

Topic: IoT Based Alcohol Detection System on High School Students

1. Team 9 Members:

Names	Reg. No.
1. Vincent NTAMBARA	220004734
2. Adeline NIBAGWIRE	215028654
3. Mathieu ITANGUKWISHATSE	215040884
4. Judith BIZIMANA	220000087
5. Valantine NYIRASAFARI	220015365

2. Project Supervisor: Prof Kayalvizhi Jayavel

3. General objective of the system.

The main objectives of this project is to test the influence of alcohol in high school students individually, to eradicate the habit of taking alcohol while at school which is causing conflicts between students themselves and even between students and teachers.

The specific objectives of the system

- Testing students before they enter school premises to find out who might be under alcohol influence.
- This goes along with luggage checking to stop unauthorized hazardous materials on school ground.
- To have competent and high performing students.

4. System Requirements

i. Hardware requirements

- **MQ-2 Gas sensor**

The Grove Gas Sensor (MQ2) module is useful for gas leakage detection (School, home and industry). It is suitable for detecting H₂, LPG, CH₄ and CO. When the target alcohol gas exist, the sensor's conductivity is higher along with the gas concentration rising. MQ-2 gas sensor has high sensitivity to Alcohol, and has good resistance to disturb of gasoline, smoke and vapor.

This sensor provides an analog resistive output based on alcohol concentration. When the alcohol gas exist, the sensor's conductivity gets higher along with the gas concentration rising.



Figure 1: MQ-2 Gas sensor

- **ESP8266 NodeMCU**

NodeMCU Development Kit/Board consists of ESP8266 Wi-Fi chip. ESP8266 chip has GPIO pins, serial communication protocol, etc. ESP8266 is a low-cost Wi-Fi chip developed by Expressive Systems with TCP/IP protocol. It is used to process the instructions form of codes, interprets them and gives out the results.

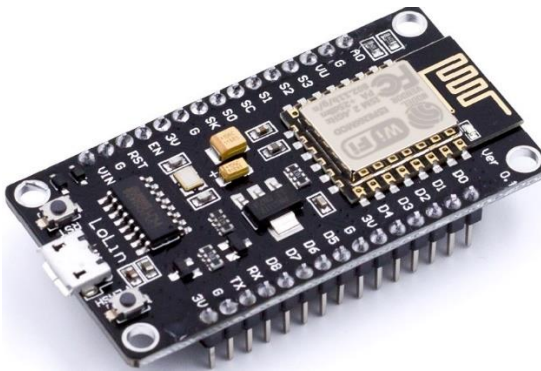


Figure 2: A NodeMCU

- **Jump wires**

A jump wire is an electrical wire, or group of them in a cable, with a connector or pin at each end (or sometimes without them – simply "tinned"), which is normally used to interconnect the components of a breadboard or other prototype or test circuit, internally or with other equipment or components, without soldering. Individual jump wires are fitted by inserting their "end connectors" into the slots provided in a breadboard, the header connector of a circuit board, or a piece of test equipment.

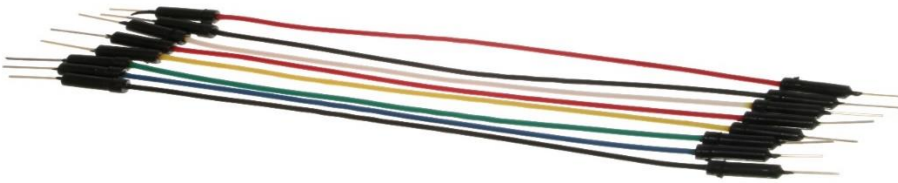


Figure 3: Jump wires

- **Breadboard**

A breadboard is a rectangular plastic board with a bunch of tiny holes in it. These holes let you easily insert electronic components to prototype (meaning to build and test an early version of) an electronic circuit, like this one with a battery, switch, resistor, and an LED (light-emitting diode).

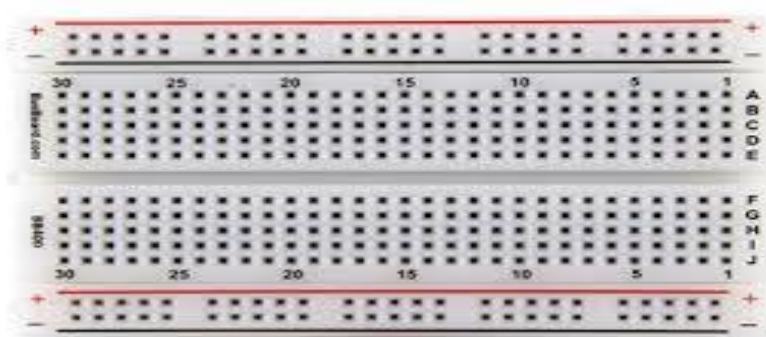


Figure 4: Breadboard

- **LED:** Do some action based on the detected level of alcohol (ON/OFF)
- **Resistor 10K Ω :** Avoid the LED to burn by regulating the current that pass through it

ii. Software requirements

- **Arduino IDE 1.8.9**

The Arduino IDE is open-source software, which makes it easy to write codes and upload them to the board. It runs on Windows, Mac OS X, and Linux. This software can be used with any Arduino board.

- **Blynk App**

Blynk is a Platform with iOS and Android apps to control Arduino, Raspberry Pi and the likes over the Internet. It is a digital dashboard where you can build a graphic interface for your project by simply dragging and dropping widgets. When connected to system through codes, it reads the alcohol values and displays them on the smartphone.

5. Methodology

i. System Block Diagram

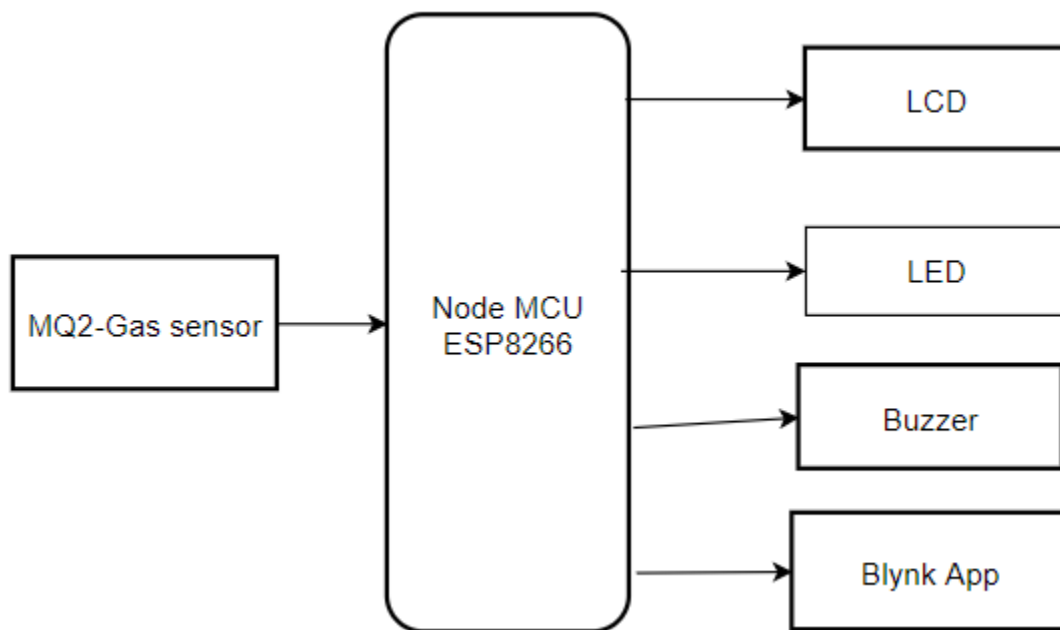


Figure 5: System Block Diagram

ii. How it works?

When a student comes, he/she will breathe on top of the MQ2 sensor, immediately MQ2 starts sensing then sends the findings to the microprocessor for processing/checking for alcohol components (Alcohol, Smoke, Co). If it find that the alcohol exceed the threshold, the LED will turns ON, the LCD displays the “**Alcohol detected!**” message with the alcohol values notifying that the student is under alcohol influence and the buzzer gives a sound alarm, If not LED will be OFF, there will be no sound from buzzer and the LCD will show this message “**No alcohol found!**” in the same time the Blynk App will display the Alcohol status for that student.

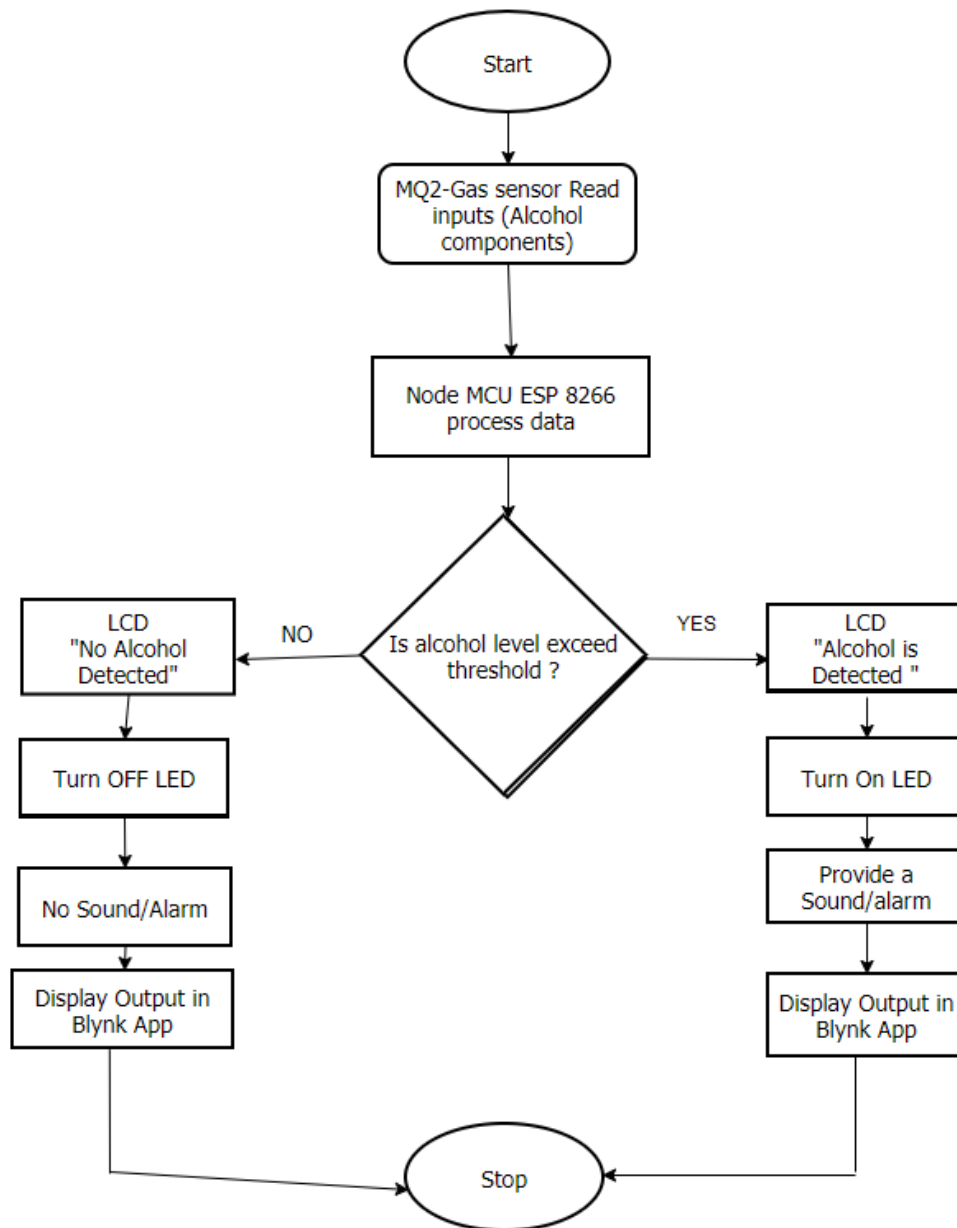


Figure 6: System Flow chart

iii. Steps for Designing Blynk Application UI

- First of all, download Blynk App for IoT and Install on an android smartphone
- Open the blynk application.
- Create an account/ Use existing Gmail Account
- Click on the new project and enter the project name.
- Click on the choose device and select NodeMCU.
- Make sure the connection type is set to WiFi.
- Finally, click on the create button, an authentication token will be sent on your email id, which later will be used in the NodeMCU programming.
- Click anywhere on the screen and search for the notification widget and add it.
- Again click on the screen and this time search for the Gauge and add it.
- Click on the gauge.
- Set the gauge name.
- Click on the PIN and select virtual pin V1, V2, V3
- Change the font size.
- Click on PUSH and select like 2 second (expected time to see the results from hardware).

iv. Working Codes

```
#define BLYNK_PRINT Serial

#include <ESP8266WiFi.h>

#include <BlynkSimpleEsp8266.h>

#include <MQ2Lib.h>

char auth[] = "Op-TZXZTtw8ZJrmzyRupHbu2fn5Xkf1J";

char ssid[] = "vincent";

char pass[] = "vincent1";

int pin = A0;

float alcohol = 0, co = 0, smoke = 0;

BlynkTimer timer;

MQ2 mq2(pin, true);

void sendSensor()
```

```

{
  Blynk.virtualWrite(V1, alcohol*100);
  delay (200);
  Blynk.virtualWrite(V2, co*100);
  delay (200);
  Blynk.virtualWrite(V3, smoke*100);
}

void setup()
{
  //Debug console
  Serial.begin(9600);
  mq2.begin();
  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 80);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);
  //timer.setInterval(1000L, sendSensor);
}

void loop()
{
  Blynk.run();
  timer.run();

  float* values= mq2.read(true); //set it false if you don't want to print the values in the Serial

  //Reading specific values:
  //lpg = values[0];

```

```
alcohol = mq2.readLPG();  
  
//co = values[1];  
  
co = mq2.readCO();  
  
//smoke = values[2];  
  
smoke = mq2.readSmoke();  
  
Serial.print ("CO: "); // output to the serial port  
  
Serial.println (co);  
  
Serial.print ("alocoho: "); // output to the serial port  
  
Serial.println (alcohol);  
  
Serial.print ("Smoke: "); // output to the serial port  
  
Serial.println (smoke);  
  
delay(2000);  
  
}  
  
}
```

6. Snapshots



Alcoho_Detectionl_System \$

```

#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include <MQ2Lib.h>
char auth[] = "Op-TZXZTtw8ZJrmzyRupHbu2fn5Xkf1J";
char ssid[] = "vincent";
char pass[] = "vincent1";
int pin = A0;
float alcohol = 0, co = 0, smoke = 0; BlynkTimer timer;
MQ2 mq2(pin, true);
void sendSensor()
{
  Blynk.virtualWrite(V1, alcohol*100);
  delay (200);
  Blynk.virtualWrite(V2, co*100);
  delay (200);
  Blynk.virtualWrite(V3, smoke*100);
}
void setup()
{
  //Debug console
  Serial.begin(9600);
  mq2.begin();
  Blynk.begin(auth, ssid, pass);
  // You can also specify server:
  //Blynk.begin(auth, ssid, pass, "blynk-cloud.com", 80);
  //Blynk.begin(auth, ssid, pass, IPAddress(192,168,1,100), 8080);
  //timer.setInterval(1000L, sendSensor);
}
void loop()
{
  Blynk.run();
  timer.run();
  float* values= mq2.read(true); //set it false if you don't want to print the values in the Serial

```

```
//Reading specific values:lpg = values[0];
alcohol = mq2.readLPG();
//co = values[1];
co = mq2.readCO();
//smoke = values[2];
smoke = mq2.readSmoke();
Serial.print ("CO: "); // output to the serial port
Serial.println (co);
Serial.print ("alocoho: "); // output to the serial port
Serial.println (alcohol);
Serial.print ("Smoke: "); // output to the serial port
Serial.println (smoke);
delay(2000);
}

/* MQ-3 Alcohol Sensor Circuit with Arduino */
/*SHUBHAM GUPTA - EMBEDDED ENGINEER*/

}
```

Figure 7: Alcohol Detection System codes screenshot

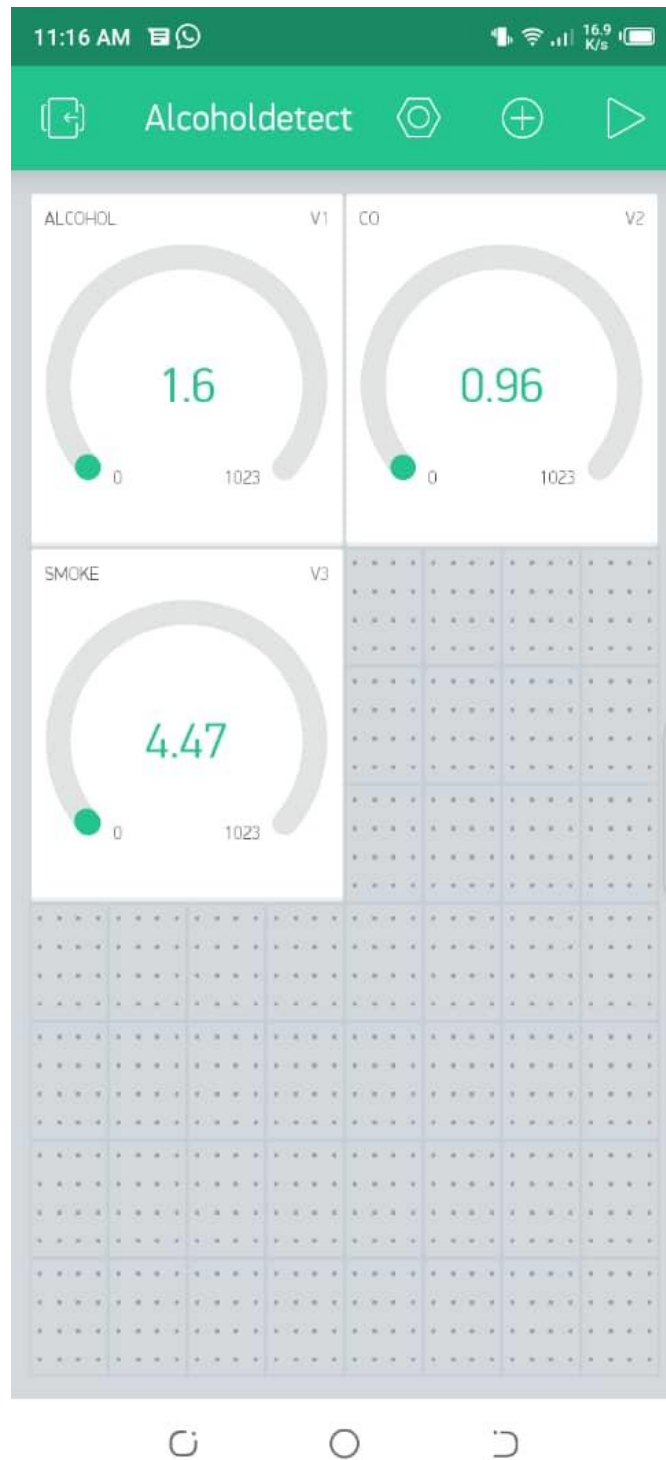


Figure 8: Alcohol detection System results on Blynk App Screenshot

7. Where else can this system be applied?

This project can be applied in different areas like:

- Traffic police (to stop accidents)
- This system can also be used in various companies or organizations to detect alcohol consumptions of employees.
- Detecting the presence of gases in the air such as methane, butane, LPG and smoke

8. References

YouTube Link: <https://www.youtube.com/watch?v=F0vK-lbHOXU>

Other Links:

1. <https://www.projectsof8051.com/arduino-workplace-alcohol-detector-with-reporting-through-sms/>
2. <https://create.arduino.cc/projecthub/yashikibrahim/alcohol-detection-sensor-dc32cd>
3. <http://www.ijetajournal.org/volume-2/issue-2/IJETA-V2I2P14.pdf>
4. <https://www.iosrjournals.org/iosr-jce/papers/Conf.17025-2017/Volume-2/11.%2066-71.pdf?id=7557>
5. <https://www.projectsof8051.com/arduino-workplace-alcohol-detector-with-reporting-through-sms/>
6. Lee, Assessing the Feasibility of Vehicle-Based Sensors to Detect Alcohol Impairment. 2010, National Highway Traffic Safety Administration: Washington, DC.

9. GitHub Link

<https://github.com/anibagwire/ACEIoT-Assignments>