DESIGN AND IMPLEMENTATION OF DRIVER MONITORING SYSTEM USING RASPBERRY PI

A Major Project Report submitted

In partial fulfilment of the requirement

For the award of degree of

BACHELOR OF TECHNOLOGY

In

Electronics and Communication Engineering

By

MEENAL MALIK & PUNEET BANSAL

(03914802814) (07714802814)

Under the Supervision of

Dr. Navneet Yadav

MAHARAJA AGRASEN INSTITUTE OF TECHNOLOGY

SECTOR-22, Rohini, New Delhi

Affiliated to

Guru Gobind Singh Indraprastha University, Delhi

April, 2018

DECLARATION

We, hereby declare that the project report entitled, **Driver Monitoring System**, submitted by us to Maharaja Agrasen Institute of Technology, in partial fulfilment of the requirement for the award of the degree of **Bachelors of Technology** in **Electronics and Communication Engineering** is a record of bonafide project work carried out under the guidance of **Dr. Navneet Yadav**. We further declare that the work reported in the project has not been submitted and will not be submitted for the partial fulfilment of any degree or diploma in this institute or any other institute or university.

Date: April 6th 2018

Meenal Malik	Puneet Bansal
(03914802814)	(07714802814)

CERTIFICATE

This is to certify that the project work entitled, **Driver Monitoring System,** is the bonafide work of **Puneet Bansal (07714802814) and Meenal Malik (03914802814)** submitted in partial fulfilment for the award of degree of **Bachelors of Technology** in **Electronics and Communication Engineering** at Maharaja Agrasen Institute of Technology, New Delhi.

The project work comprises of original work and has not been submitted anywhere else for any other degree to the best of my knowledge.

Date: April 6th 2018

Dr. Navneet Yadav	Dr. Sunil Mathur
(Project Supervisor)	(HOD, ECE)

ACKNOWLEDGEMENT

We owe the deepest gratitude to **Dr. Navneet Yadav**, our supervisor on the project, **Driver Monitoring System,** for his guidance and constant support without which this project would not have materialized.

His profound experience, incomparable expertise along with a kind and supportive nature has been a substantial asset for us throughout the learning experience and also boosts our morale to face new challenges.

Inspite of his hectic schedule and overwhelming onus of responsibilities, he always extended full attention, patient hearing, inspirational guidance which in some way or the other have largely contributed to the conceptualization of the form and context of the work and we will remain ever grateful to him.

Puneet Bansal
(07714802814)

Meenal Malik
(03914802814)

TABLE OF CONTENTS

ABSTRACT

Drowsiness and drunken driving causes the road accidents. This paper proposes a real time detection of driver's drowsiness as well as alcohol intoxication and subsequently alerting them. The main aim of this proposed system is to reduce the number of accidents due to driver's Drowsiness and alcohol intake to increase the transportation safety. This proposed system contains 8-megapixels digital USB camera, Raspberry-pi loaded with Raspbian-OS, Alcohol sensor (MQ-3) is used to detect the intake of alcohol in percentage. Driver face monitoring system is a real-time system that can detect driver fatigue and distraction using machine vision approaches. In this, a new approach is introduced for driver drowsiness detection based on the symptoms related to face and eye regions. In this method, face template matching and horizontal projection of top-half segment of face image are used to extract drowsiness symptoms from face and eye, respectively. The extracted symptoms from eye region are (1) percentage of eye closure, (2) eyelid distance changes with respect to the normal eyelid distance, and (3) eye closure rate. These symptoms related to eye region are used for fatigue detection. In the proposed system, a fuzzy expert system combines the symptoms to estimate level of driver drowsiness. There are three main contributions in the introduced method: (1) simple and efficient drowsiness and alcohol intoxication detection based on face template matching and data from MQ3 respectively, (2) adaptive symptom extraction from eye region without explicit eye detection, and (3) normalizing and personalizing the extracted symptoms using a short training phase. These three contributions lead to develop an adaptive driver eye/face monitoring. Experiments show that the proposed system is relatively efficient for estimating the driver drowsiness and alcohol consumption percentage.

KEYWORDS: Drowsiness detection, Alcohol intoxication, Raspberry pi, OpenCV.

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Improvement of public safety and the reduction of accidents are amongst the most important goals of the Intelligent Transportation Systems. One of the most important factors in accidents, especially on rural roads, is the driver fatigue and monotony. Fatigue reduces driver perceptions and decision making capability to control the vehicle. Researches show that usually the driver is fatigued after 1 hour of driving. In the afternoon early hours, after eating lunch and at midnight, driver fatigue and drowsiness is much more than other times. In addition, drinking alcohol, drug addiction, and using hypnotic medicines can lead to loss of consciousness. The first challenge is how to define fatigue exactly and how to measure it. Despite the progress of science in physiology and psychology, there is still no precise definition for fatigue. Certainly, due to the lack of precise definition of fatigue, there is not any measurable criterion or tool. Drowsiness is a multidimensional feature that researchers over the past decade have found difficult to define. Indeed, it is one of the leading contributing factors in traffic accidents worldwide. Solving the problem became critical when the design of earlier accident prevention systems was found ineffective for alerting the driver. Therefore, **a real-time fatigue detection system** is essential in order to eliminate or reduce the risk of a driver having an accident. However, a precise definition for fatigue is not defined yet, but there is a relationship between fatigue and some symptoms including body temperature, electrical resistance of skin, eye movement, breathing rate, heart rate, and brain activity. One of the first and most important symptoms of fatigue appears in the eye. There is a very close relationship between Psychomotor Vigilance Task (PVT) and the percentage of eyelid closure over time (PERCLOS). PVT shows the response speed of a person to a visual stimulation. Therefore, almost in all driver face monitoring systems, eye closure detection is the first symptom used to measure fatigue.

Eye is the most important area of the face where the symptoms of fatigue and distraction appears. Therefore, many of the driver face monitoring systems detect driver fatigue only based on the symptoms extracted from the eyes. The symptoms related to eye region include PERCLOS, eyelid distance, eye blink speed, eye blink rate, and gaze direction.

It is observed that the driver is determined to be fatigued only if the eyes are closed for several consecutive frames within a specific time period; otherwise, the driver is blinking his or her eyes, and a diagnosis of fatigue would be false. Lee et al. proposed a system that uses two fixed cameras to capture images of the driver and the road, respectively. The images are then mapped to global coordinates to monitor the driver's sight line. These systems require extra cameras to be installed to capture driver facial images. Our proposed system utilizes haar cascades method to detect the face of the driver and then facial landmark detection to perform eye detection task. Lin et al. proposed a real-time brain-computer interface (BCI) system to monitor human physiological and cognitive states by analyzing EEG signals. They demonstrated that the amplitude of an EEG peak value, which is estimated by a drowsiness detection system, may predict a driving error. In others, the ECG electrodes are placed at the driver seat and measurements are carried out when the driver's back side is in contact with the electrodes, but ECG signals can't be measured if the driver seat cloth is too thick as the contact distance between body and electrodes is increased. Meanwhile, Yang et al. integrated the sensors in a wearable shirt to measure ECG signals. Most drivers are not willing to wear such shirts during driving as it may feel uncomfortable and most importantly the shirt is difficult to clean and wash. Our proposed system avoids such limitations as it can be mounted anywhere in front of the driver in order to detect his/her face, integrating the circuit in a small box with a webcam placed in front of the driver. Liang et al. developed an in-vehicle system to detect driver distraction that applies SVMs, trained on all the data collected in simulator experiments. Three factors were investigated: how distraction can be defined; which data should be input to the model; and how the input data should be summarized. The above systems predict the driver vigilance index based on driving behaviour and vehicle movement instead of driver actual condition in real-time. The alertness prediction is more accurate by analyzing on the driver facial features and biomedical signals. The focus and objective of this study was to develop a reliable, well-controlled and non-intrusive drowsiness monitoring system that comprises the following aspects:

- Detecting the driver drowsiness(using EAR)
- Detecting whether the driver is drunk or not(using MQ3)
- Updating the status on a cloud platform
- Updating the status of the driver on an android app and representing it with the help of graphs.

CHAPTER 2

COMPONENTS AND MODULES

This chapter describes the different modules and components being used in the hardware, along with their specifications and other necessary information in respect to the project.

1. ADS 1115

The ADS1113, ADS1114, and ADS1115 devices (ADS111x) are precision, low-power, 16-bit, I^2C-compatible, analog-to-digital converters (ADCs) offered in an ultra-small, leadless, X2QFN-10 package, and a VSSOP-10 package. The ADS111x devices incorporate a low-drift voltage reference and an oscillator. The ADS1114 and ADS1115 also incorporate a programmable gain amplifier (PGA) and a digital comparator. These features, along with a wide operating supply range, make the ADS111x well suited for power- and space-constrained, sensor measurement applications.

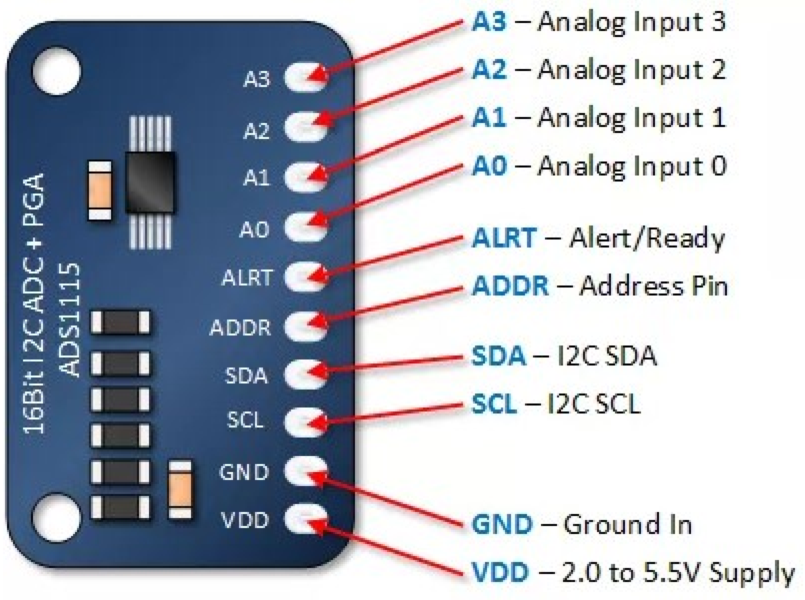


Fig.2.1. ADS1115 Pin Diagram

The ADS111x perform conversions at data rates up to 860 samples per second (SPS). The PGA offers input ranges from ±256 mV to ±6.144 V, allowing precise large- and small-signal measurements. The ADS1115 features an input multiplexer (MUX) that allows two differential or four single-ended input measurements. Use the digital comparator in the ADS1114 and ADS1115 for under- and overvoltage detection.

The ADS111x operate in either continuous-conversion mode or single-shot mode. The devices are automatically powered down after one conversion in single-shot mode; therefore, power consumption is significantly reduced during idle periods.

Features

- Ultra-Small X2QFN Package: 2 mm × 1.5 mm × 0.4 mm
- Wide Supply Range: 2.0 V to 5.5 V
- Low Current Consumption: 150 µA (Continuous-Conversion Mode)
- Programmable Data Rate: 8 SPS to 860 SPS
- Single-Cycle Settling
- Internal Low-Drift Voltage Reference
- Internal Oscillator
- I^2C Interface: Four Pin-Selectable Addresses
- Four Single-Ended or Two Differential Inputs (ADS1115)
- Programmable Comparator (ADS1114 and ADS1115)
- Operating Temperature Range: –40°C to +125°C

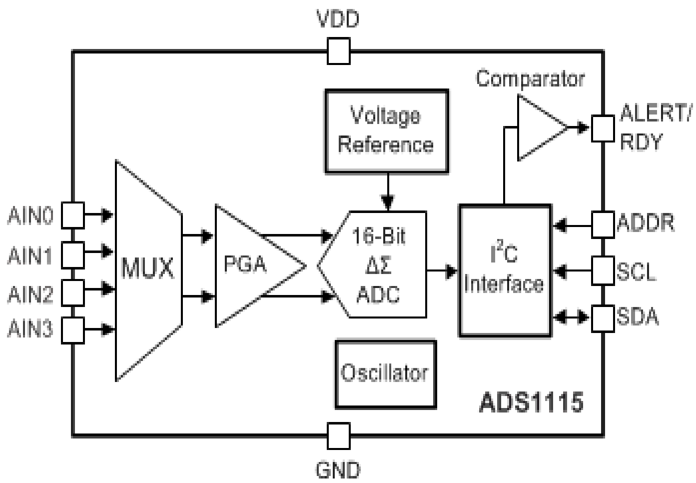


Fig.2.2. Functional Diagram of ADS1115

2. Raspberry Pi

It is a credit card-sized computer which was originally designed for education, inspired by the 1981 BBC Micro. Due to its small size, it was quickly adopted by tinkerers, makers, and electronics enthusiasts for projects that require more than a basic microcontroller (such as Arduino devices). The Raspberry Pi is a complete Linux computer and can provide all the expected abilities that implies with effective results, at a low power consumption level. Here, we have employed Raspberry Pi3 model.

Fig.2.3. Raspberry Pi 3

3. WebCam

A webcam – short for ‘web camera’ – is a digital camera that’s connected to a computer. It can send live pictures from wherever it’s sited to another location by means of the internet. Many desktop computer screens and laptops come with a built-in camera and microphone, but if yours doesn’t, you can add a separate webcam at any time.

Fig.2.4. WebCam

There are various types. Some are plugged into computers through USB ports, but others are wireless (Wi-Fi). Other features might include:

- an integral microphone
- the ability to pan and tilt
- in-built sensors that can detect movement and start recording
- a light that, when on, will let you know that the camera is in use.

There’s a wide range of things that you can do with a webcam. The most common is to video chat over the internet using Skype – see our Skype guides for all the information you need to get going.

4. Alcohol Sensor MQ3:

This module is made using Alcohol Gas Sensor MQ3. It is a low cost semiconductor sensor which can detect the presence of alcohol gases at concentrations from 0.05 mg/L to 10 mg/L. The sensitive material used for this sensor is SnO_2, whose conductivity is lower in clean air. It’s conductivity increases as the concentration of alcohol gases increases. It has high sensitivity to alcohol and has a good resistance to disturbances due to smoke, vapor and gasoline. This module provides both digital and analog outputs. MQ3 alcohol sensor module can be easily interfaced with Microcontrollers, Arduino Boards, Raspberry Pi etc.

This alcohol sensor is suitable for detecting alcohol concentration in your breath, just like your common breathalyzer. It has a high sensitivity and fast response time.

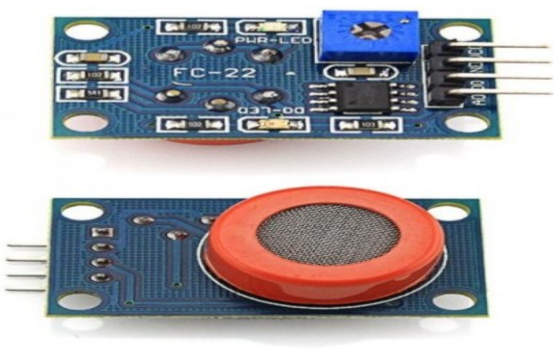


Fig.2.5. MQ3 sensor module

Sensor provides an analog resistive output based on alcohol concentration. The drive circuit is very simple, all it needs is one resistor. A simple interface could be a 0-3.3V ADC.

Features

- 5V operation
- Simple to use
- LEDs for output and power
- Output sensitivity adjustable
- Analog output 0V to 5V
- Digital output 0V or 5V
- Low Cost
- Fast Response
- Stable and Long Life
- Good Sensitivity to Alcohol Gas
- Both Digital and Analog Outputs
- On-board LED Indicator

Technical Data

- Concentration : 0.05 mg/L ~ 10 mg/L Alcohol
- Operating Voltage : 5V ±0.1
- Current Consumption : 150mA
- Operation Temperature : -10°C ~ 70°C

5. Buzzer

A buzzer is a mechanical, electromechanical, magnetic, electromagnetic, electro-acoustic or piezoelectric audio signalling device. A piezo electric buzzer can be driven by an oscillating electronic circuit or other audio signal source. A click, beep or ring can indicate that a button has been pressed.

Fig.2.6. Buzzer

CHAPTER 3

FACE DETECTION USING HAAR CASCADES

Object Detection using Haar feature-based cascade classifiers is an effective object detection method proposed by Paul Viola and Michael Jones in their paper, "Rapid Object Detection using a Boosted Cascade of Simple Features" in 2001. It is a machine learning based approach where a cascade function is trained from a lot of positive and negative images. It is then used to detect objects in other images. Here, we have used it for face detection. Initially, the algorithm needs a lot of positive images (images of faces) and negative images (images without faces) to train the classifier. Then we need to extract features from it. For this, haar features shown in below image are used. They are just like our convolutional kernel. Each feature is a single value obtained by subtracting sum of pixels under white rectangle from sum of pixels under black rectangle.

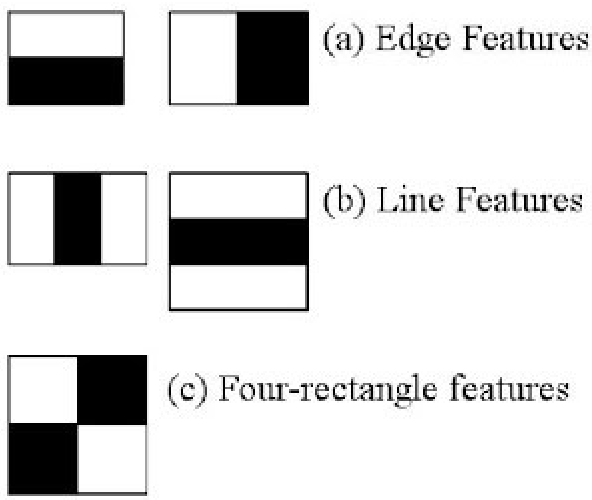


Fig.3.1. Haar Features

Now all possible sizes and locations of each kernel is used to calculate plenty of features. For each feature calculation, we need to find sum of pixels under white and black rectangles. To solve this, they introduced the integral images. It simplifies calculation of sum of pixels, how large may be the number of pixels, to an operation involving just four pixels. It makes things super-fast.

But among all these features calculated, most of them are irrelevant. For example, consider the image below. Top row shows two good features. The first feature selected seems to focus on the property that the region of the eyes is often darker than the region of the nose and cheeks. The second feature selected relies on the property that the eyes are darker than the bridge of the nose. But the same windows applying on cheeks or any other place is irrelevant. The best features out of 160000+ features are selected with the help of **Adaboost**.

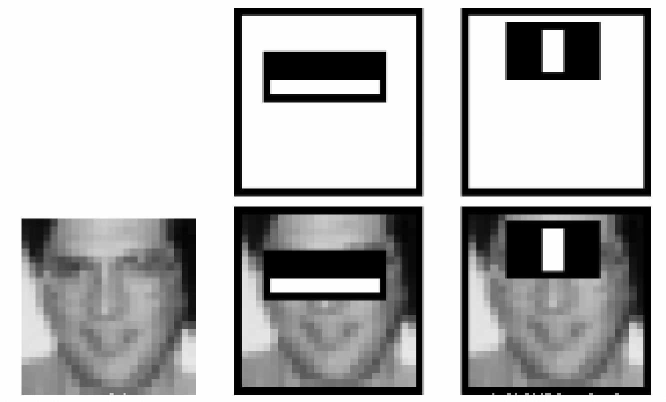


Fig.3.2. Feature Extraction

For this, we apply each and every feature on all the training images. For each feature, it finds the best threshold which will classify the faces to positive and negative. But obviously, there will be errors or misclassifications. We select the features with minimum error rate, which means they are the features that best classifies the face and non-face images. (The process is not as simple as this. Each image is given an equal weight in the beginning. After each classification, weights of misclassified images are increased. Then again same process is done. New error rates are calculated. Also, new weights. The process is continued until required accuracy or error rate is achieved or required number of features are found).

Final classifier is a weighted sum of these weak classifiers. It is called weak because it alone can't classify the image, but together with others forms a strong classifier. The paper says even 200 features provide detection with 95% accuracy. Their final setup had around 6000 features. .

In an image, most of the image region is non-face region. So it is a better idea to have a simple method to check if a window is not a face region. If it is not, discard it in a single shot.

Don't process it again. Instead focus on region where there can be a face. This way, we can find more time to check a possible face region.

For this they introduced the concept of **Cascade of Classifiers**. Instead of applying all the 6000 features on a window, group the features into different stages of classifiers and apply one-by-one. (Normally first few stages will contain very less number of features). If a window fails the first stage, discard it. We don't consider remaining features on it. If it passes, apply the second stage of features and continue the process. The window which passes all stages is a face region.

OpenCV comes with a trainer as well as detector. If you want to train your own classifier for any object like car, planes etc. you can use OpenCV to create one.

FACIAL LANDMARK DETECTION

The facial landmark detector included in the dlib library is an implementation of the *One Millisecond Face Alignment with an Ensemble of Regression Trees* paper by Kazemi and Sullivan (2014).
This method starts by using:

1. A training set of labeled facial landmarks on an image. These images are *manually labeled*, specifying **specific** *(x, y)*-coordinates of regions surrounding each facial structure.

2. *Priors*, of more specifically, the *probability on distance* between pairs of input pixels.

Given this training data, an ensemble of regression trees are trained to estimate the facial landmark positions directly from the *pixel intensities themselves* (i.e., no “feature extraction” is taking place).
The end result is a facial landmark detector that can be used to **detect facial landmarks in *real-time*** with **high quality predictions**.
The pre-trained facial landmark detector inside the dlib library is used to estimate the location of ***68 (x, y)-coordinates*** that map to facial structures on the face.

The indexes of the 68 coordinates can be visualized on the image below:

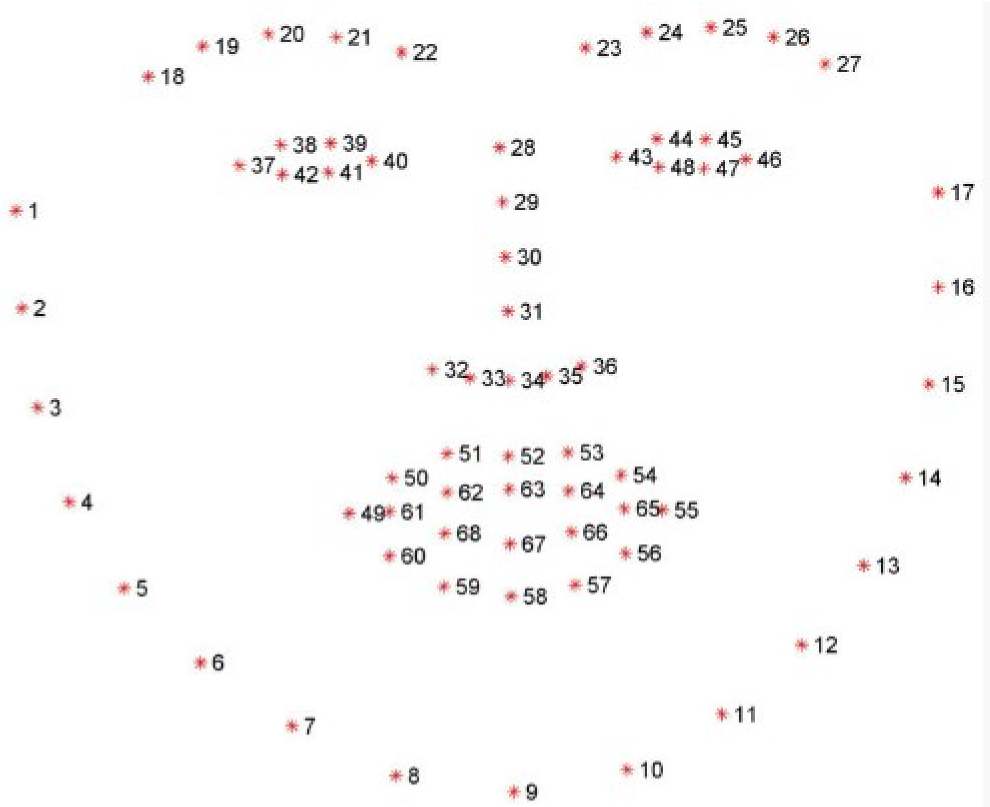


Fig.3.3. Facial Landmark Points

CHAPTER 4

FUNCTIONING AND PRACTICAL IMPLEMENTATION OF THE SYSTEM

The working is divided into 4 tasks:

- Detecting the driver drowsiness(using EAR)
- Detecting whether the driver is drunk or not(using MQ3)
- Updating the status on a cloud platform
- Updating the status of the driver on an android app and representing the same with the help of graphs.

I. Driver drowsiness detection:

First, we'll setup a camera that monitors a stream for faces. If a face is found, we apply facial landmark detection and extract the eye regions. Now that we have the eye regions, we can compute the eye aspect ratio to determine if the eyes are closed.

Now, what is EAR?

In terms of blink detection, we are only interested in two sets of facial structures — *the eyes.* Each eye is represented by 6 *(x, y)*-coordinates, starting at the left-corner of the eye (as if you were looking at the person), and then working clockwise around the remainder of the region:

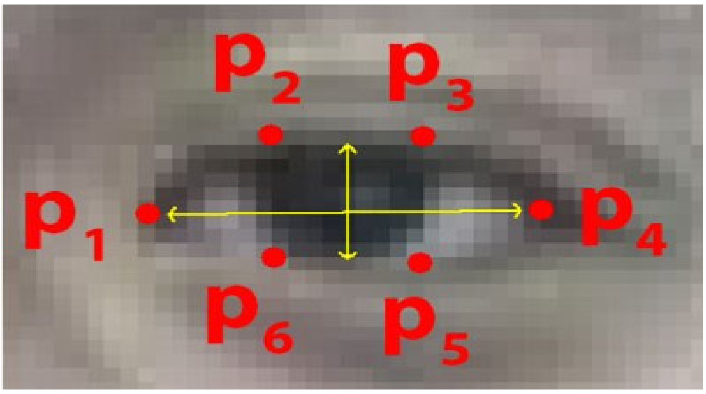


Fig.4.1. Eye Coordinates representation

There is a relation between the *width* and the *height* of these coordinates.

Based on the work by Soukupová and Čech in their 2016 paper, *Real-Time Eye Blink Detection using Facial Landmarks*, we can then derive an equation that reflects this relation called the *eye aspect ratio* (EAR):

$$\text{EAR} = \frac{\|p_2 - p_6\| + \|p_3 - p_5\|}{2\|p_1 - p_4\|}$$

Where *p1, …, p6* are 2D facial landmark locations.

The numerator of this equation computes the distance between the vertical eye landmarks while the denominator computes the distance between horizontal eye landmarks, weighting the denominator appropriately since there is only *one* set of horizontal points but *two* sets of vertical points.

Why is this equation so interesting?

Well, as we'll find out, the eye aspect ratio is approximately constant while the eye is open, but will rapidly fall to zero when a blink is taking place.

Using this simple equation, we can *avoid image processing techniques* and simply rely on the *ratio of eye landmark distances* to determine if a person is blinking.

To make this more clear, consider the following figure from Soukupová and Čech:

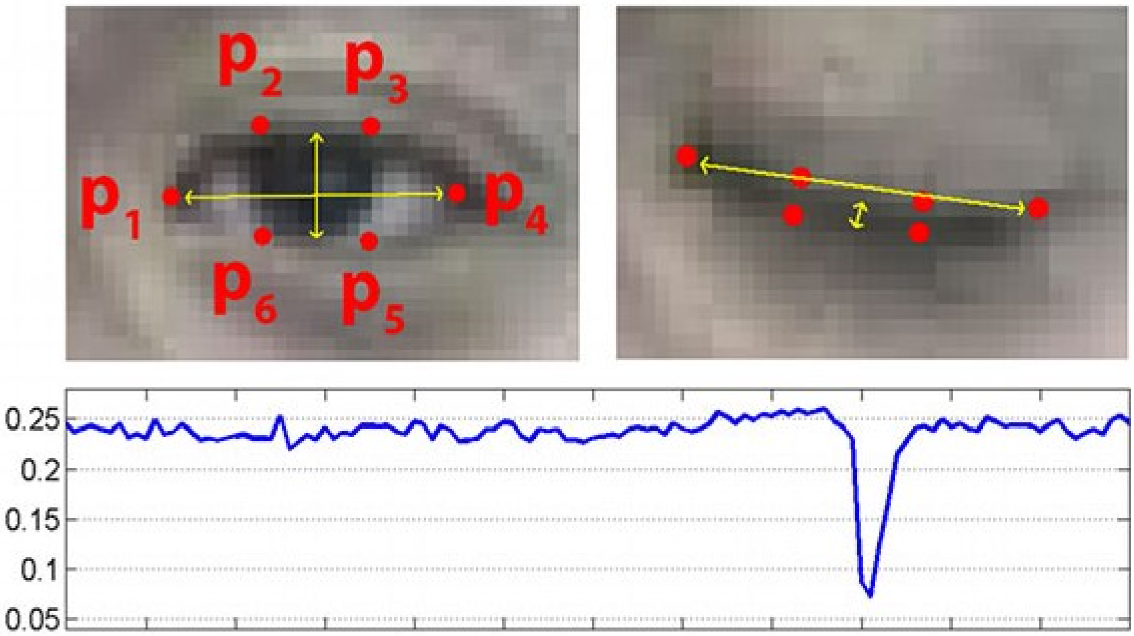


Fig.4.2. EAR graphical analysis

On the *top-left* we have an eye that is fully open — the eye aspect ratio here would be large(r) and relatively constant over time.

However, once the person blinks (*top-right*) the eye aspect ratio decreases dramatically, approaching zero.

The *bottom* figure plots a graph of the eye aspect ratio over time for a video clip. As we can see, the eye aspect ratio is constant, then rapidly drops close to zero, then increases again, indicating a single blink has taken place.

To summarize, the first step in building a blink detector is to perform facial landmark detection to *localize the eyes* in a given frame from a video stream.

Once we have the facial landmarks for both eyes, we compute the *eye aspect ratio* for each eye, which gives us a singular value, relating the distances between the vertical eye landmark points to the distances between the horizontal landmark points.

Once we have the eye aspect ratio, we can threshold it to determine if a person is blinking — **the eye aspect ratio will remain approximately constant when the eyes are open and then will rapidly approach zero during a blink, then increase again as the eye opens.**

To improve our blink detector, Soukupová and Čech recommend constructing a 13-dim feature vector of eye aspect ratios (*N*-th frame, *N – 6* frames, and *N + 6* frames), followed by feeding this feature vector into a Linear SVM for classification.

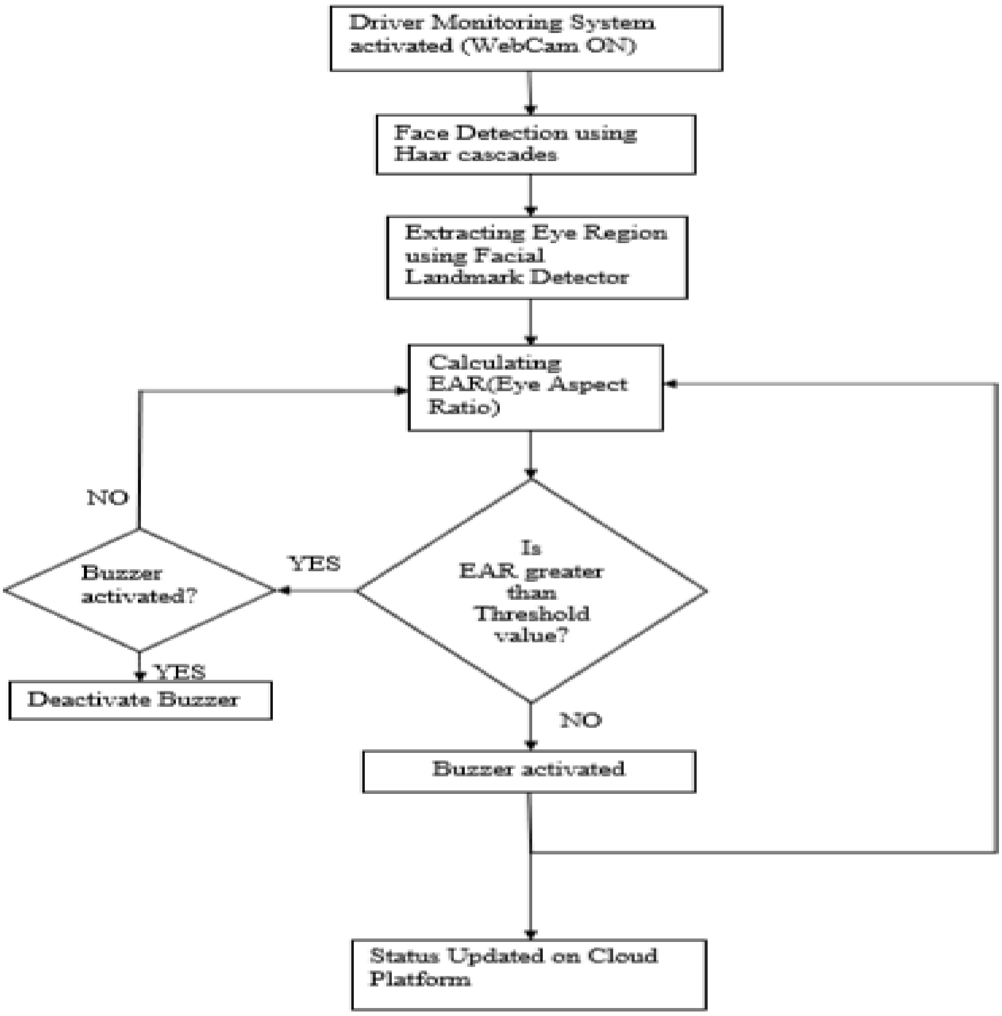


Fig.4.3. Flowchart for drowsiness detection

Table 4.1. EAR interpretation

EAR>0.3	Not Drowsy	Buzzer Deactivated	Attentive Driver
EAR<0.3	Drowsiness Detected	Buzzer Activated	Not Attentive Driver
0.15<EAR<0.3	Drowsiness Detected	Buzzer Activated	Moderately Attentive Driver
EAR<0.15	Drowsiness Detected	Buzzer Activated	Less Attentive Driver

II. Driver alcohol consumption check using MQ3 sensor:

In the other hand to detect the Alcoholic Intoxication of the driver, an alcohol gas sensor (MQ3) work as a breathalyzer to calculate alcohol intake of a driver. At the time of testing of MQ-3 Alcohol sensor there are different voltage level/samples at output. The sensor continuously vary in the range of 0-1023, because there is 10 bit ADC output. In this system of the alcoholic Intoxication is displayed in the percentage, for that purpose we program as per our condition that voltage samples is converted into percentage using Mapping Concept. If the input of alcohol is 0% to 60% at that time output is in the form of a buzzer Indication to the Driver. If the input of alcohol is Above 70% at that time Buzzer will be activated through Raspberry Pi and the status will be updated on the android app. The system begins to operate when the alcohol sensor detected BAC level from the driver. It will send the signal to raspberry pi for further process which will involve the LCD display and buzzing the alarm. BAC level detected by alcohol sensor is based on gas or alcohol concentration in ppm (parts per million). This system is tested with the help of whisky / alcoholic drinks / deodorant/ after shave lotion as the input to the experiment.

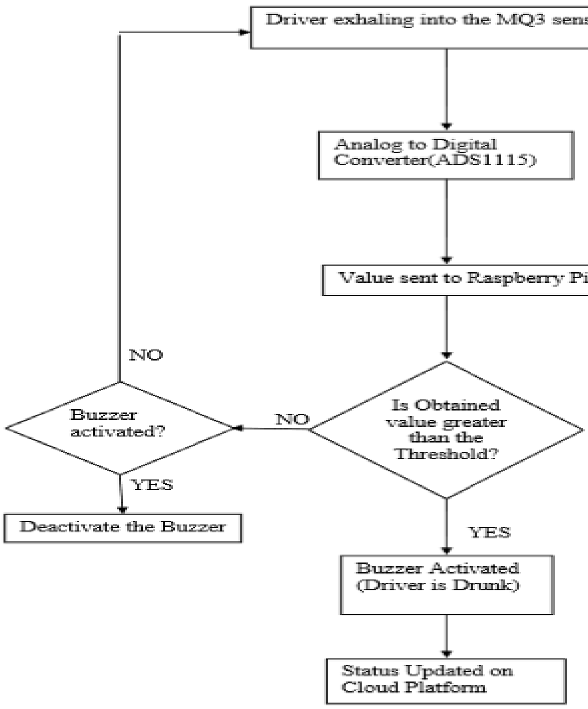


Fig.4.4. Flowchart for Alcohol Detection

Table 4.2. Alcohol Consumption Interpretation

Obtained value<Threshold(15%)	Not drunk	Buzzer Deactivated	Non-Alcoholic driver
Obtained value>Threshold(15%)	Drunk	Buzzer Activated	Alcoholic Driver

The alcohol sensor can sense an alcohol from human breath from 0 ppm until 1000 ppm. In this system the alcoholic Intoxication is displayed in the percentage, for that purpose we program as per our condition that voltage samples is converted into percentage using Mapping Concept. The result is categorized into four conditions of the driver with different value(in percentage) of BAC level which are intoxication, slightly drunk ,drunkenness and over limited drunk. For first condition, LCD will display “intoxication” when the alcohol sensor detected BAC level from 0% to 20%.This condition shows that the driver is free from alcohol and no alarm sound from the buzzer is activated. The next condition is slightly drunk with 21% to 40% drunkenness level. This means the driver is drunk with a little amount of alcohol, but need to be assisted during the driving on the road. The buzzer is activated to alert the driver and the people in vicinity that the driver is slightly drunk. The third condition drunkenness level with 41% to 60%. This means the driver is drunk with a little strong amount of alcohol, but need to be assisted during the driving on the road. The buzzer is activated to alert the driver and the people in vicinity that the driver is slightly drunk. In the last condition, the level of BAC is too high. The driver is totally unconscious and not safe for driving. Ultimately, this system helps to prevent the driver to drive in risky situation and will avoid accident occurrence on the road.

III. Updating the output on cloud platform

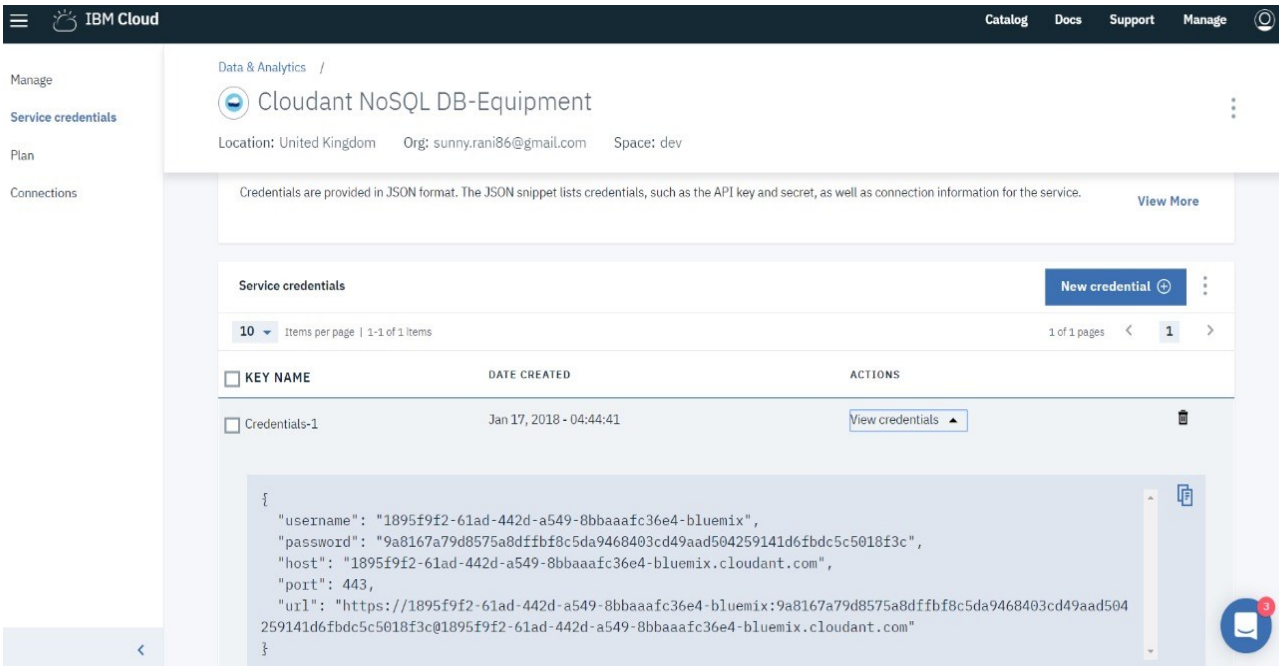


Fig.4.5. Cloud platform

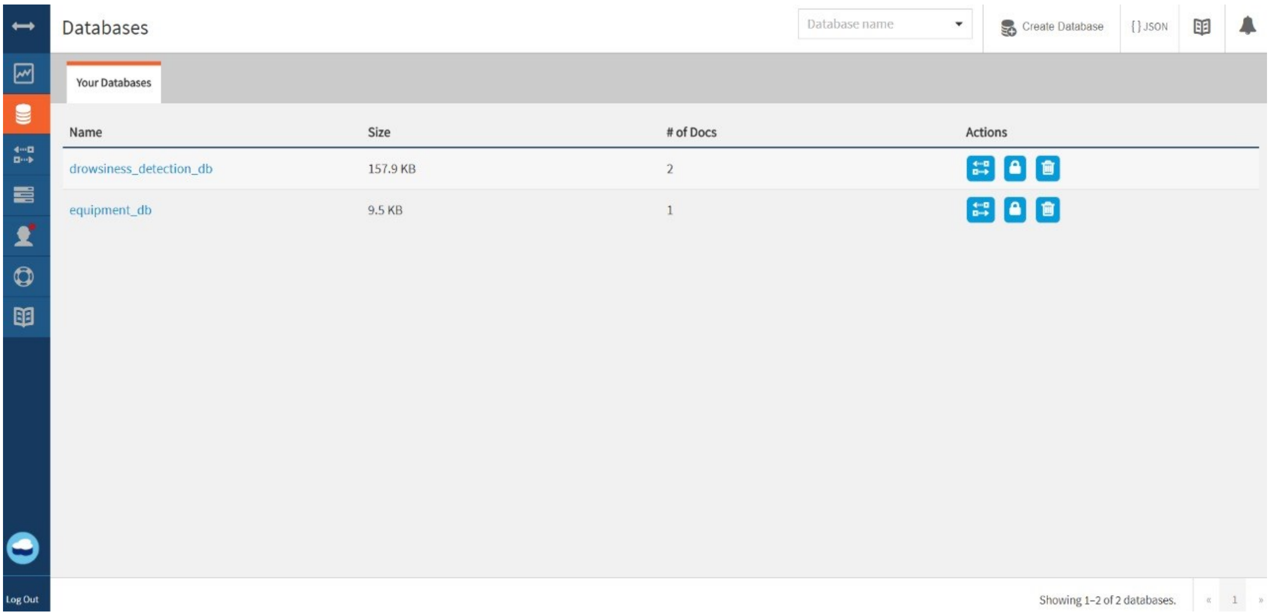


Fig.4.6. Cloud Database

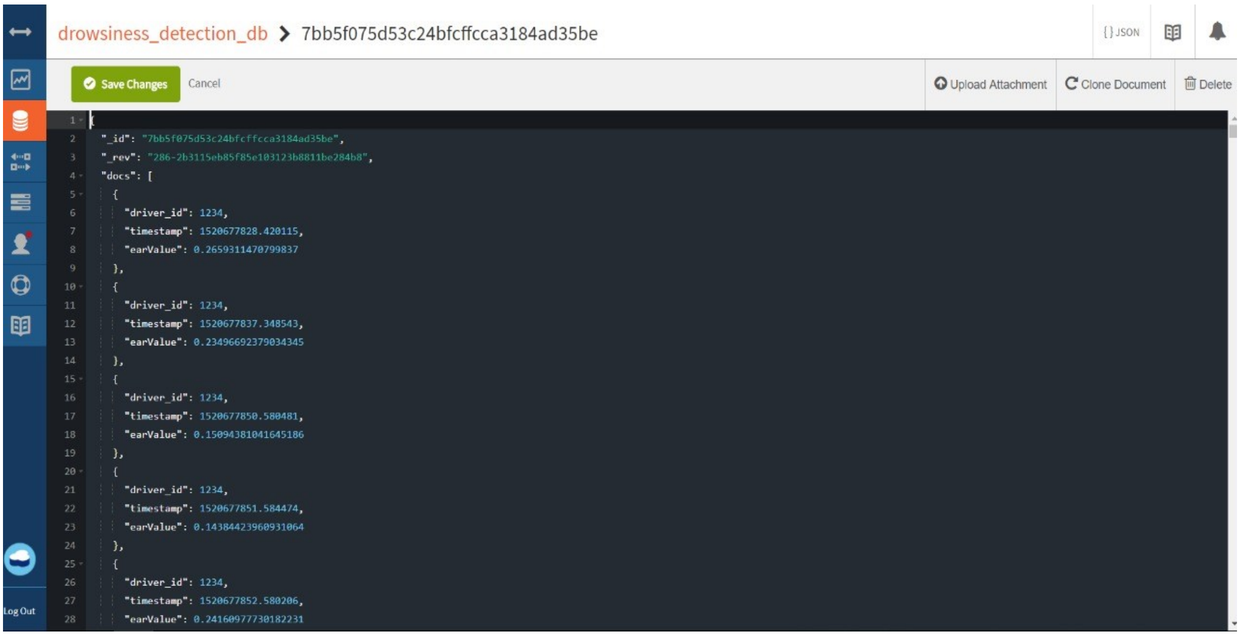


Fig.4.7. EAR Data update

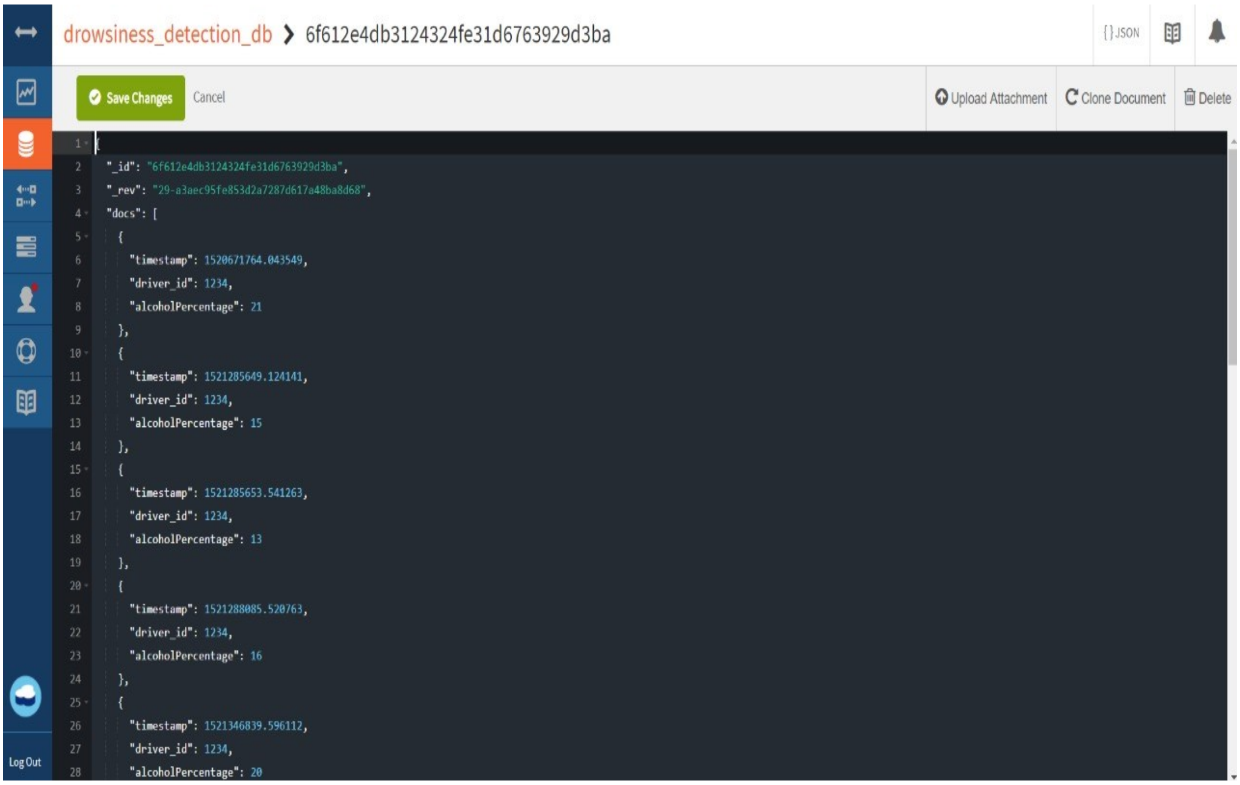


Fig.4.8. Alcohol percentage data update

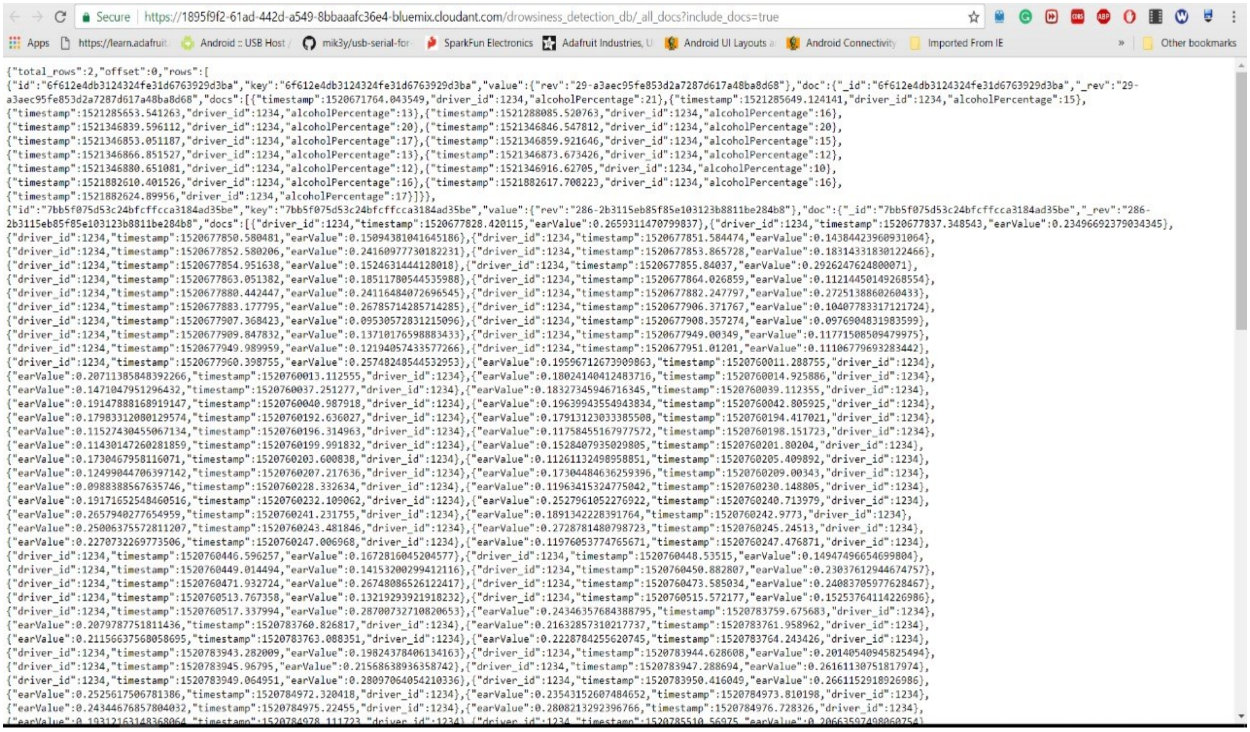


Fig.4.9. Whole database on cloud platform

IV. Procedure to implement Graphical Representation on Android app

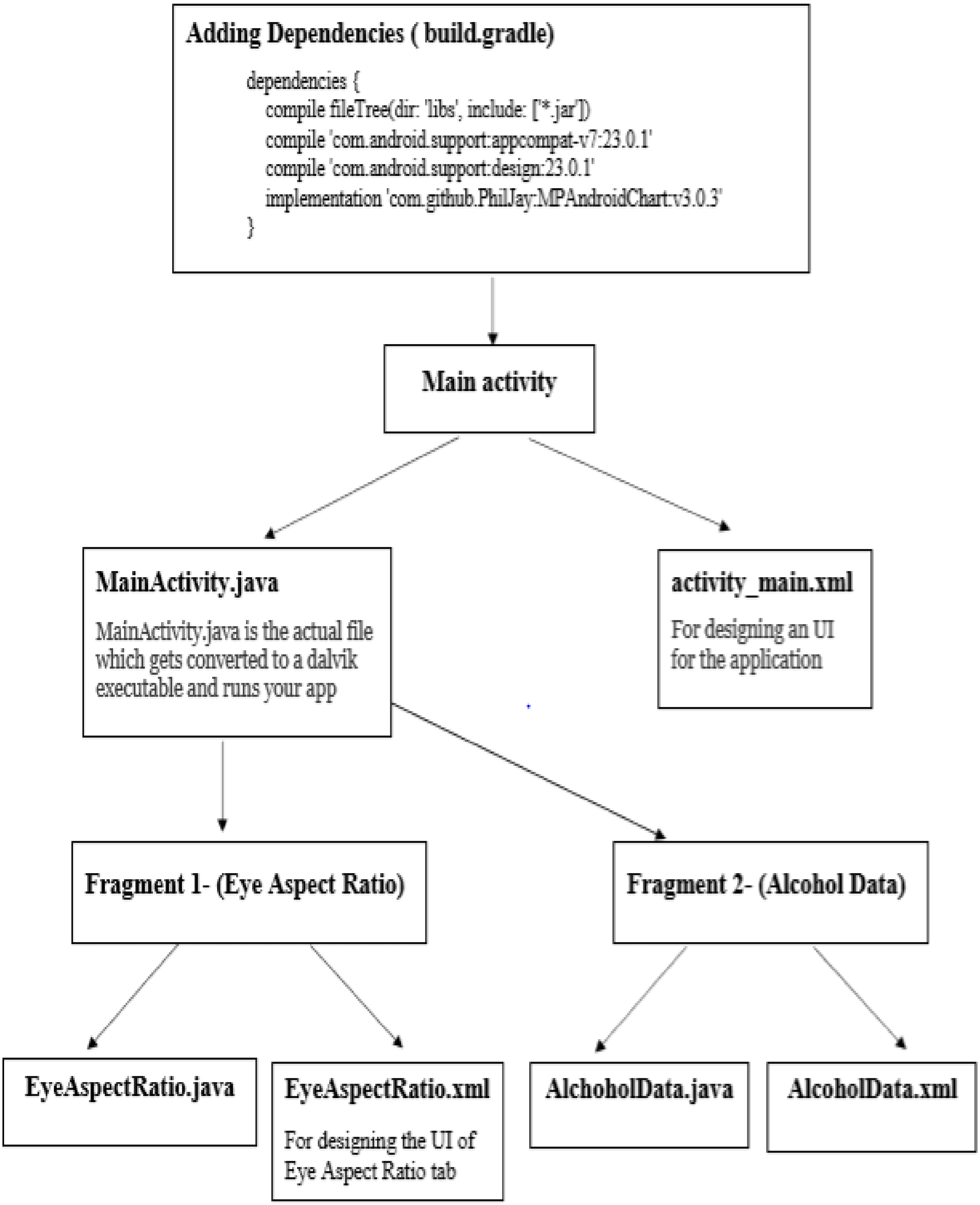


Fig.4.10. Flowchart for creation of android app

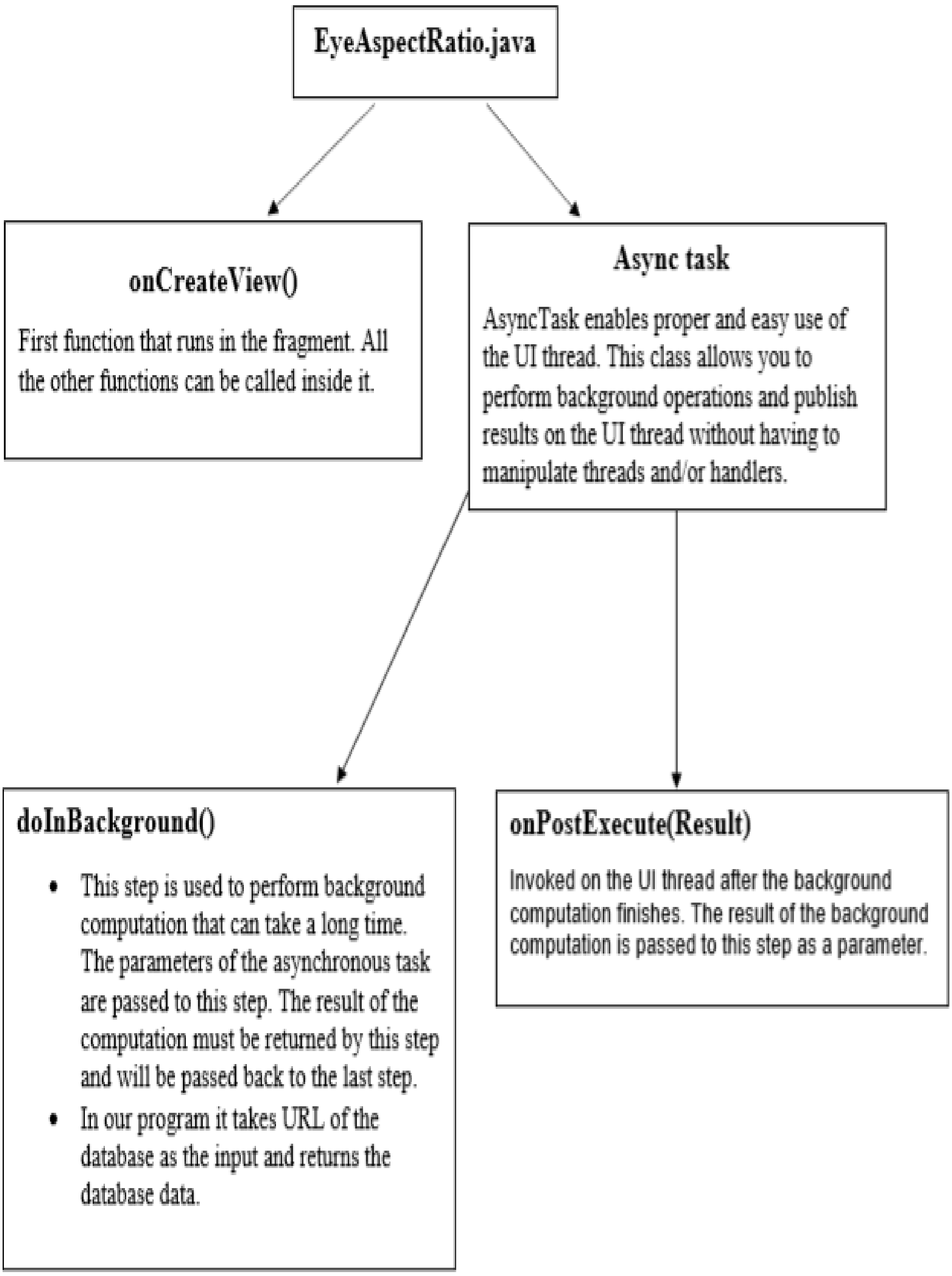


Fig.4.11. Flowchart for creation of android app (Contd.)

CHAPTER 5

RESULTS AND OUTPUT

1. Project Set-up

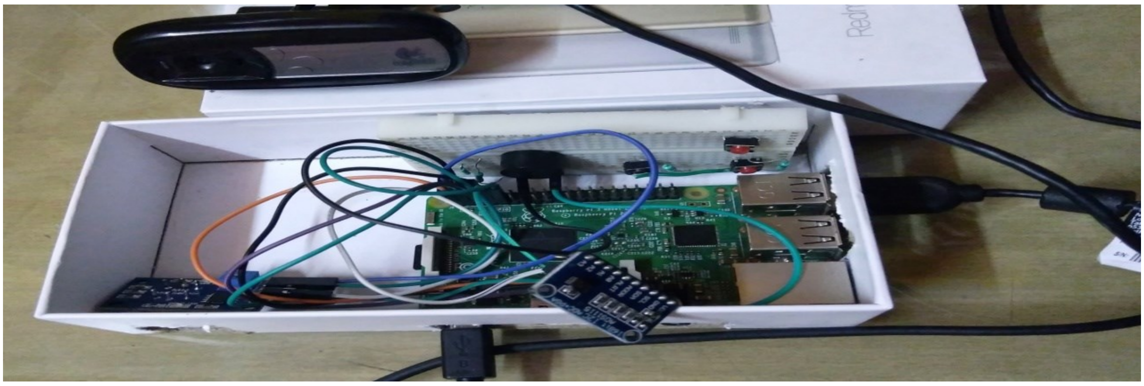


Fig.5.1. Hardware set-up

2. Eyes open(EAR value above threshold)

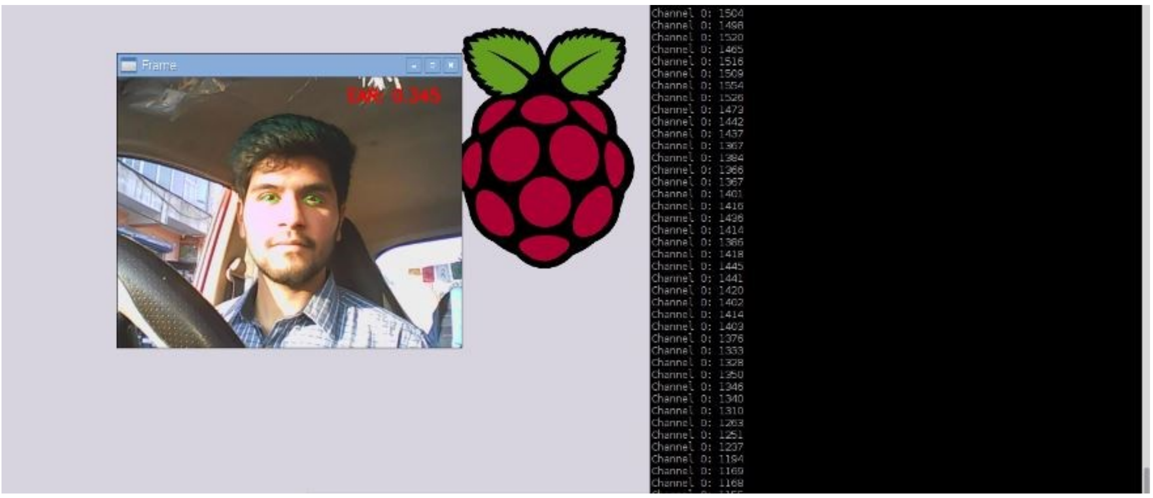


Fig.5.2. Buzzer deactivated

3. Eyes closed(EAR value below threshold)

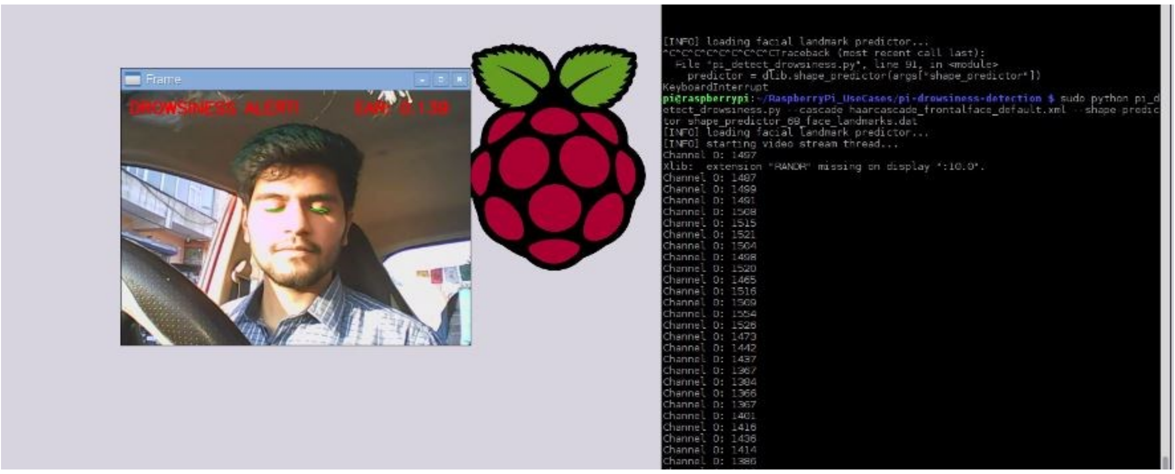


Fig.5.3. Buzzer activated

4. Moderately attentive driver

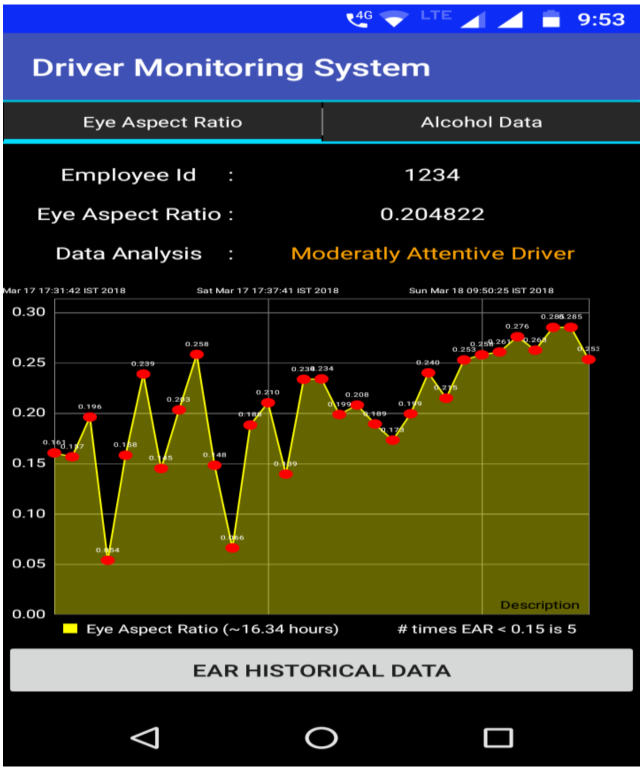


Fig.5.4. Graphical representation on android app of moderately attentive driver

5. Less attentive driver

Fig.5.5. Graphical representation of less attentive driver on android app

6. Complete EAR data analysis

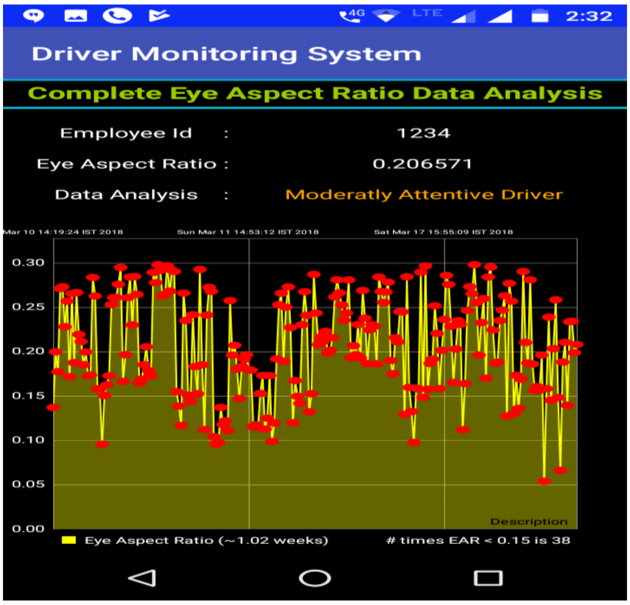


Fig.5.6. Historical graph on android app

7. Complete EAR data analysis(Zoomed in)

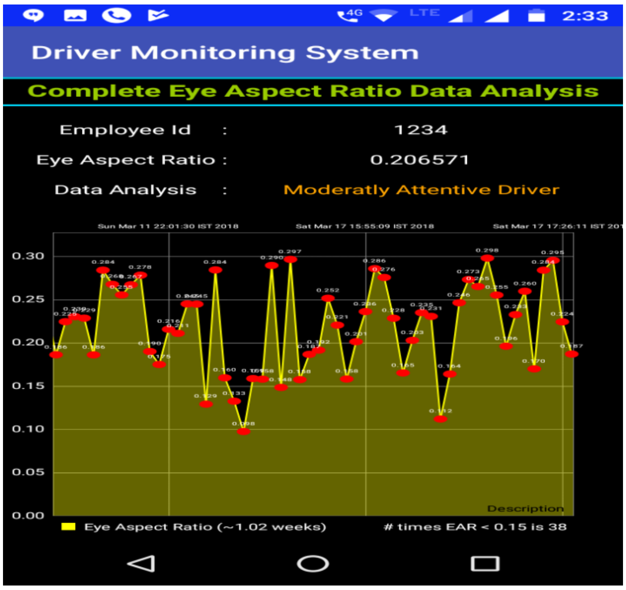


Fig.5.7. Zoomed view of historical graph on android app

8. Alcohol detection analysis

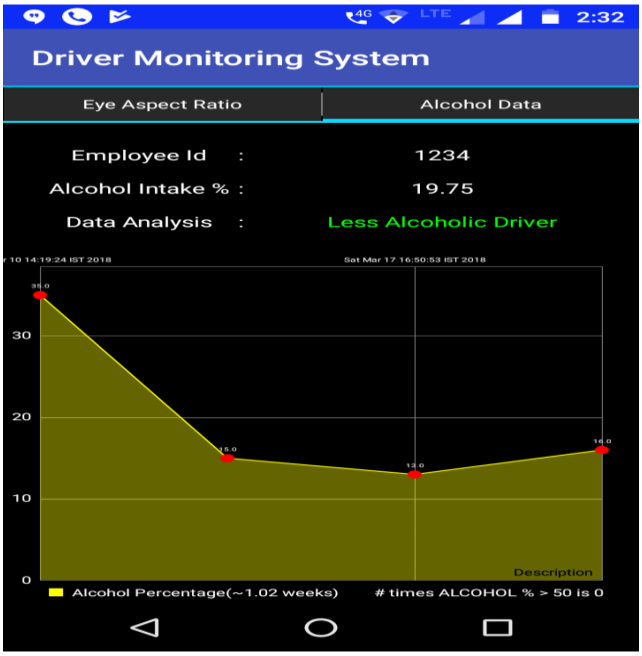


Fig.5.8. Graphical representation of less alcoholic driver on android app

9. Alcohol detection analysis

Fig.5.9. Graphical representation of less alcoholic driver on android app

REFERENCES

1. Dipjoy Sarkar and Atanu Chowdhury, “A Real Time Embedded System Application for Driver Drowsiness and Alcoholic Intoxication Detection”, International Journal of Engineering Trends and Technology (IJETT), Volume 10, Number 9, pp. 461-465,2014

2. Pandurang N. Kathar, Prof. D.L.Bhuyar, “Design and Implementation of Driver Drowsiness and Alcohol Intoxication Detection Using Raspberry PI” International Journal of Innovative Research in Computer and Communication Engineering, Vol. 4, Issue 8, August 2016

3. Varsha.E.dahiphale and Prof.sathyanarayanaR,”Computer Vision System for Driver Fatigue Detection”, International Journal of Advanced Rsearch in Electronics and communication Engineering(IJARECE), volume 04,Issue-9, pp-2331-2334,2015

4. Saeid.fazli and Parisa .Esfehani,”Tracking Eye State for Fatigue Detection”, International Conference on advanced in Computer and Electrical Engineering (ICACEE’2012), 2012

5. Tereza Soukupova and Jan Cech ‘Real-Time Eye Blink Detection using Facial Landmarks’

6. DhavalPimplaskar,Dr.M.S.Nagmode,AtulBorkar,”Real Time Eye Blinking Detection and Tracking Using OpenCV”,International Journal of Engineering Research and Application,Vol.03,Issue 05, pp-1780-1787,2013

7. V.V. Dixit, A.V. Deshpande, and D. Ganage, Face Detection for Drivers’ Drowsiness Using Computer Vision

8. SrinivasuBatchu and S. Praveen Kumar, “Driver Drowsiness Detection to Reduce the Major Road Accidents in Automotive Vehicles”, International Research Journal of Engineering and Technology (IRJET), Volume 02, Issue 01, pp. 345-349,2015

9. Mohamad-Hoseyn Sigari, Mahmood Fathy, and Mohsen Soryani ‘A Driver Face Monitoring System for Fatigue and Distraction Detection’

10. Prof.Saurabh Thakur, ShekharDure, AjinkyaKhutwad,”Advance Vehicle Control and Safety System Using Face Detection”, International Journal of Advance Research In Science And Engineering (IJARSE),Vol.No.4,Issue 05,pp.141-146,2015

APPENDIX – A

MQ3 Sensor working and theoretical analysis

MQ-3 Semiconductor Sensor for Alcohol Sensitive material of MQ-3 gas sensor is SnO_2, which with lower conductivity in clean air. When the target alcohol gas exist, The sensor's conductivity is more higher along with the gas concentration rising. Please use simple electrocircuit, Convert change of conductivity to correspond output signal of gas concentration. MQ-3 gas sensor has high sensitity to Alcohol, and has good resistance to disturb of gasoline, smoke and vapor. The sensor could be used to detect alcohol with different concentration, it is with low cost and suitable for different application. Character Configuration * Good sensitivity to alcohol gas * Long life and low cost * Simple drive circuit Application * Vehicel alcohol detector * Portable alcohol detector Technical Data Basic test loop The above is basic test circuit of the sensor.

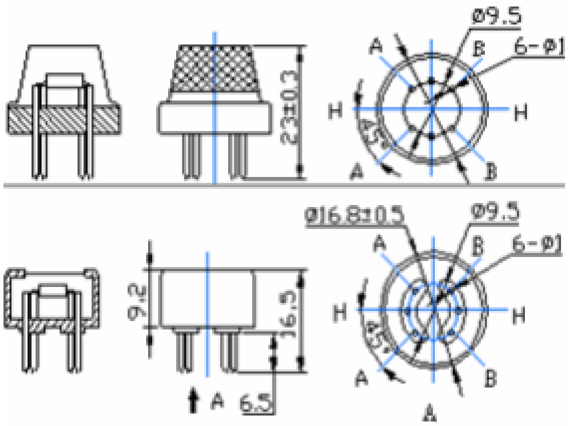


Fig.A.1. MQ3 infrastructure

The sensor need to be put 2 voltage, heater voltage（VH） and test voltage（VC）. VH used to supply certified working temperature to the sensor, while VC used to detect voltage (VRL) on load resistance （RL）whom is in series with sensor.

The sensor has light polarity, Vc need DC power. VC and VH could use same power circuit with precondition to assure performance of sensor. In order to make the sensor with better performance, suitable RL value is needed: Power of Sensitivity body(Ps): $Ps=Vc^2 \times Rs/(Rs+RL)^2$ Model No. MQ-3 Sensor Type Semiconductor Standard Encapsulation Bakelite (Black Bakelite) Detection Gas Alcohol gas Concentration 0.04-4mg/l alcohol Loop Voltage Vc ≤24V DC Heater Voltage VH 5.0V±0.2V AC or DC Circuit Load Resistance RL Adjustable Heater Resistance RH 31Ω±3Ω（Room Tem.） Heater consumption PH ≤900mW Sensing Resistance Rs 2KΩ-20KΩ(in 0.4mg/l alcohol) Sensitivity S Rs(in air)/Rs(0.4mg/L Alcohol)≥5 Character Slope α ≤0.6(R300ppm/R100ppm Alcohol) Tem. Humidity 20℃±2℃；65%±5%RH Standard test circuit Vc:5.0V±0.1V； VH: 5.0V±0.1V Condition Preheat time Over 48 hours Vc VH GND RL VRL Resistance of sensor(Rs):

Table A.1. MQ3 features

Model No.			MQ-3
Sensor Type			Semiconductor
Standard Encapsulation			Bakelite (Black Bakelite)
Detection Gas			Alcohol gas
Concentration			0.04-4mg/l alcohol
Circuit	Loop Voltage	V_c	≤24V DC
	Heater Voltage	V_H	5.0V±0.2V AC or DC
	Load Resistance	R_L	Adjustable
Character	Heater Resistance	R_H	31Ω±3Ω（Room Tem.）
	Heater consumption	P_H	≤900mW
	Sensing Resistance	R_s	2KΩ-20KΩ(in 0.4mg/l alcohol)
	Sensitivity	S	Rs(in air)/Rs(0.4mg/L Alcohol)≥5
	Slope	α	≤0.6(R_{300ppm}/R_{100ppm} Alcohol)
Condition	Tem. Humidity		20℃±2℃；65%±5%RH
	Standard test circuit		Vc:5.0V±0.1V； V_H: 5.0V±0.1V
	Preheat time		Over 48 hours

Rs=(Vc/VRL-1)×RL Sensitivity Characteristics Influence of Temperature/Humidity Fig.1 shows the typical sensitivity characteristics of Fig.2 shows the typical temperature and humidity the MQ-3, ordinate means resistance ratio of the sensor characteristics. Ordinate means resistance ratio (Rs/Ro), abscissa is concentration of gases. Rs means of the sensor (Rs/Ro), Rs means resistance of sensor resistance in different gases, Ro means resistance of in 0.4mg/l alcohol under different tem. and humidity. sensor in 0. 4mg/l alcohol. All test are under standard Ro means resistance of the sensor in environment of test conditions. 0.4mg/l alcohol, 20°C/65%RH P.S.: Sensitivity to smoke is ignite 10pcs cigarettes in 8m3 room, and the output equals to 0.1mg/l alcohol Structure and configuration Structure and configuration of MQ-3 gas sensor is shown as Fig. 3, sensor composed by micro AL2O3 ceramic tube, Tin Dioxide (SnO2) sensitive layer, measuring electrode and heater are fixed into a crust made by plastic and stainless steel net. The heater provides necessary work conditions for work of sensitive components.

Sensitivity Characteristics **Influence of Temperature/Humidity**

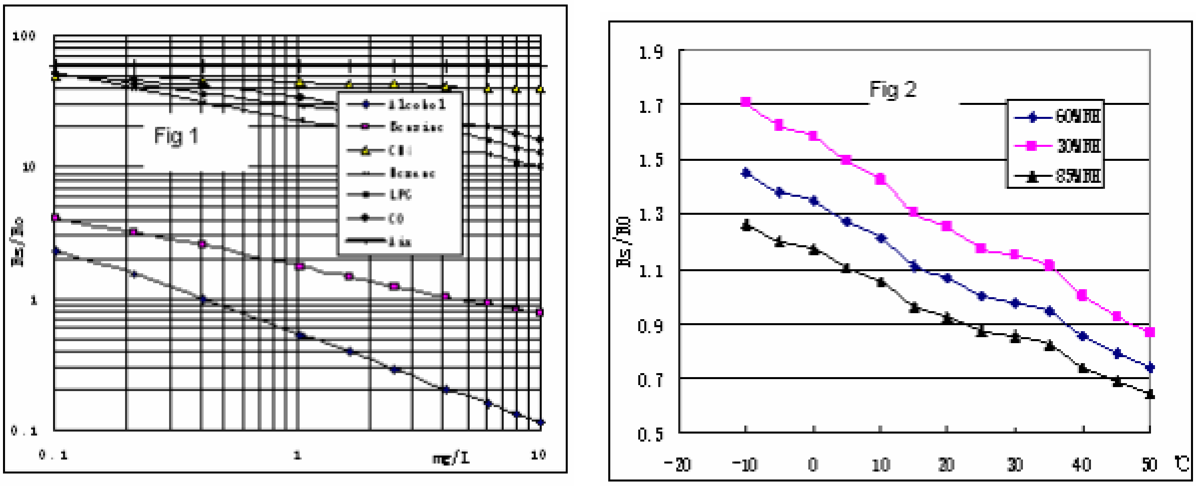


Fig.A.2. Functional Characteristics of MQ3

The enveloped MQ-4 have 6 pin, 4 of them are used to fetch signals, and other 2 are used for providing heating current. Notification 1 Following conditions must be prohibited 1.1 Exposed to organic silicon steam Organic silicon steam cause sensors invalid, sensors must be avoid exposing to silicon bond, fixature, silicon latex, putty or plastic contain silicon environment 1.2 High Corrosive gas If the sensors exposed to high concentration corrosive gas (such as H_2Sz, SOX, Cl_2, HCl etc), it will not only result in corrosion of sensors structure, also it cause sincere sensitivity attenuation. 1.3 Alkali, Alkali metals salt, halogen pollution The sensors performance will be changed badly if sensors be sprayed polluted by alkali metals salt especially brine, or be exposed to halogen such as fluorin. 1.4 Touch water Sensitivity of the sensors will be reduced when spattered or dipped in water. 1.5 Freezing Do avoid icing on sensor'surface, otherwise sensor would lose sensitivity. 1.6 Applied voltage higher Applied voltage on sensor should not be higher than stipulated value, otherwise it cause down-line or heater damaged, and bring on sensors' sensitivity characteristic changed badly. 1.7 Voltage on wrong pins For 6 pins sensor, if apply voltage on 1、3 pins or 4、6 pins, it will make lead broken, and without signal when apply on 2、4 pins 2 Following conditions must be avoided 2.1 Water Condensation Indoor conditions, slight water condensation will effect sensors performance lightly. However, if water condensation on sensors surface and keep a certain period, sensor' sensitivity will be decreased. 2.2 Used in high gas concentration No matter the sensor is electrified or not, if long time placed in high gas concentration, if will affect sensors characteristic. 2.3 Long time storage The sensors resistance produce reversible drift if it's stored for long time without electrify, this drift is related with storage conditions.

Structure and configuration

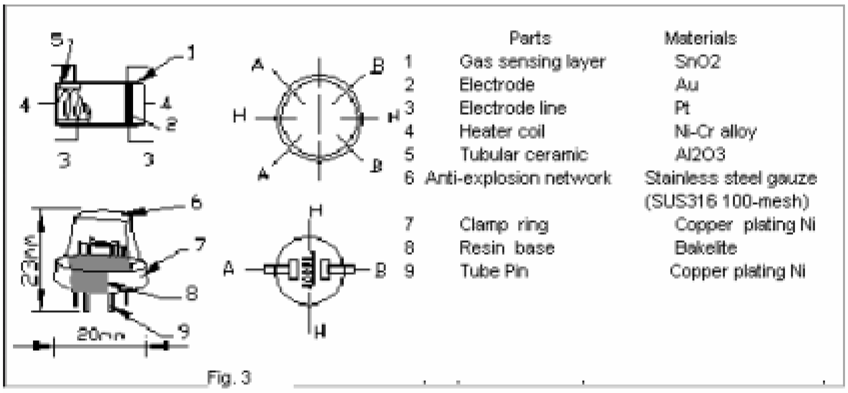


Fig.A.3. MQ3 Structure and Configuration

Sensors should be stored in airproof without silicon gel bag with clean air. For the sensors with long time storage but no electrify, they need long aging time for stbility before using. 2.4 Long time exposed to adverse environment No matter the sensors electrified or not, if exposed to adverse environment for long time, such as high humidity, high temperature, or high pollution etc, it will effect the sensors performance badly. 2.5 Vibration Continual vibration will result in sensors down-lead response then repture. In transportation or assembling line, pneumatic screwdriver/ultrasonic welding machine can lead this vibration. 2.6 Concussion If sensors meet strong concussion, it may lead its lead wire disconnected. 2.7 Usage For sensor, handmade welding is optimal way. If use wave crest welding should meet the following conditions: 2.7.1 Soldering flux: Rosin soldering flux contains least chlorine 2.7.2 Speed: 1-2 Meter/ Minute 2.7.3 Warm-up temperature : 100±20℃ 2.7.4 Welding temperature : 250±10℃ 2.7.5 1 time pass wave crest welding machine If disobey the above using terms, sensors sensitivity will be reduced.

APPENDIX – B

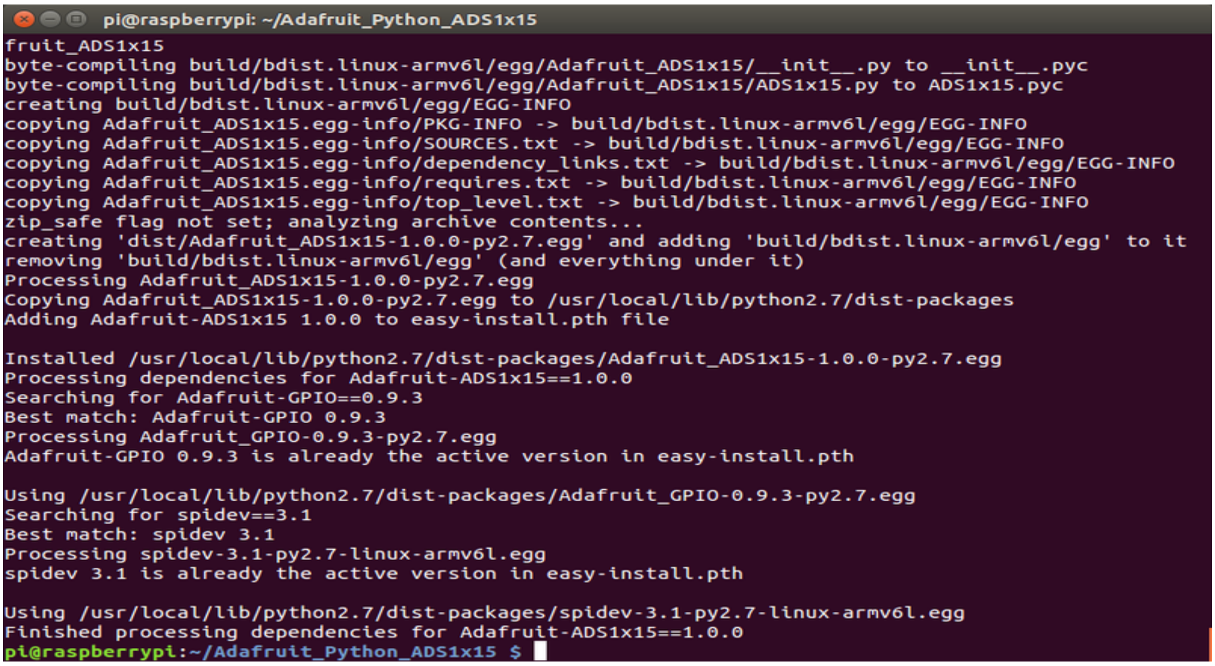
Installation procedure of Adafruit ADS1115 python library on Raspberry Pi

After wiring the ADS1115 to the Raspberry Pi, install the **Adafruit ADS1x15 Python library**.
Install the library from the Python package index with a few commands, or install the library from its source on GitHub.
Note that before installing the library Raspberry Pi must be connected to the internet through a wired or wireless network connection.

Source Install
To install from the source on Github connect to a terminal on the Raspberry Pi and run the following commands:

1. sudo apt-get update
2. sudo apt-get install build-essential python-dev python-smbus git
3. cd ~
4. git clone https://github.com/adafruit/Adafruit_Python_ADS1x15.git
5. cd Adafruit_Python_ADS1x15
6. sudo python setup.py install

One should see the library install succeed and finish with a message similar to the following:



```
pi@raspberrypi: ~/Adafruit_Python_ADS1x15
fruit_ADS1x15
byte-compiling build/bdist.linux-armv6l/egg/Adafruit_ADS1x15/__init__.py to __init__.pyc
byte-compiling build/bdist.linux-armv6l/egg/Adafruit_ADS1x15/ADS1x15.py to ADS1x15.pyc
creating build/bdist.linux-armv6l/egg/EGG-INFO
copying Adafruit_ADS1x15.egg-info/PKG-INFO -> build/bdist.linux-armv6l/egg/EGG-INFO
copying Adafruit_ADS1x15.egg-info/SOURCES.txt -> build/bdist.linux-armv6l/egg/EGG-INFO
copying Adafruit_ADS1x15.egg-info/dependency_links.txt -> build/bdist.linux-armv6l/egg/EGG-INFO
copying Adafruit_ADS1x15.egg-info/requires.txt -> build/bdist.linux-armv6l/egg/EGG-INFO
copying Adafruit_ADS1x15.egg-info/top_level.txt -> build/bdist.linux-armv6l/egg/EGG-INFO
zip_safe flag not set; analyzing archive contents...
creating 'dist/Adafruit_ADS1x15-1.0.0-py2.7.egg' and adding 'build/bdist.linux-armv6l/egg' to it
removing 'build/bdist.linux-armv6l/egg' (and everything under it)
Processing Adafruit_ADS1x15-1.0.0-py2.7.egg
Copying Adafruit_ADS1x15-1.0.0-py2.7.egg to /usr/local/lib/python2.7/dist-packages
Adding Adafruit-ADS1x15 1.0.0 to easy-install.pth file

Installed /usr/local/lib/python2.7/dist-packages/Adafruit_ADS1x15-1.0.0-py2.7.egg
Processing dependencies for Adafruit-ADS1x15==1.0.0
Searching for Adafruit-GPIO==0.9.3
Best match: Adafruit-GPIO 0.9.3
Processing Adafruit_GPIO-0.9.3-py2.7.egg
Adafruit-GPIO 0.9.3 is already the active version in easy-install.pth

Using /usr/local/lib/python2.7/dist-packages/Adafruit_GPIO-0.9.3-py2.7.egg
Searching for spidev==3.1
Best match: spidev 3.1
Processing spidev-3.1-py2.7-linux-armv6l.egg
spidev 3.1 is already the active version in easy-install.pth

Using /usr/local/lib/python2.7/dist-packages/spidev-3.1-py2.7-linux-armv6l.egg
Finished processing dependencies for Adafruit-ADS1x15==1.0.0
pi@raspberrypi:~/Adafruit_Python_ADS1x15 $
```

Fig.B.1. Installation procedure

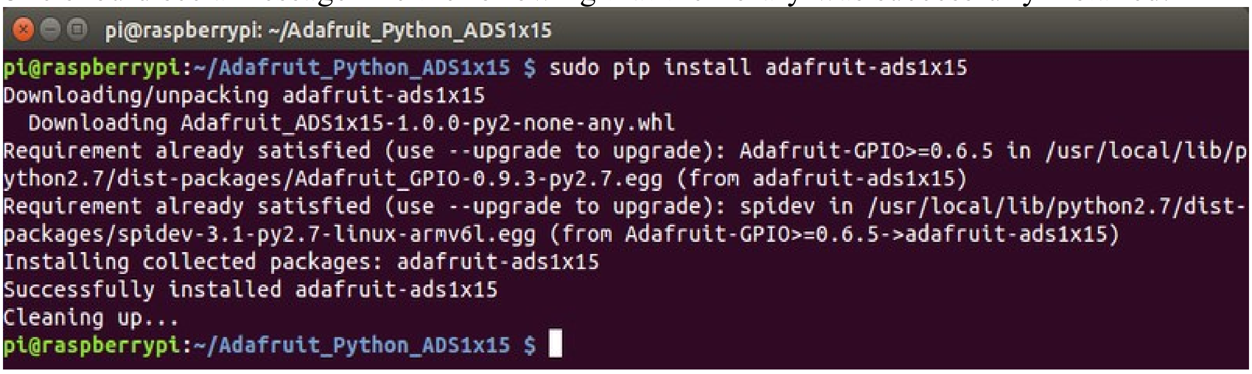
If an error occurs go back and carefully check all the previous commands were run, and that they didn't fail with an error.

Python Package Index Install

To install from the Python package index connect to a terminal on the Raspberry Pi and execute the following commands:

1. sudo apt-get update
2. sudo apt-get install build-essential python-dev python-smbus python-pip
3. sudo pip install adafruit-ads1x15

One should see a message like the following that the library was successfully installed:



```
pi@raspberrypi: ~/Adafruit_Python_ADS1x15
pi@raspberrypi:~/Adafruit_Python_ADS1x15 $ sudo pip install adafruit-ads1x15
Downloading/unpacking adafruit-ads1x15
  Downloading Adafruit_ADS1x15-1.0.0-py2-none-any.whl
Requirement already satisfied (use --upgrade to upgrade): Adafruit-GPIO>=0.6.5 in /usr/local/lib/python2.7/dist-packages/Adafruit_GPIO-0.9.3-py2.7.egg (from adafruit-ads1x15)
Requirement already satisfied (use --upgrade to upgrade): spidev in /usr/local/lib/python2.7/dist-packages/spidev-3.1-py2.7-linux-armv6l.egg (from Adafruit-GPIO>=0.6.5->adafruit-ads1x15)
Installing collected packages: adafruit-ads1x15
Successfully installed adafruit-ads1x15
Cleaning up...
pi@raspberrypi:~/Adafruit_Python_ADS1x15 $
```

Fig.B.2. Installation procedure(Contd.)

Note that on installing from the Python package index there **won't** be the example code for the library. Download these ADS1x15 examples to the Pi manually and run them.