Livrable 2

Equipe 7

Alves Gabriel, Belleux Maxence, Colson Paul, Kadiri Shahinèze, Ouach Bilal, Rebeilleau–Dassonneville Lila

Introduction

La célèbre agence d'espionnage MI7 a été mise à mal après de nombreux détournement de matériel destiné à leurs agents. Les cachettes mises à disposition n'étant plus assez sûres, un des agents de l'organisation a eu l'idée de concevoir un coffre-fort nouvelle génération basé sur des mécanismes d'authentification plus efficaces pour assurer un degré de sécurité plus élevé. Nous avons donc réalisé un prototype de ce montage pour le cinquième niveau d'authentification qui décrit son processus complet. Celui-ci présentera l'identification du modèle de notre carte, la détermination du niveau de sécurité ainsi que l'appel aux différents mécanismes d'authentification et jusqu'à l'ouverture du coffre.

L'objectif du livrable 2 est de présenter l'ensemble du code permettant la mise en place des différents mécanismes d'authentification:

- MA1: Authentification par questions/réponses
- MA2: Authentification par un code qui change au cours du temps (via BP)
- MA3: Scan rétinien (simulé)
- MA4: Scan digital (simulé)
- MA5: Basique (lettre Agent/CARD_ID lecture analogique)
- MA5 Avancé (facultatif)

Les critères suivants sont à respecter:

[3] Implémenter une description algorithmique dans un programme en langage C Arduino
Le programme est fonctionnel
Le code correspond au raisonnement algorithmique présenté dans le Check 2 - si des modifications ont été apportées, des explications sont fournies
Le programme principal est simple et fait appel à plusieurs fonctions et/ou procédures
Le programme permet de simuler tous les systèmes d'authentification pour chaque niveau de sécurité
[3] Déclarer et utiliser des fonctions dans un programme en langage C Arduino
Chaque processus d'authentification est implémenté par une fonction séparée
Le code est factorisé (pas de répétition inutile dans le code, d'autres fonctions sont créées si nécessaires)
Les fonctions sont bien déclarées et utilisées (nom pertinent, type de retour, paramètres)
[3] Utiliser les instructions d'entrée/sortie dans un programme en langage C Arduino
Le programme affiche les instructions d'authentification à l'utilisateur
Le programme permet à l'utilisateur de saisir les informations d'authentification nécessaires
Le programme interragit avec le circuit électronique pour la lecture de la tension d'entrée et les boutons poussoirs de l'authentification par code agence (boucle 6)
Le programme interragit avec le circuit électronique pour l'affichage par LEDs de l'état d'authentification par code agence et pour l'ouverture du coffre
[2] Distinguer les différents types de variables et estimer leur portée (locale, globale, constante, ...)
Le choix des noms de variables est pertinent
Le choix des types de variables est pertinent
Le choix de la portée des variables est pertinent

Montage électrique

Si le lien tinkercad ne s'affiche pas dans hedgedoc, on peut le voir directement dans tinkercad via https://www.tinkercad.com/things/iSnndKXLGn0-etape-2?sharecode=PQOCJJoqLtMwXiT4KJ7tcSnioBabaFe4f4zOBvwriM4

Code Arduino

Ici on initialise les entiers du tableau b avec le numero des broches de l'arduino auxquels ils sont connectés. Idem pour le tableau l et les broches auxquels les leds sont connectés.
Dans le setup on définit les boutons en tant qu'entrées avec valeur définie, et les leds en tant que sorties.
On fait ensuite appel aux fonctions MA1, MA3, MA4 et MA5 en arrêtant le programme si elles ne sont pas true.

```
/*Note : MA2 necessite d'etre dans la loop, il sera donc verifie
en dernier.
Reponses : MA1 : 2 puis 1 ; MA3 : message secret ; MA4
message secret numero 2 ; MA5 : A puis 1234 par exemple ;
MA2 : le code : 1110 sans randomseed
*/
const int b[4] = {7, 8, 12, 13};//broches connectees aux boutons poussoirs
const int l[5] = {2, 3, 4, 5, 6};//broches connectees aux leds
int code[4];
int etatb[4];
bool b1, b2, b3, b4 = false;

void setup() {
  for (int i = 0; i < 4; i++) {
   code[i] = random(0,2);
    Serial.println(code[i]);
  }
  Serial.begin(9600);// on initialise la vitesse à 9600 bauds

  for (int i = 0; i < 4; i++) {
    pinMode(b[i], INPUT_PULLUP);//definition des boutons en tant
  }//que sorties avec valeur bien definie (input_pullup)
  for (int i = 0; i < 5; i++) {
    pinMode(l[i], OUTPUT);//definition des leds en tant que sorties
  }
  //test des differents niveaux de securite, ma2 se fait dans la loop
  //Si le test n'est pas bon, on arrete le programme (return)
  if (!MA1()) {
   Serial.println("Erreur");
    return;
  }
  if (!MA3()) {
    Serial.println("Erreur");
    return;
  }
  if (!MA4()) {
    Serial.println("Erreur");
    return;
  }
  if (!MA5()) {
    Serial.println("Erreur");
    return;
  }
  Serial.print("Entrez le code a l'aide des boutons poussoirs");
}
```

```
Erreur
Entrez le code a l'aide des boutons poussoirs
```

MA1 est un test de deux questions à choix multiple. L'utilisateur doit choisir la réponse correspondant à celle dans le tableau rep. La fonction renvoie false si une des deux questions n'a pas été répondue correctement, true sinon.

```
bool MA1() {
  int rep[2] = {2, 1};
  int input = 0;
  Serial.println("Question 1 ? Choisir 1, 2 ou 3");
  while (Serial.available() == 0) {
    //boucle infinie jusqu'a ce que l'utilisateur rentre une valeur
  }
  input = Serial.parseInt();
  if (input != rep[0]) { //comparaison de l'entree utilisateur
   return false;       //et de la reponse
  }
  input = 0;
  Serial.println("Question 2 ? Choisir 1, 2 ou 3");
  while (Serial.available() == 0) {
  }
  input = Serial.parseInt();
  return input == rep[1];//return true si la condition est bonne
}       //false sinon
```

```
Question 1 ? Choisir 1, 2 ou 3
Question 1 ? Choisir 1, 2 ou 3
```

Dans MA3 et MA4, l'utilisateur doit rentrer une phrase qui doit correspondre à celle contenue dans la variable "message".
Elles renvoient la valeur booléenne de "entrée de l'utilisateur est égale au message".

```
bool MA3() {
  Serial.println("Entrez le message");
  String message = "message secret";
  while (Serial.available() == 0) {
  }
  String input = Serial.readString();
  return input == message;
}
```

```
Entrez le message
```

```
bool MA4() {
  Serial.println("Entrez le message");
  String message = "message secret numero 2";
  while (Serial.available() == 0) {
  }
  String input = Serial.readString();
  return input == message;
}
```

```
Entrez le message
```

Pour valider MA5, l'utilisateur doit rentrer sa lettre "code d'agent" et l'identifiant de la carte qui correspond. On renvoie la valeur booléenne de "i est compris entre 0 et 16 et i correspond à son identifiant de carte." 'i' est ici le numéro de l'agent convertir en un entier.

```
bool MA5() {
  Serial.println("Entrez votre lettre de code d'agent");
  int idcardlist[16] = {
  1234, 5678, 9012, 3456, 7890, 2345, 6789, 1230,
  4567, 8901, 2340, 5671, 8902, 1233, 4568, 7891};
  while (Serial.available() == 0) {
  }
  char agentcode = Serial.read();
  delay(10);
  Serial.println("Entrez votre id de carte");
  while (Serial.available() == 0) {
  }
  int idcard = Serial.parseInt();
  int i = agentcode - 'A'; //on convertit la lettre en int en lui
  //soustrayant la valeur de A dans la table ascii
  return i >= 0 && i < 16 && idcard == idcardlist[i];
}
```

```
Entrez votre lettre de code d'agent
Entrez votre id de carte
```

MA2 est fait dans la loop car nous devons vérifier l'état des boutons de manière permanente. C'est la dernière vérification des 5 qui est executée, ainsi on print "Authentification réussie" si cette étape est réussie. L'implémentation diffère de celle initialement faite sur flowgorithm, les détails de l'algorithme sont expliqués en commentaire du code.

```
void loop() {
  if (digitalRead(b[0]) == LOW) {//lecture de l'etat du bouton
     etatb[0] = 1;//1 si appuye
   } else {
      etatb[0] = 0;//0 sinon
   }
  if (etatb[0] == code[0]) {//comparaison entre l'etat du bouton et le 1er bit du code
   digitalWrite(l[0], HIGH);//allumer la led correspondante si condition remplie
   b1 = true;//variable check pour condition led verte
  }

  if (digitalRead(b[1]) == LOW) {
     etatb[1] = 1;
   } else {
      etatb[1] = 0;
   }
  if (etatb[1] == code[1]) {
   digitalWrite(l[1], HIGH);
   b2 = true;
  }

  if (digitalRead(b[2]) == LOW) {
     etatb[2] = 1;
   } else {
      etatb[2] = 0;
   }
  if (etatb[2] == code[2]) {
   digitalWrite(l[2], HIGH);
   b3 = true;
  }

  if (digitalRead(b[3]) == LOW) {
     etatb[3] = 1;
   } else {
      etatb[3] = 0;
   }
  if (etatb[3] == code[3]) {
   digitalWrite(l[3], HIGH);
   b4 = true;
  }

  if (b1 && b2 && b3 && b4) {//si toutes les leds rouges == HIGH, led verte == HIGH
   digitalWrite(l[4], HIGH);
   Serial.println("Authentification reussie");
   while(true); //pour ne pas print Auth reussie en boucle
  }
}
```

```
Authentification reussie
```