Final Year Project Report

Bachelor of Engineering

A gas leakage detection & accident prevention system

Student: Jiayi Feng

GUID: 2288941F

1st Supervisor: Keep blank for Moodle submission but put the name before hardcopy submission******

2nd Supervisor: Keep blank for Moodle submission but put the name before hardcopy submission ******

2019-2020

Coursework Declaration and Feedback Form

The Student should complete and sign this part

Student Name: Jiayi Feng	Student GUID:2288941F
Course Code : UESTC4006P	Course Name : INDIVIDUAL PROJECT 4
Name of 1st Supervisor: Keep blank for Moodle submission but put the name before hardcopy submission	Name of 2nd Supervisor: Keep blank for Moodle submission but put the name before hardcopy submission
Title of Project: A gas leakage detection & accident prevention system	
Declaration of Originality and Submission Information	
I affirm that this submission is all my own work in accordance with the University of Glasgow Regulations and the School of Engineering requirements Signed (Student) :	UESTC4006P
Date of Submission: 29/04/2020	
Feedback from Lecturer to Student – to be completed by Lecturer or Demonstrator	

Grade Awarded: Feedback (as appropriate to the coursework which was assessed):	
Lecturer/Demonstrator:	Date returned to the Teaching Office:

Abstract

Nowadays, combustible gas, including liquid petroleum gas (LPG), propane, and alcohol are widely utilized in both daily life and industry, with both convenience and potential danger. Particularly, the combustible gas leakage, which can easily result in heavy loss, even tragedies, especially at the industrial level, such as mining and tunnel construction. Gas leakage, whether it is natural or man-made, is unpredictable and harmful, and it can pose a great threat to both individuals and the environment. So, it is highly desirable to develop a combustible gas leakage and detection system.

In this project, I have developed a gas leakage and detection system, based on Arduino, gas sensor MQ-2, alarm components (LED, buzzer, GSM module SIM900A), and feedback component, micro-servo SG90. When the detected combustible gas concentration is lower than the threshold value, the micro servo SG90 will continue to rotate back and forth regularly, and the LED will keep shining. When the detected combustible gas concentration value is higher than the threshold one, the Arduino will give instructions to both the alarming and feedback part. For the alarming part, an LED will start blinking, a buzzer will produce a sharp and annoying noise, and an SMS (short message service) will be delivered to the system administrator's mobile phone by the GSM module SIM900A. For the feedback part, the micro-servo will also be tightened to prevent potential damage.

Compared with other gas leakage and detection system, this system has the advantages of simplicity, compatibility and has wide applications.

Acknowledgements

I am very grateful to my supervisor Lianping, for his support during the whole period, including abundant reasonable suggestions, useful information resources, and quick response to my questions, whether they are easy or difficult. Thanks to my supervisor again for his patience and time.

Besides, I am also very grateful to the UESTC-UoG joint program, the birthplace of an engineering dream. Thanks to the responsible staff and gracious professor I have met. Without their guidance and instruction, an engineering rookie will never be transferred into a qualified graduate.

Contents

Abstract ... 4
Acknowledgements ... 5
1 Introduction ... 9
1.1 Introduction of the designed system ... 10
1.2 Arduino ... 13
1.3 Schematic wire connections ... 15
1.4 Welding ... 18
2 Hardware level ... 19
2.1 The detection part: MQ-2 gas sensor ... 19
2.2 The alarming part ... 22
2.2.1 LED ... 22
2.2.2 Buzzer ... 24
2.2.3 SIM900A ... 26
2.3 The feedback part ... 30
3 Software compiling ... 32
3.1 The detection part: MQ-2 gas sensor ... 34
3.2 The alarming part ... 35
3.2.1 LED ... 35
3.2.2 Buzzer ... 36
3.2.3 SIM900A ... 37
3.3 The feedback part ... 38
4 Analysis and Discussion ... 39
4.1 Arduino ... 39
4.2 SIM900A ... 39
4.3 MQ-2 ... 39
5 Conclusions and further work ... 41
5.1 Conclusions ... 41
5.2 Suggestions for further work ... 41
References ... 43
A Appendices(codes for the project) ... 45
B University’s Plagiarism Statement ... 49

List of Figures

Figure 1. The setup of actual gas leakage detection & accident prevention system

Figure 2. The logic flow of the system design

Figure 3. The picture of Arduino module

Figure 4. The Programming interface of Arduino Software (IDE)

Figure 5. Arduino’s pin configuration

Figure 6. The schematic diagram of gas detection’s wire connection with Arduino

Figure 7. The schematic diagram of the feedback part’s wire connection

Figure 8. The schematic diagram of the wire connections between Arduino and the alarming part of SIM900A, LED, and Buzzer (from top to bottom)

Figure 9. Picture of gas sensor MQ-2 and the screw to tune the sensitivity

Figure 10. Gas sensor MQ-2’s pin

Figure 11. The picture of the LED

Figure 12. The actual effect of the yellow LED

Figure 13. The picture of the buzzer module

Figure 14. Picture of SIM900A module (without antenna and wire)

Figure 15. Picture of CH340 power supply module

Figure 16. Picture of SIM900A module and its pins

Figure 17. SIM900A’s pin connection

Figure 18. Micro-servo SG90 components

Figure 19. Schematic picture of the pins of SG90

Figure 20. The Arduino General-Purpose Input Output (GPIO)

Figure 21. Code to tune the tone of the buzzer

Figure 22. Code to write the sensor value into the serial monitor and set the period

Figure 23. Code to setup LED’s shining state for safe condition

Figure 24. Code to setup LED’s blinking state for gas leakage condition

Figure 25. The alarm ring code

Figure 26. The code to send message

Figure 27. SG90’s code

Figure 28. The gas leakage message in Sms format

List of Tables

Table 1. The working condition of the MQ-2 gas sensor

Table 2. The advantages of the MQ-2 gas sensor

Table 3. Music notes and their corresponding frequency

Table 4. Part of the AT command for SIM900A

1 Introduction

After the industrial revolution, people's growing demands for energy has increased dramatically, and combustible gas, such as liquid petroleum gas (LPG), propane, and alcohol, is one of the most fundamental suppliers for its higher energy conversion efficiency and less damage to the environment [1]. In family life, liquefied petroleum gas has been people's convenient cooking and heating fuel for years. While at the industrial level, the combustible gas will get involved in chemical ore refining in the furnace.

However, the widespread use of combustible gas has brought potential danger to people in both domestic and industrial perspective [2]. Owing to the toxic chemical substances inside, the exposure to high-density combustible gas will possibly do damage to people's respiration, and the worst case is the existence of the sparkle in the sealed space above, in most cases, mine tunnel and family room [3]. As a result, fire disasters, explosion, and poisoning happened after the gas leakage.

In order to reduce the gas leakage accident, the gas leakage and detection system are introduced. Nowadays, most of the system in the market has the effect of making noise and lighting, which means buzzer and LED in the hardware level. However, the introduced system is more multifunctional for its feedback part, the simulation of the real situation's gas valve, and the communication part. The feedback part is simplified as the micro servo, and it should be stopped when the gas leakage happened to prevent the gas emission. As for the communication part, the user will receive an SMS (short message service) once the gas leakage happened.

1.1 Introduction of the designed system

The overall system can be divided into two conditions: one that works well, which means the gas concentration is lower than the threshold value, while the other condition is gas leakage.

As for the former one, the micro-servo works well and the LED lights on continuously. The rest of the components are closed. However, when it comes to the latter gas leakage condition, the LED will blink quickly and the buzzer will make annoying noise, to give direct alert to people around. Additionally, the SIM900A will send an SMS to the administrator. The micro-servo will be stopped and return to its initial position, 0 degrees, to avoid exacerbating the gas leakage condition.

Here is the finished system in Figure 1.1.

Figure 1.2 shows the logic flow chart of the system design, which is described as gas leakage detection by MQ-2, followed by the LED blinking, SIM900A’s SMS sending, and annoying buzzer noise. Last but not least, the feedback micro-servo will be tightened and return to its initial position to extenuate gas leakage.

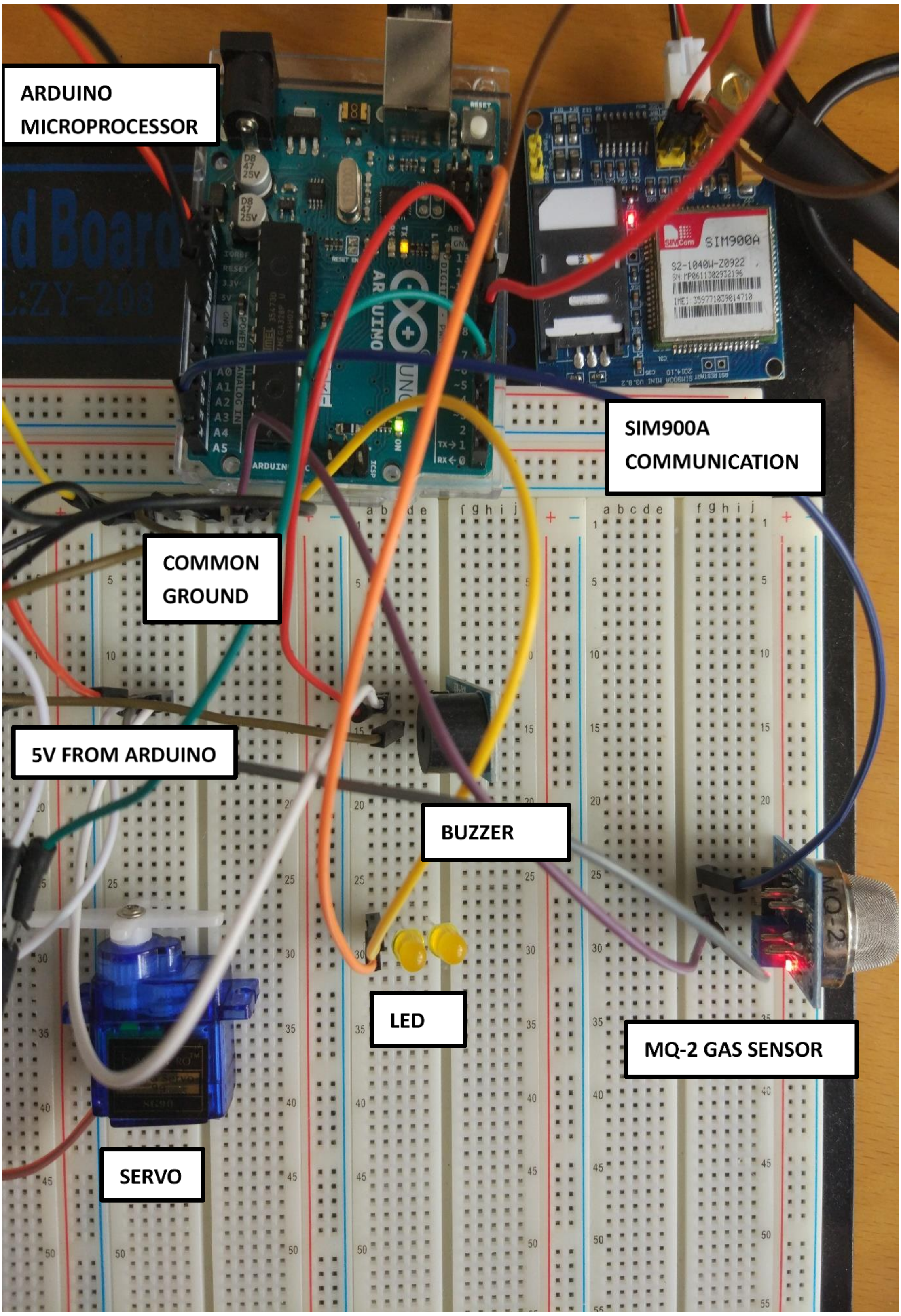

Figure 1.1. The setup of actual gas leakage detection & accident prevention system

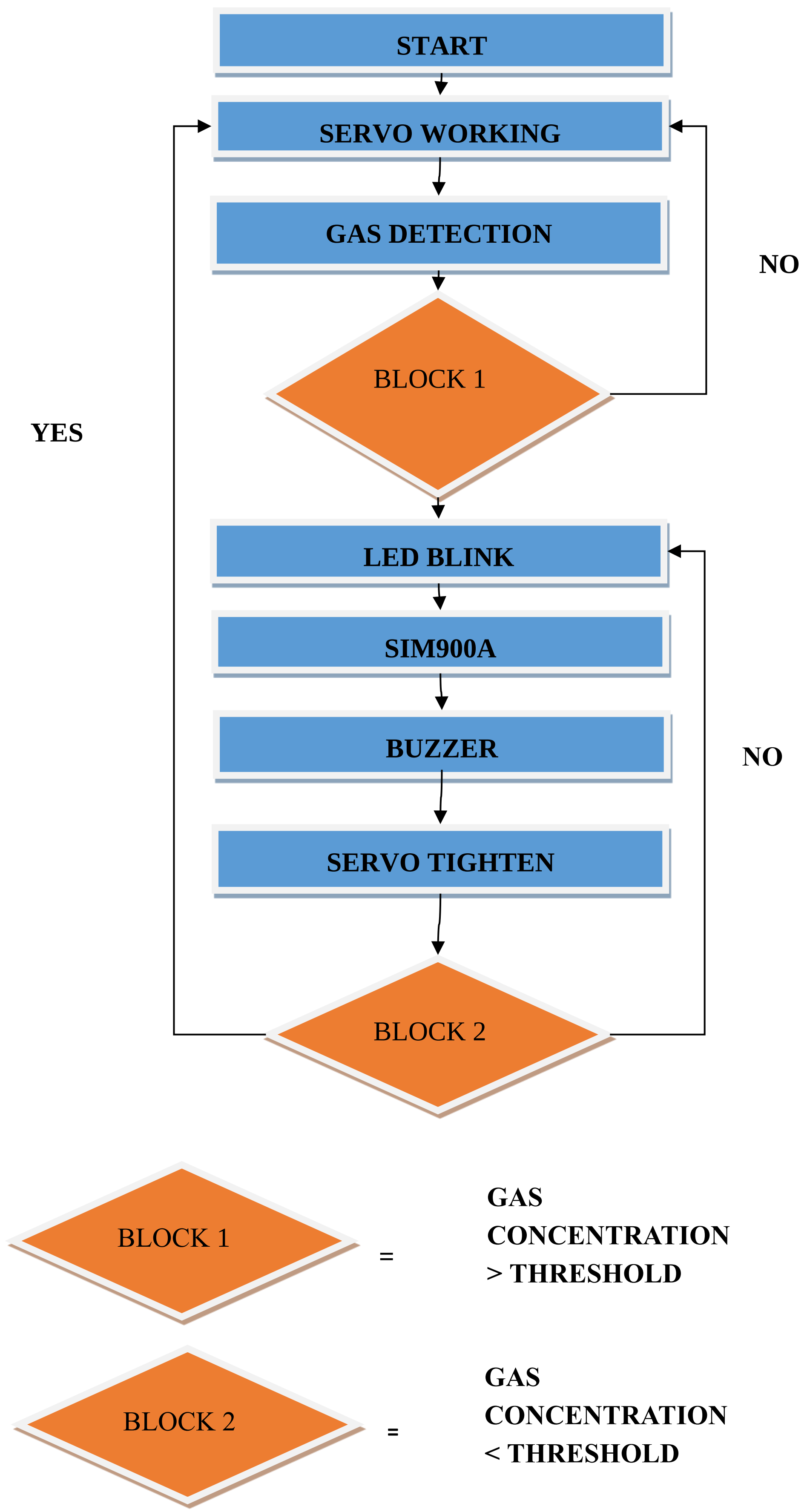

Figure 1.2. The logic flow of the system design

1.2 Arduino

The system has chosen the Arduino as the main processor to integrate all the mentioned hardware components. Arduino is an open-source platform based on beginner-friendly hardware and software, and behaves like the human's brain for the whole system. The required tasks, including read the gas sensor's input signal, lighting and bilking the LED, triggering the buzzer, sending an SMS to the administrator, and controlling the micro-servo, can be done by C/C++ based compiling instructions. Owing to the flexibility and the access to plenty of useful code examples or design experience, Arduino has been one of the best ways for hobbyists or beginners to accomplish system design tasks, not to mention its compatibility including Mac Windows, and Linux [4]. Figure 1.3 shows the Arduino UNO.

Figure 1.3. The picture of Arduino module

As for the software environment, the Arduino Integrated Development Environment (IDE) has been chosen. The Programming interface of Arduino Software (IDE) is shown in Fig. 1.4

mq2_buzzer_servo_mod

```
#include <Servo.h>

Servo servo;
int angle = 10;//initial angle of the servo

int buzzer = 10;//buzzer pin
int smokeA0 = A5;//gas sensor analog pin


int sensorThres = 400;//threshold value

void setup() {
  servo.attach(8);//PWM TO arduino's D8
  servo.write(angle);
  Serial.begin(9600);
}

void loop() {
  int analogSensor = analogRead(smokeA0);

  Serial.print("sensor value is: ");
  Serial.println(analogSensor);//check the sensor value every 0.5s
  delay(500);

  if (analogSensor > sensorThres)// Checks if it has reached the threshold value
  {
```

Figure 1.4. The Programming interface of Arduino Software (IDE)

The Arduino's pin configuration is shown in Figure 1.5. The resource comes from the Arduino's official website at https://www.arduino.cc/en/reference/board.

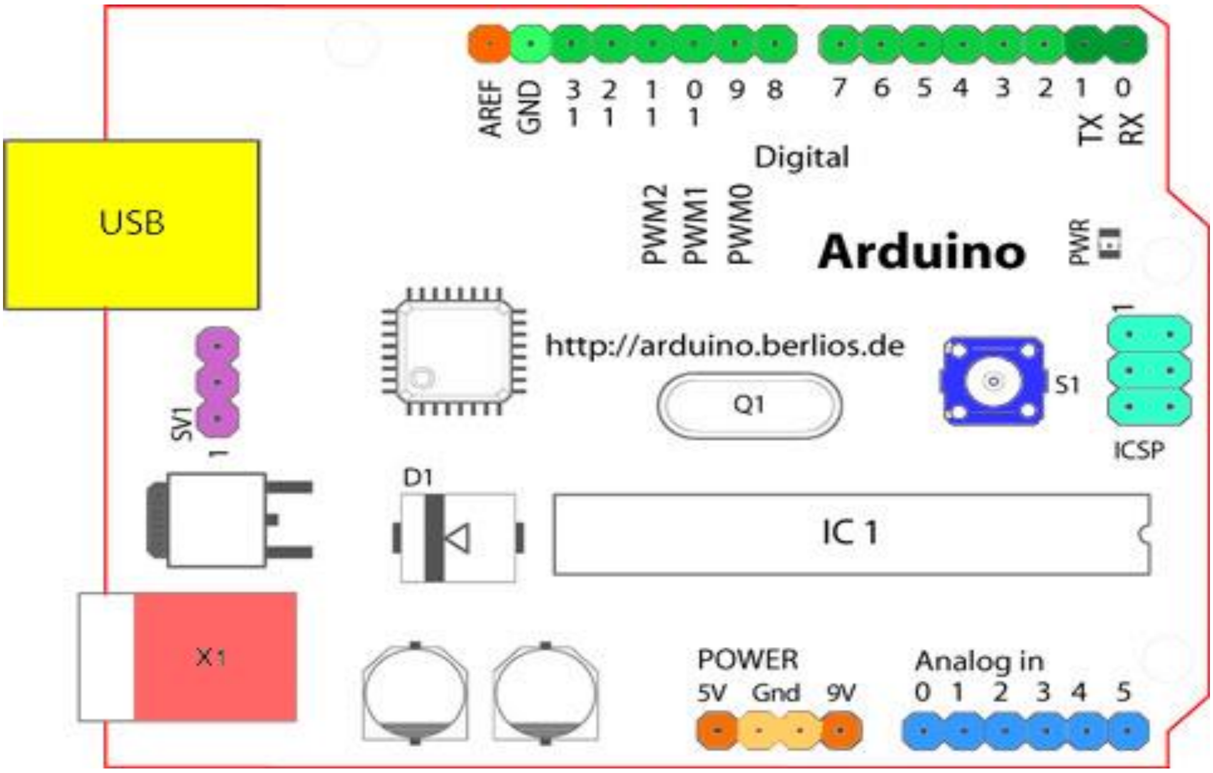

Figure 1.5. Arduino's pin configuration

1.3 Schematic wire connections

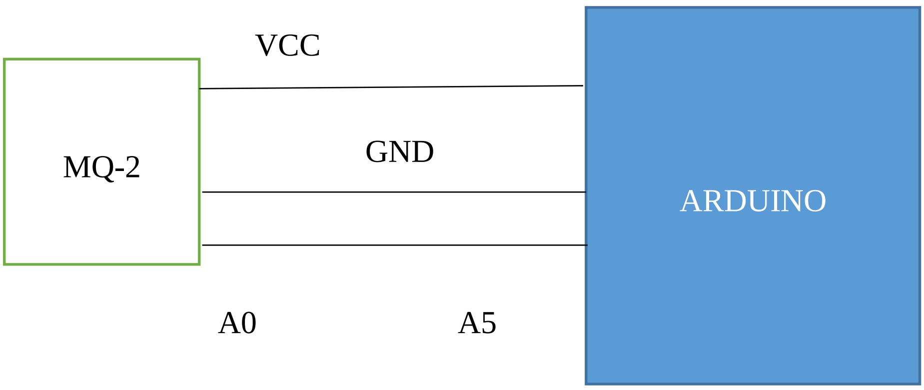

Figure 1.6. The schematic diagram of gas detection's wire connect with Arduino

Figure 1.6 shows the wire connection between Arduino and the gas detector MQ2-2.

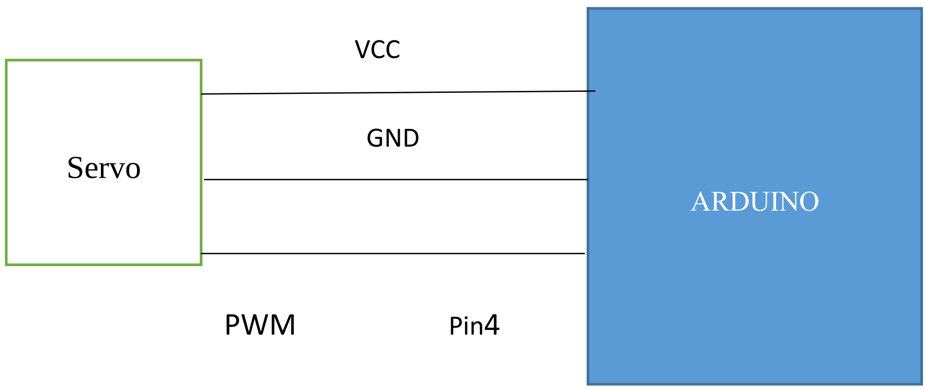

Figure 1.7. The schematic diagram of the feedback part’s wire connection

Figure 1.7 shows the wire connection between the Micro-Servo and Arduino.

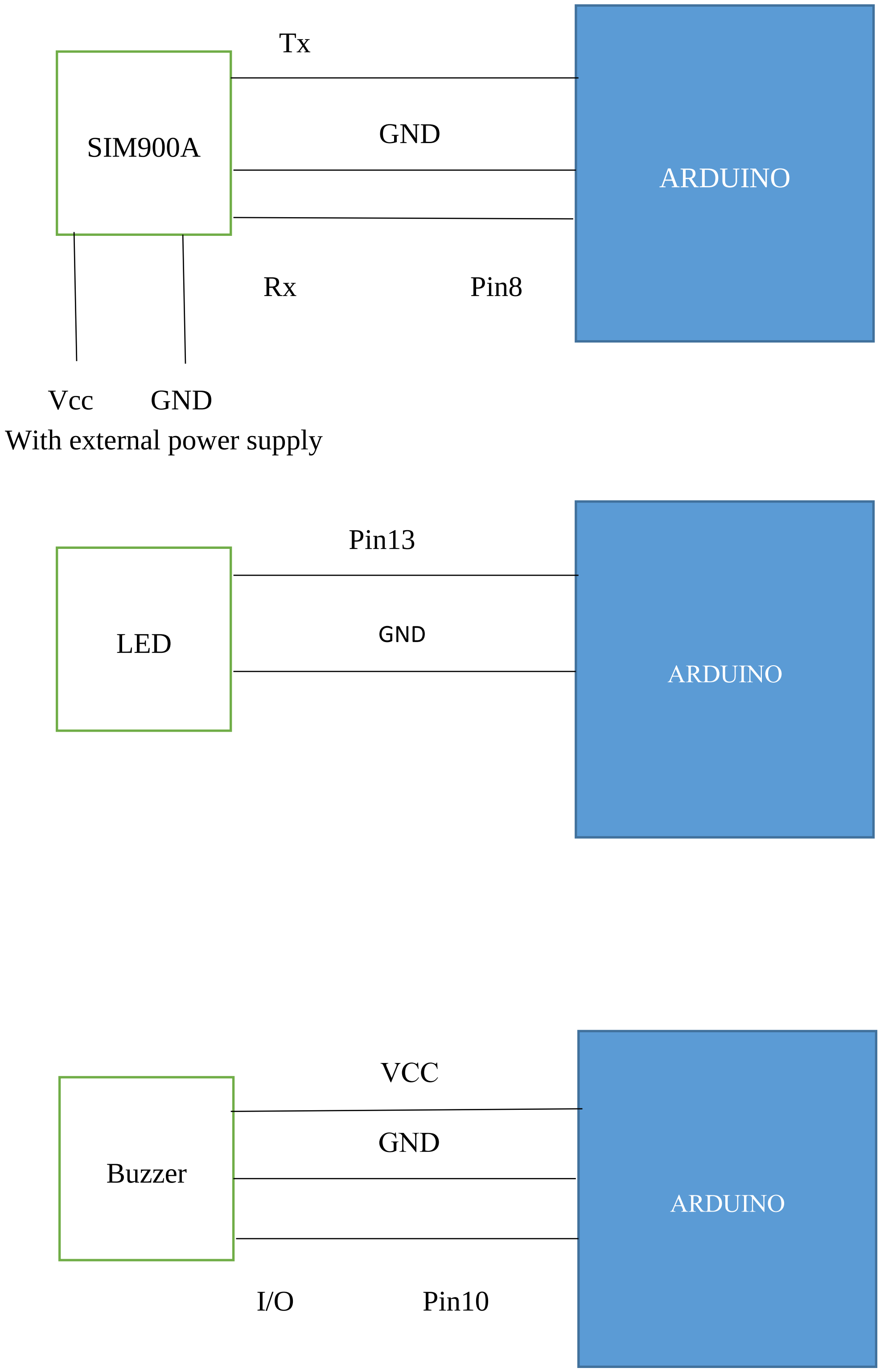

Figure 1.8. The schematic diagram of the wire connections between Arduino and the alarming part of SIM900A, LED, and Buzzer (from top to bottom)

1.4 Welding

To be honest, welding has been under consideration after the interim report. However, the unexpected coronavirus COVID-19 has brought difficulty to it. Although most of the hardware components were kept, welding in the lab room is impossible now. As a result, the direct wire connection has been utilized to finish the final year project, which results in short-circuit and open-circuit in circuit construction, and more importantly, messy overall looking. Finally, this part has to be cancelled.

2 Hardware level

2.1 The detection part: MQ-2 gas sensor

The widely utilized sensitive material of many gas sensors is tin oxide (SnO_2) for its low cost, quick response, and high sensitivity towards target gas [5]. While the target combustible gas, including Liquid Petrol Gas, Propane and Hydrogen, Methane, and other combustible steam, will lead to higher conductivity under the same concentration, compared with clean gas. Also, the increase in the combustible gas concentration will bring higher conductivity of the SnO_2. In other words, the SnO_2's conductivity correlates with the combustible gas concentration positively [6].

Understandings for the conductivity changes are explained below. Under the clean air condition, the oxygen density is higher, in contrast to the combustible gas condition. Obviously, SnO_2's surface oxygen density shares the same result mentioned above after heating at high temperatures. Therefore, oxygen adsorbed on the surface of the chosen SnO_2 material can occupy more donor electrons, and prevent current flow consequently. While Under the combustible gas condition, lower oxygen density in both the nearby atmosphere and sensor material's surface, the reverse phenomenon can happen. Less donor electrons are occupied, which allows the current to flow freely through the sensor and higher conductivity.

Conductivity changes will be converted to correspond output signal of gas concentration. In hardware design, the above analysis will be shown in the analog output voltage, originated from the MQ-2 gas sensor in Figure 2.1, whose value is positively proportional to the combustible gas concentration.

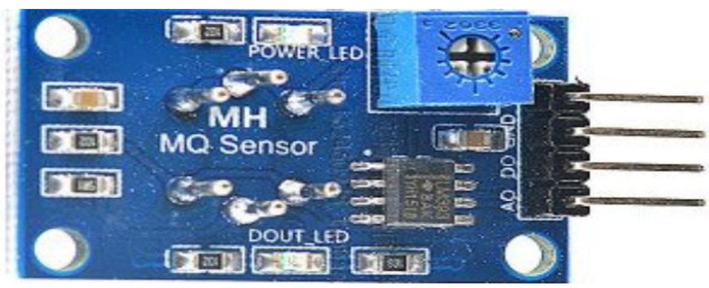

Figure 2.1. Picture of gas sensor MQ-2 and the screw to tune the sensitivity

Then the threshold voltage calibration and following combustible gas concentration check will be discussed below. As shown in Figure 2.1, the blue zone is the target. Turn the screw clockwise can increase its sensitivity and vice versa. The MQ-2 gas sensor' pin picture is shown in Figure 2.2. To calibrate the MQ-2 gas sensor, expose it to the combustible gas the user wants to detect, turn the screw until the LED in the Figure 2.2 below is glowing. The threshold voltage calibration is the preparation for the internal comparator's continuous check whether the combustible gas concentration has reached the predetermined threshold value. When it exceeds the threshold value, the digital output will output logical 1.

Figure 2.2. Gas sensor MQ-2's pin

As Figure 2.2 shows, there are four pins in MQ-2, VCC, GND, DO, AO, respectively.

VCC is the power supplier. It can be connected to Arduino 's 5V pin.

GND is the ground. It can be connected to Arduino 's GND pin.

DO is the digital output of the existence of combustible gas. It is connected to Arduino's A5, but this is not the unique pin connection, which is dependent on the written code.

AO is the analog output of the target combustible gas concentration.

Then the pin recognition and wire connection are completed.

The working condition of the MQ-2 gas sensor is shown in Table 2.1

Table 2.1. The working condition of the MQ-2 gas sensor

Detection Gas	Combustible gas and smoke
Concentration	300-10000 ppm (Combustible gas)
Load Resistance R_L	Adjustable
Heater Resistance R_H	31 Ω ± 3 Ω（Room Temperature.）
Heater consumption P_H	⩽ 900 mW

Why MQ-2 gas sensor was selected, table 2.2 shows its advantages.

Table 2.2. The advantages of the MQ-2 gas sensor

Advantages
Cheap- low cost
Compatible- good sensitivity to a wide variety of combustible gas
Sensitive- high sensitivity to the target LPG, Propane and Hydrogen
Simple- clear drive circuit and easy pin connection

Although the sensor is sensitive to multiple gasses, it cannot tell which it is. As a result, it is better for measuring changes in the known gas density, instead of detecting which is changing.

2.2 The alarming part

2.2.1 LED

The LED, short of light-emitting diode, has dominated the market for its price, simplicity, stability, and high efficiency [7]. Besides, the LED is simple enough to prevent electronic beginners from making mistakes. The first encounter with the LED was the Microelectronic System lab, and the task is to display 0-9 numbers accordingly with the help of hexadecimal. Both the project and the Microelectronic System lab share the same knowledge that higher voltage, provided by Arduino's chosen pin, will lighten the LED, and lower voltage will close them, acted as digital 0 in a digital circuit. The LED used in this project is shown in Fig. 2.3.

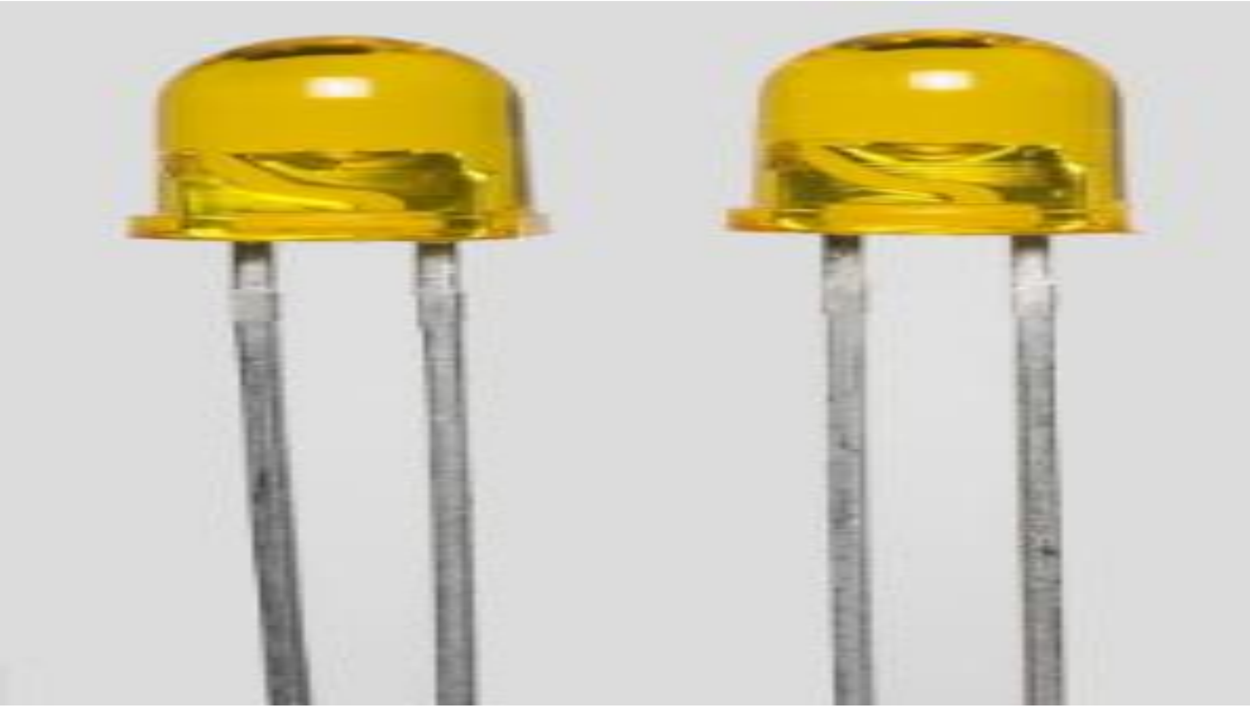

Figure 2.3. The picture of the LED

When it comes to the pin connection for the project, it is very simple. The shorter leg is the cathode, and it should be connected to the Arduino's ground. The anode, the longer leg of the LED, can be connected to any pin you want before compiling. Personally, the pin 13 has been chosen. Then the wire connection is completed.

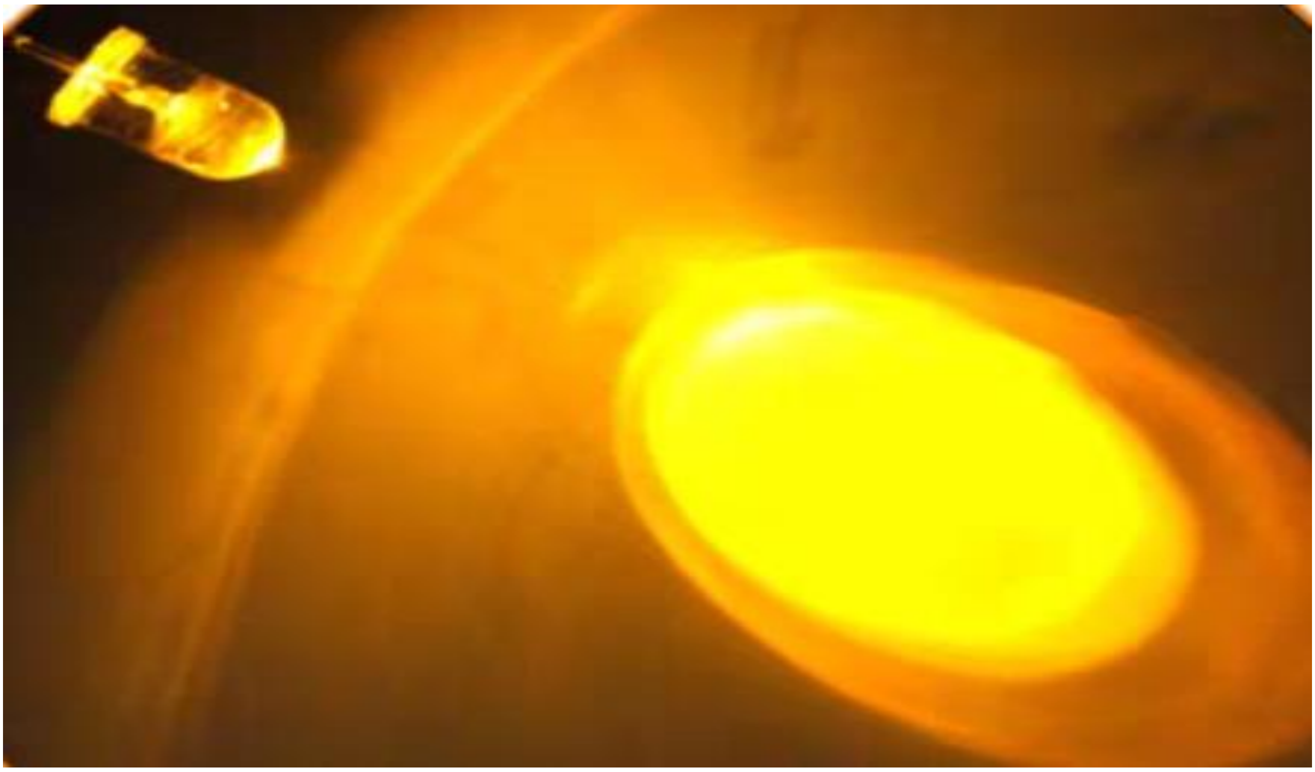

Figure 2.4. The actual effect of the yellow LED

To be honest, green and red LED can be better choices visually, the green for the safe condition, in contrast to the dangerous condition by red. However, there is no red or green LED around except the yellow one shown in Fig. 2.4. As a result, the continuous lighting was set to be the safe condition, while the blink was set to be the dangerous condition. Besides, more LED were in parallel, in order to give a more obvious alert to people.

2.2.2 Buzzer

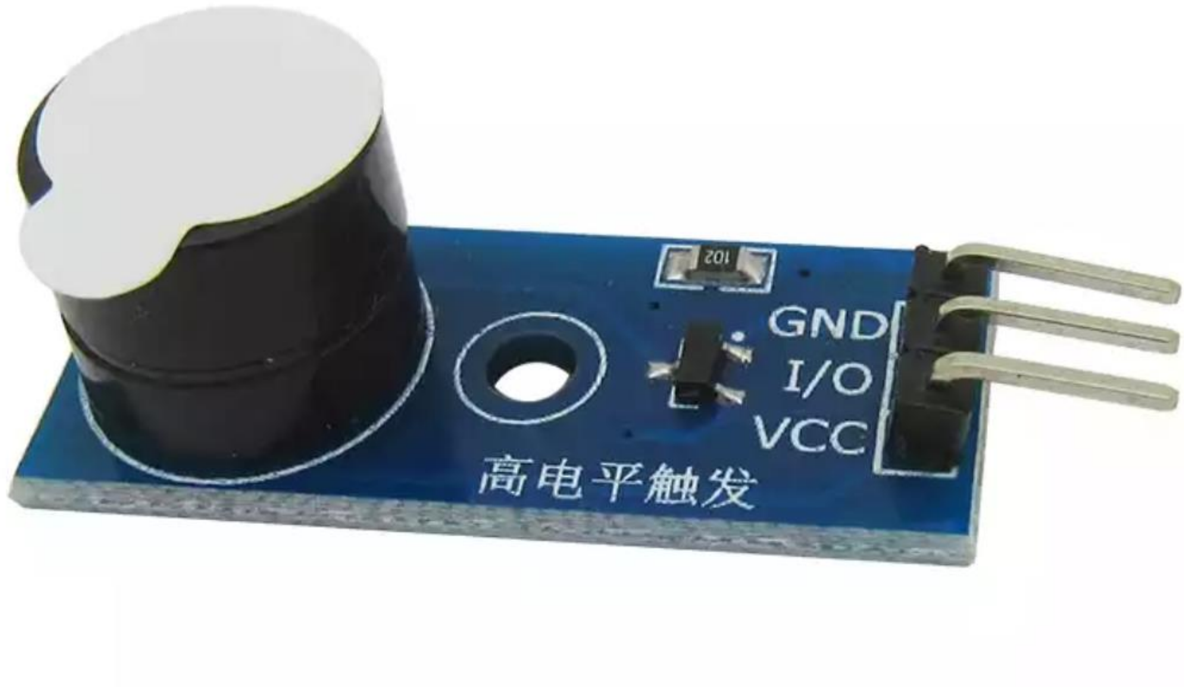

Figure 2.5. The picture of the buzzer module

The buzzer used in this project is shown Fig. 2.5, which is different from the buzzer in the course embedded processor's (EP) lab for its 3 pins, compared with 2 pins the piezo buzzer given in the EP's lab room. However, both of them are passive buzzers, and the frequency value should be given in order to make sound.

The music tone or pitch can be produced when a specific frequency value is given [8], which is shown in Table 2.3.

Table 2.3. Music notes and their corresponding frequency

Notes	Frequency (octaves)
A	440
B flat	466.16
B	493.92
C	523.28
C sharp	554.4
D	587.36

D sharp	622.24
E	659.28
F	698.48
F sharp	740
G	784
A♭ flat	830.64
A	880

The conversion table can easily transfer the music tone into frequency value, which means the music can be displayed in the buzzer form.

Making annoying noise to alert people tends to be easier. Higher frequency will bring a sharper sound, compared with the lower frequency, and the higher frequency can be reached by a longer duration of high voltage in the PWM format.

As for the pin connection, GND and VCC are connected to Arduino’s GND and 5V, respectively. The I/O will be distributed to any pin the user arbitrary set in the Arduino IDE code.

It is pin10 in this design.

2.2.3 SIM900A

The Global System for Mobile communication (GSM) has been a successful technology for years [9]. A report says that the majority (80.8%) of the cellular phone users (3.8 billion) are 2G GSM users [10]. The popularity of GSM-based cell phones contributes to the SMS applications, including portable record and emergency response [11], which is needed in this system.

The Short Message Service (SMS) is the base technology to send and receive different kinds of text messages to and from mobile telephones. The text can be combined with words, numbers, or alphabet letters. Owing to commercial and marketing purposes, the popularity of short message services has increased dramatically worldwide, with 2.38 trillion text messages and a $50 billion market worldwide by 2010 [12].

Figure 2.6. Picture of SIM900A module (without antenna and wire)

The communication hardware components SIM900A is shown in Figure 2.6, which is based on the mentioned technologies of GSM and SMS explained above, is a readily available GSM/GPRS module for various kinds of applications, such as IoT (Internet of Things) and Embedded System. SIM900A's minimum system is a dual-band GSM/GPRS module, and this module communicates through UART or RS232 Interface. In this project, its task is relatively simple. Send an alarming SMS to the administrator outside the working place, to extenuate the coming heavy gas leakage losses.

The module needs a 5V/2A power supply to drive the SIM900A into the working area, but the Arduino's maximum output current is in mA. As a result, the external power supply is needed for its stability and continuous power supply. Individually, the combination of some mobile charger and CH340 chip in Figure 2.7, which converts between USB and TTL, has been the SIM900A's power supply. Another point is the RS232 ports lying in the corner. Before using the module, serial debugging by PC is needed to prove this module works well. As for the LED in SIM900A. If the connection is well, then the LED will blink every 3 seconds. However, blinking every 1 second shows that connection goes wrong.

Figure 2.7. Picture of CH340 power supply module

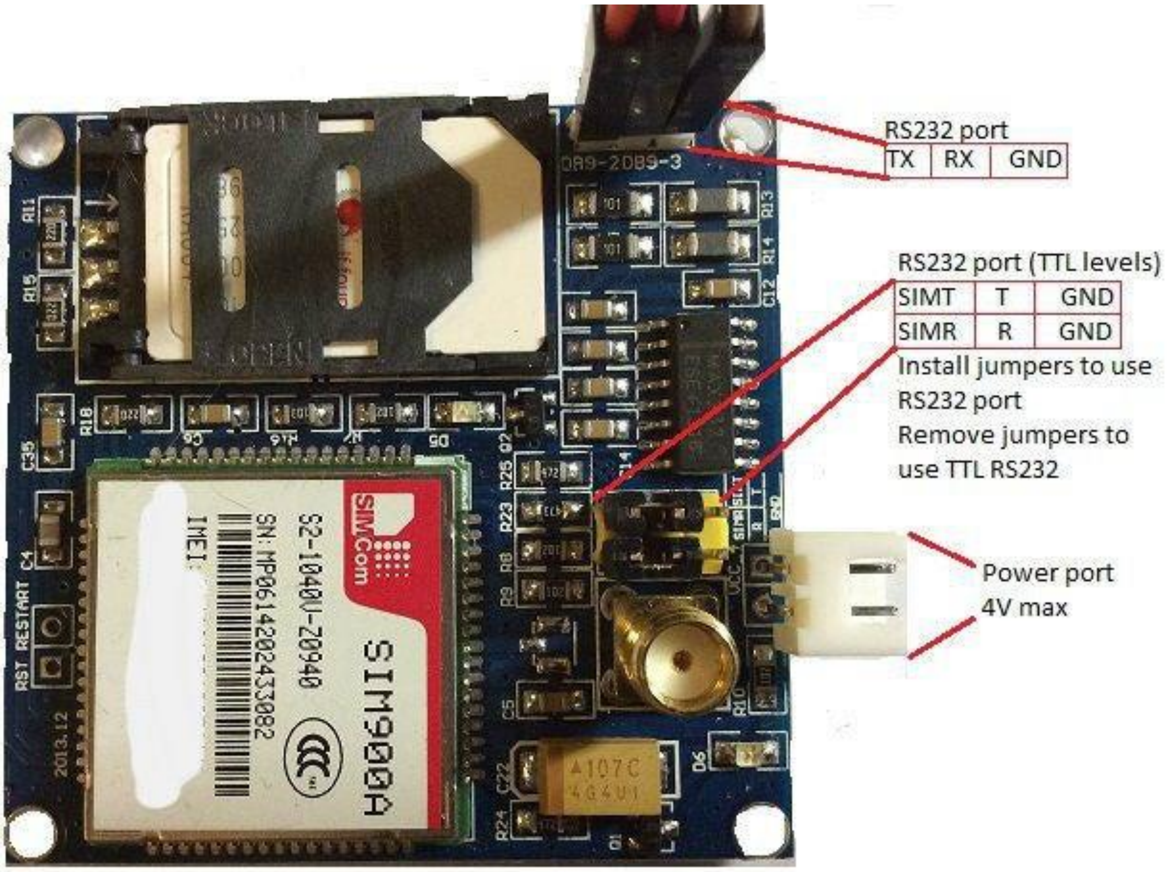

Figure 2.8. Picture of SIM900A module and its pins

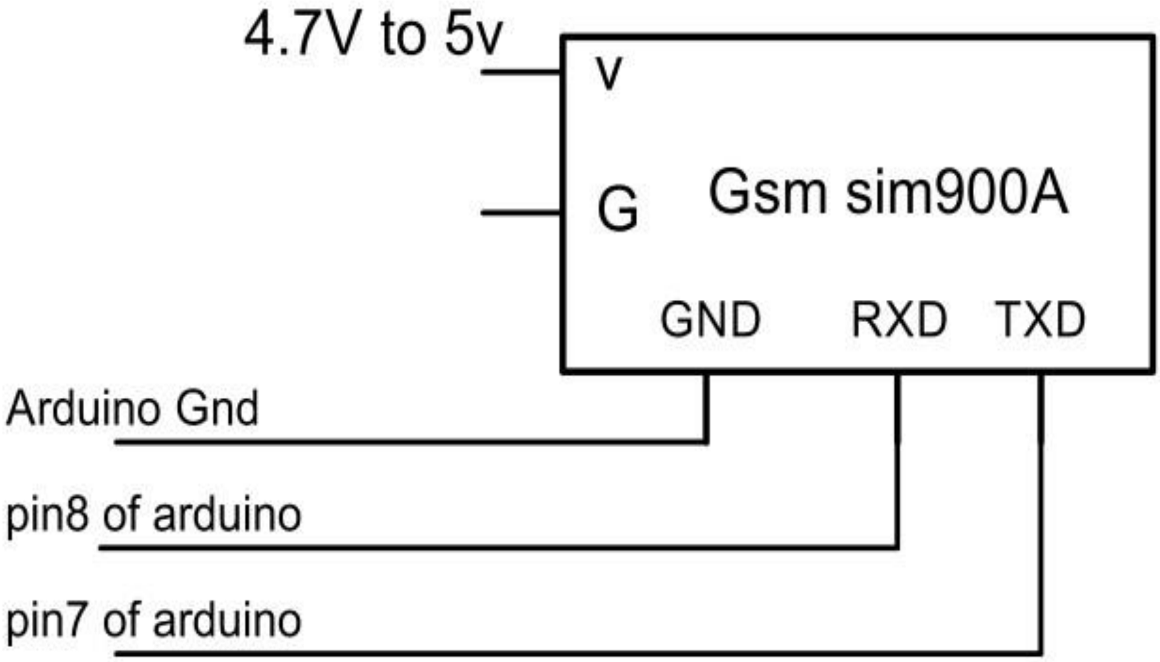

Figure 2.9. SIM900A's pin connection

The SIM900A module and its pins are shown in Fig.2.8.

The SIM900A's pins' connection is shown in In Figure 2.9, there are many pins in SIM900A, but only a few are needed, which includes VCC, GND, TXD 5V (in the VCC side and second row), RXD 5V (in the GND side and second row). VCC and GND are connected to the Arduino's VCC and GND. When it comes to the TXD 5V and RXD 5V, the choice is the Arduino D7 and D8, respectively. Another point is that the antenna is needed, do not forget about that.

2.3 The feedback part

It is familiar working with the micro-servo SG90, encountered in the year 3's TDPS courses. It will be a good choice in the design for its simplicity, high torque, and cheap price [13]. The components in micro-servo SG90 is shown in Fig. 2.10.

Inside the micro servo, pieces from the above image will be found, which includes a DC motor, a controller, and the potentiometer inside, and plastic output shaft outside.

Figure 2.10. Micro-servo SG90 components

The micro-servo is a device that turns an electrical signal into a rotary motion that can be precisely controlled, and the micro servo commonly works by using pulse width modulation (PWM) for control [14]. Owing to the manufacture standardization, almost all the servo shares the same standard, the same PWM pulse sweeping from 0.9ms to 2.1ms periodically, in accordance with 0 degree to 180 degree, under the 20ms period, the same as f=50Hz. The output shaft of the servo can be positioned to specific angular positions with specific PWM input, generated from Arduino's high voltage duration. For example, 0.9ms pulse in PWM means 0° in micro-servo.

However, the actual situation is different from the above theoretical analysis. Due to the manufacture error and other unexpected random errors, the actual angle does not correspond to the expected angle. In reality, the 0.9ms pulse does not correspond to an accurate 0 degree, and it is about 3 degrees instead. As a result, adjustment before using is needed.

Figure 2.11. Schematic picture of the pins of SG90

The Schematic picture of the pins of SG90 is shown in Fig. 2.11, which include VCC, GND, and PWM. Both VCC and GND follow the regular wiring, to Arduino's +5V and GND respectively. The PWM pin can be connected to any Arduino's digital pin, waiting for digital signal. The arbitrary chosen Arduino's digital pin can be connected with the PWM pin. In practice, the answer is pin 4. Then the wire connection is completed.

3 Software compiling

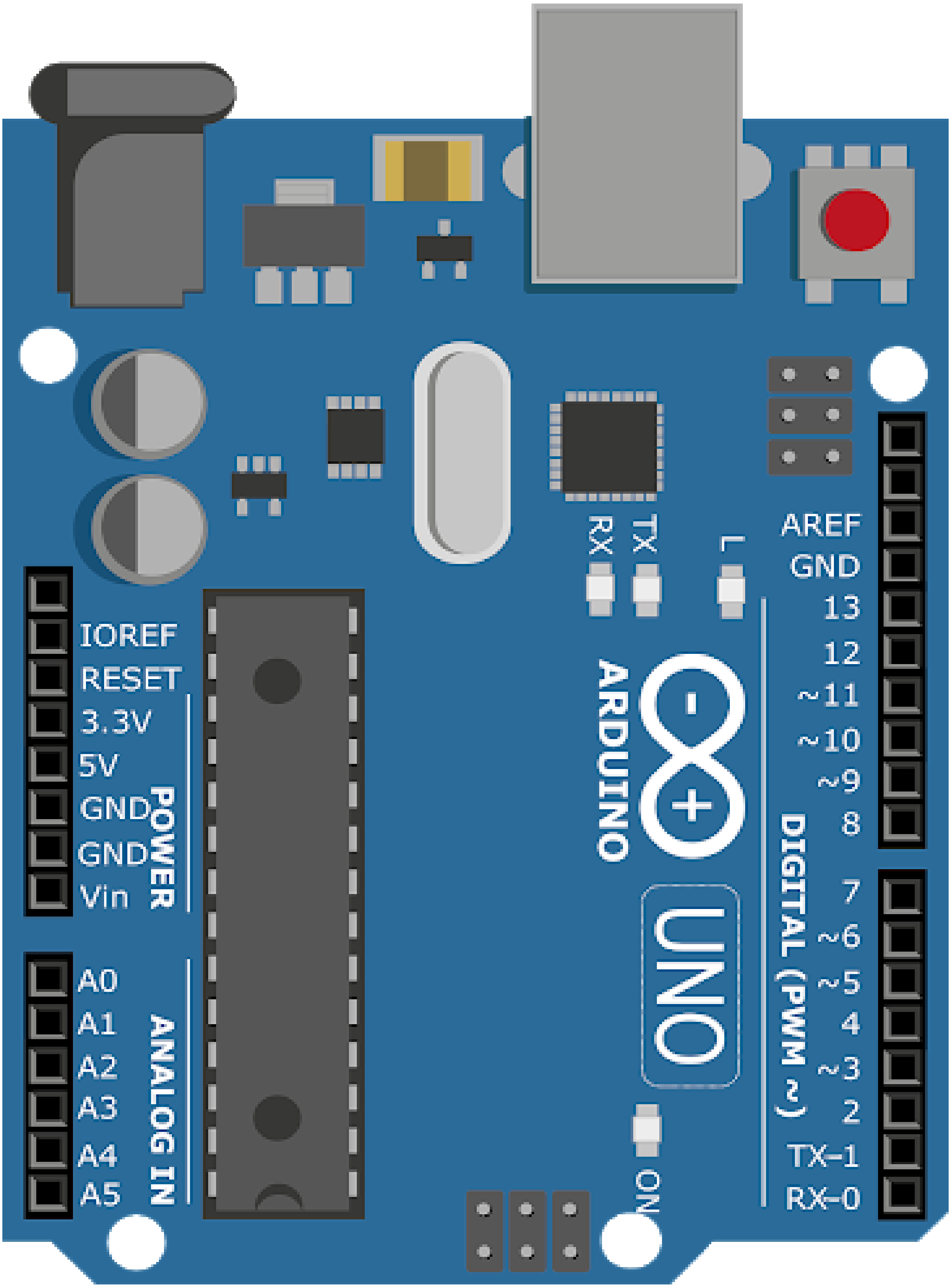

Figure 3.1. The Arduino General-Purpose Input Output (GPIO)

The Arduino Integrated Development Environment (IDE) is a cross-platform and compatible application written in C and C++, and it is used to write and upload programs to Arduino boards in Figure 3.1. In addition, plenty of examples and experiences can be exchanged and modified all over the world in its community and official websites, which helps to accomplish the whole system [15].

The Arduino IDE supplies a software library, which enables this project with many common input and output instructions. The library should be named and cited at the beginning of the whole program, which follows the parameter definition, in most cases, the pin definition. Then, only two basic functions are needed, the start parameter definition in void setup() and the unlimited loop in void loop(). While the actual function block can be extracted from the void loop() in the form of void functionname(). However, it is not necessary at the execution level.

The content in this part is mostly sub-system code. While the overall system code will be given in the appendix.

3.1 The detection part: MQ-2 gas sensor

When it comes to this programming part, the keyword is comparison, which lies between the MQ-2's surface target gas concentration and atmosphere target gas concentration. When the output gas concentration is larger than the threshold, it means the present concentration of combustible gas is beyond the regulated one, then the 'danger' signal will be sent to the buzzer for ring, SIM900A for SMS, LED for blinking respectively. If not, then the system continues its comparison every 0.5 seconds. That is the core of the code, the comparison between the present gas concentration and defined gas concentration.

The simple code can be simplified within 10 rows below in Figure 3.2. Based on the if sentence construction, more functions can be added. For instance, the sensor value can be checked every 0.5 seconds, and it can be examined in the serial monitor in Figure 3.3.

```
if (analogSensor > sensorThres)
{
  tone(buzzer, 1000, 200);
}
else
{
  noTone(buzzer);
}
```

Figure 3.2. Code to tune the tone of the buzzer

```
int analogSensor = analogRead(smokeA0);

Serial.print("sensor value is: ");
Serial.println(analogSensor);//check the sensor value every 0.5s
delay(500);
```

Figure 3.3. Code to write the sensor value into the serial monitor and set the period

3.2 The alarming part

3.2.1 LED

This part may be the simplest one. If the gas leakage concentration cannot exceed the threshold value, then the LED will be on, based on codes from Figure 3.4. However, the LED will blink quickly when the gas leakage happened, and blinking, with the help of Figure 3.5's code, is more visually obvious to people's eyes, compared with the on-state.

To be honest, the green and red LED is a better solution. Due to the COVID-19, the better solution is not available. As a result, whether the gas leakage happens will be distinguished not by colors, but by LED's blinking and shining states.

```
int LED=13;
void setup() {
  pinMode(LED, OUTPUT);
 }

void loop() {
  digitalWrite(LED, HIGH); // TRUN ON THE LED
}
```

Figure 3.4. Code to setup LED's shining state for safe condition

```
int LED=13;

void setup() {
  pinMode(LED, OUTPUT);
}

void loop() {
  digitalWrite(LED, HIGH); // TRUN ON THE LED
  delay(200);// wait for 0.5 second
  digitalWrite(LED, LOW);
  delay(200);// wait for 0.5 second
}
```

Figure 3.5. Code to setup the LED's blinking state for gas leakage condition

3.2.2 Buzzer

This part is simple too. First of all, there are two states for the buzzer. One state is the buzzer keeps silent, which means safe conditions, and no code is needed. The other state is a gas leakage, and the buzzer will make annoying police alarm noise by increasing and decreasing the frequency value step-by-step based on the Arduino board. As a result, there is one block code shown in Figure 3.6. Since the gas leakage judgment comes from the MQ-2 part, place the written buzzer code into the gas leakage part will finish the design.

```
int buzzer = 10;

void setup() {

}

void loop() {
for(int i=100 ; i<2000 ; i++){ // loop for to increase the sound frequency
  tone(buzzer, i);
  delay(10);
}
for(int i=2000 ; i>100 ; i--){// loop for to decrease the sound frequency
  tone(buzzer, i);
  delay(10);
}
}
```

Figure 3.6. The alarm ring code

As for the note and specific frequency value conversion, it helps to play music by this passive buzzer. Although playing a song is invalid in a practical situation, it is interesting during the final year project.

3.2.3 SIM900A

Since the hardware level is completed, the software level will be the important point. Before the next step, an open-source Arduino library SoftwareSerial is needed, and it is available online freely. SoftwareSerial enables serial data communication through other digital pins of Arduino. After unzipping it to the Arduino's libraries, interfacing gsm with Arduino can be taken into consideration.

Since only SMS is required in the design, other functions, such as making a call, redialling, are removed for less bugs and errors. The phone number below is adjustable for any authorized people, and explanations are listed in the code. The code to send the message is shown in Figure 3.7.

```
void SendMessage()
{
  Serial.println ("Sending Message");
  SIM900A.println("AT+CMGF=1");    //Sets the GSM Module in Text Mode
  delay(1000);
  Serial.println ("Set SMS Number");
  SIM900A.println("AT+CMGS=\"+8618608356303\"\r"); //Mobile phone number to send message
  delay(1000);
  Serial.println ("Set SMS Content");
  SIM900A.println("Sms");// Messsage content
  delay(100);
  Serial.println ("Finish");
  SIM900A.println((char)26);// ASCII code of CTRL+Z
  delay(1000);
  Serial.println ("Message has been sent ->");
}
```

Figure 3.7. The code to send message

Figure 3.8 shows the received the gas leakage message ‘Sms’ on the mobile phone. Here it is needed to be noted that the ‘ss’ is the test signal received from the SMS command, which is not needed in the design.

Figure 3.8. The received gas leakage message “Sms” on the Mobile phone

3.3 The feedback part

In a practical situation, the rotating motor shows the system works well, but it will be tightened and cut off when the gas leakage happens. For the rotating motor, it is set to rotate from 10 degrees to 180 degrees back and forth. However, no code is needed to tighten the motor, and cutting off the electricity is more direct. Since the gas leakage judgment comes from the MQ-2 part, inversely, place the written motor code in Figure 3.8 into the safe part.

At the software level, the expectation is the output shaft rotates from one end to another first. Then the code will be divided into two parts, clockwise and counter-clockwise loop. In the counter-clockwise loop, it is set to start from 10° to 180°. While in the clockwise loop, it does the reverse action.

```
#include <Servo.h>

Servo servo;
int angle = 10;

void setup() {
  servo.attach(8);//PWM TO arduino's D8
  servo.write(angle);
}


void loop()
{
 // scan from 0 to 180 degrees
  for(angle = 10; angle < 180; angle++)
  {
    servo.write(angle);
    delay(15);
  }
  // now scan back from 180 to 0 degrees
  for(angle = 180; angle > 10; angle--)
  {
    servo.write(angle);
    delay(15);
  }
}
```

Figure 3.9. SG90's code

4 Analysis and Discussion

4.1 Arduino

Arduino has chosen for various reasons. Friendly and open-source platform and community in terms of both software and hardware for beginners. Cheap hardware board and free IDE support C/C++ compiler, rather than eccentric assembly language. A wide variety of libraries for different purposes towards different applications [16]. In conclusion, a better overall performance can be reached with the help of Arduino [17].

4.2 SIM900A

Standardized AT command is needed to control the spread of SMS via a computer [18]. In this design, AT command jargons are programmed in software level, in order to send the SMS to the administrator [19]. Every AT command includes a result code which specifies the status of the command and a reply containing the data returned by the modem [20]. Part of the AT command [21] jargon is shown in Table 4.1.

Table 4.1. Part of the AT command for SIM900A

Command	Description
AT+CMGD	DELECT SMS
AT+CMGR	READ SMS
AT+CMGS	SEND SMS
AT+CMGF	SET SMS FORMAT

4.3 MQ-2

Although gas sensors play an important role in the system design, their selectivity has decreased the overall performance. This means the sensor can detect the gas leakage, but it cannot tell which it is. As a result, sensors are not the unique solution to the system construction, especially in the face of advanced technologies.

For instance, with the help of a decoupling algorithm and a pairing plot scheme, a support vector machine (SVM) is applied for gas classification effectively and accurately [22]. In addition, wireless gas sensor network (WGSN) can also be an alternative way, because it can be easy to install, modify, and maintain with a low price in the noisy industrial zone [23].

5 Conclusions and further work

5.1 Conclusions

The main objective of the project is combustible gas detection and leakage prevention, and thus ensure people's safety. As for detection, MQ-2 will compare the predetermined threshold value with the atmosphere gas detection, and then send the digital result to the Arduino. In the following step, if the result is gas leakage, LED blinks every 0.5 seconds and the passive buzzer makes annoying noise, ranging from 100 Hz to 2000 Hz, step-by-step. Additionally, since some people cannot receive the buzzer and the LED warning, an SMS will be sent by SIM900A to extenuate the coming losses. The micro-servo will also be tightened to prevent potential damage. However, if the gas leakage does not happen, the micro servo will be in the simulated working mode, rotating back and forth between 10 degrees and 180 degrees.

When it concludes, the following points will be discussed. To begin with, the design can give direct alert to people nearby, by visually and audibly visible alarm information based on LED and buzzer. Next, the alarm can be sent to administrators or users outside by an SMS based on the GSM, sent by the module SIM900A. In addition, the negative feedback loop, originated from the control theory, is introduced to extenuate the losses. When the gas leakage happened, the micro-servo will be tightened and closed to avoid potential risks until the gas concentration returns to safe region value, instead of doing nothing. Last but not least, the overall system construction cost is affordable for both families and factories. Most of the hardware components are pipeline products, cheap and mature. Modifications in hardware and software are available too for individual use.

5.2 Suggestions for further work

The project has successfully achieved the goals, gas leakage detection, alarming, prevention accordingly. Sensitive gas leakage detection, immediate gas leakage alarming and quickly gas leakage prevention responds effectively illustrate the high performance of the whole system. Besides, modifications are easy to be made based on the built structure for domestic or industrial application. However, further work is needed to improve the system.

The microprocessor is Arduino, which is a powerful and multifunctional board, but only a few ports and pins are utilized in this project. More challenging functions can be taken into consideration. For instance, output the real-time gas concentration to a big LED board, instead of simple LED lights, and hexadecimal is introduced to the system design to control the LED board naturally. In addition, the microprocessor competence should be considered. It can be Arduino used, but how about a shift from Arduino to other known or unknown microprocessors like STM32?

As for the communication hardware SIM900A, the fixed cost is certain, but the SIM card's cost is not. There are two ways for the SMS bills. One can be monthly debt, and the other one is calculated by the number of SMS sent. When the bug happened, there might be endless SMS sent to the administrator, which leads to a high cost. Monthly debt is also a waste of money and resources for users.

As a result, the solution is beyond the module's range. The fixed debt is determined by the SIM card enterprise, which is not controlled at the engineering level. Another engineering level way is the Internet, but the SIM900A module is not compatible with that. The usage of the Internet communication means another communication hardware module should be taken into consideration.

References

[1] D. B. Lata, A. Misra, and S. Medhekar, "Effect of hydrogen and LPG addition on the efficiency and emissions of a dual fuel diesel engine," *International Journal of Hydrogen Energy,* vol. 37, no. 7, pp. 6084-6096, 2012.

[2] Q. Zhang, W. Li, and D.-C. Lin, "Failure Analysis of Physical Explosion due to Gas Jetting from High-Pressure Pipeline," *Journal of Failure Analysis and Prevention,* vol. 12, no. 2, pp. 181-189, 2012.

[3] N. J. Leigh, P. L. Tran, R. J. O'Connor, and M. L. Goniewicz, "Cytotoxic effects of heated tobacco products (HTP) on human bronchial epithelial cells," *Tobacco Control,* vol. 27, no. Suppl 1, pp. s26-s29, 2018.

[4] J. C. Martínez-Santos, O. Acevedo-Patino, and S. H. Contreras-Ortiz, "Influence of Arduino on the Development of Advanced Microcontrollers Courses," *IEEE Revista Iberoamericana de Tecnologias del Aprendizaje,* vol. 12, no. 4, pp. 208-217, 2017.

[5] P. A. Russo *et al.*, "Room‐Temperature Hydrogen Sensing with Heteronanostructures Based on Reduced Graphene Oxide and Tin Oxide," *Angewandte Chemie International Edition,* vol. 51, no. 44, pp. 11053-11057, 2012.

[6] J. Zhang, X. Liu, G. Neri, and N. Pinna, "Nanostructured Materials for Room‐Temperature Gas Sensors," *Advanced Materials,* vol. 28, no. 5, pp. 795-831, 2016.

[7] J. He, S. Yang, K. Zheng, Y. Zhang, J. Song, and J. Qu, "One-pot synthesis of dispersible thermally stable organic downconversion materials under DBU catalyzation for high performance hybrid-LED lamps," *Green Chemistry,* vol. 20, no. 15, pp. 3557-3565, 2018.

[8] S. J. McFarlane, J. E. Garcia, D. S. Verhagen, and A. G. Dyer, "Alarm tones, music and their elements: Analysis of reported waking sounds to counteract sleep inertia," *PloS one,* vol. 15, no. 1, p. e0215788, 2020.

[9] J. Harris, "GSM basics: an introduction," *Microwave Journal,* vol. 39, no. 10, p. 84, 1996.

[10] A. Fehske, G. Fettweis, J. Malmodin, and G. Biczok, "The global footprint of mobile communications: The ecological and economic perspective," *IEEE Communications Magazine,* vol. 49, no. 8, pp. 55-62, 2011.

[11] C.-L. Tseng *et al.*, "Feasibility study on application of GSM–SMS technology to field data acquisition," *Computers and Electronics in Agriculture,* vol. 53, no. 1, pp. 45-59, 2006.

[12] S. Okazaki and C. R. Taylor, "What is SMS advertising and why do multinationals adopt it? Answers from an empirical study in European markets," *Journal of Business Research,* vol. 61, no. 1, pp. 4-12, 2008.

[13] Q. Zhang, L. Yan, Z. Duan, Z. Jiao, C. Gerada, and I. M. Chen, "High Torque Density Torque Motor With Hybrid Magnetization Pole Arrays for Jet Pipe Servo Valve," *IEEE Transactions on Industrial Electronics,* vol. 67, no. 3, pp. 2133-2142, 2020.

[14] M. Taghizadeh, A. Ghaffari, and F. Najafi, "Improving dynamic performances of PWM-driven servo-pneumatic systems via a novel pneumatic circuit," *ISA Transactions,* vol. 48, no. 4, pp. 512-518, 2009.

[15] Arduino-Home. [Online]. Available: https://www.arduino.cc/.

[16] A. Forsiuk, O. Pylypenko, A. Golub, Y. Zasiadko, V. Voznyy, and R. Gryshchenko, "Advisability use of Arduino controllers in automation of refrigeration devices," *Refrigeration Engineering and Technology*, vol. 55, no. 1, pp. 54-59, 2019.

[17] Y. Wang, "Design and Implementation of a Wireless Sensor Network Node Based on Arduino," *International Journal of Online Engineering (iJOE)*, vol. 13, no. 11, p. 128, 2017.

[18] S. Vougioukas and M. Roumeliotis, *A system for basic-level network fault management based on the GSM short message service (SMS)*. 2001, pp. 218-222 vol.1.

[19] A. A. Uddin, K. M. Shahriar, A. L. I. Mohammed, and R. M. Mahbubur, "Controlling and Monitoring of Electric Feeders Using GSM Network Technology," *Journal of Electrical and Electronics Engineering*, vol. 6, no. 2, pp. 5-8, 2013.

[20] F. Givehki and A. Nicknafs, "Mobile control and management of computer networks using SMS services," *Telematics and Informatics*, vol. 27, no. 3, pp. 341-349, 2010.

[21] SIMCOM, "SIM900A_AT Command Manual_ V1.03."

[22] K.-W. Jang, J.-H. Choi, J.-H. Jeon, and H.-S. Kim, "Combustible Gas Classification Modeling using Support Vector Machine and Pairing Plot Scheme," *Sensors (Basel, Switzerland)*, vol. 19, no. 22, p. 5018, 2019.

[23] F. Aliyu and T. Sheltami, "Development of an energy-harvesting toxic and combustible gas sensor for oil and gas industries," *Sensors & Actuators: B. Chemical*, vol. 231, pp. 265-275, 2016.

A Appendices (codes for the project)

```
#include <Servo.h>
Servo servo;
#include <SoftwareSerial.h>
SoftwareSerial SIM900A(7,8);

int angle = 10;//initial angle of the servo
int LED=13;// LED pin
int buzzer = 10;//buzzer pin
int smokeA0 = A5;//gas sensor analog pin
int sensorThres = 400;//threshold value

void setup() {
  servo.attach(4);//PWM TO arduino's D4
  servo.write(angle);
  pinMode(LED, OUTPUT);//LED
  SIM900A.begin(9600);   // Setting the baud rate of GSM Module
  Serial.begin(9600);    // Setting the baud rate of Serial Monitor (Arduino)
  Serial.println ("SIM900A Ready");
  delay(100);
}

void loop() {
  int analogSensor = analogRead(smokeA0);
  Serial.print("sensor value is: ");
  Serial.println(analogSensor);//check the sensor value every 0.5s
  delay(500);

  if (analogSensor > sensorThres)// Checks if it has reached the threshold value
```

```
  {
  Buzzer();
  LEDblink();
  SendMessage();
  }

  else
  {
    microservo();
    noTone(buzzer);
    LEDshine();
  }
  delay(100);
  }



void Buzzer(){
  for(int i=100 ; i<2000 ; i++){ // loop for to increase the sound frequency
  tone(buzzer, i);
  delay(10);
}
for(int i=2000 ; i>100 ; i--){// loop for to decrease the sound frequency
  tone(buzzer, i);
  delay(10);
}
  }

void LEDblink(){
  digitalWrite(LED, HIGH); // TRUN ON THE LED
```

```
  delay(200);// wait for 0.5 second
  digitalWrite(LED, LOW);
  delay(200);// wait for 0.5 second
  }


void SendMessage(){
  Serial.println ("Sending Message");
  SIM900A.println("AT+CMGF=1");    //Sets the GSM Module in Text Mode
  delay(1000);
  Serial.println ("Set SMS Number");
  SIM900A.println("AT+CMGS=\"+8618608356303\"\r");  //Mobile phone number to send message
  delay(1000);
  Serial.println ("Set SMS Content");
  SIM900A.println("Sms");// Messsage content
  delay(100);
  Serial.println ("Finish");
  SIM900A.println((char)26);// ASCII code of CTRL+Z
  delay(1000);
  Serial.println ("Message has been sent ->");
  }


void microservo(){
  // scan from 0 to 180 degrees
  for(angle = 10; angle < 180; angle++)
  {
    servo.write(angle);
```

```
  delay(15);
 }
 // now scan back from 180 to 0 degrees
 for(angle = 180; angle > 10; angle--)
 {
  servo.write(angle);
  delay(15);
 }
 }

void LEDshine(){
 digitalWrite(LED, HIGH); // TRUN ON THE LED
 }
```

B University's Plagiarism Statement

This statement is reviewed annually and the definitive statement is in the University Calendar.

The University's degrees and other academic awards are given in recognition of a student's personal achievement. All work submitted by students for assessment is accepted on the understanding that it is the student's own effort.

Plagiarism is defined as the submission or presentation of work, in any form, which is not one's own, without acknowledgement of the sources. Plagiarism includes inappropriate collaboration with others. Special cases of plagiarism can arise from a student using his or her own previous work (termed auto-plagiarism or self-plagiarism). Auto plagiarism includes using work that has already been submitted for assessment at this University or for any other academic award.

The incorporation of material without formal and proper acknowledgement (even with no deliberate intent to cheat) can constitute plagiarism. Work may be considered to be plagiarised if it consists of:

- a direct quotation;
- a close paraphrase;
- an unacknowledged summary of a source;
- direct copying or transcription.

With regard to essays, reports and dissertations, the rule is: if information or ideas are obtained from any source, that source must be acknowledged according to the appropriate convention in that discipline; and any direct quotation must be placed in quotation marks and the source cited immediately. Any failure to acknowledge adequately or to cite properly other sources in submitted work is plagiarism. Under examination conditions, material learnt by rote or close paraphrase will be expected to follow the usual rules of reference citation otherwise it will be considered as plagiarism. Departments should provide guidance on other appropriate use of references in examination conditions.

Plagiarism is considered to be an act of fraudulence and an offence against University discipline. Alleged plagiarism, at whatever stage of a student's studies, whether before or after graduation, will be investigated and dealt with appropriately by the University.

The University reserves the right to use plagiarism detection systems, which may be externally based, in the interests of improving academic standards when assessing student work.