

A PROJECT REPORT

On

# **GESTURE CONTROLLED DOOR LOCKING SYSTEM**

Submitted in partial fulfillment of the requirement of  
University of Mumbai for

**IoT Mini Project**  
In  
**Information Technology**

Submitted By  
**Manthan Gharat**  
**Midhun VM**  
**Vighnesh Srinivas**  
**Varsha Verma**

Supervisor  
**Sheetal Gawande**



**Department Of Information Technology**  
**PILLAI COLLEGE OF ENGINEERING**  
**New Panvel – 410 206**  
**UNIVERSITY OF MUMBAI**  
**Academic Year 2019 – 20**



DEPARTMENT OF INFORMATION TECHNOLOGY

Pillai College of Engineering

New Panvel – 410 206

## CERTIFICATE

This is to certify that the requirements for the report entitled '**GESTURE CONTROLLED DOOR LOCKING SYSTEM**' have been successfully completed by the following students:

<b>Name</b>	<b>Roll No.</b>
Manthan Gharat	A63
Midhun VM	A802
Vighnesh Srinivas	A73
Varsha Verma	A804

in partial fulfillment of IoT mini Project in the Department of Information Technology, Pillai College of Engineering, New Panvel – 410 206 during the Academic Year 2019 – 2020.

---

**Supervisor**

**Sheetal Gawande**



DEPARTMENT OF INFORMATION TECHNOLOGY

Pillai College of Engineering

New Panvel – 410 206

## PROJECT APPROVAL FOR

This project entitled “Gesture Control Door Lock System” by Manthan Gharat, Midhun VM, Varsha Verma, and Vighnesh Srinivas are approved for the degree of Bachelor of Engineering in Information Technology.

Examiners:

1. \_\_\_\_\_

2. \_\_\_\_\_

3. \_\_\_\_\_

Date:

Place:



DEPARTMENT OF INFORMATION TECHNOLOGY

Pillai College of Engineering

New Panvel – 410 206

## DECLARATION

We declare that this written submission for IoT Mini Project entitled “Gesture Control Door Lock System” represent our ideas in our own words and where others' ideas or words have been included. We have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any ideas / data / fact / source in our submission. We understand that any violation of the above will cause for disciplinary action by the institute and also evoke penal action from the sources which have not been properly cited or from whom prior permission has not been taken when needed.

Project Group Members:

Manthan Gharat:

---

Midhun VM:

---

Varsha Verma:

---

Vighnesh Srinivas:

---

Date:

Place:

# Table of Contents

Abstract.....	i
List of Figures.....	ii
List of Tables.....	iii
<b>1. Introduction.....</b>	<b>1</b>
<b>1.1</b> IoT Introduction.....	<b>2</b>
<b>1.2</b> Literature Survey.....	<b>3</b>
<b>1.3</b> Problem Statement.....	<b>4</b>
<b>1.4</b> Objectives.....	<b>4</b>
<b>1.5</b> Circuit Design & Block Diagram .....	<b>5</b>
<b>1.6</b> H/W and S/W Requirements .....	<b>6</b>
<b>2. Implementation of Mini Project.....</b>	<b>13</b>
<b>2.1</b> Coding implementation of mini project.....	
<b>2.2</b> Snapshots of working of project.....	
<b>2.3</b> Snapshot of Sensor data uploaded using Thingspeak.....	
<b>2.4</b> Snapshot of Project Demo and implementation uploaded on website or blogs	
<b>3 Conclusion and Future Scope</b>	
<b>4 References</b>	
<b>5 Acknowledgement</b>	

## **Abstract**

In public places sanitary conditions are always of concern, particularly of surfaces that are touched by a multitude of persons, such as door handles in rest rooms. Similar issues also arise in medical facilities. Doors that open based on presence are common in environments such as shopping malls; however they are not suited for sensitive areas, such as toilet stalls. Gesture sensors detect and enable gesture-based interfaces that work without touch. In this project, we present a concept for a gesture controlled automated door based on this sensor technology.

The project presents a platform to lock and unlock doors based on the hand gestures that will be programmed by the user which will act as the password to lock and unlock the door and as a secondary feature using face recognition admins can be given access.

## **List of Figures**

Fig. 2.2.1	Locked door recording sensor data	19
Fig 2.2.2	Unlocked door after recording proper password	19
Fig 2.3.1	Creating Channel in ThingSpeak	20
Fig 2.3.2	Creating Twitter React for the data	20
Fig 2.3.3	Graphical Representation of the data received	21
Fig 2.3.4	Tweets created at every door event	21

# **Chapter 1**

## **Introduction of IoT**

### **1.1 Fundamentals**

Today, Internet application development demand is very high. So IoT is a major technology by which we can produce various useful internet applications. Basically, IoT is a network in which all physical objects are connected to the internet through network devices or routers and exchange data. IoT allows objects to be controlled remotely across existing network infrastructure. IoT is a very good and intelligent technique which reduces human effort as well as easy access to physical devices. This technique also has autonomous control feature by which any device can control without any human interaction.



## 1.2 Literature Survey

Gesture controlled user interface (GCUI), Centre for Applied Internet Research (CAIR), Glyndŵr University, Wrexham, UK

Steeven Zeiß, Alexander Marinc, Andreas Braun, Tobias Große-Puppendahl, Sebastian Beck. They describe their approach for gesture controlled door lock system. They use capacitive proximity sensors to detect gestures and open and close the door. Their system is mainly aimed at public restrooms in order to maintain hygiene the system concept and two sets of gestures that can be used to control the door. The system was implemented in prototypical form and evaluated in a study with 16 participants. The results show that the system is intuitive to use and that the subjects would strongly prefer this system to touching door handles in public toilets, indicating that this system is a viable alternative to current solutions.

Sowmiya, U., shafiq mansoor, J. 2015 Raspberry pi based home door security through 3g dongle. Developed to connect any door with internet. In this system user also implemented PIR sensor and camera. PIR sensor used for detecting person and camera used for capturing the video of the person who comes at the door. The video was sent through 3g dongle to authorized person. They had also discussed some advantages of this system. They had concluded that the use of this system in banks, hospitals etc. But their proposed model didn't provide the facility of sending messages to the authorized people.

Senthikumar, G., Gopalkrishnan, K., Sathish Kumar, V. 2014 Embedded Image Capturing System Using Raspberry Pi System. message to the authorized person utilities. G.senthilkumar proposed a work on Embedded Image Capturing System Using Raspberry Pi. In this work, they captured the image and compared it with the database but the limitation was the system couldn't work properly in the ambient light condition

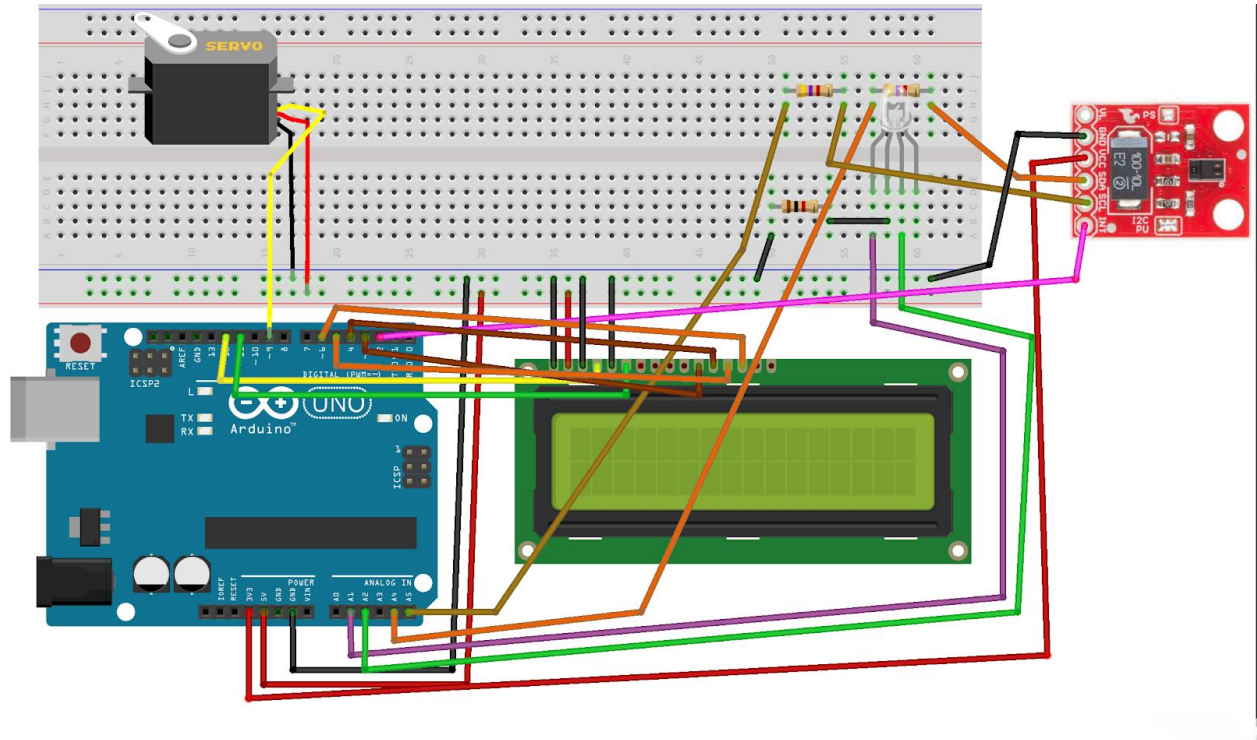
## **1.3 Problem Statement**

The project presents a platform to lock and unlock doors based on the hand gestures that will be programmed by the user which will act as the password to lock and unlock the door and as a secondary feature using face recognition admins can be given access.

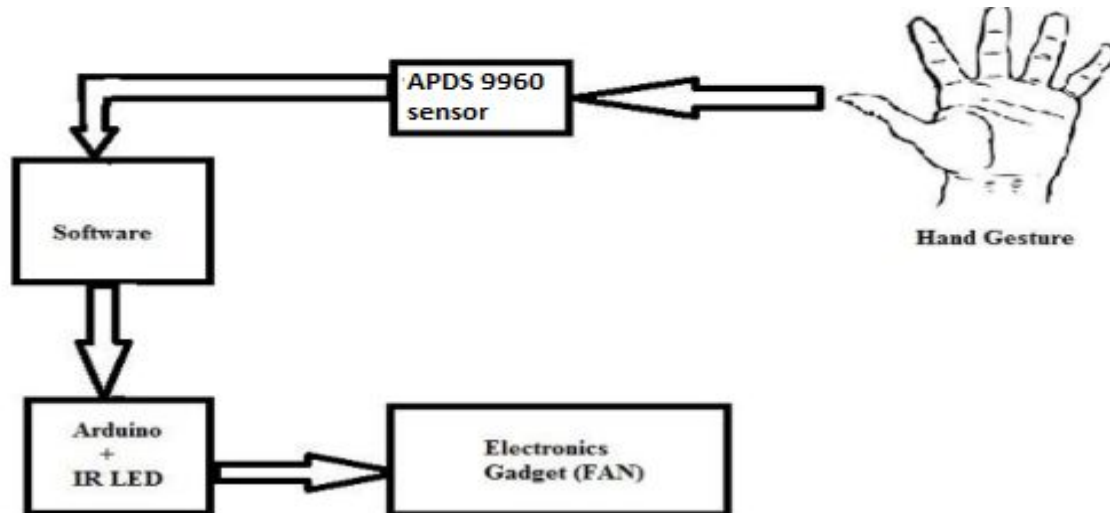
## 1.4 Objectives

- To record a particular pattern of gestures as password
- To compare and detect hand gestures which are predefined and open the door.
- To develop such a system which will help physically impaired to control door by hand gestures.
- This provides comfort and convenience for common as well, especially at home theatre.
- It provides security to documents stored in warehouses using gesture control.

## 1.5 Circuit Design



## BLOCK DIAGRAM



## **1.6 Hardware And Software**

### **HARDWARE**

- Sparkfun Apds 9960
- Breadboard
- Jumper Wires
- Servo Motor
- PIR Motion Sensor
- Arduino Uno

### **SOFTWARE**

- Arduino Uno
- 4GB RAM 32/64 bit Operating System

## Chapter 2

### Implementation of Mini Project

#### 2.1 Coding implementation of mini Project

Arduino Pin APDS-9960 Board Function

3.3V	VCC	Power
GND	GND	Ground
A4	SDA	I2C Data
A5	SCL	I2C Clock
2	INT	Interrupt

Resources:

Include Wire.h and SparkFun\_APDS-9960.h

Development environment specifics:

Written in Arduino 1.0.5

Tested with SparkFun Arduino Pro Mini 3.3V

This code is beerware; if you see me (or any other SparkFun employee) at the local, and you've found our code helpful, please buy us a round!

Distributed as-is; no warranty is given.

\*\*\*\*\*/

```
#include <Wire.h>
#include <SparkFun_APDS9960.h>
#include <Servo.h>
```

```
Servo myservo;
```

```
// Pins
#define APDS9960_INT 2 // Needs to be an interrupt pin
```

```
// Constants
```

```
// Global Variables
```

```

SparkFun_APDS9960 apds = SparkFun_APDS9960();
int isr_flag = 0;
int sub_flag1=0;
int sub_flag2=0;
int sub_flag3=0;
int sub_flag4=0;
int sub_flag5=0;
int sub_flag6=0;
int master_flag=0;

void setup() {

  myservo.attach(9); // attaches the servo on pin 9 to the servo object
  // Set interrupt pin as input
  pinMode(APDS9960_INT, INPUT);

  // Initialize Serial port
  Serial.begin(9600);
  Serial.println();
  Serial.println(F("-----"));
  Serial.println(F("Perform Your Gesture"));
  Serial.println(F("-----"));

  // Initialize interrupt service routine
  attachInterrupt(0, interruptRoutine, FALLING);

  // Initialize APDS-9960 (configure I2C and initial values)
  if ( apds.init() ) {
    Serial.println(F("APDS-9960 initialization complete"));
  } else {
    Serial.println(F("Something went wrong during APDS-9960 init!"));
  }

  // Start running the APDS-9960 gesture sensor engine
  if ( apds.enableGestureSensor(true) ) {
    Serial.println(F("Gesture sensor is now running"));
  } else {
    Serial.println(F("Something went wrong during gesture sensor init!"));
  }
}

void loop() {
  if( isr_flag == 1 ) {

```



```

detachInterrupt(0);
handleGesture();
isr_flag = 0;
attachInterrupt(0, interruptRoutine, FALLING);
}
}

```

```

void interruptRoutine() {
  isr_flag = 1;
}

```

```

void handleGesture() {
  if ( apds.isGestureAvailable() ) {
    switch ( apds.readGesture() ) {
      case DIR_UP:
        Serial.println("UP");
        sub_flag1=1;
        break;
      case DIR_DOWN:
        Serial.println("DOWN");
        sub_flag2=1;
        break;
      case DIR_LEFT:
        Serial.println("LEFT");
        sub_flag3=1;
        break;
      case DIR_RIGHT:
        Serial.println("RIGHT");
        sub_flag4=1;
        break;
      case DIR_NEAR:
        Serial.println("NEAR");
        sub_flag5=1;
        break;
      case DIR_FAR:
        Serial.println("FAR");
        sub_flag6=1;
        break;
      default:
        Serial.println("NONE");
    }
  }
}

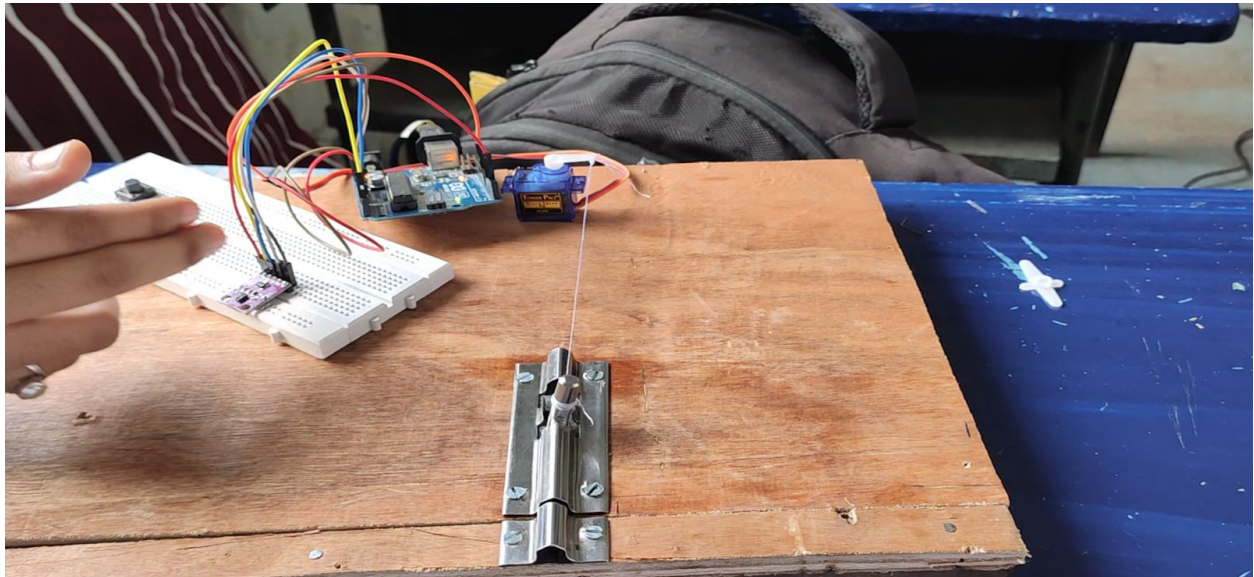
```

```
if( (sub_flag1==1) && (sub_flag2==1) && (sub_flag3==1) && (sub_flag4==1) && (sub_flag5==1)
&& (sub_flag6==1))
{
    master_flag=1;
}
if (master_flag==1)
{
    myservo.write(180);
    //delay(1000);
    myservo.write(180);
    delay(1000);

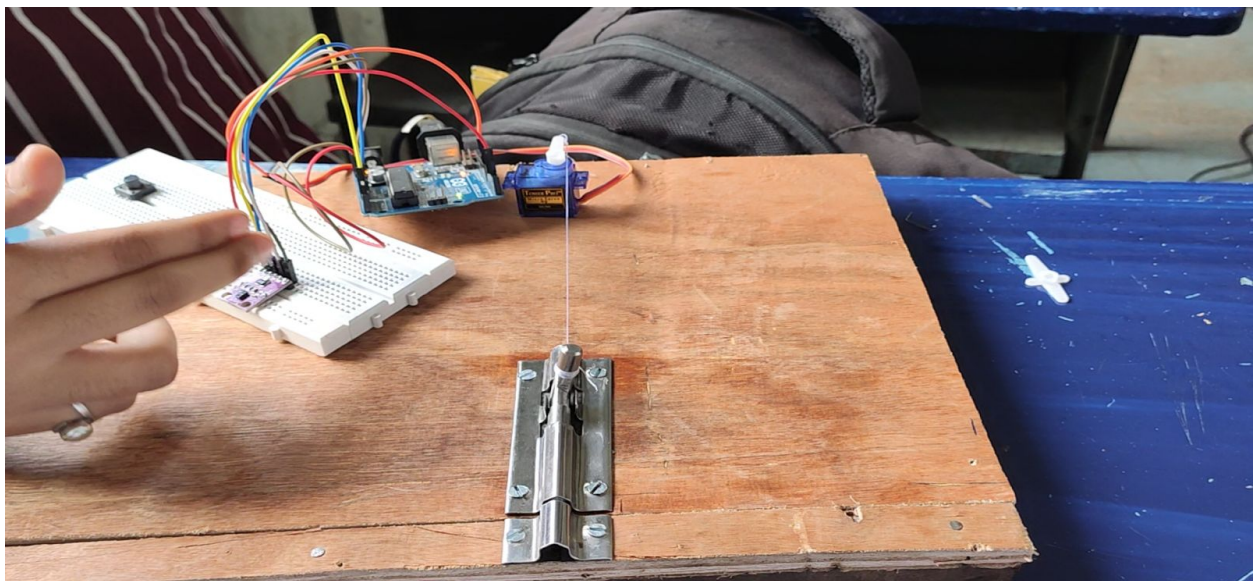
}

}
//baud rate=9600
```

## 2.2 Snapshots of working project



**Fig 2.2.1**



**Fig 2.2.2**

## 2.3 Snapshots of Sensor data uploaded using Thingspeak

The screenshot shows the 'New Channel' form in the Thingspeak web interface. The form is divided into two main sections: 'New Channel' on the left and 'Help' on the right. The 'New Channel' section contains fields for 'Name' (filled with 'door'), 'Description' (filled with 'Someone at the door'), and eight 'Field' entries. 'Field 1' is filled with 'dooropen' and has a checkbox checked. The other fields are empty. Below the fields are 'Metadata' and 'Tags' sections. The 'Help' section contains a paragraph about channels and a 'Channel Settings' section with a list of settings: Channel Name, Description, Field, Metadata, Tags, Link to External Site, Show Channel Location (with sub-items for Latitude, Longitude, and Elevation), Video URL, and Link to GitHub.

ThingSpeak™ Channels Apps Support Commercial Use How to Buy Account Sign Out

### New Channel

Name:

Description:

Field 1:  ☒

Field 2:  ☐

Field 3:  ☐

Field 4:  ☐

Field 5:  ☐

Field 6:  ☐

Field 7:  ☐

Field 8:  ☐

Metadata:

Tags:

### Help

Channels store all the data that a Thingspeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use Thingspeak apps to analyze and visualize it.

#### Channel Settings

- **Channel Name:** Enter a unique name for the Thingspeak channel.
- **Description:** Enter a description of the Thingspeak channel.
- **Field:** Check the box to enable the field, and enter a field name. Each Thingspeak channel can have up to 8 fields.
- **Metadata:** Enter information about channel data, including JSON, XML, or CSV data.
- **Tags:** Enter keywords that identify the channel. Separate tags with commas.
- **Link to External Site:** If you have a website that contains information about your Thingspeak channel, specify the URL.
- **Show Channel Location:**
  - **Latitude:** Specify the latitude position in decimal degrees. For example, the latitude of the city of London is 51.5072.
  - **Longitude:** Specify the longitude position in decimal degrees. For example, the longitude of the city of London is -0.1275.
  - **Elevation:** Specify the elevation position meters. For example, the elevation of the city of London is 35.052.
- **Video URL:** If you have a YouTube™ or Vimeo® video that displays your channel information, specify the full path of the video URL.
- **Link to GitHub:** If you store your Thingspeak code on GitHub®, specify the GitHub repository URL.

Fig 2.3.1

The screenshot shows the 'React' app configuration page in the Thingspeak web interface. The page is divided into two main sections: 'React' on the left and 'Help' on the right. The 'React' section contains a table with the following information: Name (Motion1), Condition Type (Numeric), Test Frequency (On data insertion), Last Ran (2019-10-24 21:32), Channel (door), Condition (Field 1 (dooropen) is equal to 1), ThingTweet (kuroryuu10: %%datetime%%Motion detected), Run (Each time the condition is met), and Created (2019-10-24 8:30 pm). The 'Help' section contains a paragraph about the React app and a 'Learn More' link.

ThingSpeak™ Channels Apps Support Commercial Use How to Buy Account Sign Out

Apps / React / Motion1

[Edit React](#)

Name:	Motion1
Condition Type:	Numeric
Test Frequency:	On data insertion
Last Ran:	2019-10-24 21:32
Channel:	door
Condition:	Field 1 (dooropen) is equal to 1
ThingTweet:	kuroryuu10: %%datetime%%Motion detected
Run:	Each time the condition is met
Created:	2019-10-24 8:30 pm

### Help

React works with [ThingHTTP](#) and [ThingTweet](#) to perform actions when channel data meets a certain condition. For example, you can have a mobile app report your latitude and longitude to a Thingspeak channel. When your position is within a certain distance of your house, have ThingHTTP turn on your living room lights.

[Learn More](#)

Fig 2.3.2



Fig2.3.3

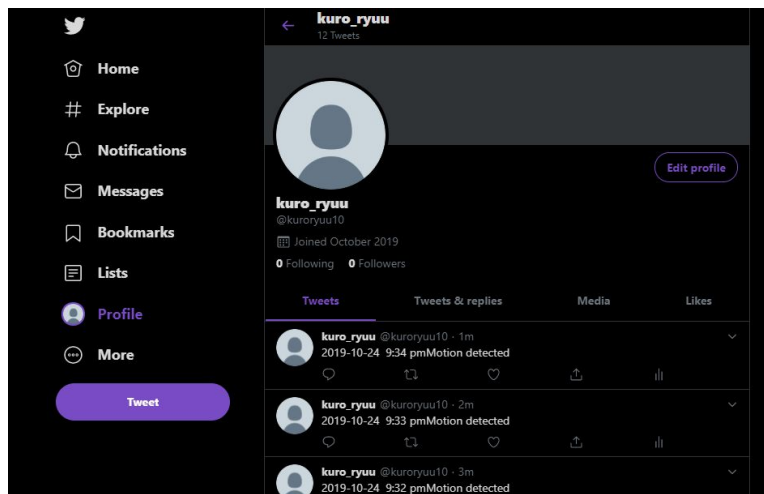


Fig 2.3.4

## **2.4 Snapshot of Project Demo and implementation uploaded on website or blogs:**

**Link: <https://youtu.be/yi-st9kR15c>**

## **Chapter 3**

### **Conclusion & Future Scope**

In this paper we presented a gesture-based door control system based on gesture sensor that allows controlling doors in public spaces without touching any surface. We have provided the system concept and a sets of gestures that can be used to open the door. The results show that the system is intuitive to use and that it is a more preferable to use this system than to touch

door handles in public toilets, indicating that this system is a viable alternative to current solutions.

However, our system signifies only a first step in this direction and there are numerous improvements that can be applied to future iterations. In the future, we plan to integrate additional sensors that measure the state.

Another addition to the system is providing a better visual feedback. It is particularly interesting to display whether a door is closed or open. We are considering using a LCD display that could provide an iconic representation of the system state if an error has occurred or if the hand is obstructing the gesture area. This feedback should ideally be available on both sides of the door.

## References

Gesture controlled user interface (GCUI), Centre for Applied Internet Research (CAIR), Glyndŵr University, Wrexham, UK

Steeven ZeiB, Alexander Marinc, Andreas Braun, Tobias Große-Puppendahl, Sebastian Beck.

They describe their approach for gesture controlled door lock system. They use capacitive proximity sensors to detect gestures and open and close the door. Their system is mainly aimed at public restrooms in order to maintain hygiene the system concept and two sets of gestures that can be used to control the door. The system was implemented in prototypical form and evaluated in a study with 16 participants. The results show that the system is intuitive to use and that the subjects would strongly prefer this system to touching door handles in public toilets, indicating that this system is a viable alternative to current solutions.

Senthikumar, G., Gopalkrishnan, K., Sathish Kumar, V. 2014 Embedded Image Capturing System Using Raspberry Pi System. message to the authorized person utilities. G.senthilkumar proposed a work on Embedded Image Capturing System Using Raspberry Pi. In this work, they captured the image and compared it with the database but the limitation was the system couldn't work properly in the ambient light condition

Sowmiya, U., shafiq mansoor, J. 2015 Raspberry pi

based home door security through 3g dongle. Developed to connect any door with internet. In this system user also implemented PIR sensor and camera. PIR sensor used for detecting person and camera used for capturing the video of the person who comes at the door. The video was sent through 3g dongle to authorized person. They had also discussed some advantages of this



system. They had concluded that the use of this system in banks, hospitals etc. But their proposed model didn't provide the facility of sending messages to the authorized people.

## Acknowledgement

We are thankful to all who provide us opportunity to complete this report. We give a special gratitude to our wonderful guide **Prof. Mimi Cherian** and **Prof. Sheetal Gawade**, who always encouraged and motivated us to do innovative things that will increase our knowledge. We would like to thank her for her generous support throughout the project. We would like to express a humble gratitude towards her for always being there and solving our queries.

We would like to show our gratitude to our Head of the Department of Information Technology **Dr. Satishkumar Verma**, for providing us with such an intriguing and informative project as well as for the good guidelines and consultations.

We would like to expand our gratitude to our principal sir **Dr. Sandeep Joshi** for letting us access all the resources efficiently for the development of our project.

Manthan Gharat

Midhun VM

Vighnesh Srinivas

Varsha Verma