

Projet Strongbox 3000

Livrable 4 - Full
StrongBox
3000

Groupe C5

JACQUES-LAURENCE NULLAR

AXEL JOLY-LEVERT

DESHANI KUMARATHAS

TASSADIT CHAIB



SOMMAIRE

INTRODUCTION.....	2
CONTEXTE DU LIVRABLE.....	2,3
MÉTHODOLOGIE DE TRAVAIL.....	4-8
CONCLUSION.....	9
PERSPECTIVES.....	9,10
BIBLIOGRAPHIE.....	11



Introduction

Suite à la conception de l'algorithme du système interne d'authentification effectuée dans le livrable 3, nous passons ainsi au livrable 4 dans lequel, une fois notre algorithme sera correcte, nous allons l'implémenter à l'aide d'une carte Arduino

Contexte du livrable

Dans ce livrable, nous allons dans un premier temps, compléter notre circuit réaliser dans le livrable afin qu'il soit bien prêt pour l'implémentation. Dans un second temps, nous devons traduire l'algorithme principale du livrable 3 en code Arduino. Une fois la carte et le code prêts, nous allons fusionner les deux sur Tinkercad et vérifier si notre coffre-fort fonctionne.

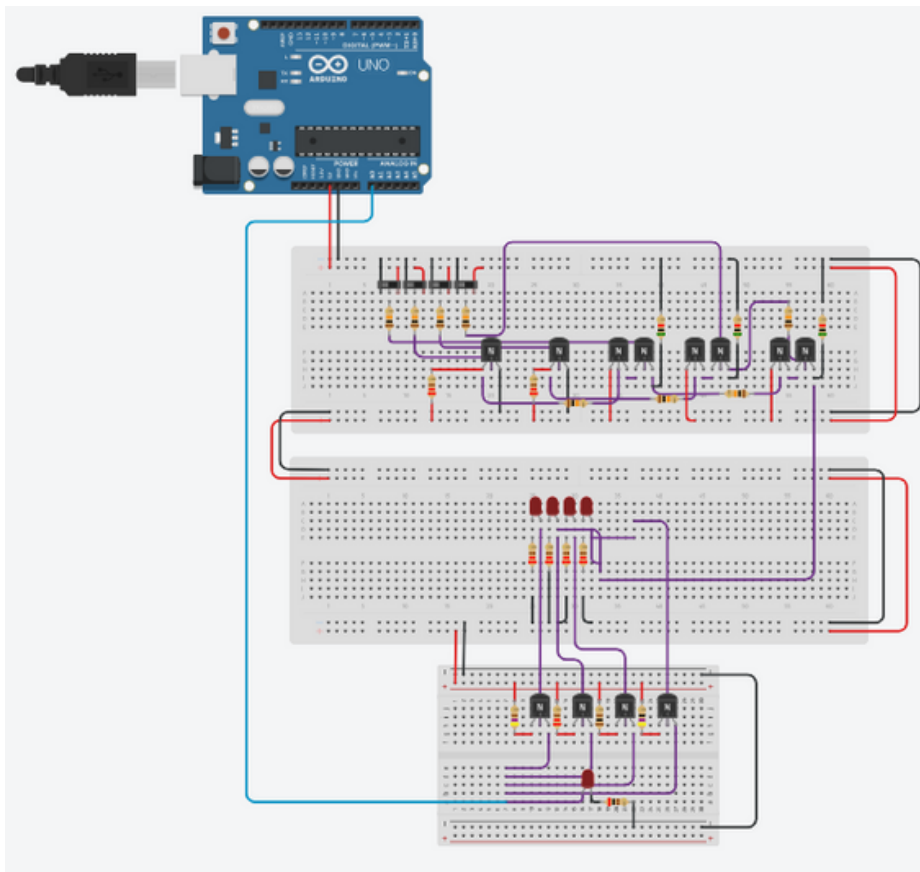
Méthodologie de travail

Les besoins:

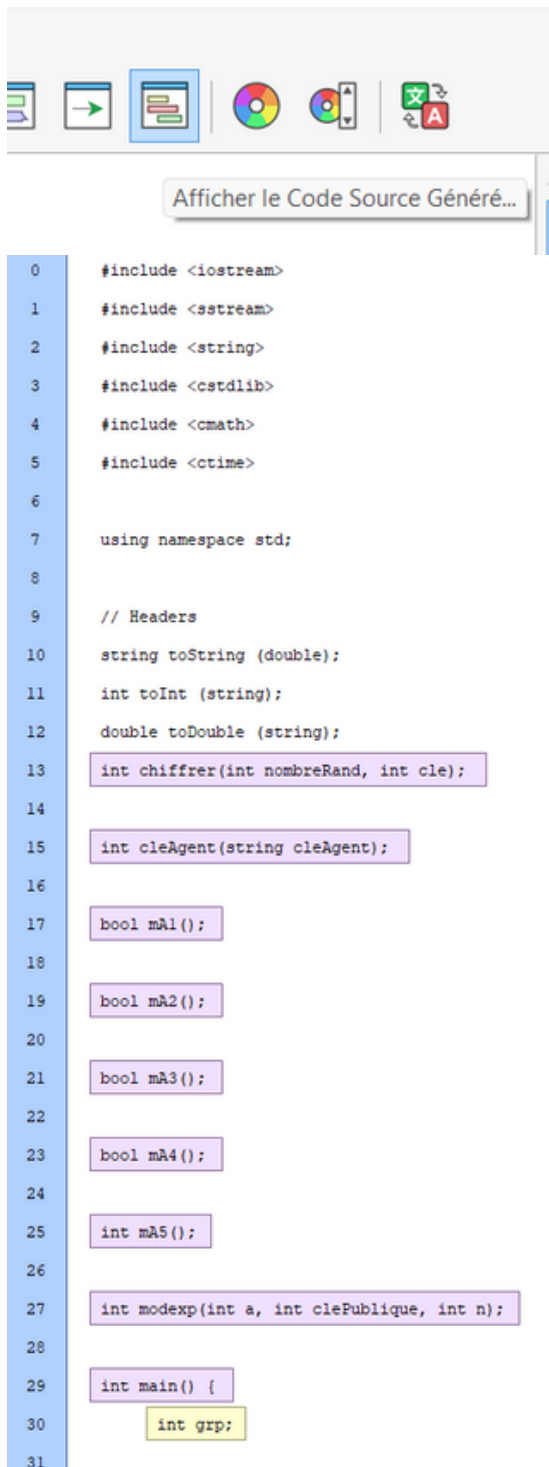
- Flowgorithm
- Tinkercad
- CLion
- C Arduino

Description détaillée du travail réalisé:

Dans un premier temps, nous avons corrigé quelques erreurs de notre circuit effectué dans le livrable 2. Nous avons changé les valeurs de nos résistances et nous avons utilisé la fonction “`analogRead()`” afin de calculer la tension R5 de chaque modèle de carte de la série E24.



En second lieu, nous avons repris le logigramme principal du coffre-fort réalisé dans le livrable 3 pour le convertir en code Arduino sur Tinkercad. Pour cela, nous avons utilisé une fonction sur Flowgorithm qui nous permet d'avoir le code du logigramme en C++.



```

0  #include <iostream>
1  #include <sstream>
2  #include <string>
3  #include <cstdlib>
4  #include <cmath>
5  #include <ctime>
6
7  using namespace std;
8
9  // Headers
10 string toString (double);
11 int toInt (string);
12 double toDouble (string);
13 int chiffrier(int nombreRand, int cle);
14
15 int cleAgent(string cleAgent);
16
17 bool mA1();
18
19 bool mA2();
20
21 bool mA3();
22
23 bool mA4();
24
25 int mA5();
26
27 int modexp(int a, int clePublique, int n);
28
29 int main() {
30     int grp;
31

```

```

32     grp = mA5();
33     if (grp < 3) {
34         cout << "Vous appartenez au Groupe 1, vous allez donc passer le niveau de sécurité 1" << endl;
35         if (mA1() == true && mA3() == true) {
36             cout << "[Niveau Sécurité 1] >> Réussi" << endl;
37             cout << "Coffre Ouvert !" << endl;
38         } else {
39             cout << "[Niveau Sécurité 1] >> Erreur" << endl;
40             cout << "Destruction du coffre" << endl;
41         }
42     } else {
43         if (grp < 6) {
44             cout << "Vous appartenez au Groupe 2, vous allez donc passer le niveau de sécurité 2" << endl;
45             if (mA1() == true && mA4() == true) {
46                 cout << "[Niveau de securite 2] >> Réussi" << endl;
47                 cout << "Coffre ouvert !" << endl;
48             } else {
49                 cout << "[Niveau Sécurité 2] >> Erreur" << endl;
50                 cout << "Destruction du coffre" << endl;
51             }
52         } else {
53             if (grp < 9) {
54                 cout << "Vous appartenez au Groupe 3, vous allez donc passer le niveau de sécurité 3" << endl;
55                 if (mA2() == true && Code() == true) {
56                     cout << "[Niveau Sécurité 3] >> Réussi" << endl;
57                     cout << "Coffre Ouvert !" << endl;
58                 } else {
59                     cout << "[Niveau Sécurité 3] >> Erreur" << endl;
60                     cout << "Destruction du coffre" << endl;
61                 }
62             } else {
63                 if (grp < 12) {
64                     cout << "Vous appartenez au Groupe 4, vous allez donc passer le niveau de sécurité 4" << endl;
65                     if (mA2() == true && mA3() == true && mA4() == true) {
66                         cout << "[Niveau Sécurité 4] >> Réussi" << endl;
67                         cout << "Coffre ouvert !" << endl;
68                     } else {
69                         cout << "[Niveau Sécurité 4] >> Erreur" << endl;
70                         cout << "Destruction du coffre" << endl;
71                     }
72                 } else {
73                     if (grp < 16) {
74                         cout << "Vous appartenez au Groupe 5, vous allez donc passer le niveau de sécurité 5" << endl;
75                         if (mA1() == true && mA2() == true && mA3() == true && mA5() == true) {
76                             cout << "[Niveau Sécurité 5] >> Réussi" << endl;
77                             cout << "Coffre ouvert !" << endl;
78                         } else {
79                             cout << "[Niveau Sécurité 5] >> Erreur" << endl;
80                             cout << "Destruction du coffre" << endl;
81                         }
82                     } else {
83                         }
84                     }
85                 }
86             }
87         }
88     return 0;
89 }

```



```

90
91 int chiffrer(int nombreRand, int cle) {
92     int n, chiffre;
93
94     n = 43 * 67;
95     chiffre = modexp(nombreRand, cle, n);
96
97     return chiffre;
98 }
99
100 int cleAgent(string cleAgent) {
101     int cle;
102
103     if (cleAgent == "A") {
104         cle = 601;
105     } else {
106         if (cleAgent == "B") {
107             cle = 619;
108         } else {
109             if (cleAgent == "C") {
110                 cle = 631;
111             } else {
112                 if (cleAgent == "D") {
113                     cle = 641;
114                 } else {
115                     if (cleAgent == "E") {
116                         cle = 647;
117                     } else {
118                         if (cleAgent == "F") {
119                             cle = 653;
120                         } else {
121                             if (cleAgent == "H") {
122                                 cle = 661;
123                             } else {
124                                 if (cleAgent == "I") {
125                                     cle = 673;
126                                 } else {
127                                     if (cleAgent == "J") {
128                                         cle = 691;
129                                     } else {
130                                         if (cleAgent == "K") {
131                                             cle = 701;
132                                         } else {
133                                             if (cleAgent == "L") {
134                                                 cle = 733;
135                                             } else {
136                                                 if (cleAgent == "M") {
137                                                     cle = 739;
138                                                 } else {
139                                                     if (cleAgent == "N") {
140                                                         cle = 751;
141                                                     } else {
142                                                         if (cleAgent == "O") {
143                                                             cle = 797;
144                                                         } else {
145                                                             if (cleAgent == "P") {
146                                                                 cle = 809;
147                                                             } else {
148                                                                 if (cleAgent == "Q") {
149                                                                     cle = 811;
150                                                                 }
151                                                             }
152                                                         }
153             }
154         }
155     }
156 }

```

```

154
155
156
157
158
159
160
161
162
163
164
165
166
167     return cle;
168 }
169
170 bool mAl() {
171     int reponse;
172     bool etat;
173
174     cout << "Quel est le projet du MI7 ?" << endl;
175     cout << "1 >> MI7 Security" << endl;
176     cout << "2 >> Strongbox 2.0" << endl;
177     cout << "3 >> StrongBox 3000" << endl;
178     cout << "4 >> Agent Double" << endl;
179     cin >> reponse;
180     if (reponse == 3) {
181
182         // Si l'utilisateur à la bonne réponse alors il peut passer à la question suivante
183         cout << "[Niveau 1] >> Réussi" << endl;
184         cout << "Quel est le nom de notre école ?" << endl;
185         cout << "1 >> ESILV" << endl;
186         cout << "2 >> ECE" << endl;
187         cout << "3 >> EMLV" << endl;
188         cout << "4 >> CESI" << endl;
189         cin >> reponse;
190         if (reponse == 4) {
191             cout << "[Niveau 2] >> Accès réussi" << endl;
192             cout << "Quel est le nom de la promotion de cette année" << endl;
193             cout << "1 >> CPI A1" << endl;
194             cout << "2 >> CPI A2" << endl;
195             cin >> reponse;
196             if (reponse == 1) {
197                 cout << "[Niveau 3] >> Réussi" << endl;
198                 cout << "Vous pouvez passer au prochain niveau de sécurité" << endl;
199                 etat = true;
200             } else {
201                 cout << "Accès refusé" << endl;
202                 etat = false;
203             }
204         } else {
205             cout << "Accès refusé" << endl;
206             etat = false;
207         }
208     } else {
209
210         // Si la réponse à la question n'est pas correcte alors l'accès à la question suivante est refusé et l'algorithme s'arrête
211         cout << "Accès refusé" << endl;
212         etat = false;
213     }
214
215     return etat;
216 }

```



```

217
218 bool mA2() {
219     bool etat;
220     bool ouverture;
221     int magent;
222
223     magent = 0;
224     int clePublique, m, mp, c;
225     string nomAgent;
226
227     cout << "Veuillez vous identifier avec votre nom" << endl;
228     cin >> nomAgent;
229     m = rand() % 2881;
230     clePublique = cleAgent(nomAgent);
231     c = chiffrer(m, clePublique);
232     while (m != magent) {
233         cout << "Message à déchiffrer" << endl;
234         cout << c << endl;
235         cin >> magent;
236     }
237     cout << "Authentification Réussie" << endl;
238     ouverture = true;
239     etat = true;
240
241     return etat;
242 }
243
244 bool mA3() {
245     bool scanEyes;
246     bool etat;
247
248     cout << "Veuillez approcher votre oeil du capteur" << endl;
249     cin >> scanEyes;
250     if (scanEyes == true) {
251
252         // Vérifies si le scan rétinien est correct
253         cout << "Scan rétinien reconnu" << endl;
254         cout << "Vous pouvez passer au prochain niveau de sécurité" << endl;
255         etat = true;
256     } else {
257         cout << "[Erreur] >> Scan rétinien non reconnu" << endl;
258         etat = false;
259     }
260
261     return etat;
262 }
263
264 bool mA4() {
265     bool scanDigital;
266     bool etat;
267
268     cout << "Veuillez approcher votre doigt du capteur" << endl;
269     cin >> scanDigital;
270     if (scanDigital == true) {
271
272         // Vérifies si le scan digital est correct
273         cout << "Scan digital reconnu" << endl;
274         cout << "Vous pouvez passer au prochain niveau de sécurité" << endl;
275         etat = true;
276     } else {

```

```

277     cout << "[Erreur] >> Scan digital non reconnu" << endl;
278     etat = false;
279 }
280
281     return etat;
282 }
283
284 int mAS() {
285     bool etat;
286     int groupe;
287     int i;
288
289     i = 0;
290     string code[16];
291
292     code[0] = "A1258";
293     code[1] = "B2865";
294     code[2] = "C1658";
295     code[3] = "D1100";
296     code[4] = "E0101";
297     code[5] = "F0110";
298     code[6] = "G0111";
299     code[7] = "H1000";
300     code[8] = "I1001";
301     code[9] = "K1010";
302     code[10] = "L1011";
303     code[11] = "M1100";
304     code[12] = "N1101";
305     code[13] = "O1835";
306     code[14] = "P8563";
307     code[15] = "Q7569";
308     bool trouvé;
309     string aTrouvé;
310
311     cout << "Entrez votre lettre et code d'agent" << endl;
312     cin >> aTrouvé;
313     trouvé = false;
314     while (trouvé != true) {
315         int compteur;
316
317         for (compteur = 15; compteur >= 0; compteur--) {
318             if (code[compteur] == aTrouvé) {
319                 cout << "Trouvé à la position : " << compteur << endl;
320                 trouvé = true;
321                 groupe = compteur;
322                 etat = true;
323             }
324         }
325     }
326
327     return groupe;
328 }
329
330 int modexp(int a, int clePublique, int n) {
331     int r;
332
333     if (clePublique < 0) {
334         exit(-1);
335     }
336     if (a == 0 || n == 1) {
337         r = 0;
338     } else {
339         r = 1;
340         while (clePublique > 0) {

```

```

341         r = r * (a % n) % n;
342         clePublique = clePublique - 1;
343     }
344 }
345
346 return r;
347 }
348
349 // The following implements type conversion functions.
350 string toString (double value) { //int also
351     stringstream temp;
352     temp << value;
353     return temp.str();
354 }
355
356 int toInt (string text) {
357     return atoi(text.c_str());
358 }
359
360 double toDouble (string text) {
361     return atof(text.c_str());
362 }

```

Ensuite, nous avons retranscrit le code C++ en code C.



Les 10 principales différences entre C et C++

C et C++ sont deux langages de programmation largement reconnus pour leur puissance de bas

```
#include <stdio.h>

int chiffrer(int nombreRand, int cle);
int cleAgent(const char* cleAgent);
int ma1();
int ma2();
int ma3();
int ma4();
int ma5();
int modexp(int a, int clePublique, int n);

int main() {
    int grp;

    grp = ma5();
    if (grp < 3) {
        printf("Vous appartenez au Groupe 1, vous allez donc passer le niveau de securite 1\n");
        if (ma1() == 1 && ma3() == 1) {
            printf("[Niveau Securite 1] >> Reussi\n");
            printf("Coffre Ouvert !\n");
        } else {
            printf("[Niveau Securite 1] >> Erreur\n");
            printf("Destruction du coffre\n");
        }
    } else if (grp < 6) {
        printf("Vous appartenez au Groupe 2, vous allez donc passer le niveau de securite 2\n");
        if (ma1() == 1 && ma4() == 1) {
            printf("[Niveau de securite 2] >> Reussi\n");
            printf("Coffre ouvert !\n");
        } else {
            printf("[Niveau Securite 2] >> Erreur\n");
            printf("Destruction du coffre\n");
        }
    } else if (grp < 9) {
        printf("Vous appartenez au Groupe 3, vous allez donc passer le niveau de securite 3\n");
        if (ma2() == 1 && ma5() == 1) {
            printf("[Niveau de securite 3] >> Reussi\n");
            printf("Coffre ouvert !\n");
        } else {
            printf("[Niveau Securite 3] >> Erreur\n");
            printf("Destruction du coffre\n");
        }
    } else if (grp < 12) {
        printf("Vous appartenez au Groupe 4, vous allez donc passer le niveau de securite 4\n");
        if (ma2() == 1 && ma3() == 1 && ma4() == 1) {
            printf("[Niveau de securite 4] >> Reussi\n");
            printf("Coffre ouvert !\n");
        } else {
            printf("[Niveau Securite 4] >> Erreur\n");
            printf("Destruction du coffre\n");
        }
    } else if (grp < 16) {
        printf("Vous appartenez au Groupe 5, vous allez donc passer le niveau de securite 5\n");
        if (ma1() == 1 && ma2() == 1 && ma3() == 1 && ma5() == 1) {
            printf("[Niveau de securite 5] >> Reussi\n");
            printf("Coffre ouvert !\n");
        } else {
            printf("[Niveau Securite 5] >> Erreur\n");
            printf("Destruction du coffre\n");
        }
    }
}

return 0;
}
```

Ensuite, nous avons retranscrit le code C en code Arduino.

<https://www.arduino.cc/reference/fr/>

```

int id;

int ma1() {
    int etat;

    Serial.println("Quel est le projet du MI7 ?");
    Serial.println("1 >> MI7 Security");
    Serial.println("2 >> Strongbox 2.0");
    Serial.println("3 >> StrongBox 3000");
    Serial.println("4 >> Agent Double");

    while (Serial.available() > 0) { // Nettoyage du buffer série
        Serial.read();
    }

    while (Serial.available() == 0) {} // Attendre l'entrée de l'utilisateur
    String reponse = Serial.readStringUntil('\n');
    reponse.trim(); // Enlève les espaces et les sauts de ligne
    delay(100); // Délai pour stabiliser le buffer série

    if (reponse == "3") {
        Serial.println("[Niveau 1] >> Reussi");
        Serial.println("Quel est le nom de notre ecole ?");
        Serial.println("1 >> ESILV");
        Serial.println("2 >> ECE");
        Serial.println("3 >> EMLV");
        Serial.println("4 >> CESI");

        while (Serial.available() == 0) {} // Attendre l'entrée de l'utilisateur
        String reponse = Serial.readStringUntil('\n');
        reponse.trim(); // Enlève les espaces et les sauts de ligne
        delay(100); // Délai pour stabiliser le buffer série

        if (reponse == "4") {
            Serial.println("[Niveau 2] >> Acces reussi");
            Serial.println("Quel est le nom de la promotion de cette annee");
            Serial.println("1 >> CPI A1");
            Serial.println("2 >> CPI A2");

            while (Serial.available() == 0) {} // Attendre l'entrée de l'utilisateur
            String reponse = Serial.readStringUntil('\n');
            reponse.trim(); // Enlève les espaces et les sauts de ligne
            delay(100); // Délai pour stabiliser le buffer série

            if (reponse == "1") {
                Serial.println("[Niveau 3] >> Reussi");
                Serial.println("Vous pouvez passer au prochain niveau de securite");
                etat = 1; // Etat défini sur True
            } else {
                Serial.println("Acces refuse");
                etat = 0; // Etat défini sur False
                return etat; // Remplace exit(-1) pour ne pas arrêter le programme Arduino
            }
        } else {
            Serial.println("Acces refuse");
            return 0; // Remplace exit(-1)
        }
    } else {
        Serial.println("Acces refuse");
        return 0; // Remplace exit(-1)
    }
}

return etat;
}

int ma2() {
    int etat;
    int magent = -1; // Initialise magent à une valeur non valide
    int clePublique, m, c;
    char nomAgent;

    Serial.println("Veuillez-vous identifier");
    while (Serial.available() == 0) {} // Attendre l'entrée de l'utilisateur
    nomAgent = Serial.read();

    // Nettoyer le buffer série pour enlever les données résiduelles
    while (Serial.available() > 0) {
        Serial.read();
    }

    m = random(2881); // Remplace srand et rand
    clePublique = cleAgent(nomAgent);
    c = chiffrer(m, clePublique);

    Serial.print("Message à déchiffrer: ");
    Serial.println(c);

```

```

while (m != magent) {
    while (Serial.available() == 0) {} // Attendre l'entrée de l'utilisateur
    magent = Serial.parseInt(); // Lire un entier depuis la communication série

    // Nettoyer le buffer série après la lecture
    while (Serial.available() > 0) {
        Serial.read();
    }

    Serial.println("Authentification autorisee");
    etat = 1;
    return etat;
}

int chiffrer(int nombreRand, int cle) {
    int n = 43 * 67;
    int chiffre = modexp(nombreRand, cle, n);
    return chiffre;
}

int cleAgent(char cleAgent) {
    int cle = 0;
    // Utiliser des conditions directes, remplace strcmp
    switch (cleAgent) {
        case 'A': cle = 601; break;
        case 'B': cle = 619; break;
        case 'C': cle = 631; break;
        case 'D': cle = 641; break;
        case 'E': cle = 647; break;
        case 'F': cle = 653; break;
        case 'H': cle = 739; break;
        case 'I': cle = 673; break;
        case 'J': cle = 691; break;
        case 'K': cle = 701; break;
        case 'L': cle = 733; break;
        case 'M': cle = 739; break;
        case 'N': cle = 751; break;
        case 'O': cle = 797; break;
        case 'P': cle = 809; break;
        case 'Q': cle = 811; break;
        default: cle = 0; // Valeur par défaut si la clé n'est pas reconnue
    }
    return cle;
}

while (m != magent) {
    while (Serial.available() == 0) {} // Attendre l'entrée de l'utilisateur
    magent = Serial.parseInt(); // Lire un entier depuis la communication série

    // Nettoyer le buffer série après la lecture
    while (Serial.available() > 0) {
        Serial.read();
    }

    Serial.println("Authentification autorisee");
    etat = 1;
    return etat;
}

int chiffrer(int nombreRand, int cle) {
    int n = 43 * 67;
    int chiffre = modexp(nombreRand, cle, n);
    return chiffre;
}

int cleAgent(char cleAgent) {
    int cle = 0;
    // Utiliser des conditions directes, remplace strcmp
    switch (cleAgent) {
        case 'A': cle = 601; break;
        case 'B': cle = 619; break;
        case 'C': cle = 631; break;
        case 'D': cle = 641; break;
        case 'E': cle = 647; break;
        case 'F': cle = 653; break;
        case 'H': cle = 739; break;
        case 'I': cle = 673; break;
        case 'J': cle = 691; break;
        case 'K': cle = 701; break;
        case 'L': cle = 733; break;
        case 'M': cle = 739; break;
        case 'N': cle = 751; break;
        case 'O': cle = 797; break;
        case 'P': cle = 809; break;
        case 'Q': cle = 811; break;
        default: cle = 0; // Valeur par défaut si la clé n'est pas reconnue
    }
    return cle;
}

```

```

int modexp(int a, int e, int n) {
    long r;
    if (e < 0) {
        return -1;
    }
    if (a == 0 || n == 1) {
        return 0;
    }
    r = 1;
    while (e > 0) {
        r = (r * (a % n)) % n;
        e--;
    }
    return r;
}

void setup() {
    Serial.begin(9600); // Initialise la communication série
    randomSeed(analogRead(0)); // Initialise le générateur de nombres aléatoires
}

```

```

int ma3() {
    int etat = 0;
    char retine[8]; // Buffer pour stocker la saisie de l'utilisateur
    const char* retine_id[] = {
        "retineA", "retineB", "retineC", "retineD", "retineE", "retineF",
        "retineH", "retineI", "retineJ", "retineK", "retineL", "retineM",
        "retineN", "retineO", "retineP", "retineQ"
    };
}

```

```

Serial.println("Veuillez approchez votre oeil du capteur : ");
while (Serial.available() == 0) {} // Attendre l'entrée de l'utilisateur

```

```

// Lire la saisie de l'utilisateur
int index = 0;
while (Serial.available() > 0 && index < 7) {
    retine[index] = Serial.read();
    index++;
    delay(10); // Petite pause pour laisser le temps de saisie
}
retine[index] = '\0'; // Ajouter un terminateur nul à la fin

```

```

int modexp(int a, int e, int n) {
    long r;
    if (e < 0) {
        return -1;
    }
    if (a == 0 || n == 1) {
        return 0;
    }
    r = 1;
    while (e > 0) {
        r = (r * (a % n)) % n;
        e--;
    }
    return r;
}

void setup() {
    Serial.begin(9600); // Initialise la communication série
    randomSeed(analogRead(0)); // Initialise le générateur de nombres aléatoires
}

```

```

int ma3() {
    int etat = 0;
    char retine[8]; // Buffer pour stocker la saisie de l'utilisateur
    const char* retine_id[] = {
        "retineA", "retineB", "retineC", "retineD", "retineE", "retineF",
        "retineH", "retineI", "retineJ", "retineK", "retineL", "retineM",
        "retineN", "retineO", "retineP", "retineQ"
    };
    Serial.println("Veuillez approchez votre oeil du capteur : ");
    while (Serial.available() == 0) {} // Attendre l'entrée de l'utilisateur
}

```

```

// Lire la saisie de l'utilisateur
int index = 0;
while (Serial.available() > 0 && index < 7) {
    retine[index] = Serial.read();
    index++;
    delay(10); // Petite pause pour laisser le temps de saisie
}
retine[index] = '\0'; // Ajouter un terminateur nul à la fin

```

```

for (int cpt = 0; cpt < 16; cpt++) {
    if (strcmp(retine_id[cpt], retine) == 0) {
        // Supposons que `id` est une variable globale définie ailleurs
    }
}

```

```

if (cpt == id) {
    Serial.println("Scan retinien reconnu");
    etat = 1;
    break;
} else {
    Serial.println("[Erreur] >> Scan retinien non reconnu");
}
}
}
return etat;
}

```

```

int ma4() {
    int etat = 0;
    char doigt[8]; // Buffer pour stocker la saisie de l'utilisateur
    const char* doigt_id[] = {
        "doigtA", "doigtB", "doigtC", "doigtD", "doigtE", "doigtF",
        "retineH", "doigtI", "doigtJ", "doigtK", "doigtL", "doigtM",
        "doigtN", "doigtO", "doigtP", "doigtQ"
    };
    Serial.println("Veuillez approchez votre doigt du capteur : ");
    while (Serial.available() == 0) {} // Attendre l'entrée de l'utilisateur
}

```

```

// Lire la saisie de l'utilisateur
int index = 0;
while (Serial.available() > 0 && index < 7) {
    doigt[index] = Serial.read();
    index++;
    delay(10); // Petite pause pour laisser le temps de saisie
}
doigt[index] = '\0'; // Ajouter un terminateur nul à la fin

```

```

for (int cpt = 0; cpt < 16; cpt++) {
    if (strcmp(doigt_id[cpt], doigt) == 0) {
        // Supposons que `id` est une variable globale définie ailleurs
        if (cpt == id) {
            Serial.println("Scan digital reconnu");
            etat = 1;
            break;
        } else {
            Serial.println("[Erreur] >> Scan digital non reconnu");
            etat = 0;
        }
    }
}
return etat;
}

```

```

int ma5() {
    int groupe = -1;
    char aTrouve[6]; // Buffer pour stocker la saisie de l'utilisateur
    const char* codes[] = {
        "A1258", "B2865", "C1658", "D1100", "E0101", "F0110",
        "G0111", "H1000", "I1001", "K1010", "L1011", "M1100",
        "N1101", "O1835", "P8563", "Q7569"
    };
}

```

```

Serial.println("Entrez votre lettre et code d'agent");
while (Serial.available() == 0) {} // Attendre l'entrée de l'utilisateur

```

```

// Lire la saisie de l'utilisateur
int index = 0;
while (Serial.available() > 0 && index < 5) {
    aTrouve[index] = Serial.read();
    index++;
    delay(10); // Petite pause pour laisser le temps de saisie
}
aTrouve[index] = '\0'; // Ajouter un terminateur nul à la fin

```

```

for (int compteur = 0; compteur < 16; compteur++) {
    if (strcmp(codes[compteur], aTrouve) == 0) {
        groupe = compteur;
        // Supposons que `id` est une variable globale définie ailleurs
        id = groupe;
        break;
    }
}

```

```

if (groupe == -1) {
    Serial.println("Erreur");
}

```

```

return groupe;
}

```

```

void niveauSecurite(int grp) {
    bool success = false;
    switch (grp) {
        case 1:
            success = (ma1() == 1 && ma3() == 1);
            break;
        case 2:
            success = (ma1() == 1 && ma4() == 1);
            break;
        case 3:
            success = (ma2() == 1 && ma5() == 1);
            break;
        case 4:
            success = (ma2() == 1 && ma3() == 1 && ma4() == 1);
            break;
        case 5:
            success = (ma1() == 1 && ma2() == 1 && ma3() == 1 && ma5() == 1);
            break;
        default:
            success = false;
    }

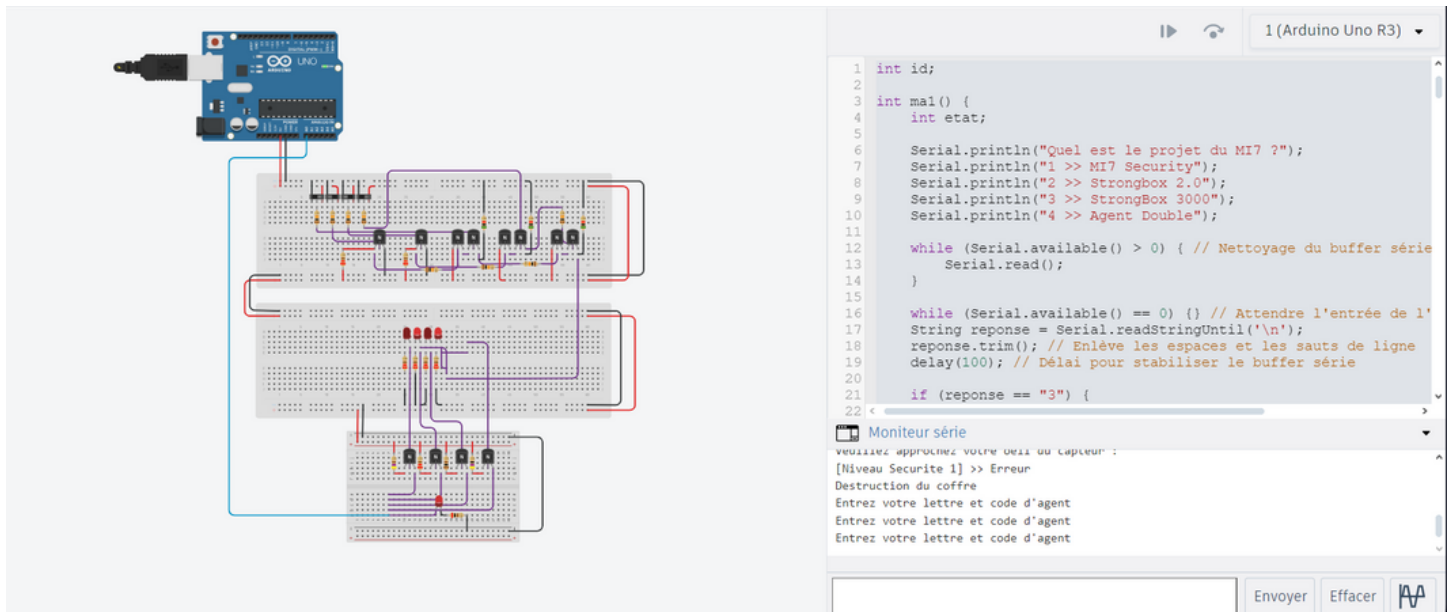
    if (success) {
        Serial.println("[Niveau Securite " + String(grp) + "] >> Reussi");
        Serial.println("Coffre Ouvert !");
    } else {
        Serial.println("[Niveau Securite " + String(grp) + "] >> Erreur");
        Serial.println("Destruction du coffre");
    }
}

void loop() {
    int grp = ma5();
    if (grp < 3) {
        niveauSecurite(1);
    } else if (grp < 6) {
        niveauSecurite(2);
    } else if (grp < 9) {
        niveauSecurite(3);
    } else if (grp < 12) {
        niveauSecurite(4);
    } else if (grp < 16) {
        niveauSecurite(5);
    }

    delay(10000); // Delai pour eviter une execution trop rapide
}

```


Nous implémentons le code dans notre circuit Tinkercad.



Lorsque nous commençons la simulation, nos LEDs s'allument pour la bonne combinaison 1001. En sortie, nous avons 0101. Nous avons rencontré des problèmes de gestion de RAM Donc, cette partie fonctionne. Ensuite, dans la barre moniteur série, notre programme du coffre-fort se lance et nous avons accès aux différents niveaux de sécurité.

Conclusion

Nous pensons donc avoir réussi notre projet car notre implémentation du code de la carte sur Tinkercad à l'aide de la carte Arduino fonctionne à merveille après de nombreux tests. Nous avons donc terminé notre projet STRONGBOX 3000.

Perspectives

Nous avons bien travaillé tous ensemble sur ce livrable. On s'est tous partagé les tâches, puis on a mis en commun. A travers ce livrable, nous avons pu apprendre à mieux utiliser Flowgorithm de façon efficace et compréhensible.

Axel : Notre travail de groupe sur l'implémentation du code du coffret sur Tinkercad a été une expérience enrichissante. Chacun d'entre nous a apporté des compétences spécifiques à la table, ce qui a grandement contribué à la qualité globale du projet. J'ai apprécié la façon dont nous avons pu résoudre les défis techniques ensemble et créer un produit final solide.

Deshani: Ce livrable sur l'implémentation du code du coffre-fort a été un défi passionnant. Le travail d'équipe a été essentiel pour le succès du projet. Chacun d'entre nous a assumé des rôles spécifiques et a travaillé de manière synergique pour élaborer un coffre-fort sophistiqué.

Tassadit: Travailler sur l'implémentation du coffre-fort dans le cadre de notre livrable a été une expérience gratifiante. Notre groupe a fait preuve d'une grande collaboration et de compétences techniques pour concevoir un programme complexe. J'ai particulièrement apprécié la manière dont nous avons résolu les problèmes ensemble, en combinant nos connaissances pour atteindre nos objectifs.

Jack-Laurence: Notre livrable sur l'implémentation du coffre-fort sur Tinkercad a été un excellent exemple de travail d'équipe et d'ingéniosité. Malgré les défis techniques auxquels nous avons été confrontés, notre groupe a travaillé de manière efficace et a réussi à produire un produit final de haute qualité. Je suis reconnaissant d'avoir eu l'opportunité de travailler avec des collègues aussi talentueux.

Bibliographie

- PROSIT UNLOCKED
- PROSIT AGENT DOUBLE
- WORKSHOP ALGORITHMIQUE
- Connaissance de Axel qui a déjà codé en C

Lien du Tinkercad: https://www.tinkercad.com/things/97IPFfJIPkg-powerful-vihelmo-gogo/editel?sharecode=-XdNHXifpLwo7ab4ATeqpbSPCq_-20qrGxx4MOmqJNk