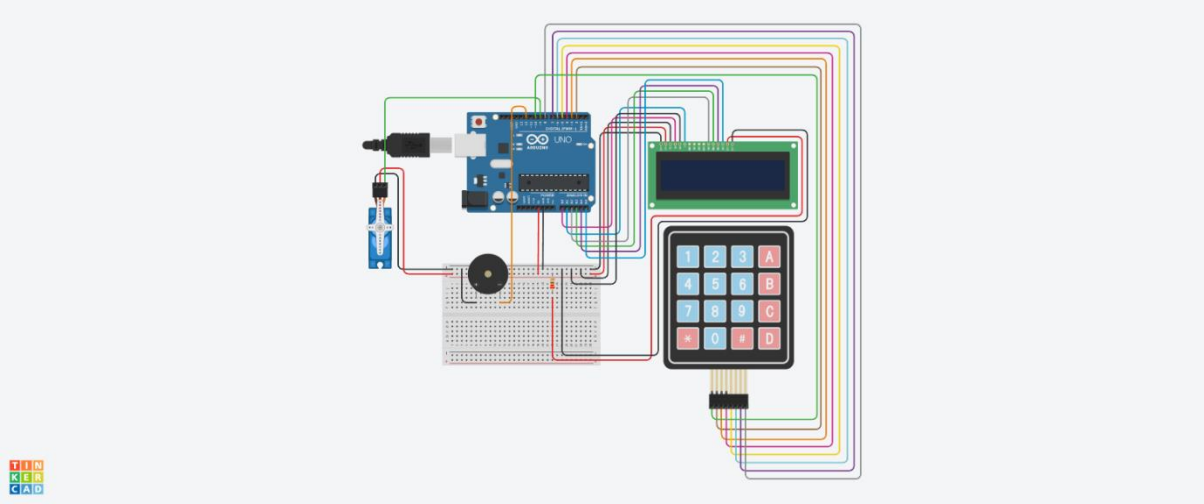
Assignment 6
Name: Rinshi Kumari
Roll No.: 210108040

1. Develop a lock system featuring a keypad, an LCD screen, and a servo motor.
a. Incorporate a buzzer that emits a sound for each incorrect attempt.
b. Implement a password reset option (e.g., entering "####" enables password reset after entering the current password).

https://www.tinkercad.com/things/ftURNkYNC6D-assignment-6-part-a/editel?sharecode=I2-Ii0J110oayoWJxJqtlzKsBuSnObPon1wEeeSNSyA



Code:

```
#include <Key.h>

#include <Keypad.h>

#include <LiquidCrystal.h>

#include <Servo.h>

Servo myservo;

int pos=0; // LCD Connections

LiquidCrystal lcd(A0,A1,A2,A3,A4,A5);

const byte rows=4;

const byte cols=4;
```

```
char key[rows][cols]={
{'1','2','3','A'},
{'4','5','6','B'},
{'7','8','9','C'},
{'*','0','#','D'},
};
byte rowPins[rows]={10,2,3,4};
byte colPins[cols]={5,6,7,8};
Keypad keypad= Keypad(makeKeymap(key),rowPins,colPins,rows,cols);
char pass[4]={'*','*','*','*'};
char actual[4]={'1','2','3','4'};
char res[4]={'#','#','#','#'};
int l=0;
int r=0;
int buzz=12;
void setup(){
  Serial.begin(9600);
  pinMode(buzz, OUTPUT);
  myservo.attach(11); //SERVO ATTACHED//
  lcd.begin(16,2);
}
void loop(){
  char k=keypad.getKey();
  if(k){
    Serial.println(k);
    Serial.println(r);
    lcd.setCursor(0,0);
    if(r==1) lcd.print("ENTER NEW PASSWORD:");
```

```
   else lcd.print("ENTER PASSWORD:");
   lcd.setCursor(l,1);
   lcd.print(k);
   pass[l]=k;
   l++;
   if(l==4){
     if(r){
          reset();
       r=0;
     }
     else correct();
     l=0;
     lcd.clear();
   }
  }
}

void displayscreen(){
          lcd.setCursor(0,0);
          lcd.print("ENTER PASSWORD");
}

void correct(){
  r=1;
  int ans=1;
  for(int i=0;i<4;i++){
          ans*=(pass[i]==actual[i]);
    r*=(pass[i]==res[i]);
  }
```

```
  if(ans) unlock();
  else if(r){
    myservo.write(0);
          lcd.clear();
    lcd.print("New Password:");
    delay(500);
    lcd.clear();
  }
  else donot();
}
void reset(){
  for(int i=0;i<4;i++){
          actual[i]=pass[i];
  }
  lcd.clear();
  lcd.print("PASSWORD CHANGED");
  delay(1000);
  lcd.clear();
}
void unlock(){
  lcd.clear();
  lcd.setCursor(0,0);
  lcd.print("UNLOCKED");
  myservo.write(255);
  delay(500);
  lcd.clear();
}
void donot(){
    lcd.clear();
```

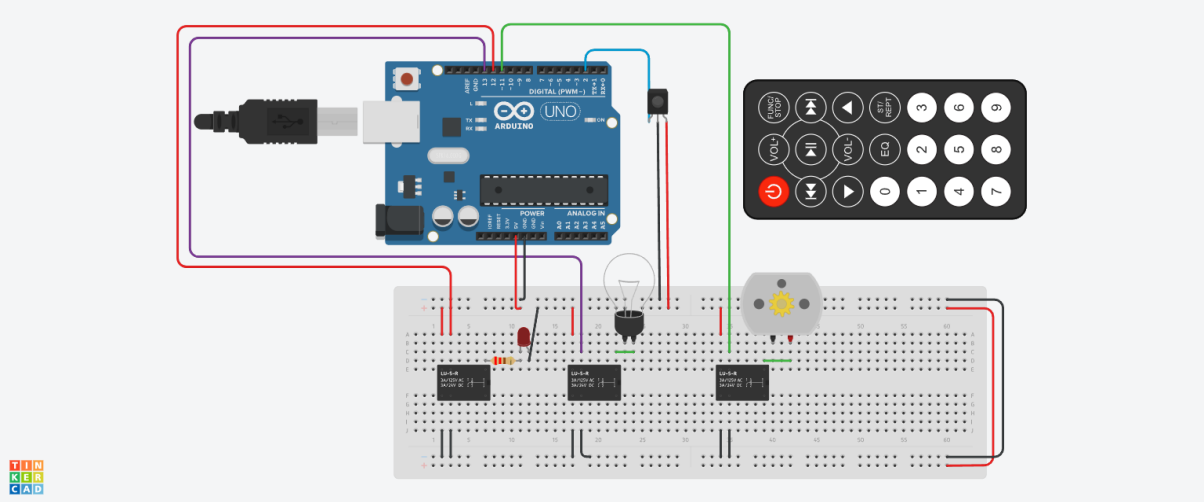
```
  lcd.setCursor(0,0);
  lcd.print("WRONG PASSWORD");
      tone(buzz, 85);
  delay(500);
  noTone(buzz);
  delay(500);
}
```

Approach:

We start by initialising the system. We set the initial password length ('l') to zero and the reset flag ('r') to zero. When 'r' is 0, we check the password to open the door; when 'r' is 1, we reset the password. The display is updated accordingly.
Entering Password: When a number is entered using the keypad, we increment 'l' and store the entered value in 'pass'. Our password length is set to 4 digits.
Password Length Check: When 'l' becomes 4, we start checking conditions. If 'r' is 1, we call the 'reset()' function.
CASE 1 (Correct Password): The 'correct()' function checks if the entered password matches the actual password or the reset code. If there is a match, it calls the 'unlock()' function.
CASE 2 (Reset Password): If the entered passcode is the reset code, 'r' is set to 1, and a message prompts the user to enter a new password. The system returns to the main loop, where a new password can be set through the 'reset()' function.
CASE 3 (Wrong Password): If the entered password is incorrect, the 'donot()' function is called. The 'donot()' function displays "WRONG PASSWORD" on the LCD and activates the buzzer.

2. Deploy a system capable of controlling three devices using an IR LED and a remote control. By pressing buttons 1, 2, and 3, you can toggle the devices on/off. Utilize a relay to operate these devices, selecting a motor, a light bulb, and an LED as the chosen devices.

https://www.tinkercad.com/things/hz23ccp9u7Z-6-b/editel?sharecode=FB8rYqLezgfzpaNk41wy_UqME-ky8Hea9M88nDu8tCw



Code:

```
#include "Adafruit_LEDBackpack.h"

#include <IRremote.h>

int button = 0;

int LED_p = 12;

int LED_s = 0;

int Motor_p = 11;

int Motor_s = 0;

int bulb_p =13;

int bulb_s = 0;

int mapCodeToButton(unsigned long code) {

 if ((code & 0x0000FFFF) == 0x0000BF00) {

  code >>= 16;

  if (((code >> 8) ^ (code & 0x00FF)) == 0x00FF) {

   return code & 0xFF;

  }

 }
```

```
  return -1;
}
int readInfrared() {
  int result = -1;
  if (IrReceiver.decode()) {
    unsigned long code = IrReceiver.decodedIRData.decodedRawData;
    result = mapCodeToButton(code);
    IrReceiver.resume();
  }
  return result;
}
void setup()
{
  IrReceiver.begin(2);
  Serial.begin(9600);
  pinMode(LED_p,OUTPUT);
  pinMode(Motor_p,OUTPUT);
  pinMode(bulb_p,OUTPUT);
}
void loop()
{
  button = readInfrared();
  if(button>1)
  Serial.println(button);
  if (button == 17) {
    if (LED_s==0) {
            digitalWrite(LED_p, HIGH);
            LED_s = 1;
    }
    else {
      digitalWrite(LED_p, LOW);
```

```
    LED_s = 0;
   }
  }
  if (button == 18) {
   if (Motor_s==0) {
          digitalWrite(Motor_p, HIGH);
          Motor_s = 1;
   }
   else {
    digitalWrite(Motor_p, LOW);
    Motor_s = 0;
   }
  }
  if (button == 16) {
   if (bulb_s==0) {
          digitalWrite(bulb_p, HIGH);
          bulb_s = 1;
   }
   else {
    digitalWrite(bulb_p, LOW);
    bulb_s = 0;
   }
  }
  delay(10);
}
```

Approach:

We are using an IR-based remote control and an IR receiver with the Arduino library . Our system comprises two key functions: 'readInfrared()' and 'mapCodeToButton()'. 'readInfrared()' is responsible for detecting IR signals from the remote control using the IR receiver, generating a hexadecimal code. The generated hex code is then passed to 'mapCodeToButton()', which simplifies it into an integer value that can be readily used for our intended task.

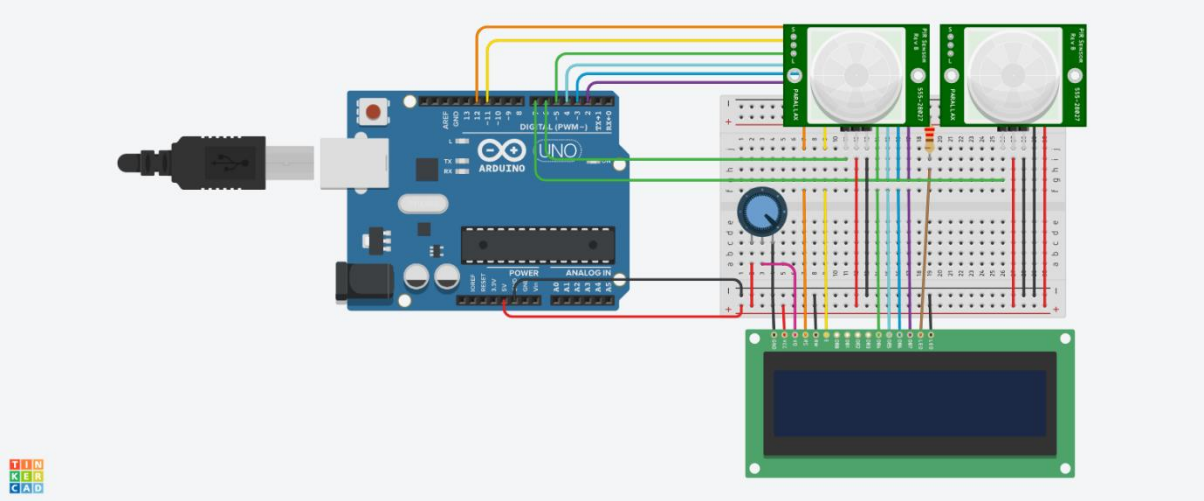
We've observed that when specific buttons are pressed on the remote, they correspond to specific integer values. For example, pressing '1' on the remote maps to '16', '2' maps to '17', and '3' maps to

'18. Device Control: Our system controls various digital devices through the use of relays. This is essential because attempting to power these devices directly from the Arduino would draw too much power, potentially causing issues. Specifically, we connect the digital pins of the Arduino as follows: '13' to an LED, '12' to a motor, and '11' to a bulb, with each device connected via its respective relay.

Button '1' on the remote control enables us to toggle the LED on/off. Button '2' on the remote turns the motor on/off. Button '3' on the remote controls the bulb, allowing us to switch it on and off as needed. Our system employs an IR-based remote control to interact with various digital devices through relays, ensuring that the appropriate amount of power is supplied to operate each device effectively. The remote's buttons are mapped to specific integer values for easy and intuitive control of these devices.

3. Utilize a PIR sensor to design a system capable of monitoring and displaying a count of individuals in a room. Ensure that the system can accurately track people entering and leaving the room.

https://www.tinkercad.com/things/2XFn8aYhlbK-people-count-using-pir-sensor/editel?sharecode=9A3ZZY56zyYtraBrN7_GpL660uFY2Ize6TrZaJL1_bI



Code:

```
#include <LiquidCrystal.h>

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

int exited=0;

int entered=0;

int count=0;

void setup() {

  lcd.begin(16, 2);

  lcd.setCursor(0, 0);

  lcd.print("No. of people:");

}

void loop() {

  if(digitalRead(6)==HIGH)

{

  entered++;

  lcd.setCursor(5,1);

  count=entered-exited;
```

```
  lcd.print(String(count));
  delay (2500);
}
if (digitalRead(7) ==HIGH)
{
  exited++;
  if(entered>=exited){
  lcd.setCursor(5, 1);
  count=entered-exited;
  lcd.print(String(count));
  delay (2500);
  }
}
  }
```

Approach:

PIR sensors consist of pyroelectric sensors housed in a round metal can with a rectangular crystal at the centre. These sensors can detect levels of infrared (IR) radiation. All objects emit a certain amount of IR radiation, and the hotter an object is, the more radiation it emits. PIR sensors are capable of detecting these variations in IR radiation. The PIR sensor is divided into two halves in a motion detector application. This design allows it to detect changes in IR radiation levels rather than just the average IR levels. The two halves of the PIR sensor are interconnected in a way that they counterbalance each other. If one half detects more or less IR radiation than the other, it triggers an output signal that goes either high or low.