

MINI_PROJET COFFRE FORT SECURITY

MISE EN SITUATION

Une équipe de technicien souhaite mettre à disposition pour des banques un coffre-fort nouvelle génération avec un mécanisme de sécurité augmenté. Le prototype doit être finalisé et réalisé dans nos salles de FABLAB :

- Une carte électronique devra être réalisé afin d'activer le programme d'ouverture du coffre. Cette carte contient un numéro d'identification que nous appellerons Proto_ID sous forme de codes d'entrée.
- Une série d'interrupteurs devront être positionnés qui déclenchera la bonne combinaison d'ouverture, ceci permettra de déterminer à l'avance le code binaire correct correspondant à la carte électronique désignée par le responsable du FABLAB.
- Le bon code activera une combinaison parmi les résistances définies par le cahier des charges (2 sorties). A chaque carte correspondra une valeur précise pour la somme des résistances activées (voir tableau des Résistances).
- Une 3^{ème} résistance sera déterminée pour chaque groupe afin d'obtenir à ses bornes une tension d'activation de 3,3v (+/- 0,15v).
- Une fois la carte validée, un programme interne à un microcontrôleur (Arduino) devient disponible. Ce programme permettra plusieurs actions :
 - o La première – obligatoire – ouverture du coffre via un servo-moteur avec une rotation de 90° sinon maintien de la position à 0° **fermeture du coffre**.
 - o Une seconde déclenchera l'allumage d'une LED verte si la carte est valide ou une LED rouge si carte invalide.
 - o Une troisième affichera le message code correct ou incorrect sur un écran LCD avec la valeur de la tension d'entrée affichée.
 - o La dernière correspondra à une sirène de 2 secondes (Buzzer) lorsque la combinaison est incorrecte.
- A ces codes peuvent être associés des mécanismes complémentaires comme de la reconnaissance digitale, rétinienne ou vocale selon le niveau de sécurité associé à la carte, facultatif.

Avant de fabriquer en série le coffre-fort, une maquette simulant le fonctionnement du système du coffre sera fabriquée.

MINI_PROJET COFFRE FORT SECURITY

Travail à réaliser :

- 1- Réaliser les simplifications logiques des solutions R1, R2 en fonction des entrées E1, E2, E3 et E4 et de la table de vérité suivante. Utiliser les tableaux de Karnaugh pour chaque sortie. Vous tiendrez compte des combinaisons qui déclencheront les résistances de sortie souhaitées en fonction du choix du responsable FABLAB : 220Ω, 330Ω.

1- Table de vérité :

Entrées (Interrupteurs)				Résistance série souhaitée		R_Totale
E4	E3	E2	E1	R1 (220 ohms)	R2 (330 ohms)	
0	0	0	0	0	1	330Ω
0	0	0	1	1	0	220Ω
0	0	1	0	0	1	330Ω
0	0	1	1	0	1	330Ω
0	1	0	0	0	1	330Ω
0	1	0	1	1	0	220Ω
0	1	1	0	0	1	330Ω
0	1	1	1	1	0	220Ω
1	0	0	0	1	0	220Ω
1	0	0	1	1	0	220Ω
1	0	1	0	0	1	330Ω
1	0	1	1	1	0	220Ω
1	1	0	0	1	1	550Ω
1	1	0	1	1	0	220Ω
1	1	1	0	0	1	330Ω
1	1	1	1	0	1	330Ω

MINI_PROJET COFFRE FORT SECURITY

Tableau de Karnaugh pour solution R1

E2E1 E4E3	00	01	11	10
00	0	1	0	0
01	0	1	1	0
11	1	1	0	0
10	1	1	1	0

Equation logique simplifiée

$$S_{R1} = E1./E2 + /E2.E4 + E1./E3.E4 + E1.E3./E4$$

Schéma logique de R1

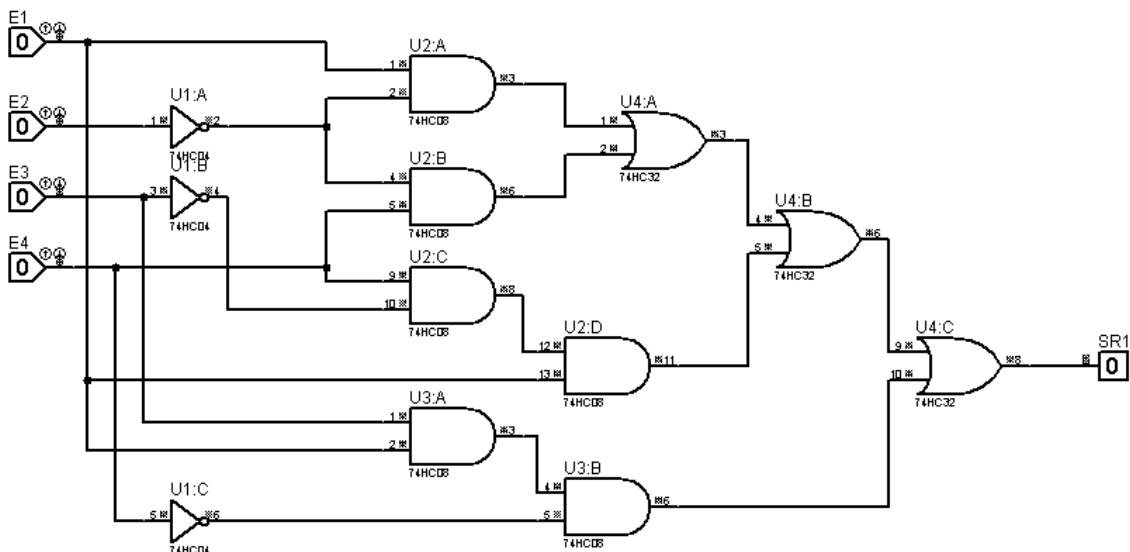


Tableau de Karnaugh pour solution R2

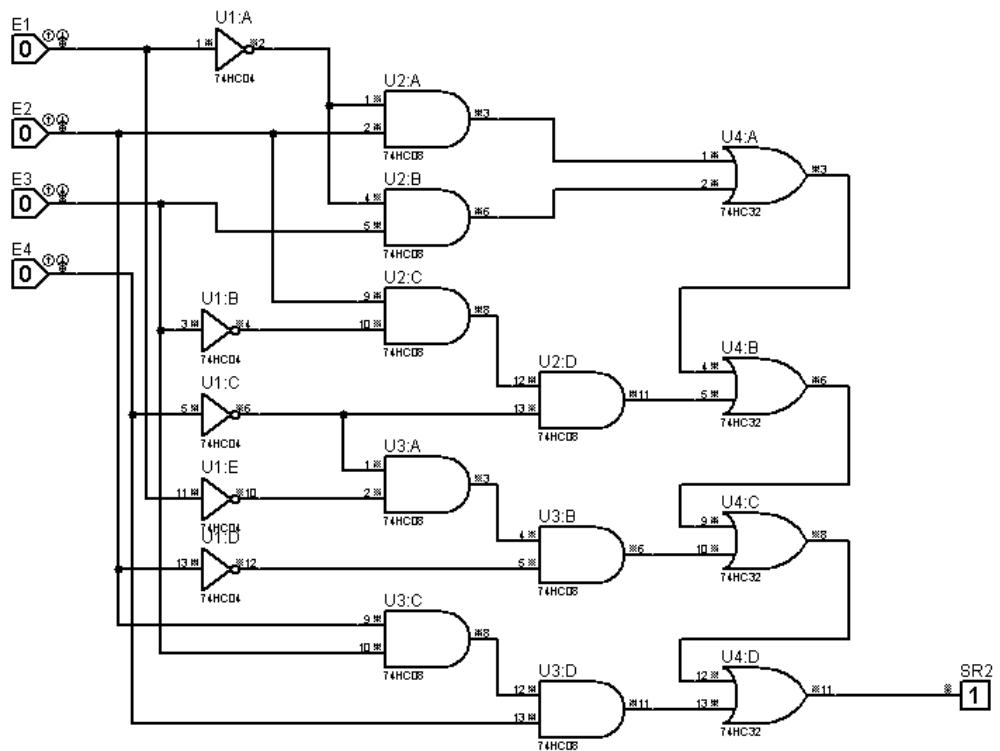
E2E1 E4E3	00	01	11	10
00	1	0	1	1
01	1	0	0	1
11	1	0	1	1
10	0	0	0	1

Equation logique simplifiée

$$S_{R1} = /E1. E2 + /E1.E3 + E2./E3./E4 + /E1./E2./E4 + E2.E3.E4$$

MINI_PROJET COFFRE FORT SECURITY

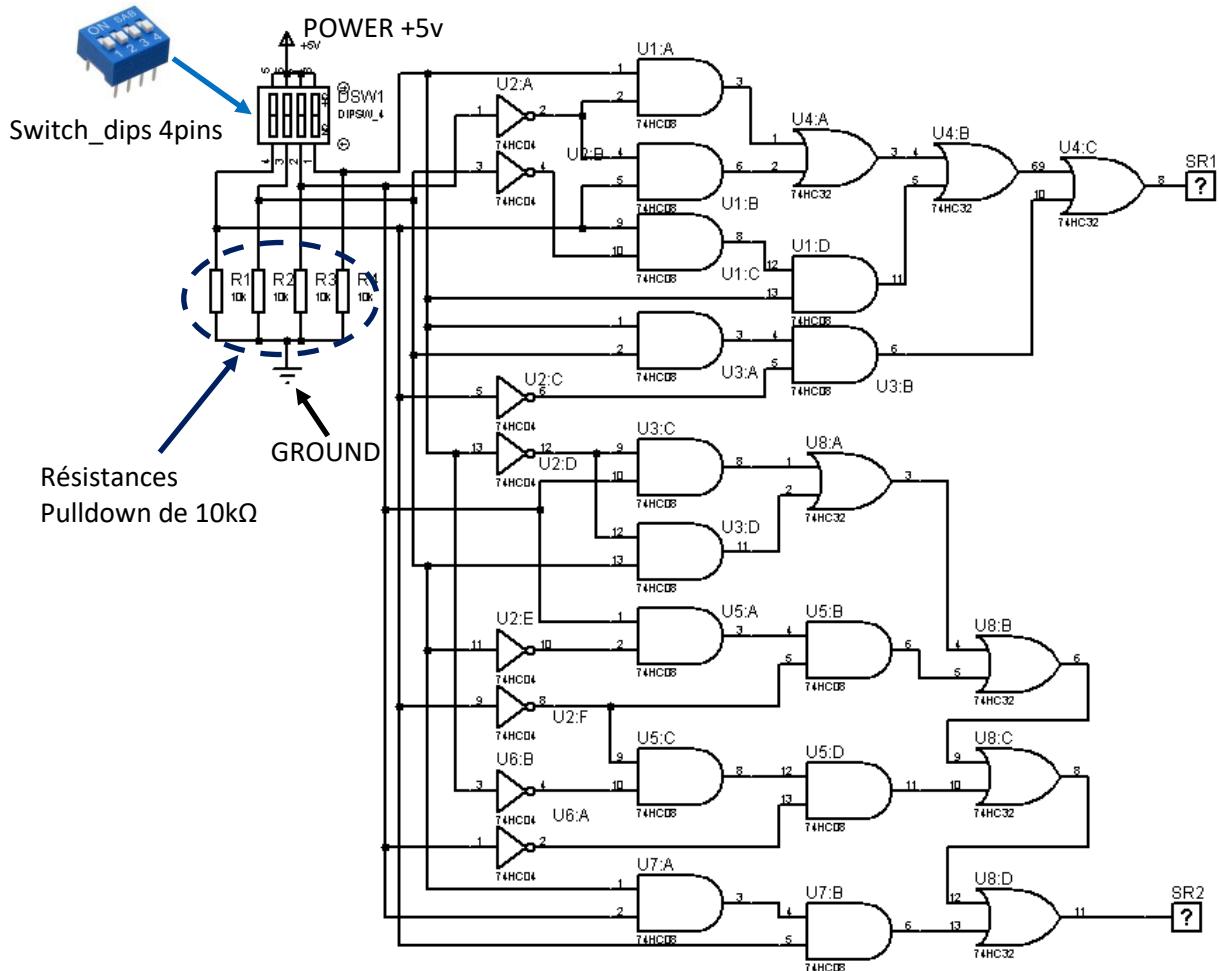
Schéma logique de R2



Pour simuler le bon fonctionnement des 2 structures logiques, on peut utiliser des LOGICSTATE pour les variables d'entrée et un LOGICPROBE (BIG) pour voir la solution en sortie.

MINI_PROJET COFFRE FORT SECURITY

- 2- Représenter le circuit logique en fonction de la résistance de sortie imposée par l'enseignant (ce sera votre PROTO_ID).



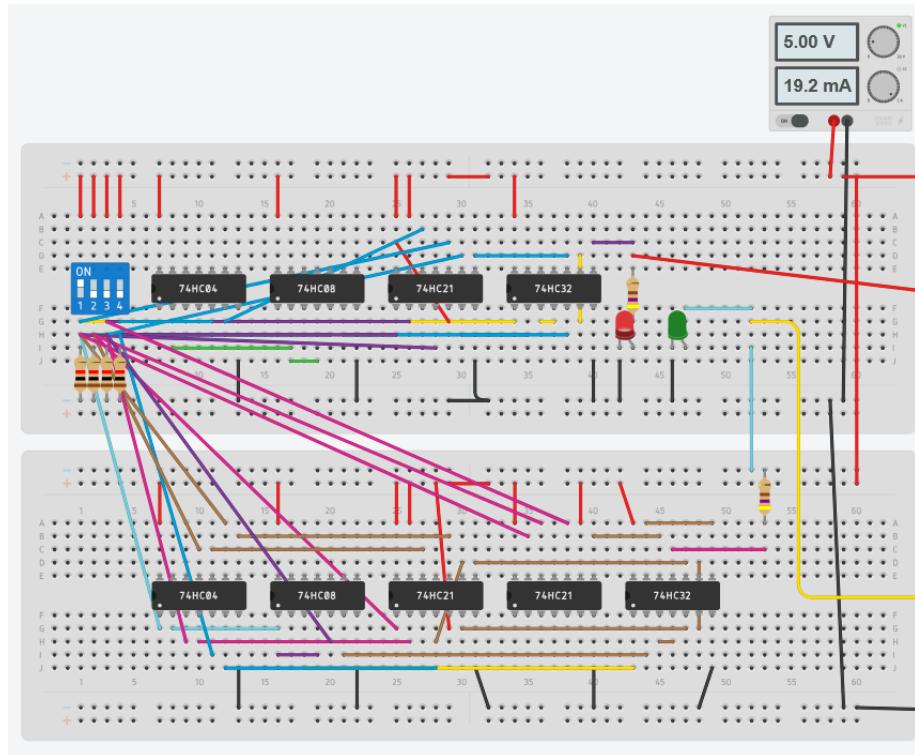
- 3- Afin de pouvoir tester l'efficacité des cartes produites, un environnement sera mis en place de manière simplifiée pour valider la reconnaissance des cartes et l'activation des résistances correctes. Pour faciliter la visualisation de l'activation des bonnes résistances vous placerez des DIPS SWITCHS 4 entrées qui simuleront les 4 variables E1 à E4, et 2 LED de couleurs différentes qui simuleront les 2 résistances de sortie R1 à R2 (***Ne pas oublier de rajouter une résistance en série à chaque LED de 470Ω pour limiter le courant.***).

Vous utiliserez les circuits suivants pour faire vos montages sur BreadBord.



- Porte NON (74HC04)
- Porte ET (74HC08)
- Porte OU (74HC32)

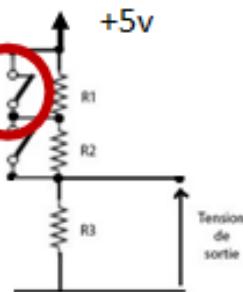
MINI_PROJET COFFRE FORT SECURITY



- 4- Montrer à l'enseignant le bon fonctionnement de votre montage réel et/ou par Tinkercad.

MINI_PROJET COFFRE FORT SECURITY

Après avoir conçu votre prototype logique, il vous est demandé de réaliser les structures qui commanderont l'activation de la ou les résistances R1 et R2.



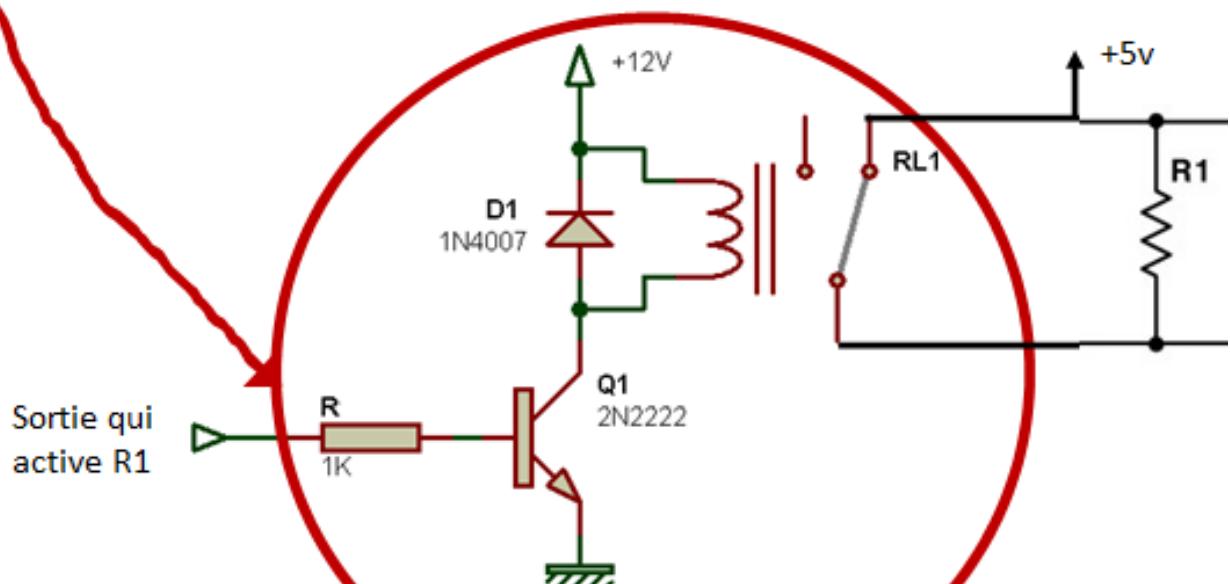
Le contact fermé est réalisé par la structure ci-dessous

Les contacts qui court-circuissent les résistances seront réalisés par des relais pilotés par des transistors type NPN (référence 2N2222A) en commutation Q1.

Lorsque le relais est soumis à une tension (sortie d'une structure logique), le contact sera activé, qui prendra en compte la résistance sélectionnée.

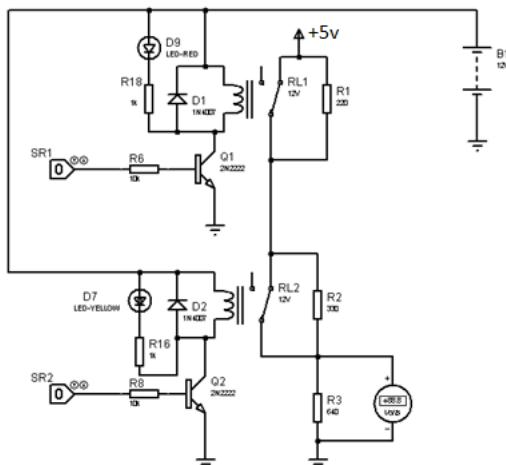
Au repos les contacts sont fermés, ce qui court-circuite la résistance.

Voici une structure qui active la résistance R1.



MINI_PROJET COFFRE FORT SECURITY

- 5- Réaliser les structures autour des 2 relais qui activeront les résistances séries R1 et R2 pour compléter votre montage.



- 6- Il vous est demandé de dimensionner la résistance (R3) afin d'obtenir une tension à ses bornes de 3,3V (+ ou - 0,15v) pour votre modèle de carte imposé par l'enseignant du FABLAB.
Pour compléter, vous donnerez sous forme de tableau pour les 3 cartes logiques, les tensions obtenues en sortie avec cette même valeur de résistance R3.

Avec R1 activée

$$VR3 = \frac{R3}{R1 + R3} * 5$$

$$R3 = \frac{220}{0,5151} = 427 \Omega \text{ valeur normalisée } 470 \Omega$$

Avec R2 activée

$$VR3 = \frac{R3}{R1 + R3} * 5$$

$$R3 = \frac{330}{0,5151} = 640 \Omega \text{ valeur normalisée } 680 \Omega$$

Avec R1 et R2 activées

$$R3 = \frac{550}{0,5151} = 1068 \Omega$$

Tableau des tensions

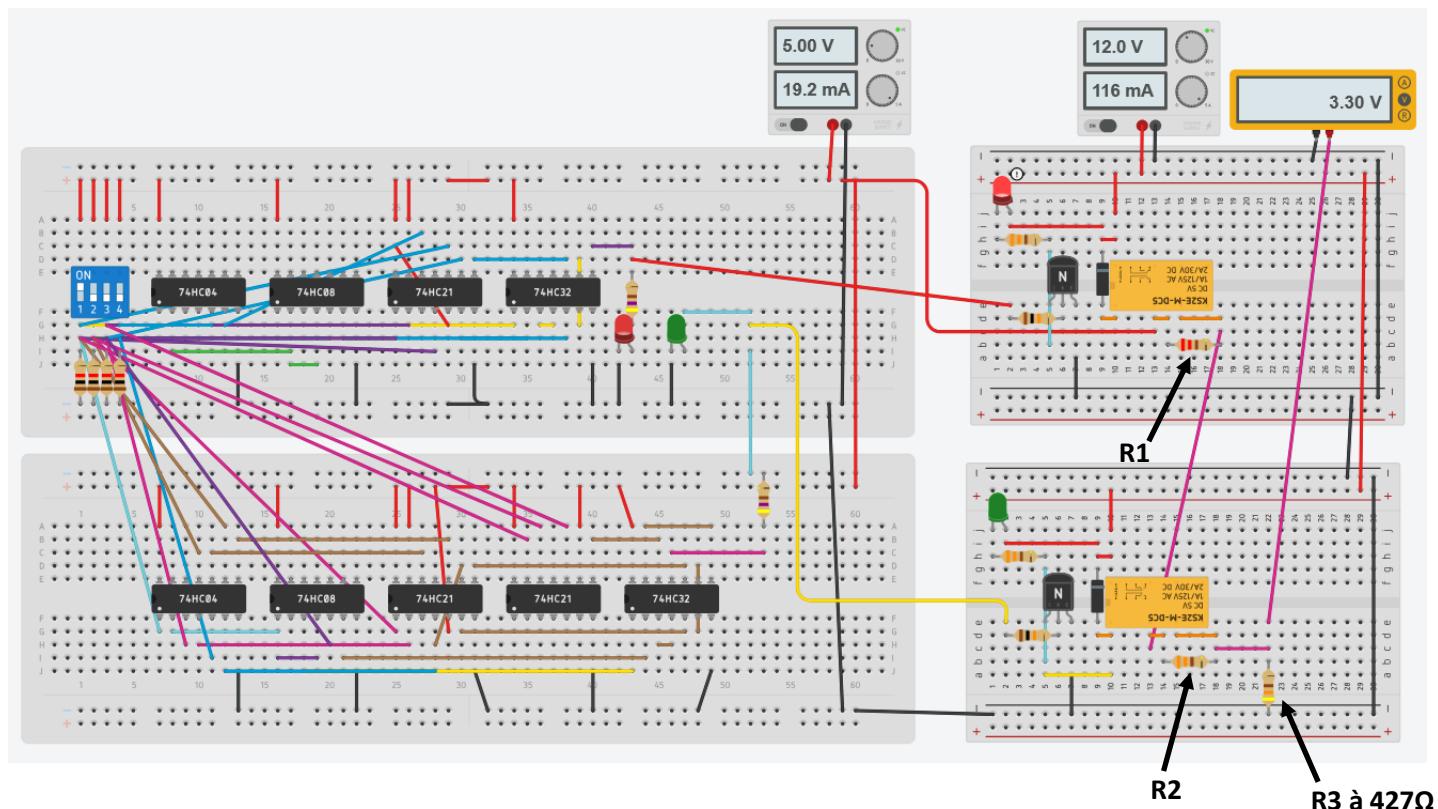
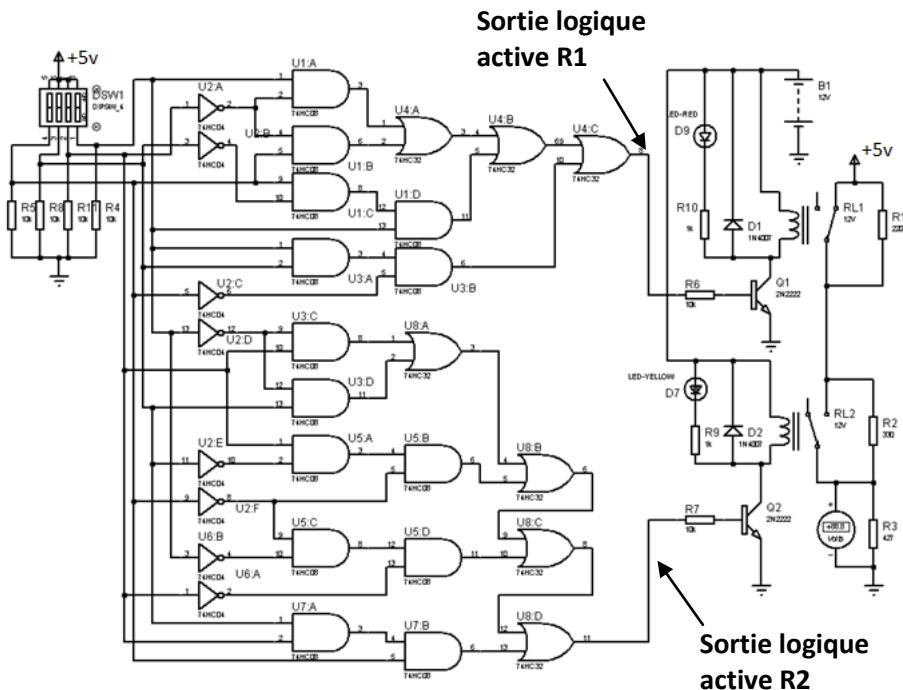
R3	Résistances actives		
	R1	R2	R1 + R2
427 Ω	3,3v	2,82v	2,19v
640 Ω	3,72v	3,3v	2,69v
1068 Ω	4,15v	3,82v	3,3v

MINI_PROJET COFFRE FORT

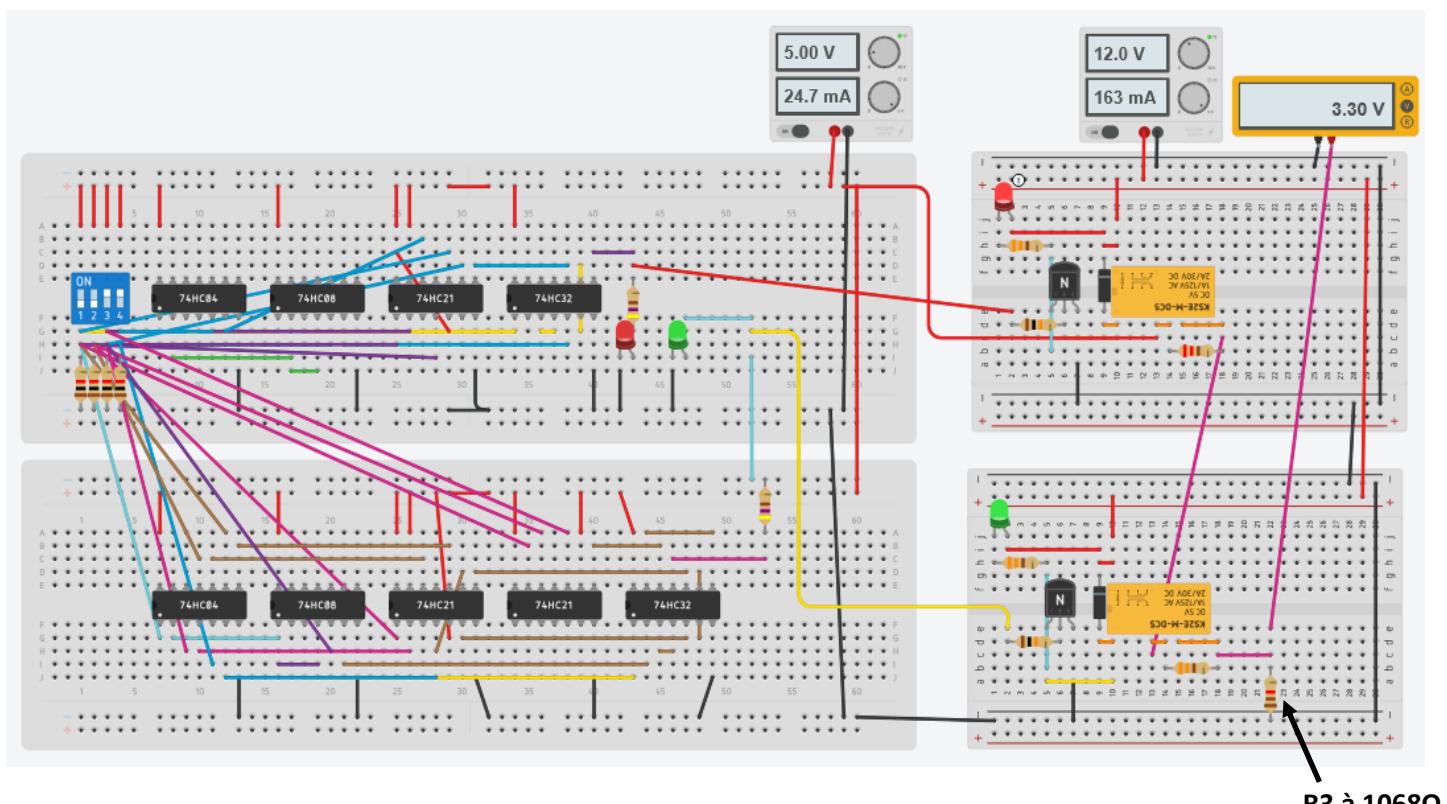
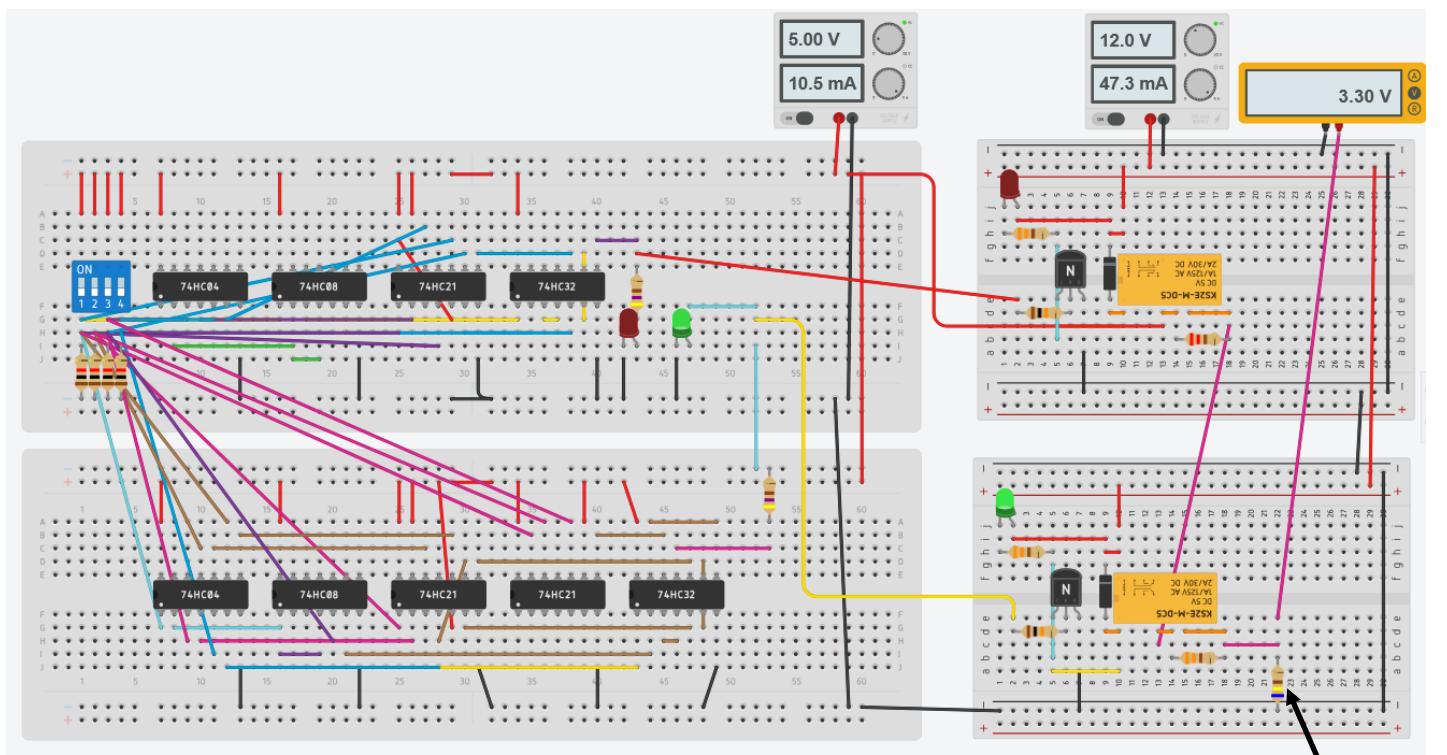
SECURITY

7- Réalisez votre montage complet par simulation en utilisant PROTEUS. Placer un voltmètre aux bornes de R3 et vérifier les tensions en fonction des combinaisons d'entrées.

Faites valider par l'enseignant



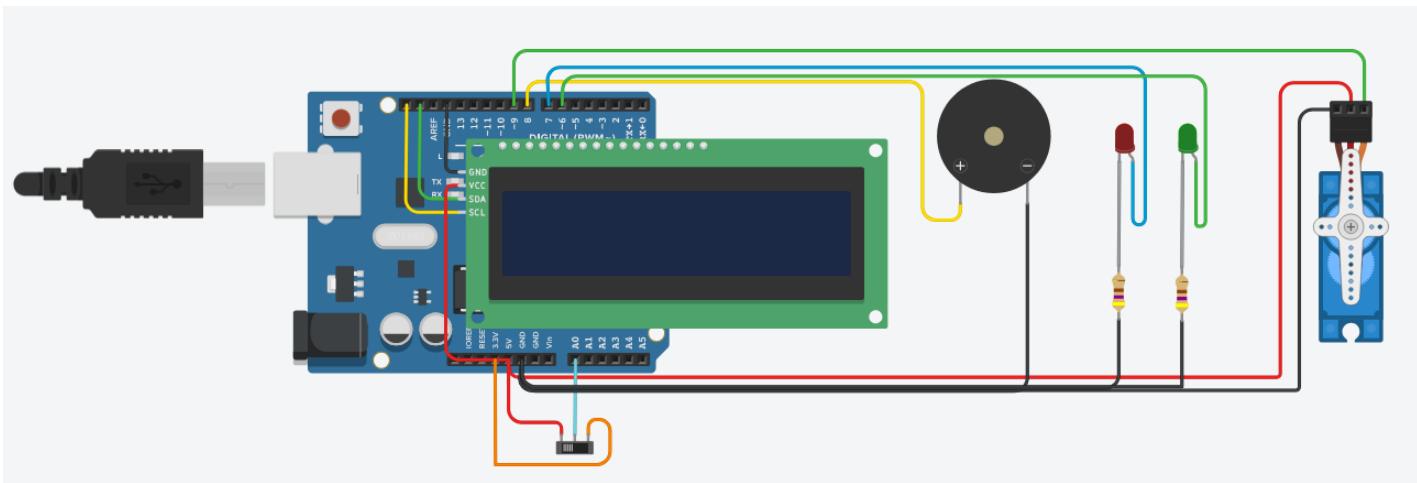
MINI_PROJET COFFRE FORT SECURITY



MINI_PROJET COFFRE FORT SECURITY

- 8- Maintenant que les prototypes logiques ont démontré que la carte pouvait être utilisée pour activer le coffre-fort, il vous reste à concevoir le programme du système interne. La tension issue de la résistance R3 doit être appliquée en une des entrées analogiques de la carte ARDUINO (**exemple A0**). Vous devez maintenant réaliser le programme en fonction de la condition de la tension de 3,3v aux bornes de R3.
- Vous afficherez le message code correct pour votre PROTO_ID et code incorrect pour les autres en utilisant un afficheur LCD (2 lignes 16 segments).
 - Une rotation de 90 degrés de votre servomoteur simulera l'ouverture du coffre pour votre carte.
 - L'affichage d'une LED verte qui simule code correct.
 - Une alarme de 2 secondes (Buzzer) et un voyant lumineux (LED rouge) seront déclenchés lors d'une carte incorrecte (tension autres que 3,3v). Montrer votre final à l'enseignant.

- **Avec un écran LCD branchement série I²C**



Programme coffre :

```
// C++ code
//
#include <Adafruit_LiquidCrystal.h>
#include <Servo.h>

int posInitiale = 0;
int Lrouge = 7;
int Lverte = 6;
int buzzer = 8;
```

MINI_PROJET COFFRE FORT SECURITY

```
Servo servo;

#define NOTE_C4 262
#define NOTE_G3 196

int melody[] = {NOTE_C4, NOTE_G3};

int noteDurations[] = {4, 4};
int seconds = 0;

Adafruit_LiquidCrystal lcd_1(0);

void setup()
{
    lcd_1.begin(16, 2);

    servo.attach(9);
    servo.write(0);
    pinMode(Lrouge, OUTPUT);
    pinMode(Lverte, OUTPUT);
    digitalWrite(Lrouge, LOW);
    digitalWrite(Lverte, LOW);
    pinMode(buzzer, OUTPUT);
}

void loop()
{
    Serial.begin(9600);
    float val = analogRead(A0);
    val=val*1000.0;
    float valIntensite = map(val,0.0,1023.0,0.0,5.0);
    valIntensite = valIntensite/1000.0;

    if ((valIntensite>=3.1)&&(valIntensite<=3.5)){
        lcd_1.clear();
        lcd_1.print("Code correct ! ");

        servo.write(90);
        digitalWrite(Lrouge, LOW);
        digitalWrite(Lverte, HIGH);
    }
}
```

MINI_PROJET COFFRE FORT

SECURITY

```

else {
    lcd_1.clear();
    lcd_1.print("Code incorrect ! ");

    servo.write(0);

    for (int thisNote = 0; thisNote < 2; thisNote++) {
        int noteDuration = 1000 / noteDurations[thisNote];
        tone(8, melody[thisNote], noteDuration);
        int pauseBetweenNotes = noteDuration * 1.30;
        delay(pauseBetweenNotes);
        noTone(8);
    }

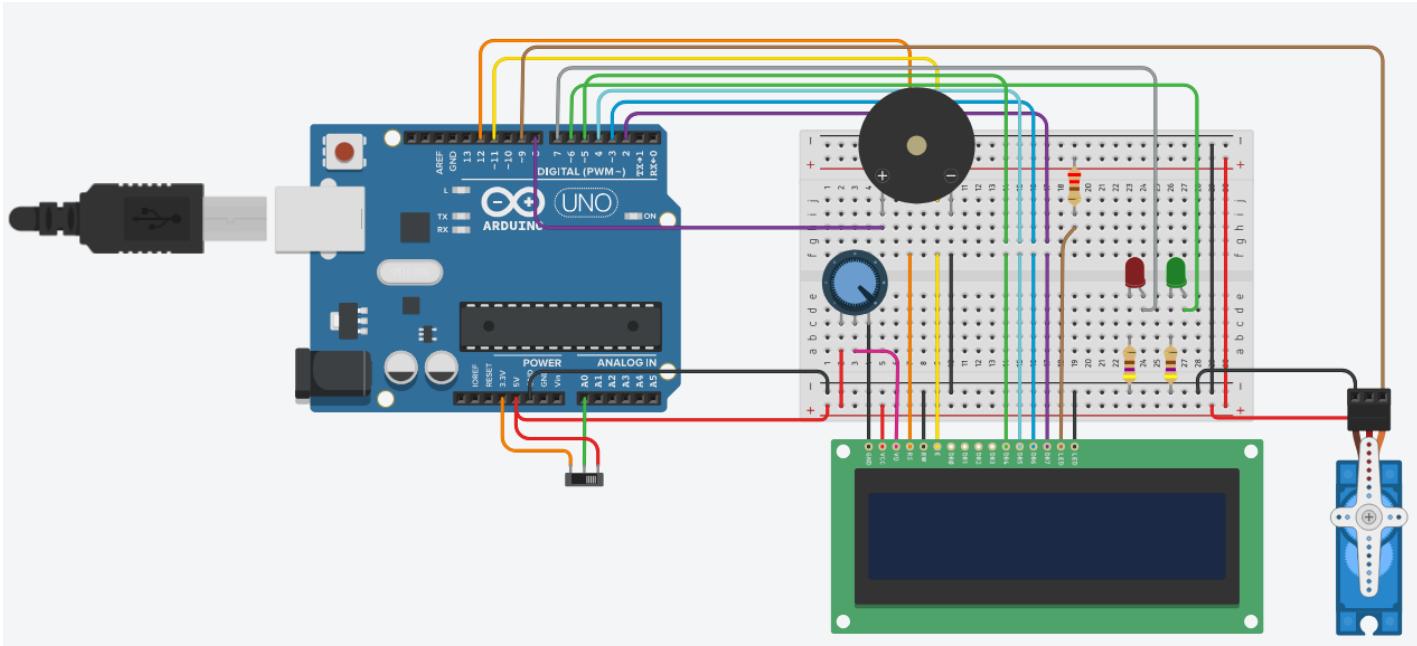
    digitalWrite(Lverte, LOW);
    digitalWrite(Lrouge, HIGH);
}

lcd_1.setCursor(0,1);
lcd_1.print(vallIntensite);
delay(100);

}

```

- Avec un écran LCD branchement parallèle



MINI_PROJET COFFRE FORT SECURITY

```
#include <LiquidCrystal.h>

#include <Servo.h>

int posInitiale = 0;
int Lrouge = 7;
int Lverte = 6;
int buzzer = 8;

Servo servo;

#define NOTE_C4 262
#define NOTE_G3 196

int melody[] = {NOTE_C4, NOTE_G3};

int noteDurations[] = {4, 4};

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);

void setup() {

lcd.begin(16, 2);
Serial.begin(9600);
servo.attach(9);
servo.write(0);
pinMode(Lrouge, OUTPUT);
pinMode(Lverte, OUTPUT);
digitalWrite(Lrouge, LOW);
digitalWrite(Lverte, LOW);

pinMode(buzzer, OUTPUT);
}

void loop() {
Serial.begin(9600);
float val = analogRead(A0);
val=val*1000.0;
```

MINI_PROJET COFFRE FORT SECURITY

```

float valIntensite = map(val,0.0,1023.0,0.0,5.0);
valIntensite = valIntensite/1000.0;

if ((valIntensite>=3.1)&&(valIntensite<=3.5)){
    lcd.clear();
    lcd.print("Code correct ! ");

    servo.write(90);
    digitalWrite(Lrouge, LOW);
    digitalWrite(Lverte, HIGH);
}
else {
    lcd.clear();
    lcd.print("Code incorrect ! ");

    servo.write(0);

for (int thisNote = 0; thisNote < 2; thisNote++) {
    int noteDuration = 1000 / noteDurations[thisNote];
    tone(8, melody[thisNote], noteDuration);
    int pauseBetweenNotes = noteDuration * 1.30;
    delay(pauseBetweenNotes);
    noTone(8);
}

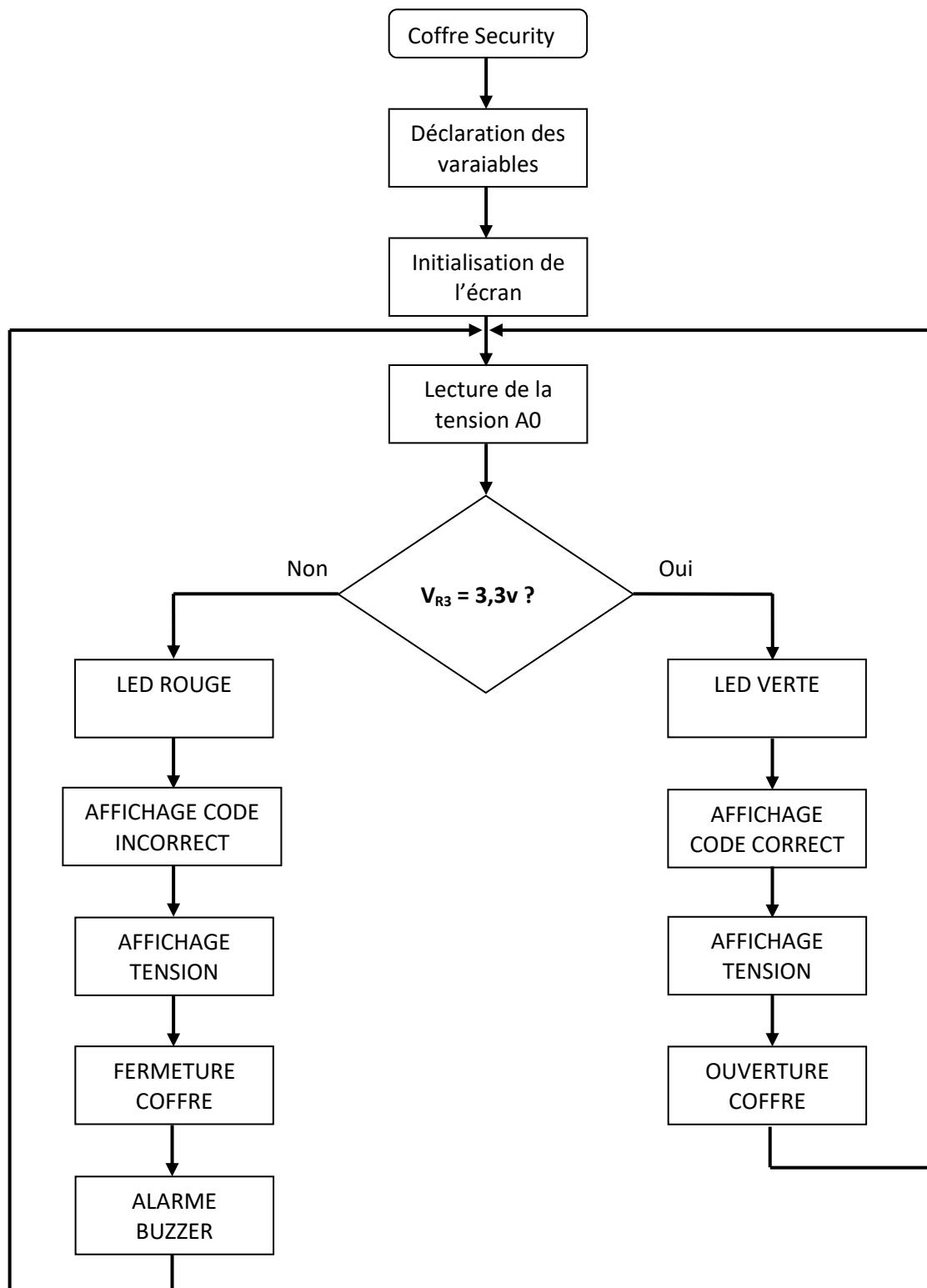
digitalWrite(Lverte, LOW);
digitalWrite(Lrouge, HIGH);
}

lcd.setCursor(0,1);
lcd.print(valIntensite);
delay(100);
}

```

MINI_PROJET COFFRE FORT SECURITY

9- Représenter votre programme en langage logigramme.



MINI_PROJET COFFRE FORT SECURITY

Barème de notation :

- 8 points pour la réalisation de votre prototype électronique numérique fonctionnel.
- 6 points pour la réalisation de votre programme Arduino.
- 6 points pour votre compte rendu (Introduction, explications, simplification logique, schéma logique, simulation Proteus, détermination de R3, tableau des tensions de sortie, logigramme, programme et conclusion...)