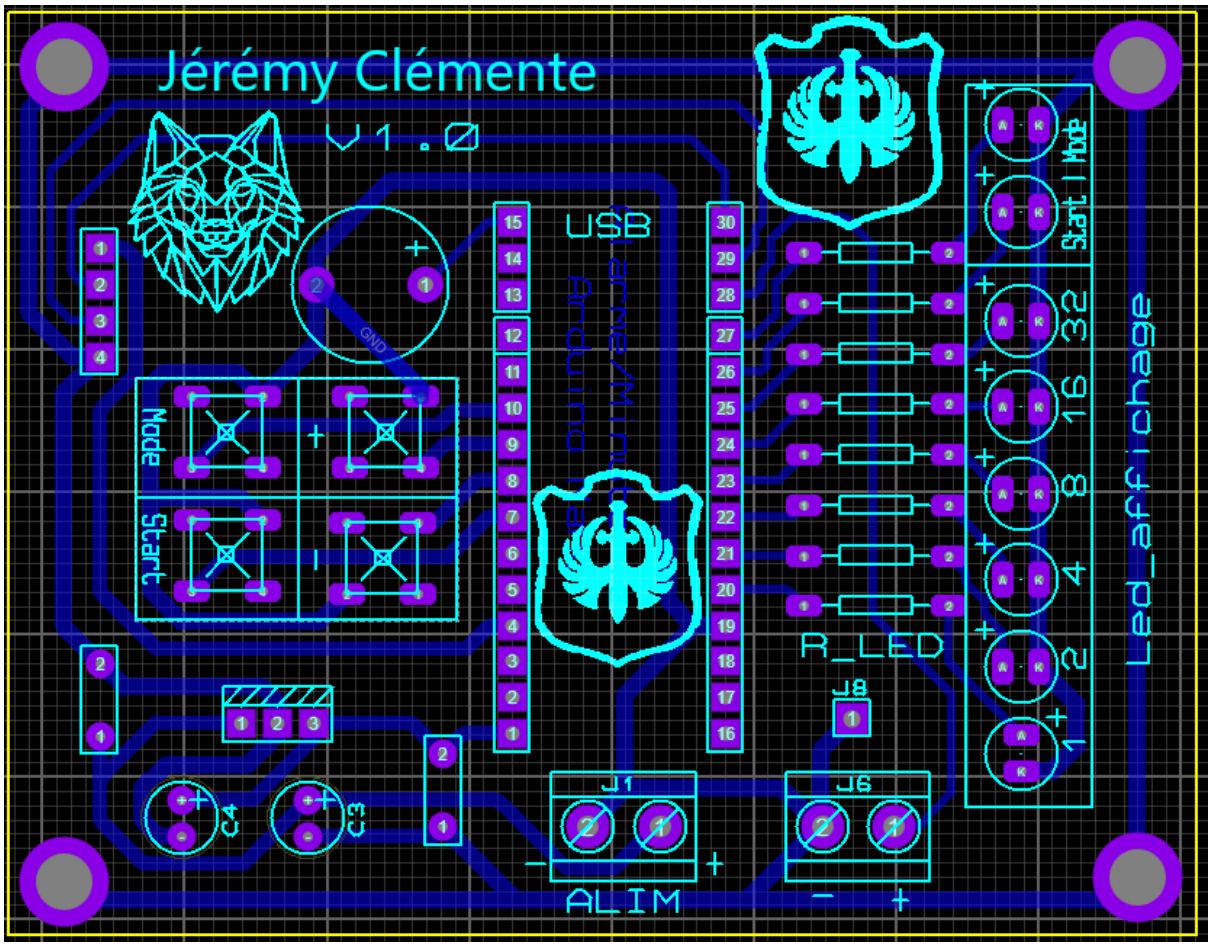
V1.0

PROJET : ALARME ET DÉTECTEUR DE MOUVEMENT ARDUINO



SOMMAIRE

Présentation Projet 3
Ce projet possède deux modes de fonctionnement : 3
Composants : 3
Carte arduino : 3
Alimentation Externe : 3
Sortie signal externe : 4
Les différentes Leds : 4
Led Bleu : 4
Led Orange : 4
Led Verte : 4
Buzzer : 4
Boutons arduino : 4
Mode alarme (Led orange éteinte) : 5
Mode Détecteur de Présence (Led orange allumée) : 5
Protheus (Isis) : 5
Protheus (Ares) : 6
IDE Arduino (code C++) : 7
Led et Buzzer : 7
Le Timer : 9
Le Buzzer : 9
Sortie externe : 9
Bouton : 9
Exemple avec le bouton start : 10
Capteur à ultrason : 10
épilogue 11
Défaut : 11
Liste d'amélioration possible : 12
Remerciements : 12
Fin du rapport : 12

PRÉSENTATION PROJET

Ce projet possède deux modes de fonctionnement :

- Minuterie ou alarme portable avec activation d'un buzzer à la fin du temps.
- Détecteur de mouvement qui active l'alarme.

Le système est portable et avec une alimention externe de 9V comme une pile par exemple. Le microcontrôleur utilisé est une Arduino nano. Il y a une partie commande avec un affichage Led et une autre partie avec des boutons pour sélectionner le mode, la durée et le départ. La carte électronique contient aussi une sortie qui est actionnée en même temps que le buzzer pour par exemple, activer des lumières ou d'autres appareils.

Composants :

- 1 Arduino nano (support)
- 2 condensateurs 100nF (non polarisé)
- *2 condensateurs 100uF (polarisé)*
- *1 7805 (convertisseur 9V en 5V)*
- *2 Camden-2*
- *1 module ultrason (support)*
- *6 Leds bleu*
- *1 Led vert*
- *1 Led orange*
- *1 Buzzer*
- *8 Resistance (200Ω-2kΩ)*
- *4 boutons arduino*

Carte arduino :

La carte arduino choisie est une Arduino Nano pour sa petite taille, ses pattes mâles qui permettent de la clipser dans un support, son coût relativement bas, sa fiabilité et ma maitrise des cartes arduino. Elle pilote tout le circuit,gère l'allumage des leds, le timer, le buzzer, ect. Elle peut être alimentée de deux façons, soit via le port usb (mode fixe),soit via une alimentation externe comme une pile de 9V (mode mobile).Elle est codée en C++ via l'ide d'arduino.

Alimentation Externe :

On peut alimenter la carte électronique via une pile de 9V en la branchant sur le CAMDEN-2 qui est devant la carte arduino, la conversion 9V en 5V se fait via l'utilisation du module 7805 et on utilise aussi deux condensateurs non polarisés et deux condensateurs polarisés pour supprimer les éventuels parasites et

harmoniques ce qui permet de garder une alimentation la plus proche possible de 5 V.

Sortie signal externe :

Le deuxième CAMDEN-2 est activé en même temps que le buzzer, c'est-à-dire lorsque le compte à rebours est à zéro (en mode alarme) ou que le capteur à ultrason détecte une présence (en mode détection de mouvement) ce signal est piloté par la carte arduino nano et peut permettre divers usages comme piloter un circuit externe, allumer des lumières, etc.

Les différentes Leds :

Il y a 3 couleurs de led utilisées sur la carte, les leds bleus, oranges et vertes.

Led Bleu :

Elles servent à témoigner de la durée restante lors du mode alarme. Elles sont codées en binaire ;allumé signifie 1 et en binaire éteint signifie 0.Le chiffre en dessous montre le temps restant, il est cumulatif, c'est-à-dire que si les leds avec les chiffres 1,2,16 sont allumés, il y a19 minutes de temps restant. L'affichage des leds change à chaque minute lors du départ de l'alarme.

Led Orange :

La led orange sert à déterminer dans quel mode est le circuit, si elle est allumée le mode est sur la détection de mouvement, sinon si elle est éteintele mode est alarme/minuterie.

Led Verte :

Elles servent uniquement en mode alarme/minuterie, si elles sont activées, cela veut dire que l'alarme est enclenchée, elles servent de témoin une fois l'alarme allumée. Le seul moyen de la désactivée est d'appuyer sur le bouton réset sur la carte arduino nano.

Les différentes leds sont connectées en série avec une résistance qui peut aller de 200Ω-2kΩ pour éviter qu'elles ne grillent.

Buzzer :

Le buzzer est alimenté via la carte arduino avec un signal oscillant du +5V au 0V pour le faire « claquer », la durée entre le 5V et le 0V est de l'ordre du milliseconde.

Boutons arduino :

La partie opérative est dotée de 4 boutons arduino, 2 servent à augmenter ou diminuer le temps en mode alarme,1 sert au départ et le dernier sert à choisir le mode alarme ou détection de présence. Il faut faire attention lors du placement des boutons arduino car certaines broches sont connectées entre elles et donc il faut éviter une liaison directe qui serait fâcheuse,le système ne pourrait pas fonctionner.

Mode alarme (Led orange éteinte) :

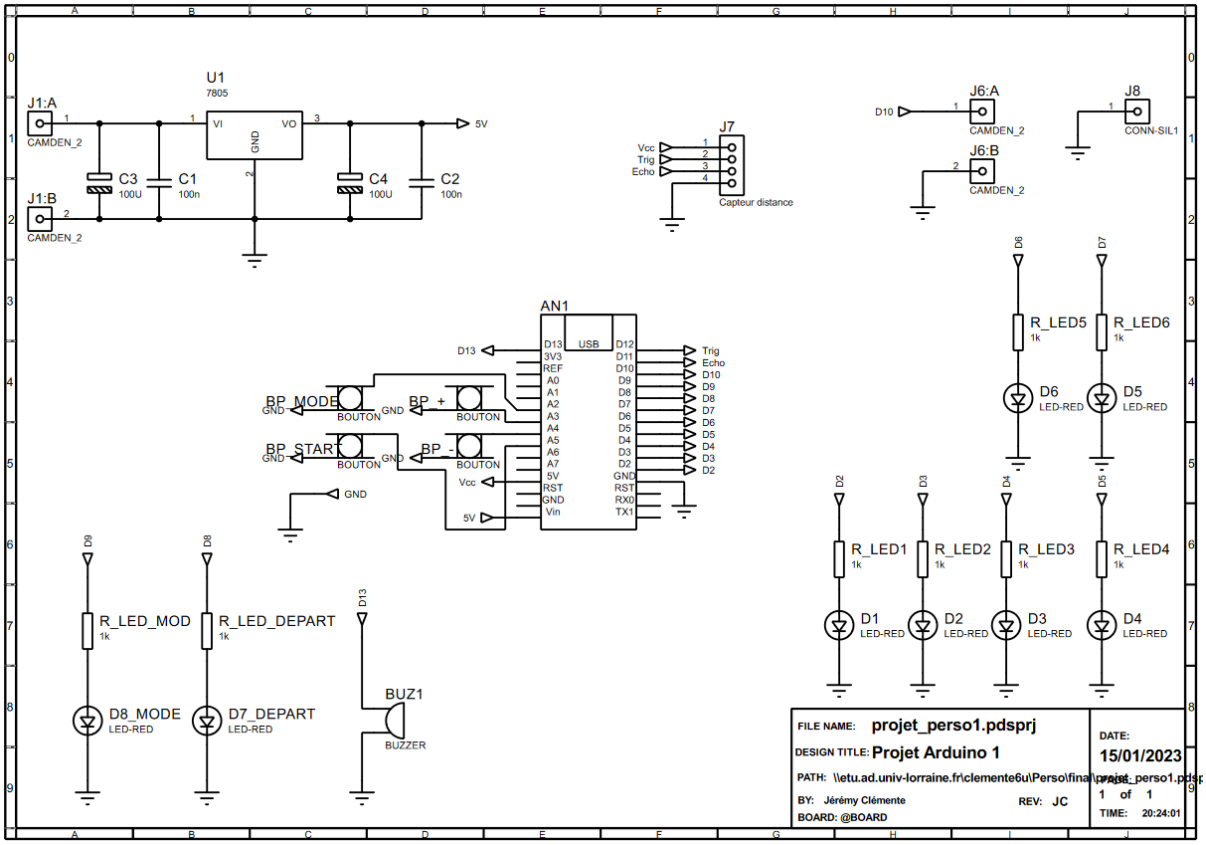
Ce mode a pour fonction de choisir une minuterie qui va de 0 à 63 mins, ce nombre est limité à 63 à cause du nombre des 6leds utiliséesvu qu' elles sont codées en binaire , le nombre maximal que l'on peut faire avec 6 leds est : $2^6 - 1 = 63$, donc on additionne la valeur des leds allumées et on obtient le temps restant ;pour choisir la durée il suffit d'appuyer sur les boutons + et - , pour lancer l'alarme il faut appuyer sur le bouton start. La durée est actualisée à chaque minute, après le départ le buzzer sonnera pendant 32 secondes à la fin du compte à rebours et après le système sera réinitialisé à0. Le seul moyen d'éteindre le compte à rebours une fois le départ lancé est d'appuyer sur le bouton reset de l'arduino nano. Les boutons de départ + et – sont bloqués lors de l'utilisation du second mode. Lors de l'activation du buzzer,un signal externe est aussi envoyé via le Camden2 à côtéde celui de l'alimentation. Il permet d'activer des circuits secondaires non présent sur la carte.

Mode Détecteur de Présence (Led orange allumée) :

Ce mode permet de détecter la présence de personne à partir de XXX cm jusqu'à la limite du capteur,quand une personne est détectée,le buzzer émettra un signal sonore ; plus la personne est proche,plus le signal sera puissant.Le système externe sera aussi alimenté quand le capteur détecte une présence.

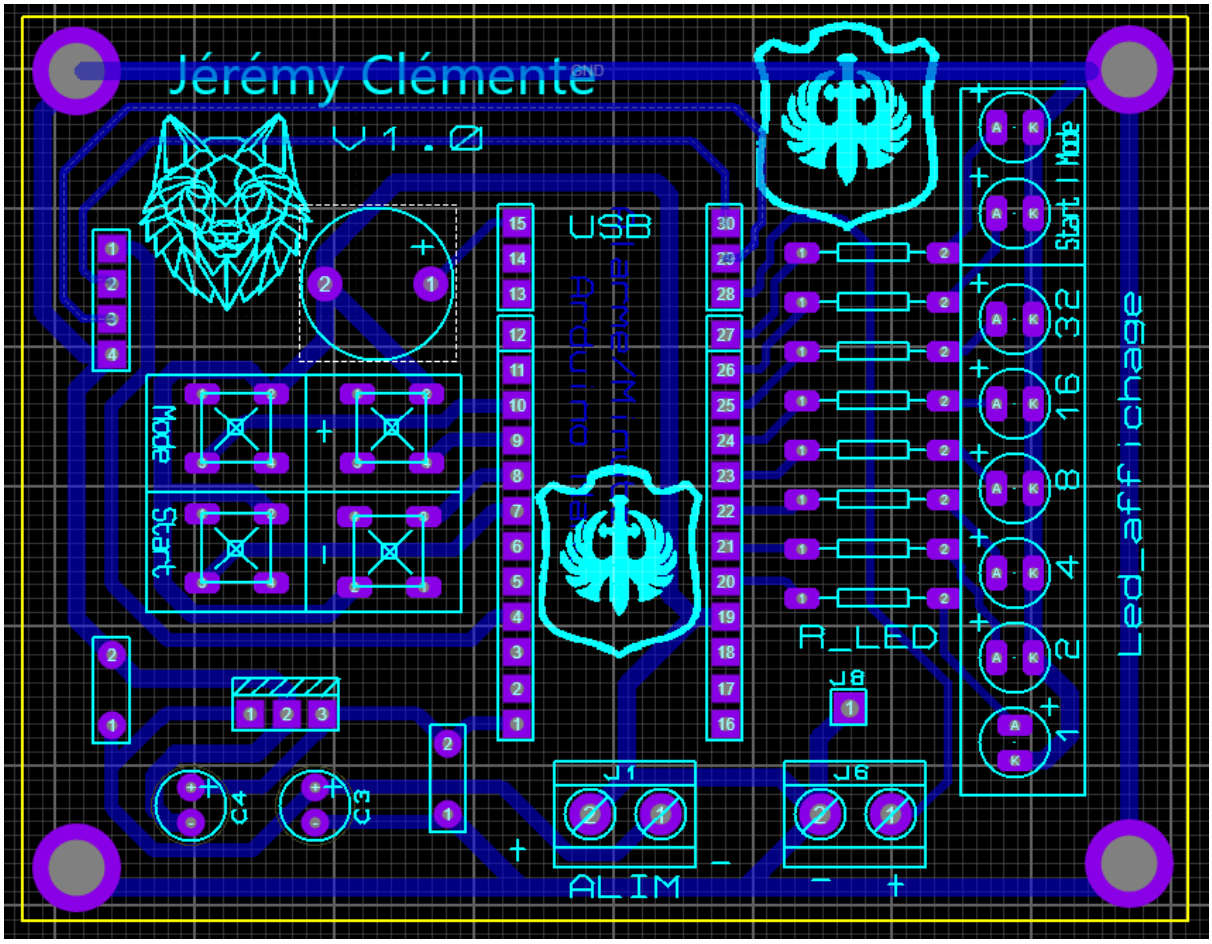
PROTHEUS (ISIS) :

Dans un premier temps,le schéma électronique a été conçu sur protheus via Isis.Certains composants n'étant pas présent de base, ilsont du être créer, comme par exemple les boutons arduino ou la carte arduino nano. Le Conn-sil présent sur le schéma est le support utilisé pour y ajouter le module à ultrasons, les Leds, Buzzer, Camden et Capteurs ultrasons, quand au bouton ils utilisent les entrées analogiques et le système d'alimentation est relié à la borne Vin de l'arduino nano.



PROTHEUS (ARES) :

Le logiciel Ares permet de faire le routage de la carte électronique grâce au schéma Isis vu précédemment. Comme certains composants ont été créés, leurs empreintes sur protheus ont du aussi être crées et certaines ont été changées comme celle du buzzer. La carte est sur du simple face et a pour dimension 75 mm x 58 mm. Certains composants peuvent être montés sur support, comme la carte arduino ou le capteur à ultrasons. Il faut faire attention lors du soudage de certains composants, comme les leds, le buzzer et les condensateurs polarisés a bien les souder dans le bon sens, sinon ils ne marcheront pas et pourraient être endommagés. Il faut aussi utiliser des composants adaptés, sinon cela ne pourrait ne pas fonctionner.



IDE ARDUINO (CODE C++) :

Cette section va expliquer comment sont codés les différents composant de la carte électronique

Led et Buzzer :

Les Leds et le buzzer sont définis sur leurs différentes broches et ensuite déclarés comme des sorties.

```
//Définition pin des led affichage
#define Led1 2
#define Led2 3
#define Led3 4
#define Led4 5
#define Led5 6
#define Led6 7
#define Led_depart 8
#define Led_mode 9

//Définition pin Buzzer
#define Buzzer 13
```

```
//definir les led en tant que sortie
pinMode(Led1, OUTPUT);
pinMode(Led2, OUTPUT);
pinMode(Led3, OUTPUT);
pinMode(Led4, OUTPUT);
pinMode(Led5, OUTPUT);
pinMode(Led6, OUTPUT);
pinMode(Led_depart, OUTPUT);
pinMode(Led_mode, OUTPUT);

//definir le buzzer en sortie
pinMode(Buzzer, OUTPUT);
```

Des variables internes sont liés aux leds, pour les leds qui servent à la minuterie les leds (bleu), on crée deux tableaux de 6 cases, car il y a 6 leds; l'un sert à savoir l'état de la led, si il faut l'allumer ou non (Tab_Led) et l'autre est la broche sur laquelle la led est concernée (Tab_led_sorti).

```
19  //Variable interne led
20  int Tab_Led[6] = { 0, 0, 0, 0, 0, 0 };
21  int i;
22  int Nbre, Nbre_copy, Quotient, Reste;
23
24  int Tab_led_sorti[6] = { 2, 3, 4, 5, 6, 7 };
```

Les variables Nbre, Nbre_copy, Quotient, Reste sert à calculer qu'elle led il faut allumer en binaire. Nbre est le temps restant en minutes, Nbre_copy est une copie de Nbre qui sert au calcul Quotient est le résultat de Nbre / 2 il sert à passer d'une led à l'autre,et reste est l'état dans lequel les leds doivent être si il reste 1 la led doit être allumée, si il reste 0 la led doit être éteinte.

On effectue déjà le calcul du reste avant le calcul du quotient et après le résultat du quotient devient Nbre . La division est une division euclidienne et on recommence 6 fois pour les six leds et remplir le tableau de l'état des leds. Puis on allume les leds en fonctions du tableau.

Il y a bien sur une sécurité dans le code qui permet d'empecher d'avoir un temps inférieur a 0 et supérieur à 63 limites du à l'affichage.

```
if (Mode == 0) {
  //LedMode
  digitalWrite(Led_mode, LOW);
  //sécurité
  if (Nbre > 63) {
    Nbre = 63;
  }
  if (Nbre < 0) {
    Nbre = 0;
  }
```

```
  //Convertion en binaire
  Nbre_copy = Nbre;
  Quotient = Nbre;

  for (i = 0; i < 6; i = i + 1) {
    Reste = Nbre_copy % 2;
    Quotient = Nbre_copy / 2;
    Nbre_copy = Nbre_copy / 2;
    Tab_Led[i] = Reste;
  }

//allumage selon le tableau

  for (i = 0; i < 6; i = i + 1) {

    if (Tab_Led[i] == 1) {
      digitalWrite(Tab_led_sorti[i], HIGH);
    } else {
      digitalWrite(Tab_led_sorti[i], LOW);
    }
  }
```

Le Timer :

Pour démarrer le timer il faut appuyer sur le bouton départ. La led départ sera allumée durant toute la durée. On actualisera les leds au bout de 1 minute. A la fin du temps le buzzer sonnera, le temps avant que le buzzer sonne dépend bien sûr du nombre lié au leds.

```
 if (Nbre > 0 && Depart == 1) {
digitalWrite(Led_depart, HIGH);
i = 0;
 while (i < 60) {
  delay(995);
   i = i + 1;
 }
Nbre = Nbre - 1;
}
```

Le Buzzer :

Le buzzer s'active une fois le compte à rebours finis et sonne pendant 32 secondes et la led verte est désactivé. On peut changer le bruit du buzzer en changeant le temps entre les phases allumer et éteinte.

```
for (i = 0; i < 4000; i = i + 1) {
  digitalWrite(Led1, LOW);
  digitalWrite(Buzzer, HIGH);
  delay(8);
  digitalWrite(Buzzer, LOW);
  delay(8);
}
digitalWrite(Led_depart, LOW);
```

Sortie externe :

Lorsque le buzzer est activée la sortie externe du circuit l'est aussi contrairement au buzzer qui alterne très rapidement pour faire son bruit la sortie du circuit l'est constamment et s'arrête après que le buzzer a fini de sonner.

Bouton :

Tous les boutons sont coder de la même façon sur un front descendant on va comparer les états après le relâchement du bouton et avant lorsque le bouton est appuyer on as donc un front descendant. Plusieurs variable sont associé a chaque bouton que une variable de lecture de l'état actuel du bouton et une variable de l'état précédent.

Exemple avec le bouton start :

```
//Variable bouton start
int btnVal_start = 0;
bool btnState_start = false;
bool oldbtnState_start = false;
void Bouton_start() { /* function testPushBtn */
  ////Read pushbutton
  btnVal_start = analogRead(bp_start);
  if (btnVal_start < 200) {
    btnState_start = true;
    if (oldbtnState_start != btnState_start) {
    }
  } else {
    btnState_start = false;
    if (oldbtnState_start != btnState_start) {
      Depart = Depart + 1;
    }
  }
  oldbtnState_start = btnState_start;
```

Capteur à ultrason :

Le capteur est actif seulement quand la led orange est allumé et allume le buzzer lorsque que le capteur détecte une personne a moins de 45 cm. Le capteur possède ces propre variable Echo et Trig, trig sert a dire au capteur d'effectuer une mesure et echo récupère le résultat. Si Echo met trop de temps a recevoir le signal on considère que la mesure n'est pas prise en compte.

```
//Variable distance
int pinTrig = 12;
int pinEcho = 11;
long temps;
float distance;
```

```
//Variable capteur ultrason
pinMode(pinTrig, OUTPUT);
pinMode(pinEcho, INPUT);

digitalWrite(pinTrig, LOW);
```

```
digitalWrite(pinTrig, HIGH);
delayMicroseconds(10);
digitalWrite(pinTrig, LOW);

temps = pulseIn(pinEcho, HIGH);

if (temps > 25000) {

}

else {
  temps = temps/2;
  distance = (temps*340)/10000.0;
}


if (distance <= 45) {
  delay(distance/2);
  digitalWrite(Buzzer, HIGH);
  delay(distance/2);
}
else {
  delay(2000);
}

digitalWrite(Buzzer, LOW);
```

ÉPILOGUE

Défaut :

Ce projet possède certain défaut qui sont les suivants :

Sur Ares la modélisation des boutons arduino sont correctes mais un peu court pour les pattes des boutons, ainsi que le camden d'alimentation est légèrement trop proche de la carte arduino.

Le système de batterie de Ares pourrais être améliorer car en cas d'inversion de la batterie cela pourrais être dangereux.

On pourrait rajouter une éventuel résistance à chaque bouton pour éviter que des parasites influences le résultat ou en cas de mauvaise modification du code. (exemple : si on déclare l'entré du bouton en sortie et que on envoie du 5 V directement au GND)

Ares on pourrais refaire le système de bouton en les espaçant un peu plus pour plus de facilité a utilisé.

Passer du mode détecteur de mouvement peut être défis compliquer. (du à cause du code)

Liste d'amélioration possible :

- Utilisation d'un décodeur pour l'affichage des leds
- Amélioration du connecteur de batterie.
- Utilisation de plusieurs afficheurs 7 segment pour remplacer l'afficheur led
- Ajout d'un bouton pour arrêter le timer une fois enclencher.
- Ajout d'un système pour changer la distance d'activation du buzzer en mode détecteur de mouvement.
- Ajout d'une possibilité de désactiver le buzzer et allumer une led a la place.

Remerciements :

Je remercie Mr Gremillet pour m'avoir aider sur la réalisation du schéma de la partie alimentation externe de la carte électronique.

Je remercie également Mr Paulmier pour m'avoir aider sur la partie réalisation de la carte électronique dans son ensemble.

Fin du rapport :

Voici la fin du Rapport sur l'alarme arduino V1.0 avec un possibilité d'amélioration vers une version supérieure, en corrigeant les différents problèmes et peut être en ajoutant quelque amélioration.