE d'Arduino 42](file:///F:\TPI\Projet_TPI_EGS\Documentation\Dossier_Projet_EGS.docx#_Toc104837799)

[Figure 19 Bouton d'installation pour "GitHub Desktop" 42](file:///F:\TPI\Projet_TPI_EGS\Documentation\Dossier_Projet_EGS.docx#_Toc104837800)

[Figure 20 Clonage du dépôt distant du projet 43](file:///F:\TPI\Projet_TPI_EGS\Documentation\Dossier_Projet_EGS.docx#_Toc104837801)

[Figure 21 Installation de la librairie "Adafruit BME280 Library" 44](file:///F:\TPI\Projet_TPI_EGS\Documentation\Dossier_Projet_EGS.docx#_Toc104837802)

[Figure 22 Installation de la librairie "RTClib" 44](file:///F:\TPI\Projet_TPI_EGS\Documentation\Dossier_Projet_EGS.docx#_Toc104837803)

[Figure 23 Installation de la librairie "Adafruit LED Backpack Library" 44](file:///F:\TPI\Projet_TPI_EGS\Documentation\Dossier_Projet_EGS.docx#_Toc104837804)

[Figure 24 Installation de la librairie "Adafruit SGP30 Sensor" 45](file:///F:\TPI\Projet_TPI_EGS\Documentation\Dossier_Projet_EGS.docx#_Toc104837805)

[Figure 25 Installation de la librairie "Adafruit NeoPixel" 45](file:///F:\TPI\Projet_TPI_EGS\Documentation\Dossier_Projet_EGS.docx#_Toc104837806)

[Figure 26 Branchement de l'horloge à l'ordinateur 45](file:///F:\TPI\Projet_TPI_EGS\Documentation\Dossier_Projet_EGS.docx#_Toc104837807)

[Figure 27 Bouton permettant la compilation du code 46](file:///F:\TPI\Projet_TPI_EGS\Documentation\Dossier_Projet_EGS.docx#_Toc104837808)

[Figure 28 Branchement de l'horloge au complet 46](file:///F:\TPI\Projet_TPI_EGS\Documentation\Dossier_Projet_EGS.docx#_Toc104837809)

## 5.6 Manuel d’installation et de mise en place

### 5.6.1 Matériel nécessaire

- L’horloge avec le montage complet :

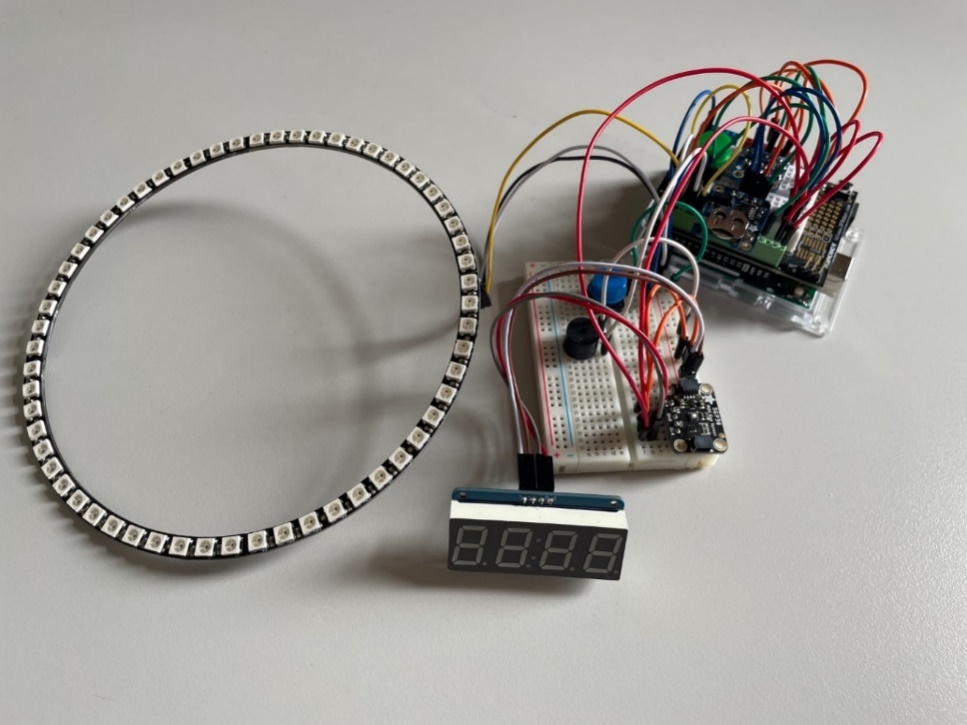


Figure 21 Projet final

- Un câble USB A-B

- Ordinateur tournant sous Windows 10

### 5.6.2 Installation des logiciels

#### 5.6.2.1 Installation de la version 1.8.12 de l’IDE d’Arduino

1) Rendez-vous sur la page des différentes versions d’IDE d’Arduino, sur le site d’Arduino, [ici](https://www.arduino.cc/en/software/OldSoftwareReleases)

2) Aller jusqu’au bloc des versions « Arduino 1.8.x », cherchez la version 1.8.12 puis cliquez sur « Windows Windows Installer », surligné en bleu sur l’image ci-dessous :

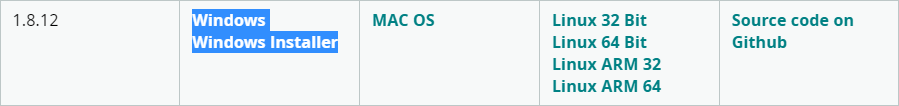


Figure 22 Installation de la version 1.8.12 de l'IDE d'Arduino

3) Terminez l’installation

#### 5.6.2.2 Installation de la dernière version de github desktop

1) Rendez-vous sur la page d’installation de « GitHub Desktop », [ici](https://desktop.github.com/)

2) Pressez sur le gros bouton de couleur :

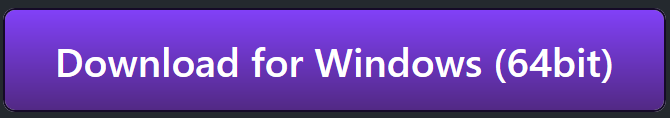


Figure 23 Bouton d'installation pour "GitHub Desktop"

3) Terminez l’installation

### 5.6.3 Clonage du projet

1) Rendez-vous sur la page du dépôt distant GitHub du projet, [ici](https://github.com/EGiorgisCPNV/Projet_TPI_EGS)

2) Cliquez sur « Code » puis « Ouvrir avec GitHub Desktop » :

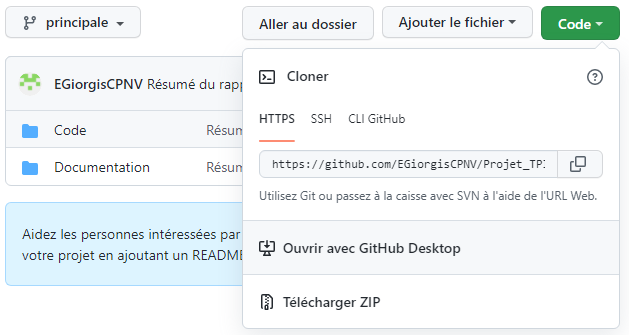


Figure 24 Clonage du dépôt distant du projet

3) Choisissez un emplacement dans votre ordinateur pour stocker le projet

### 5.6.4 Ouvrir le fichier du code du projet

1) Une fois le projet stocké en local, dirigez-vous à cet emplacement : « Code > Projet\_Principal ».

2) Ouvrez le fichier « Projet\_Principal.ino »

### 5.6.5 Installation des différentes librairies

1) Rendez-vous dans le gestionnaire de bibliothèque, pour faire cela cliquer dans le ruban en haut à gauche de l’IDE, sur les options « Outils > Gérer les bibliothèques »

2) Cherchez dans la barre de recherche en haut à droite « Adafruit\_BME280.h », dans la case « Adafruit BME280 Library » installez la version 2.2.2.

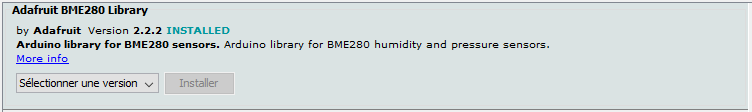


Figure 25 Installation de la librairie "Adafruit BME280 Library"

3) Cherchez dans la barre de recherche en haut à droite « RTClib.h », dans la case « RTClib » installez la version 2.0.2.

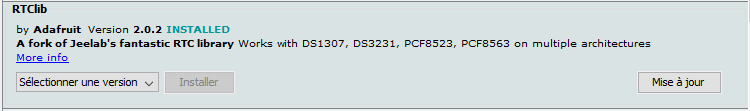


Figure 26 Installation de la librairie "RTClib"

4) Cherchez dans la barre de recherche en haut à droite « Adafruit Backpack », dans la case « Adafruit LED Backpack Library » installez la version 1.3.2.

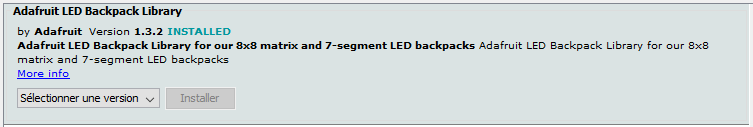


Figure 27 Installation de la librairie "Adafruit LED Backpack Library"

Attention, il se peut que cette librairie demande l’installation d’autres librairies au quelle elle dépend, si c’est le cas veuillez les installer.

5) Cherchez dans la barre de recherche en haut à droite « Adafruit\_SGP30.h », dans la case « Adafruit SGP30 Sensor » installez la version 2.0.0.



Figure 28 Installation de la librairie "Adafruit SGP30 Sensor"

6) Cherchez dans la barre de recherche en haut à droite « Adafruit\_NeoPixel.h », dans la case « Adafruit NeoPixel » installez la version 1.10.5.

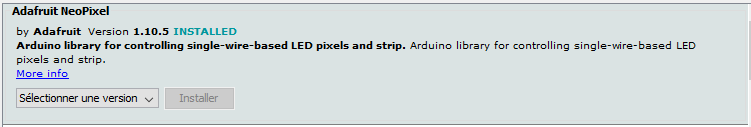


Figure 29 Installation de la librairie "Adafruit NeoPixel"

### 5.6.6 Branchement de l’horloge à l’ordinateur

1) Brancher la carte Arduino à l’ordinateur à l’aide du câble USB A-B

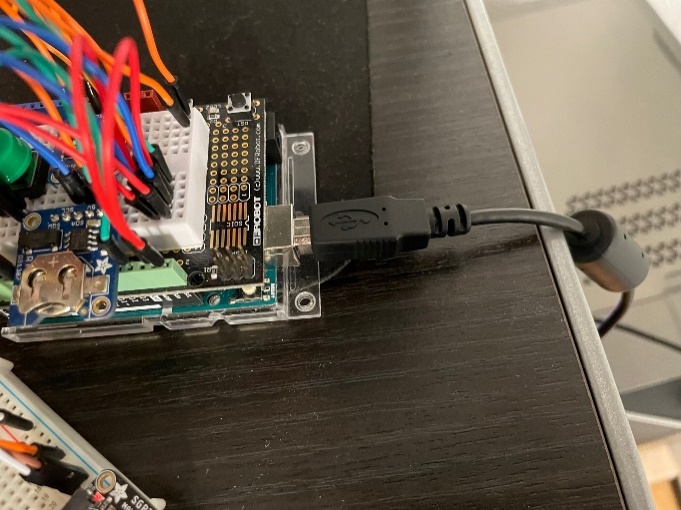


Figure 30 Branchement de l'horloge à l'ordinateur

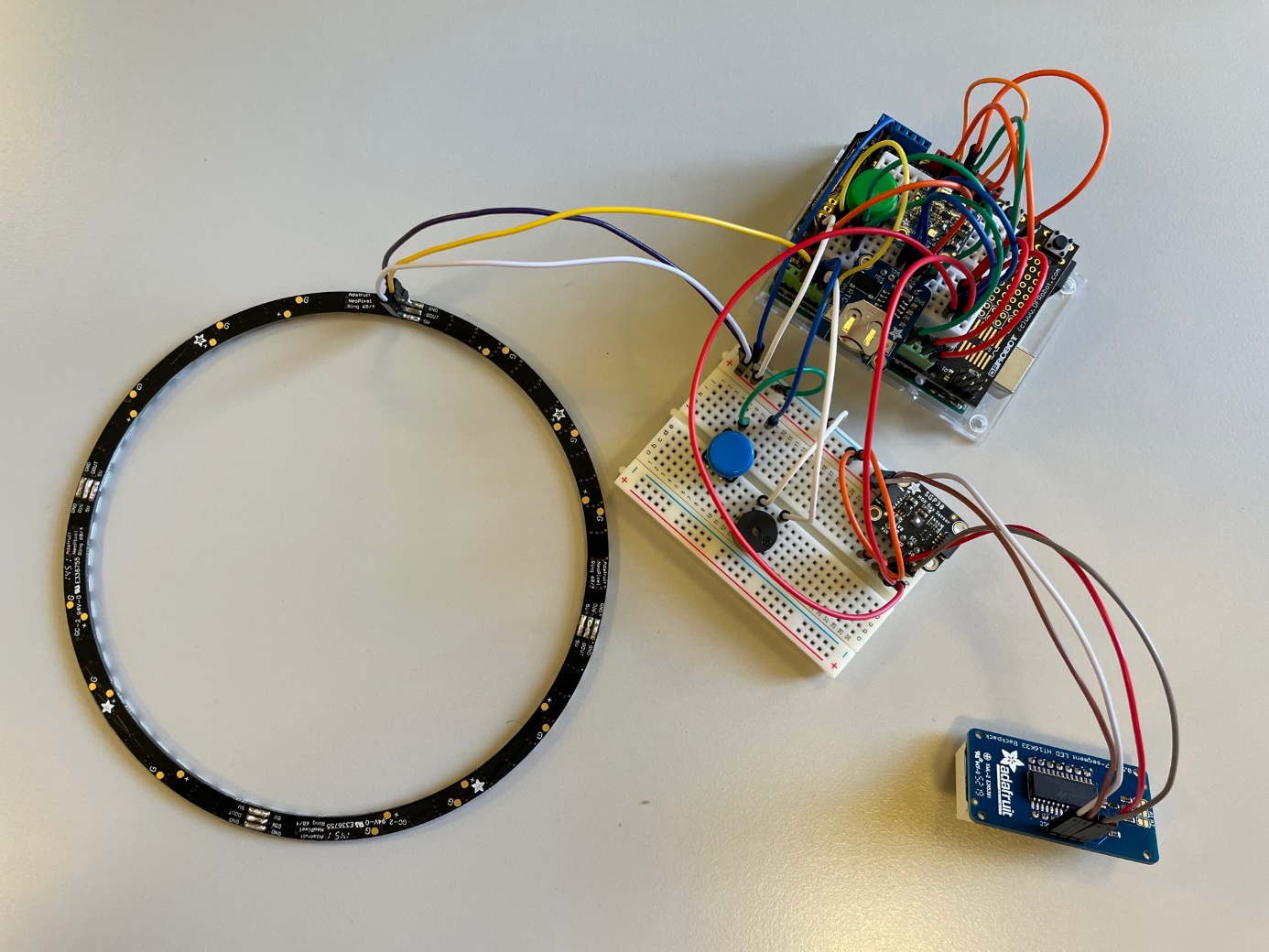
### 5.6.7 Compiler le code

1) Cliquez sur le bouton rond avec une flèche droite comme logo intérieur :



Figure 31 Bouton permettant la compilation du code

## 5.7 Manuel d’utilisation



1

7

8

4

2

9

6

3

5

Figure 32 Branchement de l'horloge au complet

Une fois le produit alimenté, l’affichage 7 segment (N°7) affichera en alternance la température / l’heure / le taux de CO2.

En pressant le bouton poussoir bleu (N°2), vous changez le mode d’affichage du 7 segments.

En pressant le bouton poussoir vert (N°2), vous changez l’option d’alerte en cas de trop haut taux de CO2.