BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Department of Electrical and Electronic Engineering

Course No.	**:** EEE 310	**Group No**	**:** 03
Course Title	**:** Communication Laboratory	**Section**	**:** C

Project Report on,

Intelligent Gesture Based Security System

Level : 3 **Term :** 1

Name	**Student ID**
Fariha Ferdousi	1706143
Nausin Tabassum Kudrot	1706158
Sadia Afrose	1706161
Raisa Mashtura	1706163
Subah Karnine	1706174

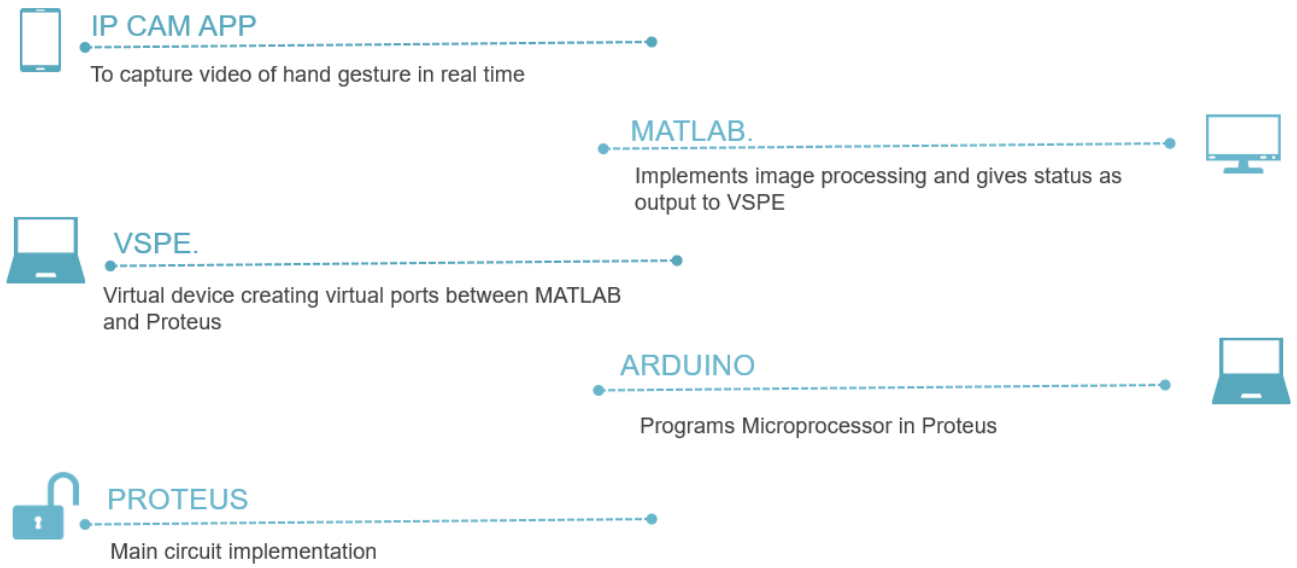
Objective:

The main objective of this project is to design a security system that can be unlocked simply by means of a stored hand gesture pattern. The user of the security system can save a gesture pattern. Afterwards, if the user is able to recreate the gesture, the systems recognizes it and unlocks. A security system like this can be used in homes, offices, hospitals etc

The reasons for coming up with this project is:

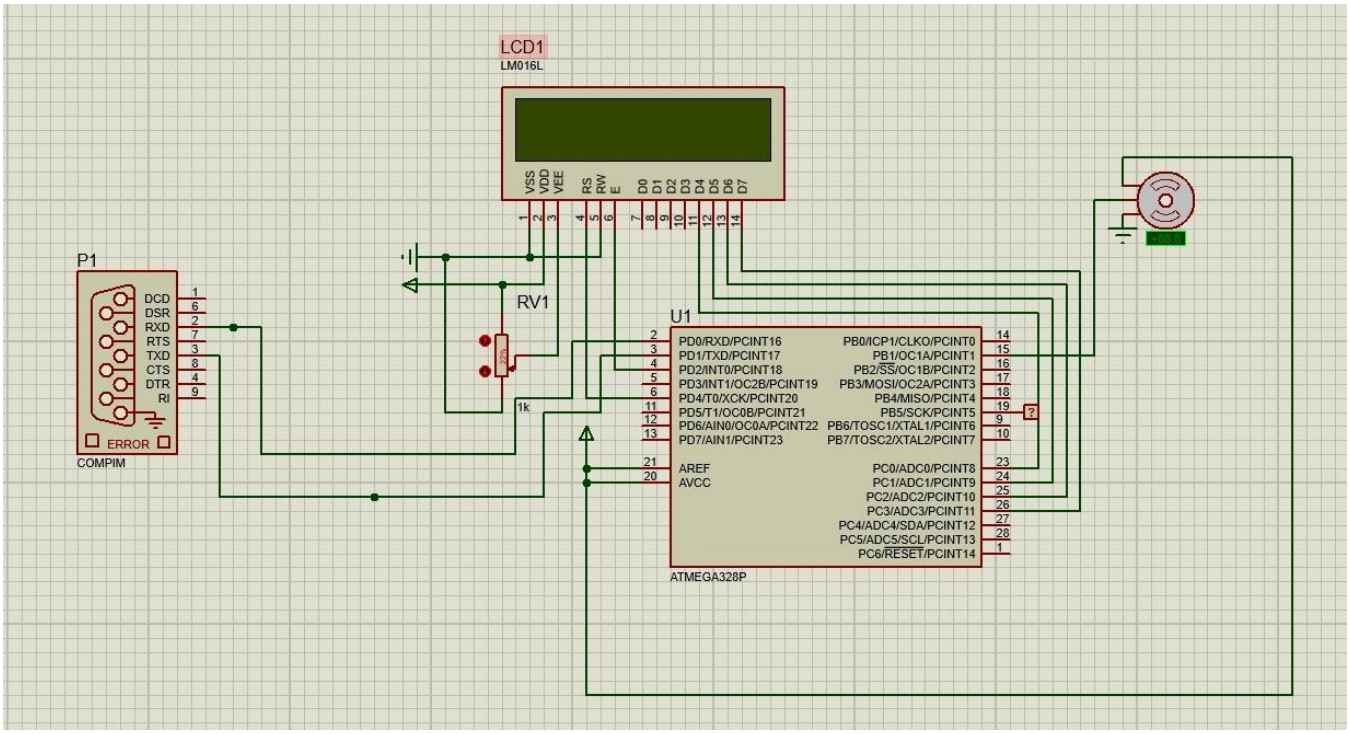
1) **Safe alternative to retina scan or finger print based security system**
   During a pandemic situation like in the current statusquo, minimizing human contact with machines is a goal
2) **Easy implementation in real life**
   Much less advanced technology needed compared to other security systems. Its also cost effective.
3) **User friendly Security system**
   User can set and reset password whenever needed.
4) **Convenient for Disabled people**
   With hand gesture recognition, interaction with computer becomes easier.

Softwares used:



Schematic Diagram :

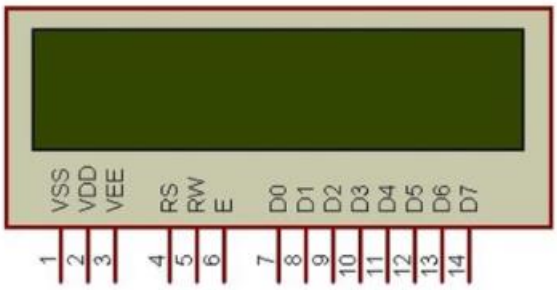
The hardware part of this project was simulated using proteus. The schematic diagram is shown below:



The list of components used in this project is given below:

- Microprocessor, ATEMEGA328P : The microprocessor is programmed in a way so that it can rotate the servo motor and display message on the LCD monitor by reading the input from the serial port. It is programmed using Arduino IDE.
- COMPIM : COMPIM is used to model physical COM interfaces in Proteus.It is used to establish serial connection between MATLAB and Proteus through VSPE.
- LCD Display, LM016L: It is a character based LCD monitor which can display maximum 2 lines including 16 characters per line.

  The schematic model of LM016L is shown below:



  The following table describes the functions of different Pins of this model:

Pin No.	Name	Description
Pin no. 1	**VSS**	Power supply (GND)
Pin no. 2	**VCC**	Power supply (+5V)
Pin no. 3	**VEE**	Contrast adjust
Pin no. 4	**RS**	0 = Instruction input 1 = Data input
Pin no. 5	**R/W**	0 = Write to LCD Module 1 = Read from LCD module
Pin no. 6	**EN**	Enable signal
Pin no. 7	**D0**	Data bus line 0 (LSB)
Pin no. 8	**D1**	Data bus line 1
Pin no. 9	**D2**	Data bus line 2
Pin no. 10	**D3**	Data bus line 3
Pin no. 11	**D4**	Data bus line 4
Pin no. 12	**D5**	Data bus line 5
Pin no. 13	**D6**	Data bus line 6
Pin no. 14	**D7**	Data bus line 7 (MSB)

- Logic Probe: A **logic probe** is a test probe used for analyzing and troubleshooting the logical states (boolean 0 or 1) of a digital circuit.
- Motor- PWMSERVO : This motor is used for unlocking the door. For appropriate inputs the motor will rotate 180 degrees from its initial position in order to unlock the door.
- POT- HG: The potentiometer is **used to adjust the bias level of the LCD** i.e. the contrast of the LCD display.

Circuit connections:

- The serial pins named TXD(pin 3) and RXD(pin 2) pins are connected to the Atmega and used for USB programs and communicating with it.
- Compim's data receiver(pin 2) and transfer pins(pin 3) were connected to ATmega's corresponding pins accordingly. Compim's physical port was set to COM4 and Baud Rate 9600. Compim captures the serial data from MATLAB and transfers it to ATmega's serial port.
- A logic probe is placed to pin 19 to display if the password is correct or incorrect.
- 5V DC Power is supplied to AVCC pin.
- A servo motor is connected to digital pin 15 for opening and closing door.

- For connecting LCD to the board VDD is connected to 5V dc voltage, Vss and RW(read/write) pin to ground as LCD is just writing data. VEE pin connected to potentiometer to adjust the bias level of the lcd. E pin connected to digital pin 4 to enable signal and RS pin connected to pin 6 for receiving input data. D4,D5,D6 and D7 pins were connected to analog pins 23,24,25,26 and 27 accordingly for data bus line.

Programming the Microprocessor:

The microproceesor takes input fom MATLAB through the COMPIM connected at pin 2 and pin 3 of ATMEGA328P.

The LCD monitor used in this project is programmed to display maximum 2 lines including 16 characters per line.

The Arduino code code reads inout from the serial port and performs the following function:

If input is '1' :

- The servo motor attached at pin 15 of ATMEGA328P (pin 9 of Arduino) rotates by 180 degrees and thus unlocks the door.
- The LCD monitor displays 'unlocked!'
- The servo motor remains at this position for 3 seconds and then rotates back to its initial position I.e. 0 degrees. Thus, the door becomes locked again.
- The LCD monitor displays 'locked!'

If input is any number other than '1' :

- The servo motor doesn't rotate at all and thus the door remains locked.
- The LCD monitor displays 'Wrong!' ; 'Try again!' .

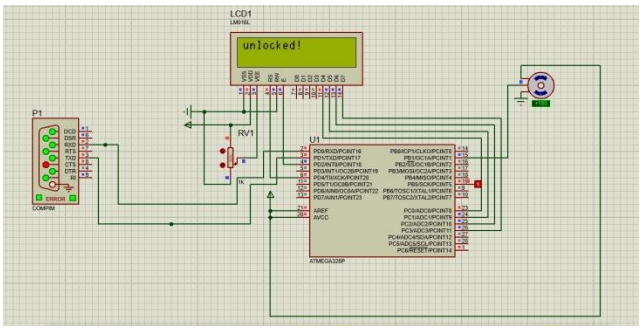


fig. 1: If correct password is enterd

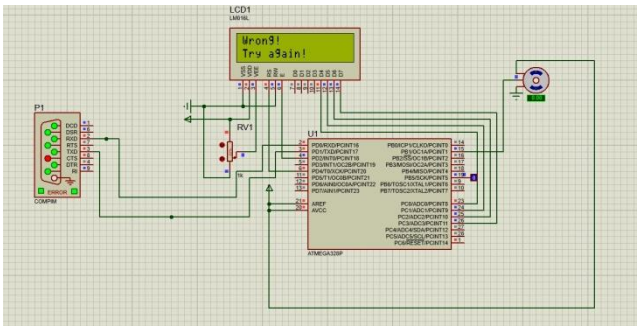
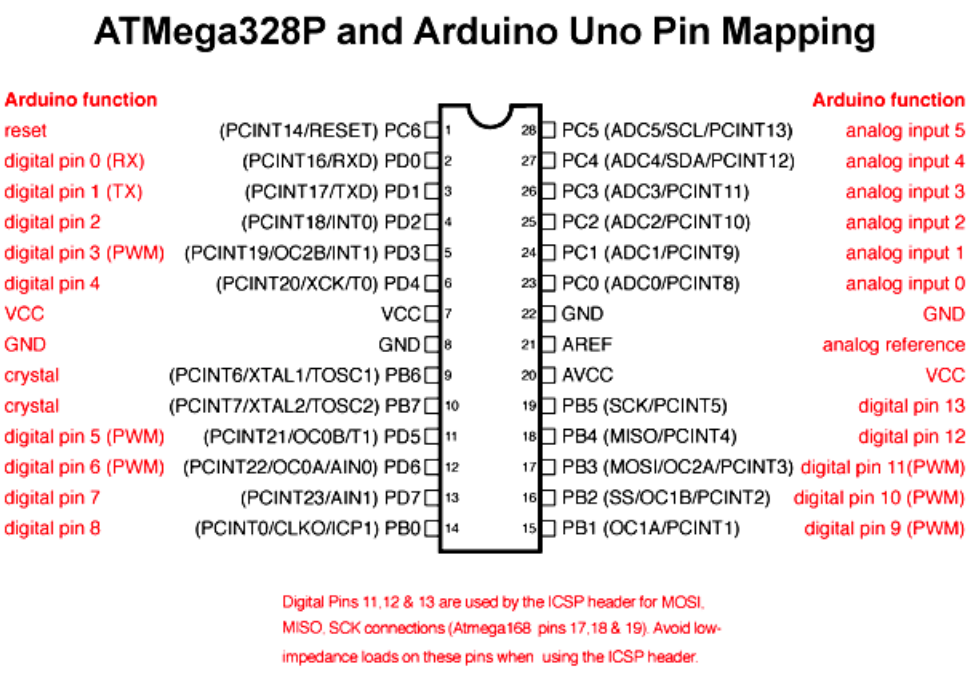


fig. 2: If incorrect password is entered

The input and output pins of Arduino are mapped with that of ATMEGA328P using the diagram below:



Arduino Code:

```
int solenoidPin = 13; //This is the output pin on the Arduino

int doi;

#include <Servo.h>

#include <LiquidCrystal.h>


// initialize the library by associating any needed LCD interface pin

// with the arduino pin number it is connected to

const int rs = 4, en = 2, d4 = A0, d5 = A1, d6 = A2, d7 = A3;

LiquidCrystal lcd(rs, en, d4, d5, d6, d7);


Servo myservo;  // create servo object to control a servo

// twelve servo objects can be created on most boards
```

```
int pos = 0;    // variable to store the servo position
void setup()
{
  Serial.begin(9600);
  myservo.attach(9);  // attaches the servo on pin 9 to the servo object
  pinMode(solenoidPin, OUTPUT); //Sets that pin as an output
   // set up the LCD's number of columns and rows:
  lcd.begin(16, 2); //the LCD has 2 lines, and can display 16 characters per line
}
void loop()
{
if(Serial.available()>0)
{
   doi = Serial.read();
   if (doi == 1)
   {
    //Serial.println(1);
    digitalWrite(solenoidPin, HIGH); //Switch Solenoid ON
    //delay(1000); //Wait .15 Second
     //digitalWrite(solenoidPin, LOW);
     for (pos = 0; pos <= 180; pos += 1) { // 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

  }
```

```
  lcd.print("unlocked!");
  delay(3000);
  for (pos = 180; pos>=0; pos -= 1) { // 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

  }
  digitalWrite(solenoidPin, LOW);
  lcd.clear();
  lcd.print("locked!");
  delay(3000);
  lcd.clear();

    }
  else
    {
      lcd.print("Wrong!");
      lcd.setCursor(0, 1);
      lcd.print("Try again!");
      delay(2000);
      lcd.clear();
    }
}
}
```

Image Processing:

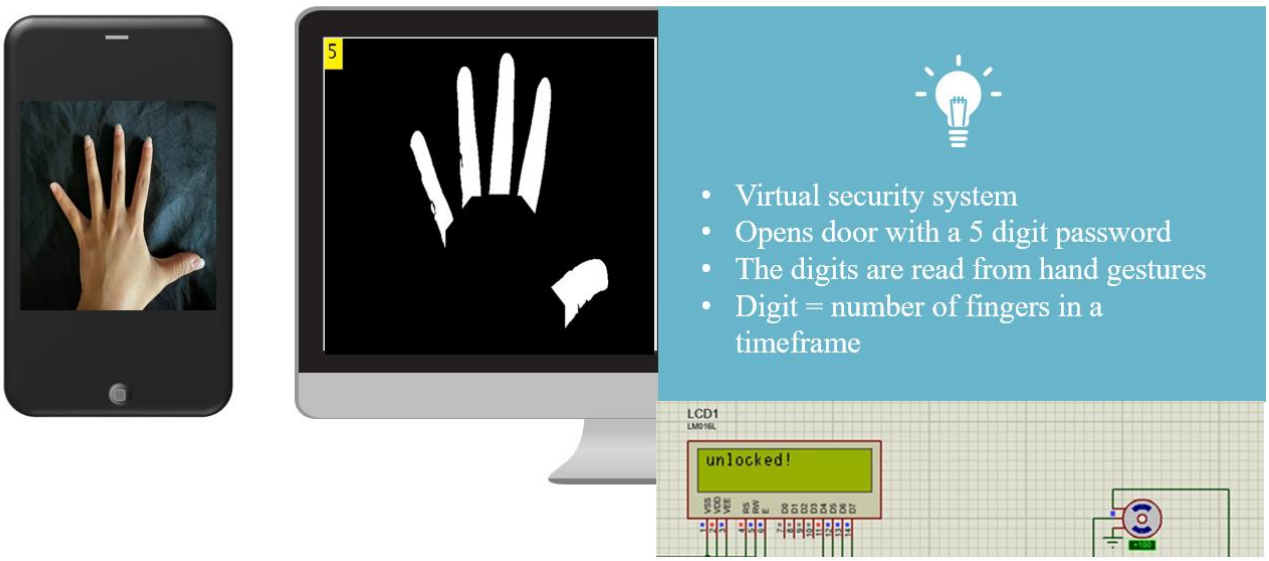


Fig. The images show a snapshot of our hand taken from a mobile phone camera and the image processing output.

The picture above shows a mobile phone acting as an IP camera. It also shows the output of an image processing

MATLAB program showing the number of fingers it has counted. On the bottom right we see that when the password matches with the saved password, the door gets unlocked with the help of a servo motor.

Flowchart of the process:

We have followed a number of steps to ensure maximum accuracy and minimum runtime.

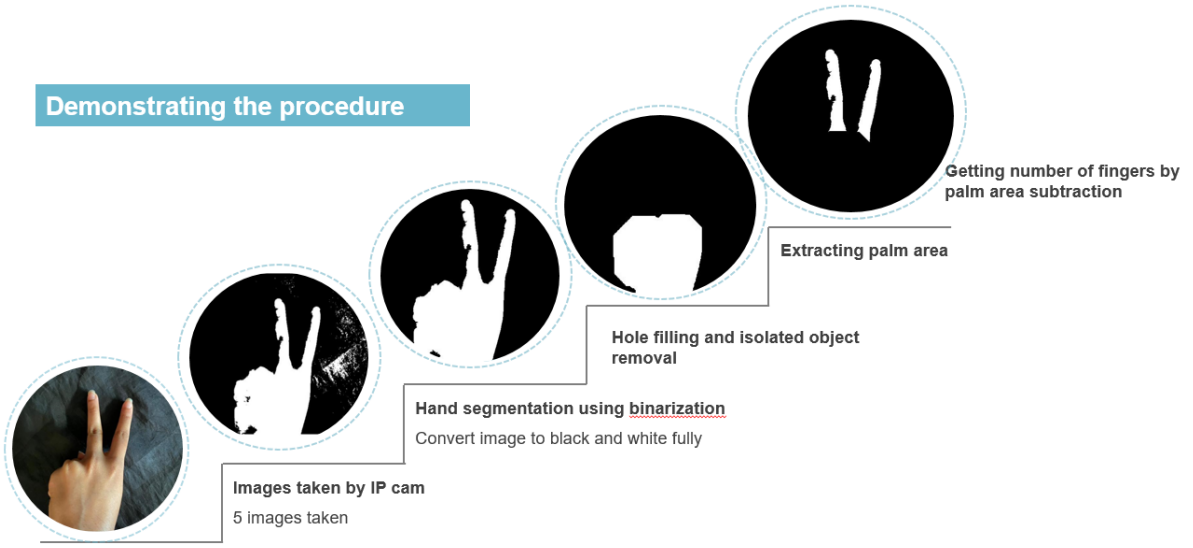


Fig. demonstration of the steps

These steps were carried out to ensure maximum accuracy while counting the number of fingers in an instance by image processing.

Hand Gesture Recognition:

This is an example of HGR used for device control. We use the following Matlab package which enables us to play the IP camera video when the IP address is provided. This enables us to counts number of fingers in real time.

We also downloaded an app in our mobile phones to act as an IP camera.

Fig. Package used

Function: Read password

The main process is carried out in a function returning a set of five digits where each digit denotes the number of fingers shown by the user. The count is taken after equal time intervals of six seconds. The loop ends after five counts.

However, MATLAB and camera must be connected to the same network. Otherwise, we would have been able to run the program where video input is provided from much larger distances.

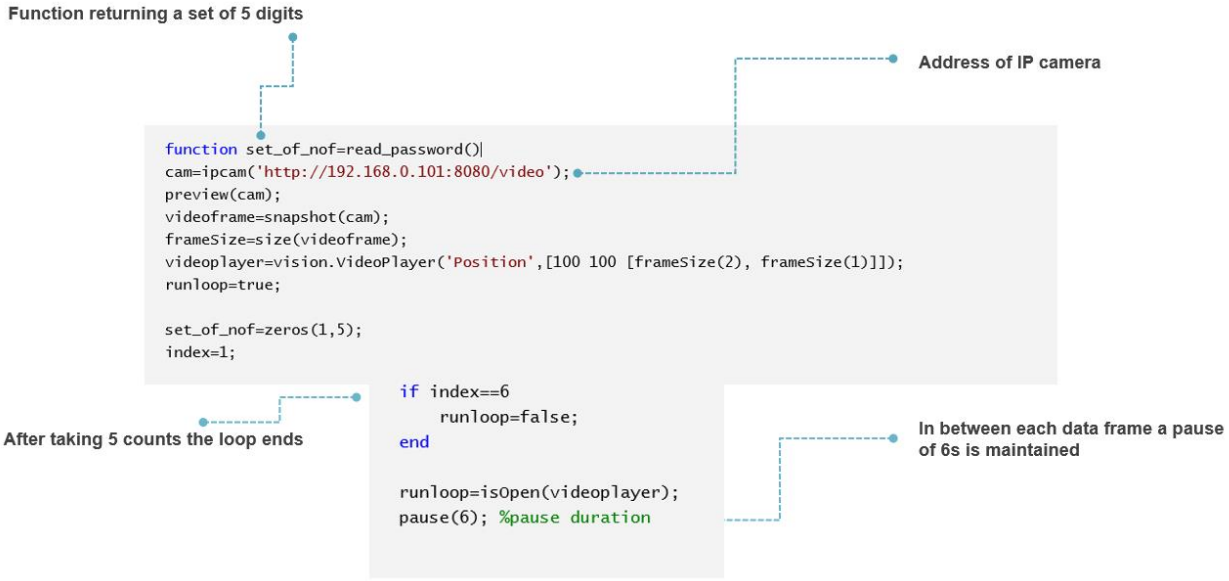


Fig. function read_password

Practical Implementation of this project:

- A series of photo is taken by the IP camera and passed to a CPU connected to the security system
- The MATLAB code is installed in the CPU
- The MATLAB code analyses the input and passes necessary arguments to the microprocessor connected to the door circuit.
- The microprocessor analyses the MATLAB command and unlocks the door if correct password is provided as input.