# A Real Time Embedded System Application for Driver Drowsiness and Alcoholic Intoxication Detection

Dwipjoy Sarkar[#1], Atanu Chowdhury[*2]

[#1]M.Tech student, [*2]Assistant professor, Department of Electronics & Communication Engineering
NIT Agartala, India
Tripura, India

*Abstract*— **This paper outlines a novel approach for the real time detection of car driver drowsiness and alcoholic intoxication. There are large numbers of road accidents which takes place due to fatigue or alcohol drinking of driver. Computer vision and alcohol gas sensor application is combined to an embedded system to achieve this goal. The proposed system is realized with an open source 5 megapixel digital camera supported embedded system board Raspberry-pi loaded with Raspbian-OS, and Python-IDLE with Open-CV installed. The Raspberry-pi system board is serially interfaced with another open source embedded system board Arduino Uno with I2C protocol, which will perform some task like issuing the alarm notification and switching off the car power source to stop the car upon receiving the positive detection message from Raspberry-pi.**

*Keywords*— **Drowsiness, alcoholic intoxication, Raspberry pi, Arduino UNO, Open CV, Embedded System, Python IDLE, Haar Cascade classifier, Linux, Raspbian, OV5647, MQ-3, I2C, relay, buzzer.**

## I. INTRODUCTION

Driver fatigue and alcohol drinking is a severe problem which results thousands of road accidents per year. It is difficult to correctly tell the exact number of sleep and alcohol drinking related accidents but traffic survey shows that driver fatigue may be a contributory factor in up to 20% and due to alcohol drinking it is about 31% of all road accidents. This project aims to create one more step towards solving of this serious problem. The paper represents a new design to detect the drowsiness and alcohol intake of car driver and perform necessary actions on real time. The design is based on computer vision and embedded system application principles. For the detection of drowsiness, it detects the eye closing rate of the person. Haar feature based cascade classifier [23] method is used for closed eye detection. On the other hand to detect the alcoholic intoxication of the driver, an alcohol gas sensor method is used which works as a breathalyzer and calculates blood alcohol content (BAC) from breath alcohol content (BrAC). There are lots of research has been performed claiming detection of driver drowsiness [1-10]. [11-15] shows different approaches for alcohol quantity measurements. The work done over here is different from [1-15], and provides a more practical and efficient design with algorithm implementation which can be easily implemented as a prototype. Development software's used in this project are all open source and available for free download. The design also includes two open source embedded application development board Raspberry pi [16] and Arduino Uno [17] to speed up real time processing. Hence the proposed design over here is cost effective and also does not compromise with the quality. Raspberry pi system board is interfaced with OV5647 [22] color CMOS QSXGA 5 megapixel camera, which is capable of capturing real time images and video. The captured frame is to be processed by Raspberry pi. Raspberry pi algorithm is implemented using Python [21] programming language with open source computer vision extension Open CV [18]. Eye close detection is based on Haar cascade classifier technique and performs several comparisons from a database of similar and dissimilar images and returns a rectangle over the detected area on matching. Eye closing rate is calculated after each 5 seconds, and if it crosses a predefined threshold value, then Raspberry pi sends a warning data to its slave device Arduino Uno over I2C serial bus. On receiving the warning message, the Arduino performs a set of tasks like issuing buzzer notification or turning off a relay to stop the car. The Arduino module is separately interfaced with MQ-3 [19] alcohol sensor, KPEG260 buzzer [20] and relay. Meanwhile the Arduino receives any warning data from Raspberry pi to perform some predefined set of task; it continuously checks alcohol content present in the air and computes blood alcohol content (%BAC) from it. If the calculated %BAC crosses the threshold limit, it will again issue same buzzer notification and will turn off the relay.

## II. METHODOLOGY

This section shows the steps towards achieving the objective of detection of drowsiness and alcoholic intoxication. Detection of drowsiness can be done in several ways like remotely measuring the heart rate or facial expression of the person to be tested. This work is the combination of face detection, eye region detection and eye closing rate detection in real time environment. OpenCV is open source software for creating computer vision related task and it is available as an extension for C, C++, Java and Python programming languages. Making a computer vision application in real time is a challenging task and it needs efficient processing power. Raspberry-pi is an ARM11 controller based small sized open source CPU with 512 MB RAM and supports 700 MHz processing speed. It supports interfacing of various low level and high level peripherals including digital camera and GPIO's. It can work with light weight Linux based operating system Raspbian which is loaded with Python-IDLE programming software. OpenCV linux version is installed to Raspberry-pi. Fig. 1 shows the basic block diagram of the proposed system. Haar Feature-based Cascade Classifier technique [23] is used for the detection face region and eye region. It is a learning based approach where a function is trained from lots of similar and dissimilar images. It is then used to detect objects in other images. Open CV is packed with a trainer as well as detector. The trainer is used to create our own classfier for an object. For proper detection 1000 eye closing images has been put as similar images and 1000 dissimilar images are also included. The resulting classifier is stored in a file with .xml extension which is used in the programming environment. On the other hand to detect the alcohol intake bye the person, an alcohol gas sensor or breathalyzer MQ-3 is interfaced with the Arduino system board which will scan whether the person in driving seat is drunken or not. Based on the detection of drowsiness or alcoholic intoxication, an alarm will be turned on or the car the car's power source can be cut down through a relay to stop the car or preventing the driver to start the car.

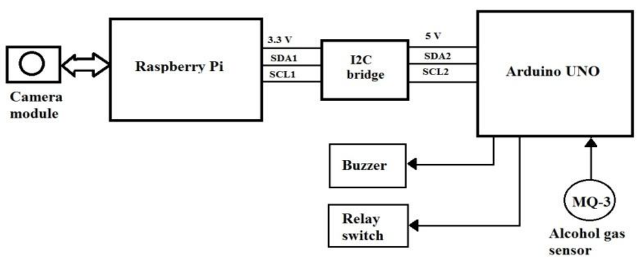


Fig. 1. Block diagram of the proposed system

## III. BLOOD ALCOHOL CONTENT (BAC) CALCULATION

The MQ-3 is an alcohol gas sensor, which measures alcohol content present in the volume of breath in mg/L (milligrams per liter). This is known as breath air content (BrAC). To determine whether a person is drunk or not the proper measure is blood alcohol content (BAC), which is the amount of alcohol content in blood volume. Observing the % BAC value, alcoholic intoxication of a person can be detected. There is a straight forward relationship between BrAC and BAC. BrAC and BAC differ by a factor of 2100; that is, for every mg of alcohol in the breath, there are 2100mg of alcohol in the blood .That means, MQ-3 measured BrAC values can be directly converted to BAC.

$$1\%\ \text{BAC} = 10\text{g/L or } 10{,}000\ \text{mg/L} \quad (1)$$

$$0.1\%\ \text{BAC} = 1000\ \text{mg/L} \quad (2)$$

A person with 0.1% BAC has 1000 mg/L of alcohol in their blood and 1000/2100 = 0.4762 mg alcohol in their breath.
So, the final formula for calculating BAC from BrAC is:

$$\%\ \text{BAC} = \text{BrAC mg/L} * 0.21 \quad (3)$$

## IV. HARDWARE AND SOFTWARE CONFIGURATION

### *A. Software configuration*

The necessary software tools required for the system includes Raspbian operating system, Python IDLE, Open CV computer vision software extension for python with Haar object detection trainer, and Arduino cross platform IDE for uploading sketch to Arduino board .

### *B. Hardware modules*

The proposed design contains, Raspberry pi model B, OV5647 camera module with connector, Arduino UNO R3 development board, MQ-3 alcohol gas sensor, BS170 mosfet, KPEG260 piezoelectric buzzer, 5 volt relay , transistor model 2N2222, diode 1N4004, 10 K and 3.3K resistors. Raspbeery pi with camera module is shown in Fig. 2, Arduino development board and MQ-3 sensor is shown in Fig. 3 and Fig. 4.

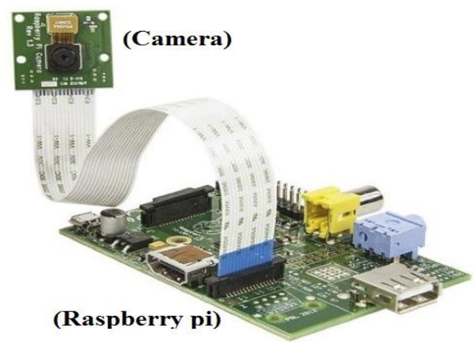


Fig. 2. Raspberry pi with camera module

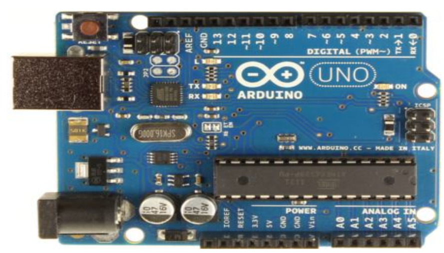

Fig. 3. Arduino UNO-R3 development board

Fig. 4. MQ-3 alcohol gas sensor

### *C. Interfacing of hardware modules*

Interfacing between several hardware sections and components is shown in Fig. 5. As the Raspberry pi model runs on 3.3 volt, so the I2C bus of Raspberry pi cannot be directly interfaced with Arduino UNO which runs on 5 volt. The interfacing is made by using a I2C level shifter which is created with BS170 mosfet [24]. Level shifter circuit is shown in Fig. 6. Raspberry pi is configured here as master device and hence Arduino is configured to work in slave mode. Alcohol sensor MQ-3 pin B1 drives an output voltage which is proportional to the percentage of alcohol gas present near the sensor atmosphere and it is connected to A/D converter pin A0 of the Arduino UNO. Piezoelectric buzzer model KPEG260 is directly connected to the Arduino digital pin D13.

Piezoelectric buzzer contains internal oscillator circuit and hence only a power supply is required to turn on the buzzer. KPEG260 can work with 3 to 5 volt supply, which is well within the range of an Arduino digital pin and that's why it is directly connected to Arduino digital pin. A 5 volt relay circuit is connected to the D12 pin of Arduino by means of a 2N2222 transistor based driver circuit, which can be used as a digitally controlled switch.

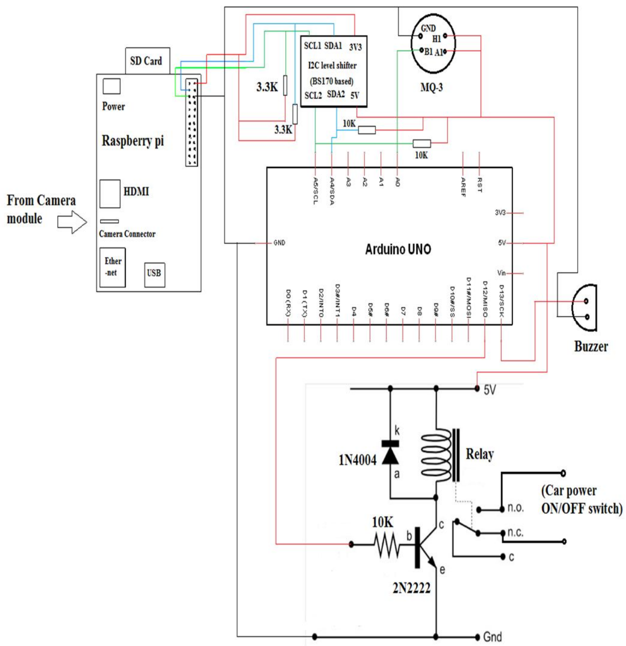


Fig. 5. Circuit diagram for the proposed model

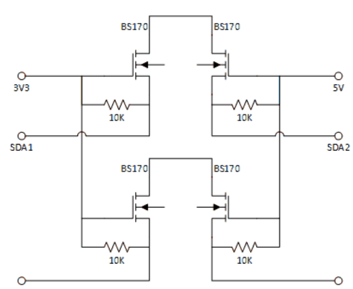


Fig. 6. I2C level shifter using BS170

## V. Programming Algorithm

### *A. Algorithm for Raspberry pi*

Programming for Rapberry pi is accomplished with Python IDLE software with Open CV computer vision extension installed. Both of the software's are free to download and open source. The overall programming algorithm for Raspberry pi module is represented with the help of a flowchart and it is shown in Fig. 7. The program will continue to run unless it is terminated using the command line interface of Raspberry pi or the device is turned off. When the program execution is initiated it will import the libraries for timer, I2C, numpy, and Open CV. Libraries are loaded with the following commands:-

```
import time
import smbus
import numpy as np
import cv2
```

The next step is to load .xml files which contain the information of objects to be detected. These are created with haar trainer by providing a huge number of pictures of desired object and without the object. So, the files are loaded in this way:-

```
face_region=cv2.CascadeClassifier('haarcascade_frontalface_default.xml')
eye_region = cv2.CascadeClassifier('haarcascade_eye.xml')
open_eye_region=cv2.CascadeClassifier('haarcascade_openeye.xml')
```

Here face_region, eye_region and open_eye_region objects are loaded with cv2.CascadeClassifier command. The next steps are almost as like as the flowchart. Camera device object is created with , cap = cv2.VideoCapture(0) . I2C device is initiated with the command, bus = smbus.SMBus(1). Raspberry pi is considered here as a master device and it needs to store the device address of the Arduino module. After the above steps, the program will enter to a while loop. Video frame is captured with the command ret, frame = cap.read(), here frame will hold the information content of the image. All the processing operations are performed on this frame. To detect regions, the colored image is converted to a gray image with the command, gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY). For detection of eye region the face region is to be detected at 1$^{st}$, because eyes are found on face and hence we can apply the searching algorithm to a certain area which will enhance the speed and efficiency of the program. Face region is extracted with the command,
faces = face_region.detectMultiScale(gray, 1.3, 5). It will return 4 quantities X coordinate, Y coordinate, length and width, using this set of information a rectangle of desired color can be drawn using :-

cv2.rectangle(frame,(x,y),(x+w,y+h),(255,0,0),2) command, which will draw a blue (B,G,R) colored rectangle around the face. Now the program is interested within the face region to locate the eye and hence region of interest is updated. Now the same procedure is followed and eye region is extracted from face region. After this step here it comes the final step, the program will search whether the eye is open or not by open eye region detection and draws a red colored rectangle within the detected region. The whole procedure follows in while loop. The program searches for the red rectangular region and calculates the area. If the area is 0, which means eye is closed and eye close count is incremented. After each 5 second it will compare the eye close count to a threshold value and if it crosses the threshold value that means the driver is in drowsiness and the program will send a warning message to the Arduino module using bus.write_byte_data() command. The further actions are performed by the Arduino module.

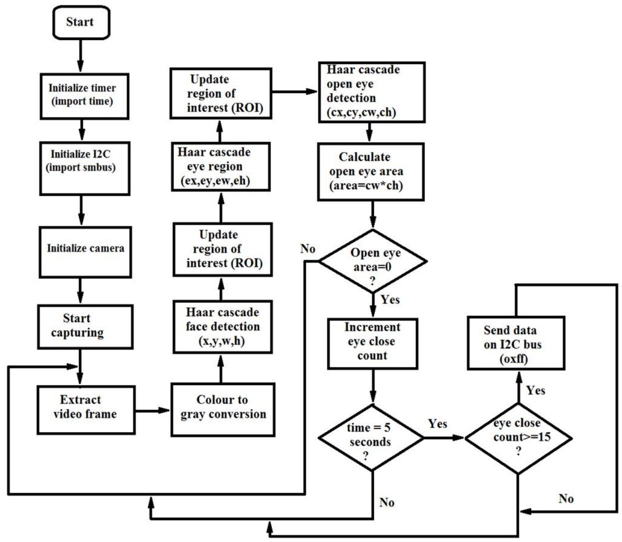


Fig. 7. Raspberry pi python programming algorithm

### *B. Algorithm for Arduino module*

Programming for Arduino module is done in Arduino IDE software. This is available for free download and also open source. The programming for the Arduino module is quite straight forward and it is shown as a flowchart in Fig. 8. Arduino module is configured here as a slave device, which can execute a set of task on receiving the data from master device. Arduino wire library (wire.h) is used for I2C communication. I2C bus is joined using the command:-

Wire.begin(address); where Raspberry pi I2C address is placed as argument in address. Receive event is configured with the command:-

Wire.onReceive(receiveEvent); here receiveEvent is the name of function which is to be defined with desired task that should be performed upon receiving data from the master module. Initialization of all the devices are performed inside Aduino IDE built in function: void setup() {}. Arduino ADC channel A0 is connected to the MQ-3 sensor and it is configured as :-

```
Int sensorPin= A0;
sensorValue = analogRead(sensorPin);
```

The sensor value is continuously checked within void loop(){} function and if sensor value crosses the threshold value, that means the driver is drunk and it will turn on the buzzer with command: digitalWrite(buzzerPin, HIGH);
In the same way it will turn off the relay module, which was previously turned on by the program with the following command: digitalWrite(relayPin, LOW);

Buzzer and relay task is also included inside void receiveEvent(){} function definition, which is executed on receiving data from Raspberry pi when the driver is in drowsiness.

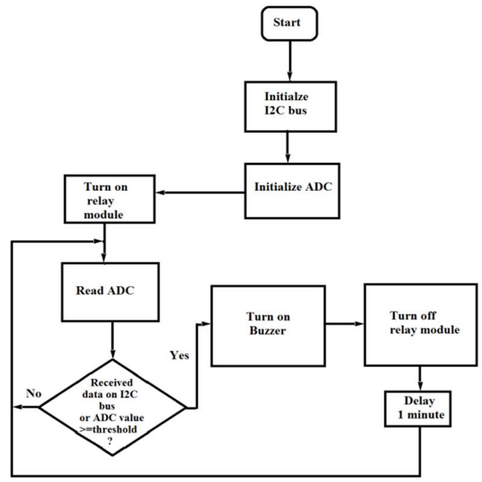


Fig. 8. Arduino UNO programming algorithm

## VI. Experimental Results

The basic framework of the system is built and still it is in the stage of research and to be fitted into a real car environment to measure the actual accuracy of the proposed system. Initial stage of the software development is done. The result shows the working of our proposed algorithm. The Raspberry pi program is built inside a linux based PC which can be transferred to Raspberry pi. At this moment the program can detect face region with blue rectangle, eye region with green rectangle and open eye region with red rectangle. Fig. 8 shows the detection of face, eye and open eye region. On the other hand in Fig. 10 as the eyes of the person are closed and hence red rectangle is not there which returns a zero as open eye area and at that instant the program throws a data to PC serial port that is captured by a hyper terminal application, which is realized in place of an I2C protocol to test the working of the program. On the other hand, alcohol sensor program for the Arduino module is tested and found properly working. Voltage samples from the MQ-3 sensor with and without the presence of an alcohol bottle (opened) is continuously converted to a range of numbers between 0-1023, which is converted as %BAC and send to a PC hyper terminal program to record the data, which is shown in the TABLE I .

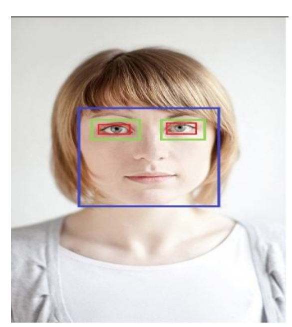

Fig. 9. Open eye detection

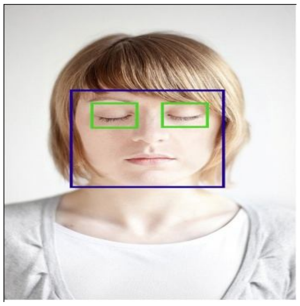

Fig. 9. Closed eye detection

TABLE I
ALCOHOL SENSOR DATA IN PC TERMINAL

| Serial No. | Before exposure to alcohol (% BAC) | After exposure to alcohol (%BAC) |
|---|---|---|
| 1 | 0.0144 | 0.0634 |
| 2 | 0.0144 | 0.0638 |
| 3 | 0.0145 | 0.0639 |
| 4 | 0.0144 | 0.0640 |
| 5 | 0.0144 | 0.0641 |
| 6 | 0.0143 | 0.0640 |
| 7 | 0.0145 | 0.0638 |
| 8 | 0.0144 | 0.0637 |
| 9 | 0.0144 | 0.0636 |

## VII. Conclusion

In this project a distinct system is designed which combines the application of computer vision with embedded systems and are targeted for reducing road accidents due to driver drowsiness and alcoholic intoxication. Development of software algorithm is completed which is partially tested and found successfully working. The research is still in continuation to develop it into a full blown system. There is much yet to improve and work on in this field.

## References


[1] R. Oyini Mbouna , S.G. Kong , and Myung-Geun Chun , "Visual Analysis of Eye State and Head Pose for Driver Alertness Monitoring," *IEEE Transactions on Intelligent Transportation Systems,* vol.14, no.3, pp.1462,1469, Sept. 2013

[2] Chin-Teng Lin et al, "A Real-Time Wireless Brain–Computer Interface System for Drowsiness Detection," *IEEE Transactions on Biomedical Circuits and Systems,* , vol.4, no.4, pp.214,222, Aug. 2010

[3] B.-G. Lee , S.-J. Jung , and W.-Y. Chung , "Real-time physiological and vision monitoring of vehicle driver for non-intrusive drowsiness detection," *IEEE Transactions on Communications,* vol.5, no.17, pp.2461,2469, November 2011

[4] Chin-Teng Lin et al, "Development of Wireless Brain Computer Interface With Embedded Multitask Scheduling and its Application on Real-Time Driver's Drowsiness Detection and Warning," *IEEE Transactions on Biomedical Engineering*, vol.55, Issue.5, pp.1582,1591, May 2008

[5] C.A. Perez , V.A. Lazcano , and P.A. Estevez , "Real-Time Iris Detection on Coronal-Axis-Rotated Faces," *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews,* vol.37, no.5, pp.971,978, Sept. 2007

[6] M.V. Ramesh , A.K. Nair , and A.T. Kunnath , "Real-Time Automated Multiplexed Sensor System for Driver Drowsiness Detection," *IEEE 7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM), 2011* , vol., no., pp.1,4, Sept. 2011

[7] Tianyi Hong , Huabiao Qin , and Qianshu Sun , "An Improved Real Time Eye State Identification System in Driver Drowsiness Detection," *IEEE International Conference on Control and Automation, 2007. ICCA 2007,* pp.1449,1453, May 30 2007-June 1 2007

[8] S. Vitabile , A. De Paola , and F. Sorbello , "Bright Pupil Detection in an Embedded, Real-Time Drowsiness Monitoring System," *24th IEEE International Conference on Advanced Information Networking and Applications (AINA), 2010* , pp.661,668, April 2010

[9] Tianyi Hong, Huabiao Qin, "Drivers drowsiness detection in embedded system," *IEEE International Conference on Vehicular Electronics and Safety, 2007. ICVES.*, pp.1,5, Dec. 2007

[10] M.J. Flores , J.M. Armingol ,and A. de la Escalera , "Driver drowsiness detection system under infrared illumination for an intelligent vehicle," *IEEE Transactions on Intelligent Transport Systems,* vol.5, no.4, pp.241,251, December 2011

[11] K. Kojima , S. Tamura , and Y. Omura , "Advanced technique to suppress subject variability for bio-impedance based alcohol-intake detection," *IEEE International Conference on Sensors, 2012,* pp.1,4, Oct. 2012

[12] K.I. Arshak et al, "NiO-$TiO_2$ thick-films for detection of alcohol vapours at room temperature," *Proceedings of IEEE Sensors, 2004.*, vol., no., pp.681,684 vol.2, Oct. 2004

[13] K. Arshak, E.G. Moore, and C. Cunniffe, "Surfactant treated drop-coated polyethylene adipate carbon black nanocomposite sensor for alcohol vapour detection," *Proceedings of the 2006 IEEE on Sensors Applications Symposium, 2006.* , pp.151,156, February. 2006

[14] R. Wagiran et al, "Development of a simple in situ instrumentation for detection of breath alcohol and gas leak," *IEEE International Conference on Semiconductor Electronics, 2002. Proceedings. ICSE 2002.*, pp.470,474, Dec. 2002

[15] Shao Jie et al, "Remote Detection of Alcohol Concentration in Vehicle Based on TDLAS," *IEEE 2010 Symposium on Photonics and Optoelectronic (SOPO),* pp.1,3, June 2010

[16] Element14 website. *Raspberry-Pi Technical Data Sheet* [online].Available: http://www.element14.com/community/docs/DOC-65470/l/raspberry-pi-technical-data-sheet

[17] Arduino.cc. *ArduinoUno Overview* [online].Available http://arduino.cc/en/Main/arduinoBoardUno

[18] OpenCV website. *Installation in Linux* [online]. Available: http://docs.opencv.org/doc/tutorials/introduction/linux_install/linux_install.html

[19] Hanwei Electronics Co., Ltd. *TECHNICAL DATA MQ-3 GAS SENSOR* [online]. Available: http://www.hwsensor.com

[20] Kingstate Electronics Corporation. *KPEG260 datasheet* [online].Available: http://www.kingstate.com.tw

[21] Python *website. Python versions* [online]. Available: https://www.python.org/download/

[22] OmniVision Technologies. *OV5647 color CMOS QSXGA (5 megapixel) image sensor datasheet* [online]. Available: http:// www.ovt.com

[23] P. Viola, M. Jones, "Rapid object detection using a boosted cascade of simple features," *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001.* , vol.1, pp.I-511,I-518, 2001

[24] Fairchild Semiconductor. *BS170 datasheet* [ online]. Available: http://www.fairchildsemi.com