HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY

OFFICE FOR INTERNATIONAL STUDY PROGRAMS

PROJECT:

THE DESIGN OF MITSUBISHI XPANDER OXYGEN SENSOR VOLTAGE SIGNAL SIMULATOR

Student's name: Lê Anh Đức

Student ID: 1852327

Class: CC18OTO

Instructor: PhD. Trần Đăng Long

Ho Chi Minh City ,2021

VIETNAM NATIONAL UNIVERSITY OF HO CHI MINH CITY
HO CHI MINH UNIVERSITY OF TECHNOLOGY
Faculty of Transportation Engineering
Department of Automotive Engineering

SOCIALIST REPUBLIC OF VIETNAM
Independence – Freedom – Happiness

THESIS MISSION

1. **Student's name:** Lê Anh Đức **- Student ID:** 1852327
2. **Major:** Automotive Engineering **- Class:** CC18OTO
3. **Thesis title:** The design of Mitsubishi Xpander oxygen sensor voltage signal simulator.
4. **Content:**
   - ❖ General information about Mitsubishi Xpander oxygen sensor signals:
   - ❖ Signals working principle.
   - ❖ Actual measurement of voltage signals on Mitsubishi Xpander.
   - ❖ Identify the problem to be solved.
   - ❖ General layout design:
     - Design the structure diagram.
     - Design the principle diagram.
   - ❖ Technical design:
     - Select the components for the product.
     - Design the timing diagram.
     - Design algorithm flowcharts.
     - Arduino programming:
       - ➢ Timer control.
       - ➢ PWM control.
       - ➢ Analog to Digital Converter control.
   - ❖ Result:
     - Simulate the working process of the system using Proteus software.
5. **Product:**

✓ Presentation report.	✓ Poster.
✓ Microcontroller program.	✓ Proteus simulation.

6. **Assigned day:** April, 2021.
7. **Finished day:** June, 2021.

The content and requirements of the thesis is already approved by the Head of Department of Automotive Engineering.

HCMC, day……. month…… year 2021 *HCMC, day……. month…… year 2021*

Head of Department **Instructor**

I. Introduction: 4
1.1 Objectives of project: 4
1.2 Scope of implementation 4
1.3 Working conditions and technical requirements: 4
1.3.1 Working conditions: 4
1.3.2 Technical Requirements: 5
II. Theoretical basis: 5
2.1 Oxygen sensor 5
2.2 Pulse Width Modulation (PWM) 6
III. General layout design: 7
3.1 General layout diagram 7
3.2 Working principles of the system: 7
3.2.1 Input and output signal 7
3.2.2 Input and output relationship 8
3.2.3 Timing diagram 13
IV. Technical design: 8
4.1 Choosing Components and microcontroller: 8
4.1.1 Choosing oxygen sensor: 8
4.1.2 Choosing electronics board: 9
4.1.3 Choosing low-pass filter 10
4.2 Algorithms 11
4.3 Code and Explanation 13
4.3.1 CODE: 13
4.3.2 Explanation: 15
V. Proteus simulation: 16
5.1 Electrical diagram 16
5.2 Diagram function 17
5.3 Simulating on proteus and in reality 17
17
VI. Conclusion: 18
VII. REFERENCE 19

I. Introduction:

1.1 Objectives of project:

Nowadays, with the development of electrics – electronics technology in automotive field, we can see many sensors installed in a car. It not only helps driver to make sure the car in normal but also help us to solve the problem of the car easily. One of the most important car sensors is oxygen which is directly relevant to the environment pollution and it is also the scope of this project.

1.2 Scope of implementation

This project aim is to simulate or testing design for deeply understanding working principle, operating of oxygen sensor in Mitsubishi Xpander and can as nearly reality as possible. Therefore, we decide to simulate this system on proteus software which has many components operate, have functions as practical system. The design will create two voltage signals OS1 and OS2 like the signals received from oxygen sensor located at before and after three-way catalyst converter of Mitsubishi Xpander by changing potentiometer.

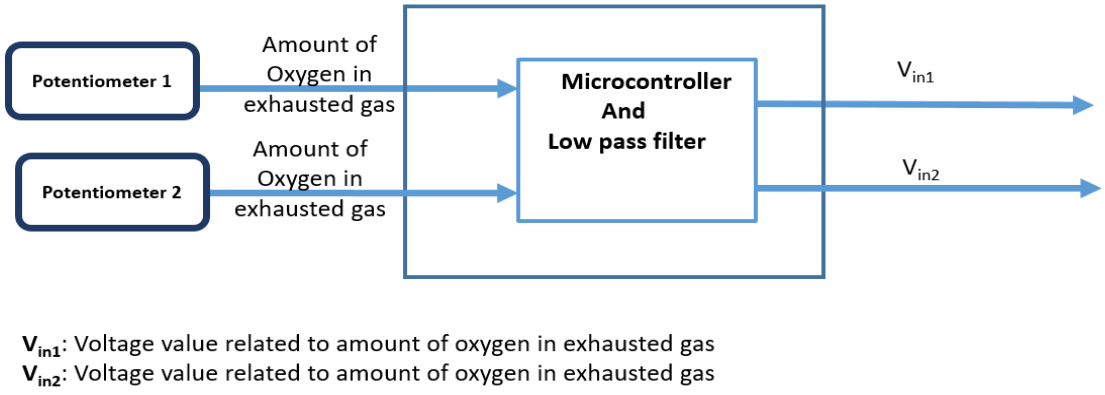

Figure 1.1 *Pictorial diagram showing the principle of the Mitsubishi Xpander oxygen sensor signal*

1.3 Working conditions and technical requirements:

1.3.1 Working conditions:

- Working in an extreme environment.
- Temperature is very high (from 315 to 343 Celsius degree).
- Under many continuously chemical reactions.

- Can be mechanical damage when operating.

1.3.2 Technical Requirements:

- Working normally in above conditions.
- Setup low pass filter to working in top condition.
- Read and process the input voltage.
- Calculate precisely the PWM.

II. Theoretical basis:

2.1 Oxygen sensor

- Oxygen sensor or lambda sensor is located within the emissions control system. The oxygen sensor will send data to the management computer located within the engine and ensure that the engine is running at top condition. The contact of the exhaust gas and the sensor probe causes the sensor to generate an electric current with a potential inversely proportional to the amount of oxygen of the exhaust gas. If the amount of oxygen is high or the mixture is "lean", the voltage will be around 0.1 volt and 0.9 volt if the mixture is "rich". The fuel injector will be regulated appropriately to ensure the ratio is close to ideal ratio (14.7: 1).
- Oxygen sensor itself produces a voltage when they got hot (about 600*F), on the tip of the oxygen sensor that plugs into the exhaust manifold is a zirconium ceramic bulb. The inside and outside of the bulb is coated with a porous layer of platinum, which serve as the electrodes. The differences in oxygen levels between the bulb and the outside atmosphere within the sensor causes voltage flow through the bulb

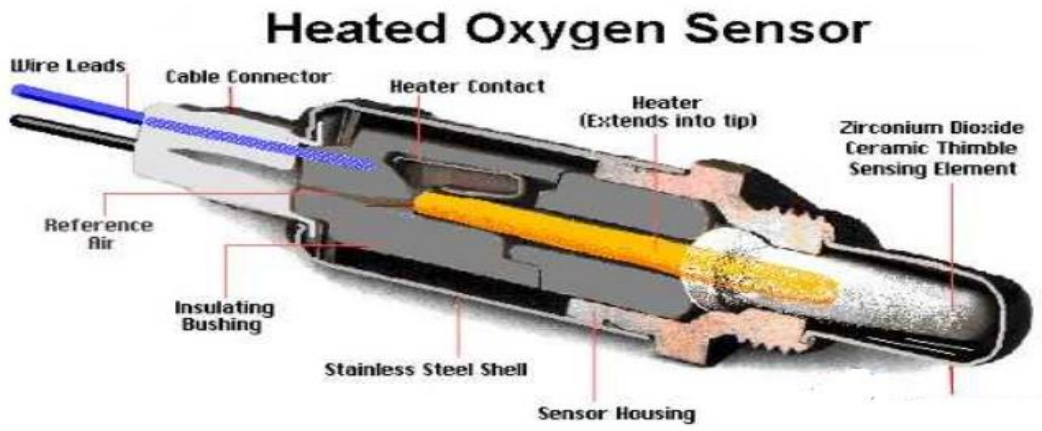

Figure 2.1 Structure of oxygen Sensor

2.2 Pulse Width Modulation (PWM)

PWM or modulation with the width of an impulse, is a widespread term in the world of electrical engineering. Its application in reality is abroad, like in the field of telecommunications, servo motors and so on. PWM can be known as a switch which constantly cycles on and off and can be produced from a digital IC such as microcontroller thus its signal will be in form of square wave. The duration where the signals stay high is called “on time” and the duration stays low is called the “off time”.

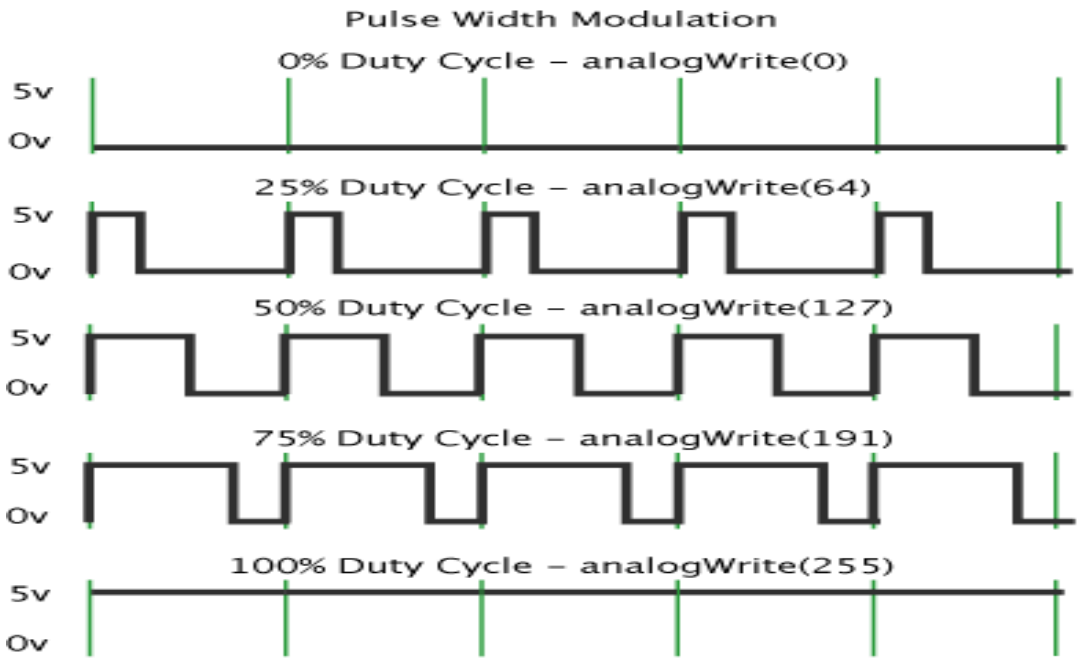

Figure 2.2 Duty cycle of PWM

III. General layout design:

3.1 General layout diagram

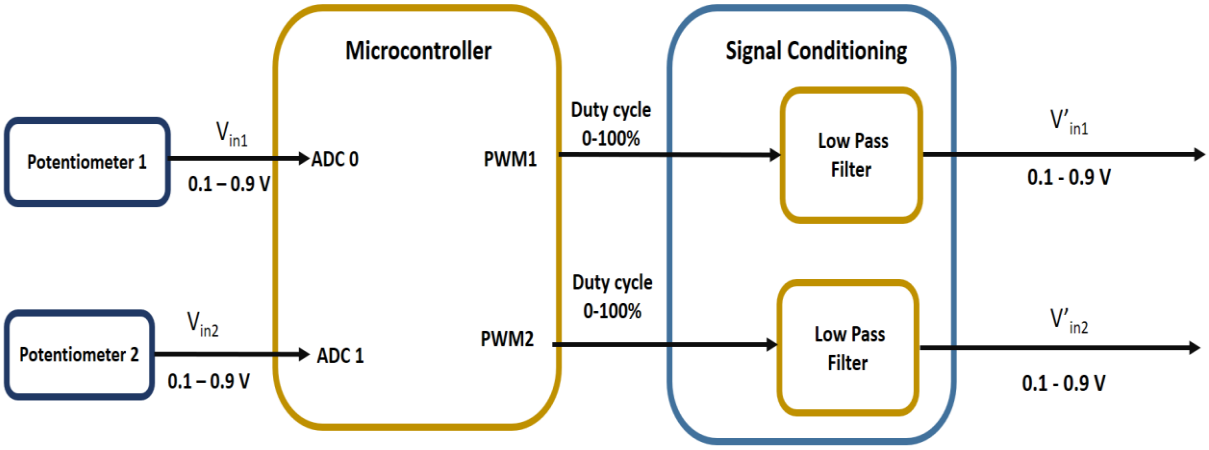

Figure 3.1: General layout diagram

There will be 2 main parts:

- **Microcontroller part**
  - ✓ Read and calculate the pedal position from the simulator potentiometer on computer
  - ✓ Calculate the width of the PWM value corresponding to the value of the potentiometer
- **Signal conditioning:**
  - ✓ Including two 2nd order low pass filters
  - ✓ Low pass filter: modify, filter, remove all unwanted high frequencies of an electrical signal, and only accept or convert all signals that we want.

3.2 Working principles of the system:

3.2.1 Input and output signal

Input	Output
- Analog input signal at A0 – voltage value generated from exhausted gas (0.1V to 0.9 V).	-Digital signal at pin 9 – PWM value, refers to the amount of oxygen of sensor 1. - Digital signal at pin 10 – PWM value, refers to the amount of oxygen of sensor 2.

Table 3.1: Input and output signal

3.2.2 Input and output relationship

- The relationship between 2 analog input value and digital output value (PWM value) is the analog to digital converter (ADC).
- The pulse width at output (pin 9 and pin 10) will be changed according to the input voltage value.

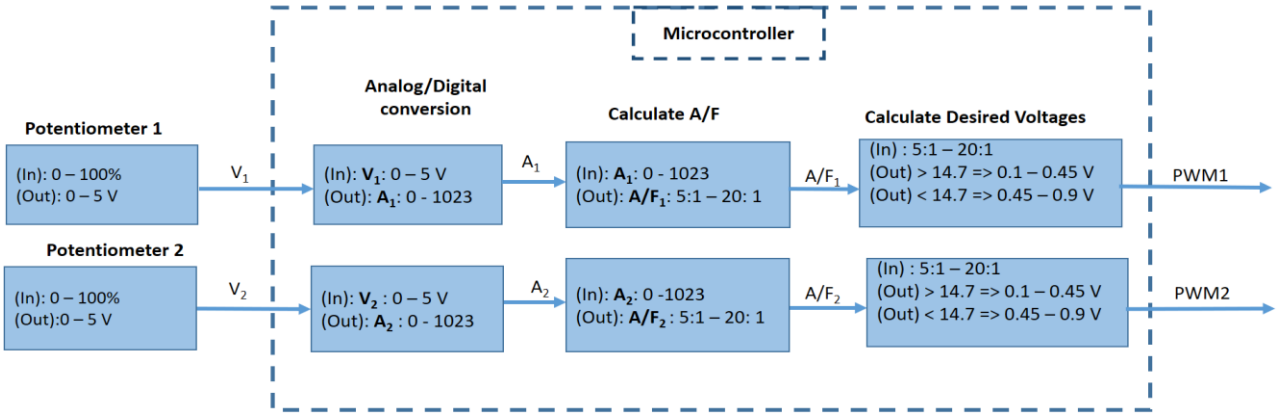

Figure 3.2: Principle diagram of oxygen sensor simulator

IV. Technical design:

4.1 Choosing Components and microcontroller board:

4.1.1 Choosing oxygen sensor:

Oxygen sensor has two main types, the narrow-band and wide band. Based on the advantages of the narrow-band sensor such as easy for checking, cheaper cost and especially its simple signal can be read by a simple ECU input, we will choose the narrow-band oxygen sensor for this project.

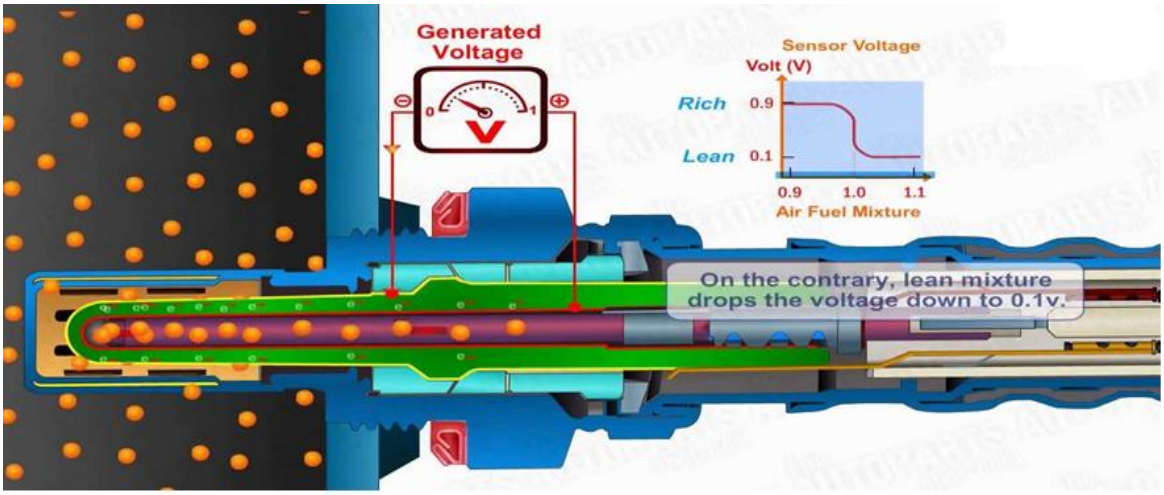

Figure 4.1. Narrow – band oxygen sensor

4.1.2 Choosing electronics board:

With basic simulation, Arduino Uno R3 is chosen for the microcontroller system. It is known as an open-source development board as it allows to use this board to interact with real-world things by uploading programs on this board. Arduino is used mostly as it is inexpensive and can be used in various forms. It is based on ATMEL (ATmega328p microcontroller)**.** It can interact with anything that is controlled by electricity in any way. It can also interact with motors, sensors, and electromagnets

- Arduino UNO R3 details:

Microcontroller	Atmega328 with 8 bits
Power	5V DC(USB only)
Working frequency	16 MHz
Digital pins	14 (6 pins for PWM)
Analog pins	6
Timer/counter	Timer0, Timer1, Timer2

Table 4.1: Arduino's parameter

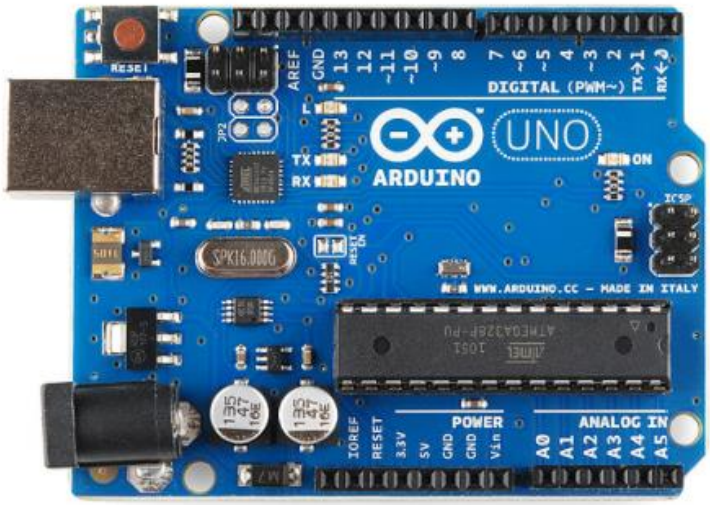

Figure 4.2 Arduino Uno R3

4.1.3 Choosing low-pass filter

There are many types of low-pass filter; first, second low pass filter and so on. In this project we will select the second one because of its advantages. It is not only easy to design but also used in many applications. Second low-pass filter, as its name, is a filter that passes signals with a frequency lower than the selected cut-off one and attenuates signals with frequencies higher than the cutoff frequency. Moreover, it is also prior to analog-to-digital conversion, digital filters for smoothing sets of data. After researching and testing, we will choose values for all components as image below:

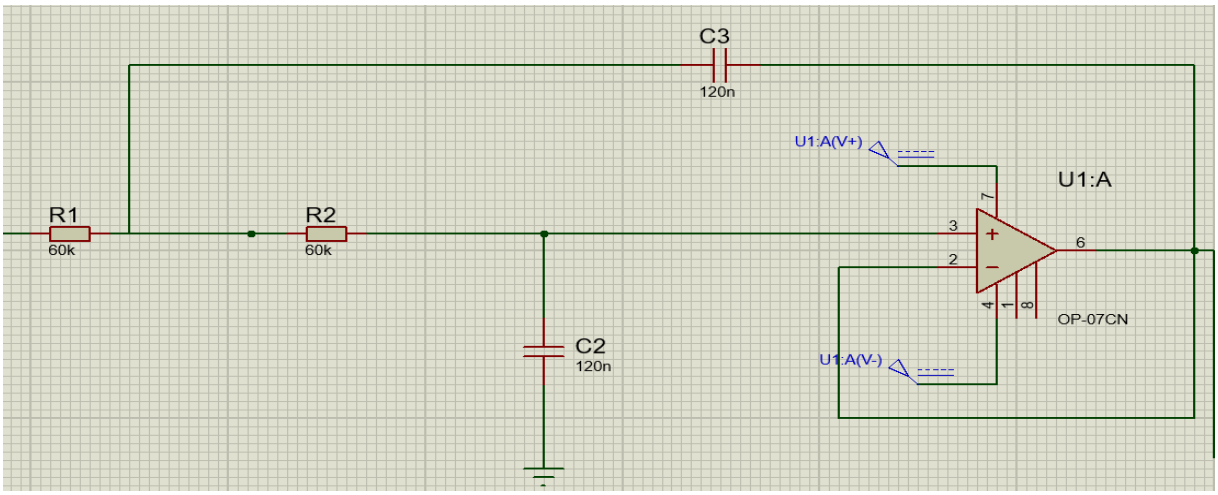

Figure 4.3: 2nd low pass filter

After testing, I choose R3 = R4 = 60k Ω, C3 = C4 = 120nF so that the system can show the signal with smooth and less turbulent, I also choose the Op-07 amplifier to adapt the requirement of my simulation circuit.

4.2 Electrical scheme:

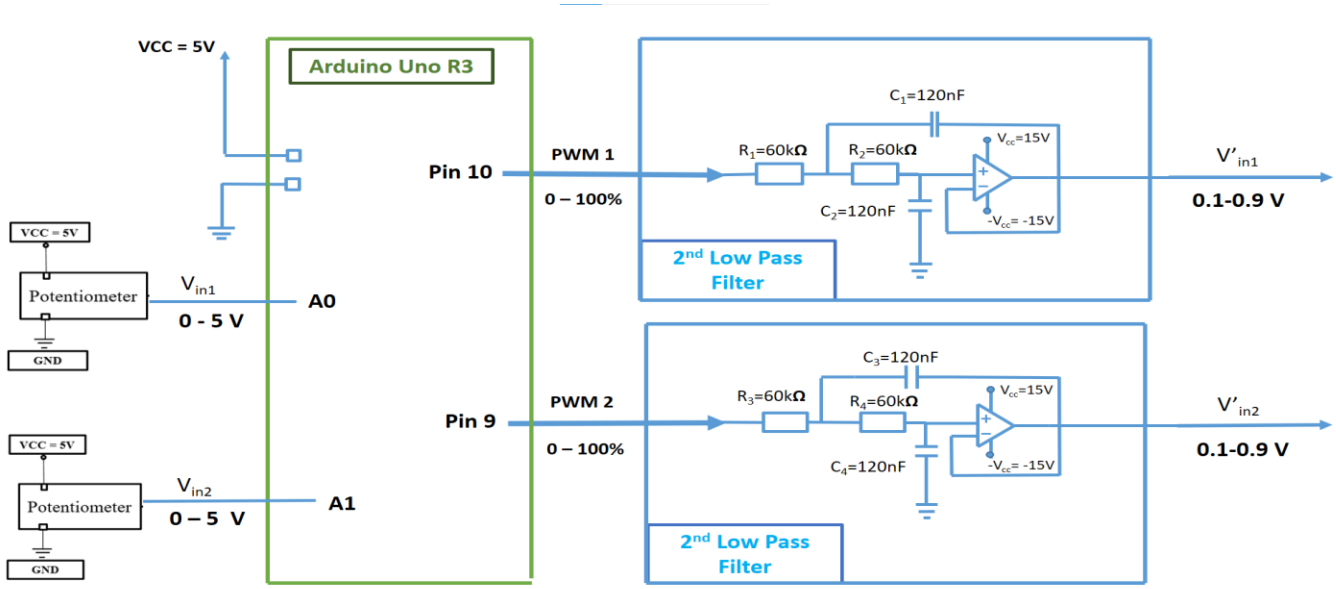

Figure 4.4: Electrical scheme of oxygen sensor simulator

4.3 Algorithms

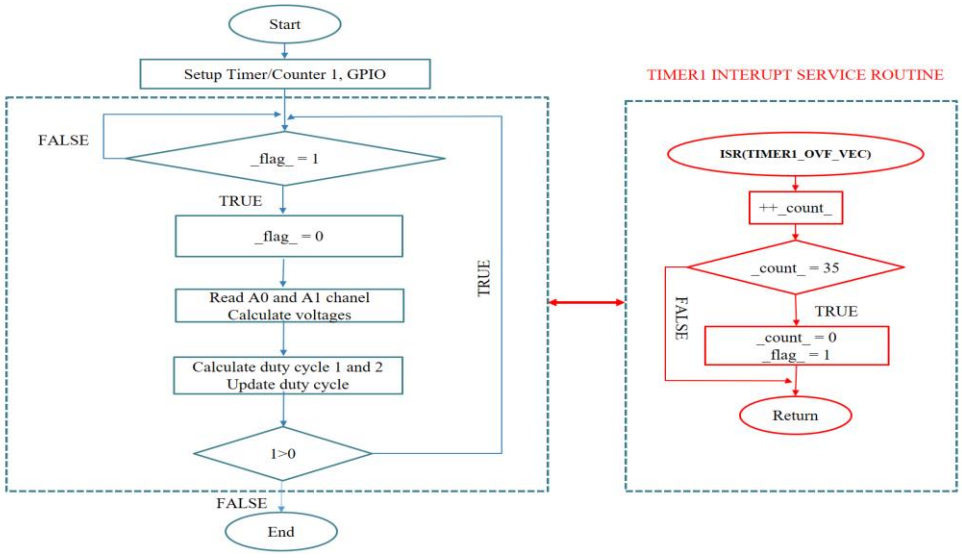

Figure 4.5: Algorithms of the program

Using Timer/Counter 1 with Fast PWM mode to create a working cycle for calculating pedal position value and calculating PWM width.

Analog input pins:

- Ain0: GPIO14, 0-1023 -> 0-5V

Digital output pins

- LPF1: GPIO9, PWM1
- LPF2: GPIO10, PWM2

*** Timer1:**

- PRESCALER = 64
- Arduino frequency=16 MHz
- Timer/Counter frequency = 16MHz / 64 = 250 kHz
- Working frequency = 10kHz
- Mode: 14 - Fast PWM

+ TOP = ICR1 = 24

+ Update of OCR1A at BOTTOM

+ TOV1 Flag Set on at TOP

*** Application:**

- Read ADC value to calculate the voltage value
- Map value convert from 0-1023 to 0.1 – 0.9 V
- Map value convert from 0-1023 to 0.1 – 0.9 V
- Calculate PWM value and assign value in OCR1A and OCR1B

*** Timer1 overflow:**

+ Set TOIE1(Timer Overflow Interrupt Enable 1): interrupt when timer overflow
+ When _count_ value reach 35 (3.5ms) ➔ Timer1 Overflow 35 times

==> Set _flag_ = 1 for communicating purpose (Serial)
Reset _count value to 0

*** Serial:**

- Display OS1 and OS2 voltage in Virtual terminal
- PWM value of OS1 and OS22 after go through LOW PASS FILTER in Digital Oscilloscope

4.4 Timing Diagram

4.4.1 Timing diagram

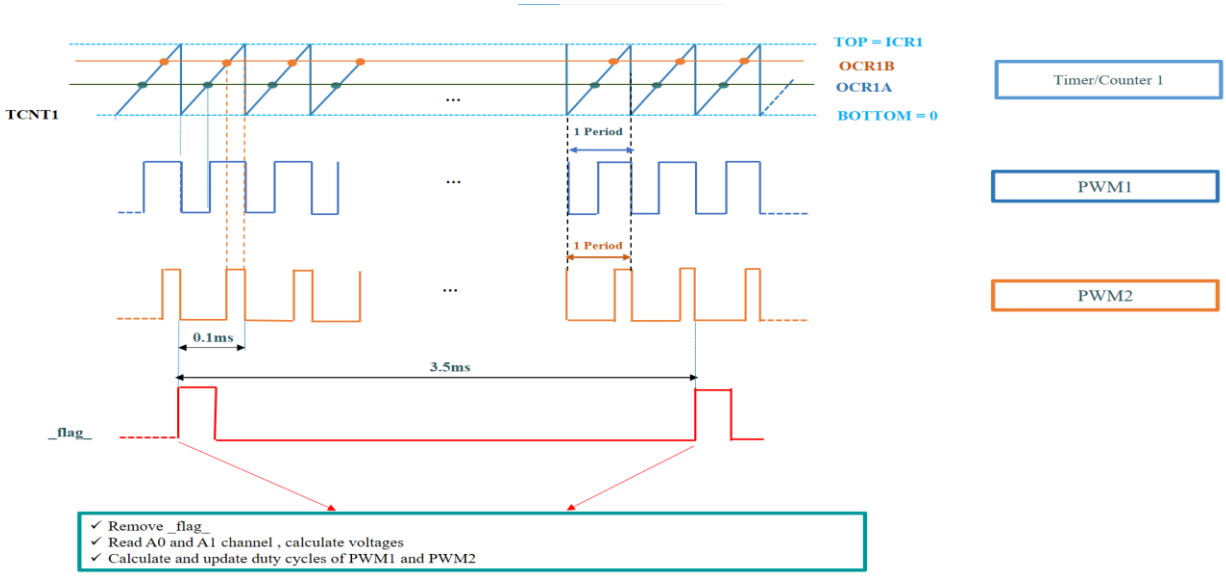

Figure 4.6 Timing Diagram of the program

- With: OCR1x (x can be A or B) depend on output we consider.

  n = number of overflow (in this project n = 35).

- Timer/Counter 1 will be created with a period of 1ms to manage the event performed in the program.
- TNCT1 will be the continuously changing variable, OCR1x and TOP will be fixed variables.
- Observe the Output signal diagram, its status will be changed according to the value of TNCT1, it will go up when TCTN1 = OCR1x and go down as TNCT1 = TOP of ICR1.

4.5 Code and Explanation

4.5.1 CODE:

```
#include <avr/interrupt.h>

bool flag = 0;

unsigned int count;

unsigned int _adc_voltage1, _adc_voltage2;
```

```
double  _vol_value1, _vol_value2 ;
void setup ()
{
  Serial.begin(9600);
  pinMode(9, OUTPUT);
  pinMode(10, OUTPUT);
  cli ();
  TCCR1A = 0;
  TCCR1B = 0;
  TIMSK1 = 0;
  TCCR1B |= (1 << WGM13) | ( 1 << WGM12 ) | ( 1<< CS11) | ( 1 << CS10 );
  TCCR1A |= (1 << WGM11) | (1 << COM1A1) | ( 1 << COM1B1 ) ;
  TCNT1 = 0;
  ICR1 = ((unsigned short int) (16000000.0/64.0/1000.0)) -1;
  OCR1A = 0;
  OCR1B = 0;
  TIMSK1 = (1 << TOIE1) ;
  sei ();
}
  void loop ()
{
  if (flag)
  {
    flag = 0;
```

```
  _adc_voltage1 = map(analogRead(A0), 0, 1023, 1000.0, 9000.0);
  _adc_voltage2 = map(analogRead(A0), 0, 1023, 1000.0,7000.0);
  _vol_value1 = _adc_voltage1 / 10000.0;
  _vol_value2 = _adc_voltage2 / 10000.0;
  OCR1A = (unsigned short int ) ((_vol_value1 / 5.0 )*ICR1);
  OCR1B = (unsigned short int ) ((_vol_value2 / 5.0 )*ICR1);
  Serial.print(_vol_value1);
  Serial.print(" ");
  Serial.println(_vol_value2);
 }
}
  ISR (TIMER1_OVF_vect)  {
  if (++ count == 35) {
    flag = 1 ;
  count = 0;
  }
 }
```

4.5.2 Explanation:

- **Beginning the program:**
  - Declare the all variables using on simulation.
  - Setup input and output of the microcontroller:
  - Analog input A0
  - Digital output pin 9 (OCR1A) to control the PWM
  - Digital output pin 10 (OCR1B) to control the PWM
- **Setup timer 1:**
  - Prescaler = 64
  - Arduino frequency=16 MHz

- Timer/Counter frequency = 16MHz / 64 = 250 kHz
- Working frequency = 1kHz
- Mode: 14 - Fast PWM
- TOP = ICR1 = 249
- Overflow Flag Set on at TOP

- **Operating:**
  - Using Map to convert 0-1023 to 0.1 – 0.9 Volt
  - Using Map to convert from 0-1023 to 0.1 – 0.9 Volt
  - Calculate PWM value and assign value in OCR1A and OCR1B
  - Timer 1 will overflow after counting 35 times.
- **Display:**
  - Display PWM value on Serial monitor according to output at pin 9.
  - Display PWM value on Serial monitor according to output at pin 9.

V. Proteus simulation:

5.1 Electrical diagram

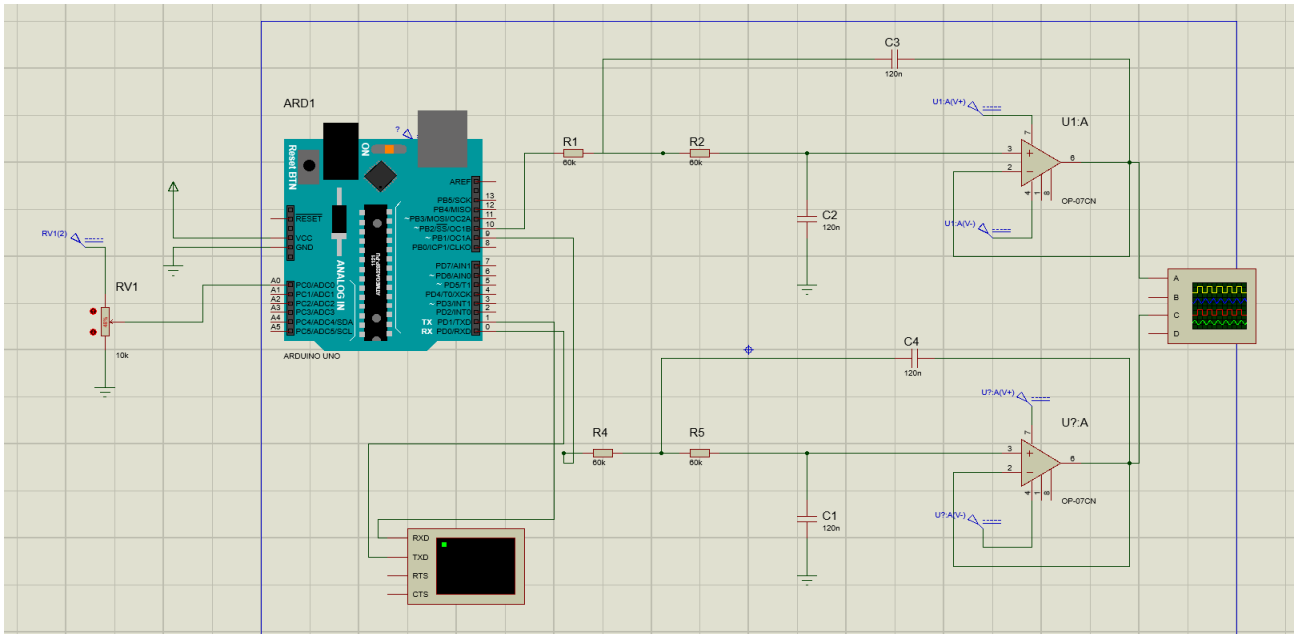

Figure 5.1 Electrical diagram in proteus simulation

5.2 Diagram function

- Potentiometer: output pin is connected to the A0 pin of the Arduino, regulates voltage as given value in reality.
- Second low pass filter acts as a filter allowing signals to pass through with a frequency lower than a selected cut-off frequency and attenuates signals with frequencies higher than the cut-off frequencies. It provides a smoother form a signal, removing the short-term fluctuations and leaving the longer-term trend.
- After processing, the result will be display on both serial monitor and digital oscilloscope.

5.3 Simulating on proteus and in reality

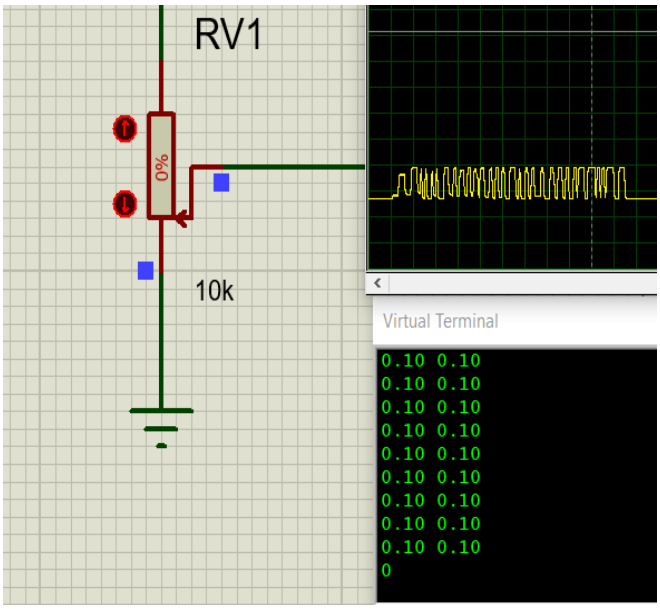

Figure 5.2 Output of sensor 1 in simulation

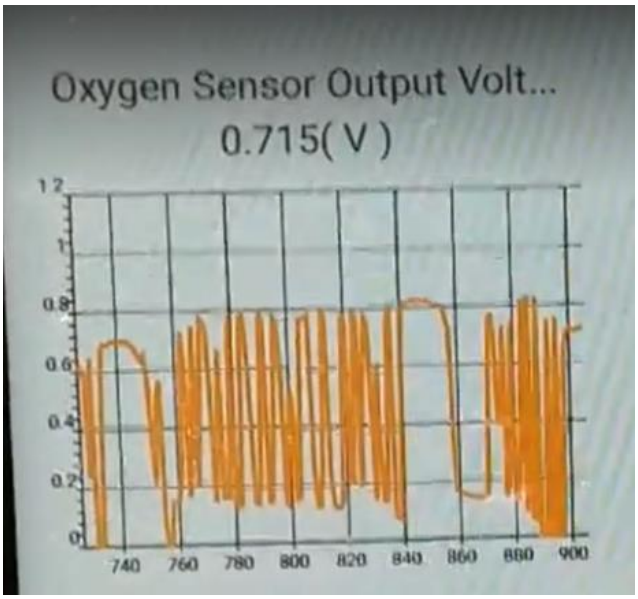

Figure 5.3 Output of sensor 1 in reality

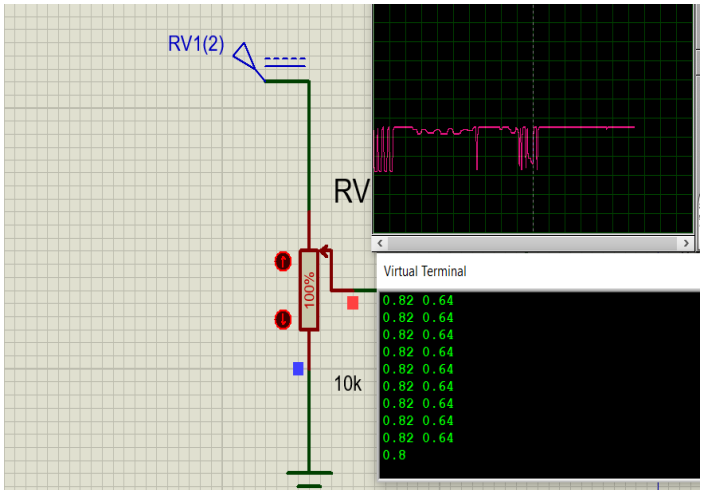

Figure 5.4 Output of sensor 2 in simulation

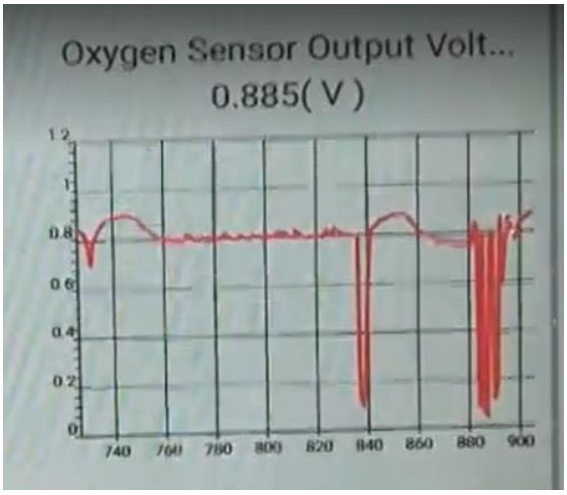

Figure 5.5 Output of sensor 2 in reality

➔ **Discussion:**

Observing above result, we can conclude that the PWM shows on oscilloscope is likely to the reality, the Virtual monitor window can show the voltage due to the changing of potentiometer. The testing system is working as the same function as reality. Both output shows the same value and we can say that both sensor (reality) working normally, one working, one checking.

VI. Conclusion:

After simulating and receiving the expected result, we can come to the conclusion that:

The system has satisfied the following conditions:

- Display two outputs that we wanted in all of conditions.
- Simulating is as like as actual.
- Calculate and display the PWM value through oscilloscope, can regulate at any situations.

However, the system also has some weakness:

- Lagging in displaying the value in proteus.
- The speed of responding is still slower than reality.
- The shape of the pulse is still not smooth.

VII. REFERENCE

[1] Electronics tutorials -2021 –Non-inverting Operational Amplifier

https://www.electronics-tutorials.ws/opamp/opamp_3.html

[2] Electronics tutorials- 2021- Analog to digital converter

https://www.electronics-tutorials.ws/combination/analogue-to-digital-converter.html

[3] Electronics tutorials – 2021 – Second low pass filter

https://www.electronics-tutorials.ws/filter/second-order-filters.html

[4] Wikipedia – 2021- Oxygen sensor

https://en.wikipedia.org/wiki/Oxygen_sensor

[5] Automotive system: “How Oxygen sensor works”

https://www.youtube.com/watch?v=Fl3aD1qJrEg

[6] Oxygen Sensors: A Detail Guide to how oxygen sensors work & what they do

https://www.fixdapp.com/blog/oxygen-sensor/

[7] Arduino Uno: Arduino Uno R3

https://www.arduino.cc/

[8] What is PWM: Pulse Width Modulation – 2018 – Aswinth Raj

https://circuitdigest.com/tutorial/what-is-pwm-pulse-width-modulation

[9] Cảm biến Oxy – Oxygen Sensor – Cấu tạo, nguyên lý và cách kiểm tra

https://oto.edu.vn/cam-bien-oxy-oxygen-sensor/