**Faculty of Engineering and Information Sciences**

**ECTE250 Deliverable Cover Sheet**

Team Name: Team A

DELIVERABLE NUMBER AND TOPIC: Deliverable 5: Integrated Arduino Subsystem

Date Submitted: 27-5-2022

Team Members (list only group members who have taken part in production of the deliverable):

1. Yasmine Abualroos

2. Obai Alashram

3. Fardin Alam

4. Mohamed Waly

Team Member Submitting: Name: Yasmine Abualroos

ID No. 6940432 Signature:

Grade:

Table of Contents

[1.0 Gas sensor integrated circuit 3](#_Toc104507717)

[1.1 Simulation 3](#_Toc104507718)

[1.2 Breadboard circuit 5](#_Toc104507719)

[1.3 Veroboard 6](#_Toc104507720)

[2.0 Primarily testing 6](#_Toc104507721)

[2.1 Arduino code 6](#_Toc104507722)

[2.2 MQ9 sensor 7](#_Toc104507723)

[2.3 MQ2 sensor 8](#_Toc104507724)

[3.0 DC Fan (Motor circuit) 8](#_Toc104507725)

[4.0 Blynk layout and communication with the app 9](#_Toc104507726)

[5.0 Appendix 11](#_Toc104507727)

[5.1 The code used for the simulated circuit 11](#_Toc104507728)

[5.2 The code used for the integrated circuit 13](#_Toc104507729)

[5.3 Poster Draft 15](#_Toc104507730)

[6.0 References 16](#_Toc104507731)

Table of Figures

[Figure 1:Sensors circuit with LCD and buzzer 3](#_Toc104505066)

[Figure 2: State 00 4](#_Toc104505067)

[Figure 3: State 01 4](#_Toc104505068)

[Figure 4: State 10 5](#_Toc104505069)

[Figure 5: State 11 5](#_Toc104505070)

[Figure 6:Integrated breadboard circuit 6](#_Toc104505071)

[Figure 7:Sensors circuit with LCD on Veroboard 6](#_Toc104505072)

[Figure 8: MQ9 value using a lighter 7](#_Toc104505073)

[Figure 9:MQ9 test value (lighter) 7](#_Toc104505074)

[Figure 10: MQ9 test using car exhaust 8](#_Toc104505075)

[Figure 11: MQ9 gas level detected (car exhaust) 8](#_Toc104505076)

[Figure 12:Integrated circuit on a breadboard with the fan 9](#_Toc104505077)

[Figure 13: Blynk app layout 10](#_Toc104505078)

[Figure 14: Blynk app pin configurations with Arduino board 10](#_Toc104505079)

[Figure 15: Notification sent by Blynk app 11](#_Toc104505080)

[Figure 16: Push notification sent to user's mobile with Blynk app 11](#_Toc104505081)

1.0 Gas sensor integrated circuit

1.1 Simulation

to begin with, the circuit was built on Tinker cad (Figure 1) having two gas sensors, a buzzer, and the LCD to test the connections before proceeding with the breadboard circuit to reduce the risk of damaging the sensors.

Diagram, schematic

Description automatically generated

Figure :Sensors circuit with LCD and buzzer

The figures below illustrate the functionality of the circuit with respect to the state logic previously done.

A close-up of a circuit board

Description automatically generated with medium confidenceA screenshot of a computer

Description automatically generated with medium confidence

Figure : State 00

When neither of the sensors detects harmful gases, the LCD displays “no gas detected” and the buzzer stays off.

Graphical user interface, diagram

Description automatically generated

Figure : State 01

When the MQ2 sensor detects gas that is greater than the accepted threshold value while the MQ9 sensor gas levels are normal, the LCD displays” smoke detected” since the MQ2 sensor is mainly used to detect smoke, CH4(methane), LPG and Butane (Components101, 2018) and the buzzer goes ON to alert the user.

Graphical user interface, application

Description automatically generated

Figure : State 10

When the MQ9 sensor detects gas that is greater than the accepted threshold value while the MQ2 sensor gas levels are normal, the LCD displays” Flammable gas” since the MQ9 sensor is sensitive to CO (Carbon monoxide) and flammable gases. (Components101, 2018) Additionally, the buzzer goes ON to alert the user.

Diagram

Description automatically generated

Figure : State 11

Lastly, when both the MQ2 and the MQ9 sensors detect gas that is higher than the threshold, the LCD displays “Multiple Gases” and the buzzer goes ON, alerting the user of the presence of harmful gases.

Note that the threshold value was chosen only for the simulation purpose, and it does not represent the actual normal safe gas levels.

For instance, as per the current Occupational Safety and Health Administration (OSHA),**50ppm** is the permissible carbon monoxide exposure limit as an 8-hour time-weighted average. (US EPA, 2014)

The maximum indoor safe carbon monoxide level is **9ppm** (parts-per-million) over 8 hours. (Gaslab.com, 2021)

1.2 Breadboard circuit

The simulated circuit was implemented on the breadboard as shown in (Figure 6) where the threshold values were altered based on the values got while testing the sensors. The LCD print code part was also modified so that the LCD displays the values of gas measured by both sensors in addition to the type.

A picture containing text, electronics

Description automatically generated

Figure :Integrated breadboard circuit

1.3 Veroboard

The sensors circuit was implemented on the Veroboard as shown in (Figure 7).

A picture containing text

Description automatically generated

Figure :Sensors circuit with LCD on Veroboard

2.0 Primarily testing

2.1 Arduino code

To test the functionality of the Arduino subsystem, a modified version of the simulated circuit code was used where the threshold value was decreased to 500 which also is just used for the demonstration purpose and does not necessarily reflect the actual safe levels etc. In addition, the first (No gas) state was considered to occur when the gas values are lower than the threshold because realistically, the gas levels would not be exactly equal to 0.

2.2 MQ9 sensor

As the MQ9 sensor can be used to detect CO which is not usually present in an indoor area in concentrations when the MQ2 sensor was tested indoors with a lighter and smoke, very low values were displayed on the LCD in the range of 29-54ppm.

A picture containing text, electronics

Description automatically generated

Figure : MQ9 value using a lighter

A picture containing diagram

Description automatically generated

Figure :MQ9 test value (lighter)

To take the test a step further, the sensor was tested using car exhaust since cars, vehicles and the burning of fossil fuels are the greatest sources of carbon monoxide (CO) found in outdoor air. (US EPA, 2016)

The values were fluctuating in the range of around 450-650ppm. The figures below illustrate a sample value displayed; 645ppm.

A picture containing text, indoor, helmet

Description automatically generated

Figure : MQ9 test using car exhaust

A picture containing text

Description automatically generated

Figure : MQ9 gas level detected (car exhaust)

By doing the above, the claims that the gas sensor can detect carbon monoxide air concentration levels between 10 and 1,000ppm and combustible gas-air concentration levels between 100 and 10,000ppm. were satisfied. (store.ncd.io, 2022)

2.3 MQ2 sensor

On the other hand, testing the MQ2 gas sensor with a cigarette lighter and smoke was enough to raise its gas values detected to a range between 135-209ppm as shown in (Figure 8) and (Figure 9) since the MQ2 sensor is highly sensitive to smoke as mentioned previously.

3.0 DC Fan (Motor circuit)

The DC fan was integrated with the sensors circuit on the breadboard and the Veroboard as shown below (Figure 8).

A picture containing indoor, computer, electronics

Description automatically generated

Figure :Integrated circuit on a breadboard with the fan

As the DC fan came with a USB cable, the USB port was first cut off to expose the positive and negative wires in the cable which were then soldered to the motor driver since the wires were too thin to be placed securely into the motor driver by simply tightening the screws. Moreover, the motor driver was connected to the Arduino to enable it to be used as a voltage regulator and to control the fan speed when there is a need to switch it ON and OFF. Furthermore, 5 AAA batteries were used to power up the fan since it needs 12V to function which makes it more portable compared to using a DC power supply.

4.0 Blynk layout and communication with the app

The Blynk app is used for the IoT aspect of the product where the app can output the values from the sensors wirelessly. The mobile layout is shown in (Figure 9). There are 2 Gauges which show the MQ2 and MQ9 sensor values. There are indicators on the layout to signify if the fan and buzzer are active as well as two buttons to control the fan and the buzzer if needed by the user.

A picture containing diagram

Description automatically generated

Figure : Blynk app layout

For communication with the board, the Blynk app requires data streams to be set which specifies which pins in the Arduino Uno board to target when looking for data. Each connection is seen in (Figure 10). The gas sensors used analogue pins A0 and A1 of the Arduino while the fan and buzzer needed only the digital pins 6 and 10 as they function as ON or OFF whereas gas sensors output multiple values.

A picture containing table

Description automatically generated

Figure : Blynk app pin configurations with Arduino board

When gas is detected, a warning notification will be sent to the user immediately. The message which has been set up is shown in (Figure 11) and the setup for the push notification to mobile is highlighted with the red box in (Figure 12).

Graphical user interface, text, application, email

Description automatically generated

Figure : Notification sent by Blynk app

Graphical user interface, application, Teams

Description automatically generated

Figure : Push notification sent to user's mobile with Blynk app

5.0 Appendix

5.1 The code used for the simulated circuit

#include <LiquidCrystal.h>

LiquidCrystal lcd (12,11,5,4,3,2);

//int const sgP = A1;

//int sensor\_1=0 ; //MQ9

//int sensor\_2=0 ; //MQ2

int threshold=670;

void setup()

{

pinMode(A1, INPUT);

pinMode(A2, INPUT);

pinMode(7,OUTPUT);

Serial.begin(9600);

lcd.begin(16,2);

}

void loop()

{

int sensor\_1 ;

int sensor\_2;

sensor\_1 = analogRead(A1);

//Serial.println(sensor\_1);

sensor\_2 = analogRead(A2);

//State00

if ((sensor\_1==0) && (sensor\_2==0))

{

lcd.clear();

lcd.setCursor(0,0);

lcd.print("No Gas detected");

digitalWrite (7, LOW);

delay(100);

}

//State01

//else if ((sensor\_1==0) && (sensor\_2==1)){

else if ((sensor\_1<threshold) && (sensor\_2>threshold)){

lcd.clear();

lcd.setCursor(0,0);

lcd.print("smoke detected");

digitalWrite(7, HIGH);

delay(100);

}

//State10

else if ((sensor\_1>threshold) && (sensor\_2<threshold)){

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Flammable Gas ");

digitalWrite(7, HIGH);

delay(100);

}

//State 11

else if ((sensor\_1>threshold) && (sensor\_2>threshold)){

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Multiple Gases ");

digitalWrite(7, HIGH);

delay(100);

}

Serial.print("sensor1: ");

Serial.print(sensor\_1);

Serial.print(" sensor2: ");

Serial.println(sensor\_2);

delay(100);

}

5.2 The code used for the integrated circuit

#include <LiquidCrystal.h>

LiquidCrystal lcd (12,11,5,4,3,2);

//int const sgP = A1;

int sensor\_1=A0 ; //MQ9

int sensor\_2=A1 ; //MQ2

int buzzer=10;

int threshold=500;

int mpin1 = 6;

int mpin2 = 7;

void setup()

{

pinMode(A1, INPUT);

pinMode(A0, INPUT);

pinMode(sensor\_1, INPUT);

pinMode(sensor\_2, INPUT);

pinMode(7,OUTPUT);

pinMode(6,OUTPUT);

pinMode(buzzer,OUTPUT);

Serial.begin(9600);

lcd.begin(16,2);

}

void loop()

{

int gas1=A0;

int gas2=A1;

//pinMode(pin1, OUTPUT);

// pinMode(pin2, OUTPUT);

gas1 = analogRead(sensor\_1);

gas2 = analogRead(sensor\_2);

//Serial.println(sensor\_1);

//sensor\_2 = analogRead(A1);

//State00

if ((gas1<=threshold) && (gas2<=threshold))

{

lcd.clear();

lcd.setCursor(0,0);

lcd.print("No Gas detected");

lcd.setCursor(0,1);

lcd.print("MQ9:");

lcd.print(gas1);

lcd.setCursor(9,1);

lcd.print("MQ2:");

lcd.print(gas2);

digitalWrite (7, LOW);

digitalWrite(6,LOW);

digitalWrite(buzzer, LOW);

delay(100);

}

//State01

//else if ((sensor\_1==0) && (sensor\_2==1)){

else if ((gas1<threshold) && (gas2>threshold)){

lcd.clear();

lcd.setCursor(0,0);

lcd.print("smoke detected");

lcd.setCursor(0,1);

lcd.print("MQ9:");

lcd.print(gas1);

lcd.setCursor(8,1);

lcd.print("MQ2:");

lcd.print(gas2);

digitalWrite (7, HIGH);

digitalWrite(6,LOW);

digitalWrite(buzzer, HIGH);

}

//State10

else if ((gas1>threshold) && (gas2<threshold)){

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Flammable Gas ");

lcd.setCursor(0,1);

lcd.print("MQ2:");

lcd.print(gas1);

lcd.setCursor(8,1);

lcd.print("MQ9:");

lcd.print(gas2);

tone(buzzer,1000,350);

digitalWrite (7, HIGH);

digitalWrite(6,LOW);

digitalWrite(buzzer, HIGH);

}

//State 11

else if ((gas1>threshold) && (gas2>threshold)){

lcd.clear();

lcd.setCursor(0,0);

lcd.print("Multiple Gases ");

lcd.setCursor(0,1);

lcd.print("MQ9:");

lcd.print(gas1);

lcd.setCursor(8,1);

lcd.print("MQ2:");

lcd.print(gas2);

digitalWrite (7,HIGH);

digitalWrite(6,LOW);

digitalWrite(buzzer, HIGH);;

}

Serial.print("sensor1: ");

Serial.print(gas1);

Serial.print(" sensor2: ");

Serial.println(gas2);

delay(100);

}

5.3 Poster Draft

Diagram

Description automatically generated

6.0 References

Components101. (2018). *MQ2 Gas Sensor*. [online] Available at: <https://components101.com/sensors/mq2-gas-sensor>

US EPA. (2014). *Carbon Monoxide’s Impact on Indoor Air Quality | US EPA*. [online] Available at: <https://www.epa.gov/indoor-air-quality-iaq/carbon-monoxides-impact-indoor-air-quality#:~:text=%5BOSHA%20PEL%5D%20The%20current%20Occupational,CFR%20Table%20Z%2D1%5D>.

Gaslab.com. (2021). *Carbon Monoxide Levels Chart*. [online] Available at: <https://gaslab.com/blogs/articles/carbon-monoxide-levels-chart>

US EPA. (2016). *Basic Information about Carbon Monoxide (CO) Outdoor Air Pollution | US EPA*. [online] Available at: <https://www.epa.gov/co-pollution/basic-information-about-carbon-monoxide-co-outdoor-air-pollution#:~:text=The%20greatest%20sources%20of%20CO,can%20affect%20air%20quality%20indoors>.

store.ncd.io. (2022). *MQ-9 Carbon Monoxide Combustible Gas Sensor ADC121C 12-Bit ADC I2C Mini Module*. [online] Available at: <https://store.ncd.io/product/mq-9-carbon-monoxide-combustible-gas-sensor-adc121c-12-bit-adc-i2c-mini-module/#:~:text=The%20MQ%2D9%20is%20capable,65%25%20%C2%B15%25%20humidity>.

‌

‌