

EC Project

Team Members:-

1. Sarthak Agarwal(210934)
2. Satish Kumar(210939)
3. Sutariya Smitkumar(211087)

Section - D

Table No. - 4

Q1. What problem are you trying to solve, and why is it important/interesting?

Ans. We are making a Number lock based door lock system which solved the following problems:-

1. **Security:** Number lock-based door locks provide a higher level of security compared to traditional lock systems.
2. **Convenience:** With Number lock-based door locks, you no longer need to carry around a physical key. This can be particularly convenient for individuals who frequently lose or misplace their keys.
3. **Key management:** Number lock-based door locks can eliminate the need for key duplication, reducing the risk of unauthorized key duplication and making key management easier.

Due to the above-mentioned problems, it will be quite interesting for us to see how the Number lock is used to replace the old existent method for door unlocking. We are students of a technical institute who are interested in cutting-edge technology and innovation. Thus, making this project is of quite relevance to us.

Q2. What are the existing solutions? Describe a few of them and list any shortcomings in them. Is your solution approach unique in some way?

Ans. There are existing solutions that solve this problem. But they have only keypad to feed password whereas we have Bluetooth app as alternative to open door.

Q3. What resources do you require to complete the project? Give a breakup of tasks that you need to accomplish week by week to complete the project.

Ans. The components that we need are as follows:-

S.N	Components Name	Quantity
1	Arduino Nano	1
2	HC-05 Bluetooth Module	1
3	Character LCD 16x2	1
4	Buzzer 5-15V	1
5	Servo Motor SG90	1
6	Breadboard	1
7	Jumper Wires	25
8	Matrix Keypad	1

Week wise breakup of tasks -

Week 1:

- Research and acquire the necessary components for the project.
- Familiarize ourselves with the basic concepts of Arduino programming, Bluetooth communication, and other components.
- Write and upload a simple Arduino program that blinks the LED light of Arduino on and off.
- Measure and show the resistor value using a Arduino code

Week 2:

- Build the circuit on a breadboard and test the servo to make sure it works properly.
- Write and upload a simple Arduino program that take input from keypad.
- Write and upload an Arduino program that reads data from keypad and process it to change the position of servo.
- Test the Bluetooth communication and make sure that it is connecting with HC05.

Week 3:

- Create an Android app that can connect to the Bluetooth module and receive fingerprint data.
- Write the code to verify the password and process it to change the position of servo.
- Test the complete system by checking it to open by keypad as well as bluetooth.

Project Description:-

We made a Number lock-based door lock system. The system gets password as input from keypad, compares it with the correct password already stored in Arduino, and using the other components of the circuit, decides whether to open the gate or have it remained closed. If it decides to open then it sends a signal to servo to change its angle and send a signal to LCD to display “Access Granted” or if password is wrong it displays “Access Denied”.

Alternatively, we made wireless control using Bluetooth module and custom-made app through which we can directly open the door. For making app, we used MIT Inventor.

Components Used:-

1. Arduino Uno-

It is the development board relying upon a microcontroller in it. It was used for controlling servo, LCD and buzzer on the basis of input it got from keypad and Bluetooth module.

2. HC05 Bluetooth Module -

It was the Bluetooth module used for sending serial data to the Arduino board based on input received.

3. Character LCD 16x2-

It is used for displaying different results we get from Arduino.

4. Buzzer-

It was used for producing a short beep sound or long duration sound, depending upon the output from Arduino.

5. Servo Motor

It is used for opening and closing gate.

6. Jumper Wires-

20 Jumper wires were used for connecting different modules.

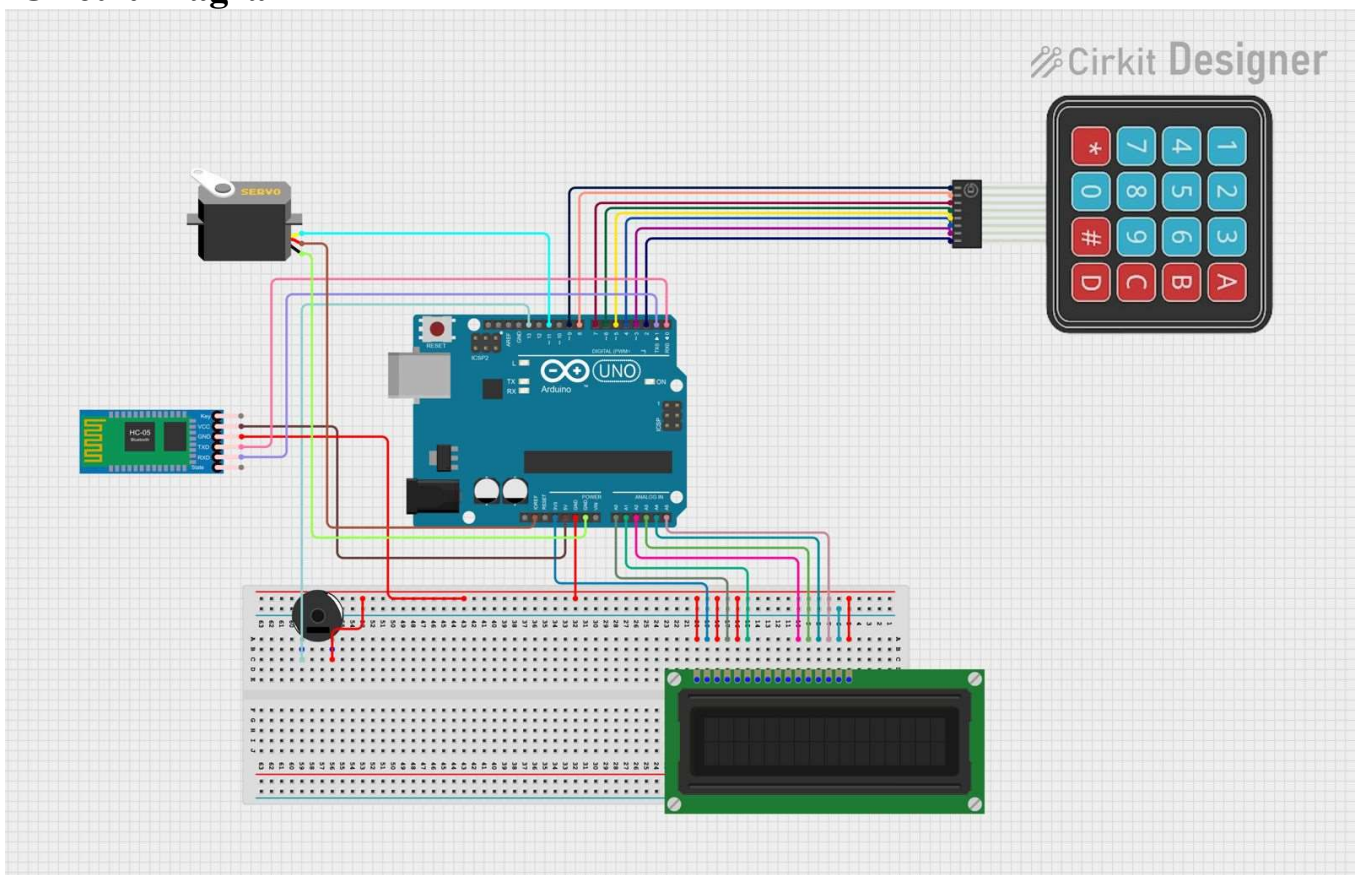
7. Matrix Keypad-

It is used for giving input to Arduino, it has 4x4 keypad with 0-9, *, #, A-D available as input.

8. Breadboard-

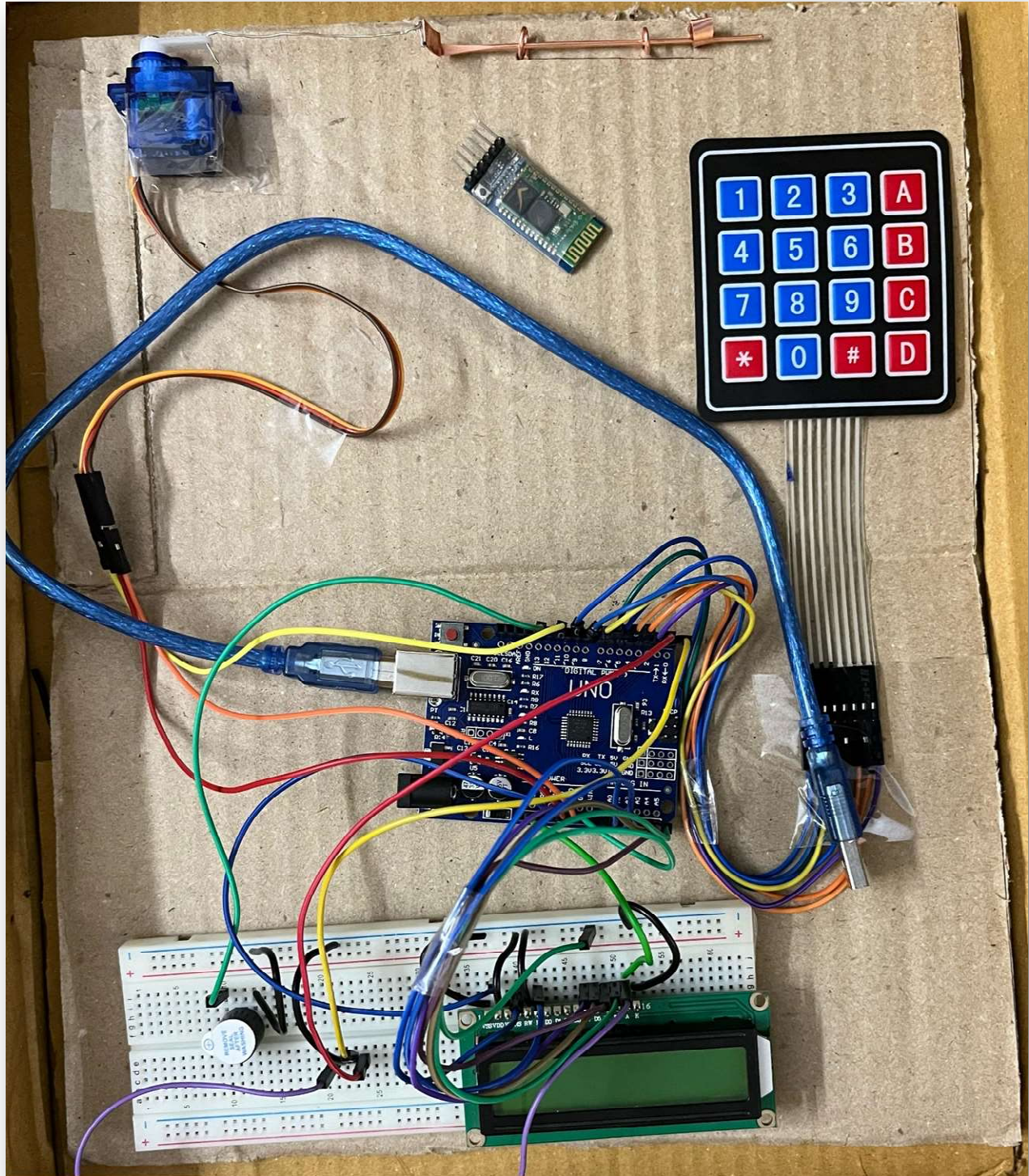
It was used for making the whole circuit.

Circuit Diagram-



- The TX and RX pins of HC05 were connected to the RX and TX of the Arduino Uno.
- V_{CC} of the Bluetooth was connected to V_{CC} (5 V) of the Uno. The GND of the Bluetooth was connected to the GND of the Uno.
- The positive terminal of the buzzer was connected to digital pin-13(D13) of the UNO, while the negative terminal was connected to the GND of the UNO.

- We connected control of servo to Digital pin-11(D11).



Working Explained-

- When we start entering data using keypad it counts till 4 digits and then compare it to Master Password and if it is correct, door opens or else it gives an 'Access Denied' message on LCD.
- If the wrong password is entered for 3 consecutive times, then a count-down of 10

seconds starts with beep sound. This keeps on increasing with more number of failed attempts.

- The TX and RX terminals were used by the HC05 to send data to the Arduino UNO when we click Open in app.
- If password is right then door is opened, else error is thrown.

Code Used-

```
#include <Keypad.h>
#include <LiquidCrystal.h>
#include <Servo.h>

Servo myservo;

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

#define Password_Lenght 5 // Give enough room for six chars + NULL char
#define buzzer 13

int pos = 0; // variable to store the servo position

char Data[Password_Lenght]; // 6 is the number of chars it can hold + the null char = 7
char Master[Password_Lenght] = "1234"; //Change PASSWORD here
byte data_count = 0, master_count = 0;
bool Pass_is_good;
char customKey;
char inputPassword[Password_Lenght];
int passwordIndex = 0;
const byte ROWS = 4;
const byte COLS = 4;
char keys[ROWS][COLS] = {
  {'1', '2', '3', 'A'},
  {'4', '5', '6', 'B'},
  {'7', '8', '9', 'C'},
  {'*', '0', '#', 'D'}
};
bool door = true;
int count = 0;
int t = 0;

byte rowPins[ROWS] = {9, 8, 7, 6}; //connect to the row pinouts of the keypad
byte colPins[COLS] = {5, 4, 3, 2}; //connect to the column pinouts of the keypad

Keypad customKeypad( makeKeymap(keys), rowPins, colPins, ROWS, COLS); //initialize an instance of class NewKeypad

void setup()
{
  myservo.attach(11);
  ServoClose();
  lcd.begin(16, 2);
  pinMode(buzzer, OUTPUT);
  Serial.begin(9600);
}

void loop()
{
  if (Serial.available()){
    readPassword();
    if(!strcmp(inputPassword, Master)){
      lcd.clear();
      ServoOpen();
    }
  }
}
```

```

    lcd.print(" ACCESS GRANTED");
    Serial.println("Door opened");
    delay(1000);
    passwordIndex = 0;
    door = 0;
}
if(strcmp(inputPassword, Master)){
    Serial.println("Wrong Password");
    lcd.clear();
    lcd.setCursor(1,0);
    lcd.print("ACCESS DENIED!");
    delay(2000);

}
}
}
if (door == 0)
{
    customKey = customKeypad.getKey();

    if (customKey == 'C')

    {

        lcd.clear();
        lcd.setCursor (3,0);
        lcd.print("LOCKING");
        delay (700);
        lcd.setCursor (10,0);
        lcd.print(".");
        delay (800);
        lcd.setCursor (11,0);
        lcd.print(".");
        delay (800);
        lcd.setCursor (12,0);
        lcd.print(".");
        delay (500);
        lcd.clear();
        delay (500);
        ServoClose();
        lcd.setCursor (5,0);
        lcd.print("LOCKED");
        delay (1700);

        door = 1;
    }
}

else Open();
}

void clearData()
{
    while (data_count != 0)
    { // This can be used for any array size,
      Data[data_count--] = 0; //clear array for new data
    }
    return;
}

void readPassword() {
    while (Serial.available() && passwordIndex < 5 ) {
        inputPassword[passwordIndex++] = Serial.read(); // Read characters into input password array
        delay(10); // Small delay for stability
    }
    inputPassword[passwordIndex] = '\0'; // Null terminate the string
}

void ServoOpen()
{

```

```

for (pos = 180; pos >= 0; pos -= 5) { // goes from 0 degrees to 180 degrees
  // in steps of 1 degree
  myservo.write(pos);           // tell servo to go to position in variable 'pos'
  delay(15);                     // waits 15ms for the servo to reach the position
}
}

void ServoClose()
{
  for (pos = 0; pos <= 180; pos += 5) { // goes from 180 degrees to 0 degrees
    myservo.write(pos);           // tell servo to go to position in variable 'pos'
    delay(15);                     // waits 15ms for the servo to reach the position
  }
}

void Open()
{
  lcd.setCursor(1, 0);
  lcd.print("INPUT PASSWORD");

  customKey = customKeypad.getKey();
  if (customKey) // makes sure a key is actually pressed, equal to (customKey != NO_KEY)
  {
    lcd.setCursor(0, 1);
    lcd.print(""); //To hide your PASSWORD, make sure its the same lenght as your password
    Data[data_count] = customKey; // store char into data array
    lcd.setCursor(data_count, 1); // move cursor to show each new char
    lcd.print("*"); // print char at said cursor
    data_count++; // increment data array by 1 to store new char, also keep track of the number of chars entered
  }

  if (data_count == Password_Lenght - 1) // if the array index is equal to the number of expected chars, compare data to master
  {
    if (!strcmp(Data, Master)) // equal to (strcmp(Data, Master) == 0)
    {
      lcd.clear();
      ServoOpen();
      lcd.print(" ACCESS GRANTED");
      lcd.setCursor(0,1);
      lcd.print("press C to close");
      door = 0;
      count=0;
    }
    else{
      if(count == 2){
        lcd.clear();

        lcd.setCursor(1,0);
        lcd.print("ACCESS DENIED!");
        delay(1000);
        t=10;
        while(t>0){

          lcd.clear();
          lcd.setCursor(1,0);
          lcd.print("Wait for sec:");
          lcd.print(t);
          digitalWrite(buzzer, HIGH);
          delay(100);
          digitalWrite(buzzer, LOW);
          delay(900);
          door = 1;
          t--;
        }
        count++;
      }
      else{
        if(count > 2){
          lcd.clear();

```



```

lcd.setCursor(1,0);
lcd.print("ACCESS DENIED!");
delay(1000);
  t=10*(count-1);
  while(t>0){

    lcd.clear();
    lcd.setCursor(1,0);
    lcd.print("Wait for sec:");
    lcd.print(t);
    digitalWrite(buzzer, HIGH);
    delay(100);
    digitalWrite(buzzer, LOW);
    delay(900);
    door = 1;
    t--;
  }
  count++;
}
else
{
  count++;

  lcd.clear();

  lcd.setCursor(1,0);
  lcd.print("ACCESS DENIED!");
  delay(1500);
  door = 1;
  digitalWrite(buzzer, HIGH);
  delay(100);
  digitalWrite(buzzer, LOW);
  delay(50);
  digitalWrite(buzzer, HIGH);
  delay(100);
  digitalWrite(buzzer, LOW);
  delay(50);
  digitalWrite(buzzer, HIGH);
  delay(100);
  digitalWrite(buzzer, LOW);

}
}
}
clearData();
}
}

```

About the App -

The app was created using MIT Inventor's drag-and-drop blocks-based interface. The logic for sending data to the HC-05 Bluetooth module based on password authentication was implemented using blocks in the 'Blocks' screen. The 'Password' component detected string input, and the 'Bluetooth Client' component was used to establish a connection and send data to the HC-05 Bluetooth module. When a password was entered and “OPEN” button was clicked, the app sent password string to the Bluetooth module. The 'List Picker' component allowed the user to select an option, and the selected option was displayed using the 'Text' block. Once the logic was implemented, the app was exported to a .apk file for installation on a device for testing or distribution.

Final Result -

The final project can be seen in the following video –

https://drive.google.com/file/d/1zfzw2SEAE_8aYkC3RRwngbsMrtelkLf3/view?usp=sharing