Visvesvaraya Technological University, Belagavi

Shri Dharmasthala Manjunatheshwara College of Engineering and Technology.

Dharwad-580002

Project Report on

Gesture Controlled Enactor(GCE)

Under the Guidance of

Prof. Govind Negalur

Project Team

MANAS UDAY NAYAK	**2SD12CS049**
PALLAVI NAIK	**2SD12CS065**
RADHIKA SUBRAY NAYAK	**2SD12CS075**
KRUPA HEGDE	**2SD12CS130**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

2015-2016

S. D. M. College of Engineering & Technology

Dharwad – 580002

Department of Computer Science & Engineering

Certificate

Certified that the project work entitled ***Gesture Controlled Enactor*** is an original work carried out by ***Manas Uday Nayak(2SD12CS049), Pallavi Naik(2SD12CS065), Radhika Subray Nayak(2SD12CS075) and Krupa Hegde(2SD12CS130)*** in partial fulfillment for the award of degree of Bachelor of Engineering in ***Computer Science & Engineering*** of **S.D.M. College of Engineering & Technology under the Visvesvaraya Technological University, Belagavi** during the year 2015 – 16. The project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the award of Bachelor of Engineering Degree.

Signature of the Guide **Signature of the HOD** **Signature of the Principal**

Name of the student:

University Seat Number:

Name and signature of the examiners:

Sl. No.	Name	Signature with date
1		
2		

Dean (Academic Program) **PRINCIPAL**

Abstract

The proliferation on IoT technology is finding opportunities in many fields. This project is aimed to make use of IoT technology to perform operations and to control things using gestures. Each gesture conveys specific meaning and this is the base to deliver jobs. There are many techniques to capture and recognize gestures. The application makes use of accelerometer to capture gestures. Using this, Gesture Controlled Enactor (GCE) prototype is developed which is useful especially for disabled people. Widening of this technology can find many applications in the field of automation.

ACKNOWLEDGEMENT

Major-project is a platform to build something out of our knowledge, not only to test our capabilities, but also to get motivated to do something more and much better in the future. When one reason behind the success of the project is our sincere efforts, hard work and talents, meanwhile, the other main reason is the kind assistance, guidance, suggestions and blessings of many people. We possess a special gratitude towards them and it is our duty to express it through words.

We express our profound gratitude and indebtedness to those who have extended their invaluable help and co-operation directly or indirectly during the progress of the project and thereby ensuring successful completion.

We are thankful to S.D.M College of Engineering & Technology Dharwad and Department of Computer Science Engineering for providing us the opportunity to do this project.

We are highly indebted to Prof. Govind Negalur for his guidance and constant supervision, constant encouragement, motivation, support, directions and timely suggestions throughout the project.

We are immensely grateful to Prof. S.M.Joshi, Head of CSE Department, for his help and needful advice throughout our project work.

We would also like to express our sincere thanks to our honourable Principal Dr. S.B. Vanakudre, SDMCET, Dharwad, for providing the facilities and support.

We are also thankful to Prof. Mayur Patil and Prof. S S Navalgund for their help regarding the project. Also we would like to express our gratitude towards all other faculty members who with their kind co-operation were present to help us out at all odd hours of the day.

We thank all our non-teaching staff for extending their timely help and kind co-operation. Finally, we would like to thank our parents, all our friends and the other people who directly or indirectly contributed towards the successful completion of the project.

Table of Contents

Abstract	i
Acknowledgement	ii
Table of Contents	iii
List of Figures	v
List of Tables	vi

Chapter 1 Introduction………………………………………………………................	1
1.1 Making Sense of Gesture..	1
1.2 Making Sense of IoT..	1
1.3 Motivation and Goals...	3
Chapter 2 Literature Survey……………………………………………………………..	5
Chapter 3 System Requirement Specification..	7
3.1 Functional Requirements………………………………………………………..	7
3.2 Non Functional Requirements…………………………………………………..	7
3.3 System Requirements………………………………………………………......	8
3.3.1 Hardware Requirements………………………………………………	8
3.3.2 Software Requirements…………………………………………….….....	8
Chapter 4 System Modelling Designs……………………………………………………….	9
4.1 System Design Chart...	9
4.2 Use Case Diagram…………………………………………………………………	10
4.3 Sequence Diagrams………………………………………………………………..	11
Chapter 5 Hardware……………………………………………………………………………	17
5.1 Introduction………………………………………………………………………..	17
5.2 Components……………………………………………………………………….	17
5.2.1 Arduino UNO R3..	17
5.2.2 ADXL3xx Accelerometer...……………………………………………	20
5.2.3 ESP8266 Wifi Module……………………………………………………..	22
5.3 Project Cost………………………………………………………………………..	24

Chapter 6 Implementation……………………………………………………………….. 25
6.1 Introduction ………………………………………………………………… 25
6.2 Reading Accelerometer Data………………………………………………… 26
6.3 Data Processing in Arduino …………………………….…………… 26
6.4 Controlling Light and Fan …………………………………………………... 28
6.5 Sending Data to Cloud……………………………………………………….. 28
6.6 Receiving Push Notification on Android App……………………………….. 28
6.7 Snapshots…………………………………………………………………….. 29
Chapter 7 Testing ……………………………………………………………….... 32
7.1 Test Case designs……………………………………………………………. 33
Chapter 8 Conclusion…………………………………………………………………... 35
8.1 Conclusion and Outcomes………………………………………………… 35
8.2 Future Scope…………………………………………………………………. 36

List of Figures

1	Fig: 4.1-System Design chart......……………………………………………………...	9
2	Fig: 4.2-Use case diagram of GCE……………..……………………………….…...	10
3	Fig: 4.3.1-Sequence diagram for turn on light………………………..………………	11
4	Fig: 4.3.2-Sequence diagram for turn off light………………………...……………..	12
5	Fig: 4.3.3-Sequence diagram for turn on fan………………………...……................	13
6	Fig: 4.3.4-Sequence diagram for turn off fan……………………….....……………..	14
7	Fig: 4.3.5-Sequence diagram for make call………………………...………………...	15
8	Fig: 4.3.6 -Sequence diagram for locate phone……...…………..……………………...	16
9	Fig: 5.2.1-Arduino Uno……...…..………………………………………………………	17
10	Fig: 5.2.2 ADXL335……………………………………………………………...	20
11	Fig: 5.2.3-ESP8266………………………………………………………………….	22
12	Fig: 6.1-Functional Diagram……………………………………………………...	25
13	Fig. 6.3-Predefined Gestures………………………………………………………..	27
14	Fig: 6.7.1-Parse dashboard………………………………………………………..	29
15	Fig: 6.7.2- Registered devices in parse.com……………………………………...	30
16	Fig: 6.7.3 –Parse.com analytics…..……………………..……………………………	30
17	Fig: 6.7.4-Temboo Dashboard……...………………………………………………..	31
18	Fig. 8.2-Future Road Map……………………………………………………………	36

List of Tables

Table: 5.2.1 Arduino uno specifications………….........………………………………. 19
Table: 5.3 Component costs…………………………………..……………………….... 24

CHAPTER 1

INTRODUCTION

1.1 Making Sense of Gesture

Gestures are an important aspect of human interaction. People naturally use hand motions to communicate with other people. Gesture based communication is not a new technique, in the pre historic time when language was not developed man used gestures to communicate. Gestures however became attractive for spontaneous interaction with consumer electronics and mobile devices in the context of pervasive computing. A variety of spontaneous gestures, such as finger, hand, body, or head movements are used to convey information in interactions among people. Gestures can hence be considered a natural communication channel with numerous aspects to be utilized in human computer interaction.

Till date, most of our interactions with computers are performed with traditional keyboards, mouse and remote controls designed mainly for stationary interaction. Gesture-based interfaces can enrich and diversify interaction options and provide easy means to interact with the surrounding environment especially for handicapped people who are unable to live their lives in a traditional way. Sometimes, the verbal communication also has limitations in such cases the gesture based communication becomes more effective.

1.2 Making sense of IoT

In most organizations, information travels along familiar routes. Proprietary information is lodged in databases and analyzed in reports and then rises up the management chain. Information also originates externally—gathered from public sources, harvested from the Internet, or purchased from information suppliers. But the predictable pathways of information are changing: the physical world itself is becoming a type of information system. In what's called the Internet of Things, sensors and actuators embedded in physical objects—from roadways to pacemakers—are linked through wired and wireless networks, often using the same Internet Protocol (IP) that connects the Internet. These networks churn out huge volumes of data that flow to computers for analysis. When objects can both sense the environment and communicate, they become tools for understanding complexity and responding to it swiftly. What's revolutionary in all this is that these physical information systems are now beginning to be deployed, and some of them even work largely without human intervention.

Building Blocks

The concept of the Internet of Things is not new (the term itself was coined in 1999), but it is now in the process of becoming a reality thanks to the confluence of several key factors.

First, while still challenging, it is easier and cheaper than ever to produce hardware – some components are open sourced (e.g. Arduino microcontrollers); 3D printing helps with rapid prototyping; specialized providers like Dragon Innovation and PCH can handle key parts of the production process, and emerging marketplaces such as Grand St. help with distribution. Crowdfunding sites like Kickstarter or Indiegogo considerably de-risk the early phase of creating hardware by establishing market demand and providing financing.

Second, the world of wireless connectivity has dramatically evolved over the last few years. The mobile phone (or tablet), now a supercomputer in everyone's hand, is becoming the universal remote control of the Internet of Things. Ubiquitous connectivity is becoming a reality (Wi-Fi, Bluetooth, 4G) and standards are starting to emerge (MQTT). The slight irony of the "Internet of Things" moniker is that things are often connected via M2M (machine to machine) protocols rather than the Internet itself.

Third, the Internet of Things is able to leverage an entire infrastructure that has emerged in related areas. Cloud computing enables the creation of "dumb" (simpler, cheaper) devices, with all the intelligence processed in the cloud. Big data tools, often open sourced (Hadoop), enable the processing of massive amounts of data captured by the devices and will play a crucial role in the space.

Scope of IoT

IoT can connect devices embedded in various systems to the internet. When devices/objects can represent themselves digitally, they can be controlled from anywhere. The connectivity then helps us capture more data from more places, ensuring more ways of increasing efficiency and improving safety and security.

IoT is a transformational force that can help companies improve performance and deliver better results. Businesses in the utilities, oil & gas, insurance, manufacturing, transportation, infrastructure and retail sectors can reap the benefits of IoT by making more informed decisions, aided by the torrent of interactional and transactional data at their disposal. Gesture Controlled Enactor is a prototype which uses IOT technology and gesture to perform tasks.

1.3 Motivation and Goals

Today human-machine interaction is moving away from mouse and pen and is becoming pervasive and much more compatible with the physical world. With each passing day the gap between machines and humans is being reduced with the introduction of new technologies to ease the standard of living. Gestures have played a vital role in diminishing this abyss. In this project, an analysis of "Human-Machine Interaction" using gestures has been presented. Gestures can be captured with the help of an accelerometer. Hence our target interest is to use hand motion based gesture interfaces.

Mobile devices, such as PDAs, mobile phones, and other portable personal electronic devices provide new possibilities for interacting with various applications, if equipped with the necessary devices especially with the proliferation of low-cost MEMS (Micro-Electro- Mechanical Systems) technology. In addition to the technical challenges of implementing gesture recognition, there are also social challenges. Gestures must be simple, intuitive and universally acceptable.

In the modern society, often people prefer to do certain jobs while being engaged in different job. In all such cases gesture based communication is easier. For instance, gesture based technology is used to vary the volume while driving car, while playing games but the technologies that are user friendly as well as affordable to disabled people are inadequate. It is therefore aimed to develop a prototype which is affordable and useful to everyone.

Therefore **objectives** of the project are to:

- Develop an application which helps in locating phone or calling an emergency contact just by using gesture.
- Control home appliances using gesture.

Definition and scope

Often people forget where they have kept the mobile and keep searching for it. A gesture controlled enactor aims to give alerts or ring, when the predefined corresponding gesture is performed. This enables the user to locate the phone. This makes more sense especially for blinds.

The same Gesture controlled enactor also aims to control home appliances using gesture. When a person is away from home but forgot to switch off the lights or fan, coming back to home to switch it off is time consuming. Or consider a scenario where a person is busy working and wants to switch on light/fan he will have to go to the switch board. The GCE aims to avoid these things by gesture. By performing predefined gesture home appliances can be automated.

CHAPTER 2

LITERATURE SURVEY

There are many application evolved to control home appliances. Controlling devices using switches are the most common. Techniques are also developed which does not make use of switches instead, they make use of infrared remote control switch, wireless remote control switch, light activated switches etc. But these technologies have one or other limitations. Laser beams are harmful to mankind. Infra-red (IR) remote control are used only for short distance applications. In such case, if we have system which does not require any radiations or which is not harmful, and cannot travel for a long distance would be great impact. In this concern there are applications developed which makes use of cellphones to control devices[1].

Even this technique has limitations, Often mobiles are inaccessible or away from us and hard to reach. There are also some applications which makes use of speech to control home appliances[2], it might be a mobile app or device which has microphone. But this also has certain limitations like dumb people cannot speak or speech application may not be of use in some restricted areas where absolute silence is maintained In such cases an alternative approach would be warranted to design a wearable device like ring, which can connect or control devices using gestures.

Literature survey reveals that there are many techniques to capture gestures. A good amount of research is being carried out on this field [3] . Direct visual matching of images for gesture recognition and image retrieval using Hidden Markov Model(HMM) is also proposed[4]. A hidden Markov model is a collection of finite states connected by transitions. Meaningful gestures may be very complex, containing simultaneous motions of a number of points. However, these complex gestures should be easily specifiable. In general, gestures can be specified either by example or by description. In the former, each application has a training session in which examples of different gestures are collected for training the models. The trained models are the representations of all gestures that the system must recognize. In the latter method of specification, a description of each gesture is written in a gesture description language, which is a formal language in which the syntax of each gesture is specified. The key idea of HMM-based gesture recognition is to use multi-dimensional HMM representing the defined gestures. The parameters of the model are determined by the training data. The trained models represent the most likely human performance and are used to evaluate new incoming gestures.

Many techniques are available. But disabled people may prefer hand gesture. A sequence of real-time depth image data acquired by active sensing hardware is used for recognition of hand gesture[5]. Hand gesture recognition using RADAR are also developed and are in market, Compared to cameras, radar has very high positional accuracy and so can sense tiny motions but these involve complex process and are costly. The proliferation of accelerometer on consumer electronics brought a new vista for interaction based on gesture. For such an interaction uWave algorithm was found to be efficient[6].

uWave matches the accelerometer readings for an unknown gesture with those for a vocabulary of known gestures, or templates, based on dynamic time warping (DTW). uWave requires a single training sample per vocabulary gesture. The strength of uWave in user-dependent gesture recognition makes it ideal for personalized gesture-based interaction. With uWave, users can create simple personal gestures for frequent interaction. Its simplicity, efficiency, and minimal hardware requirement (a single accelerometer) make uWave have the potential to enable personalized gesture-based interaction with a broad range of devices. uWave requires as few as a single training sample for each gesture and delivers competitive accuracy . It is efficient and thus amenable to implementation on resource-constrained platforms. Frame-based Descriptor and multi-class SVM (FDSVM) approach is proposed for gesture recognition which uses 3D accelerometer[7]. With FDSVM, firstly, the acceleration data of a gesture is collected and represented by a frame-based descriptor, to extract the discriminative information. Then a Support Vector Machine (SVM)-based multi-class gesture classifier is built for recognition in the nonlinear gesture feature space. Compared to the other approaches, gesture with accelerometer is robust, cost effective and gives equally reliable results.

In view of this, the majority of the new generations of smart phones, PDAs, and personal electronic devices are embedded with an accelerometer for various applications. Small wireless devices containing accelerometers could be integrated into clothing, wristwatches or other personal electronic devices to provide a means for interacting with different environments [8]. By defining some simple gestures, these devices could also be used to control home appliances. This prompted to take up this project. Although, several applications exist to assist disabled people, Gesture controlled enactor is user friendly and affordable for everybody.

CHAPTER 3

SYSTEM REQUIREMENT SPECIFICATION

3.1 Functional Requirements

Functional requirements of the project are listed below:

Recognise Gesture

- When accelerometer is moved or when a gesture is performed, gestures which are predefined are matched and only for those gestures the action should be taken.
- For a gesture which is not predefined no action should be performed.
- Gesture data should be considered only when the button is pressed before moving
- accelerometer and pressed again after the gesture to indicate the beginning and end of gesture.

Performing action on recognition of gesture

- When the gesture recognised is meant to give alerts, parse.com should push the alert notification to mobile app.
- When the gesture recognised is meant to call emergency contact, parse.com should notify the app to call.
- When gesture recognised is meant to turn on and off light or fan, arduino should signal corresponding sensor.

3.2 Non Functional Requirements

1. Availability: The data values are recorded only when the accelerometer is being held/worn. Also there must be a constant access to the internet through the wifi module.
2. Reliability: User can rely on GCE because the values recorded from the accelerometer sensor are fine enough to record the pre defined gestures and are processed by efficient algorithms.
3. Cost efficiency: The prototype uses basic components which are available at low cost.

3.3 System Requirements

System requirements of Gesture Controlled Enactor are listed as below.

3.3.1 Hardware Requirements

- Arduino Uno
- ADXL3xx Accelerometer
- ESP8266 Wifi module
- Relay Switch
- Light Bulb

3.3.2 Software Requirements

- Arduino IDE
- Eclipse IDE
- Android Studio
- Parse.com
- Temboo

CHAPTER 4

SYSTEM MODELING DESIGNS

4.1 System Design Chart

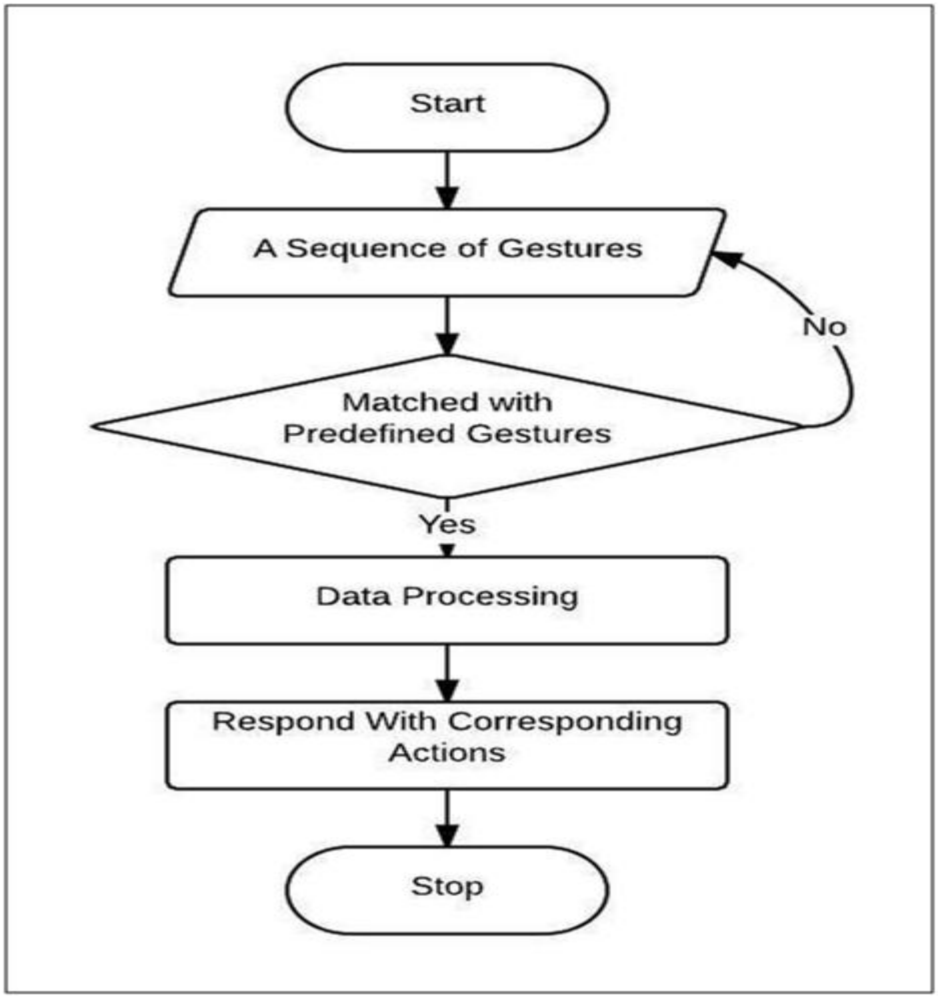


Fig: 4.1-System Design chart

The gesture data has to go through the preprocessing stage in which the cleansing of the data is performed to ensure that the data is in the exact form which is preferred by the system.

i. Pre define list of some standard gesture values and the actions that should be performed for a particular gesture.

ii.The next step is to capture the gesture using the adxl3xx accelerometer sensor. The gesture values available in the form of coordinates are noted.

iii.The gesture values obtained from step ii. are checked to see if they match with any of the pre defined values from step i.

iv.If the values match then the values are processed and goto step vi.

v. If the values do not match go back to step ii.

vi. Now the gesture values match hence perform the corresponding actions as specified under step 'i'.

4.2 Use Case Diagram

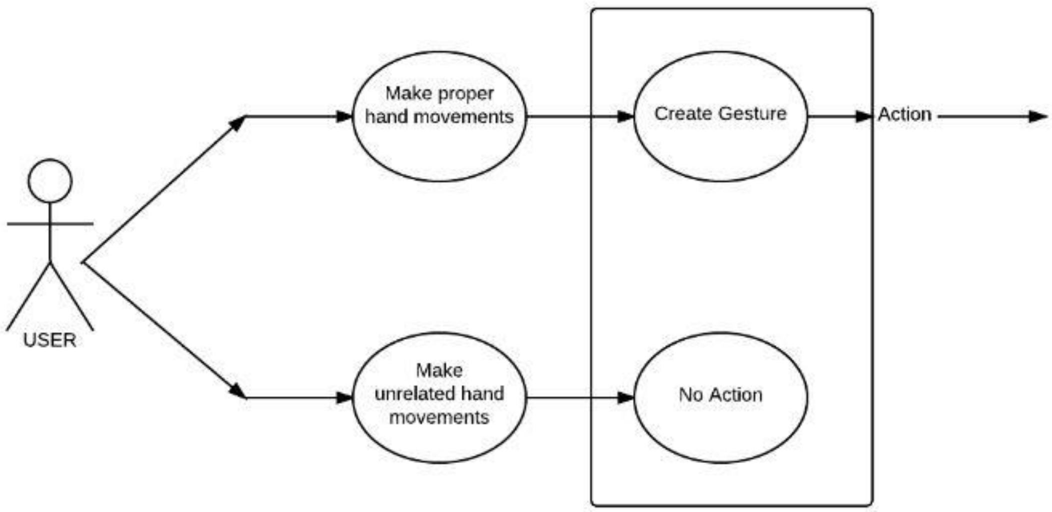


Fig: 4.2-Use case diagram of GCE

4.3 Sequence Diagram

Sequence diagram for individual action is shown in the below figures.

Turning on the Bulb or LED

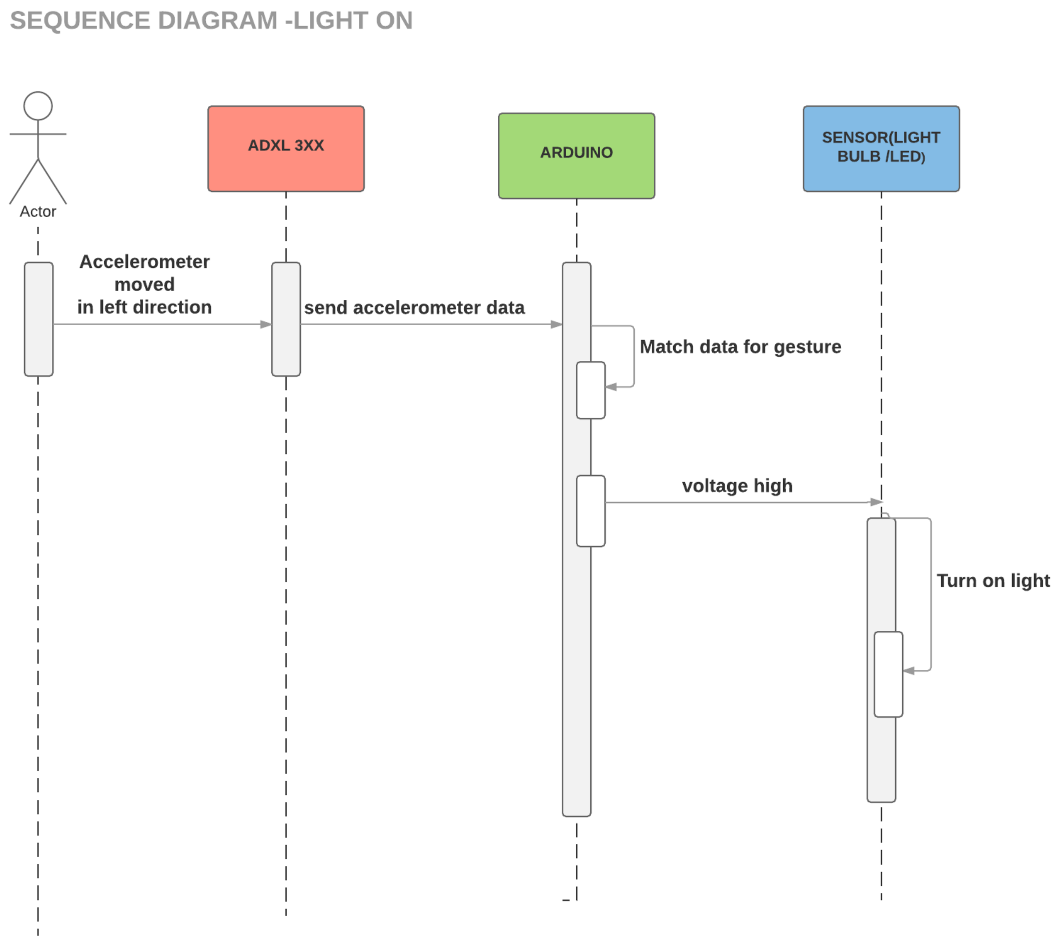


Fig: 4.3.1-Sequence diagram for turn on light

Turning off the Bulb or LED

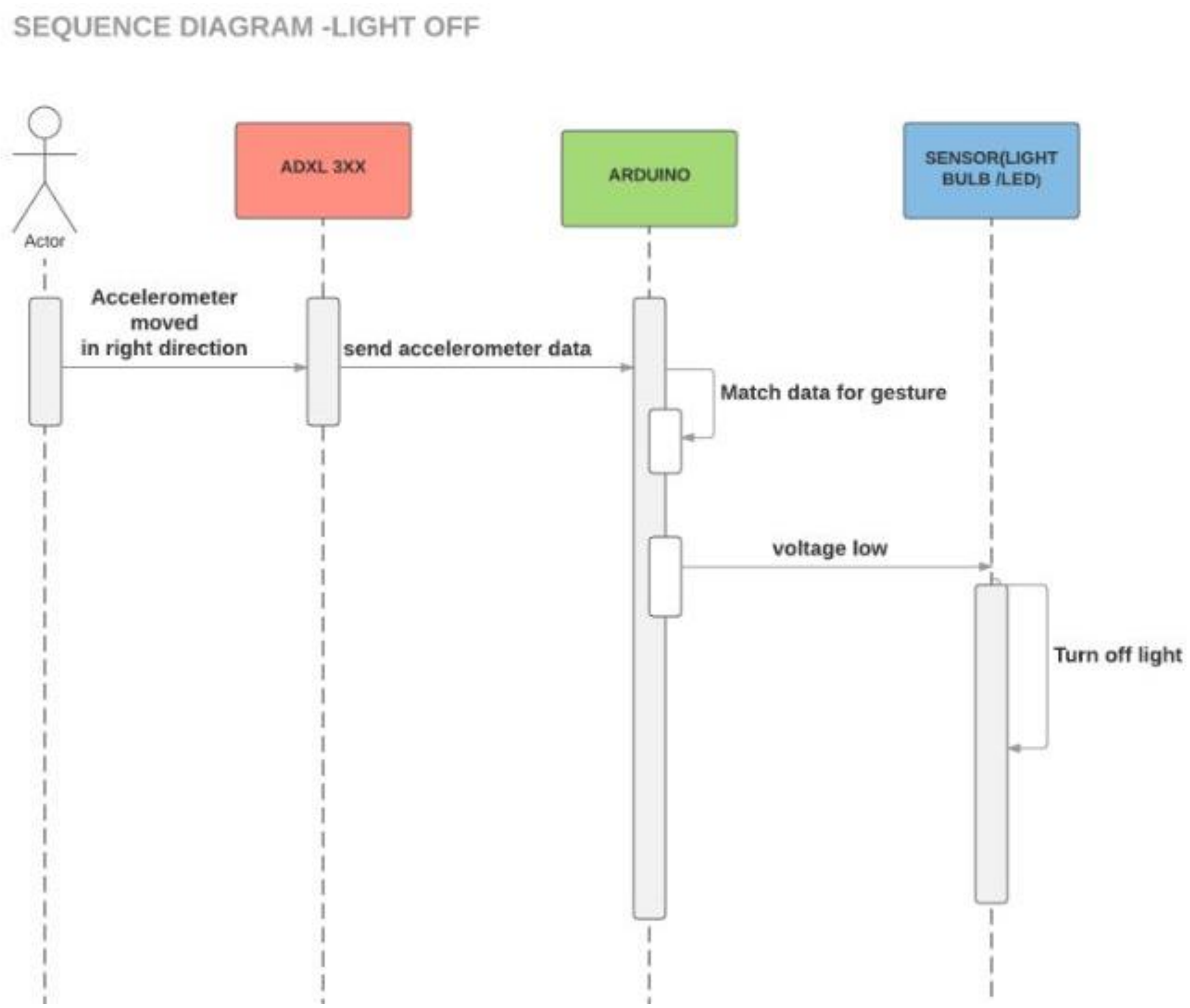


Fig: 4.3.2-Sequence diagram for turn off light

Turning On the Fan

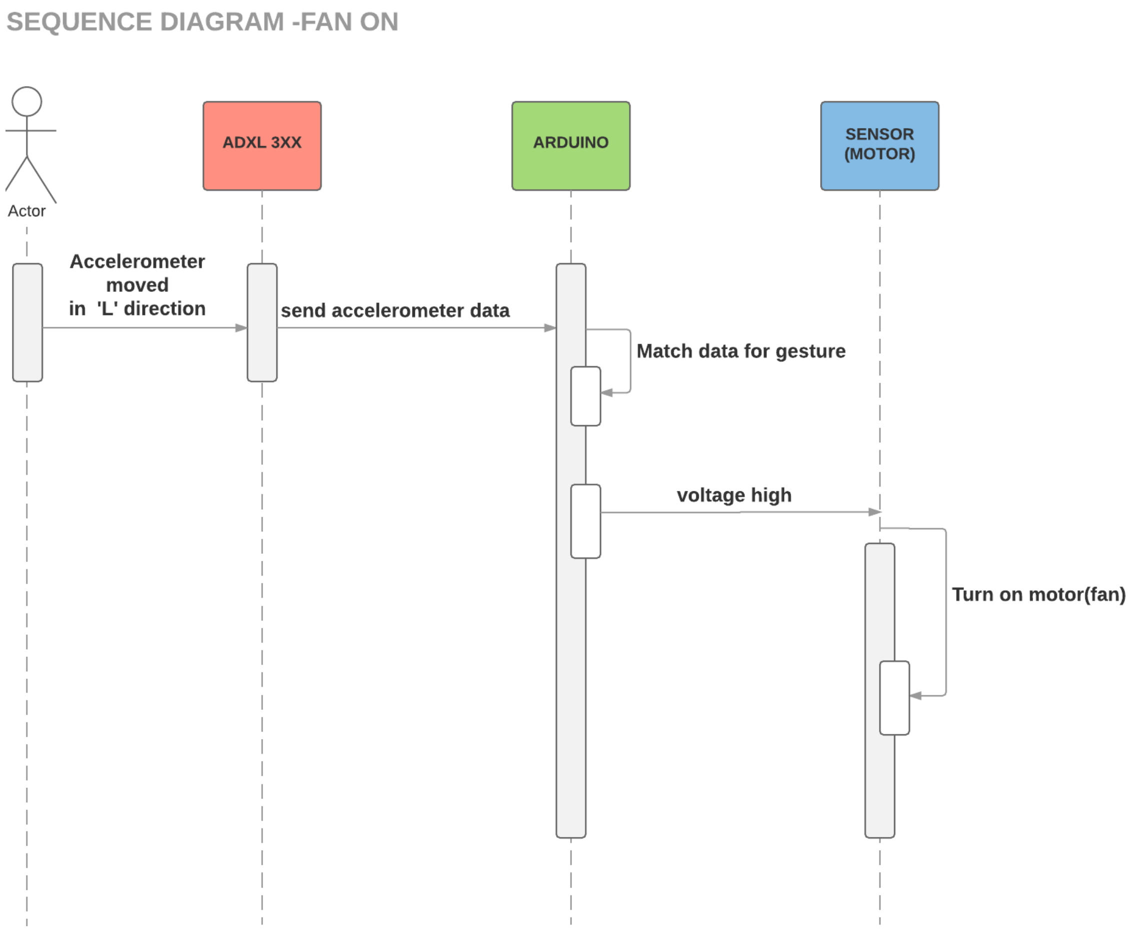


Fig: 4.3.3-Sequence diagram for turn on fan

Turning Off the Fan

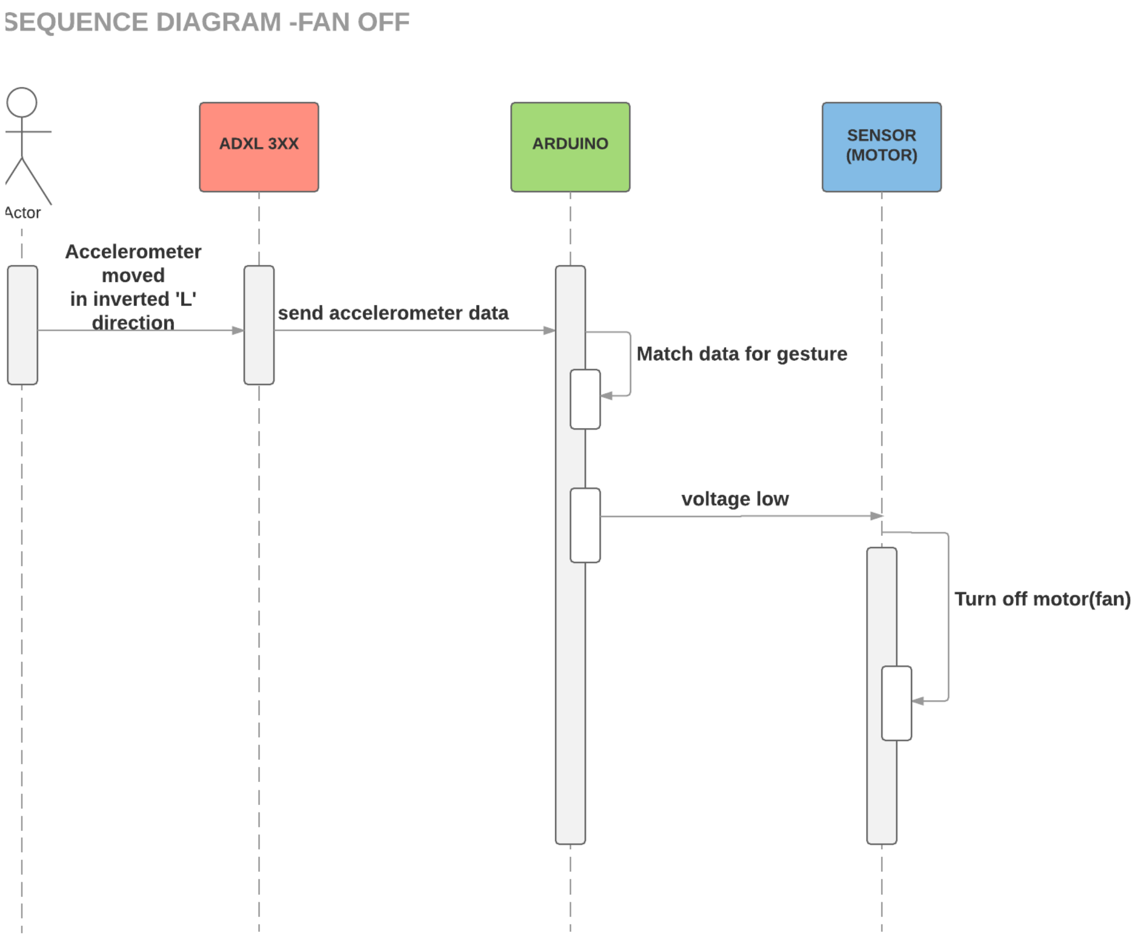


Fig: 4.3.4-Sequence diagram for turn off fan

Make an emergency contact call

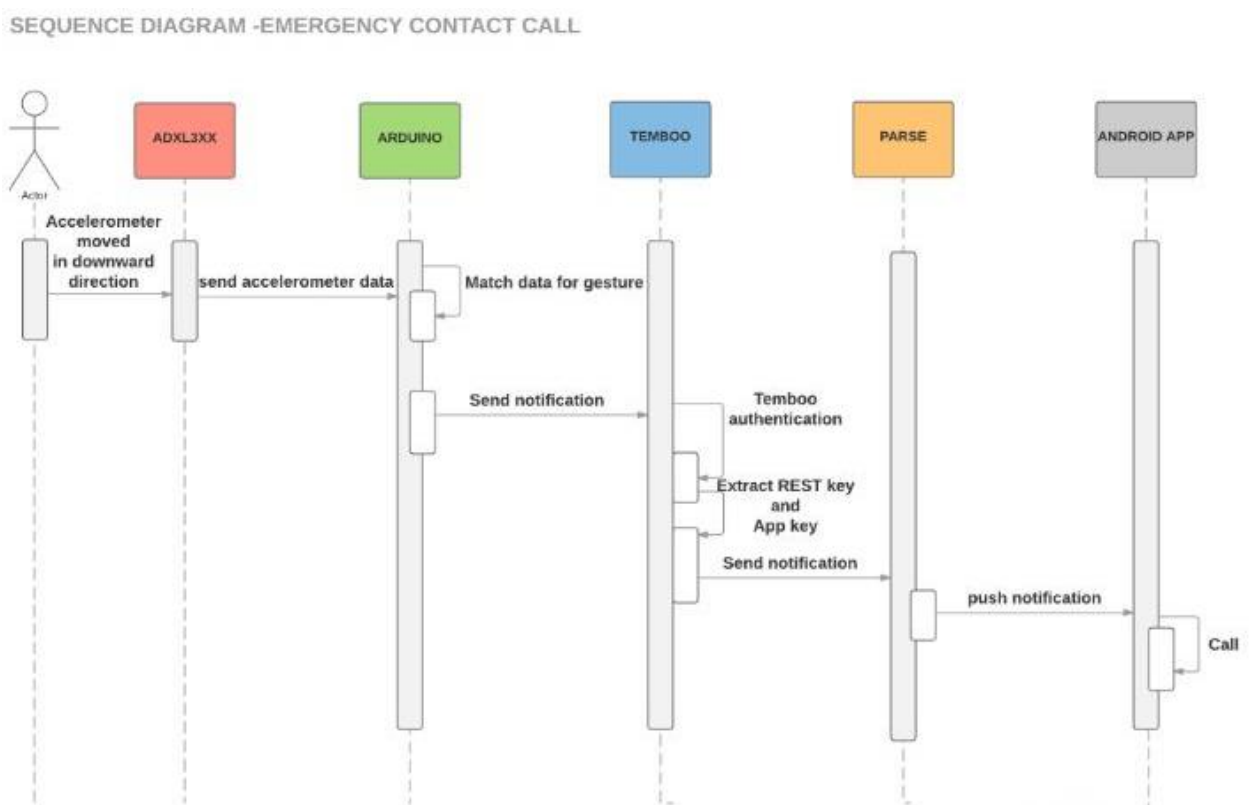


Fig: 4.3.5-Sequence diagram for make call

Locate the Phone

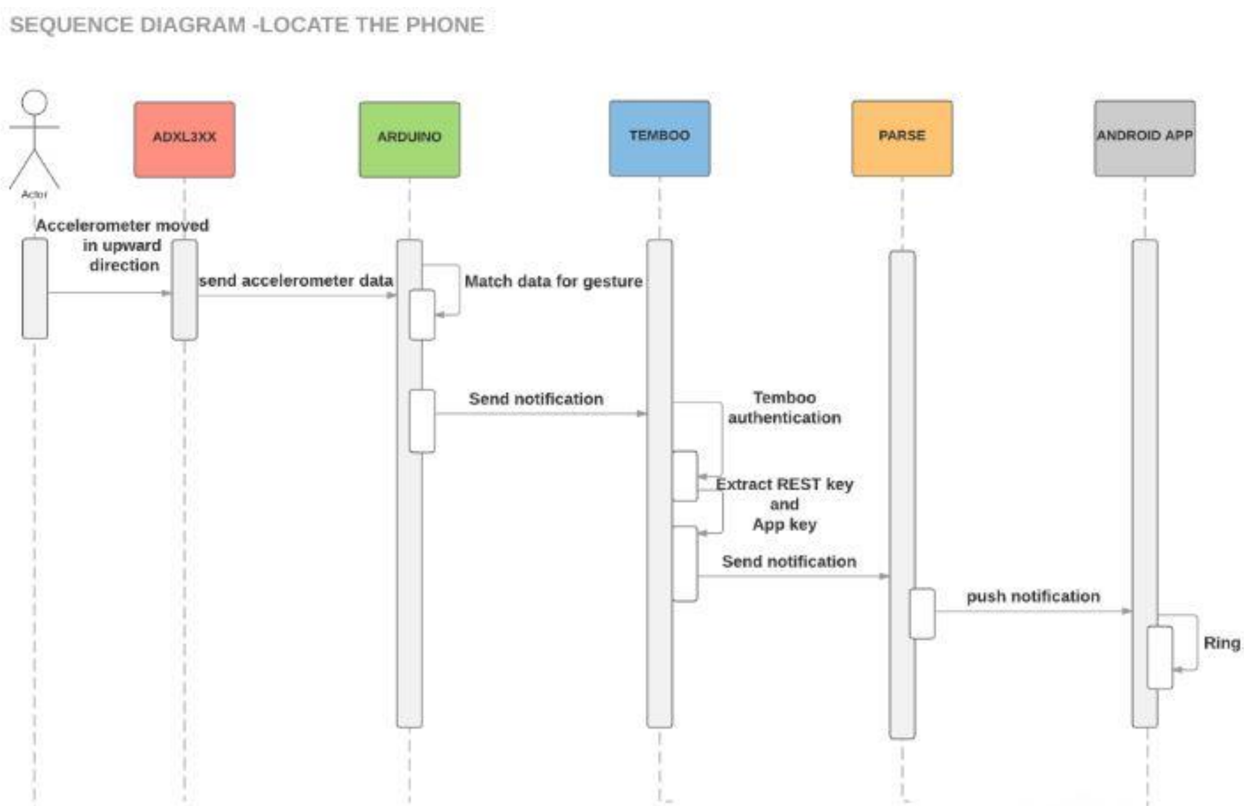


Fig:4.3.6 -Sequence diagram for locate phone

CHAPTER 5

HARDWARE

5.1 Introduction

This chapter introduces some of the prominent hardware components that is being used in the project and layout of the hardware design used for the project.

5.2 Components

Hardware components of the project are explained below.

5.2.1 Arduino UNO R3

Arduino is an open-source physical computing platform based on a simple I/O board and a development environment that implements the Processing/Wiring language.

Fig: 5.2.1-Arduino Uno

The Arduino Uno is a microcontroller board based on the ATmega328. It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller. It can be powered with a USB cable or power it with AC-to-DC adapter or battery to get started.

Arduino Architecture

Arduino's processor basically uses the Harvard architecture where the program code and program data have separate memory. It consists of two memories- Program memory and the data memory.The code is stored in the flash program memory, whereas the data is stored in the data memory. The Atmega328 has 32 KB of flash memory for storing code (of which 0.5 KB is used for the bootloader), 2 KB of SRAM and 1 KB of EEPROM and operates with a clock speed of 16MHz.

Purpose of Using Arduino

1. It is inexpensive
2. It comes with an open source hardware feature which enables users to develop their own kit using already available one as a reference source.
3. The Arduino software is compatible with all types of operating systems like Windows, Linux, and Macintosh etc.
4. It also comes with open source software feature which enables experienced software developers to use the Arduino code to merge with the existing programming language libraries and can be extended and modified.
5. It is easy to use for beginners.
6. We can develop an Arduino based project which can be completely stand alone or projects which involve direct communication with the software loaded in the computer.
7. It comes with an easy provision of connecting with the CPU of the computer using serial communication over USB as it contains built in power and reset circuitry.

Specification

Arduino uno specifications are listed as below.

Table: 5.2.1-Arduino uno specifications

Microcontroller	ATmega328
Operating Voltage	5V
Input Voltage(recommended)	7-12V
Input Voltage (limits)	6-20 V
Digital I/O Pins	14 (of which 6 provide PWM output)
Analog Input Pins	6
DC Current per I/O Pin	40 mA
DC Current for 3.3V Pin	50 mA
Flash Memory	32 KB (ATmega328) of which 0.5 KB used by bootloader
SRAM	2 KB (ATmega328)
EEPROM	1 KB (ATmega328)
Clock Speed	16 MHz

Arduino is chosen for its excellent community support and IDE, which helps in speeding up the prototyping of the project.

5.2.2 ADXL3xx Accelerometer

The ADXL335 is a small, thin, low power, complete 3-axis accelerometer with signal conditioned voltage outputs. The product measures acceleration with a minimum full-scale range of ±3 g.

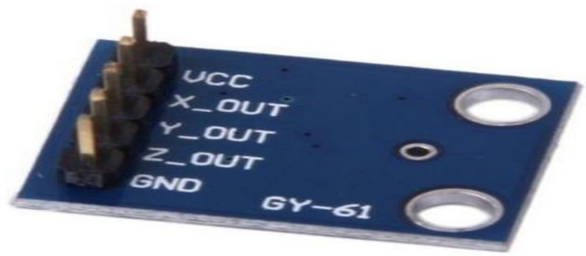


Fig: 5.2.2 ADXL335

It can measure the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion, shock, or vibration. The sensor is a polysilicon surface-micromachined structure built on top of a silicon wafer. Polysilicon springs suspend the structure over the surface of the wafer and provide a resistance against acceleration forces. Deflection of the structure is measured using a differential capacitor that consists of independent fixed plates and plates attached to the moving mass. The fixed plates are driven by 180° out-of-phase square waves. Acceleration deflects the moving mass and unbalances the differential capacitor resulting in a sensor output whose amplitude is proportional to acceleration. Phase-sensitive demodulation techniques are then used to determine the magnitude and direction of the acceleration.

The user selects the bandwidth of the accelerometer using the CX, CY, and CZ capacitors at the XOUT, YOUT, and ZOUT pins. Bandwidths can be selected to suit the application, with a range of 0.5 Hz to 1600 Hz for X and Y axes, and a range of 0.5 Hz to 550 Hz for the Z axis.The ADXL335 is available in a small, low profile, 4 mm × 4 mm × 1.45 mm, 16-lead, plastic lead frame chip scale package.

The ADXL345 is a small, thin, low power, 3-axis accelerometer with high resolution (13-bit) measurement at up to ±16*g*. Digital output data is formatted as 16-bit twos complement and is accessible through either a SPI (3- or 4-wire) or I2C digital interface.

The ADXL345 is well suited for mobile device applications. It measures the static acceleration of gravity in tilt-sensing applications, as well as dynamic acceleration resulting from motion or shock. Its high resolution (4 m*g*/LSB) enables measurement of inclination changes less than 1.0°.

Several special sensing functions are provided. Activity and inactivity sensing detect the presence or lack of motion and if the acceleration on any axis exceeds a user-set level. Tap sensing detects single and double taps. Free-fall sensing detects if the device is falling. These functions can be mapped to one of two interrupt output pins. An integrated, patent pending 32-level first in, first out (FIFO) buffer can be used to store data to minimize host processor intervention.Low power modes enable intelligent motion-based power management with threshold sensing and active acceleration measurement at extremely low power dissipation.

Principles of Operation

Most accelerometers are Micro-Electro-Mechanical Sensors (MEMS). The basic principle of operation behind the MEMS accelerometer is the displacement of a small proof mass etched into the silicon surface of the integrated circuit and suspended by small beams. Consistent with Newton's second law of motion (**F = ma**), as an acceleration is applied to the device, a force develops which displaces the mass. The support beams act as a spring, and the fluid (usually air) trapped inside the IC acts as a damper, resulting in a second order lumped physical system. This is the source of the limited operational bandwidth and non-uniform frequency response of accelerometers.

ADXL335 Features

- 3-axis sensing
- Small, low profile package

- 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

ADXL345 Features

- Ultralow power: as low as 40 µA in measurement mode and 0.1 µA in standby mode at VS = 2.5 V (typical)
- Power consumption scales automatically with bandwidth
- User-selectable resolution
- Fixed 10-bit resolution
- Full resolution, where resolution increases with g range, up to 13-bit resolution at ±16 g (maintaining 4 mg/LSB scale factor in all g ranges)
- Embedded, patent pending FIFO technology minimizes host processor load
- Tap/double tap detection
- Activity/inactivity monitoring

5.2.3 ESP8266 WiFi Module

ESP8266 is an impressive, low cost WiFi module suitable for adding WiFi functionality to an existing microcontroller project via a UART serial connection.The ESP8266 WiFi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to our WiFi network.

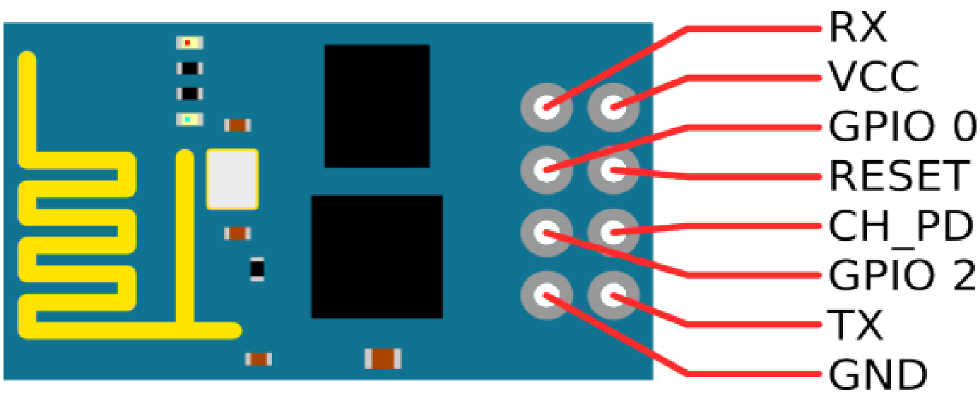


Fig: 5.2.3-ESP8266

This module has a powerful enough on-board processing and storage capability that allows it to be integrated with the sensors and other application specific devices through its GPIOs with minimal development up-front and minimal loading during runtime.

Hardware Connections

1. The ESP8266 requires 3.3 V power and not 5 V
2. The ESP8266 needs to communicate via serial at 3.3V and does not have 5V tolerant inputs, so we need level conversion to communicate with a 5V microcontroller like most Arduinos use.

Features

1. Integrated TCP/IP protocol stack
2. 1MB Flash Memory
3. Wi-Fi Direct (P2P), soft-AP
4. Integrated low power 32-bit CPU could be used as application processor
5. Integrated temperature sensor
6. Supports antenna diversity
7. SDIO 2.0, SPI, UART
8. Standby power consumption of < 1.0mW
9. Wake up and transmit packets in < 2ms

5.3 Project Cost

The table below shows the cost of the components used in project.

Table: 5.3-Component costs

Item	Quantity	Price	Total
Arduino uno R3	1	555	555
ADXL3xx Accelerometer	1	450	450
ESP8266 WIFI MODULE	1	285	285
Solderless Breadboard	1	160	160
USB Cable	1	95	95
Jumper Wires	25	5	125
Relay Switch	1	139	139

CHAPTER 6

IMPLEMENTATION

6.1 Introduction

Gesture based automation is the new way to interact with devices present in our surroundings. In this project we are using sensors instead of camera for gesture recognition, because it provides better consistency, environment independency, portability, lesser processing, low cost and energy efficiency. In gesture recognition using an accelerometer sensor, gestures are represented by characteristic patterns of incoming signal data, i.e. vectors representing the current acceleration of the the wearable device in all three dimensions. Hence, a system pipeline preparing and analyzing this vector data in order to train as well as recognize patterns for distinct gestures is required.

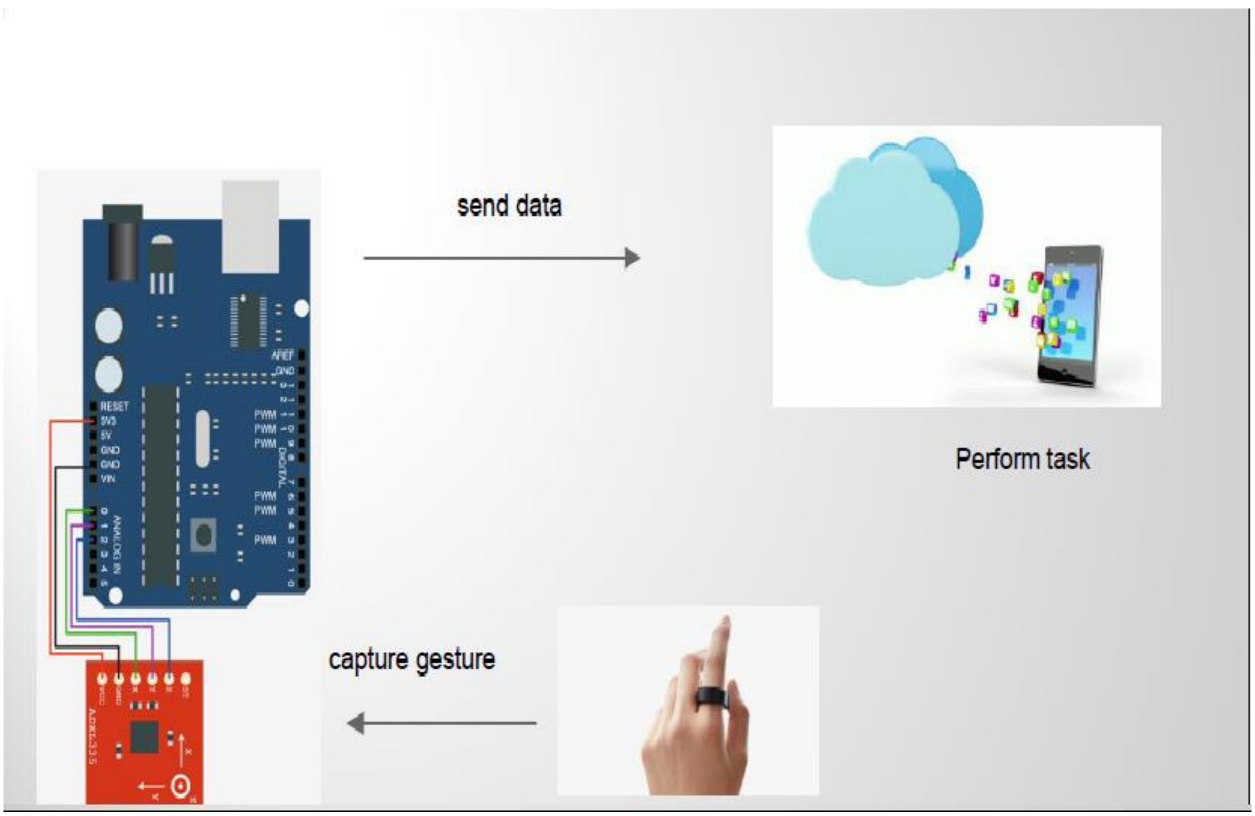


Fig: 6.1-Functional Diagram

The project mainly consists of two parts:

1) Wearable gesture capturing device: Design a wearable device consisting of accelerometers and sensors to measure the gesture input.

2) Application to control the device: Build an application which maps the gesture reading with the predefined gestures and performs the required task.

Since the actions are taken on the commands of gesture, the first task is to capture gesture and recognize it. The hand gestures used in the proposed prototype, to perform the actions are basically some simple gestures which the user will find useful and easy. These operations are schematically shown in figure 6.1.

6.2 Reading Accelerometer Data

To perform a gesture we are making use of accelerometer. Accelerometer is connected to Arduino uno. Whenever the accelerometer is moved we can receive the data of x,y,z coordinate values, speed with which the accelerometer is moved and velocity. By using static gravity value we can get the tilt of the device with respect to earth. These data is sent to arduino from accelerometer. The accelerometer data is processed and matched with predefined concept. Arduino receives the accelerometer data and recognizes the gesture to take corresponding action.

6.3 Data Processing in Arduino

Once we have the gesture data in arduino from accelerometer, we need to match the gesture data to take the corresponding action. The steps are as follows:

Predefined gestures and corresponding actions: The prototype is having some prescribed gesture data to which the corresponding actions are also defined. Gestures are explained as below and shown in figure 6.3.

Accelerometer moved in up direction: When the accelerometer is moved in up direction, the mobile phone should give alerts or should ring, so that the user can locate his phone. For example, for a blind person to locate his phone it takes lot of efforts, so by moving the wearable accelerometer in up direction the phone will ring, which helps the user to locate his phone.

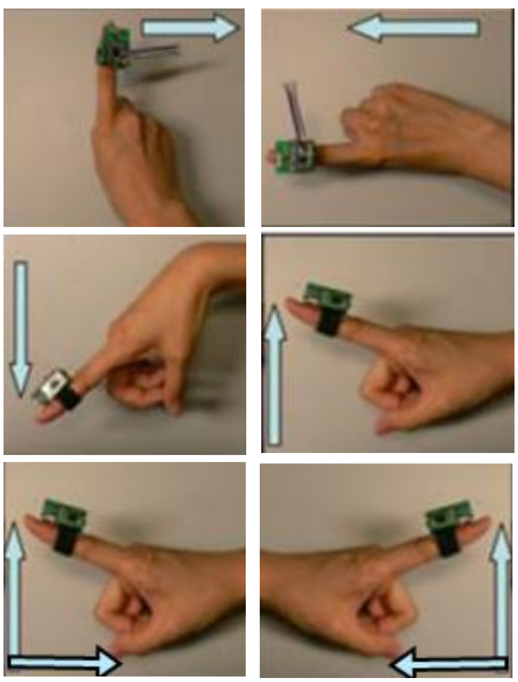
Accelerometer moved in downward direction: When accelerometer is moved downwards, an automatic call should go to the emergency contact.

Accelerometer moved in left direction: When accelerometer is moved towards left, light should turn on.

Accelerometer moved in right direction: When accelerometer is moved towards right, light should turn off.

Accelerometer moved in L shape: When the gesture is of L shape the fan should turn on.

Accelerometer moved in _| shape: When the gesture is opposite to the previous gesture fan should turn off.



To operate light

To send notifications to app

To operate fan

Fig. 6.3-Predefined Gestures

6.4 Controlling light and fan

Operations like turn on light, turn off light, turn on the fan and turn off the fan, are to be handled for particular gestures. Whenever the corresponding gesture is performed and matched arduino notifies the sensors or it handles the operation.

6.5 Sending Data to Cloud

This project makes use of Temboo to send data notifications to Parse.com using the application id of the android app and by using REST API to achieve all these above mentioned tasks we need to send data or we need to notify the devices, for that we use cloud. Once data is matched in arduino, arduino sends notification to Parse.com using Temboo agent.

Parse.com : It is a back-end tool for mobile developers that help mobile developers store data in the cloud, manage identity log-ins and handle push notifications.

Temboo: With the resources available that is with arduino uno and esp8266 it is very difficult to send data or to communicate with Parse.com as arduino uno+esp8266 doesn't directly support http requests. Third party libraries have to be used or we have to choose some other protocol. One way is to transfer the app in Parse.com to parse server and hosts the app with some cloud based backend such as mongolab. Another way is to make use of agents. Temboo is one such agent or library. Whenever the code containing Choreos is run, it calls up the Temboo platform, where the longer, more complicated code behind every Choreo is executed. Temboo's Library contains thousands of Choreos that handle API interactions, work with databases, perform code utility functions, and more.

6.6 Receiving Push Notification on Android App

An Android app which performs the action based on gesture. Android app is connected to the parse.com using app key and rest api. Parse.com can send push notifications to the Android app.

Once the app receives the notification corresponding action will be taken. To locate the phone we just need an alert or the mobile should ring. For this, when the notification from the parse says "Give alerts" the android app gives an alert notification. When the parse notification says "call emergency contact" app will call the emergency contact which is previously stored.

6.7 Snapshots

Parse Dashboard

In figure below we can see the notifications received in parse. Which are sent from arduino.

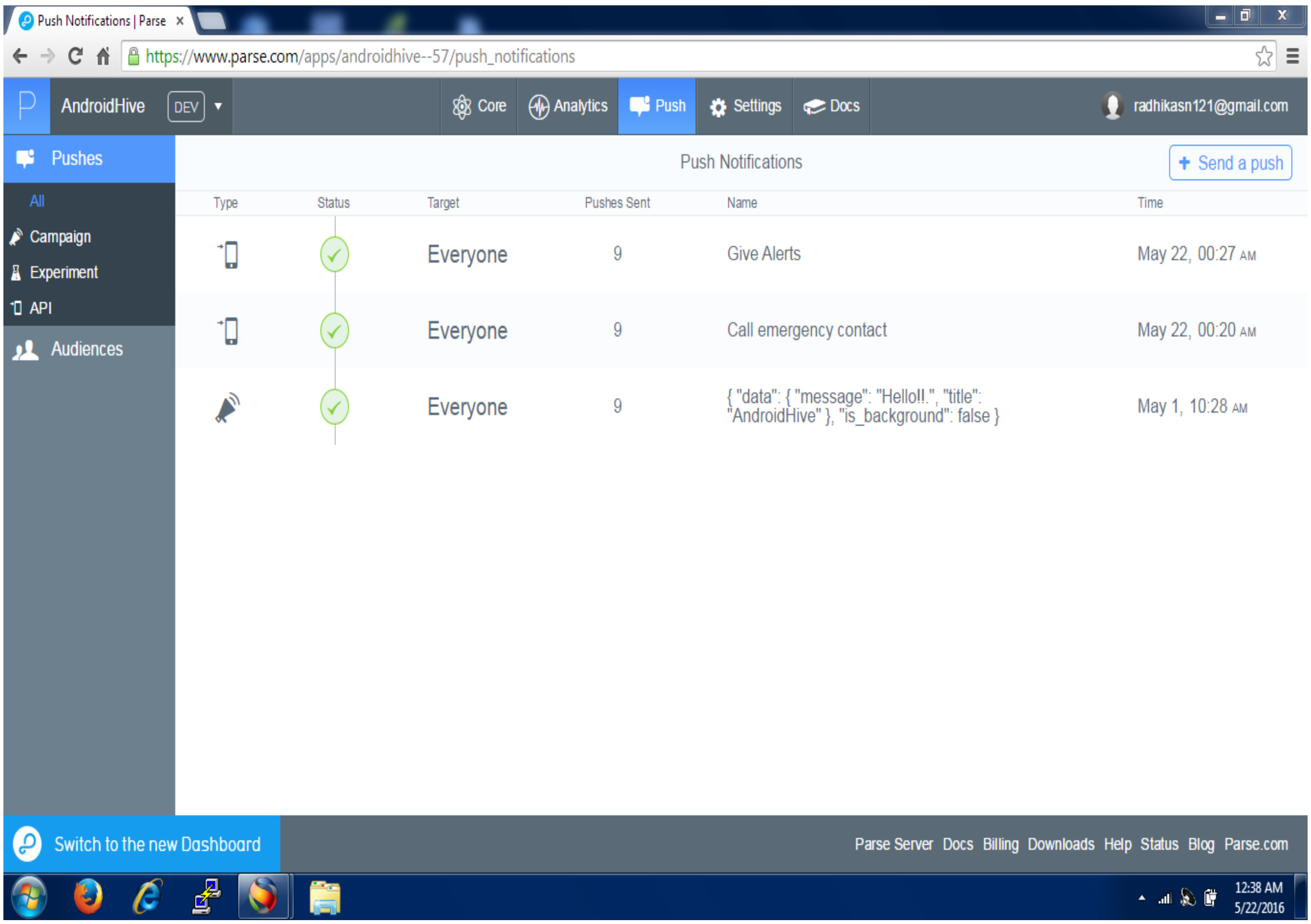


Fig: 6.7.1-Parse dashboard

Devices Registered in Parse

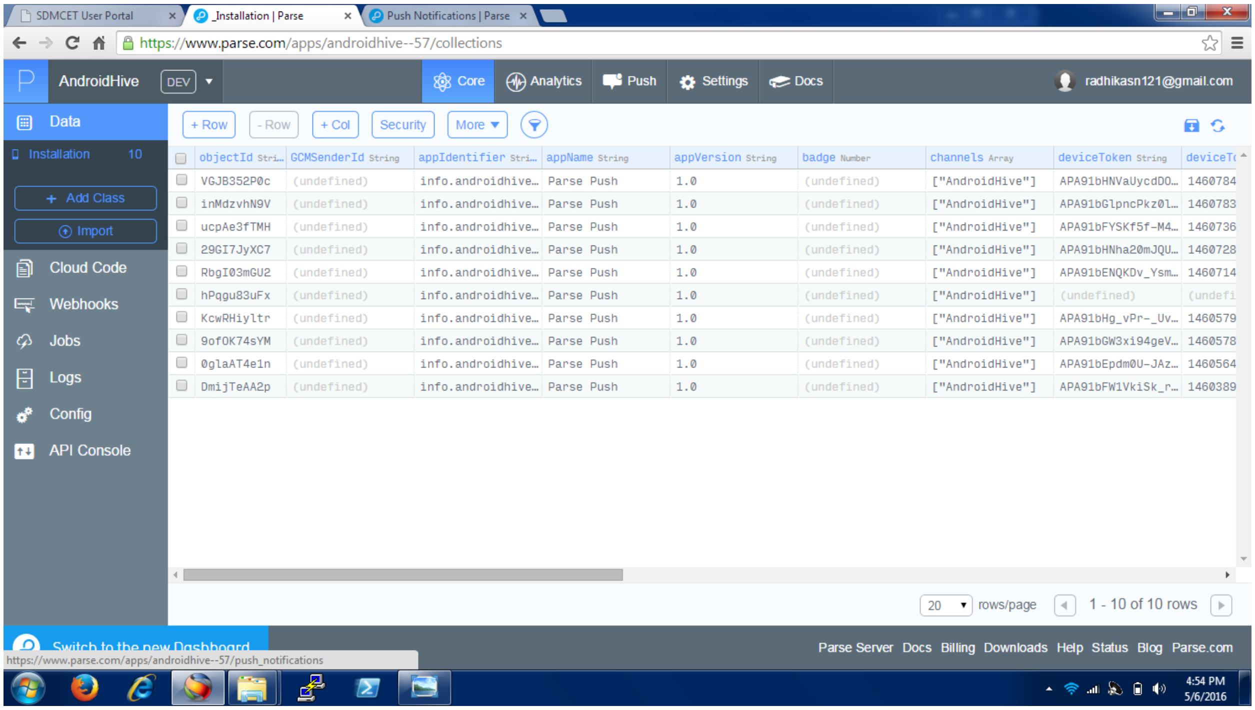


Fig: 6.7.2- Registered devices in parse.com

Analytics

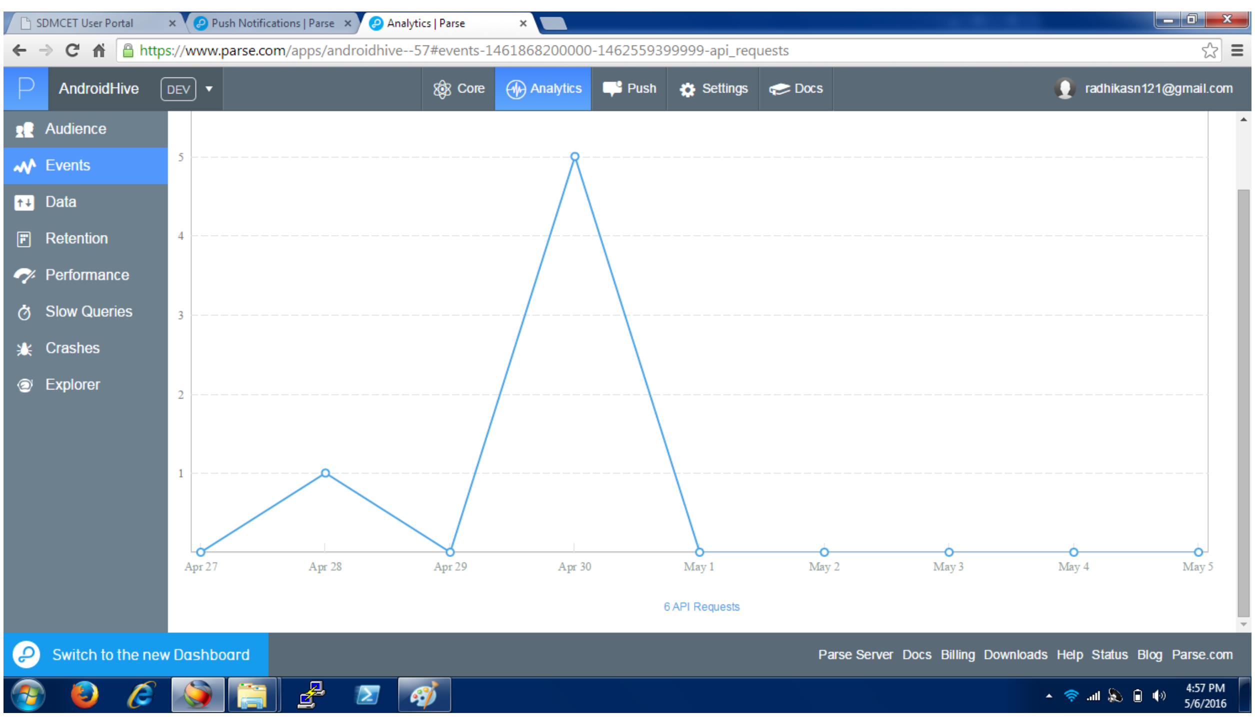


Fig: 6.7.3 –Parse.com analytics

Temboo Dashboard

Figure below shows Temboo dashboards where we can see the notification sent to the parse.com

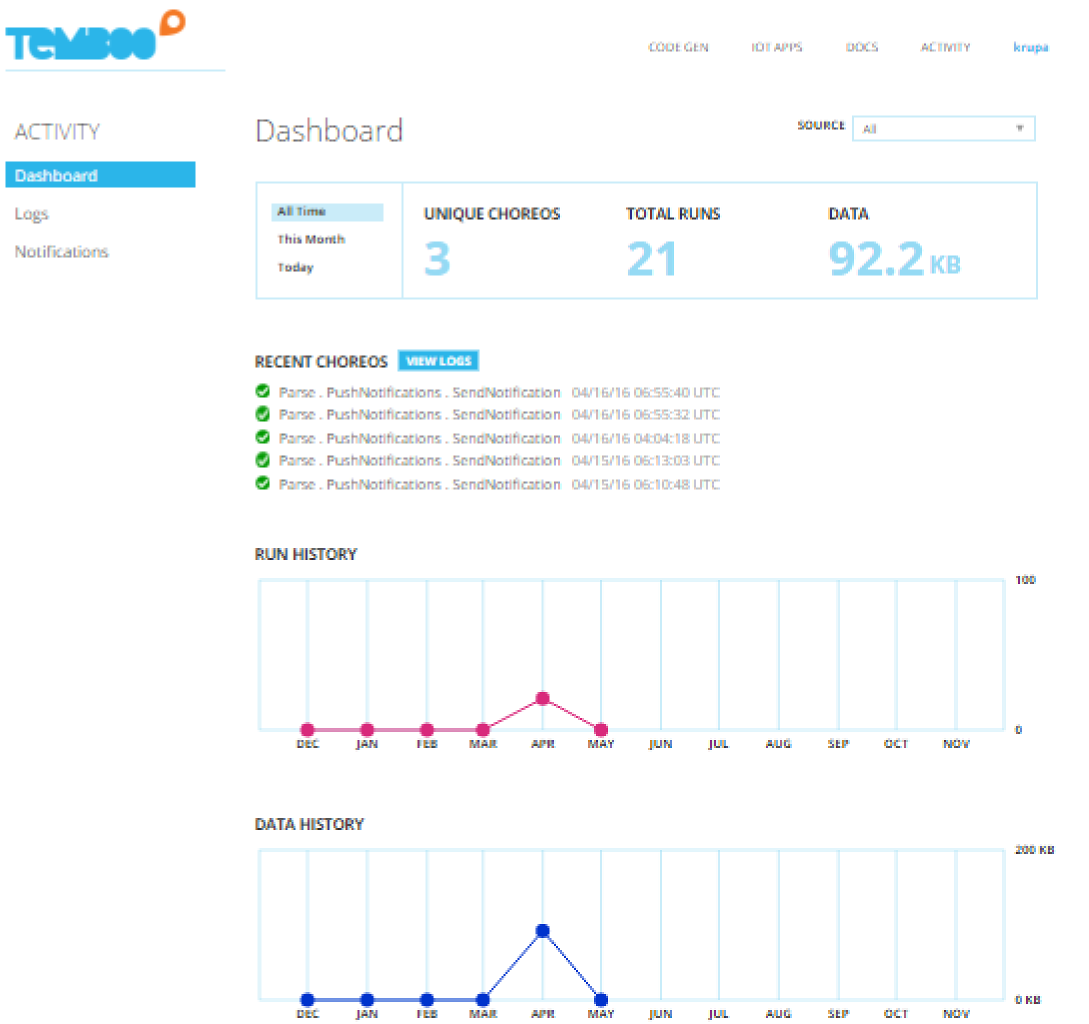


Fig: 6.7.4-Temboo Dashboard

CHAPTER 7

TESTING

7.1 Test case Designs

Test case id : T001

Test case : Matching predefined gesture

Preconditions : Accelerometer connected to arduino

Input test data : Accelerometer X, Y , Z readings

Priority : High

Steps to be executed : Accelerometer readings are recorded for performed gesture. Matching of gesture to our personalized gesture takes place.

Expected result : Perform the action like turn on/off light, give alerts/call to emergency contact as defined

Actual result : No deviation from expected result

Pass/Fail : Pass

Test case id : T002

Test case : Defining boundary for input accelerometer reading

Preconditions : Accelerometer connected to arduino

Input test data : Accelerometer X, Y , Z readings

Priority : High

Steps to be executed : Accelerometer readings are recorded for gestures performed and check whether they lie outside the valid input boundary values.

Expected result : Out of boundary values detected and appropriate message is shown in arduino console and no action is performed

Actual result : No deviation from expected result

Pass/Fail : Pass

Test case id	: T003
Test case	: Test the mechanism of reading of data
Preconditions	: Accelerometer connected to arduino
Input test data	: Gesture X, Y, Z values
Priority	: Normal
Steps to be executed	: Use "Reset Button" to start the reading the data and use the same button to stop reading the data
Expected result	: Accelerometer reading should be recorded only between the start and stop of the button.
Actual result	: No deviation from expected result
Pass/Fail	: Pass

Test case id	: T004
Test case	: To test the “Always-on” connectivity mechanism for an Iot Device (GCE)
Preconditions	: Accelerometer connected to arduino and app installed in phone
Input test data	: Input Gesture
Priority	: Normal
Steps to be executed	: Perform a predefined valid gesture and wait for the corresponding action to take place. For example, moving upwards notifies the location of the phone.
Expected result	: Instant response to the gesture input readings from the arduino with no such delay
Actual result	: No deviation from expected result
Pass/Fail	: Pass

Test case id	: T005
Test case	: To test the GCE during no internet connection
Preconditions	: Accelerometer connected to arduino and app installed in phone
Input test data	: Input Gesture
Priority	: Normal
Steps to be executed	: Perform a predefined valid gesture and wait for the corresponding action to take place
Expected result	: No response, complete downtime of the system
Actual result	: No deviation from expected result
Pass/Fail	: Pass

Test case id	: T006
Test case	: To test the authentication and security issues
Preconditions	: Accelerometer connected to arduino, app installed in phone and user should be registered.
Input test data	: Input Gesture
Priority	: Normal
Steps to be executed	: Perform a predefined valid gesture and wait for the corresponding action to take place.
Expected result	: Action should be performed only for the authenticated users
Actual result	: No deviation from expected result
Pass/Fail	: Pass

CHAPTER 8

CONCLUSION

8.1 Conclusion and Outcomes

Gesture controlled enactor has 6 predefined gestures (up, down, right, left, L and invert L). The prototype developed works successfully for these gestures. Automation is achieved in home appliances like fan and light. Locating phone and calling emergency contact via gesture is achieved. This prototype is developed in IoT platform.

By working on this project, we gained experience and knowledge regarding the field IoT. We have learnt to process accelerometer data to recognize gesture. The project enabled us to learn about hardware, how software reads the hardware and to learn gesture based human interaction with hardware. This project has given exposure to tools like parse.com. We have learnt to use agents like Temboo to connect hardware to the cloud when the hardware component is less efficient in achieving the same. Learn to use arduino IDE. The project has given an experience to subdivide the task and integrating them as a whole. The project has given us an opportunity to work in team, to use resources efficiently and to realize the importance of team spirit and teamwork in achieving the target goals.

8.2 Future Scope

The below figure shows the future road map of GCE.

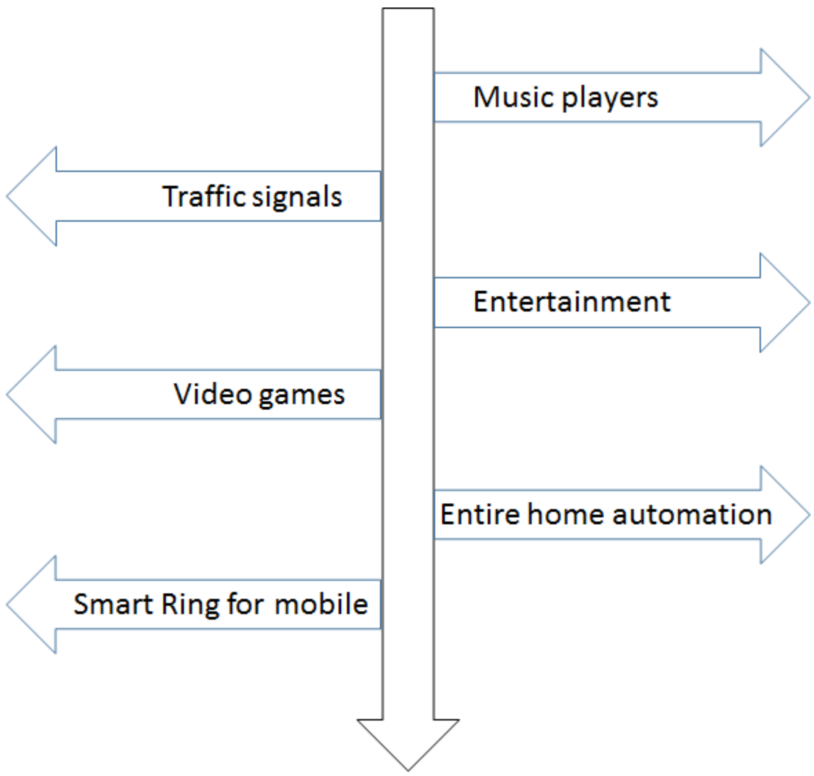


Fig. 8.2-Future Road Map

This prototype can be extended for many areas .The project is quite feasible and can be further extended to entire home automation, as an aid for physically challenged people and make them self-reliant, in gaming applications, in sports events for assisting the referees by automating their hand gestures, as an aid for traffic police and many more. It can be extended in automations like using moving without the physical contact with it, by doing gesture calling, messaging, playing music etc can be controlled, While driving cars music systems can be controlled by gestures. It can be taken in home automations like bulb, fan, tv or in opening and closing doors. The prototype can be converted to a commercial product.

REFERENCES

[1]Akinyemi,L.A., Shoewu,O.O., Makanjuola,N.T., Ajasa,A.A. and Folorunso,C.O. 2014. Design and Development of an Automated Home Control System Using Mobile Phone. World Journal Control Science and Engineering, V.2 (1), pp. 6-11

[2]Mahmud Khaled, Joarder Mahbubul Alam and Roy Anal Kanti. 2015,Voice Activated Electronic Devices Control System For Home Appliances. Scholars Journal of Engineering and Technology (SJET) V.3(1B),pp.66-70.

[3]Mitra, S. 2007 Systems, Man, and Cybernetics, Part C: Applications and reviews; IEEE Transactions, V.37(3), pp.311-324.

[4]Prabhu, Srikanth and Banerjee, R N and Ray, A k and Giridhar, N S and Subbareddy, N V and Prema, K V(2005) Gesture recognition using HMM. In : National conference on devices, intelligent systems and communications, MIT,Manipal.

[5]Xia Liu, 2004. Hand gesture recognition using depth data. pp.529-534. In: Automatic face and hand gesture recognition. Proceedings, 6th IEEE International conference.

[6]Jiayang Liu, Lin Zhong, Jehan Wickramosuriya and Venu Vasudevan. 2009. uWave: Accelerometer based personalized gesture recognition and its applications. Pervasive and mobile computing, V.5(6) pp.657-675.

[7]Jiahui Wu,Gang pan Daging Zhang. GuandeOi, Shijan Li. Gesture recognition with a 3D accelerometer.Lecture nodes in Computer Science. V.5585,pp.25-38

[8] Ahmad Akl “A Novel Accelerometer-based Gesture Recognition System” An unpublished thesis submitted to University of Toronto.