

III-SEMESTER B.TECH COMPUTER AND COMMUNICATION ENGINEERING

2022-2026 BATCH

**19CCE201-MICROCONTROLLER AND INTERFACING
TECHNIQUES**



**TITLE: PASSWORD BASED DOOR LOCK SYSTEM
USING LPC2148**

PREPARED BY:

Anirudh V - CB.EN.U4CCE22006

Ashmita D - CB.EN.U4CCE22009

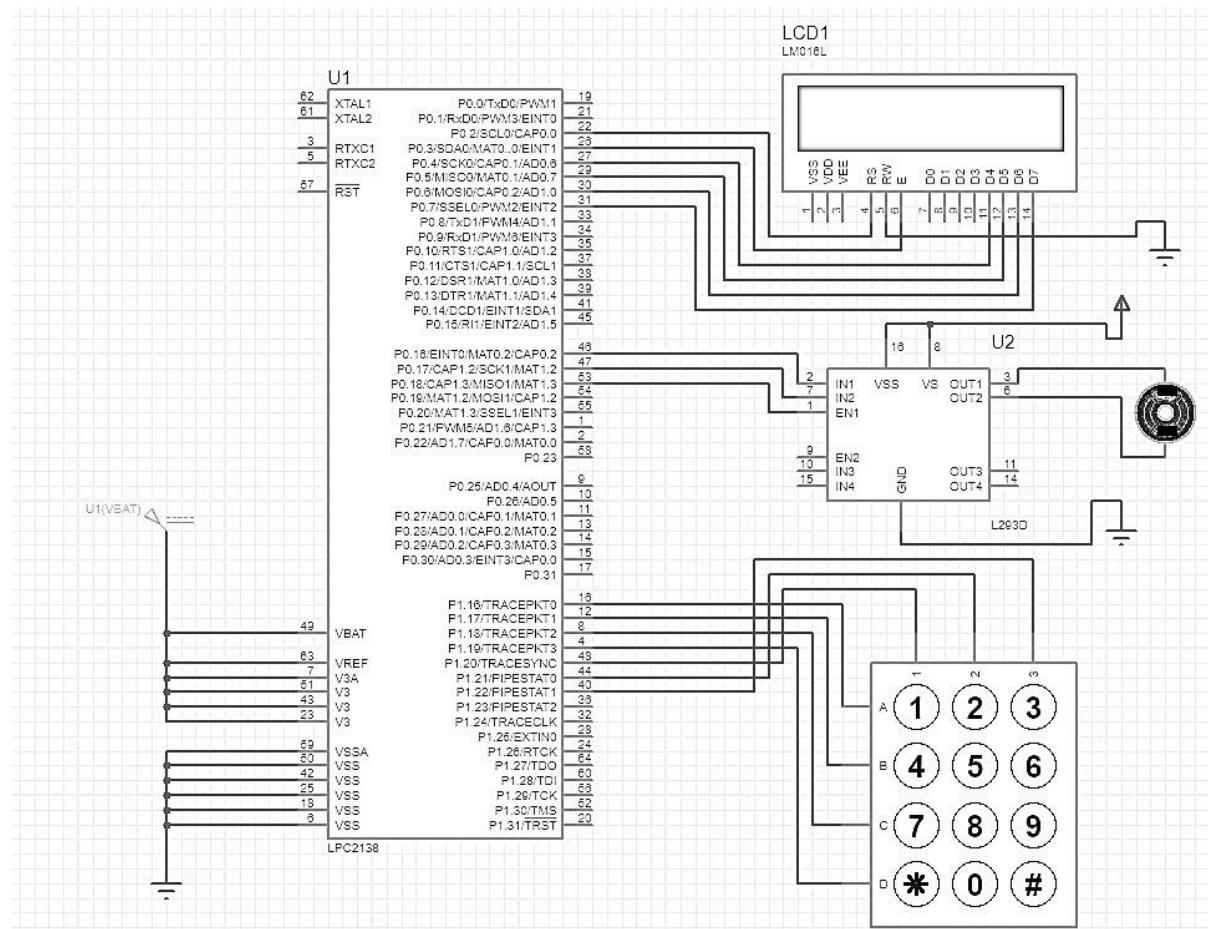
Shri Namrutha S - CB.EN.U4CCE22047

Samyuktha V - CB.EN.U4CCE22043

AIM:

The project aims to build a password-based door lock system. LPC2148 Microcontroller is used in this project for the efficient implementation. It aims to make a secure and safe locking system by assigning a unique password to each user using this system.

DESIGN:



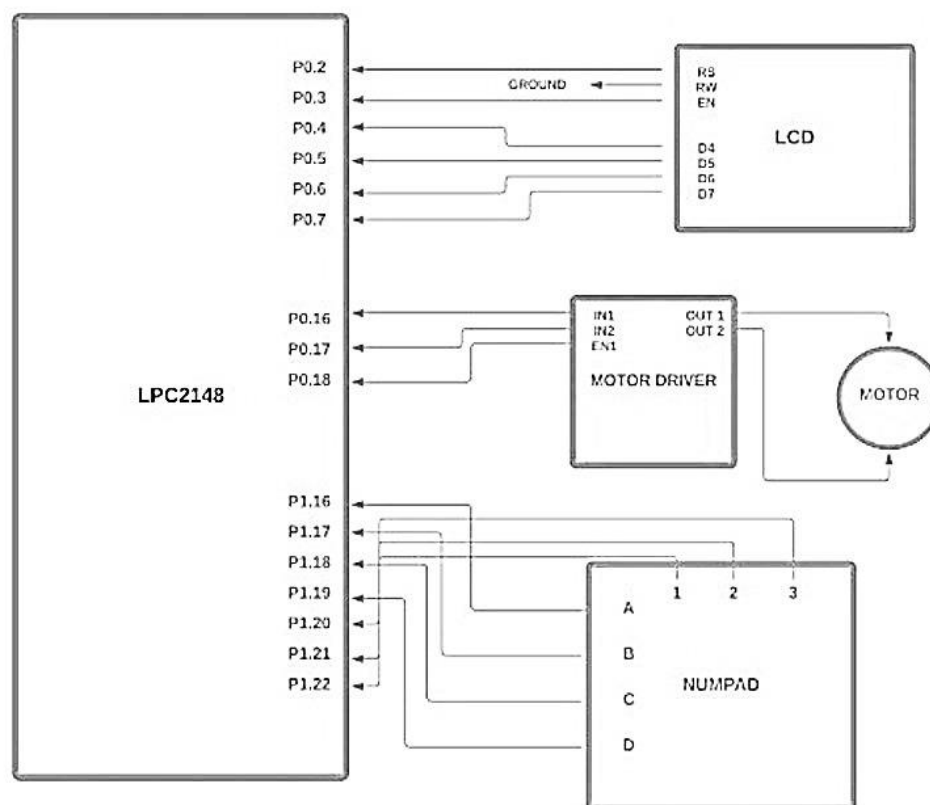
COMPONENTS USED:

- LPC2148 Microcontroller
- 16 x 2 LCD Display
- L293D Motor Driver
- DC Motor
- 4 x 3 Keypad

WORKING:

The door locking system project aims to create a secure way to control access for homes or businesses. It uses a 4x3 keypad for password entry, an LCD for messages, and a DC motor for controlled door opening. Users input a 4-digit password on the keypad, and the system displays messages on the LCD. If the correct password is entered, the door unlocks, and the DC motor moves it forward, displaying a success message before stopping and reversing. The project focuses on developing a reliable, user-friendly, and advanced door locking system that meets the modern security needs of homes and businesses.

BLOCK DIAGRAM:



CODE:

```
//include the header file for providing definition and declaration specific to lpc212x
#include <lpc214x.h>

unsigned int i;          //Declare a global variable for further usage in iteration

#define bit(x) (1 << x) // Define a macro -Logic Shift Left

typedef struct           // Define a structure named user to store username and password
{
    char username[10];
    char password[5];
} User;

//Declare an array of user structures with 20 passwords along with their names
User users[20] = {
    {"SAMYUKTHA", "1234"}, {"ASHMITA", "5678"}, {"NAMRUTHA", "9876"}, {"ANIRUDH",
"4321"}, {"TOMMY ", "0000"}, {"JERRY ", "1111"}, {"DRACO ", "2222"}, {"RAMESH",
"3333"}, {"BONNY ", "4444"}, {"HERMOINE", "5555"}, {"HARRY ", "6666"}, {"HEDWIG",
"7777"}, {"MIKO ", "8888"}, {"ELLEN ", "9999"}, {"LUKE ", "1230"}, {"RONNY ",
"0123"}, {"NEVILLE", "2148"}, {"KENNY ", "2138"}, {"GOJO ", "8051"}, {"LEVI ", "2128"};
};

void delay()            // Delay Function
{
    unsigned int temp, ct;
    for (ct = 0; ct < 30; ct++)
    {
        for (temp = 0; temp < 65000; temp++);
    }
}

//DC Motor Functions:
void forward(void);
void reverse(void);
void stop(void);

//LCD Functions
void lcd_init(void);
void cmd(unsigned char a);
void dat(unsigned char b);
```

```

void show(unsigned char *s);
void lcd_delay(void);
// Keypad Function definitions
#define c1 (IOPIN1 & 1 << 20)
#define c2 (IOPIN1 & 1 << 21)
#define c3 (IOPIN1 & 1 << 22)
// Allocating Row and Column of 4x3 Numeric Keypad
unsigned char r_loc, c_loc;
unsigned char key[4][3] = {"123", "456", "789", "*0#"};
unsigned char keypad(void);
// Main Function:
int main()
{
    unsigned char rx_arr[4];
    int count, user_index;
    VPBDIV = 0x01; // PCLK =60 MHz
    IO1DIR |= 0x0f << 16; // P1.16 to P1.19 is assigned for the numeric 4x3 phone keypad
    IO0DIR |= 0xf00fc; //P0.2 to P0.7 is assigned for the the 16x2 alphanumeric LCD
    lcd_init(); //Call the LCD Function
    //Infinite loop (An embedded program does not stop
    while (1)
    {
        cmd(0x80); // Set DDRAM address or cursor position on display to first line
        show("ENTER PASSWORD");
        cmd(0xc5); // Loading DDRAM to second line of LCD
        for (count = 0; count < 4; count++)
        {
            rx_arr[count] = keypad(); //call the keypad function
            dat('*'); // Password masking
        }

        delay();

        user_index = -1; // Initialize to an invalid index for checking the matching password

        for ( i = 0; i < 20; i++)

```

```

{
    if (users[i].password[0] == rx_arr[0] &&users[i].password[1] == rx_arr[1] &&
        users[i].password[2] == rx_arr[2] &&users[i].password[3] == rx_arr[3])
    {
        user_index = i;
        break;
    }
}
if (user_index != -1)
{
    cmd(0xc0);
    show("ACCESS GRANTED");
    forward();
    delay();
    stop();
    cmd(0xc0);
    show("WELCOME ");
    show((unsigned char *)users[user_index].username);
    delay();
    reverse();
    delay();
    stop();
}
else
{
    cmd(0xc0);
    show("ACCESS DENIED");
    delay();
}
cmd(0x01); // Clear the display
}
}
unsigned char keypad()
{

```

```

IO1PIN &= ~(0xff << 16);
IO1PIN |= 0xf0 << 16;
while (c1 && c2 && c3);
while (!c1 || !c2 || !c3)
{
    if (!c1 && c2 && c3)
        c_loc = 0;
    else if (c1 && !c2 && c3)
        c_loc = 1;
    else if (c1 && c2 && !c3)
        c_loc = 2;
    IO1CLR = 1 << 16;
    IO1SET = 0x0e << 16;
    if (!c1 || !c2 || !c3)
    {
        r_loc = 0;
        break;
    }
    IO1CLR = 1 << 17;
    IO1SET = 0x0d << 16;
    if (!c1 || !c2 || !c3)
    {
        r_loc = 1;
        break;
    }
    IO1CLR = 1 << 18;
    IO1SET = 0x0b << 16;
    if (!c1 || !c2 || !c3)
    {
        r_loc = 2;
        break;
    }
    IO1CLR = 1 << 19;
    IO1SET = 0x07 << 16;

```

```

    if (!c1 || !c2 || !c3)
    {
        r_loc = 3;
        break;
    }
}
while (!c1 || !c2 || !c3);
return (key[r_loc][c_loc]);
}

void lcd_init()
{
    cmd(0x02); // Initialize cursor to home position
    cmd(0x28); // Set the function set (4-bit interface, two line, 5x8 dots)
    cmd(0x0c); // Display ON, Cursor OFF, and Blink OFF
    cmd(0x06); // Set the Entry mode (Cursor Increment, Display Shift OFF)
    cmd(0x80); //Set DDRAM or cursor position on display
}

// Display Character Using LCD Command Write Function
void cmd(unsigned char a)
{
    IO0PIN &= 0xfffff03; // Set P0.2-P0.7 as input
    IO0PIN |= (a & 0xf0) << 0; // Pass the MSB (Last 4 bits of our input)
    IO0CLR |= bit(2); // Command input method instruction, Set register select to 0 (RS=0)
    IO0CLR |= bit(1); //Write to LCD, Set Read/Write to 0 (RW=0)
    IO0SET |= bit(3); // LCD will be enabled ON until delay is performed (EN=1)
    lcd_delay(); // Call LCD Delay Function
    IO0CLR |= bit(3); // LCD will be turned OFF, EN=0 (Ground)
    IO0PIN &= 0xfffff03; // Set P0.2-P0.7 as input
    IO0PIN |= ((a << 4) & 0xf0) << 0; //Pass the MSB (Last 4 bits of our input)
    IO0CLR |= bit(2); // Command input method instruction, Set register select to 0 (RS=0)
    IO0CLR |= bit(1); // Write to LCD, Set Read/Write to 0 (RW=0)
    IO0SET |= bit(3); //LCD will be enabled ON until delay is performed (EN=1)
    lcd_delay(); // Call LCD delay function
    IO0CLR |= bit(3); //LCD will be turned OFF, EN=0 (Ground)

```



```

}

// Display character using LCD command write function:
void dat(unsigned char b)
{
    IO0PIN &= 0xfffff03; // Set P0.2 – P0.7 as input
    IO0PIN |= (b & 0xf0) << 0; // Pass the MSB (last 4 bits of our input 8 -bit data Into input pins)
    IO0SET |= bit(2);    // Data input method instruction, set register select to 1 (RS = 1 )
    IO0CLR |= bit(1);    // Write to LCD, set Read/Write to 0 (RW =0)
    IO0SET |= bit(3);    // LCD will be enabled ON until delay is performed (EN=1)
    lcd_delay();        // Call LCD Delay function
    IO0CLR |= bit(3);    // LCD will be turned OFF, EN = 0 (Ground)
    IO0PIN &= 0xfffff03; // Set P0.2 – P0.7 as input
    IO0PIN |= ((b << 4) & 0xf0) << 0; // Pass the LSB (last 4 bits of our input 8 Bit data into i/p pin)
    IO0SET |= bit(2);    // Data Input method instruction, Set register select to 1 (Rs = 1)
    IO0CLR |= bit(1);    // Write to LCD, Set Read/Write to 0 (RW =0)
    IO0SET |= bit(3);    // LCD will be enabled ON until delay is performed (EN = 1)
    lcd_delay();        // Call LCD Delay function
    IO0CLR |= bit(3);    // LCD will be turned OFF, EN = 0 (Ground)
}

// Function for password Masking :
void show(unsigned char *s)
{
    while (*s)    // It involves a text field tht accepts any character, however, doesn't show the
                  // inputted character to the user.
    {
        dat(*s++); // Instead, it shows an asterisk.
    }
}

// Lcd Delay Function loop :
void lcd_delay()
{
    unsigned int i;
    for (i = 0; i <= 1000; i++) ;
}

```

// DC motor functions :

// three pins – input 1 pin configuring P0.16, Input 2 pin configuring P0.17 and Enable Pin configuring P0.18

void forward()

```
{
    IO0SET = bit(16) | bit(18); // Configure P0.16 and P0.18 as LOW
    IO0CLR = bit(17); // Motor rotates in forward direction
}
```

void reverse()

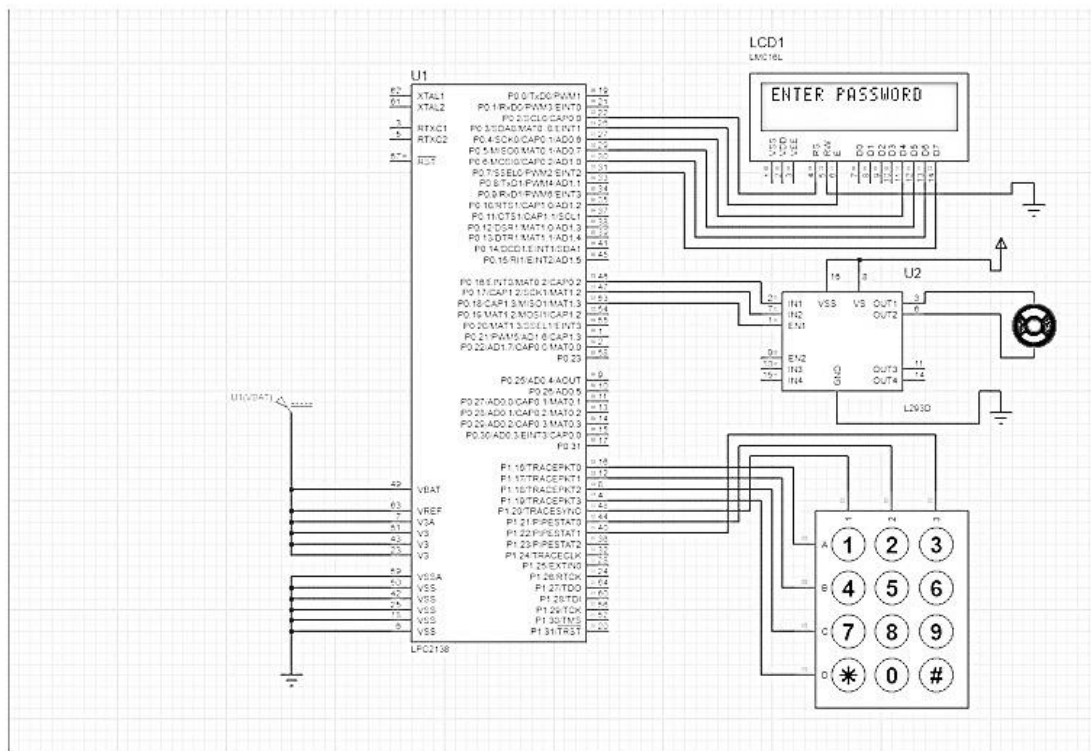
```
{
    IO0SET = bit(17) | bit(18); // Configure P0.16 and P0.18 as LOW
    IO0CLR = bit(16); // Motor rotates In reverse direction
}
```

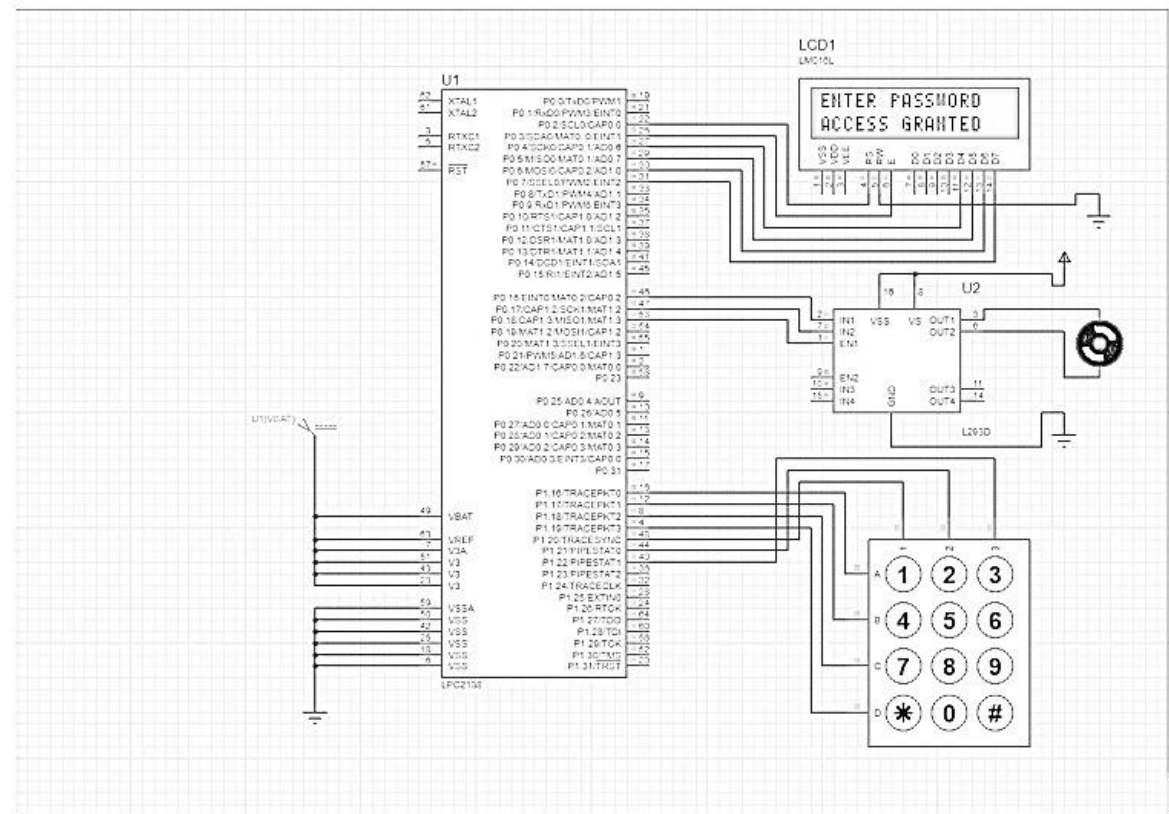
void stop()

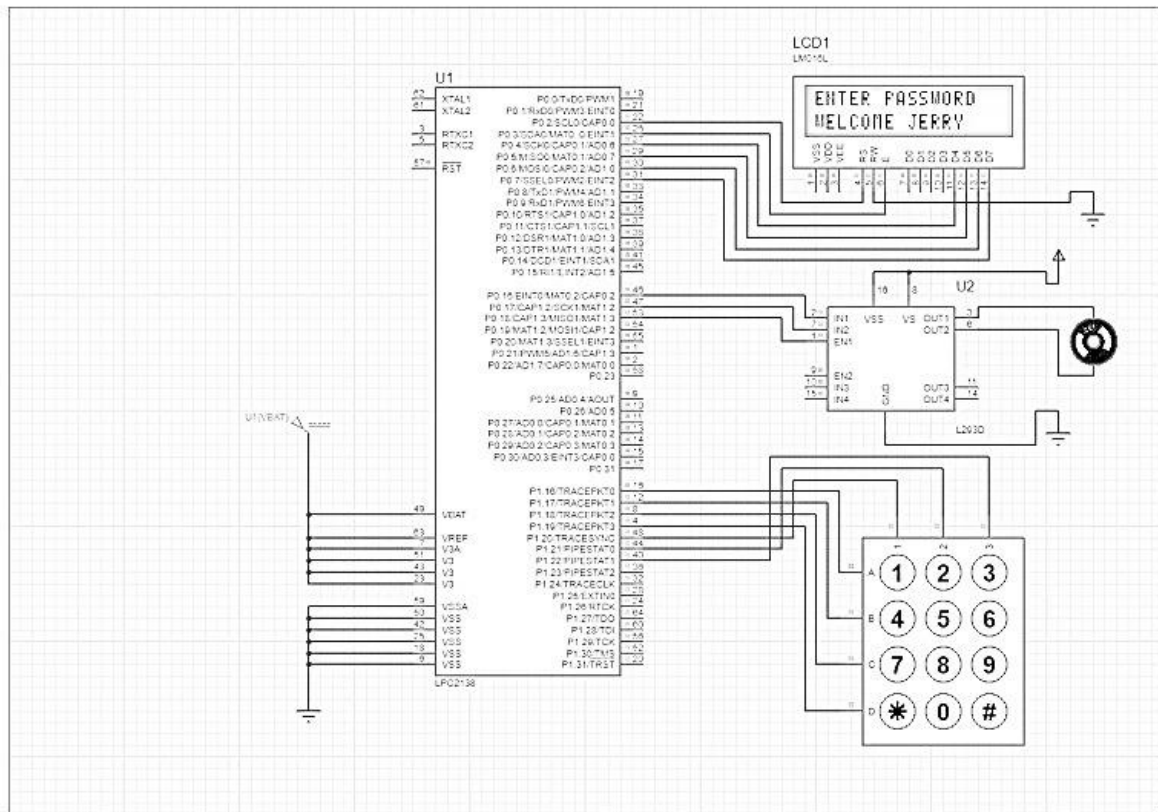
```
{
    IO0CLR = bit(18); // When P0.18 in cleared, motor rotation stops.
}
```

OUTPUT:

1. When password is correct motor rotates to indicate the door opening







2. When password is wrong "ACCESS DENIED" is displayed.

