

DUAULT GASTON

Baccalauréat
technologique, série
STI2D; spécialité SIN

Soutenance de Projet
2019

Coffre-Fort Connecté



GRUPE

- DUAULT Gaston
- CELTON Pierre
- GERARD Ludovic
- GERARD Quentin

JOLIOT-CURIE
144, Bd de Vitré
35703 RENNES CEDEX 7

2018-2019

Sommaire

1) Présentation du projet

- a. Problématique.....p3
- b. Enjeux économiques, environnementaux et sociaux....p4
- c. Composants.....p4

2) Organisation du travail

- a. Mettre en place le digicode.....p5
- b. Mettre en place les badges et le capteur de badges.....p5
- c. Mettre en place le moteur et la gâche.....p5-6
- d. Mettre en place l'accéléromètrep6
- e. Mettre en place le serveur et la base de données.....p6

3) Réalisation du projet

- a. Tâches personnelles.....p7

4) Avancement

- a. Problèmes rencontrés.....p9
- b. Programmation.....p9-10

5) Conclusion.....p10

Le projet que nous avons réalisé est un coffre-fort connecté. Ce projet a été conçu pour une entreprise qui stockera les objets de clients en sécurité. Le coffre-fort communique avec une base de données. L'administrateur réseau pourra consulter sur cette base de données l'historique d'ouverture (par qui et quand).

Problématique:

Comment notre coffre-fort se différencie-t-il d'un coffre classique?



- *ECONOMIQUE*
- *ENVIRONMENTAL*
- *SOCIAL*

- **Économique** : Le budget requis pour une maquette s'estime à 200€. Les revenus de l'entreprise sont supérieurs aux coûts des coffres. Nous pouvons donc dire que notre entreprise et notre projet sont rentables.
- **Environnemental** : Notre projet a un faible impact sur l'environnement. En effet, notre coffre a une consommation électrique minime. Notre carte Arduino ainsi que notre gâche et moteur sont alimentés sous 12V. De plus, le corps du coffre pourrait être constitué de métal recyclé.
- **Social** : Notre projet crée de l'emploi. En effet, si notre projet est vu comme une entreprise et non comme un coffre-fort unique et privé, nous aurons besoin d'employé pour s'occuper de la maintenance des coffres et de leur entretien.

- Composants** :
- Carte Arduino
 - Carte Raspberry PI 2
 - Digicode 3x4
 - Capteur RFID 125hz
 - Moteur shield MFA RE385
 - Gâche à électro-aimant
 - Module LED rouge et verte
 - Accéléromètre groove 3axis +-16g
 - Boîte aux lettres (corps du coffre-fort)

Mettre en place le digicode

Le digicode est un clavier 3x4. Il permet de pouvoir écrire son mot de passe, chaque utilisateur a son mot de passe qui correspond à chaque badge. La touche « * » permet de valider le mot de passe tandis que la touche « # » permet d'effacer le mot de passe . Cette tâche a été réalisée par Quentin GERARD.



Mettre en place les badges et le lecteur de badges

Afin de pouvoir écrire son mot de passe, il faut tout d'abord valider son badge. Le lecteur de badges lit l'adresse que porte un badge. Chaque badge a sa propre adresse. Les badges ont une fréquence de 125Khz. Pour que le badge soit validé, il ne suffit pas juste de le lire avec un lecteur de badge, il faut que celui-ci ait une adresse qui soit connue et approuvée (comme un mot de passe). Cette tâche a été réalisée par Pierre CELTON.



Mettre en place la gâche

La gâche et le moteur ne sont pas alimentés directement par la carte Arduino car elle ne dispose pas d'assez de puissance. La gâche a besoin de 12 V minimum afin de fonctionner correctement. La gâche est une gâche à électro-aimant et non à serrure. Sa fonction est de verrouiller ou déverrouiller le coffre. Cette tâche a été réalisée par Quentin GERARD.



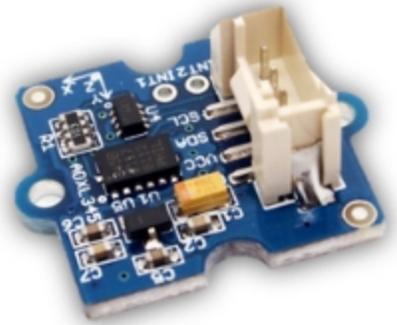
Mettre en place le moteur

Le moteur et la gâche sont pilotés depuis la carte Arduino et alimentés par un amplificateur (carte motor shield Arduino). Le moteur que nous utilisons est un moteur MFA RE385, sa tension varie de 6Vmin à 15Vmax. A vide, sa vitesse de rotation est égale à 1100tr/min. Le moteur permet de refermer le coffre automatiquement après 5 minutes d'ouverture. J'ai réalisé cette tâche.



Mettre en place l'accéléromètre

L'accéléromètre est un appareil qui détecte et mesure les accélérations. Nous utilisons un accéléromètre car il permet d'éviter les vols ou les dégradations sur le coffre-fort. C'est un accéléromètre 3 axis (x,y,z), z représente la gravité soit 9.81 m/s^2 , sa plage de mesure est de -16g à 16g. Il est piloté depuis la carte Arduino . J'ai réalisé cette tâche.



Mettre en place le serveur et la base de données

Le serveur est sur la carte Raspberry, nous l'utilisons afin de pouvoir héberger la base de données. Nous avons utilisé Apache pour pouvoir concevoir le serveur et Mysql pour pouvoir réaliser la base de données. Sur la base de données, l'administrateur réseau pourra consulter quand et par qui le coffre a été ouvert. Le serveur a été mis en place par Pierre CELTON et la base de données a été réalisée par Ludovic GERARD.



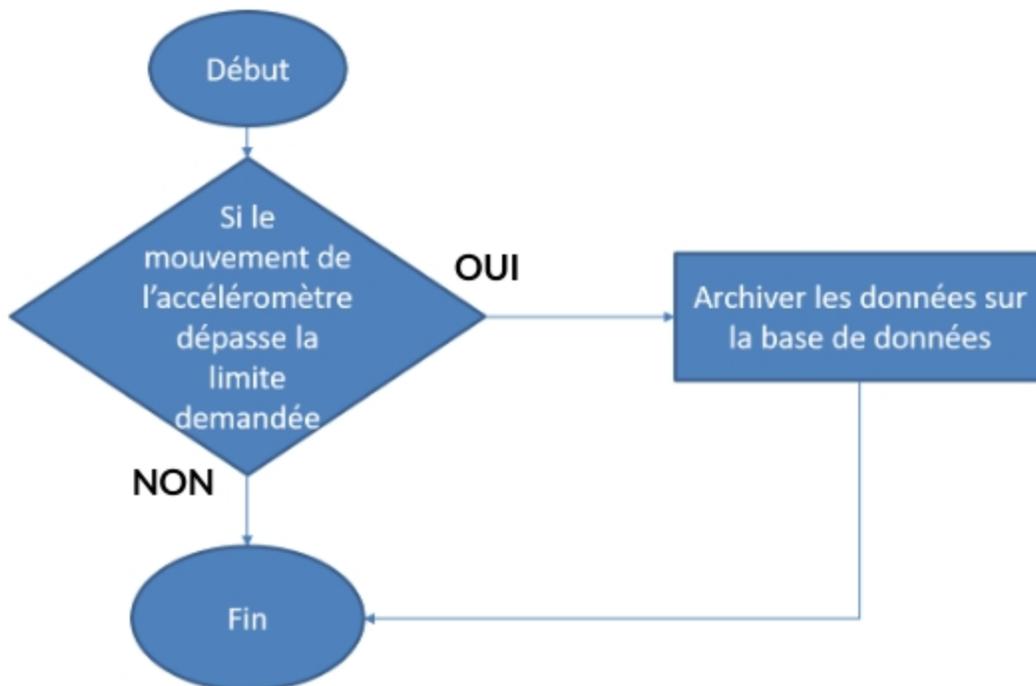
Mes tâches personnelles :

Dans ce projet, nous nous sommes réparti le travail afin de pouvoir finir notre projet dans les délais établis.

Ma première tâche attribuée a été la mise en place de l'**accéléromètre**, ensuite la mise en place du **moteur**, puis d'établir la **communication entre la carte Arduino et la carte Raspberry Pi** avec un câble USB. Et enfin d'écrire le **programme général**.

1. L'accéléromètre:

L'accéléromètre permet d'apporter plus de sécurité à notre coffre-fort : il détecte les mouvements et les chocs, si celui-ci détecte un mouvement anormal, les informations seront envoyées à la base de données.



2. Le Moteur :

Le moteur referme la porte du coffre-fort au bout de 5 minutes d'ouverture avec un cordon, un cordon relie l'arbre du moteur et la porte. Lorsque le moteur s'actionne, le cordon tire la porte du coffre. Nous avons utilisé une poulie que nous avons fixée sur l'arbre du moteur afin de fixer le cordon et qu'il puisse s'enrouler facilement et inversement.

3. Communication entre la carte Raspberry Pi et la carte Arduino :

Nous utilisons deux cartes car la carte Arduino doit être à l'intérieur du coffre pour pouvoir piloter l'accéléromètre, les digicodes, le lecteur de badges, la gâche et le moteur ainsi que les LED. Tandis que la carte Raspberry Pi doit être à l'extérieur pour pouvoir héberger le serveur et la base de données. La connexion entre les deux cartes se fait par liaison USB, pour nous cette solution nous semblait être la plus simple. Sur la carte Arduino, pour envoyer un message, nous utilisons la simple commande « Serial.print ». Un message sera envoyé toutes les deux secondes. Ensuite, sur la carte Raspberry, nous utilisons Python et la commande « Serial.serial ». La donnée ne circule que de la carte Arduino à la carte Raspberry.

Résultat sur la carte Raspberry

```
pi@EspaceRaspberryFrançais:~ $ sudo python comArd.py
Un message va etre envoye toutes les deux secondes des maintenant !

Message #0

Message #1

Message #2

Message #3
```

4. Ecrire le programme général :

Le programme général regroupe tous les programmes en un seul programme. Ma tâche consistait à l'écrire et faire en sorte de simplifier le plus possible le programme général et aussi corriger les erreurs et bugs. Puis, j'ai dû apporter des solutions aux problèmes rencontrés afin de garantir le bon fonctionnement de notre projet.

Problèmes rencontrés :

Lors de la conception de notre projet, nous avons pu constater que le système élaboré pour la fermeture automatique du coffre avait un défaut. En effet, si l'utilisateur referme le coffre manuellement avant les 5 minutes, le moteur lui ne pourra en être informé. De plus, la carte Arduino, lors de la fermeture du coffre automatique, ne pourra pas savoir quand la porte sera totalement refermée et, si le moteur continue de tirer la porte, cela pourra dégrader le mécanisme du moteur. Nous avons deux solutions, la première est d'utiliser un capteur de présence au niveau de la porte afin de savoir quand celle-ci est ouverte ou bien fermée. La deuxième solution est de ne jamais refermer le coffre manuellement.

Programmations :

L'accéléromètre :

```
double xyz[3];
double ax, ay, az;
adxl.getAcceleration(xyz);
ax = xyz[0];
ay = xyz[1];
az = xyz[2];
```

initialisation des 3 axes

```
Serial.print("X=");
Serial.print(ax);
  Serial.print(" g ");
Serial.print("Y=");
Serial.print(ay);
  Serial.print(" g ");
Serial.print("Z=");
Serial.print(az);
  Serial.println(" g");
Serial.println("*****");
```

affichage en boucle des
mesures des 3 axes en g

```
if (abs(ax) > 0.2 || abs(ay) > 0.2 || abs(az-0.82) > 0.2) {
  Serial.println("Message #" + String(i));
  Serial.println("ALERT LE COFFRE BOUGE");
}
else
{
}
```

Si l'accéléromètre dépasse plus de 20% de son poids, il faut envoyer un message au Raspberry sinon on ne fait rien

Le Moteur :

```

if (millis() - dateM > date_moteur) {
  value = 100;
  digitalWrite(M1, LOW);
  analogWrite(E1, value);
  dateM2 = millis();
}

```

Si le coffre est ouvert plus de 5 minutes, le moteur est actionné

```

if(millis() - dateM2 > date_moteur2) {
  value = 1;
  analogWrite(E1, value);
}

```

Si le moteur est actionné depuis 3 sec alors il faut l'arrêter

```

else {
  value = 1;
  analogWrite(E1, value);
}

```

sinon on ne fait rien

- "digitalWrite (M1, LOW)" signifie que le moteur se met en marche et "LOW" désigne le sens de rotation.

- "analogWrite" signifie la vitesse à laquelle le moteur doit tourner, "value = 0" la vitesse est nulle, "value = 255" vitesse maximum.

Conclusion :

Ce projet permet de pouvoir conserver des objets en toute sécurité. Grâce à la base de données, notre coffre-fort connecté se distingue d'un coffre-fort classique. Des améliorations pourraient être apportées, nous pourrions faire ouvrir le coffre-fort automatiquement et y ajouter une batterie externe qui permettrait d'en faire un coffre-fort mobile. Ce projet était très intéressant, il m'a permis de pouvoir acquérir des connaissances en programmation en langage C++ et aussi quelques connaissances en Python. Cela ne fait que renforcer mon projet d'orientation vers un IUT Informatique.

```
int badg1 = 0;
bool psd = false;
int date_gache = 15000;
int date_badge = 15000;
int date_moteur = 2000;
int date_moteur2 = 5000;
int date2;
int dateM;
int dateM2;

const int NB_BADGE = 4;
char tab_badge[NB_BADGE][20] = {"0000000000000", "050080891F13", "070081B6F9C9", "08002526474C"};
char tab_code[NB_BADGE][20] = {".....", "123456", "654321", "000000"};

void setup() {
    Serial.begin(9600);
    pinMode(LED_V, OUTPUT);
    pinMode(LED_R, OUTPUT);
    setup_accelerometre();
    setup_badge();
    setup_digicode();
    setup_gache();
    //setup_moteur();
}

void loop() {

    badge();

    if(badg1 != 0){
        Serial.println(badg1);
        gererClavier();

        if(psd == true){
            Serial.println("VALIDE");
            gache();
        }

        else {
            Serial.println("NON VALIDE");
        }
    }
    moteur();
    accelerometre();
}
```

```
void setup_badge() {
  // juste pour voir si la carte reboote
  Serial.println("Goodnight moon!");

  // set the data rate for the SoftwareSerial port
  mySerial.begin(9600);
}

void badge() { // run over and over
  char buffer_[20] = "";
  if (mySerial.available())
  {
    mySerial.readBytesUntil(3, buffer_, 20);
    Serial.println(buffer_ + 1);
  }

  if ((badgel == 0) || (millis() - date > date_badge))
  {
    badgel = 0;
    for (int i=0; i<NB_BADGE; i++)
    {
      if (strncmp(tab_badge[i], buffer_ + 1, 12) == 0) {
        Serial.println("Badge # VALIDE");
        badgel = i;
        date = millis();
      }
    }
  }
}
```

```

char keys[ROWS][COLS] = {
  {'1', '2', '3'},
  {'4', '5', '6'},
  {'7', '8', '9'},
  {'*', '0', '#'}
};
byte rowPins[ROWS] = {12, 11, 10, 9};
byte colPins[COLS] = {3, 2, 13};

Keypad keypad = Keypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS);

char buffer_[7] = {0};
char passwd[7] = {0};

void setup_digicode() {
  wdt_enable (WDTO_8S);
}
int compteur = 0;

void gererClavier()
{
  wdt_reset (); // remet à zéro le timeout du watchdog
  Serial.print("---");
  Serial.print(buffer_);
  Serial.println("---");

  char customKey = keypad.getKey();

  if (customKey != NO_KEY) {
    if (customKey == '#') {
      compteur = 0;
      buffer_[compteur] = 0;
      //passwd[compteur] = 0;
    }
    if (customKey == '*') {
      Serial.println("---");
      Serial.println(buffer_);
      Serial.println("---");
      if (strcmp(tab_code[badgel], buffer_, 6) == 0) {
        Serial.println("mot de passe VALIDE:");
        Serial.println(buffer_);
        compteur = 0;
        buffer_[compteur] = 0;
        psd = true;
        digitalWrite(LED_V ,HIGH);
        digitalWrite(LED_R ,LOW);
      }
      else if (strcmp(tab_code[badgel], buffer_, 6) != 0) {
        Serial.println("mot de pas NON valide");
        compteur = 0;
        buffer_[compteur] = 0;
        psd = false;
        digitalWrite(LED_V ,LOW);
        digitalWrite(LED_R ,HIGH);
      }
    }
  }
  else{
    buffer_[compteur] = customKey;
    compteur++;
    buffer [compteur] = 0;
  }
}

```

```
int E2 = 6;
int M2 = 7;

void setup_gache()
{
    pinMode(E2, OUTPUT);
    pinMode(M2, OUTPUT);
}

void gache()
{
    int value = 255;
    if (psd == true)
    {
        digitalWrite(M2, LOW);
        analogWrite(E2, value);
        date2 = millis();
        dateM = millis();
    }

    if (millis() - date2 > date_gache){
        digitalWrite(M2, LOW);
        analogWrite(E2, 0);
    }
}
```