# Android Controlled Home Automation System

*A report submitted in partial fulfilment of the requirements degree of*

Bachelor of Technology

*in*

## Computer Science and Engineering

*by*

# Biswajit Singh and Biswajit Bajaj

*Under the supervision of*

## PROF. Eekian Wong

**Department of Computer Science Engineering**

**Institute of Engineering & Management**

****

Department of Electronics and Communication Engineering

Institute of Engineering & Management

Gurukul, Y-12, Block –EP, Sector-V, Salt Lake Electronics Complex Kolkata-700091, West Bengal, India

|  |  |  |
| --- | --- | --- |
|  | **Table of Contents** | |
| Acknowledgement............................................................... | | i |
| Abstract....................................................................... | | .…..ii |
| 1. Introduction | |  |
| 1.1 | **Overview**........................................................... | 1 |
| 1.2 | Project Objectives……...... | ……….….....................2 |

* 1. Project Scope…………...………………................3
  2. Technology exposure that project provides……....4

1. Background Theory…………………………….........5

|  |  |  |  |
| --- | --- | --- | --- |
| 2.1 Hardware & Programming Language Description..... | | 6-9 |  |
|  | |  |  |
| 2.1 | .1 Hardware Description…………………………....6-8 | |  |
| 2.1.2 | Software Description…………………………..... | 9 |  |
|  |  |  |  |

3.Requirements------------------------------------------------10

4.Design---------------------------------------------------11-12

5.Hardware Schematic ----------------------------------13-14

6.App Description----------------------------------------15-16

|  |  |  |
| --- | --- | --- |
|  |  |  |
|  |  |  |
|  |  |  |

7. Implementation-----------------------------------------------17-29

8. Testing ---------------------------------------------------------30-33

9. Conclusion----------------------------------------------------34

10. References------------------------------------------------------34

**Acknowledgement**

This major project would not have been possible without the valuable assistance of many people to whom we are indebted, in particular, our project coordinator Eekian Wong of Institute of Engineering and Management

We would also like to thank “Department of Computer Science”, IEM Kolkata for

providing us with the necessary components for our project. Our thank also goes to

all the teachers of Computer Science Department who helped us in many

difficult situations regarding the project and provided with the necessary advice.

A special word of thanks is to our class mates and our families for providing us the moral support. A word of thanks to our friends Manish Jindal , Biswadeep Sikder, who helped us in many ways.

i

**Abstract**

In modern times, people prefer more of automatic systems rather than manual systems.

With the Influence of Internet in people’s life lots of new technologies are coming up.

One of the latest, emerging and trending technologies is the ‘Internet of Things’. This

technology is expected to rule the world within a few years. Home Automation System

uses the technology of Internet of Things for monitoring and controlling of the electrical

and electronic appliances at home from any

remote location by simply using a Smartphone. Implementation of a low cost, flexible

home

automation system is presented. It enhances the use of wireless communication which

provides the user with remote control of various electronic and electrical appliances

ii

1

1. **Introduction**

**1.1 Overview**

Overview Homes of the 21st century will become more and more self controlled and automated due to the comfort it provides, especially when employed in a private home. A home automation system is a means that allow users to control electric appliances of varying kind. Many existing, well-established home automation systems are based on wired communication. This does not pose a problem until the system is planned well in advance and installed during the physical construction of the building. But for already existing buildings the implementation cost goes very high. In contrast, Wireless systems can be of great help for automation systems. With the advancement of wireless technologies such as Wi-Fi, cloud networks in the recent past, wireless systems are used every day and everywhere.

2

**1.2 Project Objectives :**

The Internet of Things (IoT) is the inter networking of physical devices ,vehicles,buildings, and other items embedded with electronics,, software, sensors ,acutators,and network connectivity which enables these objects to collect and exchange data .

We tried to work on IoT as a part of final year Project . Home Automation is one of the most widely used applications of IoT . IoT enables us to make home automation products which make lives of people a lot simpler.

In our Iot based project we tried to explore the area of home automation making use of Rapberry pi and Arduino . this is a simple prototype of what the actual product would represent .

Our product notifies the user whenever someone is standing infront the users door.it captures the images of the person standing infront the door and sends it to the user . The user on receiving the photos of the person can control the opening and closing of the door as he wishes with the help of an android app .

**3**

**1.3. Scopes**

The project aims at designing a prototype for controlling the home appliances that can

be controlled wirelessly via a mobile application . The application is run on any mobile device having bluetooth connectivity. The system can be used in wide range of areas.

This system is designed to assist and provide support in order to fulfill the

needs of elderly and disabled in home. Household appliances can be easily controlled via a Mobile/Tablet. Status of light, fan and other electrical appliances can be known.

The system integrated with different features can be applied in the following fields.

* **The system can be used in homes to small offices**

The system can be used from home to offices to control the Doors according to the users needs.

* + **For the development of technology friendly environment**

The system incorporates the use of technology and making smart home automation. By the use of day to day gadgets we can utilize them for different prospective.

4

1. **Technology Exposures That Project Provides:**
2. Google’s Android open source technology.
3. Interfacing cameras and sensors to the Raspberry Pi.
4. Interfacing Bluetooth Module to the Arduino.
5. Embedded programming.

5

**2. Background**

The “Home Automation” concept has existed for many years. The terms “Smart Home”, “Intelligent Home” followed and has been used to introduce the concept of networking appliances and devices in the house. Home automation Systems (HASs) represents a great research opportunity in creating new fields in engineering, and Computing. HASs includes centralized control of lighting, appliances, security locks of gates and doors and other systems, to provide improved comfort, energy efficiency and security system. HASs becoming popular nowadays and enter quickly in this emerging market. However, end users, especially the disabled and elderly due to their complexity and cost, do not always accept these systems.

Due to the advancement of wireless technology, there are several different of connections are introduced such as GSM, WIFI, and Bluetooth. Each of the connection has their own unique specifications and applications. Among the four popular wireless connections that often implemented in HAS project, we have made use of Bluetooth with its suitable capability.

This project forwards the design of home automation and security system using Raspberry pi, a credit sized computer and Arduino . Raspberry pi provides the features of a mini computer, additional with its GPIO pins where other components and devices can be connected. GPIO registers of raspberry pi are used for the output purposes. We have design a power strip that can be easily connected to GPIO Pins of the Raspberry pi. The home appliances are connected to the input/output ports of Raspberry pi and the Arduino . The android running OS in any phone connected to a Bluetooth Device can access the status of the home appliances via an application. It presents the design and implementation of automation system that can monitor and control home appliances via android phone or tablet.

**6**

**2.1. Hardware and Programming Language Description**

**2.1.1. Hardware Description:**

**Raspberry pi –**

The Raspberry Pi is a series of small single-board computers developed in the United

Kingdom by the Raspberry Pi Foundation to promote the teaching of basic computer

science in schools and in developing countries. The original model became farmore

popular than anticipated,[7] selling outside of its target market for uses such as robotics. Peripherals are not included with the Raspberry Pi. Some accessories however have been included in several official and unofficial bundles.



7

**Arduino :**

Arduino is an open source, computer hardware and software company, project,

and user community that designs and manufactures single-board

microcontrollers and microcontroller kits for building digital devices and

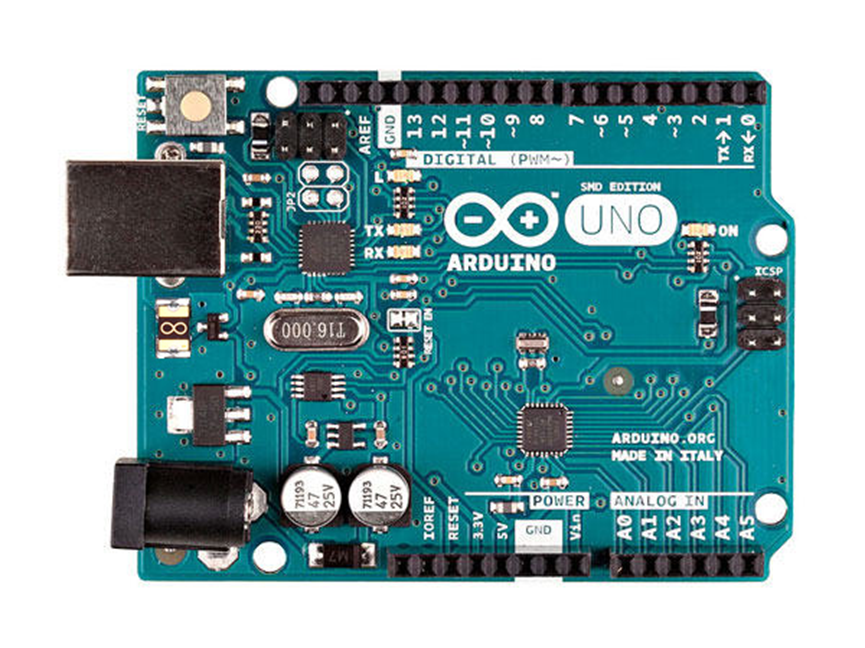
interactive objects that can sense and control objects in the physical world. The

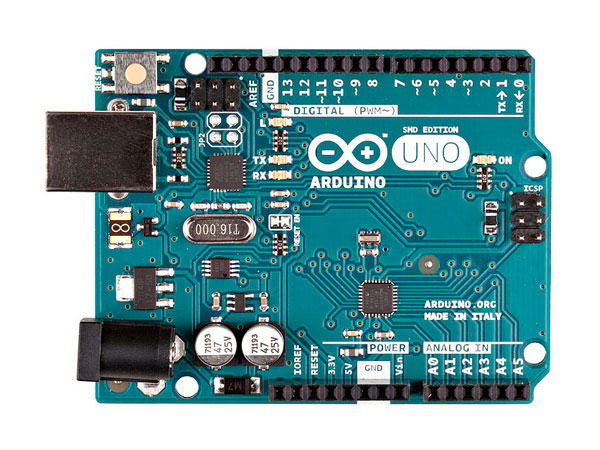
project's products are distributed as open-source hardware and software, which

are licensed under the GNU Lesser General Public License (LGPL) or the GNU

General Public License (GPL),[1] permitting the manufacture of Arduino boards

and software distribution by anyone





**Ultrasonic sensors:** 8

ultrasonic sensors are used for detecting the presence of an obstacle . they detect an

obstacle by the means of producing ultrasonic sound and computing the distance of

the object from the obstacle.



**Servomotor:** we made use of servomotor in out project to implement

gate control. The Arduino is connected to the servomotor . when the arduin receives

information from the android device it rotates the servomotor.

 9

**2.1.2. Software Description**

We have used two different programming languages for our project. For the development of the application on android, we have used Java Platform. Android Software Development kit incorporates Eclipse software where Java programming is performed.

Eclipse software is used to write the codes for the application under Java Platform. Raspbian OS is used at the raspberry pi. Server is established at raspberry pi. Python Language is used to write the codes of server, and to control the GPIO Pins of OS.

**5.2.1. Java:**

Java is a set of several computer software products and specifications from Oracle Corporation that provides a system for developing application software and deploying it in a cross-platform computing environment. Java is used in a wide variety of computing platforms from embedded devices and mobile phones on the low end, to enterprise servers and supercomputers on the high end.

**5.2.2. Python:**

Python is an interpreter, interactive, object-oriented programming language. It incorporates modules, exceptions, dynamic typing, very high level dynamic data types, and classes. Python combines remarkable power with very clear syntax. It has interfaces too many system calls and libraries, as well as to various window systems, and is extensible in C or C++

**5.2.3. Raspbian:**

Raspbian is a free Operating System based on Debian optimized for the raspberry pi hardware. Raspbian comes with more than 35000 packages; pre-combined software bundled in a nice format for easy installation on Raspberrypi

5.2.4

**C: C**  Is  general-purpose,  imperative computer  programming language, supporting structured programming, lexical variable scope and recursion.

**3. Requirements.** 10

Android controlled Smart Home Automation should be able to control the home appliances wirelessly with effectively and efficiently.

**1.Detecting the presence of a person infront the door**

Our product makes use of ultrasonic sensors to detect a human within a specified range

**2.Capturing photos of the person**

One of the main objectives of our project was to enable our product to successfully

capture photos of the person standing infront the door

**3.Sending push notifications to the user**

This was one of the primary objectives our our project as well. we had to enable our

product to send push notifications to the user. Our product uses python inbuilt email

package to send photos of the captured person as attachments.

**4.Controlling the Door via Application**

We made an android app that allows the user to control the opening and closing of the door of his/her house .

**5.Extensible platform for future enhancement**

The application is to be highly extensible, with possibility of adding features in the future as needed.

11

**4.Design :**

In this project we tried to build a home automation product which allows

its users to see who in infront the door of there house and control the opening/closing

of the door using an app. We used Raspberry

Pi , Arduino and Google’s Android open source technology to come up

with the product. In this section we are going to take a brief look at the

proposed system for our project which describes the working of our

product.

We wanted to design our product in such a way that it would

send pictures of the person to the owner of the house whenever he/she

would stand infront the users gate hence allowing the user to control the

gate via a phone as per his/her wish. So we can see that our system needs

to capture photos of a person whenever it comes infront the gate. Its quite

clear that distance is a major constraint which we have to take care of

here.

**Ultrasonic Sensor :**

To achieve the following we made use of an ultrasonic sensor .

ultrasonic sensor detects the presence of an object in a given range by

producing ultrasonic sound . we attached the ultrasonic sensor to the gpio

pins of the raspberry pi and made it work in a way such that it would be

able to detect objects in distances within 100 cm .

**Pi Camera:**

For the purpose of

capturing photos we needed a camera module . raspberry pi comes with

an inbuilt camera module so we didn’t make use of any external camera.

We used the pi camera and coded it in a way such that it would be

triggered to capture photos whenever the sensor detects an object.

12

**Notification:**

Now we needed to transfer the captured photos to the users phone . so

we made use of python’s email package which sends those photos to as

attachments to the user in the form of a push notification . once the

user receives these photos he would control the opening of the gate using

his phone .

we made an android app which fulfils this purpose .

**Arduino and Servomotor :**

For making the gate we made use of Arduino whose pins have been

connected to the servomotor .We have used arduino’s Bluetooth module

to make it discoverable . when the Arduino receives information from

the mobile device it rotates the servo hence resulting in the opening and

closing of the gate fulfilling the purpose of our project.

13

**5.HARDWARE SCHEMATIC**

In this section we are going to have a look at the hardware schematic for a better

understanding of our project .

Firstly we worked with sensors . the pins of the ultrasonic sensors were connected

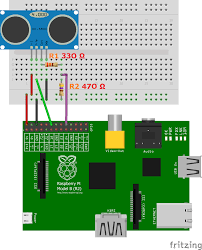
to the gpio pins of the raspberry pi .there are mainly 4 pins we deal with in ultrasonic

sensors namely the vcc,gnd,trig,echo. The vcc of the sensor is connected to the 5v of

the pi . the gnd pin goes into the gnd pin of the pi . the trigger pin the connected to the

pin number 23 and the echo pin is connected to the pin number 24 . the following

diagram represents the connection of sensors with the raspberry pi



After working with sensors we started working on the image capture part. We used a pi camera for the purpose of capturing photos . the raspberry pi comes with an inbuilt camera module so we didn’t make any use of an external camera . given below is an image of a pi camera .



14

When the image gets transferred to the user phone the user can control the door using

an app. For the purpose of gate control we made use of an Arduino and a servomotor.

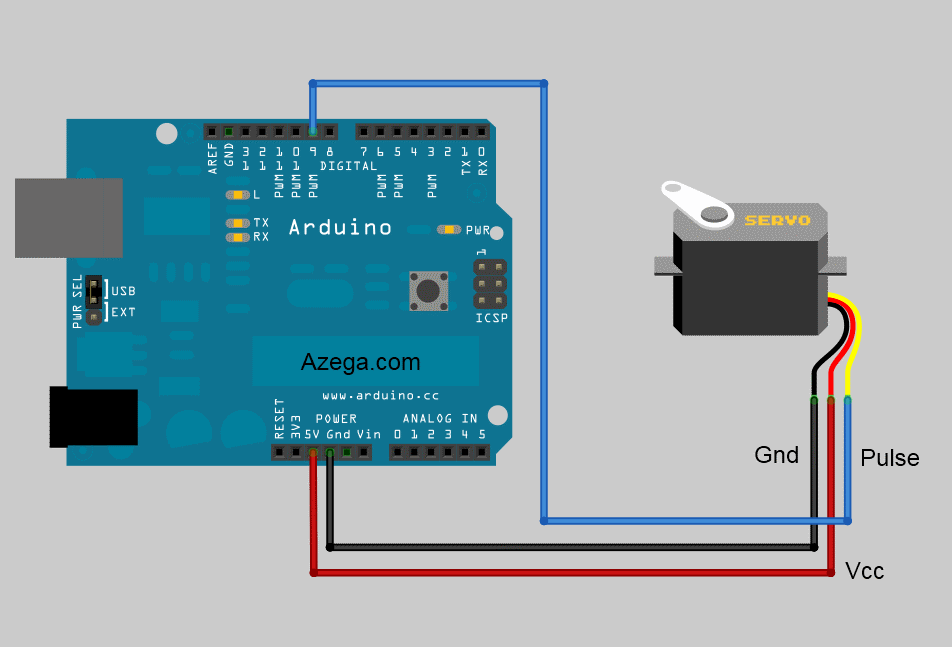
The pins of the servomotor are connected to the Arduino . whenver the Arduino

receives information for gate control it will rotate the servo . the vcc of the servo is

connected to the 5V . the gnd is connected to the gnd of Arduino .and the pulse pin is

connected to the pin number 12. Below mentioned is a digram which represents the

hardware connection between the servo and the Arduino

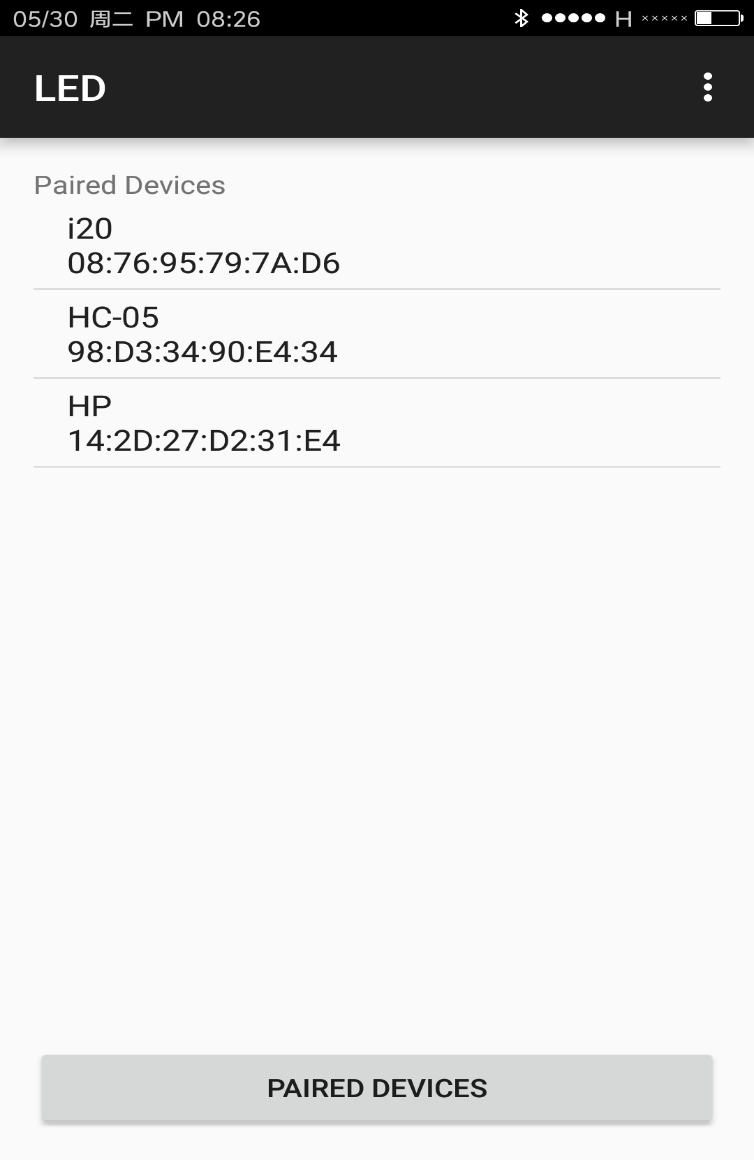


15

**6. Application Description:**

Application Consists of Graphical User Interfaces. It consists of following different activities.

1. **Device List activity** : this activity allows the user to see a list of Bluetooth devices in range . It does so by storing the Bluetooth devices in an Array list and displaying the Array list to the user.

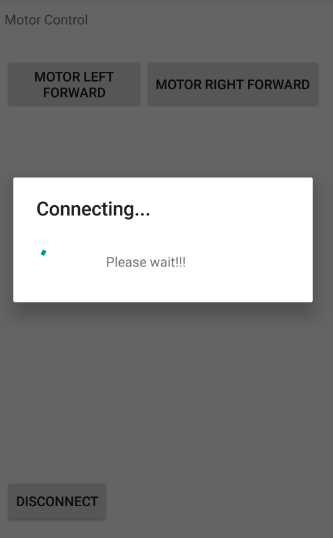


16

2.**Trasmission Activity** : this activity allows the android device to communicate

with the Arduino through the Bluetooth module. Bluetooth Socket is used for the

bidirectional half duplex pathway for character passing .



17

**7.Implementation/Coding**

In this section we are going to have a look at the implementation phase for our project

1)First we started working with the ultrasonic sensor .it makes use of sound waves to detect the presence of an obstacle .below mentioned code snippet shows how the sensor computer the distance of the obstacle from it .

while True:

GPIO.output(TRIG, False)

print("Waiting For Sensor To Settle")

time.sleep(2)

GPIO.output(TRIG, True)

time.sleep(0.00001)

GPIO.output(TRIG, False)

while GPIO.input(ECHO)==0:

pulse\_start = time.time()

while GPIO.input(ECHO)==1:

pulse\_end = time.time()

**pulse\_duration = pulse\_end - pulse\_start**

**distance = pulse\_duration \* 17150**

**distance = round(distance, 2)**

18

2) In this section we started working on image transfer . the pi camera basically captures images when triggered by the sensor and store those images as .jpg files in the desktop .

This creates a camera object .

Camera=piCamera()

Picamera is activated to capture photos if any object comes within a distance of 10 cm.

if **distance < 10:**

**print("Distance:",distance,"cm")**

**sleep(5)**

**camera.capture('/home/pi/Desktop/image'+str(cnt)+'.jpg'**)

cnt = cnt+1

19

3) In this section we worked on sending push notifications to the user . once the photos were captured by the pi our main objective was to send those photos to the user in the form of a notification . one of the simplest forms in which notifications can be sent is through email. So we decided to use pythons inbuilt email package and send those photos as attachments to the user . we make use of **SMTP protocol** which handles sending emails and routing email between mail servers. Python provides an **smtplib** module which defines an SMTP client session object that can be used to send mail to any internet machine with an SMTP .

// we define the variables as per our requirement here

**emailfrom = "singhbiswajit79@gmail.com"**

**emailto = "biswajitsingh27@yahoo.com"**

**fileToSend = "ronaldo.jpg"**

**fileToSend1="big.jpg"**

**username = "singhbiswajit79@gmail.com"**

**password = "biswajitsingh28"**

// we define the message project thereafter

**msg = MIMEMultipart()**

**msg["From"] = emailfrom**

**msg["To"] = emailto**

**msg["Subject"] = "help I cannot send an attachment to save my life"**

**msg.preamble = "help I cannot send an attachment to save my life"**

#1

ctype, encoding = mimetypes.guess\_type(fileToSend)

if ctype is None or encoding is not None:

ctype = "application/octet-stream"

maintype, subtype = ctype.split("/", 1)

20

//the following snippet identifies the type of media file we want to transfer .

**if maintype == "text":**

**fp = open(fileToSend)**

**# Note: we should handle calculating the charset**

**attachment = MIMEText(fp.read(), \_subtype=subtype)**

**fp.close()**

**elif maintype == "image":**

**fp = open(fileToSend, "rb")**

**attachment = MIMEImage(fp.read(), \_subtype=subtype)**

**fp.close()**

**elif maintype == "audio":**

**fp = open(fileToSend, "rb")**

**attachment = MIMEAudio(fp.read(), \_subtype=subtype)**

**fp.close()**

**else:**

**fp = open(fileToSend, "rb")**

**attachment = MIMEBase(maintype, subtype)**

**attachment.set\_payload(fp.read())**

**fp.close()**

**encoders.encode\_base64(attachment)**

**attachment.add\_header("Content-Disposition", "attachment", filename=fileToSend)**

21

here we attach our photos to the message object

**msg.attach(attachment**)

here we can see that the message object gets converted to a string and passed as a parameter . For sending the mail, we have to convert the object to a string, and then

use the same prodecure as above to send using the SMTP server..

**server = smtplib.SMTP("smtp.gmail.com:587")**

**server.starttls()**

**server.login(username,password)**

**server.sendmail(emailfrom, emailto, msg.as\_string())**

**server.quit()**

22

4) In this section we implemented the app which would control our gate . the app

basically

consists of two activities namely **device list activity** and **transmission activity**. The

device list activity shows the Bluetooth enabled devices in range in the form of a list view

.the Bluetooth enable devices get stored inside the arraylist and it is thereafter displayed

as a list view to the user . the transmission activity basically enables the android device to

communicate with the arduino’s Bluetooth module via **Bluetooth sockets** . Bluetooth

Socket is used for the bidirectional half duplex pathway for character passing . On

receiving the information from the android device the Arduino rotates the servomotor

hence resulting in the opening closing of the gate .

Following is our Device lsit activity

**public class DeviceList extends ActionBarActivity {**

//in the following code snippet we define the variables needed for our activity below .

**Button btnPaired;**

**ListView devicelist;**

**private BluetoothAdapter myBluetooth = null;**

**private Set<BluetoothDevice> pairedDevices;**

**public static String EXTRA\_ADDRESS = "device\_address**";

@Override

20

protected void onCreate(Bundle savedInstanceState) {

super.onCreate(savedInstanceState);

setContentView(R.layout.activity\_device\_list);

// the view objects are getting linked to view objects present in xml layout files

23

**btnPaired = (Button)findViewById(R.id.button);**

**devicelist = (ListView)findViewById(R.id.listView**);

// the following code snippet uses Bluetooth adapter to see if the device supports Bluetooth connectivity . if it doesn’t support Bluetooth connectivity it displays a toast with text showing bluettoth device is not available . //

**//if the device has bluetooth**

**myBluetooth = BluetoothAdapter.getDefaultAdapter();**

**if(myBluetooth == null) {**

**Toast.makeText(getApplicationContext(), "Bluetooth Device Not Available", Toast.LENGTH\_LONG).show();**

**//finish apk**

**finish();**

**}**

**else if(!myBluetooth.isEnabled()) {**

**Intent turnBTon = new Intent(BluetoothAdapter.ACTION\_REQUEST\_ENABLE);**

**startActivityForResult(turnBTon,1);**

**}**

//The onclick listener listens to an event call. When the user presses a button it calls the paireddeviceslist() method .//

**btnPaired.setOnClickListener(new View.OnClickListener() {**

**@Override**

**public void onClick(View v)**

**{**

**pairedDevicesList();**

**}**

**});**

**}**

24

Lets have a look at the pairedDevicesList method

**Private void pairedDevicesList() {**

**pairedDevices = myBluetooth.getBondedDevices();**

**ArrayList list = new ArrayList();**

**if (pairedDevices.size()>0) {**

**for (BluetoothDevice bt : pairedDevices) {**

**list.add(bt.getName() + "\n" + bt.getAddress());**

**}**

**}**

**else {**

**Toast.makeText(getApplicationContext(), "No Paired Bluetooth Devices Found.", Toast.LENGTH\_LONG).show();**

**}**

As you can see pairedDevicesList method populates the arraylist by storing the

Bluetooth enabled devices . if no Bluetooth device is found it simply returns a text

showing no Bluetooth device is found

25

The interface for Bluetooth Sockets is similar to that of TCP sockets: Socket and ServerSocket. On the server side, use a BluetoothServerSocket to create a listening server socket. When a connection is accepted by the BluetoothServerSocket, it will return a new BluetoothSocket to manage the connection. On the client side, use a single BluetoothSocket to both initiate an outgoing connection and to manage the connection.

// Following are the code snippets for the transmission activity

public class Transmission extends ActionBarActivity {

// we are the initialising view objects in this part .

**Button btnOn, btnOff, btnDis, btnRight, btnStop;**

**SeekBar brightness;**

**TextView lumn, value;**

**String address = null;**

**ProgressDialog progress;**

**BluetoothAdapter myBluetooth = null;**

**BluetoothSocket btSocket = null;**

**boolean isBtConnected = false;**

static final UUID myUUID = UUID.fromString("00001101-0000-1000-8000-00805F9B34FB");

// the view objects are getting linked to view objects present in xml layout files

**btnOn = (Button)findViewById(R.id.button2);**

**btnOff = (Button)findViewById(R.id.button3);**

**btnDis = (Button)findViewById(R.id.button4);**

**btnRight = (Button)findViewById(R.id.button5);**

**btnStop = (Button)findViewById(R.id.button6);**

**brightness = (SeekBar)findViewById(R.id.seekBar);**

**lumn = (TextView)findViewById(R.id.lumn);**

**value = (TextView) findViewById(R.id.textView4);**

26

new ConnectBT().execute();

// the onclick listener listens to an event . when the right button is pressed it calls the right function which rotates the servomotor in the right direction

**btnRight.setOnClickListener(new View.OnClickListener() {**

**@Override**

**public void onClick(View view) {**

**right();**

**}**

**});**

// the onclick listener listens to an event . when the left button is pressed it calls the right function which rotates the servomotor in the left direction

**btnLeft.setOnClickListener(new View.OnClickListener() {**

**@Override**

**public void onClick(View view) {**

**left();**

**}**

**});**

// the onclick listener listens to an event . when the strop button is pressed it calls the right stop function which stop rotating the servo in the right direction .

**btnStop.setOnClickListener(new View.OnClickListener() {**

**@Override**

**public void onClick(View view) {**

**rightStop();**

**}**

**});**

// the onclick listener listens to an event . when the strop button is pressed it calls the disconnect function which disconnects from the device.

26

**btnDis.setOnClickListener(new View.OnClickListener() {**

**@Override**

**public void onClick(View v) {**

**Disconnect();**

**}**

**});**

**//** this is the right function which uses IO stream to retrieve the output stream object and send characters using Bluetooth sockets.

**private void right() {**

**if (btSocket != null) {**

**try {**

**btSocket.getOutputStream().write("TR".toString().getBytes());**

**}**

**catch (Exception e) {**

**msg("Error");**

**}**

**}**

**}**

//This is the right stop function

**private void rightStop() {**

**if (btSocket != null) {**

**try {**

**btSocket.getOutputStream().write("TS".toString().getBytes());**

**BufferedInputStream istream = new BufferedInputStream(btSocket.getInputStream());**

27

**BufferedReader reader = new BufferedReader(new InputStreamReader(istream));**

**value.setText(reader.readLine());**

**}**

**catch (Exception e) {**

**msg("Error");**

**}**

**}**

**}**

This is the left function which uses IO stream to retrieve the output stream object and send character using Bluetooth sockets.

**private void left() {**

**if (btSocket!=null) {**

**try {**

**btSocket.getOutputStream().write("TO".toString().getBytes());**

**}**

**catch (IOException e) {**

**msg("Error");**

**}**

**}**

**}**

private void msg(String s) {

Toast.makeText(getApplicationContext(),s,Toast.LENGTH\_LONG).show();

}

Once the socket is connected, whether initiated as a client or accepted as a server, open the IO streams by calling getInputStream() and getOutputStream() in order to retrieve InputStream and OutputStream objects, respectively, which are automatically connected to the socket . In the above code the server socket uses getOutputStream() toretrieve the Output Stream object and send data to it .

theArduino device which is the client receives this data , it will rotate the servomotor to the right direction.

28

4) Inthis section we implemented the gate control part making use of the Arduino. When the Arduino receives information from the android device it rotates the servomotor in a specified direction which would ideally result in the opening and closing of the gate .

Following is the code for the gate control via servomotor using Arduino.

#include<SoftwareSerial.h>

#include<Servo.h>

**int enablePin = 5;**

Servo servo;

char command;

String string;

#define led 12

// this is the setup function

**void setup()**

**{**

**digitalWrite(enablePin, HIGH);**

**Serial.begin(9600);**

**pinMode(led, OUTPUT);**

**servo.attach(led);**

**servo.write(90);**

**}**

This is the loop function where we write our main code

**void loop()**

**{**

**digitalWrite(13, LOW);**

**if (Serial.available() > 0)**

29

**{string = "";}**

**while(Serial.available() > 0)**

**{**

**command = ((byte)Serial.read());**

**if(command == ':')**

**{**

**break;**

**}**

**else**

**{**

**string += command;**

**}**

**delay(1);**

**}**

**if(string == "TO")**

**{**

**servo.write(0);**

**}**

**if(string == "TR")**

**{**

**servo.write(120);**

**}**

**}**

As you can see we run an infinite loop. When the Arduino receives the charater ‘TR’,it rotates the servo by 120 degress. On the other hand when it receives the character ‘TO’ it sets itself back to the default position .

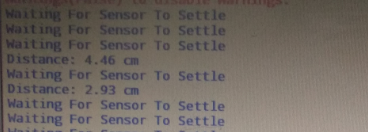
30

**8.Testing**

In this section we are going to have a look at all the test cases which have implemented as a part of the final year project

#**Test case 1**

1. **Title**:Obstacle Detection.
2. **Device to be tested**:Ultrasonic Sensor.
3. **Details:** this test checks checks for the presence of an obstacle infront the gate.this device put to test here is the ultrasonic sensor. If a person comes within the specified range of the sensor it should notify the user about the presence of a person along with its distance from the gate .
4. **Result:**Pass
5. **Screenshot** :



31

#**Test Case 2**

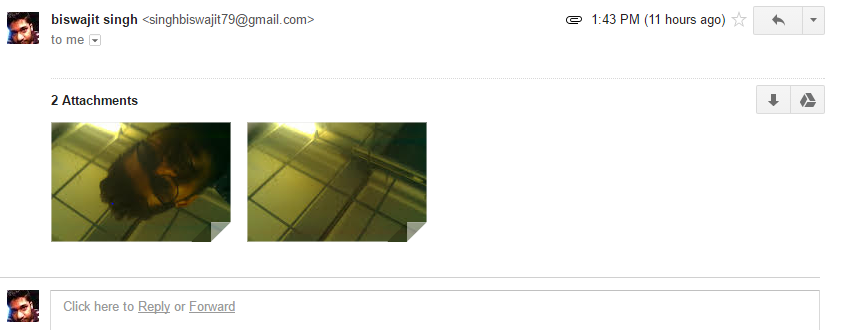
1. **Title:** Photo capture
2. **Device to be tested:**pi Camera
3. **Details**: this **case checks** if the pi camera is triggered by the sensor when an object comes within the range of the sensor and can successfully capture the photos of the person and store in it .jpg format on the desktop
4. **Result:**Pass
5. **Screenshot**:



32

**#Test case 3**

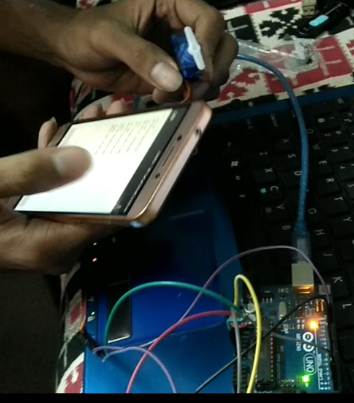
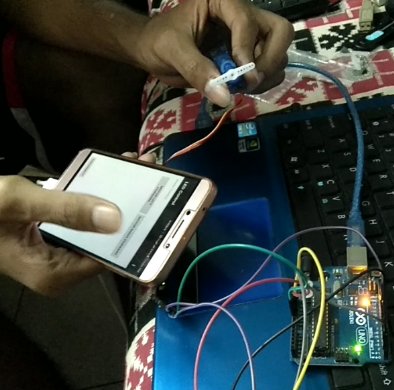
1. **Title:**Push Notification
2. **Details:** this test case checks if user successfully receives the photos of the person standing infront the gate as a push notification.
3. **Result**:Pass
4. **Screenshot:**



33

**#TestCase4**

1. **Title:**Gate Control
2. **Device to be Tested :**Arduino
3. **Details:** this test case checks whether the gate can be controlled by the user through the phone. It basically checks if servomotor is rotating as desired .
4. **Result**: Partial Pass
5. **Screenshot**:

34

**9. Conclusion:**

The prime objective of our project is to use the Smartphone to control the users door effectively. The user receives photos of the person standing infront his/her gate via a push notification .after receiving these photos he can easily control the opening /closing of the gate via an android app .

In this way, automation process is carried out. This is a simple prototype. Using this as a reference further it can be expanded to many other programs.

One limitation of using ultrasonic sensors is that fact that it is used to detect

the presence of any kind of object and not only a human object . So cases might arrive

where sensors trigger the camera to take photos which ultimately turn out to be

useless hence wasting the users time.

To overcome the limitation of our project we can integrate image processing into our

product . we can use image processing libraries like opencv and run a face detection

algorithm which will tell the product to send notifications to the user only when a face

is detected in the picture hence making sure the user only receives photos of human

objects.

This project is based on the Raspberry pi,Arduino, Android platform Java and Python.

These platforms are Free Open Source Software. So the overall implementation cost is low and can be easily configured

**10.References**

* 1. <https://docs.python.org/>
  2. <http://developer.android.com/training/index.html>
  3. <http://www.raspberrypi.org/>
  4. <http://stackoverflow.com/>
  5. <http://electronics.howstuffworks.com/>