



Rapport de Stage

Paul Bammez

Période du stage : Du 3 mai au 4 juin

Tuteur de stage : François Huchez

Etablissement / Formation : EPSI – Bachelor Devops

Entreprise d'accueil : Horloges Huchez – 4 rue de la Croix, 60420 Ferrières

Sommaire :

- I. Présentation de l'entreprise**
- II. Programmation du glas**
 - a. Présentation de la mission**
 - b. Solutions choisies et réalisation de la mission**
 - i. Etude du projet et familiarisation à Arduino**
 - ii. Réalisation de la sonnerie du glas**
 - iii. Automatisation des impulsions**
 - iv. Changement de certains composants et du câblage**
 - v. Câblage avec le véritable tableau de commandes**
 - c. Notice de fonctionnement et de l'installation**
 - d. Présentation du code final**
- III. Horloge Réglice**
 - a. Fonctionnement de l'horloge**
 - b. Missions à réaliser**
 - i. Recherche du motoréducteur**
 - ii. Recherche d'une carte**
 - iii. Choix des capteurs**
 - 1. Capteur optique**
 - 2. Capteur à effet Hall**
- IV. Réalisation de test**
 - a. Présentation du fonctionnement**
 - b. Exemple d'un calcul**
 - c. Tests réalisés**
- V. Configuration de badgeuses**
- VI. Annexes**

I. Présentation de l'entreprise

Horloges Huchez est une entreprise créée en 1832 par August-Lucien Vérité qui comprend aujourd'hui 17 employés. Il était passionné par l'horlogerie et a décidé de créer son entreprise. Traditionnellement experte campaniste, l'entreprise a élargie son offre. Horloges Huchez intervient en matière de campaniste et horloges monumentales, également tout ce qui concerne gestion de temps et le contrôle d'accès, la distribution d'heure et la sonorisation.

Horloges Huchez regroupe des clients en Europe, aux USA et au Moyen-Orient et donc est aujourd'hui un acteur majeur dans le marché des métiers du temps.

L'horlogerie monumentale comprend toutes les phases de fabrication d'une horloge, de l'aspect mécanique à l'aspect visuel, certaines horloges monumentales ont même été faites en collaboration avec des architectes et des designers.

Campaniste concerne la conception, réalisation, installation, restauration et l'entretien des cloches et horloges d'édifices comme des églises, cathédrales, gares, etc.

Concernant la gestion de temps avec les badgeuses, ils utilisent un logiciel de pointage « Thémis » qui est utilisé depuis 12 ans. L'installation se fait sur le poste du client avec également la base de données qui se trouvera sur le poste du client. Et actuellement un nouveau logiciel de pointage est en cours de développement « Réglice », ce logiciel est déjà utilisé par de nombreux Burger King pour gérer les présences et les heures de travail. La base de données est sur le cloud et le problème est que s'il y a un problème sur le logiciel ou autre, tous les clients ayant ce logiciel sont impactés et les modifications sont faites à tout le monde. Le logiciel Réglice ne va pas remplacer Thémis car ils sont différents et en fonction de la demande du client un logiciel peut être plus adapté que l'autre, de plus certaines personnes n'aiment pas avoir leurs données sur un serveur.

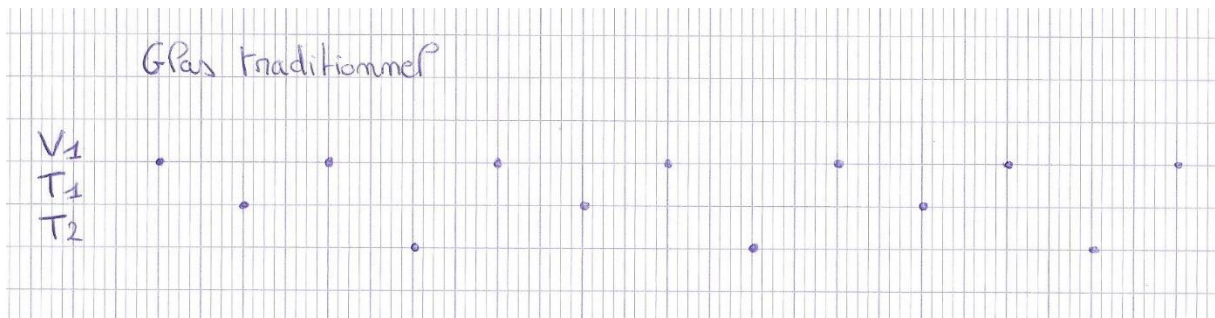
Les alertes et sonorisations comprennent toutes les alertes de sécurité ou musicales dans les établissements de santé, les lieux publics, les usines, les magasins et les établissements scolaires avec des sonneries musicales.

II. Programmation du Glas :

a. Présentation de la mission :

Dans le cadre de mon stage j'ai programmé le glas d'une commune nommée Saint-Remy-en-l'Eau. Le glas est la sonnerie des cloches signalant l'agonie, la mort ou les obsèques d'une personne, dans la tradition catholique et orthodoxe. Anciennement le glas était sonné dès le commencement de l'agonie d'une personne jusqu'à sa mort, ou alors entre sa mort et ses funérailles, et actuellement il est sonné uniquement pendant les obsèques.

Leur demande est spéciale car cette commune ne désire pas un glas traditionnel. Un glas traditionnel composé d'une cloche à la volée et de deux cloches à tintement à un ordre de sonnerie bien précis : on a la cloche à la volée qui se balance de gauche à droite et qui déclenche une impulsion lorsqu'elle passe au milieu et cette impulsion fait sonner une fois sur deux la première cloche à tintement puis la deuxième.



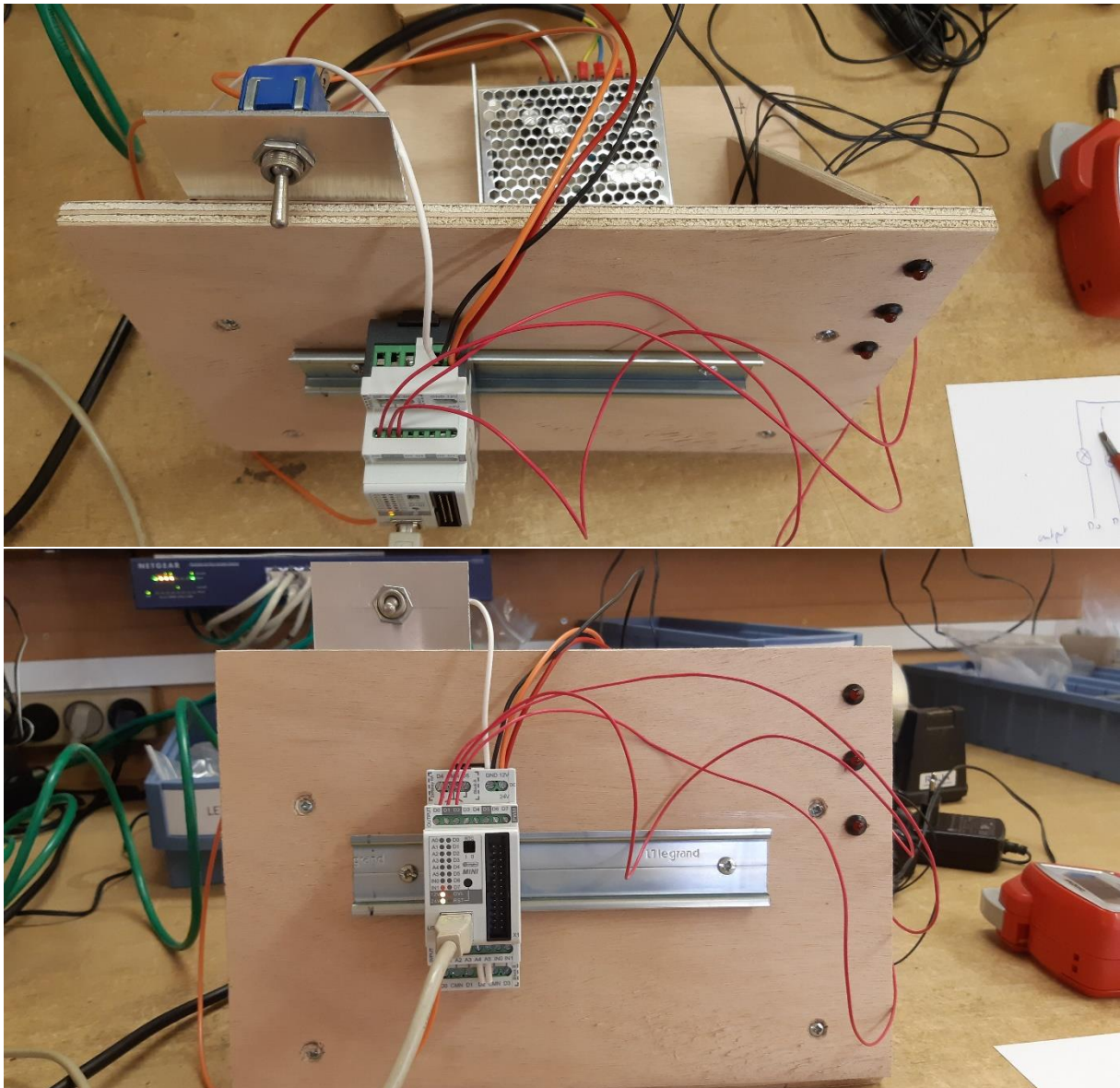
Cloche à la volée



Cloche à tintement



Or la demande de Saint-Remy-en-l'Eau consiste également à faire balancer une cloche à la volée qui elle aussi déclenche une impulsion quand elle passe au milieu. Mais contrairement au glas traditionnel, les deux premières impulsions font sonner la cloche à tintement numéro une puis la troisième impulsion fait sonner la deuxième cloche à tintement et ainsi de suite.



Ensuite je me suis familiarisé avec ce langage que je n'avais utilisé que très peu de fois au lycée. J'ai regardé des tutos et la façon de coder en Arduino sur internet pour savoir comment déclarer les variables, leur forme, etc.... J'ai donc simplement commencé par essayer de faire fonctionner les leds.

```
Code_arduino | Arduino 1.8.13 (Windows Store 1.8.42.0)
Fichier Edition Croquis Outils Aide

Code_arduino$

#include <ControlLine.h>

#define D2 CONTROLLING_D2
#define D3 CONTROLLING_D3
#define D4 CONTROLLING_D4

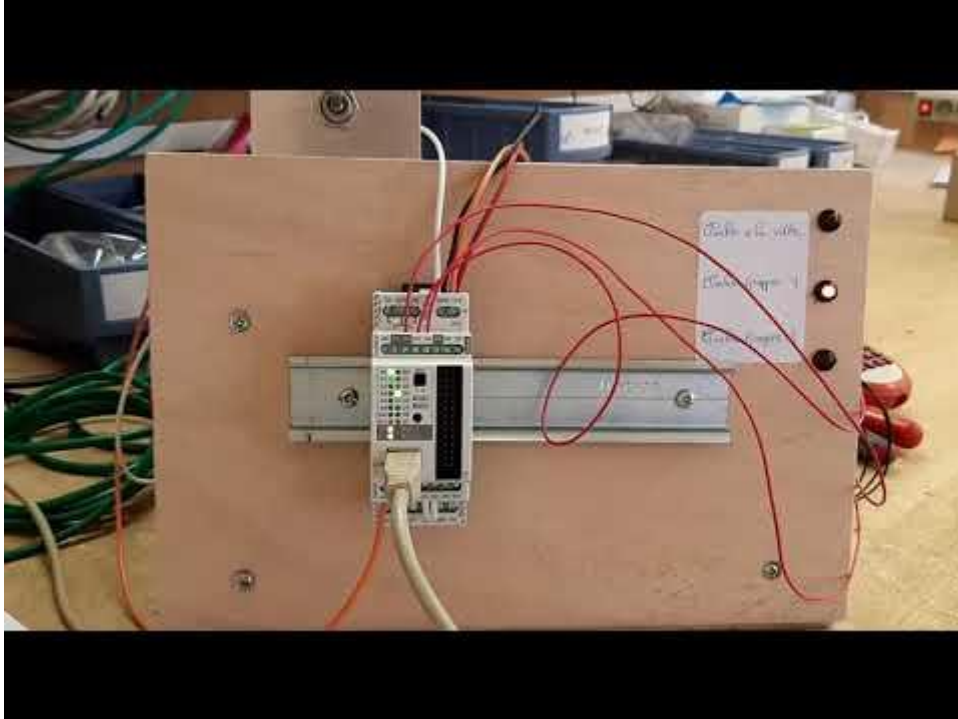
void setup() {
  // put your setup code here, to run once:
  pinMode(D2, OUTPUT);
  pinMode(D3, OUTPUT);
  pinMode(D4, OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:

  digitalWrite(D2, HIGH);
  digitalWrite(D3, HIGH);
  digitalWrite(D4, HIGH);
}
```


ii. Réalisation de la sonnerie du glas :

Après avoir réussi à allumer les leds qui représentent les cloches, il a fallu faire en sorte qu'elles ne soient pas illuminées constamment mais qu'elles suivent la sonnerie du glas.

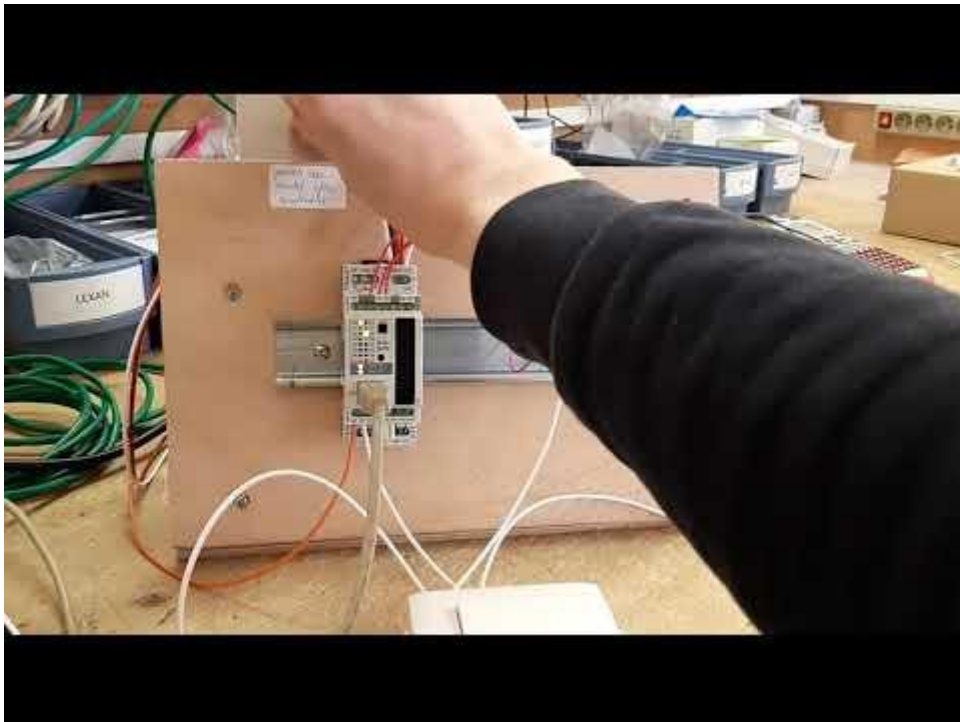


J'ai ajouté un interrupteur on-off qui permet de passer au mode glas car on ne souhaite pas que le glas sonne en continu. Donc lorsque l'on actionne l'interrupteur le glas commence jusqu'à ce qu'on repasse l'interrupteur sur off.

Ensuite je me suis occupé de faire en sorte que les leds qui représentent les cloches à tintement s'illuminent seulement lorsqu'elles reçoivent une impulsion. J'ai donc simulé l'impulsion à l'aide d'un interrupteur.

Donc lorsque le mode glas n'est pas activé et que l'on appuie sur l'interrupteur pour créer une impulsion, les leds qui représentent les cloches à tintement ne s'allument pas, tandis que lorsque le mode glas est activé, on prend en compte les impulsions lorsque j'appuie sur l'interrupteur et donc on fait clignoter la led lors que l'impulsion.

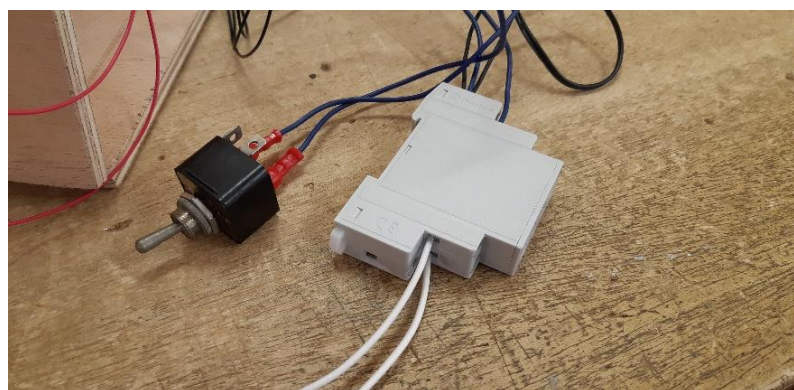
Ensuite il faut savoir que lorsque l'on démarre le mode glas la cloche à la volée doit se balancer pendant 30 secondes sans prendre en compte les impulsions car ces 30 secondes permettent à la cloche d'être à la bonne vitesse. Donc j'ai ajouté le fait que pendant 5 secondes (j'ai mis 5 sec au lieu de 30 sec pour les phases de test pour ce ne soit pas trop long) après le démarrage du mode glas, lorsque l'on appuie sur l'interrupteur pour créer les impulsions, celles-ci ne sont pas prises en compte pendant les 5 premières secondes.

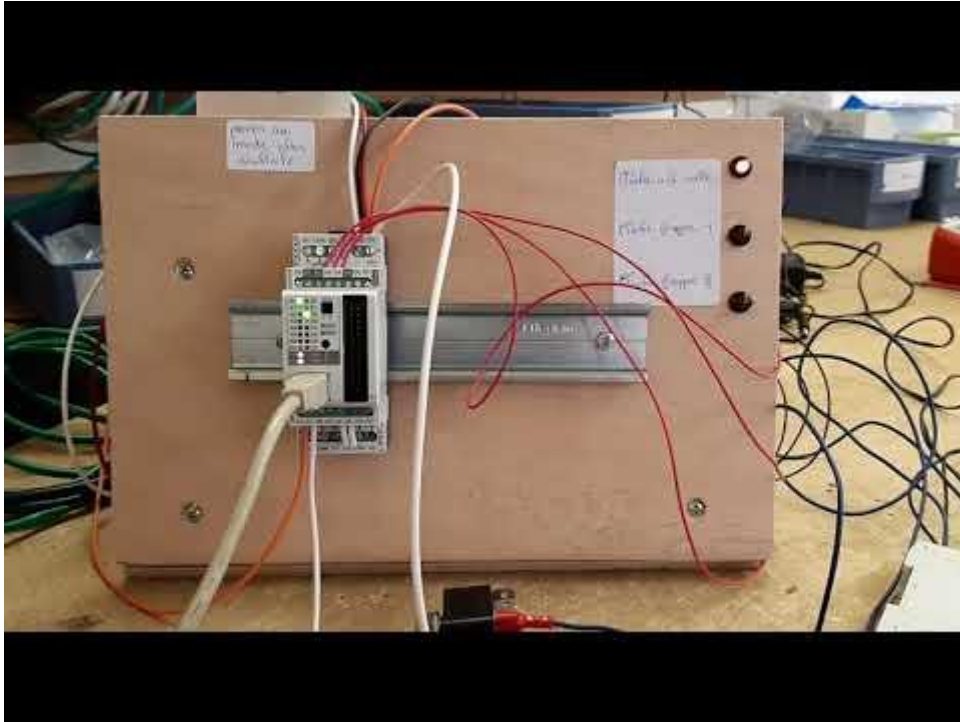


Et lorsque l'on désactive le mode glas on a toujours la led qui représente la cloche à la volée qui reste allumée 30 secondes (5 secondes pour les tests) le temps que celle-ci s'arrête complètement.

iii. Automatisation des impulsions :

Puis nous avons pu automatiser les impulsions à l'aide d'un relais temporisé modulaire réglé pour qu'il crée une impulsion toutes les 3 secondes. On y a relié un interrupteur qui permet d'arrêter les impulsions ou de les activer.

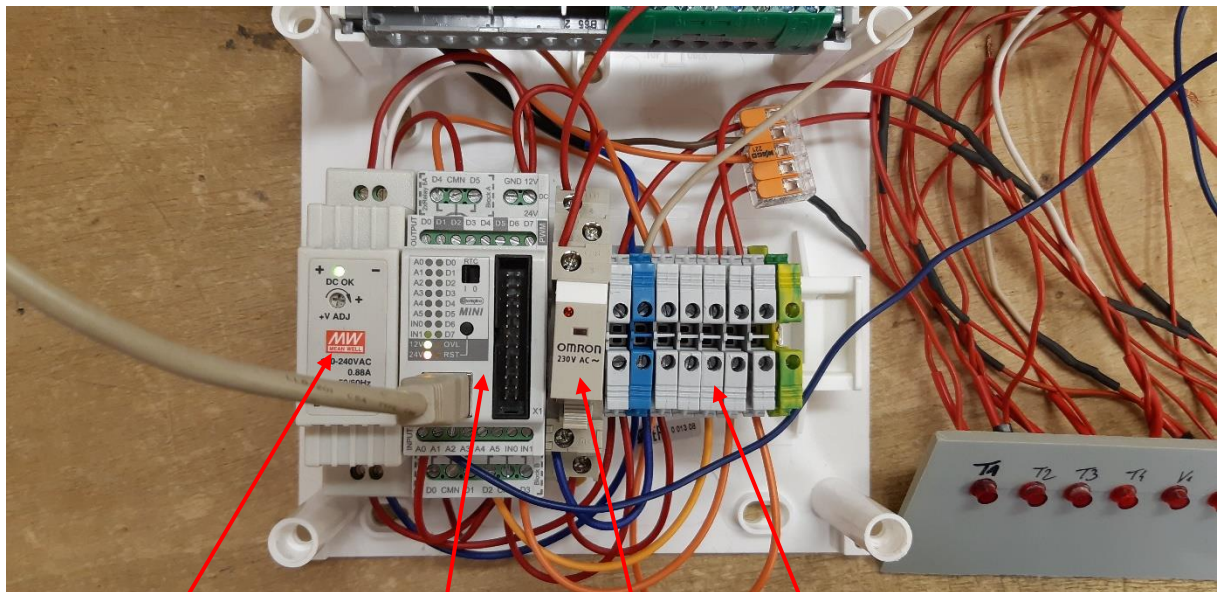




Ensuite il a fallu que je pense au fait que si on désactive le mode glas alors qu'on ne souhaité pas le désactivé et donc qu'on le réactive avant les 30 secondes qui permettent à la cloche à la volée de s'arrêter ou bien que quelqu'un décide de jouer avec l'interrupteur qui démarre le mode glas et que cette personne l'active et désactive en boucle. Pour éviter que cela crée des bugs, j'ai fait en sorte que dans tous les cas si le glas était actif et qu'on le désactive, on va forcément attendre 30 secondes le temps que la cloche à la volée s'arrête.

iv. Changement de certains composants et du câblage :

On a ensuite changé l'alimentation 24V par une alimentation 240V AC / 12V DC car les cloches ont besoin d'une tension de 240V pour fonctionner. On a donc tout décâbler pour tout recâbler proprement avec une alimentation 240V, un relais car certains composants ont besoin de 24V et non 240 et également des bornes de raccordement.



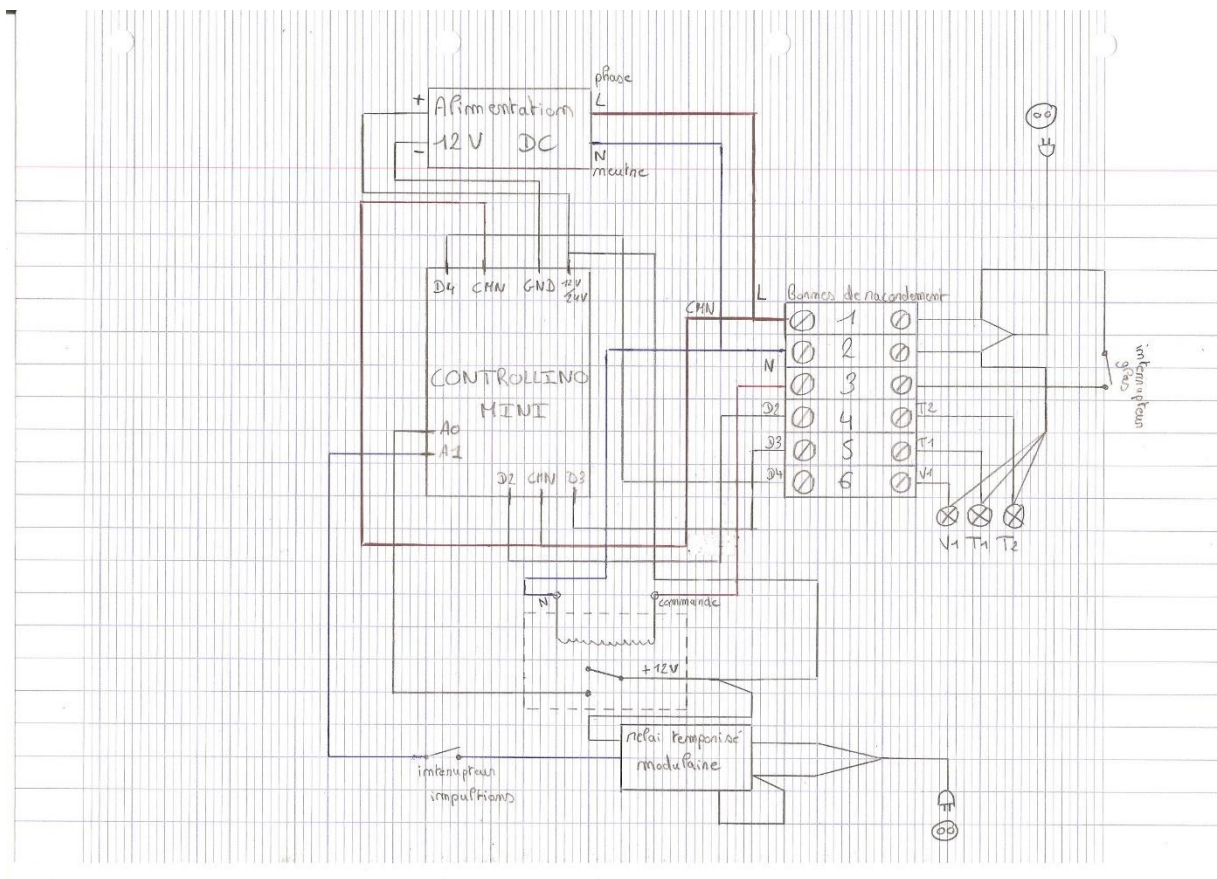
Alimentation 240V

Module Arduino

Relai

Bornes de raccordement

J'ai finalement fait le schéma électrique de l'installation, car il permet de la décrire précisément. Un schéma électrique permet de donner toutes les explications et données sur les équipements à installer et montre également les relations entre les éléments.



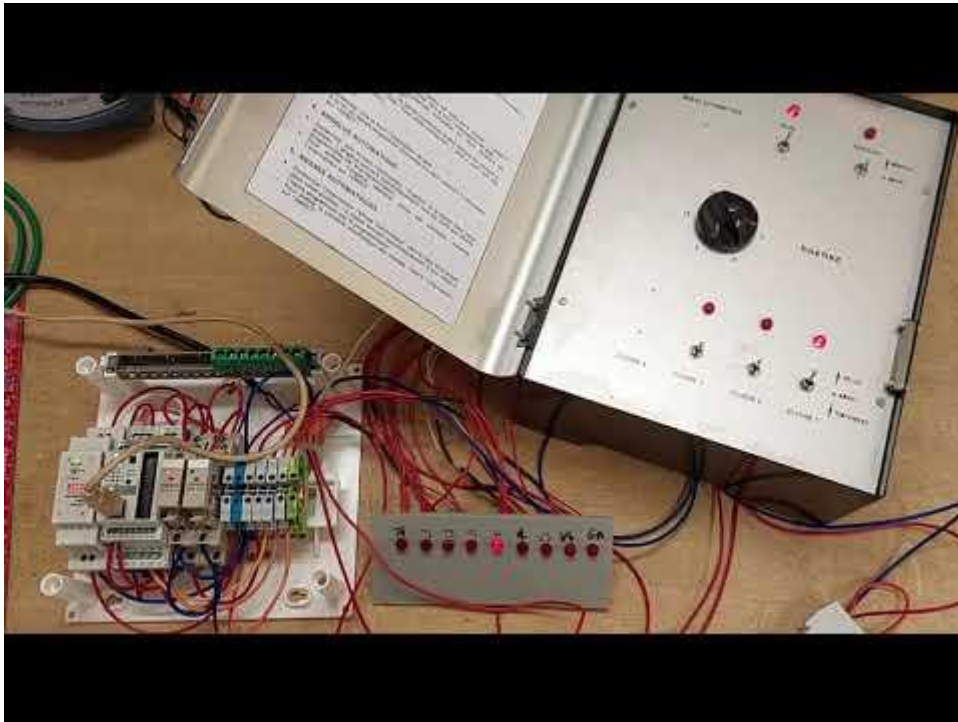
v. Câblage avec le véritable tableau de commandes :

La prochaine étape était de passer à un nouveau câblage afin d'utiliser le vrai tableau de commande utilisé dans les églises pour se rapprocher le plus possible de la réalité.

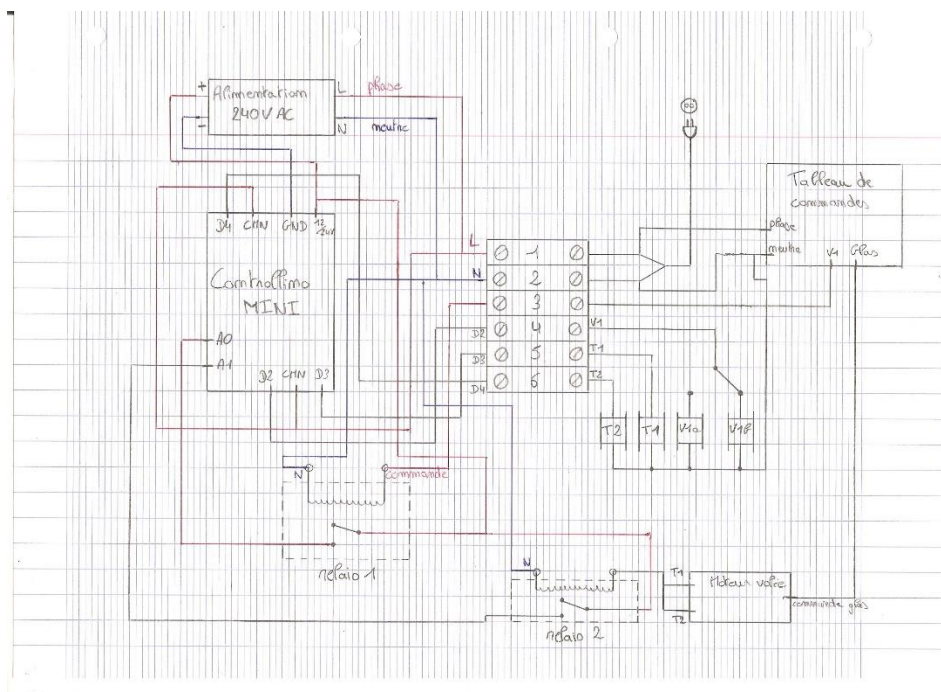


Ici le fonctionnement est légèrement différent. Pour activer le glas il faut tourner la molette noire au milieu du tableau de commandes sur le temps que l'on souhaite que le glas sonne. Ensuite il suffit d'activer l'interrupteur de la cloche 1 sur volée, cet interrupteur déclenche la cloche à la volée. Ensuite il faut activer l'interrupteur « Glas » sur marche, celui-ci sera le déclencheur des 30 secondes après lesquelles les cloches à tintement vont commencer à sonner afin d'être sûr que les 30 secondes soient bien respectées. La cloche à la volée va commencer par se balancer toute seule, et les deux cloches à tintement vont commencer à sonner 30 secondes après avoir déclenché la commande glas. Puis lorsque la minuterie est finie, le moteur de la volée se désactive et le tintement des cloches est arrêté. La cloche à la volée va s'arrêter progressivement du fait de son inertie.

J'ai donc fait le câblage final avec le tableau de commandes. Et avec quelques petites modifications dans le code, on a obtenu le résultat final qui est le suivant :



J'ai donc également réalisé le schéma électrique final, or j'ai eu quelques difficultés car au lieu de représenter le schéma du montage que j'avais à ce moment, il fallait que le fasse comme serait l'installation dans l'église, donc avec le moteur de la cloche à la volée, les contacteurs, etc... et non avec les leds.



c. Notice du fonctionnement et de l'installation :

J'ai réalisé la notice du projet dans laquelle est expliquée le fonctionnement du glas avec l'explication des différents interrupteurs etc.... Il y a également le câblage à réaliser en atelier pour l'installation des différents modules comme l'alimentation et les relais dans le coffret prévu à cet effet. Il explique également le câblage à réaliser sur site afin de relier au tableau de commandes et après toutes les étapes pouvoir faire fonctionner le glas (voir annexe page 23).

d. Présentation du code final :

Pour coder j'ai utilisé l'IDE Arduino. Le code est composé en trois parties : tout d'abord on déclare les variables (cadre rouge), pour déclarer les entrées et sorties (cadre bleu) puis on a la boucle qui se répète indéfiniment et c'est là que l'on entre notre code (cadre vert).

```
#include <Controllino.h>

#define cloche_volee CONTROLLINO_D2 //déclare cloche à la volée
#define cloche_tintement_1 CONTROLLINO_D3 //déclare cloche à tintement 1
#define cloche_tintement_2 CONTROLLINO_D4 //déclare cloche à tintement 2
const int V1 = A0; //déclare l'interrupteur de la volée en entrée A0
const int glas = A1; //déclare l'interrupteur du mode glas en entrée A1
int nbr_impulsion = 0; //déclare la variable qui permet de compter le nombre d'impulsions
int variable_30sec_debut = 0; //déclare la variable des 30sec au déclenchement du mode glas
```

```
void setup() {
  // put your setup code here, to run once:

  pinMode(cloche_volee, OUTPUT); //déclare la cloche à la volée en sortie
  pinMode(cloche_tintement_1, OUTPUT); //déclare la cloche à tintement 1 en sortie
  pinMode(cloche_tintement_2, OUTPUT); //déclare la cloche à tintement 2 en sortie
  pinMode(V1, INPUT); //déclare l'interrupteur de la volée en entrée
  pinMode(glas, INPUT); //déclare l'interrupteur du mode glas en entrée
  Serial.begin(9600); //défini le nombre de symboles transmissibles par seconde
}
```

```
void loop() {
  // put your main code here, to run repeatedly:

  if (digitalRead(V1) == 0){ //Si l'interrupteur de la volée = 0 : on ne fait rien et reme
    digitalWrite(cloche_volee, LOW);
    digitalWrite(cloche_tintement_1, LOW);
    digitalWrite(cloche_tintement_2, LOW);
    variable_30sec_debut = 0;
    nbr_impulsion = 0;}

  else{ //activation de la volée
```

Ce que le code fait :

Si l'interrupteur de la cloche à la volée est sur OFF, rien ne se passe.

```
void loop() {  
  // put your main code here, to run repeate  
  
  if (digitalRead(V1) == 0){ //Si l'interr  
    digitalWrite(cloche_volee, LOW);  
    digitalWrite(cloche_tintement_1, LOW);  
    digitalWrite(cloche_tintement_2, LOW);  
    variable_30sec_debut = 0;  
    nbr_impulsion = 0;}  
}
```

Si celui-ci passe sur ON, on fait sonner la cloche à la volée.

Si l'interrupteur du mode Glas passe sur ON, et si on a pas encore attendu les 30 secondes nécessaires à ce que la cloche à la volée atteigne la bonne vitesse, alors on attend 30 secondes avant de prendre en compte les impulsions.

```
else{ //activation de la volée  
  digitalWrite(cloche_volee, HIGH); //fait fonctionne  
  if(digitalRead(glas) == 1){ // si l'interrupteur du  
    if(variable_30sec_debut < 1){ //tant que 30 sec son  
      delay(30000); //attend 30 sec le temps que la clo  
      variable_30sec_debut = variable_30sec_debut + 1;}  
    }
```

Ici on fait sonner 2 fois d'affiler la cloche à tintement numéro 1.

```
while(nbr_impulsion < 2){ //permet de faire sonn  
  if (digitalRead(glas) == 1){ //si l'interrupteur  
    digitalWrite(cloche_tintement_1, HIGH); // all  
    delay(500);  
    digitalWrite(cloche_tintement_1, LOW); // étein  
    delay(500); //delai de 0,5 sec car impulsion p  
    nbr_impulsion = nbr_impulsion + 1; //incrément  
  }  
}
```

Puis on fait sonner 1 fois la cloche à tintement numéro 2

```
.  
  
if(digitalRead(glas) == 1 && nbr_impulsion > 1){ //1  
  digitalWrite(cloche_tintement_2, HIGH);  
  delay(500);  
  digitalWrite(cloche_tintement_2, LOW);  
  delay(500);  
  nbr_impulsion = 0; //remet le nombre d'impulsions  
}
```

Pour finir si on rebasculer l'interrupteur de la volée, on la fait s'arrêter.

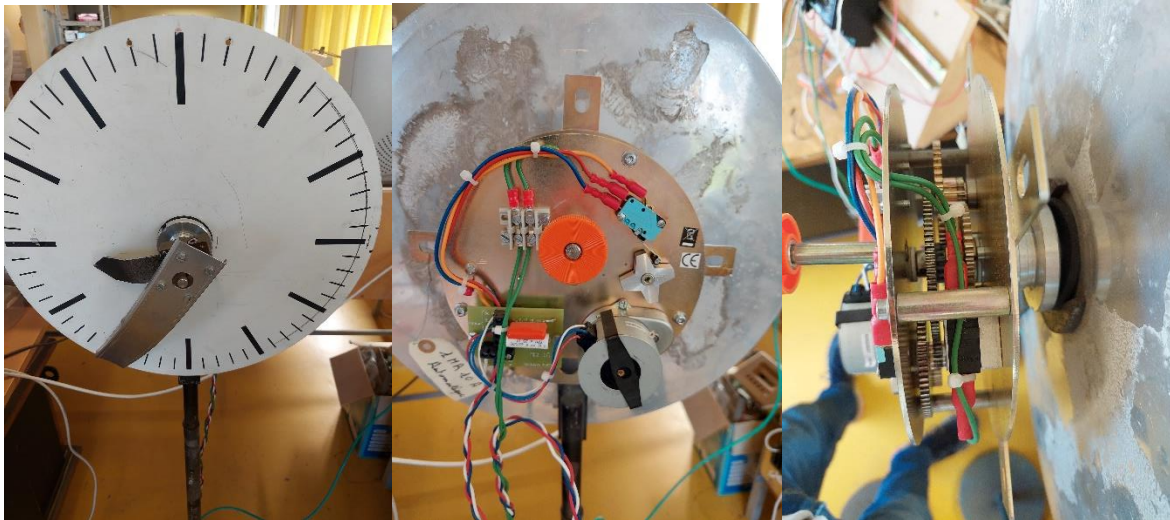
```
}  
if(digitalRead(V1) == 0){ //si désenclench  
  digitalWrite(cloche_volee, LOW);  
  variable_30sec_debut = 0;  
  nbr_impulsion = 0;}  
}  
}
```


III. Horloge Réglice

a. Fonctionnement de l'horloge :

La deuxième partie de mon stage fut dédiée à l'étude de composants de l'horloge Réglice.

L'horloge Réglice est une horloge déjà existante connectée au logiciel Réglice qui est un des deux logiciels de gestion de temps de l'entreprise.



Cette horloge est le projet d'un ancien stagiaire, le code avait été fait en Arduino sur un module Controllino. Elle fonctionne avec un capteur mécanique qui permet de détecter les minutes paires et impaires. Il y a également un autre capteur qui détecte la position à 12 h 00 car à cette heure l'horloge vérifie si elle n'est pas en retard ou en avance auquel cas elle attendrait qu'il soit réellement 12 h 00 ou rattraperait son retard.

b. Missions à réaliser :

J'ai été mis sur ce projet avec Florian un autre stagiaire qui était en stage en même temps que moi. Florian est en 3^{ème} année de licence informatique.

Notre objectif était de trouver d'autres capteurs pouvant durer plus longtemps qu'un capteur mécanique, changer le motoréducteur qui est trop peu puissant et changer de carte pour une carte plus adaptée et moins coûteuse.

i. Recherche du motoréducteur :

Nous avons donc commencé par chercher le motoréducteur. Les caractéristiques demandées était un motoréducteur brushless car ils fonctionnent plus longtemps que les autres motoréducteurs comme ceux à balai. La vitesse de rotation devait être entre 10 et 100 tr/min, une tension d'alimentation de 24V, une taille pas trop importante et pour un prix inférieur à 100€.

Le problème rencontré était que pour un moteur brushless il fallait compter au minimum 140€. Les seules solutions étaient de prendre un motoréducteur à balai afin de respecter le budget ou d'augmenter le budget aux alentours de 150€. Nous avons donc exposé le problème et la décision a été d'augmenter le budget pour prendre un motoréducteur brushless car ils durent beaucoup mieux dans le temps.

Le motoréducteur choisis a donc été le suivant. Il respecte toutes les caractéristiques demandées, c'est un motoréducteur brushless avec une vitesse de sortie maximale de 80 tr/min que l'on peut câbler en 24V, tout en étant pas trop volumineux (63.5 X 92 X 56 mm) et pour un prix de 150€.



ii. Recherche d'une carte :

Ensuite nous avons cherché la carte qui serait la plus adaptée au projet. La carte précédente était une Controllino maxi, c'est une carte qui coûte plus de 200€ tandis que le budget pour la future carte était de 50€.

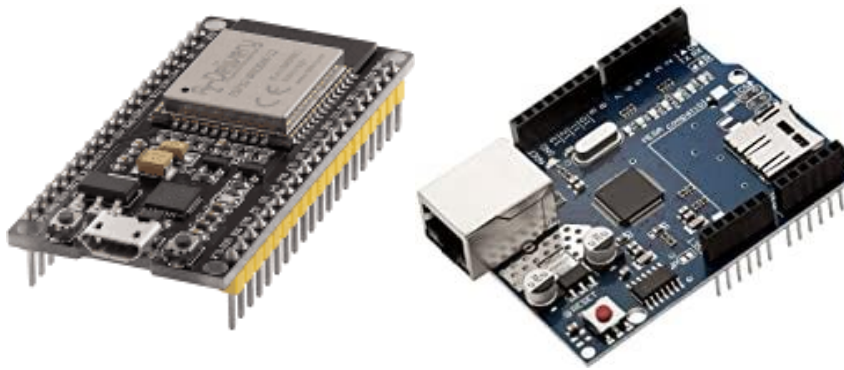
Il fallait prendre en compte la connectivité, les capacités et le nombre de ports.

Deux choix s'offraient à nous. Il y avait les cartes Arduino classiques comme la UNO ou la MEGA ou bien une carte ESP32. Aucune de ces cartes ne possède de port Ethernet pour se connecter au réseau donc il va falloir ajouter un Shield Ethernet. En revanche la ESP32 a la possibilité de se connecter en wifi et en Bluetooth, elle possède un id unique contrairement à une carte Arduino classique et elle fonctionne entre -40° et +125°. De plus elle possède 38 entrées/sorties ce qui suffit pour ce projet.

Il est possible de trouver une carte ESP32 aux alentours de 8 euros. Et pour le shield Ethernet à ajouter, on en trouve pour 10€.

Carte ESP32

Shield Ethernet



iii. Choix des capteurs :

Le dernier composant à choisir fut les capteurs pour savoir si on est sur une minute paire ou impaire et pour savoir quand on est à 12 H 00. Le capteur installé sur l'horloge était un capteur mécanique à levier or ce genre de capteur a tendance à s'abîmer beaucoup plus vite que des capteurs qui fonctionnent sans contact mécanique qui eux sont presque inusables.

Capteur précédent :



Après une recherche de capteurs de position, les deux capteurs retenus ont été le capteur à effet Hall et le capteur optique. Le capteur à effet Hall réagit à la présence d'un champ magnétique et le capteur optique réagit lors du passage d'un élément devant son faisceau lumineux.

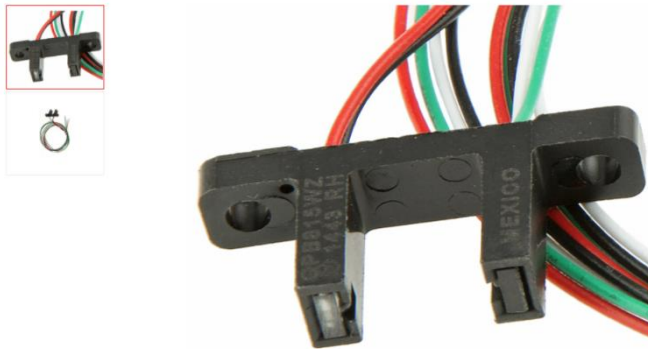
1. Capteur optique :

Un capteur optique fonctionne avec une lumière infrarouge émise avec une séquence d'impulsions extrêmement courte et rapide, ce qui permet une détection fiable des mouvements rapides, or s'il y a une luminosité extérieure trop importante son efficacité peut être altérée. Donc il faut que l'endroit où le capteur se trouve soit protégé des lumières extérieures.

Ici un disque comme sur l'image suivante passerait devant la lumière infrarouge et donc en fonction de si la lumière est captée ou non, on sait si on est sur une minute paire ou impaire. Et le système serait identique pour savoir quand les aiguilles sont sur 12 H 00.



Le capteur optique choisi est le suivant car il y a déjà les câbles permettant de le faire fonctionner et la distance entre l'émetteur et le récepteur est d'environ 9.5mm ce qui permet au disque de passer entre deux. Son prix est de 4€ unité et il fonctionne entre -40° et +80°.

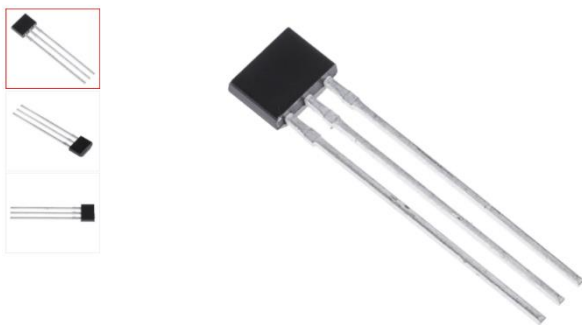


2. Capteur à effet Hall :

Le capteur à effet hall est composé de trois broches, deux pour l'alimentation et une pour le signal de sortie. Pour les deux broches que l'on doit relier à l'alimentation, on en relie une au pôle négatif et une au pôle positif. La troisième broche émet un signal lorsqu'elle rencontre un champ magnétique.

Certains capteurs sont unipolaires et ne peuvent détecter qu'un seul pôle de l'aimant (le nord ou le sud) alors que d'autres capteurs sont bipolaires et détectent aussi bien le pôle Nord que le pôle Sud de l'aimant. Ce genre de capteur est très précis et n'est pas très grand.

Le capteur retenu est le suivant car il n'est pas facile de trouver un capteur à effet hall avec le câblage déjà réalisé qui convient au projet. Celui-ci coûte 2€ et fonctionne entre -40° et +150°.



Il n'y a pas eu de choix entre le capteur optique et à effet hall car vu les prix très faibles les deux solutions pourront être testées afin de déterminer la meilleure solution.

IV. Réalisation de tests :

a. Présentation du Fonctionnement

J'ai pu réaliser des tests pour un des développeurs qui s'occupe de la réalisation du calcul des heures de travaux et heures supplémentaires sur le logiciel Réglice.

Le calcul se fait avec les pointages réalisés, ce qui génère le temps travaillé chaque jour, semaine et mois.

Avant de lancer le calcul il faut paramétrer le compteur et l'écrtage. Concernant le compteur, on choisit sur quoi se base le calcul : soit en fonction du nombre d'heures totales, soit en fonction des heures du jour, on choisit également si on fait le calcul à la journée, à la semaine ou au mois. Et finalement on choisit la majoration des heures supplémentaires.

Ensuite l'écrtage sert à définir le temps minimum d'heures supplémentaires pour qu'elles soient comptabilisées, également le nombre d'heures supplémentaires maximum. Il est également possible de choisir si on prend uniquement les heures supplémentaires positives ou aussi en négatif.

b. Exemple d'un calcul :

Dans l'exemple qui suit, le mode de calcul est en fonction des heures totales faites par journée. Le seuil de début est de 6 h 00, et le seuil de fin est de 8 h 00 de travail. Et avec une majoration de 100%.

Gestion des compteurs

Gestion des Compteurs

Vos compteurs : HS1 à la journée + test

Information du compteur

Désignation du compteur :
HS1 à la journée

Désignation entête édition:
HS1

Mode de comptage selon :
Base d'un total

Seuil de Début :
06.00.00

Seuil de Fin :
08.00.00

Sélectionner un type de total pour ce compteur :
Jour

Majoration (en %):
100

REPORT

OPTION

Visible : ON

Partage : OFF

Modifier

Annuler

L'écrtage se fait à la semaine avec un maximum de 6 h 00 d'heures supplémentaires par semaine.

Gestion des écrêtages

Gestion des Écrêtages

Vos compteurs : Écrêtage : HS1 à la journée + test

Paramètre de l'écrêtage

Désignation de l'écrêtage :
Écrêtage : HS1 à la journée - Total semaine

Total à écrêter :
HS1 à la journée - Total semaine

ECRETAGES

OPTION

Visible : ON

Partage : OFF

Modifier

Annuler

Ensuite on lance le calcul. On obtient donc les logs avec les pointages à la journée ce qui donne le calcul des heures travaillées à la journée. On a également le calcul semaine et le calcul mois (période de paie).

Puis nous avons le calcul des heures supplémentaires. Comme on peut le voir, les heures supplémentaires ne dépassent pas 2 h 00 dans la même journée car le seuil de début était 6 h 00 et le seuil de fin 8 h 00. Les heures supplémentaires à la semaine sont donc calculées avec un écrêtage de 6 h 00 maximum, et finalement les heures supplémentaires sur toute la période de paie.

| |
|-----------------------------------------------------------------------------------|
| Pointage (12/03/2020) : |
| Pointage 1 : 12/03/2020 - 16:57:53 |
| Pointage 2 : 12/03/2020 - 20:07:49 |
| Pointage 3 : 12/03/2020 - 21:00:00 |
| Pointage 4 : 13/03/2020 - 01:01:37 |
| Calcul (Jour) : 12/03/2020 => 07:11:33 |
| Pointage (13/03/2020) : |
| Pointage 1 : 13/03/2020 - 17:01:58 |
| Pointage 2 : 13/03/2020 - 21:00:36 |
| Pointage 3 : 13/03/2020 - 21:49:01 |
| Pointage 4 : 14/03/2020 - 01:01:38 |
| Calcul (Jour) : 13/03/2020 => 07:11:15 |
| Calcul (Semaine) : 01/03/2020 - 01/03/2020 => 07:29:42 |
| Calcul (Semaine) : 02/03/2020 - 08/03/2020 => 38:48:38 |
| Calcul (Semaine) : 09/03/2020 - 15/03/2020 => 29:46:28 |
| Calcul (Période de paie) : 01/03/2020 - 31/03/2020 => 76:04:48 |
| Calcul (Jour) [HS1 à la journée] : 01/03/2020 => 01:29:42 |
| Calcul (Jour) [HS1 à la journée] : 02/03/2020 => 01:50:13 |
| Calcul (Jour) [HS1 à la journée] : 03/03/2020 => 02:00:00 |
| Calcul (Jour) [HS1 à la journée] : 04/03/2020 => 01:24:12 |
| Calcul (Jour) [HS1 à la journée] : 07/03/2020 => 01:38:48 |
| Calcul (Jour) [HS1 à la journée] : 08/03/2020 => 01:30:12 |
| Calcul (Jour) [HS1 à la journée] : 09/03/2020 => 01:23:11 |
| Calcul (Jour) [HS1 à la journée] : 10/03/2020 => 02:00:00 |
| Calcul (Jour) [HS1 à la journée] : 12/03/2020 => 01:11:33 |
| Calcul (Jour) [HS1 à la journée] : 13/03/2020 => 01:11:15 |
| Calcul (Semaine) [HS1 à la journée] : 01/03/2020 - 01/03/2020 => 01:29:42 |
| Calcul (Semaine) [HS1 à la journée] : 02/03/2020 - 08/03/2020 => 06:00:00 |
| Calcul (Semaine) [HS1 à la journée] : 09/03/2020 - 15/03/2020 => 05:45:59 |
| Calcul (Période de paie) [HS1 à la journée] : 01/03/2020 - 31/03/2020 => 13:15:41 |

c. Tests réalisés.

Après qu'on m'ait expliqué le fonctionnement des calculs, le paramétrage du compteur et de l'écrêtage, j'ai réalisé des tests en essayant de réaliser toutes les possibilités. Durant les tests j'ai donc essayé de mettre des valeurs incohérentes dans le paramétrage, j'ai regardé si tous les paramètres étaient bien pris en compte dans les logs comme l'écrêtage, etc.... J'ai ensuite donné les erreurs que j'avais trouvées qui étaient du type : l'écrêtage pas pris en compte, le seuil de début qui pouvait être plus grand que le seuil de fin et le fait qu'il n'y avait pas de message d'erreur lorsque l'on rentrait des valeurs incohérentes dans certains champs.

v. Configuration de badgeuses :

L'entreprise achète des badgeuses qu'il faut ensuite configurer. Je me suis donc occupé de configurer quelques badgeuses pour qu'elles soient prêtes au moment où il y a des commandes de badgeuses.

La configuration consiste à coller les étiquettes Huchez, configurer l'adresse IP, mettre la badgeuse en français, modifier le format de la date, personnaliser et désactiver certains raccourcis etc...


J'ai également réalisé une procédure de la configuration pour que d'éventuelles personnes ayant besoin de configurer ces badgeuses ultérieurement puissent le faire simplement. (Voir annexe page 24).




VI. Annexes


Annexe 1 – Notice d'installation et de fonctionnement du glas

Glas « boíteux » Saint-Remy-en-l'Eau





00420 Ferrières - France
Tel : 03 44 51 11 22 - Fax : 03 44 51 22 00
www.horloges-huchez.fr www.saintremy-l'eau.com

Version 2 du 31/05/2021
Glas de Saint-Remy-en-l'Eau
Notice d'installation et d'utilisation


1. Introduction

Félicitations !
Vous venez de recevoir un module Arduino Controllino Mini, une alimentation 240V AC / 12V DC, 2 relais et 7 bornes de raccordement.

Afin de bénéficier d'un fonctionnement correct et durable, nous vous recommandons de bien suivre les instructions de cette notice.

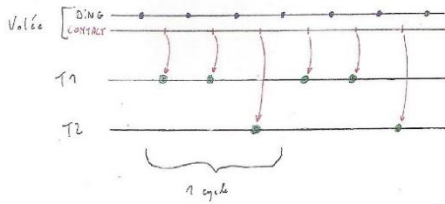
1.1 Principe du glas « boíteux »

On part du principe du glas romain :

- 1 x cloche en volée, équipée d'un combinatoire de glas, qui déclenche un contact sec chaque fois que la cloche passe à son point bas (= position de repos).
- Ce contact sec déclenche un marteau linteur, qui va donc battre au même rythme que la cloche en volée, entre deux coups de cloche en volée.

La différence pour ce glas boíteux est que :

- on aura 2 cloches en linteur (au lieu d'une seule)
- on frappe 2 coups sur la première, puis un coup sur la deuxième
- et ainsi de suite...



SOMMAIRE

| | | |
|-------|-------------------------------------------------|----|
| 1. | INTRODUCTION | 3 |
| 1.1 | PRINCIPE DU GLAS « BOITEUX » | 3 |
| 2. | FONCTIONNEMENT | 4 |
| 3. | MONTAGE / CABLAGE | 5 |
| 3.1 | CABLAGE EN ATELIER | 5 |
| 3.1.1 | Alimentation du module Arduino | 6 |
| 3.1.2 | Alimentation des relais | 7 |
| 3.1.3 | Cablage de l'alimentation | 8 |
| 3.1.4 | Cablage de l'interrupteur du mode glas | 9 |
| 3.1.5 | Cablage du Controllino aux cloches | 10 |
| 3.2 | MONTAGE SUR SITE | 11 |
| 3.2.1 | Cablage au tableau de commandes | 11 |
| 3.2.2 | Cablage des contacteurs et des cloches sur site | 12 |
| 3.3 | Essais & mise en service | 13 |
| 4. | Notes | 14 |

HISTORIQUE DU DOCUMENT

| Ed | Date de maj | Modifications |
|----|-------------|-----------------------------------|
| 01 | 18/05/2021 | Première version par PAUL BANNIER |
| 02 | 31/05/2021 | Correction par François HUCHEZ |
| | | |
| | | |

3/16

Annexe 2 – Procédure des badgeuses

Procédure de mise en fonction des badgeuses LB50 et LB150

Respecter la procédure suivante afin de configurer correctement la badgeuse.

Initialisation :

Étape 1 :
Coller l'étiquette au dos de la badgeuse.
La référence de la badgeuse avec LB50-CD-ETH si le modèle est une LB50, ou LB150-CD-ETH si il s'agit d'une LB150.

Étape 2 :
Coller le bouton sur le devant de la badgeuse en respectant la photo ci-dessous. Le flash doit rester visible.

Étape 3 :
Brancher le câble d'alimentation et le câble ethernet.

Étape 4 :
Allumer la badgeuse en appuyant sur le bouton situé en bas à gauche.
L'écran de démarrage devrait apparaître. Appuyez ensuite sur le bouton pour accéder à l'écran d'accueil. Cette touche permet de valider l'action en cours.

Déplacements :

Utiliser les touches encadrées en rouge ci-dessous afin de se déplacer sur l'écran.

Paramétrage du module :

Étape 1 :
Se déplacer sur l'écran "Personnalisation" Appuyer sur "User Interface" et choisir la langue "Français".

Étape 2 :
Appuyer sur l'écran "Paramétrage" puis sur "ENTRETIEN" et choisir "Config".

Étape 3 :
Appuyer sur l'écran "Système" puis sur "Date/Heure".

Étape 4 :
Appuyer sur "Config" puis sur "Date/Heure".
Puis faire les paramètres suivants :
- Mode du jour : 5
- Date : 01/01/2017
- Heure : 01:00
- Fz : 01:00
- Fz : 01:00
- Fz : 01:00
- Fz : 01:00
- Fz : 01:00
- Fz : 01:00

Étape 5 :
Appuyer sur "Paramétrage" puis sur "Entretien".

Étape 6 :
Appuyer sur l'écran "Personnalisation" puis sur "User Interface" et choisir la langue "Français".

Étape 7 :
Appuyer sur l'écran "Paramétrage" puis sur "ENTRETIEN" et choisir "Config".

Quand la configuration est terminée, appuyer sur le bouton "ENTRETIEN" pour valider.

Configuration avec l'écran :

- Appuyer sur l'écran "Paramétrage" puis sur "ENTRETIEN" et choisir "Config".
- Appuyer sur l'écran "Système" puis sur "Date/Heure".
- Appuyer sur "Config" puis sur "Date/Heure".
- Appuyer sur "Config" puis sur "Date/Heure".
- Appuyer sur "Config" puis sur "Date/Heure".
- Appuyer sur "Config" puis sur "Date/Heure".
- Appuyer sur "Config" puis sur "Date/Heure".
- Appuyer sur "Config" puis sur "Date/Heure".
- Appuyer sur "Config" puis sur "Date/Heure".
- Appuyer sur "Config" puis sur "Date/Heure".

Félicitations vous avez configuré la badgeuse.