ALCOHOL DETECTION SYSTEM FOR CAR USING ARDUINO UNO

DEV PATEL

COMPUTER SCIENCE 207

UNIVERSITY OF REGINA

TREVOR THOMES

DECEMBER 6TH, 2019

TABLE OF CONTENTS:

INTRODUCTION:

The project made for this term was the construction of the human breath analyzer for the car it detects the alcohol level and displays that in the LCD if someone is drunk and wants to drive the car the system won't let the driver start the engine. This is the hacker's safety system for the car.

The basic idea of the system is safety purpose, there is an alcohol sensor named MQ135 detects the alcohol level in the environment and there is an LCD that shows the level and indicates the different modes which will be discussed further with more explanations in this paper.

Few hackers made this project which simply measures the alcohol level and displays that moreover, I have a new concept that I wanted to implement in it and

that was the locking system, I will also elaborate this locking part in this paper briefly.

INSPIRATIONS:

When I was browsing for the final projects I was watching different videos on the YouTube meanwhile I saw an advertisement published by the SGI it was about driving high is DUI, then I got inspired to control the accidents which are happening because of drinking and driving and drugs and many more, then I started searching alcohol sensor I came to know to the MQ sensors there are various type of sensors which are for gas and alcohol then I found the MQ135 which detects the alcohol. As I had to modify someone's projects which are already out there and implementing my new thing to that project, I started searching for the alcohol projects and few people worked on the MQ135 and I found the how2electronic.com which helped me with the projects.

From how2electronic.com I got a lot of information regarding the project for instance there are few more alternatives to the MQ135 and get introduced with few libraries. Also, he used the clear circuit diagrams in his presentation so it was easy for me to hook up the wires and components.

The way he arranged his project was good enough to understand any person. He also used alcohol to show how the sensor is working and displaying the data related to the sensor. He also explained the sensor and LCD connections.

DESIGNING PROCESS:

The design process was simple. The system is made of LCD, anMQ135 sensor, servo motor, breadboard, Arduino Uno and some jumper wires. The photograph of the system can be viewed in Appendix A.

The LCD is joined at the right part of the breadboard. The MQ sensor and the servo motor are joined besides the LCD. I have attached the sensor and the

servo motor with few jumper wires and those wires directly go to the breadboard. The reason behind why I put the sensor and the servo motor on sides of the LCD is I can show the whole working process of the system starting from testing alcohol then display the readings and the action of the servo motor altogether.

The original prototypes of the project can be found in Appendix B. It includes the design of the project. Moreover, you will notice a few changes now.

BUILDING PROCESS:

The building process was not that complex in the initial prototype. I have wired up all components with the jumper wires so it looks a bit messy but it can be easily identified. There is one LCD, one servo motor, one MQ sensor attached with the breadboard. There are around 18 connections including the ground and +5 volt pin.

First of all, there is a 16*2 LCD attached to the breadboard which is the highlight of the whole system because it is the biggest part of the system and it displays the interesting things on itself. There are 16 pins in total on that. There are

8 data pins, 2 positive pins, one negative, one ground and one contrast pin. In this system I used 12 pins in total from the LCD, 4 pins go to the ground, 2 pins go to the +5volts, and the 6 of them are data pins all of them are directly joined to the Arduino Uno.

Next is the MQ sensor, it has four pins in it we are using three of them and they are the Vcc, ground and the analog pins. The analog pin is attached to the pin A0. The features of the MQ135 are, it has wide detecting scope, fast response and high sensitivity, it is stable, it can be used as a digital or analog sensor, there is a potentiometer at the back of the sensor where we can set the sensitivity level moreover we can use this sensor in order to detect not only alcohol but also the NH3, smoke, CO2 and many more.

The third essential part of the system is the servo motor and it is to represent the locking system. It has three pins the yellow, red and, brown. The yellow pin goes to pin 10 and it gets commands from there to drive the motor, the red pin goes to the Vcc and the brown goes to the ground and the servo is now read to get commands from the MQsensor.

There are two more important pins joined to the breadboard from the Arduino to give power and to finish the circuit and those are Vcc and the ground which are at the corner of the breadboard.

USER MANUAL FOR SYSTEM:

The system is very easy to set up and use. First of all, one needs to download the code onto the Arduino UNO. Arduino UNO supplies 5v and it is enough for the system at this scale. Refer to Appendix c for the code information. Once you get the code, there are few steps to follow to hook up things.

First of all, we need two pins that directly go to the breadboard from the Arduino UNO and those are Vcc and the ground. Then we are setting up the LCD into the breadboard, there are 16 pins as I mentioned earlier. The first pin from the left side goes to the ground, second goes to the Vcc, the third goes to the ground again, fourth and the sixth go to the PIN 7and 6 respectively, the fifth one again goes to the ground then we will skip four pins D0 to D3 then we will attach D4, D5, D6,

D7 to the pin num 5,4,3,2 respectively and last two-pin goes to the Vcc and the ground.

Continuously, we will go to the MQsensor, it has three pins to join and those pins are clearly mentioned in the module we just need to follow the steps we are hooking up the analog pin to the A0 and Vcc to the +5v and the GND pin to the ground refer the Appendix D to get idea of the MQsensor's pin.

Moving ahead, we will plug the servo motor into the breadboard, there are also three pins yellow, orange, and brown and those are going to the pin 10 to the Arduino, Vcc, and ground respectively and we are done with the hooking up things. To get more ideas of the circuit and hooking up things refer to Appendix E and F.

HOW IT WORKS:

When you run the code to the system the sensor is detecting the alcohol level in the nearby environment so if it detects the alcohol it will print the level in the numbers and if it passes the threshold it will print the "alert..!!" and the servo motor rotates

to the 180 degrees. Here the servo motor playing the main role and it is the highlight of the whole system. If the servo motor rotates it indicates that there is an interruption between the start power button and the ignition system. So if any drunk person sits in the car and he is so drunk then the sensor will detect the level and it will not let him start the car.

SETBACKS AND FAILURES:

The biggest difficult part for me was to implement the whole system into the original car. As I do not own the car and no one will be ready to give their car to experiment thing so I had to use the servo motor to illustrate the power brake system for the engine.

Another problem I faced that to get parts, I order the parts from the Amazon and the shipping date was different for every pat and when I gathered all parts and was attaching the parts to the Arduino I came to know my LCD is not working for some reason so needed to reorder the LCD. The funny part was that for the first time I was attaching the LCD incorrectly so I wasted my lot of time in the LCD.

Other than that project was going smooth as I was working alone I did not face the group problems.

Another problem I faced my servo motor and the LCD is flickering frequently, I searched a lot but I didn't find the solution. On the online forums, it says that if we have high power then this problem may occur but for the Arduino, we need 5v I cannot reduce then that. I still did not find the solution for that and it would be my failure in this project.

MILESTONES:

The project should be considered a success because the system is responding when it detects the alcohol near the sensor so it fixes the main aim however the flickering bug needs to be fixed. It will continue to be improved after the end of the semester as well.

One thing that was not accomplished as I expected as my starch goal. It was going to be in the real car as a live demo with the locking system but due to time limits and the lack of knowledge with the cars I was not able to finish that up. This can be

done in the future where there is no time limit on that. I will try my best to work on that properly.

There is also one argument raised by my friend and that was what if someone else is drunk in the car, not the driver and the driver want to start the car and it won't allow starting then that will be the question, I would consider this point as my setback as well as my milestone for the future.

CONTRIBUTION:

Dev Patel: documentation, Github, improvising the code and the hardware, testing, presentation. I also would like to give credit to the how2electronics.com I referred to their project and got some idea for the circuit part and coding. Overall the guidelines of how2electronics.com and my work turned out this project well.

CONCLUSION:

As it is my first project I enjoyed it a lot it was so interesting and the important thing for me I got to learn a lot of things and every time I got intruded to new

interesting and knowledgeable topics and it was such fun for me. Moreover, this course also helped me to improve my academic knowledge as well as skills. I would love to work more on the Arduino and love to play with that as a hacker. I would love to finish my all setbacks as well as my future milestones and would like to come up with the new thing.

REFERENCES:

https://www.how2electronics.com/

https://github.com/trevortomesh/CDC/blob/master/CS207/CS207-Example-Writeup.pdf

https://app.schoology.com/page/2274079133

https://github.com/trevortomesh/OSHRepo

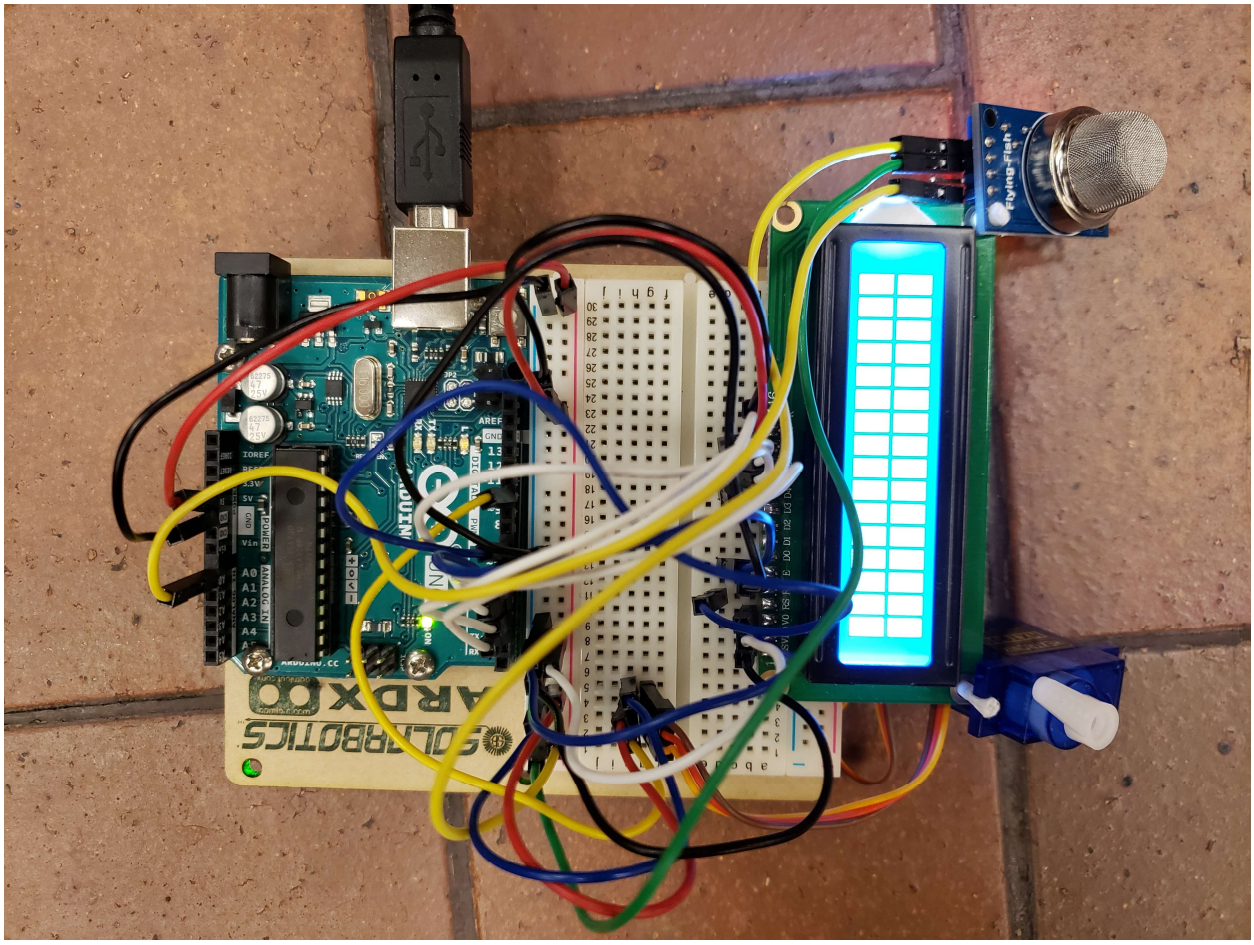
https://components101.com/sensors/mq135-gas-sensor-for-air-quality

https://components101.com/motors/mg996r-servo-motor-datasheet

https://components101.com/16x2-lcd-pinout-datasheet

https://www.youtube.com/watch?v=yWKfuP9BhhI

APPENDIX A:



APPENDEX: B

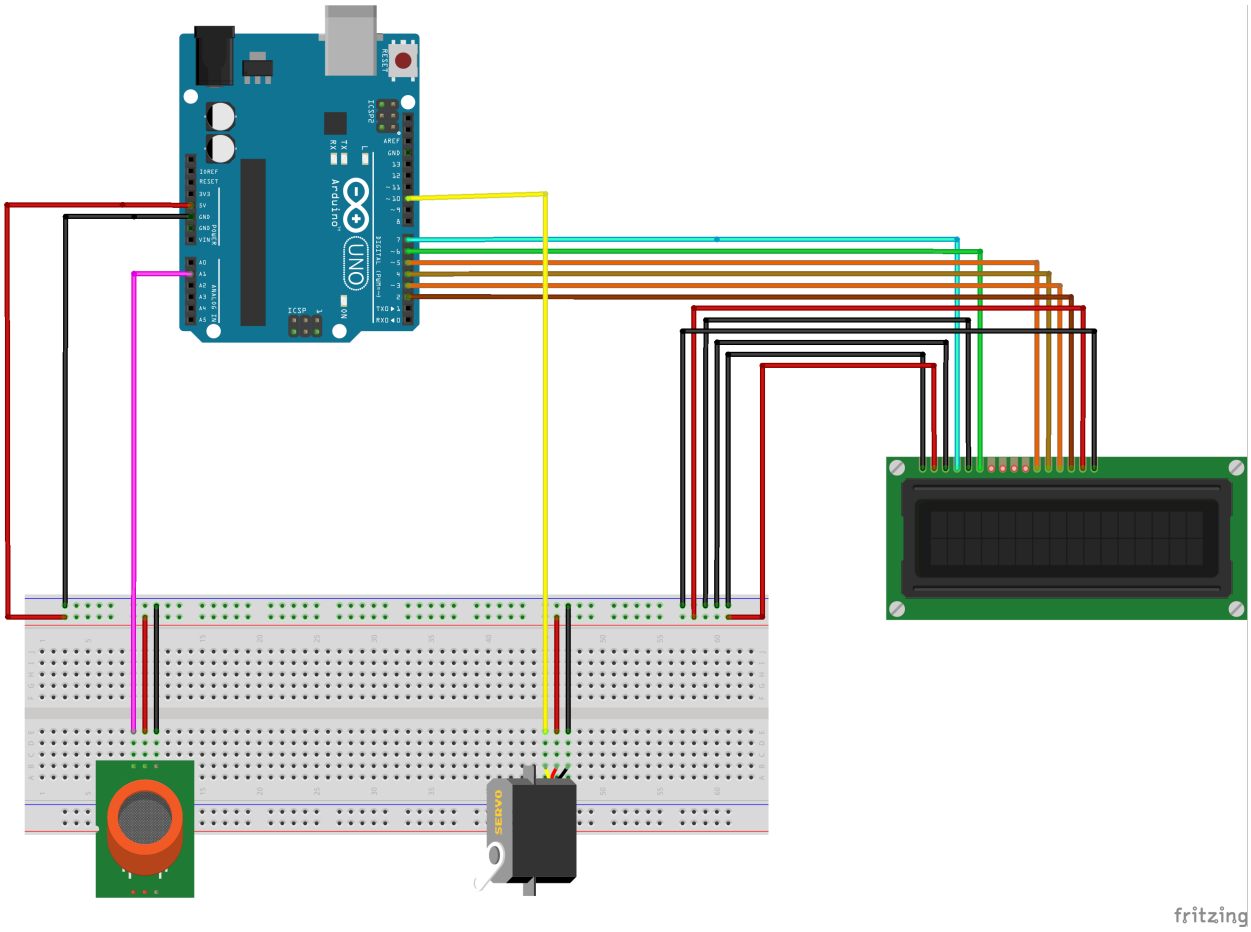
APPENDIX: C

```
#include <LiquidCrystal.h>
#include <Servo.h>
LiquidCrystal lcd(7, 6, 5, 4, 3, 2); // set the digital pins to display on LCD


Servo myservo;  // create servo object to control a servo
// a maximum of eight servo objects can be created

int pos = 0;    // variable to store the servo position


int sensorPin = A0; // set the alcohol sensor pin
int value;    //variable to store the sensor value

void setup()
{
  Serial.begin(9600); // set the baud rate
  lcd.begin(16, 2); // set the cursor of the LCD
  myservo.attach(10);  // attaches the servo on pin 10 to the servo object
}
void loop()
{
  int Value = analogRead(sensorPin);
  value = analogRead(A0);
  lcd.print("Alcohol Lev.:");
  lcd.print(value - 50);
  Serial.print(value);
  if (value - 50 > 400) // condition to display alert

  {
    //digitalWrite(ledPin,HIGH);
    lcd.setCursor(0, 2);
    lcd.print("Alert....!!!");// printing Alert...!!! to the lcd
    Serial.print ("Alert");
    for (pos = 0; pos < 180; pos += 1) // goes from 0 degrees to 180 degrees
```

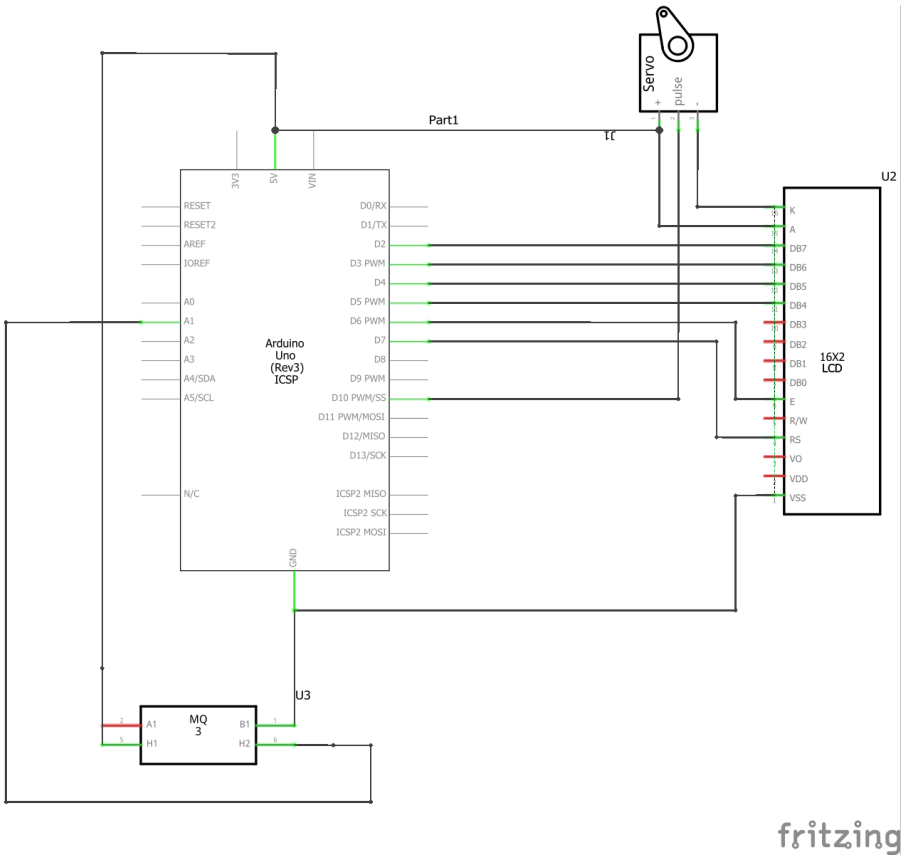
```
  { // in steps of 1 degree
    myservo.write(pos);           // tell servo to go to position in variable 'pos'
    //  delay(15);                // waits 15ms for the servo to reach the position
  }
 }
 else {
  lcd.setCursor(0, 2);
  lcd.print(".....Normal.....");
  Serial.print("Normal");
  for (pos = 180; pos >= 1; pos -= 1) // goes from 180 degrees to 0 degrees
  {
    myservo.write(pos);           // tell servo to go to position in variable 'pos'

  }
 }

 delay(500);
 lcd.clear();
}
```

APPENDIX D:

APPENDIX: E



APPENDIX:F


Servo
+
pulse
-
Part1
U2
K
A
DB7
DB6
DB5
DB4
DB3
DB2
DB1
DB0
E
R/W
RS
VO
VDD
VSS
16X2
LCD
3V3
5V
VIN
RESET
RESET2
AREF
IOREF
A0
A1
A2
A3
A4/SDA
A5/SCL
N/C
D0/RX
D1/TX
D2
D3 PWM
D4
D5 PWM
D6 PWM
D7
D8
D9 PWM
D10 PWM/SS
D11 PWM/MOSI
D12/MISO
D13/SCK
ICSP2 MISO
ICSP2 SCK
ICSP2 MOSI
GND
Arduino
Uno
(Rev3)
ICSP
U3
MQ
3
A1
H1
B1
H2
fritzing