HO CHI MINH UNIVERSITY OF TECHNOLOGY

OFFICE FOR INTERNATIONAL STUDY PROGRAMS

PROJECT:

DESIGN OF MITSUBISHI EXPANDER CRANKSHAFT AND CAMSHAFT POSITION SENSOR SIMMULATOR

Student's name: Trần Đoàn Đăng Khôi

Student ID: 1752303

Class: CC18OTO

Instructor: Mr. Trần Đăng Long

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

MISSION OF UNDERGRADUATE THESIS

1. **Student's name:** Trần Đoàn Đăng Khôi **- Student ID:** 1752303
2. **Major:** Automotive Engineering **- Class:** CC18OTO
3. **Thesis title:** Design of Mitsubishi Expander crankshaft and camshaft position sensor simulator
4. **Content:**
   - ❖ Introduction of the project.
   - ❖ Working conditions and technical requirements.
   - ❖ General layout design:
     - Design the structure diagram.
   - ❖ Technical design:
     - Select the components for the product.
     - Design the timing diagram.
     - Design algorithm flowcharts.
     - Arduino programming:
   - ❖ 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**

CONTENTS

I. INTRODUCTION

Improvement of automobiles is an actual problem for a large number of scientists and engineers, which they are engaged in to improve the consumer properties of machines and increase their reliability and safety. They are exploring all the new components that provide better performance and lower costs. One of the important components of a modern car is sensors. Crankshaft and camshaft position sensor play an important role in improving productivity and reducing engine fuel consumption

⇨ The project is The design of Mitsubishi Expander crankshaft and camshaft position sensor simulator

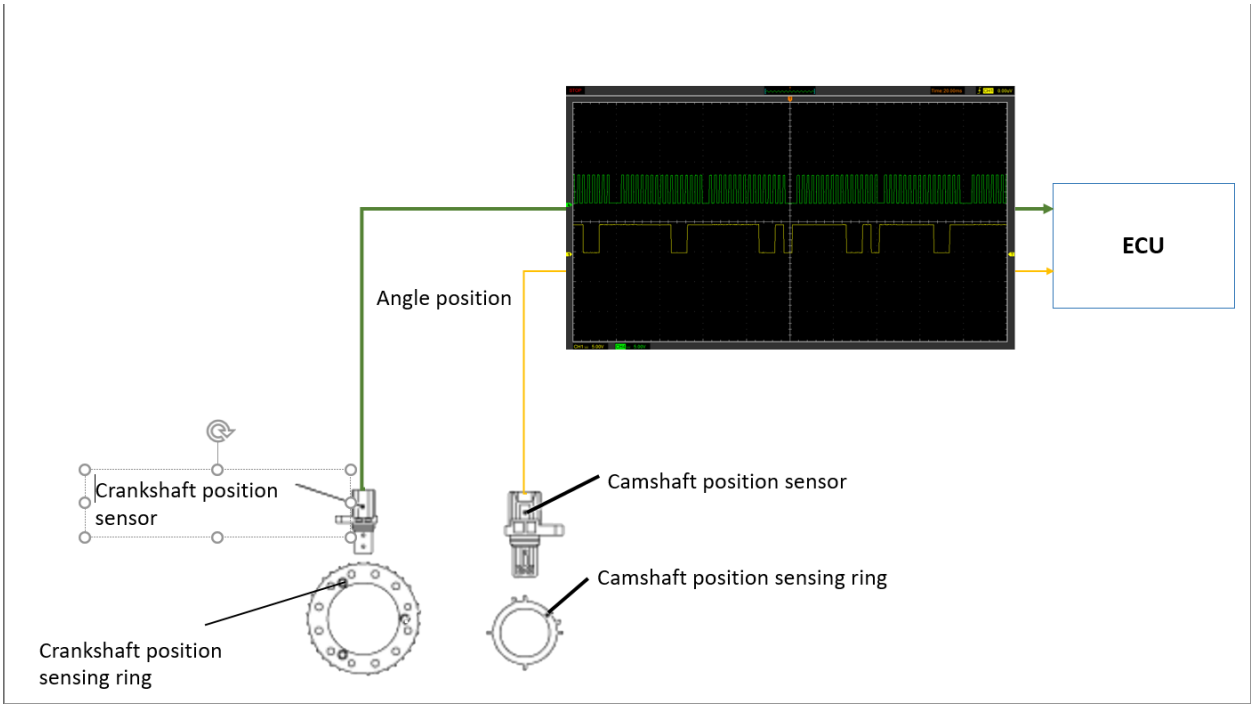

Figure 1.1: Pictorial diagram showing the operating principle of Mitsubishi Expander crankshaft and camshaft signals.

1.1 Working principle of crankshaft and camshaft position sensor on Mitsubishi Expander:

- Crankshaft angle sensor is installed on the inlet side of the cylinder block. The crank angle sensor monitors rotation of crankshaft sensing ring (36 teeth including 3 missing teeth) installed on the crankshaft and converts to voltage (pulse signal) that is output to engine-ECU. Engine-ECU uses crank angle sensor's output pulse to detect crank angle.

- A camshaft position sensor is installed on the inlet of the cylinder head. The camshaft position sensor monitors rotation of the camshaft position sensing ring (6 teeth) and converts to voltage (pulse signal) that is output to engine-ECU. Upon receiving this output voltage, the engine-ECU effects feedback control to optimize the phase of the camshaft. Engine-ECU uses a combination of the camshaft position sensor output pulse signal and crank angle sensor output

pulse signal to identify cylinders in the compression process. This information will help to fine tune spark timing and injector pulse.

1.2 Mitsubishi Expander crankshaft and camshaft signal simulator

Base on the working principle of the system. The objectives of this project is to design a Mitsubishi Expander car crankshaft and camshaft sensor signal simulator. Base on the operating principle of the system, this simulator will simulate the speed of Mitsubishi Expander engine and generate two digital output signals, which are the signals of crankshaft and camshaft respectively.

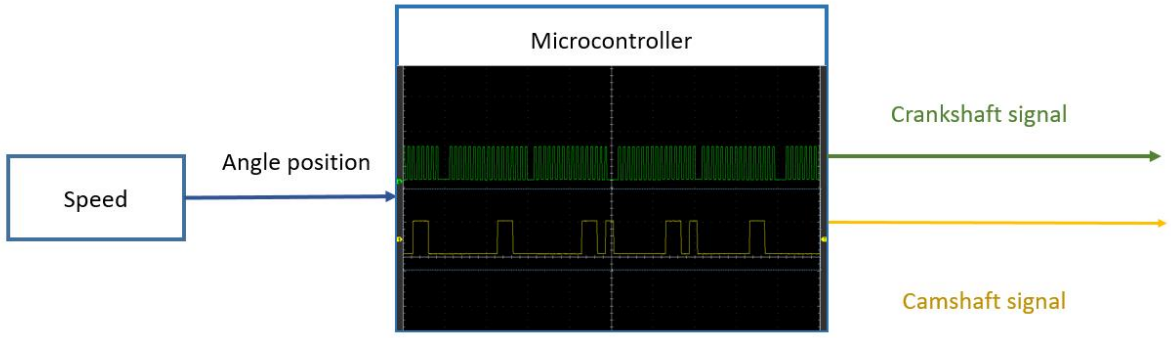

Figure 1.2: Pictorial diagram showing the operating principle of Mitsubishi Expander crankshaft and camshaft signals simulator

1.3 Working conditions:

The environmental conditions for crankshaft and camshaft position sensor include:

- High working in high frequency
- Vibrations and shocks
- High temperatures
- All the electrical components are in good conditions and tested carefully before using.

1.4 Technical requirements:

The microcontroller:

- Calculate exact the speed of crankshaft and camshaft
- Have two digital output signal for crankshaft and camshaft
- Accurately calculate the position of the camshaft and crankshaft

II. GENERAL LAYOUT DESIGN

2.1 **General layout diagram**

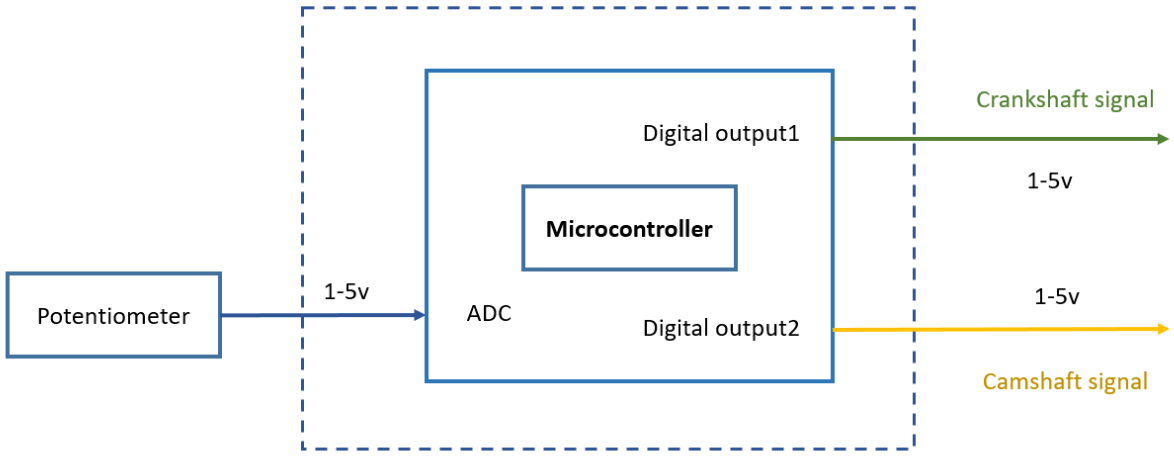

Figure 2.1: General layout design

2.2 Working principles of microcontroller system

2.2.1 Input signal

- Input signal: one analog input signal – voltage value from potentiometer, which reflects speed of crankshaft and camshaft.

2.2.2 Output signal

- Output signal: two digital output signals of crankshaft and camshaft.

2.2.3 The relationship between input value and digital output value

- The relationship between analog input value and digital output value is the analog-to-digital converter (ADC).
- Depending on input voltage from potentiometer, microcontroller calculate speed (RPM) crankshaft and camshaft.
- Depending on RPM, calculate desired frequency:
  + Frequency of round per second (RPS)
  + Frequency each tooth of crankshaft
  + Interrupt frequency.
- From desired frequency, calculate and TOP value (ICR1)

III. TECHNICAL DESIGN

3.1 Choosing microcontroller board and components

3.1.1 Arduino

Arduino is a microprocessor used for programming to interact with hardware devices, designed based on open source code for both hardware and software with a common programming language (C/C++), which can be linked Combined with programs like LabVIEW, MATLAB to take advantage of the power of these programs in programming and simulation.

Advantages of Arduino:

- Simple structure, easy to use, low cost.
- Familiar C/C++ programming language.
- Working in a multi-OS environment, it can run on Windows or Mac IOS.
- Design programs can be loaded from a computer via a convenient USB cable.
- Having a diverse and rich library due to owning a large number of users. This greatly shortens the project completion time.

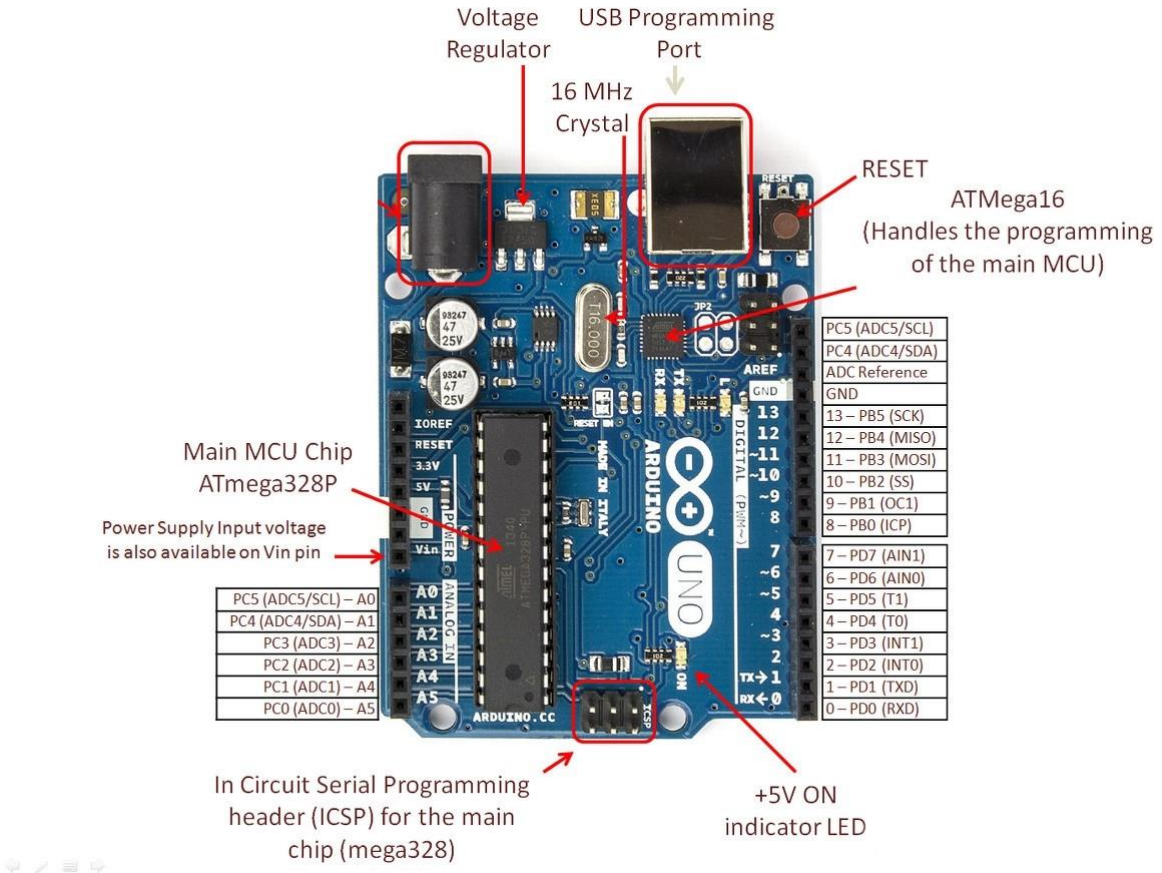

Figure 3.1: Arduino UNO R3 ATmega328p board

Choose Arduino UNO R3 to read voltage value from the simulator potentiometer on the computer and calculate the width of the PWM value corresponding to the value of the potentiometer.

Specifications:

- Microcontroller: ATmega328P
- Operating Voltage: 5V
- Digital I/O Pins: 14 (of which 6 provide PWM output).
- Analog Input Pins: 6
- Clock Speed: 16 MHz

3.2 Electrical scheme

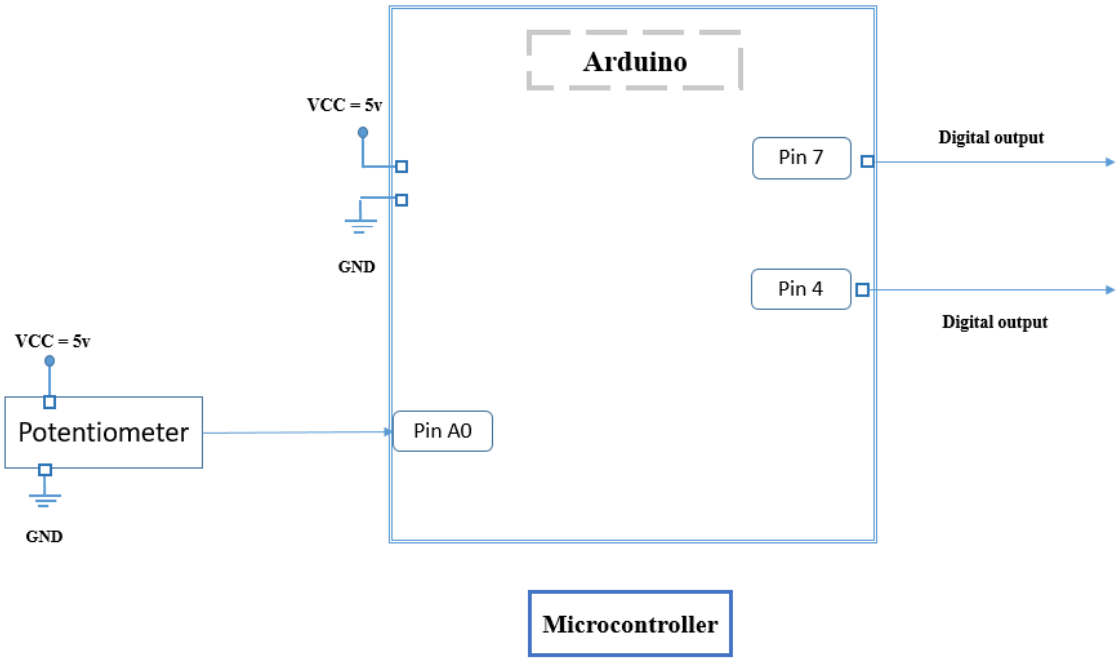

Figure 3.2 : Electrical scheme

3.3 Timing diagram

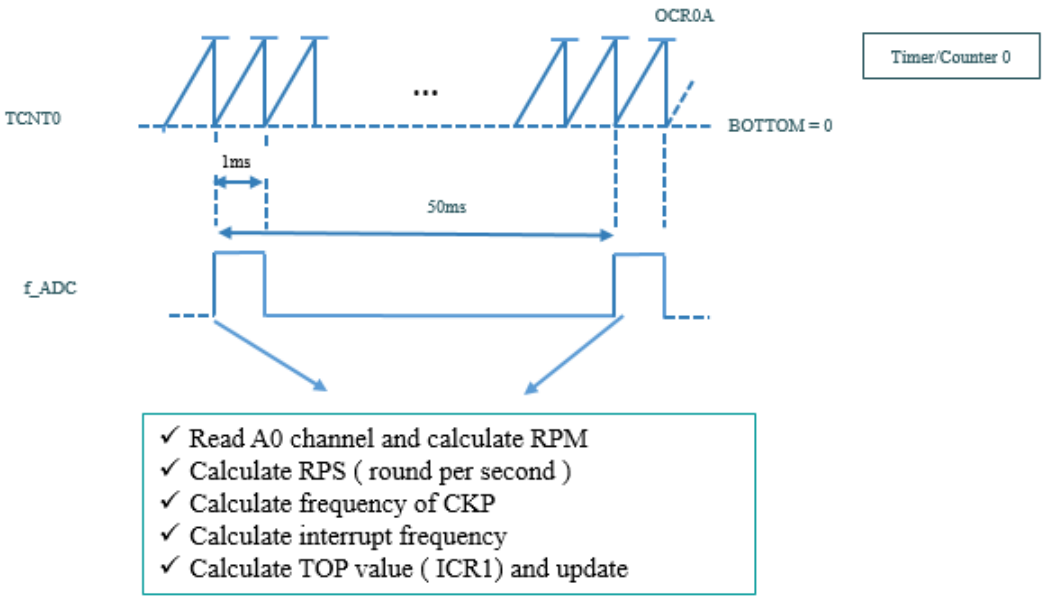

Figure 3.3: Timing diagram of Timer 0

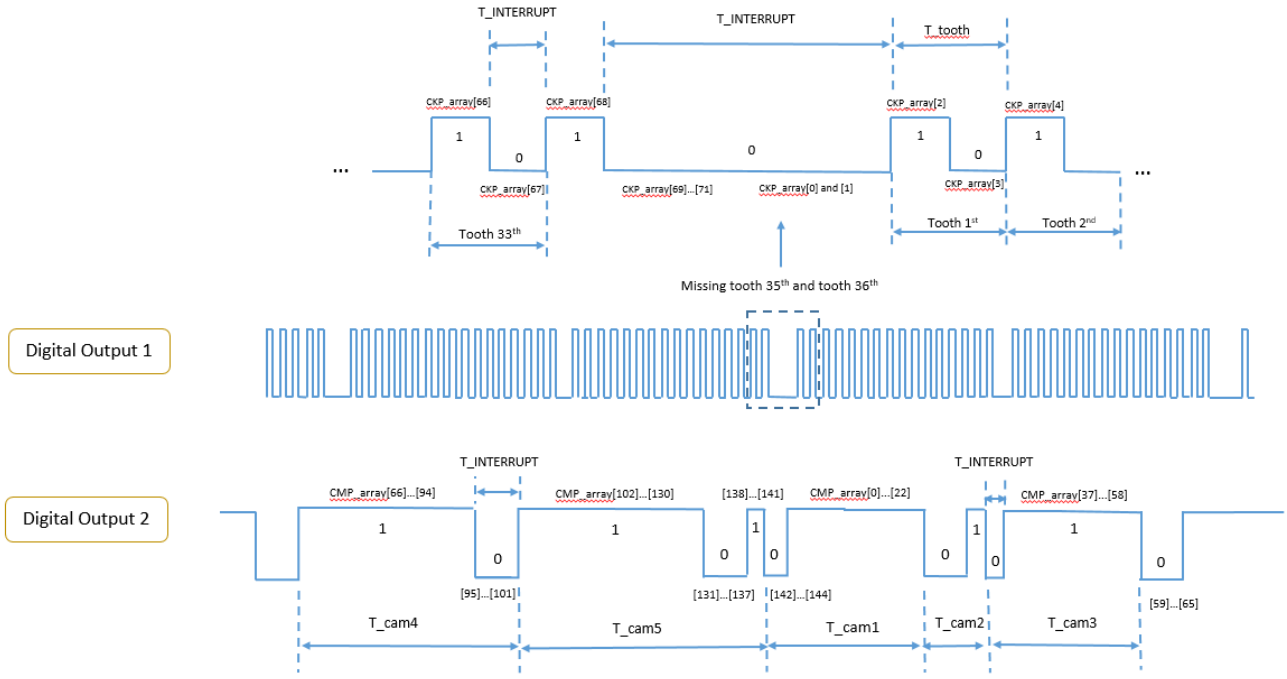

Figure 3.4: Timing diagram of Timer1

3.4 Frequency diagram

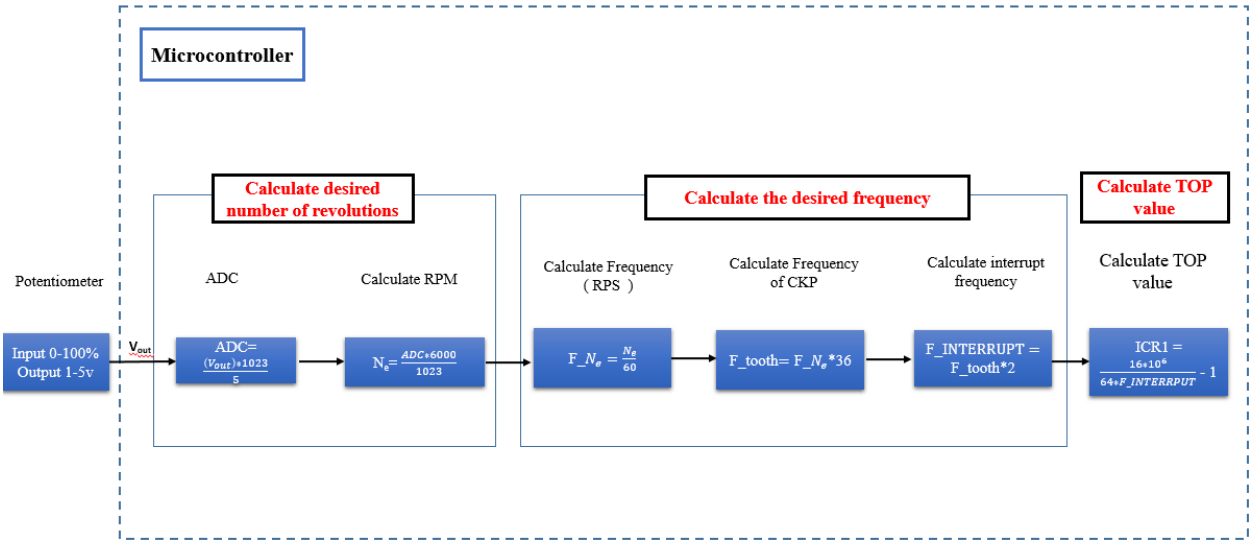

Figure 3.5: Frequency diagram

3.5 Algorithms

3.5.1 Algorithm for main program

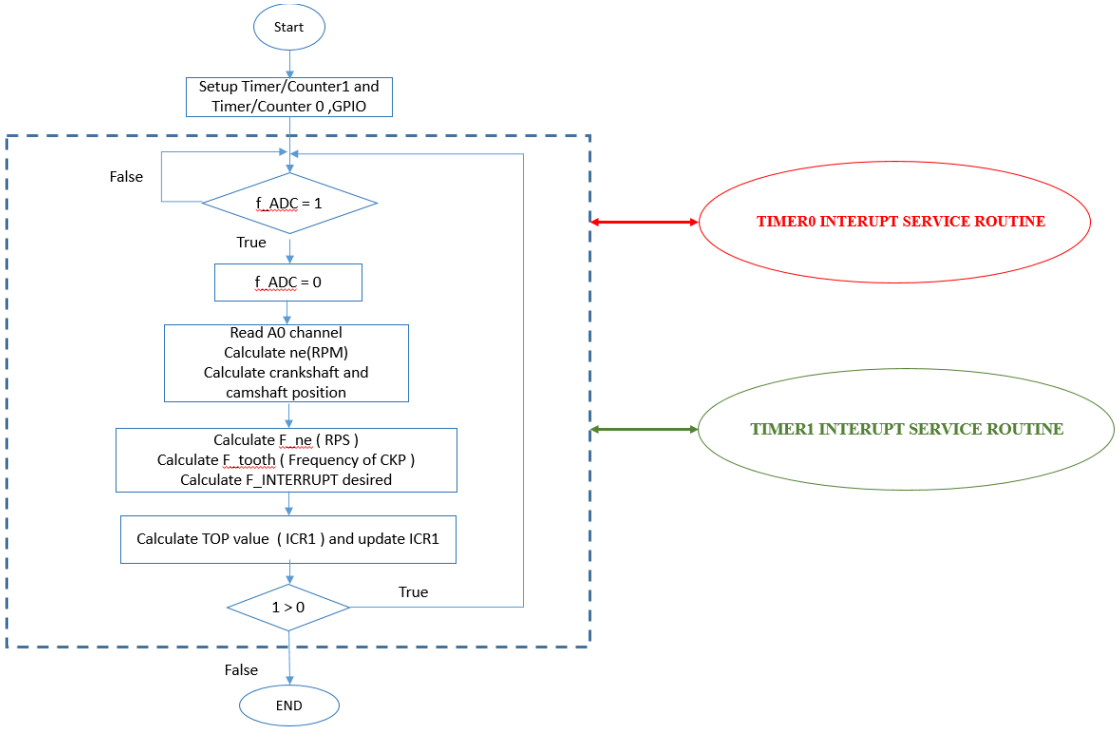

Figure 3.6: Algorithm of main program

3.5.2 Algorithm for Timer 0

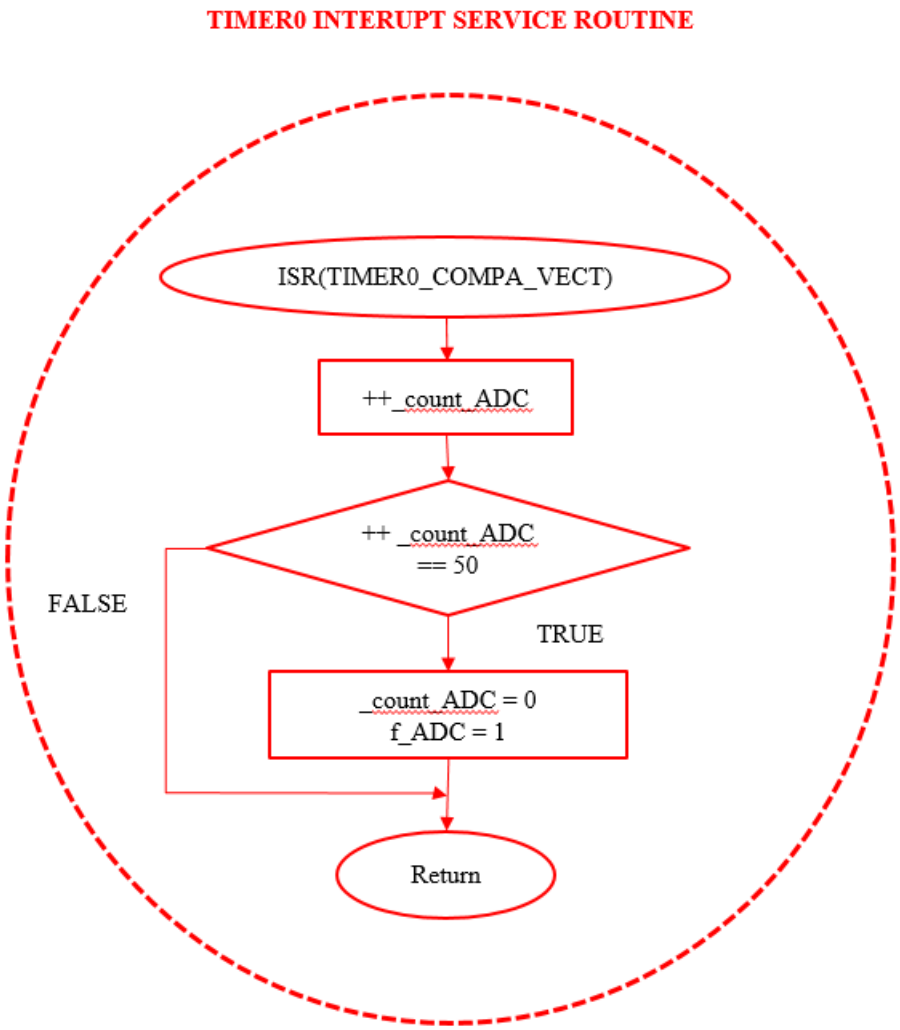

Figure 3.7: Algorithm of Timer0 program

3.5.3 Algorithm for Timer 1

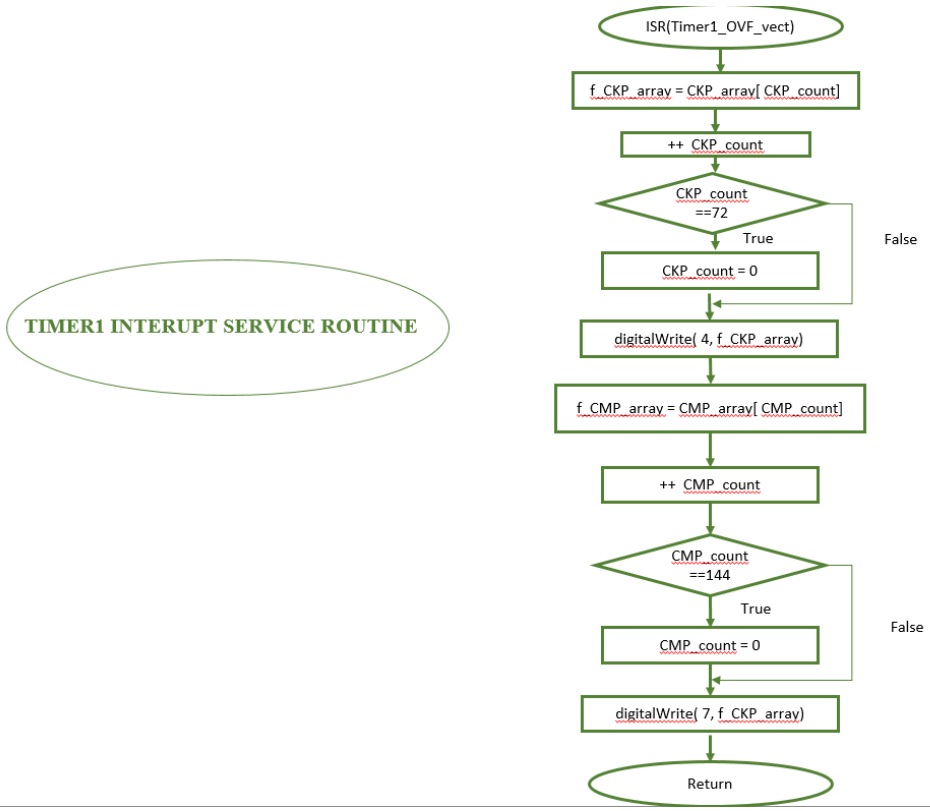

Figure 3.8: Algorithm for Timer1

*** Program function:**

Using timer0 with mode CTC to

- **Analog input pins**
  - ✓ Ain0: GPIO14, 0-1023 -> 1-5V
- **Digital output pins**
  - ✓ Pin 7
  - ✓ Pin 4
- **Timer0:**
  - ✓ PRESCALER = 64
  - ✓ Arduino frequency=16 MHz
  - ✓ Timer/Counter frequency=16MHz/64=250000 Hz
  - ✓ Working frequency: 1000 Hz
  - ✓ Mode: CTC
  - Top = OCR0A = 249
  - Update of OCR0A at BOTTOM
  - Timer Compare Match Flag Set on at TOP
- **Timer1:**
  - ✓ PRESCALER = 64
  - ✓ Arduino frequency=16 MHz
  - ✓ Timer/Counter frequency=16MHz/64=250000 Hz

- ✓ Working frequency: change depending on
- ✓ Mode: Fast PWM

- Duty cycle = 50%
- Top = ICR1
- Update of ICR1A at BOTTOM
- TOV1 Flag set on at TOP

*** Application:**

Main program:

- Read ADC value to calculate the voltage value
- Map value convert from 0 - 1023 to 1 - 5V
- Map value convert from 1-5V to 600 – 6000 (RPM) : n_e
- Calculate the number of revolution per second (RPS)

  $F\_n_e = \frac{n_e}{60}$
- Calculate frequency of each tooth of crankshaft

  $F\_tooth = F\_n_e * 36$
- Calculate interrupt frequency

  $F\_INTERRUPT = F\_tooth*2$
- Calculate TOP value (ICR1)

  $ICR1 = 16*10^6 / (64* F\_INTERRUPT) - 1$

Timer 0 Interrupt Compare Match

Increase variable “count_ADC” to 50

- ⇨ Set flag_ADC = 1
- ⇨ Reset count_ADC to 0

➔ Meaning: Every 50ms will turn on flag to calculate and update in Loop once

Timer 1 overflow:

***Create digital output of crankshaft**

- Set variable f_CKP_array to assign the value from array CKP_array
- Set counter variables CKP_count to determine the element of array
  + When timer interrupt, the counter variables are incremented.
  + When CKP_count == 72, reset CKP_count to 0
- Output pulse by command digitalWrite

***Create digital output of cankshaft**

- Set variable f_CMP_array to assign the value from array CMP_array
- Set counter variables CMP_count to determine the element of array
  + When timer interrupt, the counter variables are incremented.
  + When CKP_count == 144, reset CMP_count to 0
- Output pulse by command digitalWrite

IV. PROTEUS SIMULATION

4.1 Electrical diagram

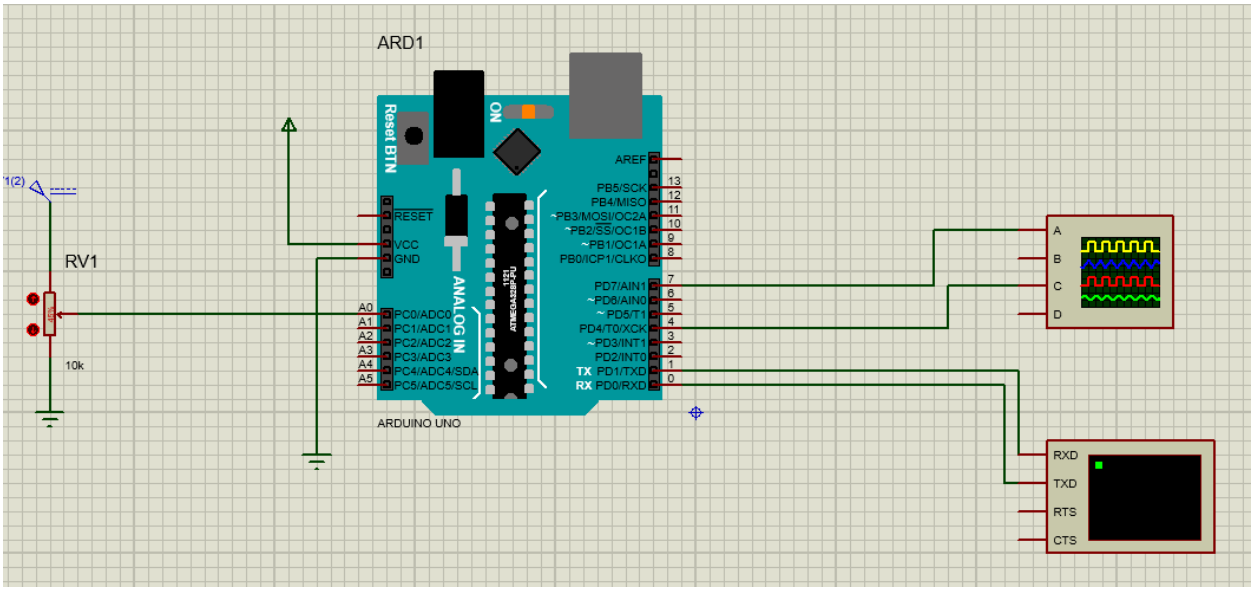

Figure 4.1: Electrical diagram in Proteus

4.1.1 Diagram function

Component	Connection	Function
Potentiometer	➢ Connect pin A0 of Arduino board	Generate voltage value to simulate speed of crankshaft and camshaft
Arduino Uno	➢ Connect to Potentiometer through pin A0 ➢ Pin 7 connect to Port A ➢ Pin 4 connect to Port C ➢ TXD connect to RXD of Virtual terminal ➢ RXD connect to TXD of Virtual terminal	Read and calculate voltage from potentiometer ⇨ Generate digital output pulse at pin 4 and pin 7
Display	• Oscilloscope: ➢ Port A connected to pin 7 of Arduino ➢ Port C connected to pin 4 of Arduino • Virtual terminal: ➢ Pin RXD connected to pin TXD of Arduino ➢ Pin TXD connected to pin RXD of Arduino	• Oscilloscope : display digital output signal of Crankshaft and Camshaft • Virtual terminal: display RPM and frequency.

4.1.2 Simulating result discussion

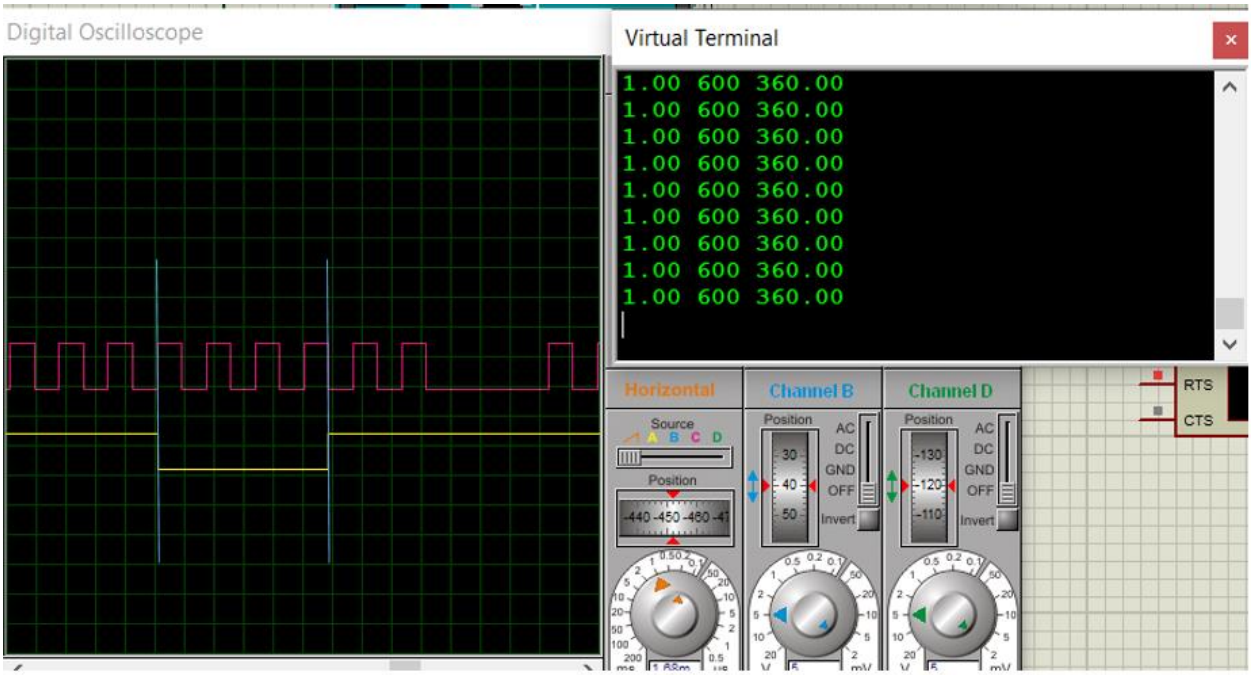

Figure 4.2: Digital output of crankshaft and camshaft at 600RPM

Figure 4.3: Digital output of crankshaft and camshaft at approximately 3000 RPM

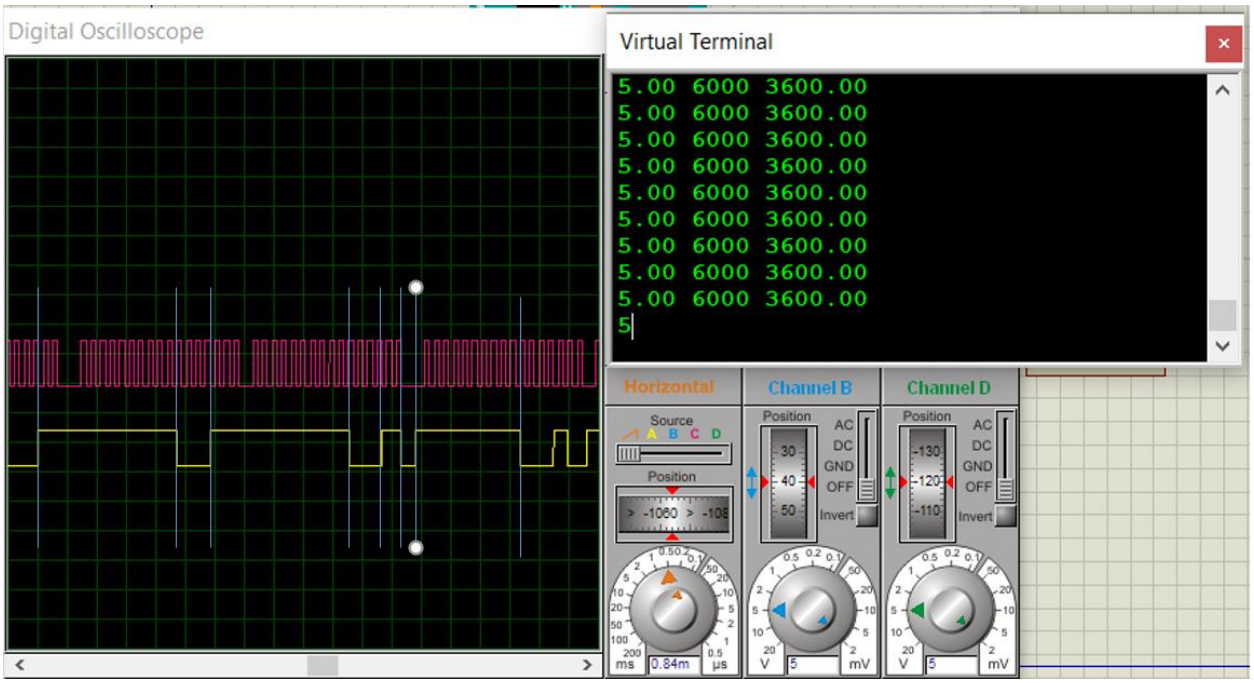

Figure 4.4: Digital output of crankshaft and camshaft at 6000 RPM (maximum)

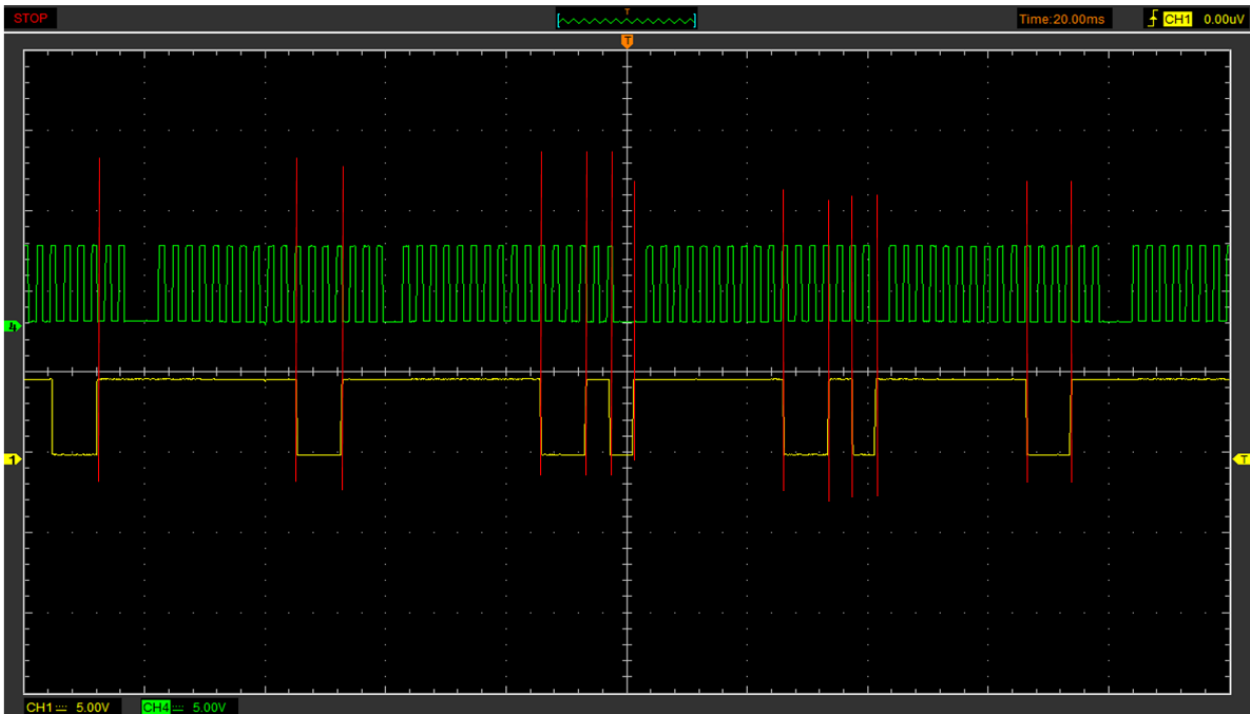

Figure 4.5 : Sample of digital output signal of M.EXPANDER

4.2 Discussion

The criterion of project:

- Accurately simulate the position of each crankshaft pulse relative to the camshaft
  - ⇨ To exactly determine the ignition timing or fuel injection timing.

From observation between simulated signal and actual signal of Crankshaft and Camshaft.

For any change of speed, we completely generate digital output of crankshaft and camshaft without any deviation, 100% accuracy

V. CONCLUSION

Through simulating the signal of crankshaft and camshaft position sensor, we have better understanding of how the sensor works and basic usage of Proteus and Arduino software

Thanks to these programs, it helps to satisfy the following conditions:

- Simulate speed of crankshaft and camshaft
- Calculate desired frequency
- Generate two digital output signal for crankshaft and camshaft

VI. APPENDIX

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

int CKP_count =0 ;

int CMP_count =0 ;

int n ;

int count_ADC ;

bool CKP_array[72]=
{0,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,0,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,1,0,
1,0,1,0,1,0,1,0,1,0,1,0,1,0,0,0} ;

bool
CMP_array[144]={1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,1,1,1,0,0,0,0,1,1,1,1,1,1,1,1,1,1,
1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,1,1,1,1,1,1,1,

1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1
,1,0,0,0,0,0,0,0,1,1,1,1,0,0,0 };

unsigned int ne ;

const float _CLK_PRE = 16000000.0/64 ;

float _adc_value, _volt_value  ;

float  F_ne ;

float  F_tooth ;

float F_INTERRUPT ;

bool f_ADC ;
```

```
bool f_CKP_array;
bool f_CMP_array;


 void setup()
 {
 Serial.begin(9600);

pinMode(4, OUTPUT);
pinMode(7, OUTPUT);

cli();
//setup timer 0
TCCR0A = 0;
TCCR0B = 0;
TCNT0=0;
TIMSK0 = 0;

TCCR0B |= (1 << CS01  )|(1 << CS00) ;
TCCR0A |= (1 << WGM01) | (1 << COM0A1);
OCR0A =   (unsigned short int)(16000000/64/1000.0) -1 ;
TIMSK0 |= (1 << OCIE0A)  ;


TCCR1A = 0;
TCCR1B = 0;
TCNT1=0;
TIMSK1 = 0;
TCCR1B |= (1 << WGM13) | (1 << WGM12) | (1 << CS11) | (1 << CS10);
TCCR1A |= (1 << WGM11) | (1 << COM1A1)| (1 << COM1A0)  ;
TIMSK1 |= (1 << TOIE1);
```

```
ICR1 = 249 ;
 sei();
}

   void loop()
{
    if(f_ADC)
{
  f_ADC = 0;
 _volt_value = map(analogRead(A0), 0 , 1023 , 1000, 5000 );
ne = map(_volt_value , 1000 , 5000 , 600 , 6000 );
F_ne = ne/60 ;     //round per sencond
F_tooth = F_ne*36 ;  //tan so của CKP
F_INTERRUPT = F_tooth*2 ;
ICR1 = (unsigned short int)( _CLK_PRE/F_INTERRUPT ) -1  ;
Serial.print( _volt_value/1000.0);
Serial.print( " " );
Serial.print( ne );
Serial.print( " " );
Serial.println(F_tooth);

}
}

ISR(TIMER0_COMPA_vect)
{

  if(++ count_ADC == 50 )
  {
    f_ADC = 1 ;
    count_ADC = 0;
```

```
  }
}

ISR(TIMER1_OVF_vect)
{

 f_CKP_array = CKP_array[CKP_count];
 ++ CKP_count ;
 if(CKP_count == 72 )
 {CKP_count = 0; }
  digitalWrite( 4, f_CKP_array );
  f_CMP_array = CMP_array[CMP_count];
   ++ CMP_count ;
  if(   CMP_count == 144 )
 {CMP_count = 0; }
  digitalWrite ( 7, f_CMP_array );}
```

VII. REFERENCES

[1] "Hall Camshaft Sensor Measurement". *Tiepie-Automotive.Com*, 2021, https://www.tiepie-automotive.com/en/articles/camshaft-sensor-hall.

[2] 2021, https://www.electronics-tutorials.ws/electromagnetism/hall-effect.html. Accessed 31 July 2021.

[3] "Shieldsquare Captcha". *Iopscience.Iop.Org*, 2021, https://iopscience.iop.org/article/10.1088/1757-899X/252/1/012099/pdf.

[4] "How They Work - Denso". *Denso-Am.Eu*, 2021, https://www.denso-am.eu/products/automotive-aftermarket/engine-management-systems/camshaft-crankshaft-sensors/how-they-work/.

[5] Atmel corporation- 01/2015 - 1600 Technology Drive, San Jose, CA 95110 USA - Atmega328p Datasheet https://bitly.com.vn/99w3u9

[6] 2021, https://learnmech.com/crankshaft-position-sensor-function-types-working/. Accessed 30 July 2021.

[7] 2021, https://learnmech.com/camshaft-position-sensor-function-types-working/. Accessed 30 July 2021.