VOICE AND GESTURE CONTROLLED HOME AUTOMATION

Submitted by

Rajat Bhatia(16BCE0513)
Rishabh Jaiswal(16BCE0893)
Sidhant Joshi(16BCE2044)
Shambhavi Rai(16BCE2084)
Akshat Sood(16BCE2168)

Under the guidance of

Prof. Naresh K

Prepared For

CSE 2006 – MICROPROCESSOR AND INERFACING

School of Computer Science and Engineering

CERTIFICATE

This is to certify that the project work entitled ‘VOICE AND GESTURE CONTROLLED HOME AUTOMATION’ that is being submitted by for Microprocessor and Interfacing is a record of bonafide work done under my supervision. The contents of this Project work, in full or in parts, have neither been taken from any other source nor have been submitted for any other CAL course.

ACKNOWLEDGEMENT

We would like to express our gratitude to our faculty Prof. Naresh K who gave us the valuable guidance for the project.It also helped us in doing a lot of work and we came to know about so many things. We are really thankful to him.

Our thanks extends to each one of us individually too for maintaining cooperation and a proper working environment around in developing the project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

1.INTRODUCTION

1.1 Objective

Nowadays, people have smartphones with them all the time. So it makes sense to use these to control home appliances. Presented here is a home automation system using a simple Android app, which you can use to control electrical appliances with voice commands. Commands are sent via Bluetooth to Arduino Uno. Also we will be using accelerometer to control electrical appliances with hand gestures. So you need not get up to switch on or switch off the device.Also with the help of ultrasonic sensor we will be automating a two level garage system based on the movement of the car.

1.2 Motivation

Home automation industry is growing rapidly. This is fuelled by the need to provide supporting systems for the elderly and the disabled, especially those who live alone. Coupled with this, the world population is confirmed to be getting older. Home automation systems must comply with the household standards and convenience of usage.

Automation is, unsurprisingly, one of the two main characteristics of home automation. Automation refers to the ability to program and schedule events for the devices on the network. The programming may include time-related commands, such as having your lights turn on or off at specific times each day. It can also include non-scheduled events, such as turning on all the lights in your home when your security system alarm is triggered.Once you start to understand the possibilities of home automation scheduling, you can come up with any number of useful and creative solutions to make your life better. The other main characteristic of cutting-edge home automation is remote monitoring and access. While a limited amount of one-way remote monitoring has been possible for some time, it's only since the rise in smartphones and tablets that we've had the ability to truly connect to our home networks while we're away. With the right home automation system, you can use any Internet-connected device to view and control the system itself and any attached devices.

Home automation gives you access to control devices in your home from a mobile device anywhere in the world. The term may be used for isolated programmable devices,but home automation more accurately describes homes in which nearly everything -- lights, appliances, electrical outlets, heating and cooling systems -- are hooked up to a remotely controllable network.The truth is that home automation systems have the power to simplify everything within your home and life. For example, homeowners often struggle with technologies that don’t work together. They hate the numerous remote controls cluttering coffee tables, and the wires tangled throughout rooms, but they don't know they have any other options. Today’s home automation technologies eliminate the confusion.

1.3 Background

This project presents the development of home appliances based on voice command and using gesture commands. We should be able to develop using our knowledge design,implement and configure a device that meets user requirements.

In regard with the project,a gesture sensor is used to turn on and off the devices based on the movement of the hands. Google application has been used as voice recognition and it processes the voice input from the smart phone. The voice input has been captured by the android and has been sent to the Arduino Uno. Bluetooth module in Arduino Uno receives the signal and processes the input signal to control the light and fan. This is very beneficial as a person doesn't has to get up and walk all the way to the switch board to turn off or on the device.The proposed system intends to control electrical appliances with relatively user-friendly interface and ease of installation.A large number of research papers have been published on home automation using various technologies.

Also the one level garage system has been automated based on the movement of the car.This overrides the need to manually control the garage door for parking of a car.

2. PROJECT DESCRIPTION AND GOALS

We will be automating a room and a garage. In the room we will be able to switch on or switch off the fans and lights, all at once or one at a time,with the help of either voice commands or with the help of hand gestures. We can also regulate the fan speed and the brightness of the lights through voice commands. We will be giving voice commands to an android mobile application and that command will be delivered to the components with the help of a bluetooth module. For gesture controll we will be using accelerometer that will basically measure the angle of rotation and accordingly switch on or switch off the components(fans,lights) as per the code fed to the arduino. For example, if we rotate the accelerometer to the left, then the lights go off. If we rotate the accelerometer towards the right then lights go on. Using voice commands, if we say "Lights on" to the phone speaker operating on the android operating system the lights will be on and if we say "Lights off" then the lights will be off. We can also give commands to a specific light by saying "Light One off" or "Light One On". Same applies for the fans.

We will also automate a one-level garage. In the one-level garage we will be using ultra-sonic sensors to detect if a car is coming.If the car comes then the garage door opens with the help of the servo motor and as soon as it enters the garage the garage door closes.

3.Technical Specifications

While the overall goals, strategies and objectives have been stated, the specifications of the components will be determined as they are identified for their applicability in the project. The
overall system will meet the specifications stated below.

ADXL335

Features and Benefits

- 3-axis sensing
- Small, low-profile package
- 4 mm × 4 mm × 1.45 mm LFCSP
- Low power - 350 μA (typical)
- Single-supply operation
  1.8 V to 3.6 V
- 10,000 g shock survival
- Excellent temperature stability
- BW adjustment with a single capacitor per axis
- RoHS/WEEE lead-free compliant

Applications

- Cost sensitive, low power, motion- and tilt-sensing applications
- Mobile devices
- Gaming systems
- Disk drive protection
- Image stabilization
- Sports and health devices

Bluetooth Sensor HC-05

Hardware features

Typical -80dBm sensitivity.

- Up to +4dBm RF transmit power.
- Low Power 1.8V Operation, 3.3 to 5 V I/O.
- PIO control.
- UART interface with programmable baud rate.

- With integrated antenna.
- With edge connector.

HC-05 module is an easy to use Bluetooth SPP (Serial Port Protocol) module, designed for transparent wireless serial connection setup. Serial port Bluetooth module is fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) 3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses CSR Bluecore 04-External single chip Bluetooth system with CMOS technology and with AFH (Adaptive Frequency Hopping Feature). It has the footprint as small as 12.7mmx27mm.

Arduino Uno

Power

The Arduino Uno can be powered via the USB connection or with an external power supply. The power source is selected automatically.

External (non-USB) power can come either from an AC-to-DC adapter (wall-wart) or battery. The adapter can be connected by plugging a 2.1mm center-positive plug into the board's power jack. Leads from a battery can be inserted in the Gnd and Vin pin headers of the POWER connector.
The board can operate on an external supply of 6 to 20 volts. If supplied with less than 7V, however, the 5V pin may supply less than five volts and the board may be unstable. If using more than 12V, the voltage regulator may overheat and damage the board. The recommended range is 7 to 12 volts.

The power pins are as follows:

- VIN. The input voltage to the Arduino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- 5V.This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 12V), the USB connector (5V), or the VIN pin of the board (7-12V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage your board. We don't advise it.
- 3V3. A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- GND. Ground pins.
- IOREF. This pin on the Arduino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs for working with the 5V or 3.3V.

Memory

The ATmega328 has 32 KB (with 0.5 KB used for the bootloader). It also has 2 KB of SRAM and 1 KB of EEPROM (which can be read and written with the EEPROM library).

Input and Output

Each of the 14 digital pins on the Uno can be used as an input or output, using pinMode(), digitalWrite(), and digitalRead() functions. They operate at 5 volts. Each pin can provide or receive a maximum of 40 mA and has an internal pull-up resistor (disconnected by default) of 20-50 kOhms. In addition, some pins have specialized functions:

- Serial: 0 (RX) and 1 (TX). Used to receive (RX) and transmit (TX) TTL serial data. These pins are connected to the corresponding pins of the ATmega8U2 USB-to-TTL Serial chip.
- External Interrupts: 2 and 3. These pins can be configured to trigger an interrupt on a low value, a rising or falling edge, or a change in value. See the attachInterrupt() function for details.
- PWM: 3, 5, 6, 9, 10, and 11. Provide 8-bit PWM output with the analogWrite() function.
- SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). These pins support SPI communication using the SPI library.
- LED: 13. There is a built-in LED connected to digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off. The Uno has 6 analog inputs, labeled A0 through A5, each of which provide 10 bits of resolution (i.e. 1024 different values). By default they measure from ground to 5 volts, though is it possible to change the upper end of their range using the AREF pin and the analogReference() function. Additionally, some pins have specialized functionality:
- TWI: A4 or SDA pin and A5 or SCL pin. Support TWI communication using the Wire library. There are a couple of other pins on the board:
- AREF. Reference voltage for the analog inputs. Used with analogReference().
- Reset. Bring this line LOW to reset the microcontroller. Typically used to add a reset button to shields which block the one on the board.

Communication

The Arduino Uno has a number of facilities for communicating with a computer, another Arduino, or other microcontrollers. The ATmega328 provides UART TTL (5V) serial communication, which is available on digital pins 0 (RX) and 1 (TX). An ATmega16U2 on the board channels this serial communication over USB and appears as a virtual com port to software on the computer. The '16U2 firmware uses the standard USB COM drivers, and no external driver is needed. However, on Windows, a .inf file is required. The Arduino software includes a serial monitor which allows simple textual data to be sent to and from the Arduino board. The RX and TX LEDs on the board will flash when data is being transmitted via the USB-to-serial chip and USB connection to the computer (but not for serial communication on pins 0 and 1).

A SoftwareSerial library allows for serial communication on any of the Uno's digital pins.

The ATmega328 also supports I2C (TWI) and SPI communication. The Arduino software includes a Wire library to simplify use of the I2C bus; For SPI communication, use the SPI library.

Programming

The Arduino Uno can be programmed with the Arduino software (download). Select "Arduino Uno from the Tools > Board menu (according to the microcontroller on your board

The ATmega328 on the Arduino Uno comes preburned with a bootloader that allows you to upload new code to it without the use of an external hardware programmer. It communicates using the

original STK500 protocol.

You can also bypass the bootloader and program the microcontroller through the ICSP (In-Circuit Serial Programming) header using Arduino ISP or similar.

The ATmega16U2 (or 8U2 in the rev1 and rev2 boards) firmware source code is available. The ATmega16U2/8U2 is loaded with a DFU bootloader, which can be activated by:

- On Rev1 boards: connecting the solder jumper on the back of the board (near the map of Italy) and then resetting the 8U2.
- On Rev2 or later boards: there is a resistor that pulling the 8U2/16U2 HWB line to ground, making it easier to put into DFU mode.

You can then use Atmel's FLIP software (Windows) or the DFU programmer (Mac OS X and Linux) to load a new firmware. Or you can use the ISP header with an external programmer (overwriting the DFU bootloader).

Automatic (Software) Reset

Rather than requiring a physical press of the reset button before an upload, the Arduino Uno is designed in a way that allows it to be reset by software running on a connected computer. One of the hardware flow control lines (DTR) of the ATmega8U2/16U2 is connected to the reset line of the ATmega328 via a 100 nanofarad capacitor. When this line is asserted (taken low), the reset line drops long enough to reset the chip. The Arduino software uses this capability to allow you to upload code by simply pressing the upload button in the Arduino environment. This means that the bootloader can have a shorter timeout, as the lowering of DTR can be well-coordinated with the start of the upload.

This setup has other implications. When the Uno is connected to either a computer running Mac OS X or Linux, it resets each time a connection is made to it from software (via USB). For the following half-second or so, the bootloader is running on the Uno. While it is programmed to ignore malformed data (i.e. anything besides an upload of new code), it will intercept the first few bytes of data sent to the board after a connection is opened. If a sketch running on the board receives one-time configuration or other data when it first starts, make sure that the software with which it communicates waits a second after opening the connection and before sending this data.

The Uno contains a trace that can be cut to disable the auto-reset. The pads on either side of the trace can be soldered together to re-enable it. It's labeled "RESET-EN". You may also be able to disable the auto-reset by connecting a 110 ohm resistor from 5V to the reset line.

USB Overcurrent Protection

The Arduino Uno has a resettable polyfuse that protects your computer's USB ports from shorts and overcurrent. Although most computers provide their own internal protection, the fuse provides an extra layer of protection. If more than 500 mA is applied to the USB port, the fuse will automatically break the connection until the short or overload is removed.

Physical Characteristics

The maximum length and width of the Uno PCB are 2.7 and 2.1 inches respectively, with the USB connector and power jack extending beyond the former dimension. Four screw holes allow the board

to be attached to a surface or case. Note that the distance between digital pins 7 and 8 is 160 mil (0.16"), not an even multiple of the 100 mil spacing of the other pins.

Ultrasonic sensor :

Features:

- Operating Voltage: 5V DC
- Operating Current: 15mA
- Measure Angle: 15°
- Ranging Distance: 2cm - 4m

HC-SR04 is an ultrasonic ranging module that provides 2 cm to 400 cm non-contact measurement function. The ranging accuracy can reach to 3mm and effectual angle is < 15°. It can be powered from a 5V power supply.

HC-SR04 Specifications

- Working Voltage: DC 5V
- Working Current: 15mA
- Working Frequency: 40Hz
- Max Range: 4m
- Min Range: 2cm
- Measuring Angle: 15 degree
- Trigger Input Signal: 10µS TTL pulse
- Echo Output Signal Input TTL lever signal and the range in proportion
- Dimension 45 * 20 * 15mm

4.Design Approach and details

4.1.1 Design Approach

Fig 4.1.1: Block Diagram

CIRCUIT DIAGRAM:

For Garage:

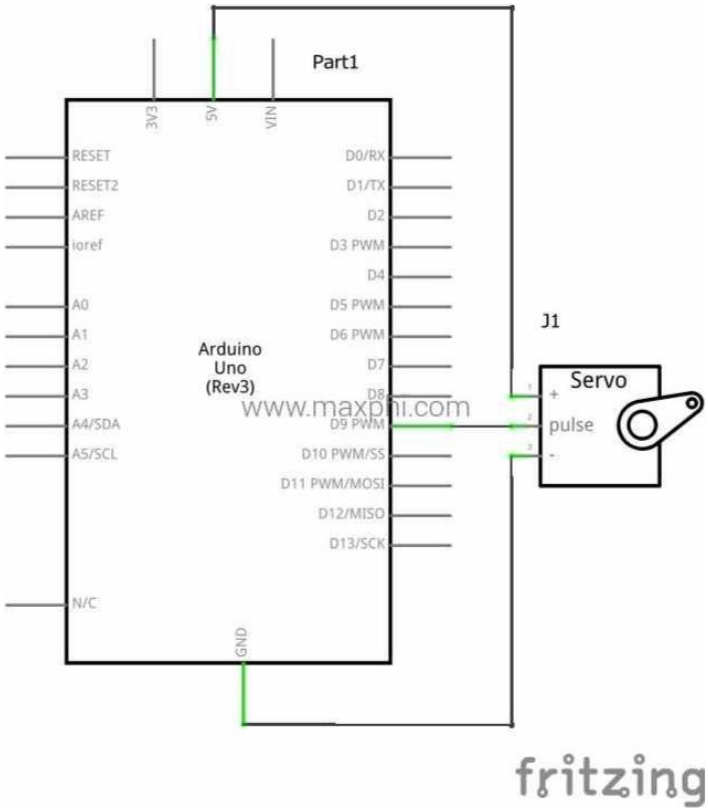


Fig 4.1.2: Garage Circuit

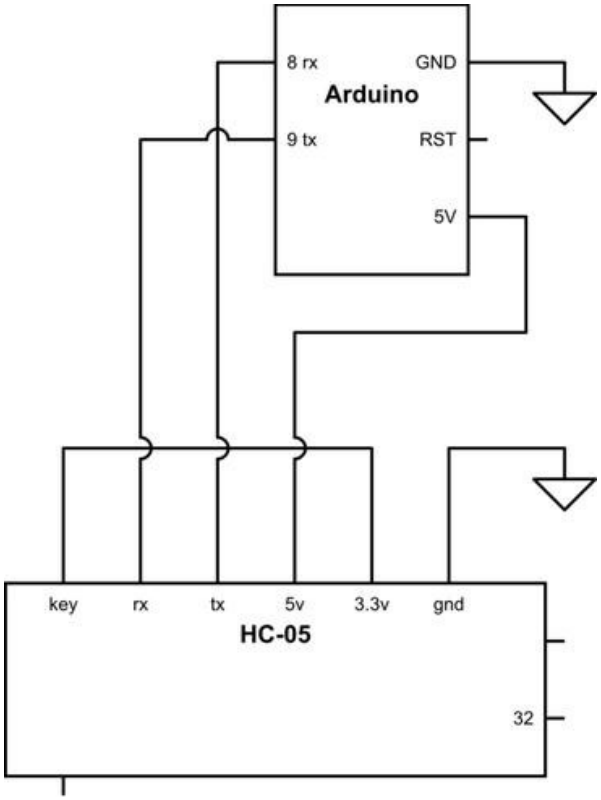


Fig 4.1.3: Bluetooth Circuit

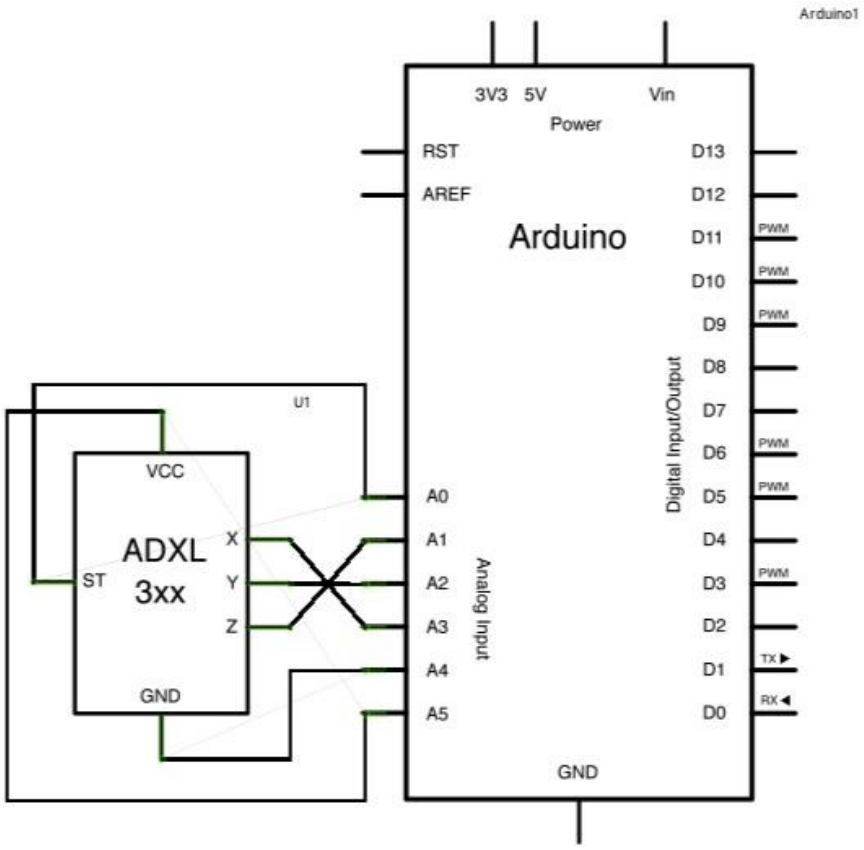
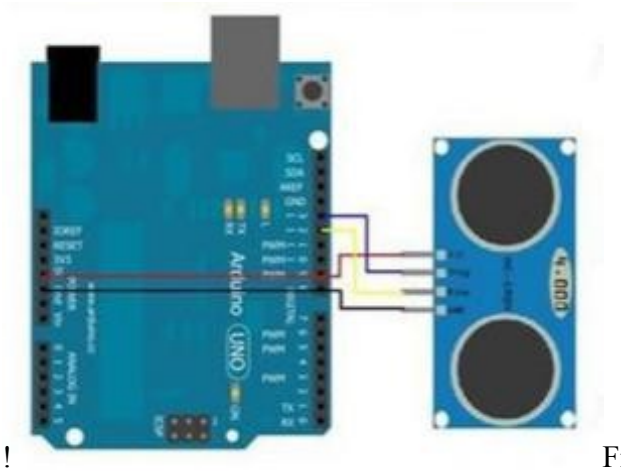


Fig 4.1.4:
Accelerometer Circuit

8!



! Fig 4.1.5:for ultra-sonic sensor interfacing.

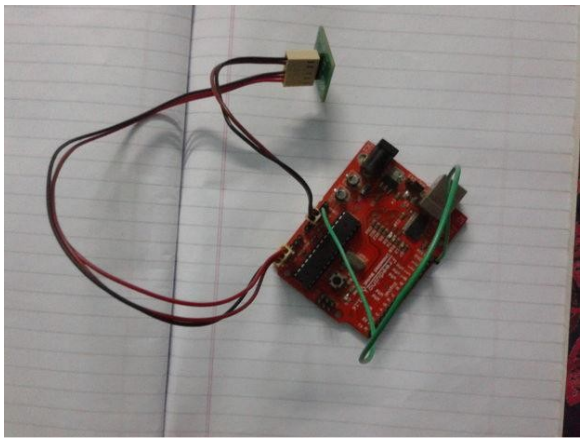
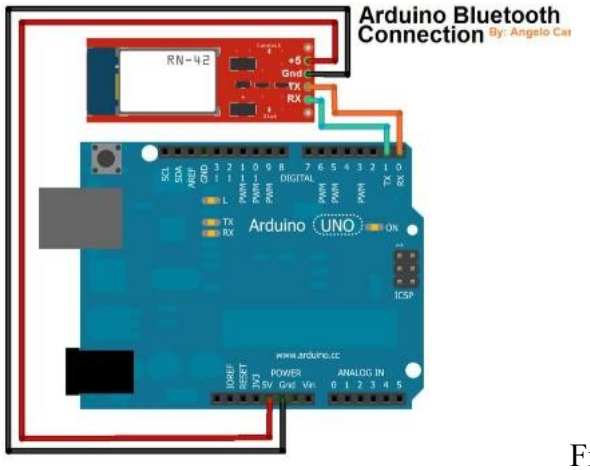


Fig 4.1.6:for accelerometer



! Fig 4.1.7:for bluetooth interfacing

9!

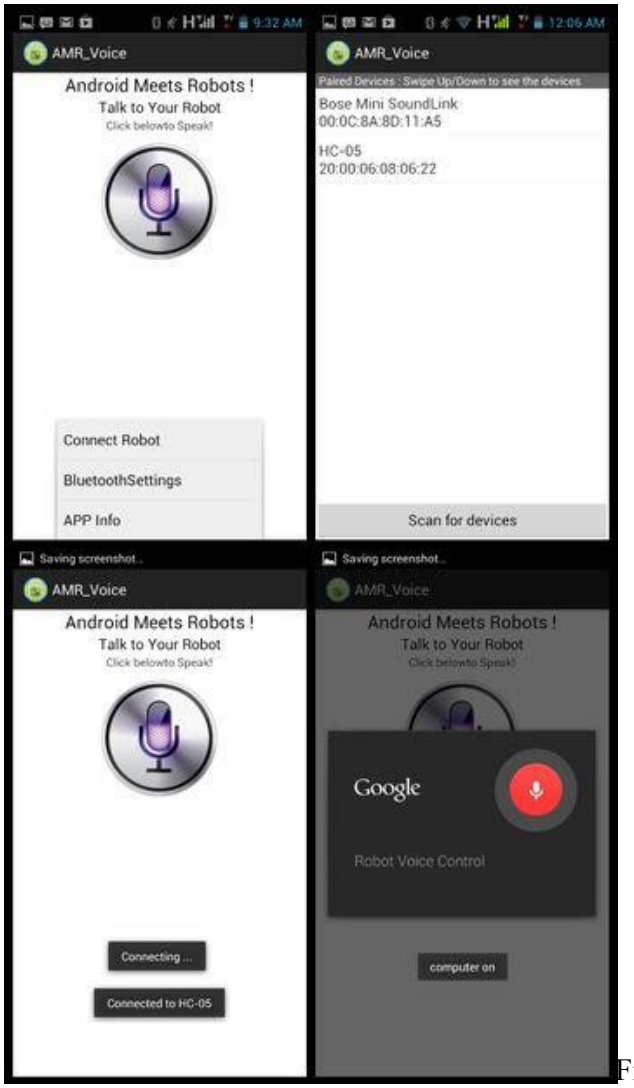


Fig 4.1.8: android app screenshot

4.1.2 Algorithm

Arduino Codes:

For gesture controll:

```
const int ap1 = A5;

const int ap2 = A4;

const int ap3 = A3;

int led = 9;

int led1 = 10;

int sv1 = 0;

int ov1 = 0;

int sv2 = 0;
```

```
int ov2= 0;
int sv3 = 0;
int ov3= 0;
void setup() {
  /  initialize serial communications at 9600 bps:
  Serial.begin(9600);
}
void loop() {
  analogReference(EXTERNAL);    //connect 3.3v to AREF
  /  read the analog in value:
  sv1 = analogRead(ap1);
  /  map it to the range of the analog out:
  ov1 = map(sv1, 0, 1023, 0, 255);
  /  change the analog out value:
  delay(2);
/
  sv2 = analogRead(ap2);
  ov2 = map(sv2, 0, 1023, 0, 255);
 //
  delay(2);
/
  sv3 = analogRead(ap3);
  ov3 = map(sv3, 0, 1023, 0, 255);
  /  print the results to the serial monitor:
  Serial.print("Xsensor1 = " );
  Serial.print(sv1);
  Serial.print("\t output1 = ");
  Serial.println(ov1);
```

```
  Serial.print("Ysensor2 = " );
  Serial.print(sv2);
  Serial.print("\t output2 = ");
  Serial.println(ov2);
  Serial.print("Zsensor3 = " );
  Serial.print(sv3);
  Serial.print("\t output3 = ");
  Serial.println(ov3);
if(sv2 < 500)
{
  digitalWrite(led, HIGH);
}
if(sv2 > 500)
{
  digitalWrite(led, LOW);
}
if(sv1 > 500)
{
  digitalWrite(led1, HIGH);
}
if(sv1 < 500)
{
  digitalWrite(led1, LOW);
}
}
```

For Voice Controll:

```
String voice;
```

```
int
led1 = 2, //Connect LED 1 To Pin #2
led2 = 3, //Connect LED 2 To Pin #3
led3 = 4, //Connect LED 3 To Pin #4
led4 = 5, //Connect LED 4 To Pin #5
led5 = 6;
int led6 = 10;
//Connect LED 5 To Pin #6
//-------------------------Call A Function-----------------------------//
void allon(){
    digitalWrite(led1, HIGH);
    digitalWrite(led2, HIGH);
    digitalWrite(led3, HIGH);
    digitalWrite(led4, HIGH);
   digitalWrite(led5, HIGH);
   analogWrite(led6, 255);
}
void alloff(){
    digitalWrite(led1, LOW);
    digitalWrite(led2, LOW);
    digitalWrite(led3, LOW);
    digitalWrite(led4, LOW);
    digitalWrite(led5, LOW);
    analogWrite(led6, 0);
}
//-------------------------------------------------------------------//
void setup() {
  Serial.begin(9600);
```

```
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);
  pinMode(led4, OUTPUT);
  pinMode(led5, OUTPUT);
}
//--------------------------------------------------------------------//
void loop() {
  while (Serial.available()){ //Check if there is an available byte to read
  delay(10); //Delay added to make thing stable
  char c = Serial.read(); //Conduct a serial read
  if (c == '#') {break;} //Exit the loop when the # is detected aft er the word
  voice += c; //Shorthand for voice = voice + c }

  if (voice.length() > 0) {
    Serial.println(voice);
//--------------------------------------------------------------------//
  //----------Control Multiple Pins/ LEDs----------//
      if(voice == "*all on" ) {allon();} //Turn Off All Pins (Call Function)
  else if(voice == "*all of" ){alloff();} //Turn On All Pins (Call Function)

  //----------Turn On One-By-One----------//
  else if(voice == "*TV on") {digitalWrite(led1, HIGH);}
  else if(voice == "*bathroom lights on") {digitalWrite(led2, HIGH);}
  else if(voice == "*computer on") {digitalWrite(led3, HIGH);}
  else if(voice == "*bedroom lights on") {digitalWrite(led4, HIGH);}
  else if(voice == "*fan on") {digitalWrite(led5, HIGH);}
  else if(voice == "*low light"){analogWrite(led6, 50);}
```

```
  else if(voice == "*medium lights"){analogWrite(led6, 150);}
  else if(voice == "*bright lights"){analogWrite(led6, 255);}

  //----------Turn Off One-By-One----------//
  else if(voice == "*TV off" ) {digitalWrite(led1, LOW);}
  else if(voice == "*bathroom lights off" ) {digitalWrite(led2, LOW);}
  else if(voice == "*computer of" ) {digitalWrite(led3, LOW);}
  else if(voice == "*bedroom lights off" ) {digitalWrite(led4, LOW);}
  else if(voice == "*fan of") {digitalWrite(led5, LOW);}
  }}
//-------------------------------------------------------------------//
//voice="";}} //Reset the variable after initiating
```

For garage:

```
#include <Servo.h>
#define trigPin 11
#define echoPin 10
#define trigPin1 13
#define echoPin1 12
Servo servo;
Servo servo1;
void setup()
{
Serial.begin (9600);
pinMode(trigPin, OUTPUT);
pinMode(echoPin, INPUT);
pinMode(trigPin1, OUTPUT);
pinMode(echoPin1, INPUT);
```

```
servo.attach(9);
servo1.attach(6);
}
void loop()
{
long duration, distance, duration1, distance1;
digitalWrite(trigPin, LOW);
digitalWrite(trigPin1, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
digitalWrite(trigPin1, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
digitalWrite(trigPin1, LOW);
duration = pulseIn(echoPin, HIGH);
duration1 = pulseIn(echoPin1, HIGH);
distance = (duration/2) / 29.1;
distance1 = (duration1/2) / 29.1;
if((distance<3)&&(distance1>3))
{
Serial.println("the distance is less than 5");
servo.write(180);
delay(2000);
servo.write(0);
}
else if((distance1<3)&&(distance<3))
{
Serial.println("the distance is less than 5");
```

```
servo1.write(180);
servo.write(180);
delay(2000);
 servo1.write(0);
 servo.write(0);
}
else
{
 servo1.write(0);
 servo.write(0);
}
}
```

4.2 Codes and Standards

IEEE 802.15.1: WPAN / Bluetooth - is a working group of the Institute of Electrical and Electronics Engineers (IEEE) IEEE 802 standards committee which specifies wireless personal area network (WPAN) standards.

1293-1998 - IEEE Standard Specification Format Guide and Test Procedure for Linear, Single-Axis, Non-Gyroscopic Accelerometers-

Arduino Uno R3-Arduino/Genuino Uno is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins , 6 analog inputs, a 16 MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button

4.3 Constraints,Alternatives and Trade-offs

Constraints

- BLUETOOTH RANGE 100M
- LENGTH OF ACCELEROMETER WIRE
- VOICE RECOGNITION WILL BE DONE IN ENGLISH
- ONLY WORKS ON ANDROID OPERATING SYSTEM

Trade –offs

- We will be doing the project on small scale thus reducing the expenditure.

5. Schedule, Tasks and Milestones

There are three major milestones as well as several smaller tasks that must be achieved in order to reach the milestones. The three major milestones are:

- Switching on and off the LED lights and fans using hand gesture
- Switching on and off the LED lights and fans by voice control
- Automatic car parking in garage

Tasks were split up among group members according to each member's level of expertise and comfort. However, the other group members provided assistance if needed. Also, each tasks` respective member is held accountable if the task fails.

6. Project demonstration

7. Marketing and Cost Analysis

7.1 Marketing Analysis

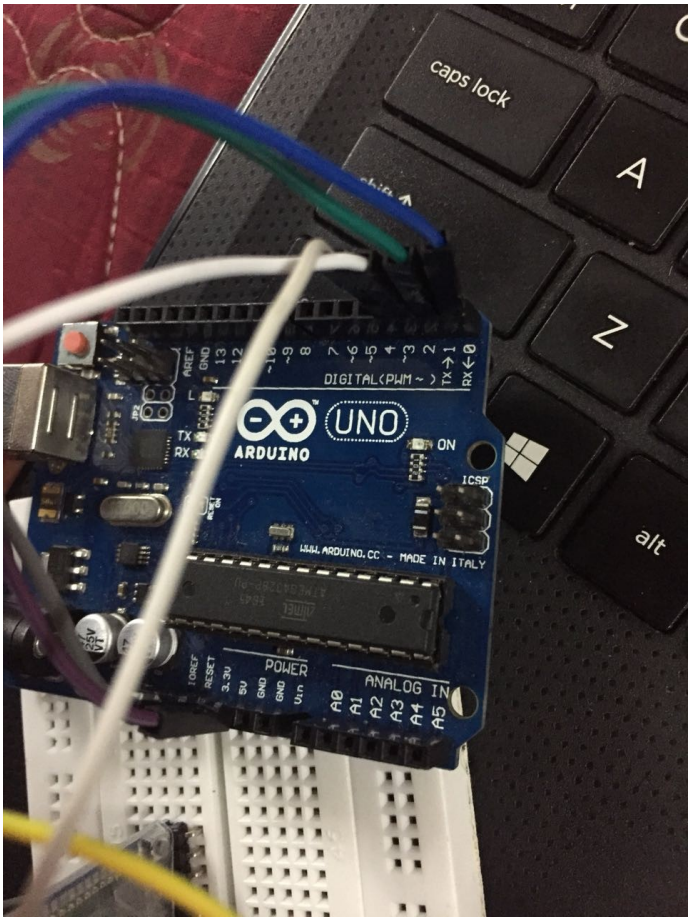
Home Automation is undeniably a resource which can make a home environment automated. People can control their electrical devices via these Home Automation devices. We think this product have high potential for marketing in the future. At the moment the components are a bit too high to be able to produce these devices for an interesting price. But the new stream of home automation systems has developed into a vast one and the current market is flooded with a flurry of home automation systems and device manufacturers. Factors such as the significantly growing IoT market, presence of a large number of manufacturers expanding their product portfolios, and the increasing importance of home monitoring systems are driving the growth of the home automation system market. The global home automation system market was valued at USD 39.93 Billion in 2016 and is expected to grow significantly at 11.3% in the next five areas.

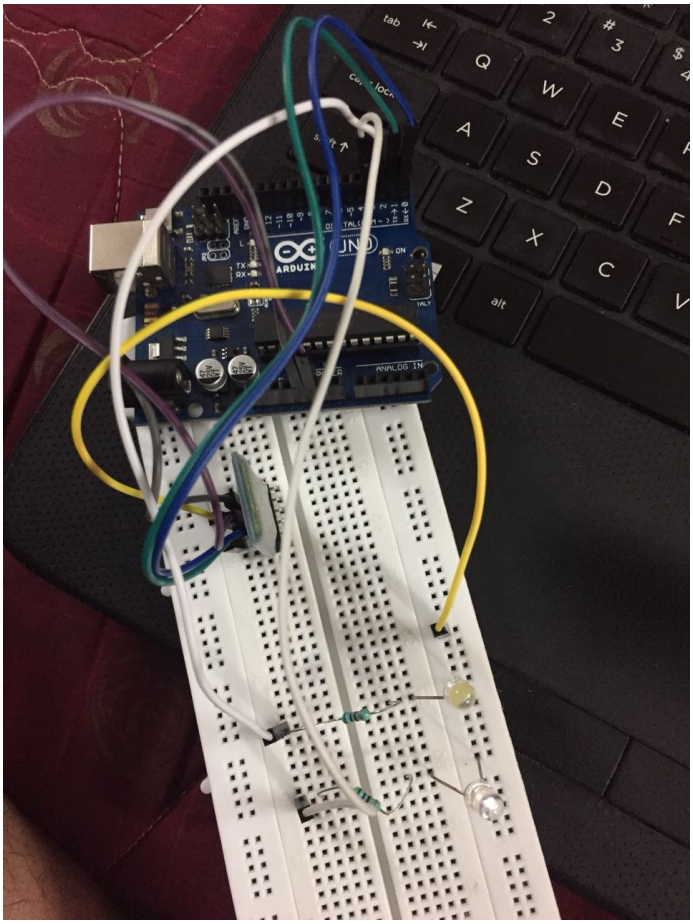
7.2 Cost Analysis

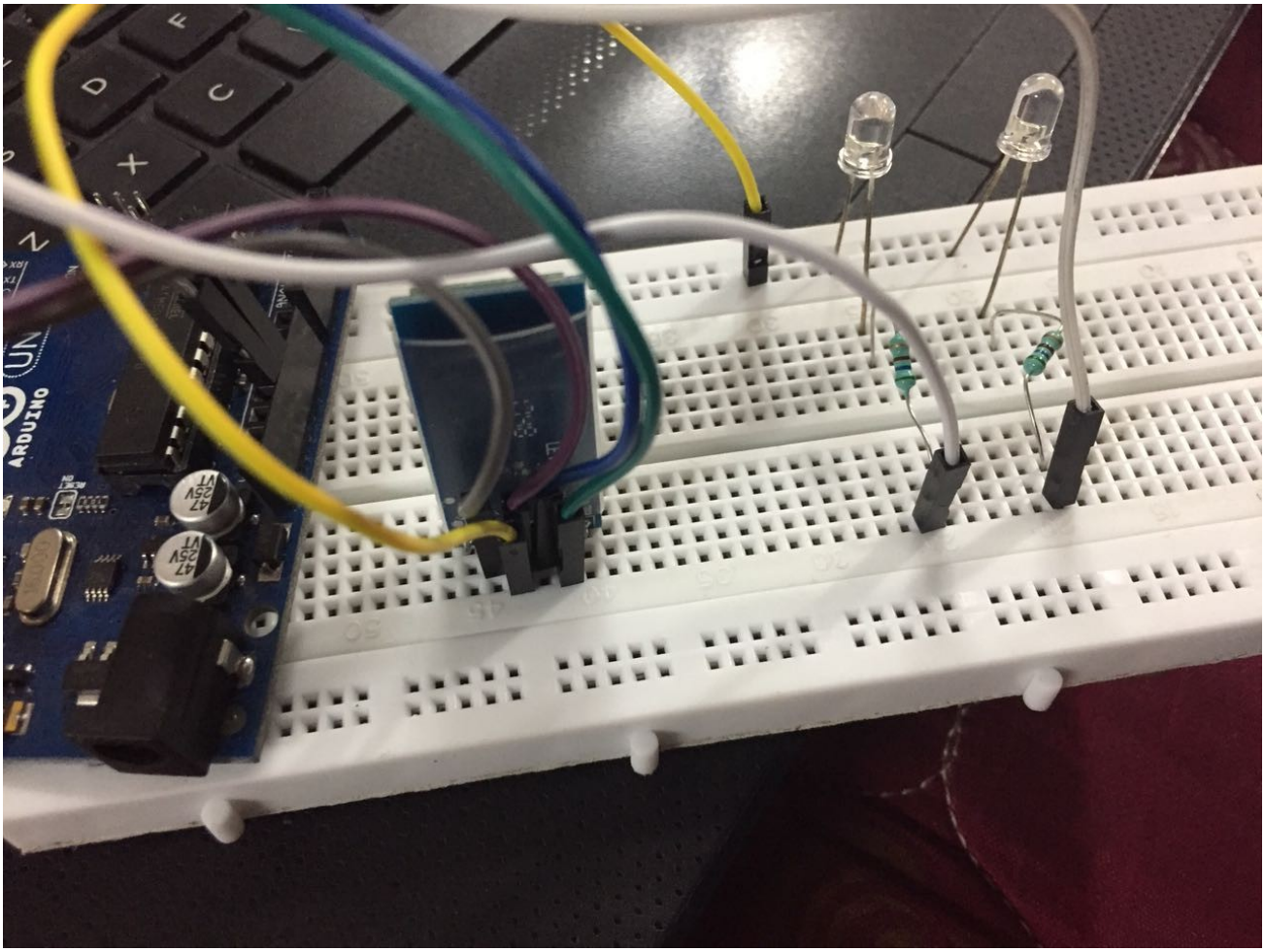
The following table provides a detailed cost analysis on the required parts for home automation and garage system.

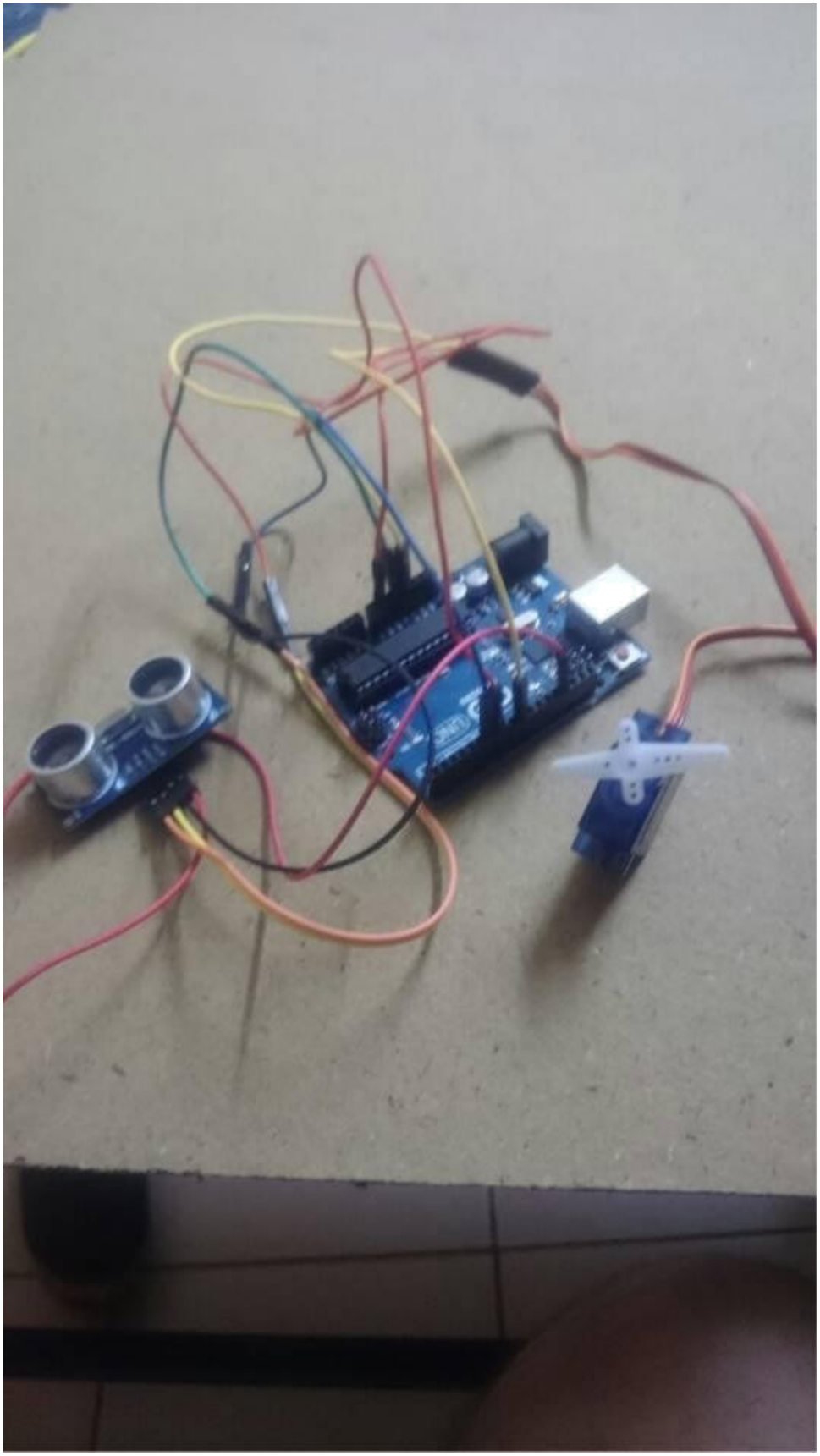
Sr. No.	Parts List	Quantity	Price	Total Amount
1.	Arduino-UNO	3	550	1650
2.	HC-05 Bluetooth module	1	275	275
3.	Jumper Wires	10feet	10	100

6. Project Demonstration









7. Marketing and Cost Analysis

7.1 Marketing Analysis

Home Automation is undeniably a resource which can make a home environment automated. People can control their electrical devices via these Home Automation devices. We think this product have high potential for marketing in the future. At the moment the components are a bit too high to be able to produce these devices for an interesting price. But the new stream of home automation systems has developed into a vast one and the current market is flooded with a flurry of home automation systems and device manufacturers. Factors such as the significantly growing IoT market, presence of a large number of manufacturers expanding their product portfolios, and the increasing importance of home monitoring systems are driving the growth of the home automation system market. The global home automation system market was valued at USD 39.93 Billion in 2016 and is expected to grow significantly at 11.3% in the next five areas.

7.2 Cost Analysis

The following table provides a detailed cost analysis on the required parts for home automation and garage system.

Sr. No.	**Parts List**	**Quantity**	**Price**	**Total Amount**
1.	Arduino-UNO	3	550	1650
2.	HC-05 Bluetooth module	1	275	275
3.	Jumper Wires	10feet	10	100
4.	Ultrasonic sensor	1	200	200
5	Servo Motor	1	330	330
6	Accelerometer	1	200	200
7	LED’s	20	1	20

4.	Ultrasonic sensor	1	200	200
5	Servo Motor	1	330	330
6	Accelerometer	1	200	200
7	LED’s	20	1	20

Table 7.2

8.Summary

The project goals are carefully read, understood and achieved through this prototype. A thorough analysis is done and seen through the constraints and trade-offs. The design analysis is done initially through the basic architecture and then the circuit is designed. Functionality of each component and how it can be used is understood and explained. The algorithm for the microcontroller is made, and coded in Arduino IDE. The system is made within the stipulated and anticipated time. Market analysis is done with the existing system and a cheaper and efficient system is made which can be later launched in the market.

In future,it is possible to use WiFi technology over Bluetooth so that automation can be done from a greater distance.Also two level garage system can be converted to a multi-level garage system with more than two levels.

9.References

1.Piyare, Rajeev, and M. Tazil. "Bluetooth based home automation system using cell phone." *Consumer Electronics (ISCE), 2011 IEEE 15th International Symposium on*.IEEE, 2011.

2.A. R. Al-Ali and M. Al-Rousan (2004).“Java-Based Home Automation System.”IEEE. 2. 498 - 504.

3.Alkar, Ali Ziya, and UmitBuhur. "An Internet based wireless home automation system for multifunctional devices." *IEEE Transactions on Consumer Electronics* 51.4 (2005): 1169-1174.

4.BarisYuksekkaya, A. AlperKayalar, M. BilgehanTosun, M. KaanOzcan, and Ali ZiyaAlkar, “A GSM, Internet and Speech Controlled Wireless Internet Home Automation System”, IEEE Transactions on Consumer Electronics, Vol. 52, No. 3, AUGUST 2006

5.Obaid, Thoraya, et al. "ZigBee based voice controlled wireless smart home system." *International Journal of Wireless & Mobile Networks* 6.1 (2014): 47.