**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 2](#_Toc104272530)

[1.1 Introduction 2](#_Toc104272531)

[1.2 Organisation 2](#_Toc104272532)

[1.3 Objectifs 3](#_Toc104272533)

[1.4 Objectifs personnels 3](#_Toc104272534)

[1.5 Planification initial 4](#_Toc104272535)

[2. Analyse / Conception 6](#_Toc104272536)

[2.1 Stratégie de test 6](#_Toc104272537)

[2.2 Uses cases scénario 8](#_Toc104272538)

[2.3 Risques techniques 11](#_Toc104272539)

[2.4 Dossier de conception 12](#_Toc104272540)

[2.5 Dépôt distant 17](#_Toc104272541)

[3. Réalisation 18](#_Toc104272542)

[3.1 Dossier de réalisation 18](#_Toc104272543)

[3.2 Description des tests éffectué 22](#_Toc104272544)

[3.3 Liste des documents fournis 27](#_Toc104272545)

[4. Annexes 28](#_Toc104272546)

[4.1 Sources – Bibliographie 28](#_Toc104272547)

[4.2 Glossaire 29](#_Toc104272548)

[4.3 Table des illustrations 29](#_Toc104272549)

# 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

### 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

### 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 initial

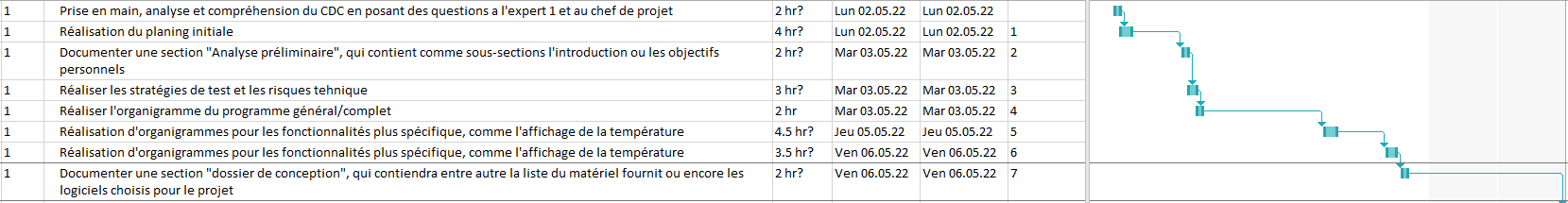


Figure 1 Semaine 1

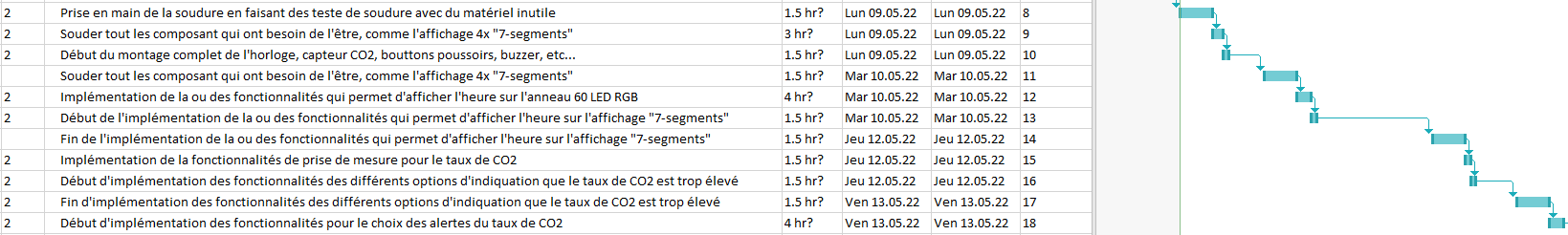


Figure 2 Semaine 2

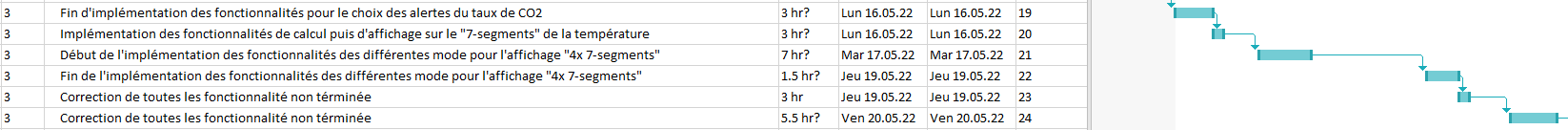


Figure 3 Semaine 3

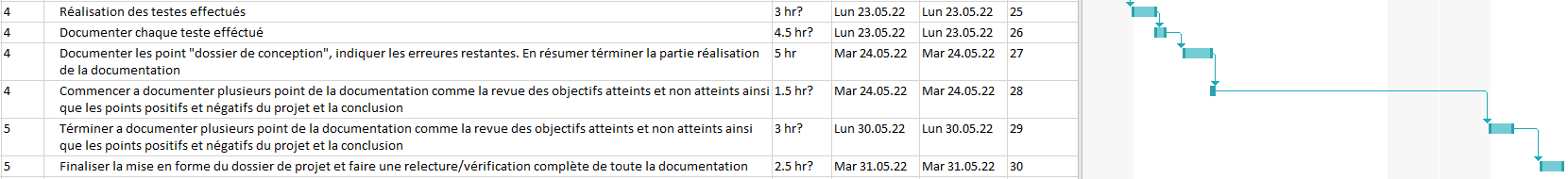


Figure 4 Semaines 4 et 5

# 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 indiquée 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 situé 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 Dossier de conception

### 2.4.1 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…)

### 2.4.2 Système d’exploitation utilisé

* Windows 10 Education, Version 21H2

### 2.4.3 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

### 2.4.4 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"

## 2.5 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.

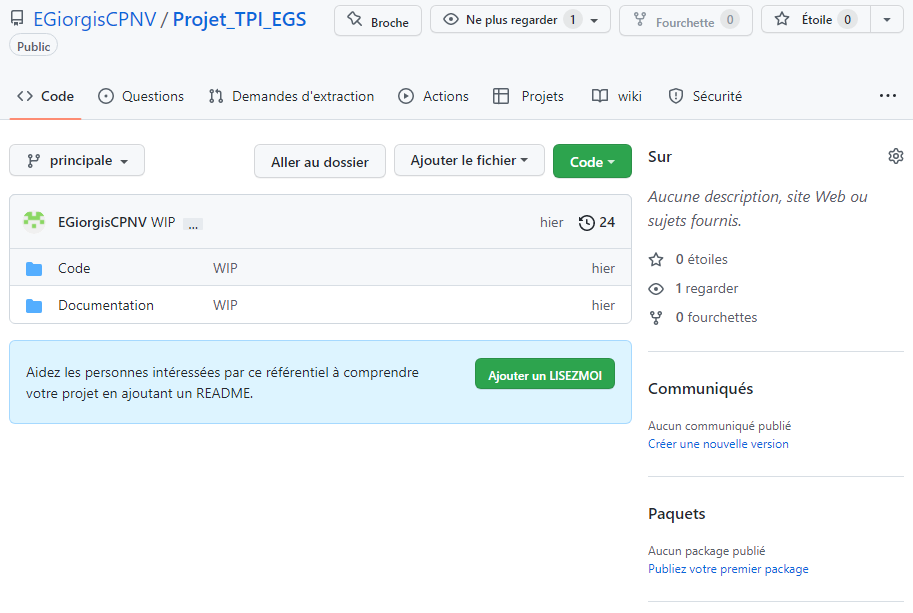


Figure 9 Dépôt distant GitHub du projet

# 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 Branchement final de l’horloge au complet

7

2

1

8

5

6

3

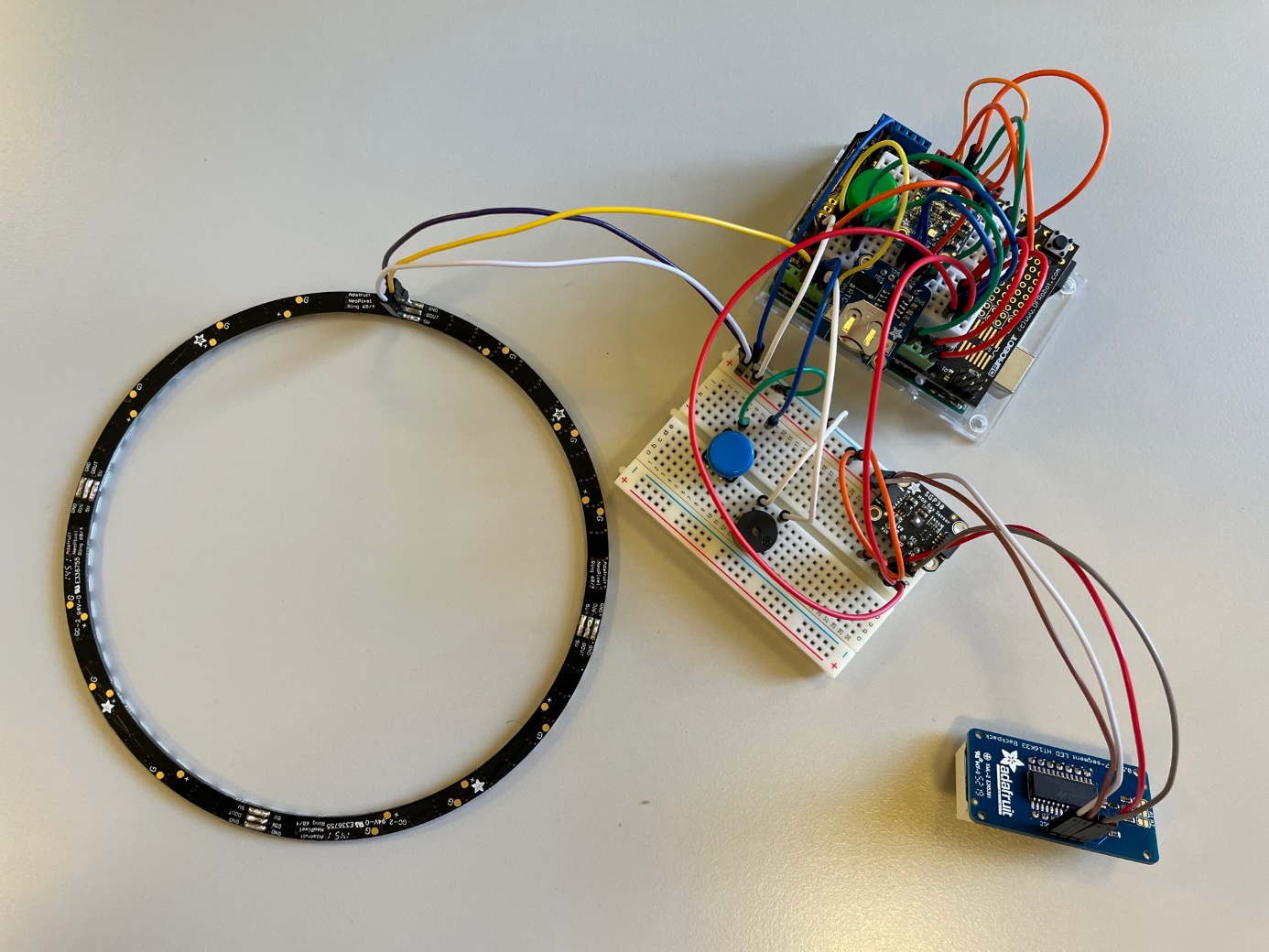


Figure 10 Branchement de l'horloge

4

9

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

### 3.1.4 Projet final

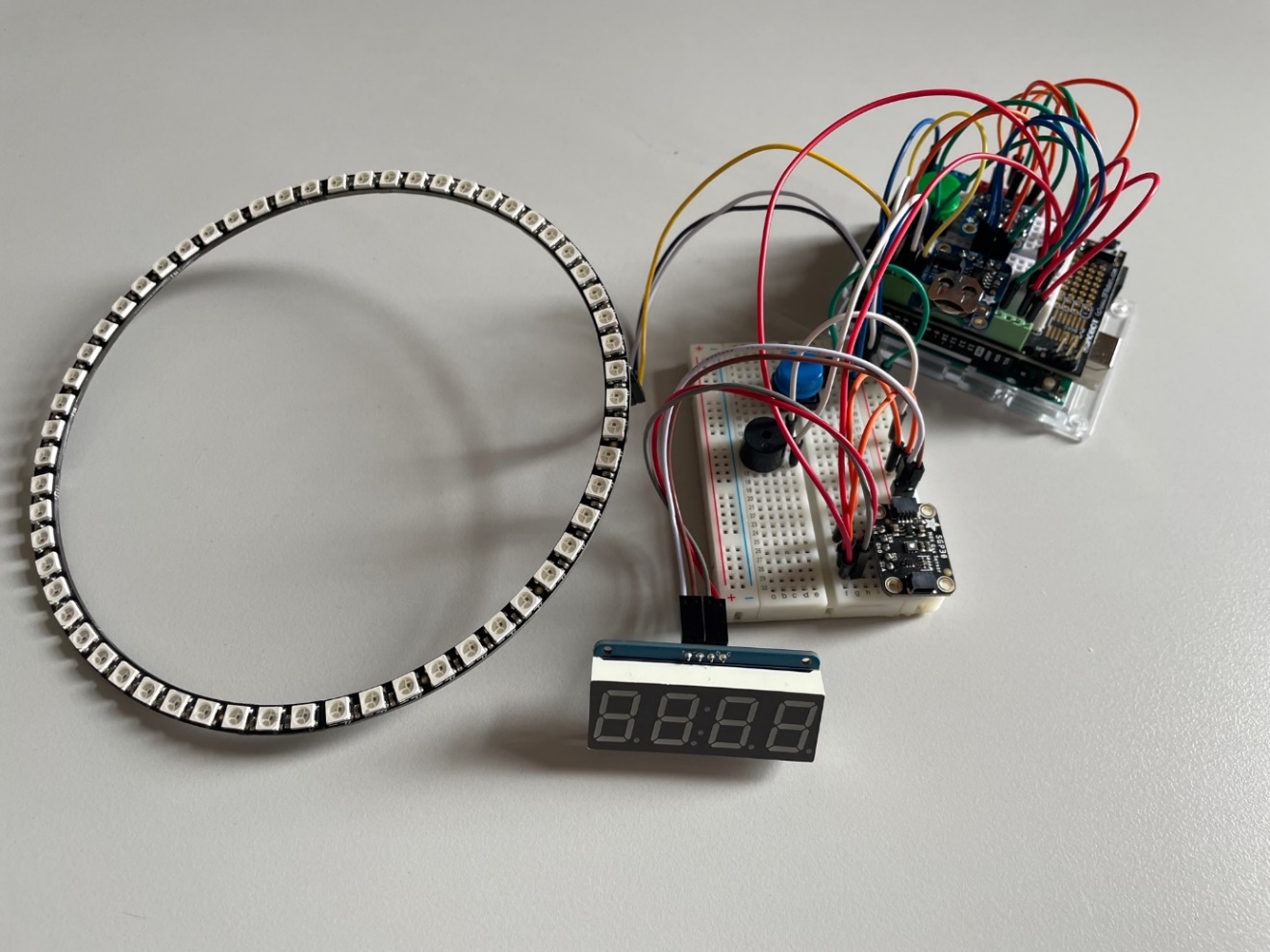


Figure 11 Projet final

### 3.1.5 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 éffectué

### 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.

**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.

**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.

**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.

**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 qui clignote chaque seconde.

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

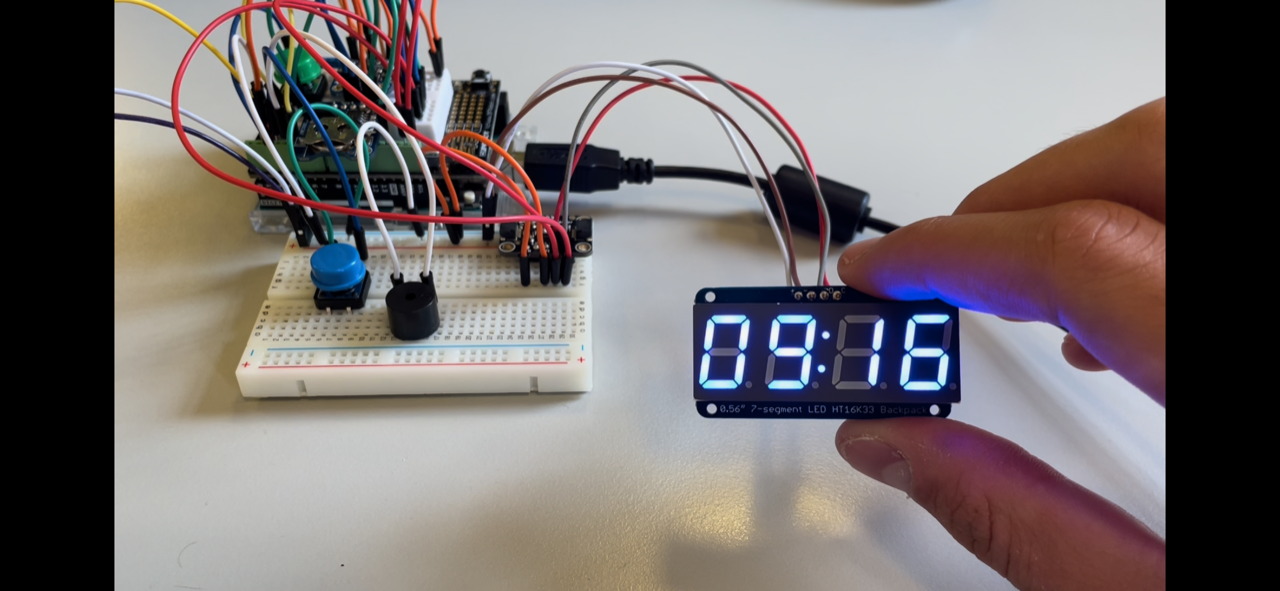


Figure 11 Résultat final du 2ème test

**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.

**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.

**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.

**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.

**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 est 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.

**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 Liste des documents fournis

### 3.3.1 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 initiale [MS Project]

- Planification finale [MS Project]

- Les tests effectués [Excel]

- Manuel d’installation et de mise en place [Word]

- Résumé du rapport TPI [Word]

- 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

### 3.3.2 Format papier

- Dossier de projet

- Journal de travail

- Procédure d’installation et de mise en service

# 4. Annexes

## 4.1 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>

## 4.2 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.

**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.

## 4.3 Table des illustrations

[Figure 1 Semaine 1 4](file:///C:\TPI\Projet_TPI_EGS\Documentation\Dossier_Projet_EGS.docx#_Toc104216898)

[Figure 2 Semaine 2 4](file:///C:\TPI\Projet_TPI_EGS\Documentation\Dossier_Projet_EGS.docx#_Toc104216899)

[Figure 3 Semaine 3 5](file:///C:\TPI\Projet_TPI_EGS\Documentation\Dossier_Projet_EGS.docx#_Toc104216900)

[Figure 4 Semaines 4 et 5 5](file:///C:\TPI\Projet_TPI_EGS\Documentation\Dossier_Projet_EGS.docx#_Toc104216901)

[Figure 5 Diagramme de flux du programme général 13](#_Toc104216902)

[Figure 6 Diagramme de flux pour l'option "Génération de l’alerte" 14](#_Toc104216903)

[Figure 7 Diagramme de flux pour l'option "Lecture du bouton poussoir option d’alerte" 15](#_Toc104216904)

[Figure 8 Diagramme de flux pour l'option "Lecture du bouton poussoir mode d’affichage 7 segments" 16](#_Toc104216905)

[Figure 9 Dépôt distant GitHub du projet 17](file:///C:\TPI\Projet_TPI_EGS\Documentation\Dossier_Projet_EGS.docx#_Toc104216906)

[Figure 10 Branchement de l'horloge 20](file:///C:\TPI\Projet_TPI_EGS\Documentation\Dossier_Projet_EGS.docx#_Toc104216907)

[Figure 11 Projet final 21](file:///C:\TPI\Projet_TPI_EGS\Documentation\Dossier_Projet_EGS.docx#_Toc104216908)

[Figure 11 Résultat final du 2ème test 23](file:///C:\TPI\Projet_TPI_EGS\Documentation\Dossier_Projet_EGS.docx#_Toc104216909)