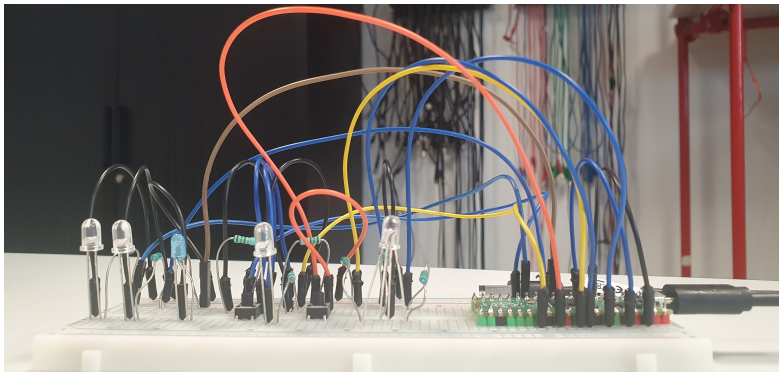

Télécom Saint-Étienne

Rapport de projet application électronique FISE 1 Groupe A 2021-2022

Programmateur lave-linge

Benjamin ALLAIN & Jossua Mignon

Semestre 6 - Bloc Signaux et Systèmes numériques

Contents

1 Rappel du sujet

Les déchets électroménagers sont un problème pour l'environnement et la santé. Symboles de la consommation de masse, ils sont très utiles dans notre vie quotidienne mais deviennent un fardeau lorsqu'ils ne sont plus utilisables. Outre l'usure légitime et compréhensible d'un appareil au bout d'une longue et parfois intense utilisation, l'obsolescence programmée accélère le cycle de remplacement. Un des points faibles est très souvent le programmateur (électromécanique ou électronique).

Proposition :

- Remplacez la partie programmation d'un lave-linge classique par un système électronique avec une interface utilisateur simple d'utilisation et que le système soit communiquant pour envoyer des notifications à l'utilisateur.

2 Introduction

Le sujet proposé entre dans le cadre général des enjeux de société autour du développement durable, de l'upcycling et de la redirection écologique. Depuis quelques années ont fleuris les "repair café", qui consistent à mettre à disposition du public des outils et des méthodes pour réparer leur matériel, généralement de l'électroménager. Au cœur de la logique de ces événements réside la lutte contre le gaspillage et l'obsolescence programmée, qui nécessite l'acquisition d'un certain savoir faire et de connaissances. Les tutoriels sur internet sont généralement très utiles pour résoudre un problème précis, réparer un objet d'une marque donnée, mais il est rare de trouver une méthodologie permettant de s'attaquer a priori à n'importe quel problème de réparation d'objet électroménager. D'autant que ces matériels utilisent des circuits de puissance, potentiellement dangereux pour un utilisateur qui ignorerait les problèmes de sécurité.

La première partie de ce projet consiste à entrer dans ce nouveau paradigme, cette nouvelle logique, pour définir la méthodologie générale d'ingénierie permettant l'analyse du besoin, sur le cas précis du programmateur de lave-linge quelle que soit sa marque, pour permettre à des utilisateurs possédant des connaissances minimales de résoudre le problème. La seconde partie consiste à appliquer cette méthodologie sur un cas concret de machine dont le programmateur est HS.

2.1 Le Hacking

A l'origine, l'obsolescence planifiée des objets a été théorisée par Bernard London dans son fascicule *l'obsolescence planifiée* [1]. Cet agent immobilier américain à imaginé cette théorie pour en finir avec la grande dépression durant laquelle les américains ne pouvaient plus acheter d'appareils mais que l'industrie continuait d'en fabriquer en grosse quantité et simplement du fait des innovations technologiques. Cela a eu pour conséquence de créer des stocks qui ne se dilapidaient pas. Dans son ouvrage, London critique le fait que les consommateurs préfèrent user un produit jusqu'à ce qu'il ne soit plus utilisable plutôt que d'en changer. Dans son contexte cette théorie est idéale, elle permet de solutionner les problèmes liés à la grande dépression. Seulement, un élément qui n'a pas été pris en compte est l'impact écologique et environnemental d'un tel mode de consommation. L'impact d'un mode de vie qui aujourd'hui encore et ce même si la grande dépression est finie. En effet, l'obsolescence programmée est aujourd'hui encore un pendant de l'industrie. On distingue deux modes d'obsolescence : évolutive et fonctionnelle. La première s'appuie sur le fait que le constructeur n'apporte pas d'amélioration ni de possibilité d'amélioration à ses appareils (Certains smartphones ne pouvant plus avoir accès à des mise à jours). La seconde est plus classique, elle repose sur le fait que l'industriel va pousser son appareil à la panne (moteur non adapté à son utilisation) avec en général peu de moyen de réparer la panne à moindre coût et simplement.

Le hacking est une pratique de plus en plus ancrée dans notre société. Cette culture laisse entrevoir une nouvelle façon d'accéder au savoir par des moyens détournés, comme la pratique de disciplines scientifiques comme l'informatique et l'électronique en loisir. Plus qu'une contre-culture isolante pour

ses membres, le hacking s'articule au système de socialisation de base de chaque individu. Subséquemment, le hacking pousse à remettre en question le rapport de chacun à la technologie de plus en plus présente et complexe nous entourant. De par la démocratisation de l'apprentissage par sois même, notamment dans les domaines tel que l'informatique et l'électronique, l'individu apporte une plus-value à son socle de connaissance acquis dans le cadre plus "conforme" de la scolarité.

Du fait que le hacking se base sur un apprentissage en autodidaxie et une importance du partage avec les pairs, la philosophie même de cette culture repose sur une notion de libre accès, de simplicité de conception afin d'être compréhensible par la majorité. Cela, couplé au fait que la plupart des hackers se professionnalisent dans un milieu proche de leur activité de hacking, laisse à penser qu'au fur et à mesure du temps, la conception même des objets technologiques du quotidien finiront par être affectée par cette mentalité. Les premier signes se font d'ailleurs de plus en plus voir (https://frame.work/de/en). Dans *The Hackor Manifesto* [2], M. Blankenship met en exergue les propos ci-dessus. En effet, l'auteur exprime le fait que les apprentissages classiques ne le satisfaisait pas. Il met aussi en lumière que le hacking est avant tout une histoire de communauté *Changer le travail ou changer la société ? Les hackers entre conformation à l'ordre social et volonté d'innover*, thèse de Eric Zufferey. -The Hackor Manifesto, Loyd Blankenship [3]
Un des impacts de l'obsolescence programmée des objets et bien sûr l'augmentation générale d'appareils. Or, tandis que la consommation augmente, le nombre de déchets augmente tout autant.

3 Cœur du projet

3.1 Schémas Fonctionnels

Une des premières étapes de notre projet est d'identifier à quoi va devoir répondre la solution technique ainsi que de déterminer les éléments qui y seront liés.

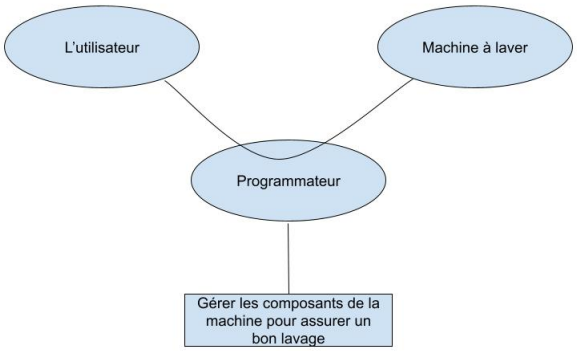

Figure 1: Bêtes à cornes

Le programmateur de la machine à laver devra réaliser plusieurs fonctions:

- Gérer de façon simple et intuitive la communication avec l'utilisateur
- Gérer différents programmes de la machine
- Gérer le bon fonctionnement du lavage notamment grâce à des capteurs

Voici le schéma fonctionnel d'une machine à laver:

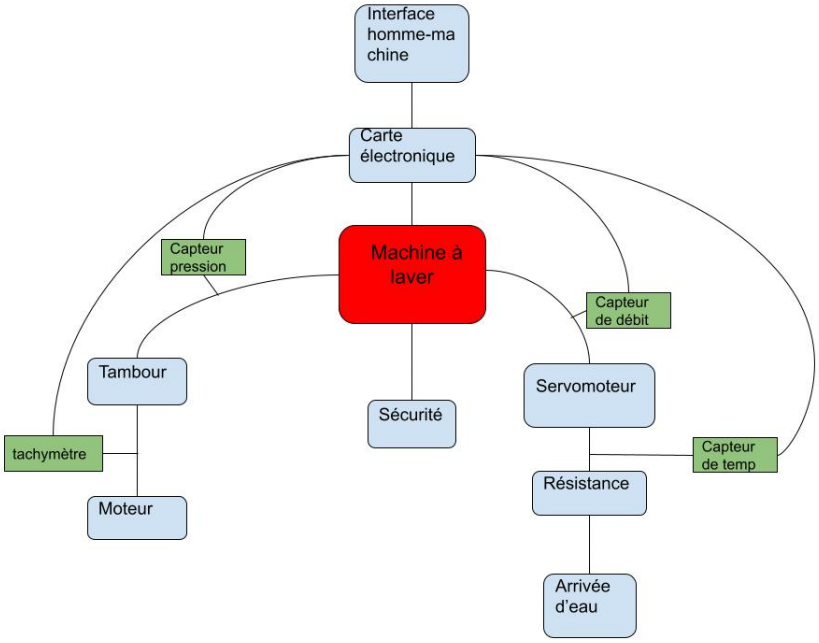

Figure 2: Schéma fonctionel machine à laver

Ces schémas, basés sur nos recherches, nous ont permis de commencer à conceptualiser le rapport entre la carte et la machine à laver. Cela nous a permis de savoir ce que nous cherchions dans la machine. Il ne manquait plus qu'à nous la procurer.

3.2 Retro Engineering

A l'aide d'annonces sur Leboncoin et d'appels à des pros et des déchetteries, nous avons réussi à trouver une machine à laver auprès d'un réparateur. Nous avons été surpris de nous apercevoir que nous ne pouvions pas simplement récupérer une machine à la déchetterie, ces derniers les revendent pour pièces ou recyclage.

Nous avons ensuite commencé à démonter la machine pour identifier les composants reliés et gérés par la carte. Ceci était très nombreux et nécessite une certaine rigueur. Chaque composant doit être soigneusement identifié afin de pouvoir savoir comment il faudra le gérer à l'avenir.
Par chance notre machine avait dans la carcasse un plan représentant tous les composants reliés à la carte et leur pin sur cette dernière. Ce plan nous a beaucoup simplifié la tâche, nous pensons que cette démarche serait à mettre en place dans toute l'industrie afin de pouvoir simplifier les réparations.
Nous avons pu identifié 3 grandes familles de composants:

- Ceux de puissances (moteur tambour et moteur pompe) nécessitant de gérer des gros ampérages.
- Ceux fonctionnant en numérique, tel que électromécanique de la porte, les boutons de l'IHM fonctionnant à l'aide d'un simple signaux digital.
- Ceux fonctionnant en analogique (capteurs pressions, températures, tachymètre) renvoyant une valeur analogiques.

3.3 Choix de la carte de programmation

Afin de remplacer la carte électronique de la machine, il nous a fallu choisir une carte permettant de pouvoir gérer les différents composants identifiés précédemment et pouvoir être simplement programmable. Le marché actuel propose une multitude de choix ! Les principaux acteurs de ce marchés

sont Arduino (UNO) et Raspberry (Pi Pico). L'avantage de la Raspberry Pi pico 2040 est quelle est capable de fournir du 32 bits contrairement a l'Arduino UNO qui fournit du 8 bits. De plus, il est plus facile d'envoyer un message à l'utilisateur avec les Raspberry (via Bluetooth ou e-mail). Enfin, elle est moins cher que l'Arduino UNO.

Notre choix c'est donc porté sur la Raspberry Pi Pico. Le premier problème que nous avons du faire face est que pour notre projet est qu'il faudra contrôler de l'électronique de puissance. Or, la carte ne sort que $300\,mA$, et il faudra $16\,A$ pour faire fonctionner correctement une machine à laver, notamment à cause des moteurs.

Figure 3: Machine récupérée grâce à un professionnel

Figure 4: Démontage de la machine à laver

3.4 Conception du code

Pour la partie software, il nous fallait créer au moins un programme de lavage. Pour se faire, nous nous sommes renseigné sur la programmation de machine à laver. Cependant, ce genre d'informations sont difficiles à obtenir, nous voyons encore une fois les freins qui sont posés par les industriels. Nous avons donc observé des machines en fonctionnement afin de déterminer ce qu'elle faisait. Nous avons identifié ces étapes principales:

- Fermer la porte
- Lancer le programme
- La machine se remplit
- Contrôler la température
- Stopper l'arrivée d'eau
- Injecter de la lessive
- Faire tourner le tambour dans un sens

Lors du lavage, la pression interne doit être surveillée ainsi que le verrouillage de la porte et le bouton de marche/arrêt de la machine pour stopper le système en cas de problème.

3.4.1 Gestion de de puissance

Afin d'apprendre à gérer l'électronique de puissance, nous avons décidé de faire clignoter une lampe 220V à l'aide de la Raspberry. Pour se faire nous avons utilisé un relai électromécanique pour faire la liaison entre la carte et la lampe. En utilisant l'interface Arduino IDE, nous avons pu programmé dans un langage proche du C (*voir code arduino*).

Cependant, nous nous sommes retrouvés limités par le relais. En effet, en essayant de faire varier l'intensité de la lampe en PWM, nous nous sommes aperçu que le relais ne parvenait pas à changer d'état assez vite pour avoir un PWM. Ce dernier étant limité par le fait qu'il est composé de pièces mécaniques amovibles ne pouvant se mouvoir assez rapidement. Pour pallier ce problème, il nous faudrait utiliser un transistor. De même, pour faire varier le sens de rotation, il nous faudrait utiliser un pont en H.

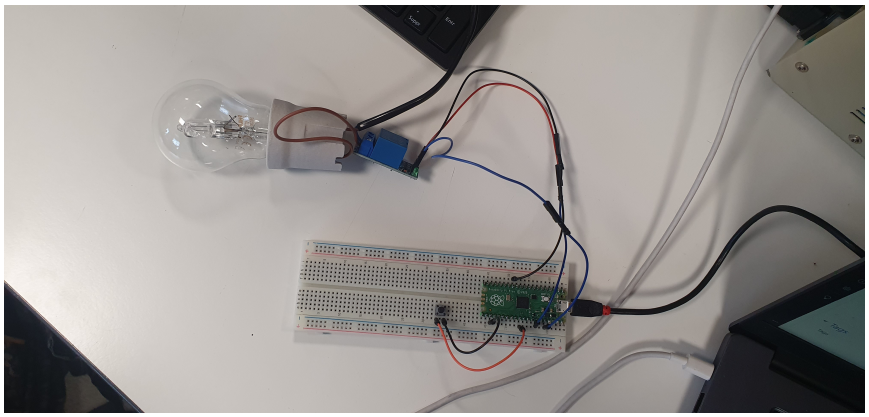

Figure 5: Gestion de puissance avec une lampe 220V

3.4.2 Gestion des LEDS

Pour faciliter la mise en place de notre circuit et afin de faciliter son débugguage, nous avons dans un premier temps remplacer les véritables composants par des LEDS. C'est le cas pour la gestion de la porte du moteur et de la pompe. Si la porte est fermée, alors la LED qui correspond à la porte s'allume et inversement. Pour le moteur, nous avons utilisé une LED RVB, une lumière bleu correspond à une rotation en sens horaire du moteur et rouge une rotation en sens anti-horaire. Enfin, la pompe fonctionne de la même manière que celle du la porte, une LED de plus est rajouté pour simuler le niveau d'eau dans la machine. Cette dernière aura une intensité proportionnel au niveau d'eau présent.

Pour simuler l'interface homme-machine nous avons rajouté deux poussons poussoirs : un qui correspond à l'état Marche/Arrêt de la machine et le deuxième pour le verrouillage de la porte.

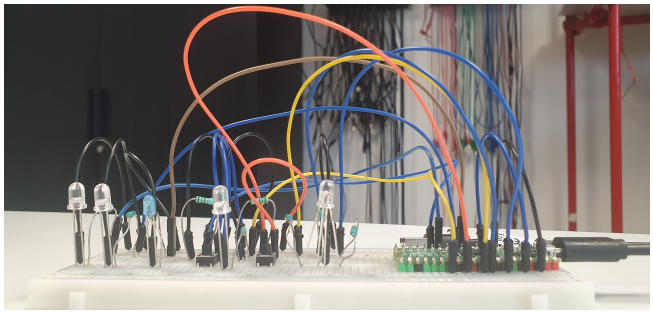

Figure 6: Montage électronique

3.4.3 Contrôle de la pression

Afin de simuler la gestion d'une entrée analogique, nous avons utilisé un potentiomètre. Face à certaines difficultés rencontrées avec le Raspberry, notamment pour les entrées analogique (Impossibilité d'importer certain module sur Arduino IDE), nous avons migré sur Arduino Uno. L'avantage de notre projet et sa grande flexibilité pour changer de support. En effet, il ne faut pas modifier grand chose pour que le programme développé avec la Raspberry Pi Pico fonctionne sur UNO.

3.4.4 Gestion des interruptions et explications du changement de carte

Une autre difficulté est le fait de lire à tout moment la valeur du capteur de pression et de contrôler l'appuie sur le bouton marche/arrêt ou le contrôle de la porte, ceci à tout moment de l'exécution de la boucle. Pour se faire, nous avons étudié les interruptions. Ces dernières permettent de à tout moment d'interrompre la loop afin d'exécuter une fonction prédéfinie. L'interruption sur une sortie s'exécute quand cette sortie change d'état dans notre cas. Les sorties d'interruption sur Arduino UNO sont les sorties 2 et 3.

Pour la gestion de l'interruption sur le potentiomètre (simulant la pression), il nous a fallu passer par un watchdog, permettant une interruption à un temps donné. A ce moment nous prenons la valeur du potentiomètre puis nous la comparons à la limite de pression. Une fois toutes ces simulations faites, nous créons sur breadboard une simulation du programme.

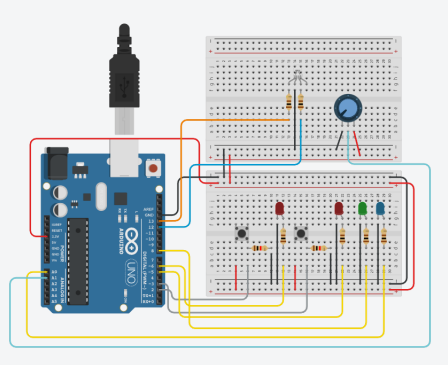

Figure 7: Montage électrique final sur simulation

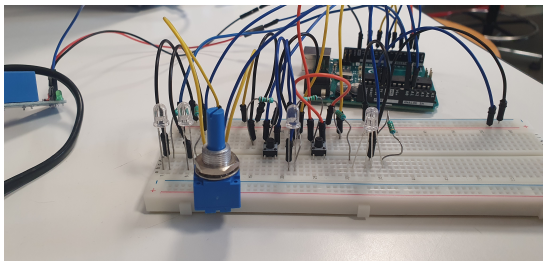

Figure 8: Montage électrique final

4 Critique

Dans les premières séance, nous avons eu du mal à comprendre les véritables enjeux de ce projet. Effet, comme il s'agit de Retro-Engineering, la méthodologie à adopter pour ce projet diffère de ce que nous avons l'habitude de faire. Nous avons du eu l'impression de ne pas avancer pendant plusieurs semaines sans avoir de visibilité sur le travail à fournir.

5 Pour aller plus loin

Notre projet était un beau projet d'électronique, qui peut encore se continuer. Les avancements que nous aurions pu apporter sont:

-Remplacer la LED RVB par un véritable moteur. Chose que nous avons commencer à faire (pour la partie montage électrique) mais nous n'avons pas aboutit.

-Ajouter un PWM au niveau du moteur afin de faire varier sa vitesse de rotation.

-Analyser chaque éléments de la machine pour les caractériser. Par exemple connaitre valeur maximum de la pression, vitesse maximum du moteur, de la température. Cependant, cela était compliqué à trouver car il s'agit de matériel propriétaire.

Enfin, la partie de concrétisation du projet, mais aussi la plus complexe aurait été d'adapter la simulation à la machine, afin de la hacker.

6 Bibliographie

1. *L'obsolescence planifiée* - Bernard London

2. *The Hackor Manifesto* - M. Blankenship

3. *Changer le travail ou changer la société ? Les hackers entre conformation à l'ordre social et volonté d'innover* - Eric Zufferey

4. *Le Grand Livre d'Arduino* - Erik Bartmann

7 Annexe

Voici le code Arduino utilisé:

```
const byte ledporte = 10;
const byte boutonporte = 2; //LE port 2 permet une interruption
const byte boutonmarche = 3;// Le port 3 permet une interruption
const byte ledmarche = 13;
const byte ledpompe = 12;
const byte ledjauge = A0; //Sur port analogique pour faire varier l  intensite  lumineuse
const byte rouge = 8;
const byte bleu = 7;
const byte curseur = A1; //Sur port analogique
const byte lampe = 5;

volatile byte state = LOW;
int compteurporte =0; //Compte le nombre de clique sur le bouton porte
int compteurmarche =0; //Compte le nombre de clique sur le bouton marche
bool executeporte = false;
bool executemarche = false;
int valcurseur = 0;

void setup() {
  Serial.begin(7000);

  pinMode(ledporte, OUTPUT);
  pinMode(ledmarche, OUTPUT);
  pinMode(ledpompe, OUTPUT);
  pinMode(ledjauge, OUTPUT);
  pinMode(lampe, OUTPUT);

  pinMode(boutonporte, INPUT_PULLUP);
  pinMode(boutonmarche, INPUT_PULLUP);

  attachInterrupt(digitalPinToInterrupt(boutonporte), arretporte, CHANGE); //Interruption par le
       bouton de la porte lancant la sequence arreteporte a tout changement de valeur du PIN
  attachInterrupt(digitalPinToInterrupt(boutonmarche), arretmarche, CHANGE);

  cli(); // Desactive l  interruption  globale
  bitClear (TCCR2A, WGM20); // WGM20 = 0
  bitClear (TCCR2A, WGM21); // WGM21 = 0
  TCCR2B = 0b00000110; // Clock / 256 soit 16 micro-s et WGM22 = 0
  TIMSK2 = 0b00000001; // Interruption locale autorisee par  TOIE2
  sei(); // Active l  interruption  globale
}
```

```
void loop() {
digitalWrite(lampe, HIGH); // Le relais etant inverse (a 1 par defaut) lampe a HIGH eteint la lampe
  if (executeporte ==true && executemarche == true){ //On verifie que nos deux appuies sont bien
      realises
    digitalWrite(ledpompe, HIGH); //On lance la pompe amenant l eau
    for (int i=0; i<=255; i+=51){ // On augmente la jauge d eau jusqu a remplir la machine
      analogWrite(ledjauge, i);
      Serial.print("Dans la boucle ");
      Serial.println(i);
      delay(1000);
    }




    digitalWrite(ledpompe, LOW); //On eteint la pompe car la machine est pleine
    digitalWrite(rouge, HIGH); // Le moteur tourne dans un sens
    digitalWrite(bleu, LOW);
    digitalWrite(lampe, LOW); // On allume la lampe afin de prouver qu on peut gerer les composants
        demandant de la puissance
    delay(1000);
     digitalWrite(lampe, HIGH); // On eteint la lampe
    delay(2000);
    }


    digitalWrite(rouge, LOW);
    digitalWrite(bleu, HIGH); // On allume le moteur dans l autre sens, representer par une autre
        couleur de led
    digitalWrite(lampe, LOW); // On redemontre la puissance
    delay(1000);
    digitalWrite(lampe, HIGH);

    delay(2000);
    digitalWrite(rouge, LOW);
    digitalWrite(bleu, LOW); // Le moteur s'eteint
    digitalWrite(ledpompe, HIGH); // La pompe se relance
    for (int i=255; i<=0; i-=51){ // On vide l'eau de la machine
      analogWrite(ledjauge, i);
      delay(1000);
    }
    digitalWrite(ledpompe, LOW);
    delay(2000);
  }
}



void arretporte() {
  if(digitalRead(boutonporte)==true){
    compteurporte++; // Si on detecte un appui sur le bouton porte on incremente le compteur de 1
  }
  if(compteurporte&1){ //On verifie que le nombre d appuie est impair
a    executeporte =true; // Si c est le cas, on passe execute porte true, la condition est remplie
    digitalWrite(ledporte, HIGH); //on allume la LED temoin
  }



  else if(compteurporte%2 == 0){ // Dans le cas ou le nombre de clique est pair
  digitalWrite(ledporte, LOW);
  executeporte = false; // On ne valide pas la condition
```

```
        digitalWrite(rouge, LOW);
        digitalWrite(bleu, LOW);
        digitalWrite(ledjauge, LOW);
        digitalWrite(ledpompe, LOW);
        digitalWrite(ledmarche, LOW);
        digitalWrite(ledporte, LOW);
        digitalWrite(lampe, HIGH); //On eteint l ensemble du systeme pour le preserver
    exit(0); // On quitte la loop, le programme ne peut plus executer, l utilisateur est oblige de reset
          la carte pour pouvoir relancer la machine ceci afin de lui laisser le temps de comprendre d ou
          vient la panne.
  }

}


void arretmarche() { //Le fonctionnement est le meme que precedemment.
  if(digitalRead(boutonmarche)==true){
    compteurmarche++;
  }
  if(compteurmarche&1){
    executemarche =true;
    digitalWrite(ledmarche, LOW);
  }
  else if(compteurmarche%2 == 0){
  digitalWrite(ledmarche, LOW);
  executemarche = false;
        digitalWrite(rouge, LOW);
        digitalWrite(bleu, LOW);
        digitalWrite(ledjauge, LOW);
        digitalWrite(ledpompe, LOW);
        digitalWrite(ledmarche, LOW);
        digitalWrite(ledporte, LOW);
        digitalWrite(lampe, HIGH);
  exit(0);
  }
}


byte varCompteur = 0; // La variable compteur

// Routine d'interruption
ISR(TIMER2_OVF_vect) {
  TCNT2 = 256 - 250; // 250 x 16 microS = 4 ms
  if (varCompteur++ > 125) { // 125 * 4 ms = 500 ms (demi-periode)
    varCompteur = 0;
    valcurseur = map(analogRead(curseur), 0, 675, 0, 255); //On lit la valeur du potentiometre qu
         on map pour l avoir entre 0 et 255
    if ( valcurseur >150){ //Si la valeur est superieure a 150, soit la pression du systeme trop eleve
        digitalWrite(rouge, LOW);
        digitalWrite(bleu, LOW);
        digitalWrite(ledjauge, LOW);
        digitalWrite(ledpompe, LOW);
        digitalWrite(ledmarche, LOW);
        digitalWrite(ledporte, LOW);
        digitalWrite(lampe, HIGH);
        exit(0); // Le systeme se met en securite
    }
  }
}
```