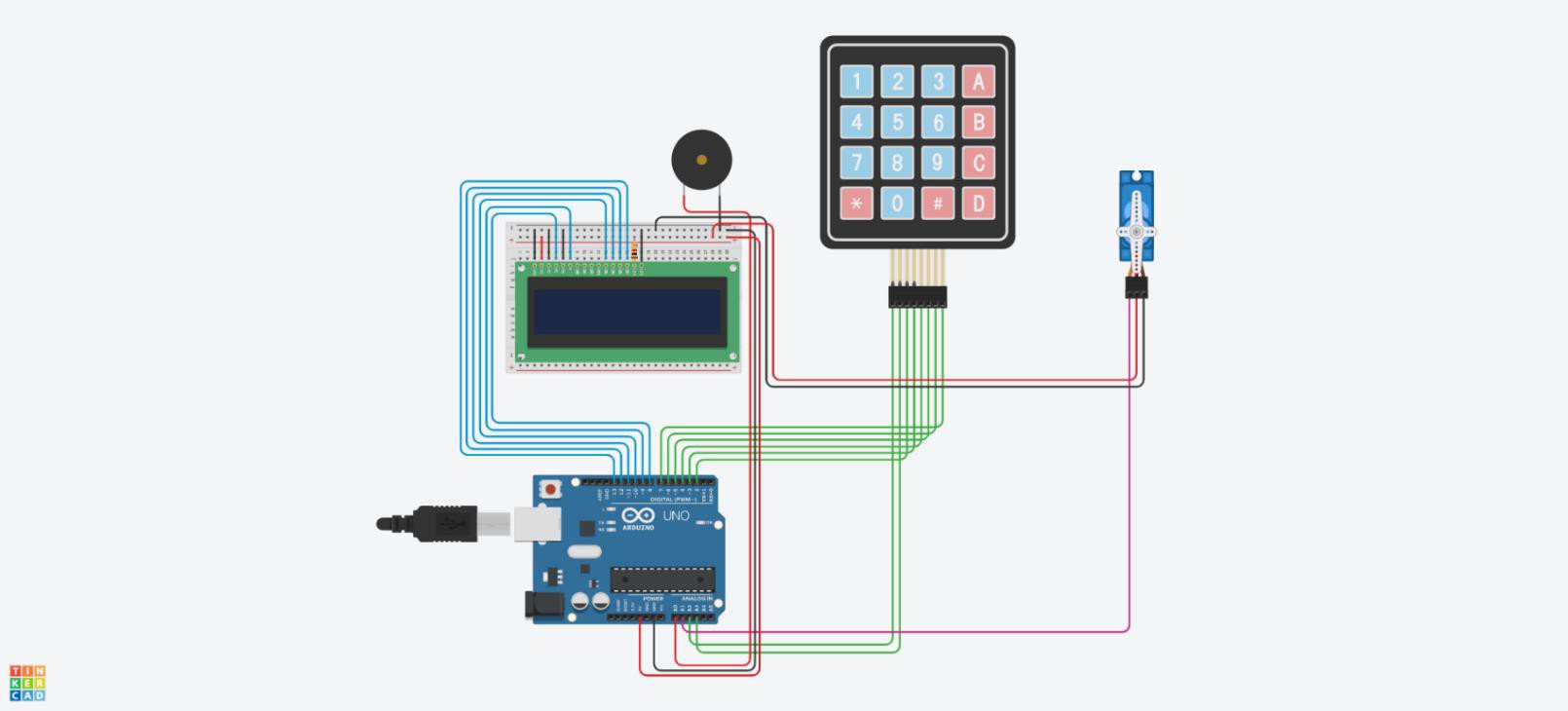
# Travail à faire

***Q1.*** *Donnez le nom des différents composants et le fonctionnement de ceux-ci.*

|  |  |  |
| --- | --- | --- |
| *Nom du*  *composant* | *Image du composant* | *Fonctionnement* |
| *Carte arduino* |  | *Gestion électronique des composants* |
| *Servo moteur* |  | *maintenir une opposition à un effort statique et dont la position est vérifiée en continu et corrigée en fonction de la mesure.* |
| *Buzzer* |  | *Alerte Sonore* |
| *Digicode* |  | *Permet d’entrer des codes dans l’arduino* |
| *Lecteur LCD* |  | *Interface homme machine* |

***Q2.*** *Raccordez les composants avec les câbles :*

* D’alimentation en rouge
* De masse en noir
* D’information rentrant en vert
* D’information sortante en jaune



***Q3.*** *Créer l'algorithme votre propre système en vous appuyant sur le cours Algorigramme*

*Evènement Action*

*Ajout de badge Ferme le servo Code bon Ouvre le servo*

*Code faux Buzzer Réinitialisation*



Début

Passer le badge

Délai d’ouverture

OUI

NON X 3

Code bon

Ferme le servo

Ouvre le servo

Buzzer sonne

Ferme le servo

Réinitialisation code

***Q4.*** *Commentez le code de votre programme dès que nécessaire.*

Rappel : il suffit de rajouter **//** avant d’ajouter votre commentaire Exemple : #include <Servo.h> //ajoute la bibliothèque Servo au code #include <LiquidCrystal.h> **//ajoute la bibliothèque LCD**

#include <Keypad.h> **//ajoute la bibliothèque pavé numérique**

#include <Servo.h> **//Ajoute la bibliothèque servo**

LiquidCrystal LCD( 8, 9, 10, 11, 12, 13 ) ; **//Assigne pin LCD**

Servo servo ; **// Renomme servo**

const int buzzerPin = 14 ; **//Assigne pin buzzer**

const int servoPin = 15 ; **//Assigne pin servo**

const byte ROWS = 4, COLS = 4 ; **// Donne le nombre de lignes et de colonnes**

char keys[ROWS][COLS] =

{

{'1','2','3','A'},

{'4','5','6','B'},

{'7','8','9','C'},

{'\*','0','#','D'}

# } ; //Assigne les touches

byte rowPins[ROWS] = { 16, 17, 2, 3 } ;

byte colPins[COLS] = { 4, 5, 6, 7 } ; **//Assigne les pin du pavé num**

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

char password[16], string[16] ;

int flag\_h\_setpassword = 1, flag\_inputpassword = 0, flag\_inputstring = 0, flag\_opendoor = 1, flag\_state = 0, flag\_remoteopen = 0, flag\_lockdown = 0 ;

int count = 0, trial\_count=0, pos = 0, state = 0 ;

# // flag\_setpassword = flag for setting the password,

**// flag\_stringinput = flag for taking input the string,**

void setup() {

for(int k=8 ; k<14 ; k++) { pinMode(k,OUTPUT) ;

}

LCD.begin(16, 2) ;

pinMode(buzzerPin, OUTPUT) ; pinMode(servoPin, OUTPUT) ; servo.attach(servoPin) ;

# // for Bluetooth-Module

Serial.begin(9600) ; **// Assigne la vitesse de communication**

LCD.setCursor(0,0) ; **// Donne la zone d’écriture** LCD.print(" WELCOME !!") ; **//Ecrit le texte** LCD.setCursor(0,1) ;

LCD.print("Set a Password :") ;

InitializePassword(), InitializeString() ; **//initialise le code**

CloseDoor() ; **// Ferme le porte**

}

void loop() {

# //Keypad

if( trial\_count < 3 ) {

char key = kpd.getKey() ; **//storing pressed key value in a char**

if( key != NO\_KEY ) {

if( flag\_h\_setpassword == 1 ) { H\_SetPassword() ;

}

if( key == '#' ) {

if( flag\_inputpassword == 1 && count > 0 ) { flag\_inputpassword = 0 ;

password[count] = '\0' ; H\_EnterPassword() ;

}

else if( flag\_inputstring == 1 && count > 0 ) { flag\_inputstring = 0 ;

string[count] = '\0' ;

if( Compare\_Password\_and\_String() == 1 ) { LCD.clear() ;

LCD.print(" VERIFIED ! ") ;

Serial.println("UNLOCKED") ; trial\_count = 0 ;

for( int i=0 ; i<1 ; ++i ) {

tone(buzzerPin, 500, 100) ; **//le buzzer sonne**

delay(230) ; tone(buzzerPin, 100, 100) ; delay(230) ;

OpenDoor() ; delay(4000) ; CloseDoor();

InitializeString(), H\_EnterPassword() ;

}

}

else {

LCD.clear() ;

LCD.print("Wrong Password !") ; delay(1000) ;

Serial.println("Someone unsuccessfully attempted to open the lock !") ;

++trial\_count ; tone(buzzerPin, 100, 1000) ; delay(1000) ; H\_EnterPassword() ;

}

}

}

else if( flag\_inputpassword == 1 || flag\_inputstring == 1 ) { LCD.print(key) ;

delay(100) ; LCD.setCursor(count,1) ; LCD.print('\*') ;

if( flag\_inputpassword == 1 ) password[count] = key ; else if( flag\_inputstring == 1 ) string[count] = key ;

++count ;

}

}

}

else {

LCD.clear() ; LCD.setCursor(0,0) ;

LCD.print("SYSTEM LOCKDOWN!") ;

tone(buzzerPin, 1000, 1000) ; delay(1500) ; flag\_lockdown = 1 ;

}

}

void InitializePassword() { for( int i=0 ; i<16 ; ++i )

password[i] = 0 ;

}

void InitializeString() {

for( int i=0 ; i<16 ; ++i ) string[i] = 1 ;

}

void H\_SetPassword() { LCD.clear() ; LCD.setCursor(0,0) ; LCD.print("Set a Password :") ; LCD.setCursor(0,1) ; flag\_h\_setpassword = 0 ;

flag\_inputpassword = 1, count = 0 ;

}

void H\_EnterPassword() { CloseDoor() ; LCD.clear() ; LCD.setCursor(0,0) ;

LCD.print("Enter Password :") ; LCD.setCursor(0,1) ; flag\_inputstring = 1, count = 0 ;

}

int Compare\_Password\_and\_String() { int i ;

for( i=0 ; password[i]!='\0' && string[i]!='\0' ; ++i ) { if( password[i] != string[i] )

return 0 ;

}

if( password[i] == '\0' && string[i] == '\0' ) return 1 ;

else return 0 ;

}

void OpenDoor() {

if( flag\_opendoor == 1 ) return;

for( pos=15 ; pos<=100 ; ++pos ) { servo.write(pos) ;

delay(15) ;

}

flag\_opendoor = 1 ;

}

void CloseDoor() {

if( flag\_opendoor == 0 ) return;

for( pos=100 ; pos>=15; --pos ) { servo.write(pos) ;

delay(15) ;

}

flag\_opendoor = 0 ;

}