ABSTRACT

Digital is certainly a unique system. It gives the same function as the analog system which has been widely used throughout our life and the innovation of digital appliances has given great impact in our work system. Therefore in this project, a model of an Adjustable Speed Limit Digital Speedometer with Speed Alarm is developed in the laboratory using a PIC microcontroller which will be equipped with a 12-buttons keypad, a two dual digits 7-segment display and also a buzzer. A custom made programming was developed using assembly language for the PIC microcontroller which will be the key of the whole project. In the end of the project, the built model of an Adjustable Speed Limit Digital Speedometer with Speed Alarm will able to display the actual running speed of the vehicle in km/h reading and also the desired speed limit entered by the user in digital mode while the secondary purpose is to have an adjustable speed limit controller as an input for the setting of the speed limit together with a speed alarm that buzzes continuously when the speed limit being set is exceeded by the current running vehicle speed.

ABSTRAK

Sistem digital sememangnya satu sistem yang unik. Ia mempunyai fungsi yang sama seperti sistem analog yang telah digunakan secara meluas sepanjang hayat kami dan dengan adanya inovasi perkakasan-perkakasan digital, sememangnya ia telah memberi kesan yang besar didalam sistem kerja kami. Oleh itu di dalam projek ini, satu model Had Kelajuan Boleh Ubah Meter Laju dengan Loceng Amaran telah dimajukan di dalam makmal dengan menggunakan satu mikropengawal PIC yang dilengkapi dengan satu 12-butang kekunci, 2 unit paparan 7 ruas dengan 2-digit, dan juga satu loceng isyarat. Satu aturcara juga telah ditulis untuk mikropengawal PIC dengan menggunakan bahasa himpunan yang merupakan kunci kepada keseluruhan projek ini. Pada akhir projek ini, model Had Kelajuan Boleh Ubah Meter Laju dengan Loceng Amaran yang telah dibina boleh memaparkan kelajuan kenderaan yang sebenar dalam bacaan km/j dan juga had laju yang telah ditetapkan oleh pengguna dalam bentuk digital disamping mempunyai satu kawalan had kelajuan boleh ubah kepada pengguna untuk menetapkan had kelajuan berserta dengan satu loceng isyarat yang akan berdengung sekiranya kelajuan kenderaan semasa telah melebihi had kelajuan yang telah ditetapkan oleh pengguna.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

AC	-	Alternating Current
DC	-	Direct Current
Ω	-	Ohm
F	-	Farad
A	-	Ampere
V	-	Voltage
RMS	-	Root Mean Square
P	-	Peak
W	-	Watt
Hz	-	Hertz
LED	-	Light Emitting Diode
LDR	-	Light Dependent Resistor
s	-	Second
T_P	-	Time Period of Signal 1
T_{PO}	-	Time Period of Signal 2
C	-	Calibration Number
f	-	Frequency
k	-	Kilo
M	-	Mega
μ	-	Micro
p	-	Pico
m	-	Milli
T	-	Turns
N	-	North
S	-	South

mm	-	Millimeter
Km/h, km/h	-	Kilometer per hour
RPM, rpm	-	Revolution per minute
I/O	-	Input/Output
TMR0	-	TIMER0
TMR1	-	TIMER1
EEPROM	-	Electrically Erasable Programmable Read-Only Memory

CHAPTER I

PROBLEM STATEMENT

1.1 Introduction

Anyone who drives a car inevitably exceeds the posted speed limit on occasions, either deliberately or due to lack of attention. But these days, it's really not a good idea to speed. Apart from the obvious safety considerations, there are lots of speed cameras about and it's all too easy to cop a heavy fine and maybe even lose you license.

You do not have to be a speed demon either. On a long trip, you can gradually creep up as you become used to the road conditions. It's also quite difficult to stick to the speed limit in a 60km/h zone after you have been driving at high speed on the open road as driving at 60km/h seems agonizing slow after driving at 100km/h. In this situation, an Adjustable Speed Limit Digital Speedometer with Speed Alarm can keep you on your toes and ensure that you stick within the posted limit.

Another situation where it's easy to inadvertently exceed the speed limit is if you are using a cruise control. Now while cruise control are a great help when it comes to maintaining a set speed but they do have one inherent limitation which is the speed of the car can increase beyond the set limit on downhill stretches and once again an Adjustable Speed Limit Digital Speedometer with Speed Alarm can instantly warn you when you have overstepped the mark.

1.2 Objective

The main objective of this project is to build a model of car digital speedometer that will display the actual speed of the vehicle itself in the km/h reading and also displaying the desired speed limit entered by the user in digital mode while the secondary purpose is to have an adjustable speed limit controller as an input for the setting of the speed limit.

Last but not least, a speed alarm that buzzes continuously when the speed limit is exceeded base on the speed limit entered by the user through the adjustable speed limit controller which in the case will be a 12-buttons keypad.

This project illustrates the importance and steps of knowing how the microcontroller work which includes programming and hardware interfacing to the microcontroller chip.

1.3 Scopes of Project

Covered in this project are:

i. Designation and construction of the electronic processing circuit

ii. Designation and construction of the adjustable speed limit controller and display circuit with speed alarm
iii. Basic construction of microcontroller circuit
iv. Programming of the PIC Microcontroller using assembly language

1.4 Thesis Organization

This thesis consists of five chapters and the outline of each chapter is stated as follows:

Chapter I give brief description of Problem Statement behind this project. The introduction, objective and scopes of this project are being discussed in detail.

As for Chapter II, it gives all the Literature Review behind this project and among the topic being discussed here are the background, theory implemented and also the project components in this project.

In Chapter III is the Methodology which gives the description of the whole project process which includes introduction, tools related to the project, hardware circuit's development and also software's development of the project.

Meanwhile Chapter IV will be focusing on the Results and Observations which includes the input and output signal observations of the project together with some analysis on comparison between actual vehicle speeds being displayed with calculated vehicle speed.

Last but not least, Chapter V presents the Discussion, Conclusion and Recommendations of the built project model based on the overall results and observations in Chapter IV.

CHAPTER II

LITERATURE REVIEW

2.1 Background

Digital is certainly a unique system. It gives the same function as the analog system which has been widely used throughout our life and the innovation of digital appliances has given great impact in our work system.

Thus, for example the speedometer of our vehicles is in analog system and the digital technology slowly took over this “job”, although it hasn’t been widely use. Therefore, steps must be considered first in order to change an analog speedometer to a digital systems.

As we know, the speedometer implement on our vehicles are mostly in analog system and the disadvantages of this system are an average reading is displayed by a needle, which can be lag behind rapid speed changes thus did not show the exact speed and meters tend to be somewhat fragile where the higher tendency of malfunction to the speedometer needle might occur. Furthermore, the older generation preferred this digital display type of display as it was not necessary to focus on them as much as was necessary to see a needle especially at night. After all, digital speedometer is easier to read at a glance as we need only to see a number rather than a needle and the scale to which it points as in the analog speedometer.

Although the digital speedometer is available in the market but not all the models are implement with it. In other words, only some of the luxurious car model came with the digital speedometer instead of the analog type which we use to see around us.

Therefore, this project has it purpose, that is to create a digital speedometer to overcome all the problems above and also to understand more on how the digital speedometer that is currently available in the market works. Apart from that, the purpose of the project is also to have an adjustable speed limit with speed alarm to tell the driver that he or she is driving exceeding the set speed limit. Besides that, with the speed alarm implemented, we can eliminates the necessity of looking at the digital speedometer or the analog speedometer all the time which can make us distracted from driving and thus reducing the rate of accident that can happen.

2.2 Theory Implemented

In previous year, the digital speedometer project did by Rudy Timbang (2003) before this is implement's by using analog to digital converter, logic circuit such as binary code decimal, 7-segment decoder to display the digital speed of the vehicle. Therefore, it involved a more complex circuit compare to what is needed if implement using the microcontroller chip although the latter needed it to be program. Besides that, the past digital speedometer also does not include an adjustable speed limit controller with speed alarm.

As this project is a microcontroller base project, therefore wide knowledge in how the microcontroller work's is the most important part in this project. Among the knowledge needed here are programming of the microcontroller chip with all the hardware well connected to it and also designing the hardware circuit which is suitable to support the microcontroller chip.

Apart from that, knowledge in designing and constructing the hardware circuit is also crucial besides the skills in soldering and troubleshooting of the hardware circuit will be also an important factor to this project.

2.3 Project Components

Generally, this project consists of 4 mains component which is the electronic signal processing circuit, the input circuit which include a keypad, an output circuit which have a 4 digits 7-segment display together with a speed alarm and last of all, the speed limit controller which is the microcontroller. As for the speed sensor component, it will not be applied in this project as the input to the electronic signal processing circuit will be replaced by AC sine wave signal generated from a function generator. This is because both applied the same concept where the speed sensor is actually a coil with magnets mounted on the driveshaft, as the driveshaft rotate then the magnets will induce AC sine wave voltage to the coil. Figure 2.1 show the project block diagram developed for this project.

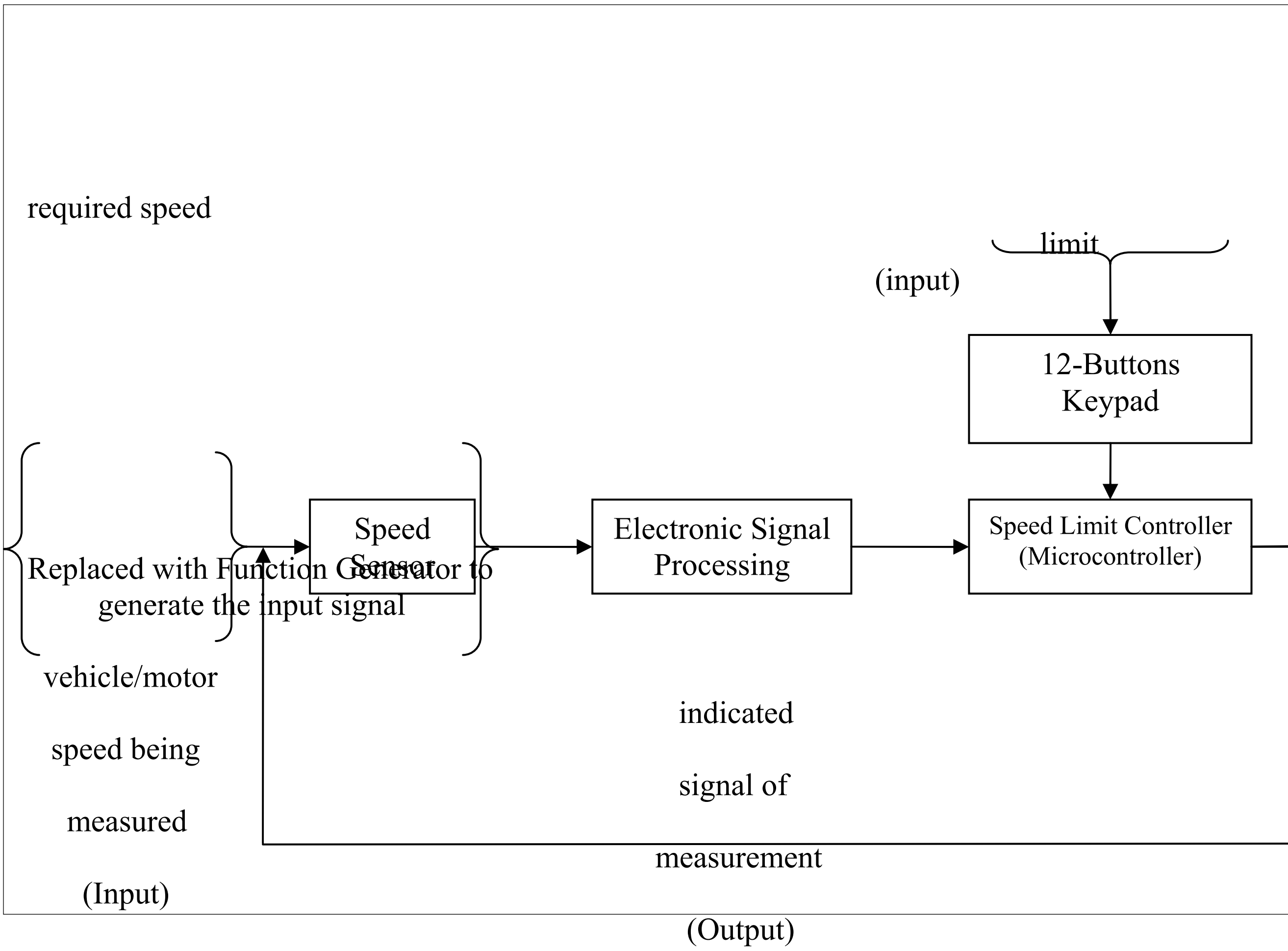

Figure 2.1: Project Block Diagram

CHAPTER III

METHODOLOGY

3.1 Introduction

The Adjustable Speed Limit Digital Speedometer with Speed Alarm will consist of 3 major parts that will be developed accordingly. The major parts are the circuit designation and basic construction of the electronic signal processing circuit, follow by the designation and construction of the adjustable speed limit controller and display circuit with speed alarm and finally will be the full circuit implementation using PIC microcontroller.

First of all, the circuit designation and basic construction of the electronic signal processing circuit will be carry out as it is important to detect the actual running speed of the vehicle. The speed sensor is an inductive type and therefore it will consists of a coil together with 4 magnets (2 'N' and 2 'S') pole each which are mounted on a driveshaft and as the magnet spin past, they will then induce an AC sine wave voltage into the coil. Instead of constructing the speed sensor, alternatively we can obtain the speed signal from the vehicle's engine management computer if this can be identified or even by replacing it with a function generator to generate the input AC sine wave signal.

Therefore, I will choose the latter to proceed further in this project which is by replacing it with a function generator to generate the input AC sine wave signal as the project I am developing here is basically a model which is developed inside the laboratory. Next, the AC sine wave signal received from the function generator will be process by the electronic signal processing circuit and the latter will converts the signal received into a +5V pulse train that will later feed into the PIC microcontroller as microcontroller chip's can only receive signals between 0 to +5V. After all, the basic concept of this digital speedometer will be the number of pulses received from the speed sensor within a set period of time before being process with a calibration value which will then indicates the actual speed of the running vehicle.

Besides that, the construction and designation of a voltage regulator that will provide 5V DC to the microcontroller from the car battery which supply +12V DC and the construction and designation of the electronic signal processing circuit will also be carry out at the same time. After that, to proceed further on will be the designation and construction of the adjustable speed limit controller and display circuit with speed alarm. During this process, circuits which will be design and built here are a 12-buttons keypad which will function as an input setting for the user to enter the desired speed limit, a 4 digits 7-segment display which will show the actual speed of the running vehicle and also the speed limit set by the user, a speed alarm that will buzzes continuously when the speed limit is exceeded base on the speed limit entered by the user through the 12-buttons keypad.

Apart from that, a basic microcontroller circuit is to be built in order for the microcontroller to works. With all the circuits ready, the next step is interfacing all the built circuits together to form the core of the hardware needed in order for this project to be a success. All the built circuits as mentioned are then connected to one another using the PCB male and female type of connector.

Finally will be the full circuit implementation using PIC microcontroller. As we all know a microcontroller without program in it will just be a useless chip. Therefore, during this whole process, the programming of the PIC microcontroller with all the hardware well connected will be carrying out here. The algorithm of the programming will be written based on all the hardware connected to this microcontroller chip using software development tool for PIC microcontroller which in this case will be the MPLAB IDE software. Meanwhile, the implementation of the program written earlier into the microcontroller chip is done using a JDM programmer with ICPROG program. Meanwhile for the PIC microcontroller chip, the model that I have chosen to use here will be the PIC16F876A model because it utilized almost all the I/O ports available.

Apart from all the above, instead of using a speed sensor as mention before we can instead obtain the speed signal from the vehicle's engine management computer this can be identified and accessed. Therefore, during the testing and demonstration session as mentioned earlier I will replace the speed sensor with a function generator to generate the speed signal which will be the AC sine wave signal. By varying the frequency of the AC sine wave signal generated, it will give us different car's speed. This AC sine wave signal generated is simply fed to the 1kΩ resistor at the speed input while the coil and also the magnet assembly will be removed. Figure 3.1 represent the general procedure to be taken for this project.

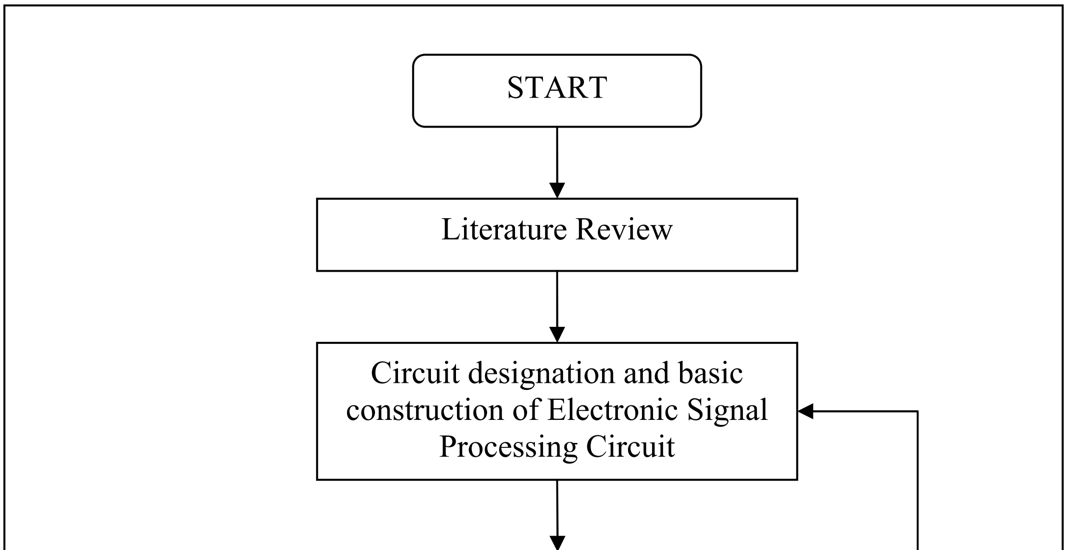

Figure 3.1: Flow chart showing the general procedure to be taken for this project

3.2 Tools Related to the Project

3.2.1 Hardware Testing and Constructing Tools

As briefed above, this project will involved both the hardware circuit's construction and programming of the PIC microcontroller chip. Therefore, I was required to use and utilized some of the equipments available in the laboratory such as function generator, multimeter, DC supply and also the oscilloscope especially during testing and troubleshooting the built circuit.

Apart from that, as for the circuit construction, tools such as solder and sucker, proto board, strip board and many others more, just to mention some here are needed in order to construct these circuit.

3.2.2 Programming Tools

3.2.2.1 ICPROG

ICProg is a free program that will take assembled code (filename ending in .HEX) and drive the programming circuit correctly to load the code into the PIC chip. ICProg was written by a freelance programmer, and although the interface looks simplistic, it is actually extremely well made. This program also has the advantage in that it can program many other types of chips, not just PIC chips and not just chips made by Microchip. In fact, it can program hundreds of chips.

3.2.2.2 MPLAB IDE

MPLAB Integrated Development Environment (IDE) is a free, integrated toolset for the development of embedded applications employing Microchip's PICmicro® and dsPIC® microcontrollers. MPLAB IDE runs as a 32-bit application on MS Windows®, is easy to use and includes a host of free software components for fast application development and super-charged debugging. MPLAB IDE also serves as a single, unified graphical user interface for additional Microchip and third party software and hardware development tools. Moving between tools is a snap, and upgrading from the free simulator to MPLAB ICD 2 or the MPLAB ICE emulator is done in a flash because MPLAB IDE has the same user interface for all tools.

3.2.2.3 JDM Programmer

The JDM programmer is a low-cost solution for getting a program into the internal flash memory of the PIC and in fact is one of the simplest PIC programmers available in the market. In this project, I built my own JDM Programmer in order for me to load my program into the internal flash memory of the PIC16F876A. Besides that, this programmer built by me here can be also used to program all the PIC16F87X microcontroller series. Figure 3.2 show the JDM Programmer Circuit.

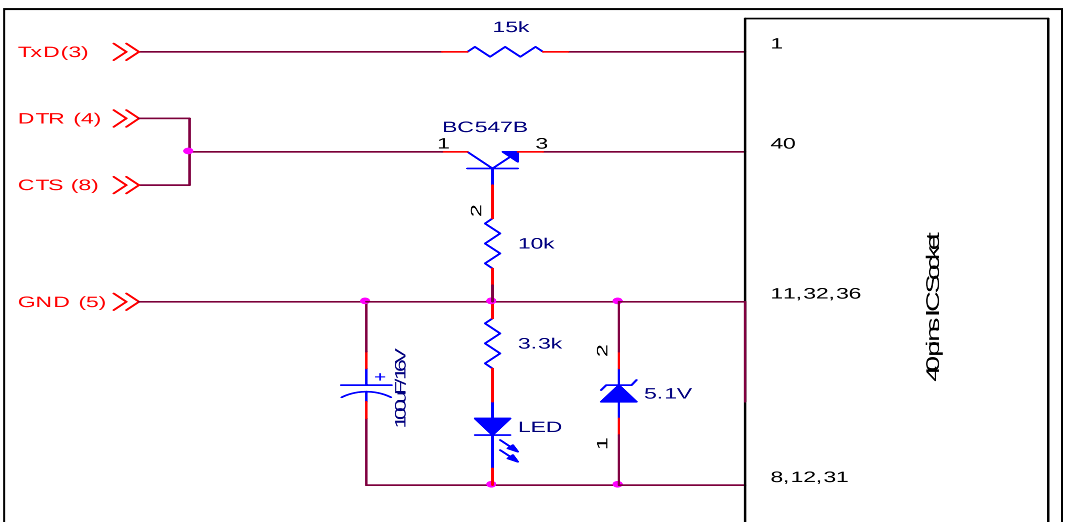

Figure 3.2: JDM Programmer Circuit

3.3 Hardware Circuits Development of the Project

3.3.1 +5V Voltage Regulator Circuit

In this circuit, a +12V rail is derived from the vehicle battery via the ignition switch in practical but in this project a DC Power Supply is being use to provide the +12V instead. The +5V Voltage Regulator Circuit is represented in Figure 3.3.

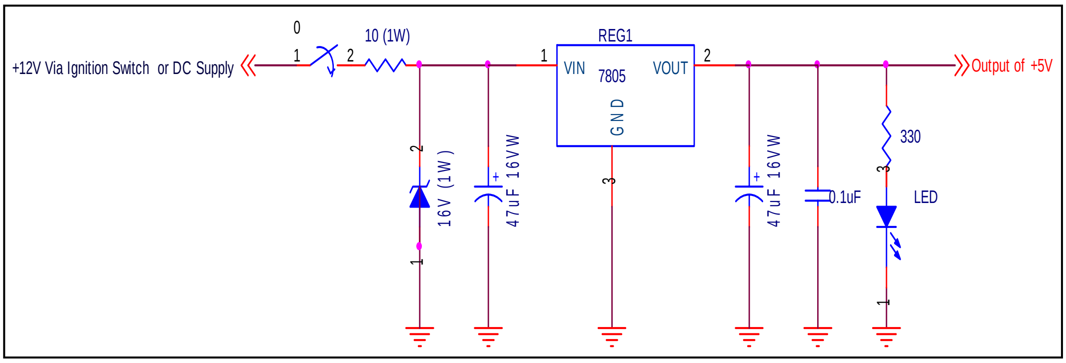

Figure 3.3: +5V Voltage Regulator Circuit

As shown in Figure 3.3, a 10Ω 1W resistor and 47µF decouple the supply, while a 16V (1W) zener diode will protects the circuit from transient voltage spikes above 16V. The decoupled ignition supply voltage is fed to regulator REG1, which provides a +5V rail. This rail is then used to power all the circuitry in this project with the exception of IC LM358 which is powered directly from the decoupled +12V rail. A second 47µF capacitor plus several 0.1µF capacitors are used to decouple the regulator's output.

3.3.2 Speed Sensor with Electronic Signal Processing Circuit

In this circuit, the speed sensor will consist of a coil which mounts on the chassis, plus two magnets which mount on drive shaft. As the magnets spin past, they will induce a voltage into the coil and this is detected by comparator stage LM358. Figure 3.4 show the Speed Sensor with Electronic Signal Processing Circuit.

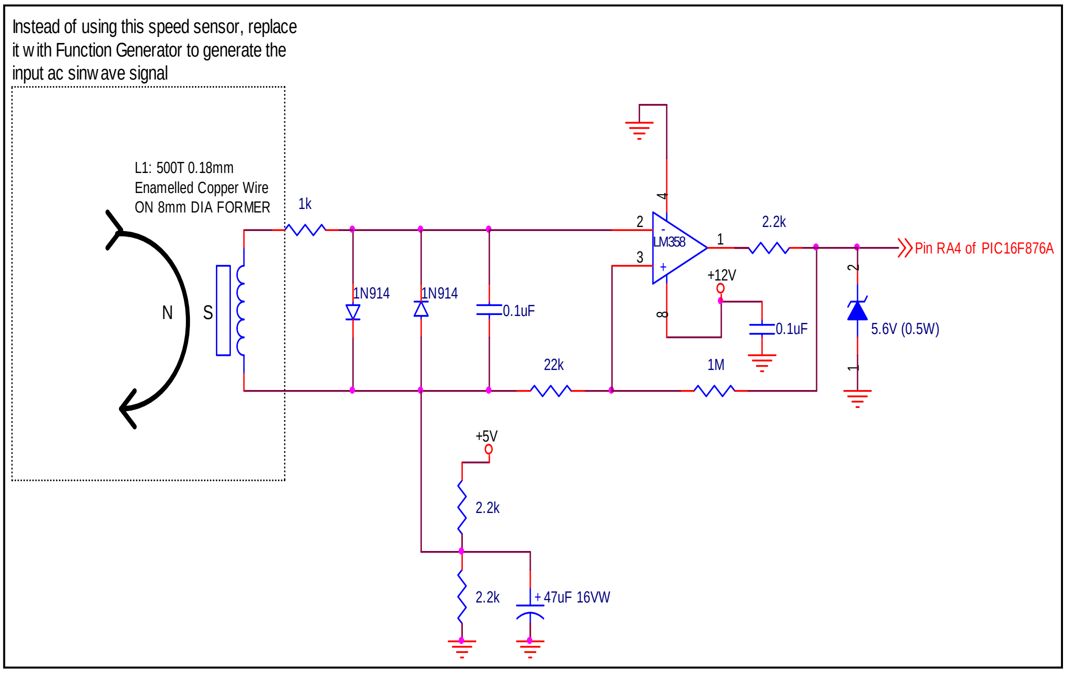

Figure 3.4: Speed Sensor with Electronic Signal Processing Circuit

Alternatively, by replacing the speed sensor with a function generator we can then generate the speed sensor signal which in this case will be a sine wave type even without applying it in the vehicle as the testing can be done in the laboratory. This can be easily done by setting the function generator to provide a 300mV_{RMS} or 0.424V_P sine wave output to the input or the coil input terminal of the electronic signal processing circuit as shown in Figure 3.4 above. In this case, LM358 will functions as an inverting comparator by processing the signal and amplified the received signal from the speed sensor to 10V pulse train and the output from IC2a is fed to the microcontroller chip via a 2.2kΩ limiting resistor. This is to convert the 10V pulse train to a +5V pulse train at the input of the microcontroller as the zener diode of 5.6V (0.5W) will clamp the voltage received from this circuit to a maximum of 5.6V.

In other words, the limiting resistor will work together with the zener diode of 5.6V (0.5W) in order to convert the 10V pulse train to a +5V pulse train and also to provide safety to the microcontroller chip as the 10V pulse train can destroy the microcontroller chip as we all know.

3.3.3 Speed Alarm with 4 Digits 7-Segment Display Circuit

In this circuit, the 4 digits 7-segment display is driven by the microcontroller in multiplex fashion. Figure 3.5 is related to the Speed Alarm with 4 Digits 7-Segment Display Circuit.

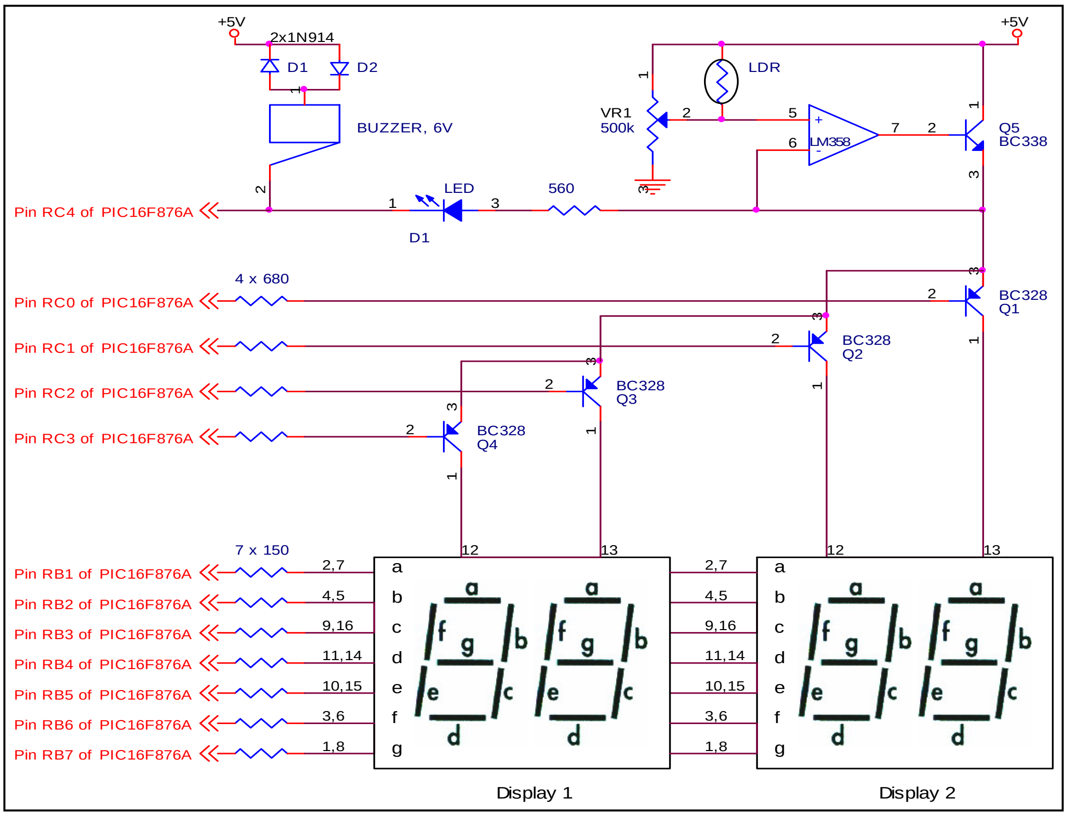

Figure 3.5: Speed Alarm with 4 Digits 7-Segment Display Circuit

As shown in Figure 3.5, the individual segment are driven directly from the microcontroller outputs via 150Ω current limiting resistors, while the individual display were drive by the microcontroller output via switching transistors of Q1-Q4 and also the 680Ω resistors. Besides that, the same LM358 as in Figure 3.4 was used to control the display brightness together with transistor Q5, LDR sensor and also the horizontal trimpot (500kΩ). Finally, the microcontroller wills also drives the alarm LED and the buzzer when the speed limit is exceeded. The two parallel diodes in series with the buzzer were used to prevent any low volume tone from being produced due to modulation of the microcontroller output.

3.3.4 12-Buttons Matrix Keypad with 74C922 Encoder Chip Circuit

On the other hand, as for the 12-Buttons Matrix Keypad with 74C922 Encoder Chip Circuit is shown in Figure 3.6.

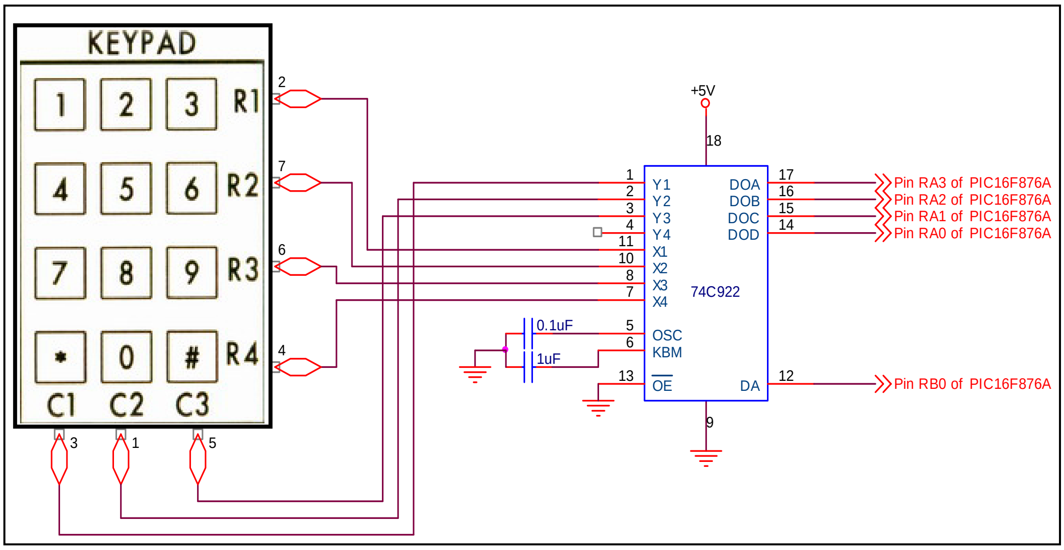

Figure 3.6: 12-Buttons Matrix Keypad with 74C922 Encoder Chip Circuit

A keypad encoder which is the MM74C922 was used in order to make the programming of the microcontroller much easier as later on during the programming of the microcontroller chip, we need not to take care the de-bouncing which occurred on the entire mechanical switch as mechanical consideration is contact bounce. Thus if using only normal connection without the encoder chip, then we have to do extra complicated programming to eliminate the de-bouncing while if using the encoder chip, de-bouncing were eliminate in this chip without us need to do extra works in programming.

The chip scans the keypad waiting for a key press. When a key is pressed pin 12 of the encoder chip will goes high. Then a 4 bit binary code corresponding to the key number as shown in Table 3.1 is sent to pins 14-17 of the chip thus making the programming of the microcontroller pretty straight forward, by writing a instruction to read the state of pin 12 of the MM74C922.

Table 3.1: Output of Keypad Encoder Chip MM74C922

Key Number	**Binary Code**			
	From A	***From B***	***From C***	***From D***
	To (RA3)	***To (RA2)***	***To (RA1)***	***To (RA0)***
'1'	0	0	0	0
'2'	0	0	1	0
'3'	0	0	0	1
'4'	1	0	0	0
'5'	1	0	1	0
'6'	1	0	0	1
'7'	0	1	0	0
'8'	0	1	1	0
'9'	0	1	0	1
'*'	1	1	0	0
'0'	1	1	1	0
'#'	1	1	0	1

As soon as the signal received from the pin 12 of the encoder chip is high, the microcontroller will read the states of from pin 14 to 17. First by reading the pin 17 (RA3) and pin 16 (RA2), the microcontroller will determine the row of the key that is being pressed before proceeding to read the pin 15 (RA1) and pin 14 (RA0) to determine the column of the key which is being pressed.

Finally by determining the row and column of which key is being pressed, automatically the key number of the pressed key can be obtain. After that, we will require to convert the 4 bit binary code which represents the key number to a decimal value using the microcontroller before saving the result as the set speed limit. As for the 0.1μF and 1μF capacitors, they are used for the scanning and de-bouncing times respectively. Meanwhile, pin 4 of MM74C922 was left unconnected as the keypad we are using here is a 3×4 matrix instead of a 4×4 matrix type.

3.3.5 Basic Microcontroller Circuit

Finally, we will need a basic microcontroller circuit where this is basically the fundamental circuit in order for the microcontroller to work. This Basic Microcontroller Circuit is shown in Figure 3.7.

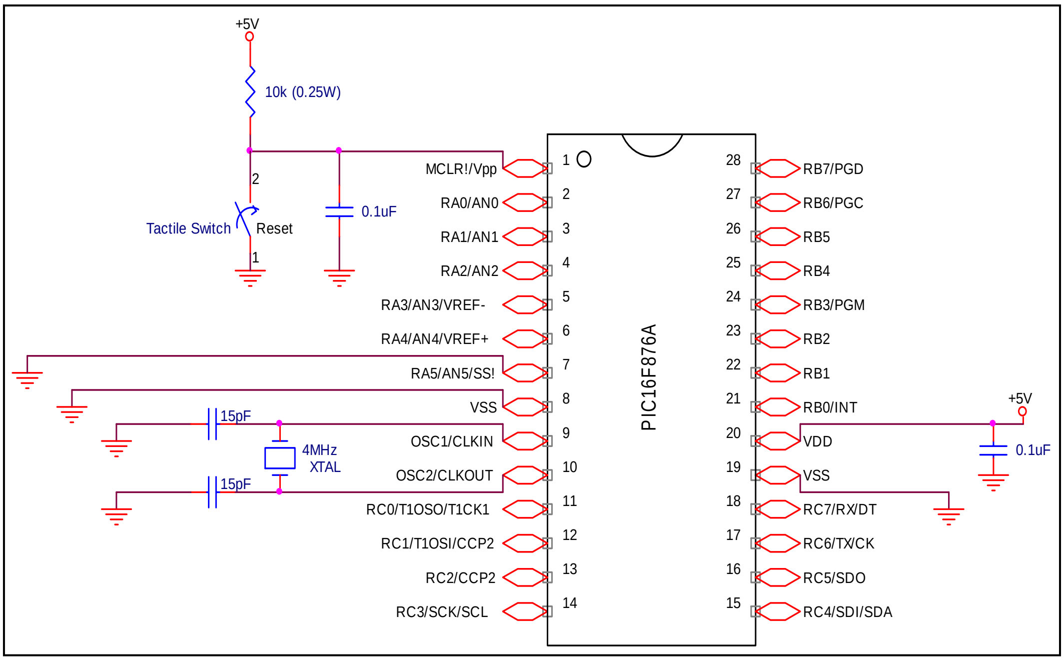

Figure 3.7: Basic Microcontroller Circuit

The most important part in this circuit will be the crystal oscillator which will provide pulses to the microcontroller, a +5V dc supply together with ground and also not to forget about the reset circuit connected to pin 1 of the PIC16F876A microcontroller. The reset circuit is important if the microcontroller program gone awry which might be cause by the hardware's short circuit and also others unforeseen circumstances.

3.3.6 Parts List

Before we can build any circuits, we will need the appropriate parts and components with the correct specification in order for the built circuit to work successfully. Therefore, in this section all components which are needed to build this project are listed out systematically started from Table 3.2 which shows the Parts List for JDM Programmer Circuit.

Table 3.2: Parts List for JDM Programmer Circuit

No.	**Type of Components**	**Description**	**Quantity**
1.	Resistor	15 kΩ	1
2.	Resistor	3.3 kΩ	1
3.	Resistor	10 kΩ	1
4.	LED	Green (5mm)	1
5.	Transistor	BC547B	1
6.	Capacitor	100 µF/16V (Electrolyte)	1
7.	Zener Diode	5.1 V (0.5W)	1
8.	Socket Pin	* Appropriate	3
9.	D-Connector	Male	1
10.	D-Connector	Female	2
11.	Strip Board	* Appropriate	1
12.	Jumper Wire	* Appropriate	-

Meanwhile, Table 3.3 shows the Parts List for +5V Voltage Regulator Circuit and Table 3.4 represent the Parts List for Speed Sensor with Electronic Signal Processing Circuit.

Table 3.3: Parts List for +5V Voltage Regulator Circuit

No.	Type of Components	Description	Quantity
1.	Resistor	10 Ω (1W)	1
2.	Resistor	330 Ω (0.25W, 1%)	1
3.	Voltage Regulator	5V 1A	1
4.	Capacitor	47μF 16VW (PC Electrolytic)	2
5.	Capacitor	0.1μF (MKT Polyester)	1
6.	LED	Green (5mm)	1
7.	Zener Diode	16V (1W)	1
8.	Slide Switch	SPDT Vertical	1
9.	Crocodile Clip	Red and Black	2
9.	Strip Board	* Appropriate	1
10.	Jumper Wire	* Appropriate	-

Table 3.4: Parts List for Speed Sensor with Electronic Signal Processing Circuit

No.	Type of Components	Description	Quantity
1.	Resistor	1 kΩ (0.25W, 1%)	1
2.	Resistor	2.2 kΩ (0.25W, 1%)	3
3.	Resistor	22 kΩ (0.25W, 1%)	1
4.	Resistor	1 MΩ (0.25W, 1%)	1
5.	Diode	1N914	2
6.	Zener Diode	5.6V (0.5W)	1
7.	Capacitor	47μF 16VW (PC Electrolytic)	1
8.	Capacitor	0.1μF (MKT Polyester)	2
9.	Dual Op-Amp	LM358(shared part with section 3.3.7.4)	1
10.	IC Socket	8 pins	1
11.	Strip Board	* Appropriate	1
12.	Jumper Wire	* Appropriate	-

As for the Parts List for Speed Alarm with 4 Digits 7-Segment Display Circuit is represented in Table 3.5 while Table 3.6 shows the Parts List for 12-Buttons Matrix Keypad with 74C922 Encoder Chip Circuit.

Table 3.5: Parts List for Speed Alarm with 4 Digits 7-Segment Display Circuit

No.	Type of Components	Description	Quantity
1.	Resistor	150 Ω (0.25W, 1%)	7
2.	Resistor	560 Ω (0.25W, 1%)	1
3.	Resistor	680 Ω (0.25W, 1%)	4
4.	Horizontal Trimpot	500 kΩ	1
5.	Diode	1N914	2
6.	Buzzer	6V	1
7.	LED	Red (5mm)	1
8.	LDR	Sensor	1
9.	Dual Op-Amp	LM358(shared part with section 3.3.7.3)	1
10.	7-Segment Display	Dual Digit	2
11.	Strip Board	* Appropriate	1
12.	Jumper Wire	* Appropriate	-

Table 3.6: Parts List for 12-Buttons Matrix Keypad with 74C922 Encoder Chip Circuit

No.	Type of Components	Description	Quantity
1.	Capacitor	0.1µF (Ceramic)	2
2.	Capacitor	15pF (Ceramic)	2
3.	Keypad	3x4 Matrix	1
4.	Encoder Chip	74C922	1
5.	IC Socket	18 pins	1
6.	Strip Board	* Appropriate	1
7.	Jumper Wire	* Appropriate	-

Lastly, will be Table 3.7 which will represent the Parts List for Basic Microcontroller Circuit.

Table 3.7: Parts List for Basic Microcontroller Circuit

No.	Type of Components	Description	Quantity
1.	Capacitor	0.1μF (Ceramic)	1
2.	Capacitor	1μF (Ceramic)	1
3.	Resistor	10 kΩ (0.25W)	1
4.	Tactile Switch	-	1
5.	IC Socket	28 pins	1
6.	Microcontroller	PIC 16F876A	1
7.	Crystal Oscillator	4 MHZ	1
8.	Strip Board	* Appropriate	1
9.	Jumper Wire	* Appropriate	-

3.3.7 PIC16F876A Microcontroller

As mentioned before, the controller use to control the whole project here is the PIC16F876A microcontroller. PIC16F876A is a 28 pins device and there are 3 I/O ports in this microcontroller namely port A, port B, and port C. Almost all the pins in each port are fully utilized here and Table 3.8 will summarize the use of all the I/O ports together with the pins distribution in this project.

Table 3.8: Usage of the microcontroller's port

Port	Pin	Usage
A	RA0-RA3	Digital Input from pin A-D of 12-Buttons Matrix Keypad with 74C922 Encoder Chip Circuit
	RA4	External Input Transition Signal for Timer0 from Speed Sensor with Electronic Signal Processing Circuit
B	RB0	Digital Input from Data Enable pin of 12-Buttons Matrix Keypad with 74C922 Encoder Chip Circuit
	RB1-RB7	Digital Output for segment a-g of the Speed Alarm with 4 Digits 7-Segment Display Circuit
C	RC0-RC3	Digital Output for 4 common anode of the Speed Alarm with 4 Digits 7-Segment Display Circuit
	RC4	Digital Output for buzzer of the Speed Alarm with 4 Digits 7-Segment Display Circuit

3.4 Software's Development of the Project

This section here will discuss on the programming aspect of the project. The whole programming of the PIC 16F876A microcontroller in this project is done using only assembly language. This project actually have a few subroutines and basically what are being discussed here are the 3 majors one while the others can be easily understand through the comment in Appendix C of Programming Source Code.

First of all, before we start to write the program, initialization of the microcontroller need to be done first where we need to initialize all the input and output of the microcontroller together with the features available in the microcontroller that we are going to applied such as TMR0, TMR1 and others through the microcontroller Special Function Registers. Among the Special Function Registers are the STATUS Register, OPTION_REG Register, INTCON Register, PIE1 Register, PIR1 Register, and many others which are related in order for me to develop my program for the project here.

In the main subroutine of the programming in this project will be the user input part which in this case will be the signal received from the 3×4 matrix keypad. In this case, polling method in programming is implemented in the main subroutine to scan whether there is any key that had been pressed from the keypad in which have been discussed in detail in section 3.3.4 regarding how the keypad communicate with the microcontroller. If the microcontroller didn't receive any signals from the 3×4 matrix keypad then it will keeps on displaying the actual vehicle speed and update the actual speed each second. Therefore, in Figure 3.8 is the Flow Chart of Adjustable Speed Limit Digital Speedometer with Speed Alarm for Main Subroutine which is generally on how the main subroutine is written.

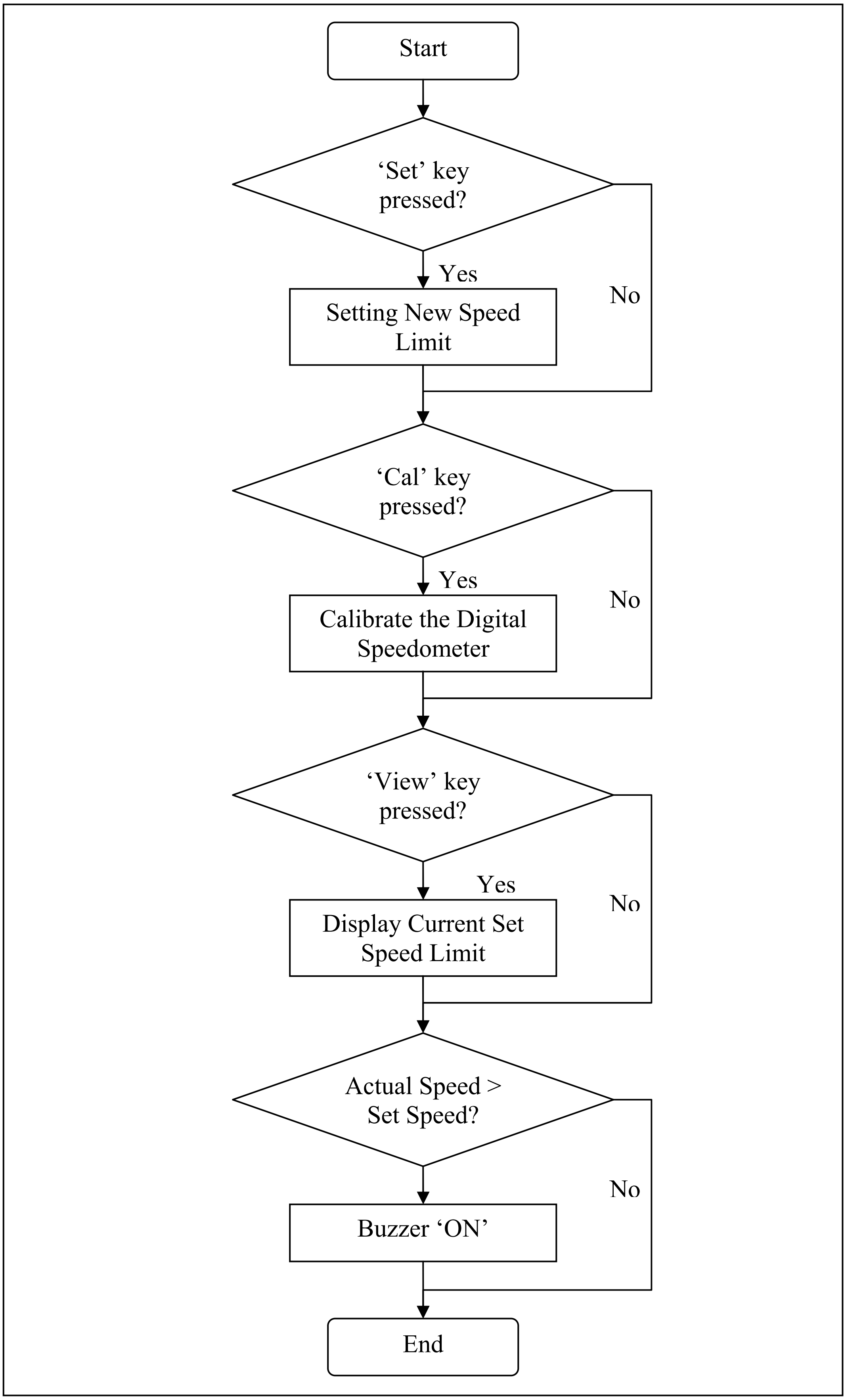

Figure 3.8: Flow Chart of Adjustable Speed Limit Digital Speedometer with Speed Alarm for Main Subroutine.

Apart from the main subroutine as mentioned above, basically the program is written to counts the number of pulses that came from the speed sensor or the speed signal from the vehicle's engine management computer if this can be identified over a fixed time period. The fixed time period in this program will be a second. Figure 3.9 represents the Flow Chart of Adjustable Speed Limit Digital Speedometer with Speed Alarm for Timer Interrupt Subroutine while Figure 3.10 shows the Flow Chart of Adjustable Speed Limit Digital Speedometer with Speed Alarm for Pulse Counter Subroutine. Both this flow chart will generally explain to us on how the subroutine is written.

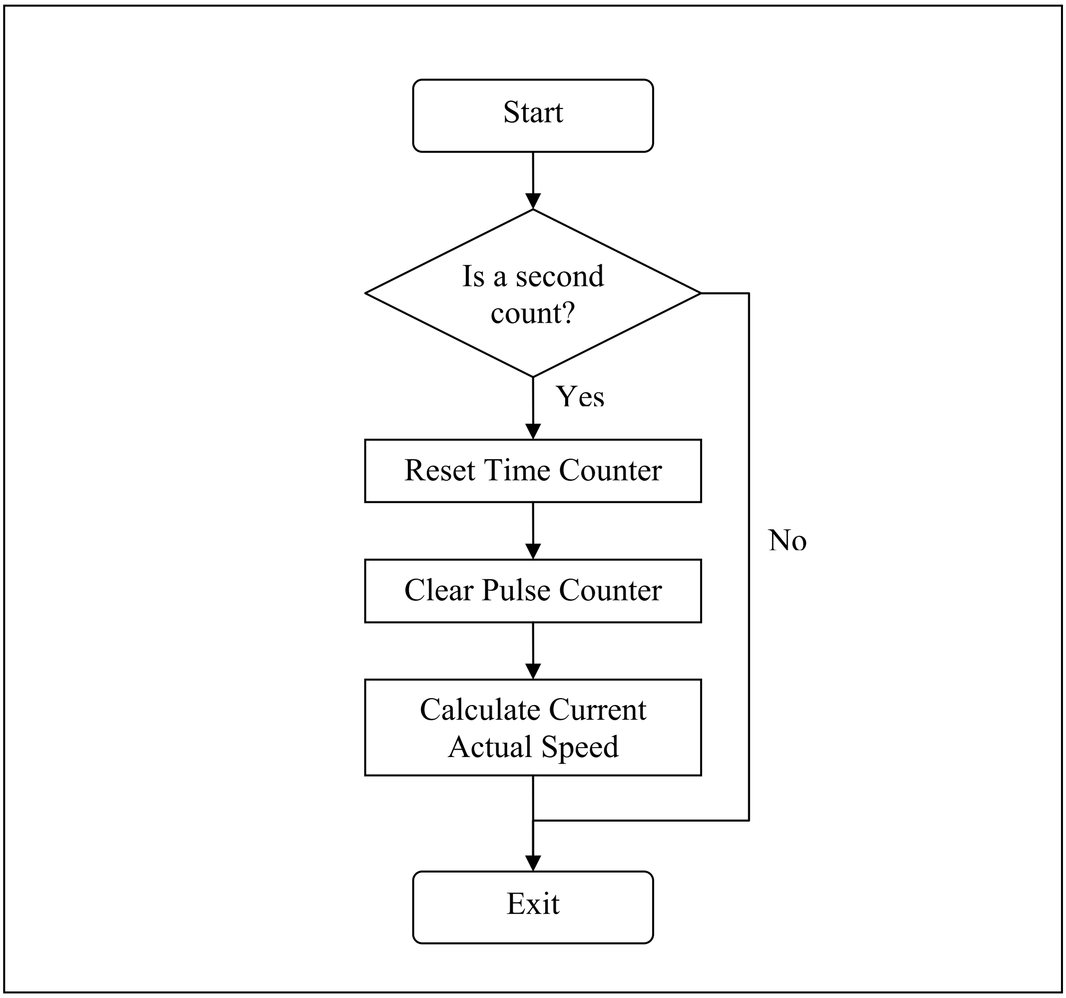

Figure 3.9: Flow Chart of Adjustable Speed Limit Digital Speedometer with Speed Alarm for Timer Interrupt Subroutine.

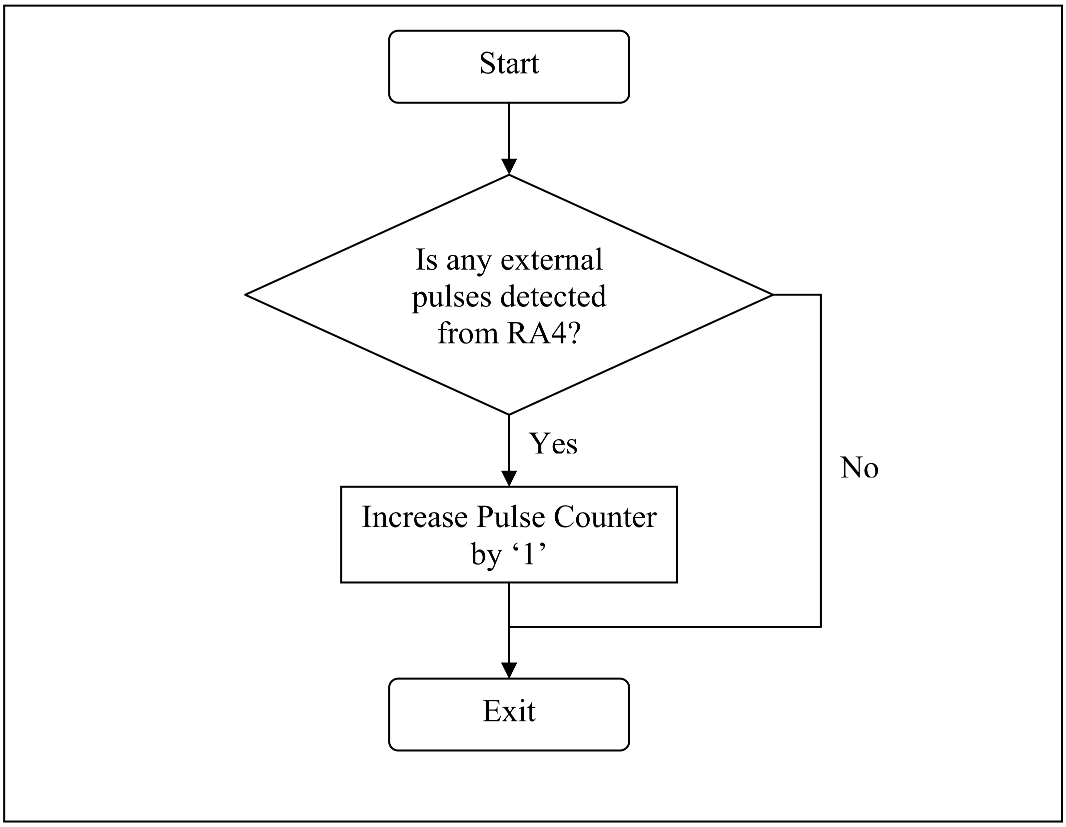

Figure 3.10: Flow Chart of Adjustable Speed Limit Digital Speedometer with Speed Alarm for Pulse Counter Subroutine.

The Timer Interrupt Subroutine as shown in the Figure 3.9 is implemented using the TMR1 available in the PIC 16F876A microcontroller together with another temporary register. As for the pulse counter subroutine, as shown in Figure 3.10. It is implemented using TMR0 as pulse counter together with another temporary register to store the counted pulses that is more than 256 pulses that came from the speed sensor or function generator.

Then, the total number of pulses that is counted in a second will be process by multiplying and dividing it with a calibration value. There are 25 different calibration values which are available in this model. The calibration value that is chosen or determined by the user will be stored permanently in EEPROM. This can only be change if we recalibrate it with another calibration value. During calibration, the 7-segment display will also display the output of ‘CALU’ which means that the model here is in calibrating state

The program is written in such that have different calibration value is because different type of vehicle and also numbers of magnet that are placed at the rotating shaft if we are using a speed sensor then it will generate different frequency value and with this, it will give us different number of pulses counted in a second which is the fixed period that is set earlier in the program.

As mentioned before, the counter period is fixed to a second and therefore the equation derived for total number of pulses counted in a second is as below:

$$Total\ Number\ Of\ \ Pulses\,(per\,\mathrm{sec}\,ond) = \frac{Fixed\ Period\,(s)}{T_{PO}\,(ms)}$$

(3.1)

Where the Fixed Period is always set to a second while T_{PO} is the Time Period variable for each pulse that can be obtained from the electronic signal processing circuit. The microcontroller will increase its pulse counter by one each time it detects a falling edge as shown in Figure 4.3 of Chapter IV from the input signal.

Therefore in order to determine the actual speed of any vehicle, different calibration value is needed in order to support all types of car's which have different frequency value generated, even though the entire cars is running at the same speed. The equation (3.2) will show the formula used to calculate the actual speed of the vehicle.

$$Speed(km/h) = \frac{Total\ Number\ Of\ \ Pulses \times 5km/h}{C}$$

(3.2)

Where C is the variable for different number of pulses counted per 5km/h which this C value might be different from one car to another and therefore equation (3.3) is related to the calibration value in this project.

$$Calibratio\,n\ Value = \frac{5km/h}{C}$$

(3.3)

The value of C can be varied from 1 to 25 and Figure 3.11 represents the different calibration value that is available in this model.

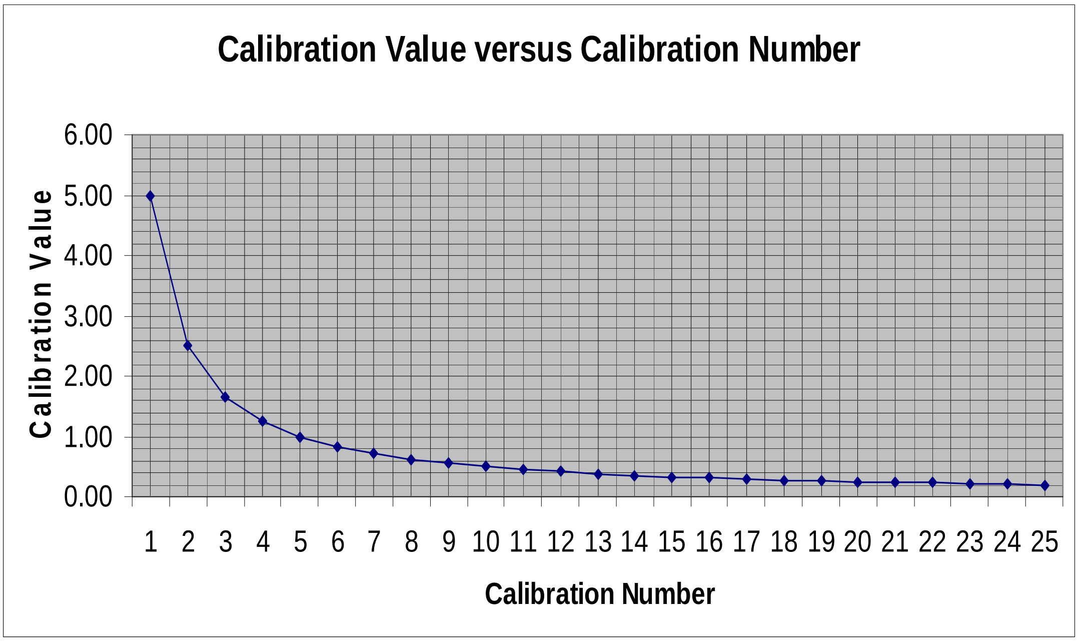

Figure 3.11: Calibration Value versus Calibration Number

As shown in Figure 3.11, for Calibration Number 1 the value for C is 1 while in Calibration Number 25, the maximum for C is 25 and by the default the value for C in this model are 5. The update time for displaying the actual speed of the car is a second meaning that the display will update the actual speed of the car via the 7-segment each one second and practically, for most of the cars if we are running at the frequency of 100 Hz then the actual car speed should be around 100km/h which means the value of C will be 5 if we are using 4 magnets at the driveshaft or the microcontroller should have counted pulses more or close to 100 during the one second period. When doubling the number of magnets at the driveshaft then we need to recalibrate the C value to 10 while reducing the number of magnets at the driveshaft to only 2, we need to have the C value set at 2 or 3 instead of 5 or 10.

More magnets will basically generate more pulses in a same fixed period and vice versa. As mentioned before, when the magnets spin past the coil, it will cut the fluxes that exist at the coil and by doing so, a voltage will be induced into the coil.

The type of voltage that is being induced here will be a sine wave type. Meanwhile in this project, the maximum speed that can be display is 256km/h. If more than the maximum speed of 256km/h then the 7-segment display here will display the output of '-Err' which means that actual vehicle speed cannot be obtain.

After the actual speed is obtained, the value must be converted to decimal form as the current actual speed is in the hexadecimal form and in order to do so, another subroutine is being written in order to do the conversion. The conversion from hexadecimal to decimal form is important in order to display it through the 7-segment display for the sake of convenience to the user.

Apart from that, a subroutine for the keypad input was also developed. As mentioned in section 3.3.4 where the keypad encoder is concerned, it will send out different binary code to represent the key that is pressed and basically the subroutine here will check the binary code that is detected from the keypad encoder chip to the input of the microcontroller and with the help of a look up table which is developed by me, the 4 digits 7-segment display will then display the actual number that have been pressed. In this case, the microcontroller will send out appropriate signal as in Table 4.1 of Chapter IV to the output ports which is connected to the Speed Alarm with 4 Digits 7-Segment Display Circuit to display the number which had been pressed by the user.

As for the speed alarm, the speed limit entered by the user will be stored in the EEPROM until we re-entered the speed limit with another value. The buzzer wills buzz continuously and the red LED will also come along only if the actual speed of the car detected is the same or more than the set speed limit being entered earlier by the user. The buzzer will only stopped buzzing and the red LED will go off once the actual speed detected is less than the set speed limit. Besides that, by setting the most significant digit more then '0' will automatically switch off the speed alarm. Storing the data in EEPROM or in another word storing the data in a non-volatile memory will prevents the data stored earlier being lost after we reset the microcontroller or cutting off the power supply to the whole Adjustable Speed Limit Digital Speedometer with Speed Alarm circuit model. Therefore, by storing the data in the EEPROM, this model here shall retain all data even if the power supply is removed or the speedometer is removed from the car.

3.4.1 User Guide for the Key's Function of 3×4 Matrix Keypad in this Project

0. Initially all the set speed limit is '0000' by default

1. Press '0' key, the default value for set speed limit will be shown.

2. Press ''*' key to reset the speed limit.

3. Press '*' key to reset the setting of speed limit if the initial input execution during step no.2 is wrong.
   E.g. If we want to input '0120' but when the actual we key in '013' before we key in the last digit which in this case is 0, we can reset the whole new input by pressing the '' key, then we will start the new input from the first digit.*
   While pressing the'#' key after any key between 0-9 will then exit the setup process and only update the previous digit entered before the '#' key is pressed.

4. Press '0' key to view the speed limit entered.

5. Press '*' key and followed by the '#' key will then calibrate the digital speedometer.

6. Pressing the '*' or '0' keys will leave the calibration mode and display the initial set speed value, while pressing the '#' key will recalibrate the speedometer.

7. Press '*' and followed by pressing any number greater than '0' for first digit or most significant digit will 'off' the speed alarm.

CHAPTER IV

RESULTS AND OBSERVATIONS

4.1 Input and Output Signal Observations of the Project

As this project is developed and designed in the laboratory using function generator to produced the AC sine wave as an input replacement for the speed sensor in this project. Therefore, it is important to observe all the signal involved starting right from the beginning which is signal generated from the function generator to the signal that feed into the PIC microcontroller. This entire signal can be observed and viewed only through the laboratory oscilloscope.

Therefore, Figure 4.1 shows the Conversion Process of Input Signal to Output Signal of the Project using function generator as input replacing the speed sensor.

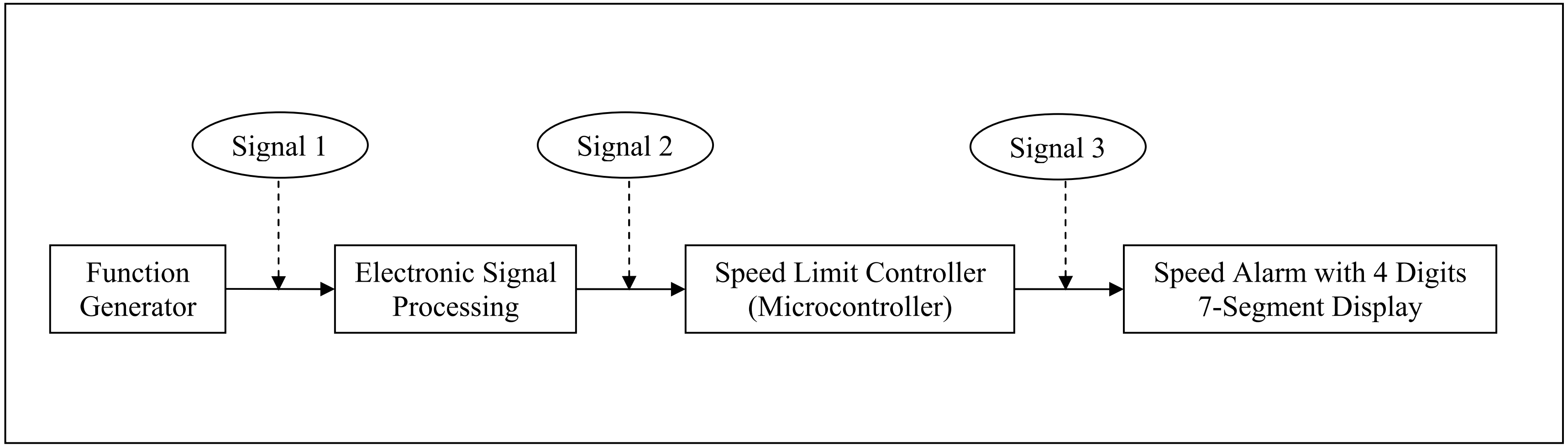

Figure 4.1: Conversion Process of Input Signal to Output Signal of the Project using Function Generator as Input replacing the Speed Sensor

where:

- Signal 1 is Input signal from Function Generator to Electronic Signal Processing Circuit
- Signal 2 is Input signal from Electronic Signal Processing Circuit to Microcontroller
- Signal 3 is Data being sent to the Speed Alarm with 4 Digits 7-Segment Display Circuit from the microcontroller after being processed by the microcontroller itself.

The Signal 1 and Signal 2 as in Figure 4.1 is obtained through the oscilloscope and is represented in Figure 4.2 and Figure 4.3 respectively as below.

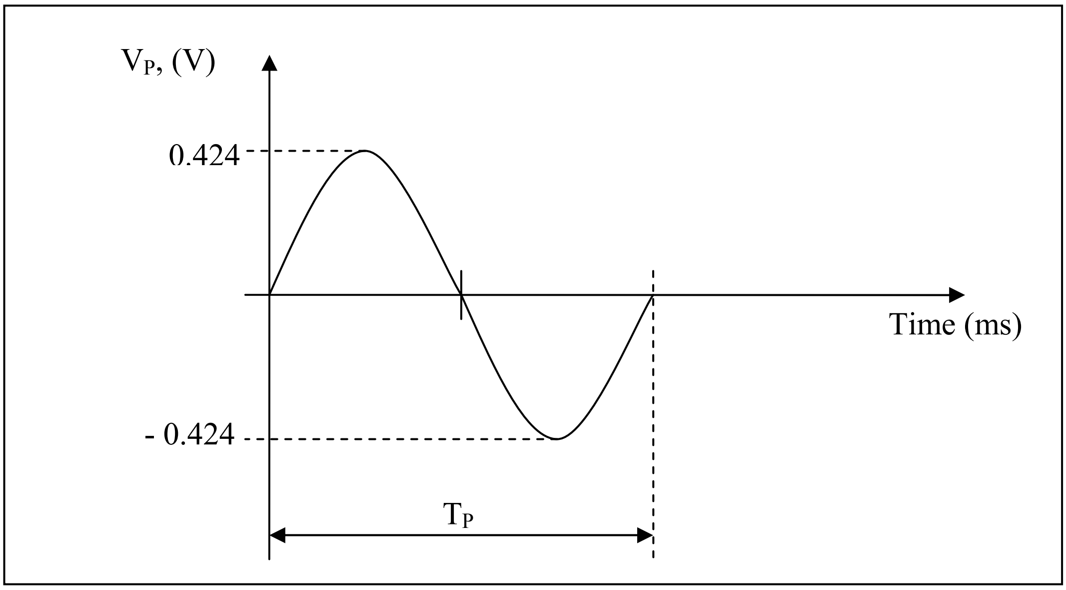

Figure 4.2: Signal 1 - Input signal from Function Generator to Electronic Signal Processing Circuit

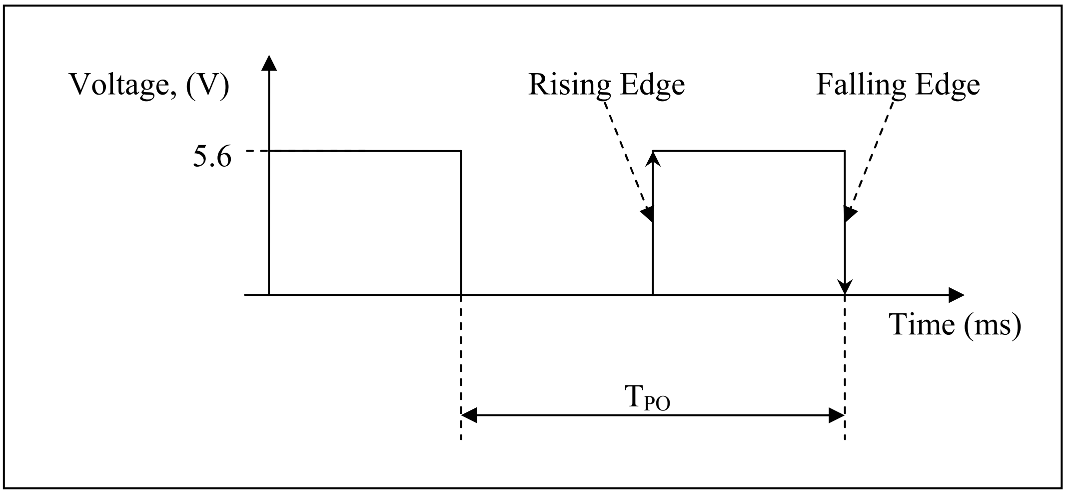

Figure 4.3: Signal 2 - Input signal from Electronic Signal Processing Circuit to Microcontroller

As shown in Figure 4.2 and Figure 4.3 respectively, both the variables for time period in Signal 1 is represented by 'T_P' while Signal 2 is represented by 'T_{PO}'. By obtaining the value for both the time period of both the signal, we can definitely obtain the frequency of the input by simply applying the simple equation as below where T_{PO} is actually equal to T_P.

$$f = \frac{1}{T_P}$$

(4.1)

Both the time period for Signal 1 and Signal 2 is actually the same because the Electronic Signal Processing circuit will only convert the sine wave signal to a pulse train signal which can only be applicable to the microcontroller. These have been widely discussed in Chapter III. The function generator should be set to provide a $300mV_{RMS}$ sine wave to Signal 1 which is also equal to 0.424 V_P or any voltage value as long as it is below 1 V_P. This is because in practical, the magnets are not strong enough to induce the coil in high voltage. In fact, by obtaining the value for $T_{PO,}$ we can also counts the number of pulses which is received by the microcontroller as explained in Chapter III before.

Meanwhile as for Signal 3, it will be data being sent by the microcontroller to the two dual digits 7-segment display. By sending output of 0V to the 7-segment it will 'ON' any one of the 7-segment display. As for Table 4.1 which in Figure 4.1 will be the Signal 3, shows the data being sent to the 7-segment display to display the actual speed of the vehicle. Apart from that, the microcontroller will also send the output of 0V to the common anode of any one of the 7-segment display on which digit which should be 'ON' as this project consists of 4 digits of 7-segment display.

Table 4.1: Signal 3 - Data being sent to the Speed Alarm with 4 Digits 7-Segment Display Circuit from the Microcontroller

Output from microcontroller to the 7-segment display							Display
Segment							
g	f	e	d	c	b	a	
1	0	0	0	0	0	0	‘0’
1	1	1	1	0	0	1	‘1’
0	1	0	0	1	0	0	‘2’
0	1	1	0	0	0	0	‘3’
0	0	1	1	0	0	1	‘4’
0	0	1	0	0	1	0	‘5’
0	0	0	0	0	1	0	‘6’
1	1	1	1	0	0	0	‘7’
0	0	0	0	0	0	0	‘8’
0	0	1	1	0	0	0	‘9’
1	0	0	0	1	1	0	‘C’
0	0	0	1	0	0	0	‘A’
1	0	0	0	1	1	1	‘L’
1	0	0	0	0	0	1	‘U’
0	1	1	1	1	1	1	‘-’
0	0	0	0	1	1	0	‘E’
0	1	0	1	1	1	1	‘r’

4.2 Comparison between Actual Vehicle Speeds being Displayed with Calculated Vehicle Speed

As mentioned before in Chapter III, the actual speed of the vehicle can be obtained using both the equation (3.1) and (3.2). Theoretically, the Fixed Period is set to 1 second but practically the microcontroller will not obtained exactly 1 second timing and in this project the microcontroller could only produce 1.04856 second instead of 1 second timing. This value can be calculated using the below equation:

$$Fixed\ Period(s) = \left(\frac{4MHz}{4}\right)^{-1} \times 65535 \times 8 \times 2 \qquad (4.2)$$

$$= 1.04856$$

Where 4MHz is the Oscillator Frequency being used, 65535 is the maximum value that TIMER1 module can support where TIMER1 module is a 16-bit timer/counter consisting of two 8-bit registers, 8 is the Prescale value that is selected for TIMER1 module and finally 2 is the number of times the microcontroller needed to execute TIMER1 module in order to obtained the timing of 1.04856s.

Therefore for all the results being obtained through testing in laboratory, the equation (4.2) is being use to calculate the Total Number of Pulses (per second).

$$Total\ Number\ Of\ Pulses\ (per\ second) = \frac{1.04856(s)}{T_{PO}(ms)} \qquad (4.3)$$

This is due to the nature of the microcontroller implementation concerning the clock frequency which in this case is 4 MHz that can only produce 1.04856s instead of the actual 1s. Meanwhile, as to calculate the actual speed of the vehicle, the same equation (3.2) can be applied.

4.2.1 Analysis of Results Obtained

This project will not be completed without any data being obtained through testing in the laboratory using the related equipment which in this case will be the DC Power Supply, Function Generator and also Oscilloscope.

Therefore, there will be 5 results being obtained using different calibration value where Analysis 1 is being done when the calibration value is equal to 5km/h /5, Analysis 2 is when the calibration value is equal to 5km/h /8, Analysis 3 is when the calibration value is equal to 5km/h /10, Analysis 4 is when the calibration value is equal to 5km/h /16 and finally Analysis 5 is being obtained when the calibration value is equal to 5km/h /25. Basically, by using different calibration value the speed of the vehicle is also different although the input frequency is the same.

After each analysis is being done, it will be followed by a graph being plot to relate the comparison between the input frequency and the speed of the vehicle for both the actual speed display and also the calculated actual speed.

4.2.1.1 Analysis 1: Calibration Value is 5km/h /5 where C=5

From Function Generator				Analysis of Data Being Feed Into Microcontroller		
Input Voltage (V_P)	Input Freq. (Hz)	T_P(ms)	Actual Freq.(Hz)	T_{PO} (Time/Pulse), ms	Number of Pulses/Sec	Calculated Ac (km/
0.424	0	0.00	0.00	0.00	0	0
	10	98.30	10.17	98.30	11	11
	20	51.00	19.61	51.00	21	21
	30	33.60	29.76	33.60	31	31
	40	24.80	40.32	24.80	42	42
	50	19.95	50.13	19.95	53	53
	60	16.65	60.06	16.65	63	63
	70	14.35	69.69	14.35	73	73
	80	12.40	80.65	12.40	85	85
	90	11.10	90.09	11.10	94	94
	100	10.06	99.40	10.06	104	104
	110	9.32	107.30	9.32	113	113
	120	8.26	121.07	8.26	127	127
	130	7.74	129.20	7.74	135	135
	140	7.34	136.24	7.34	143	143
	150	6.78	147.49	6.78	155	155
	160	6.32	158.23	6.32	166	166
	170	5.98	167.22	5.98	175	175
	180	5.64	177.30	5.64	186	186
	190	5.40	185.19	5.40	194	194
	200	5.12	195.31	5.12	205	205
	210	4.86	205.76	4.86	216	216
	220	4.65	215.05	4.65	225	225
	230	4.34	230.41	4.34	242	242
	240	4.13	242.13	4.13	254	254
	250	4.02	248.76	4.02	261	261

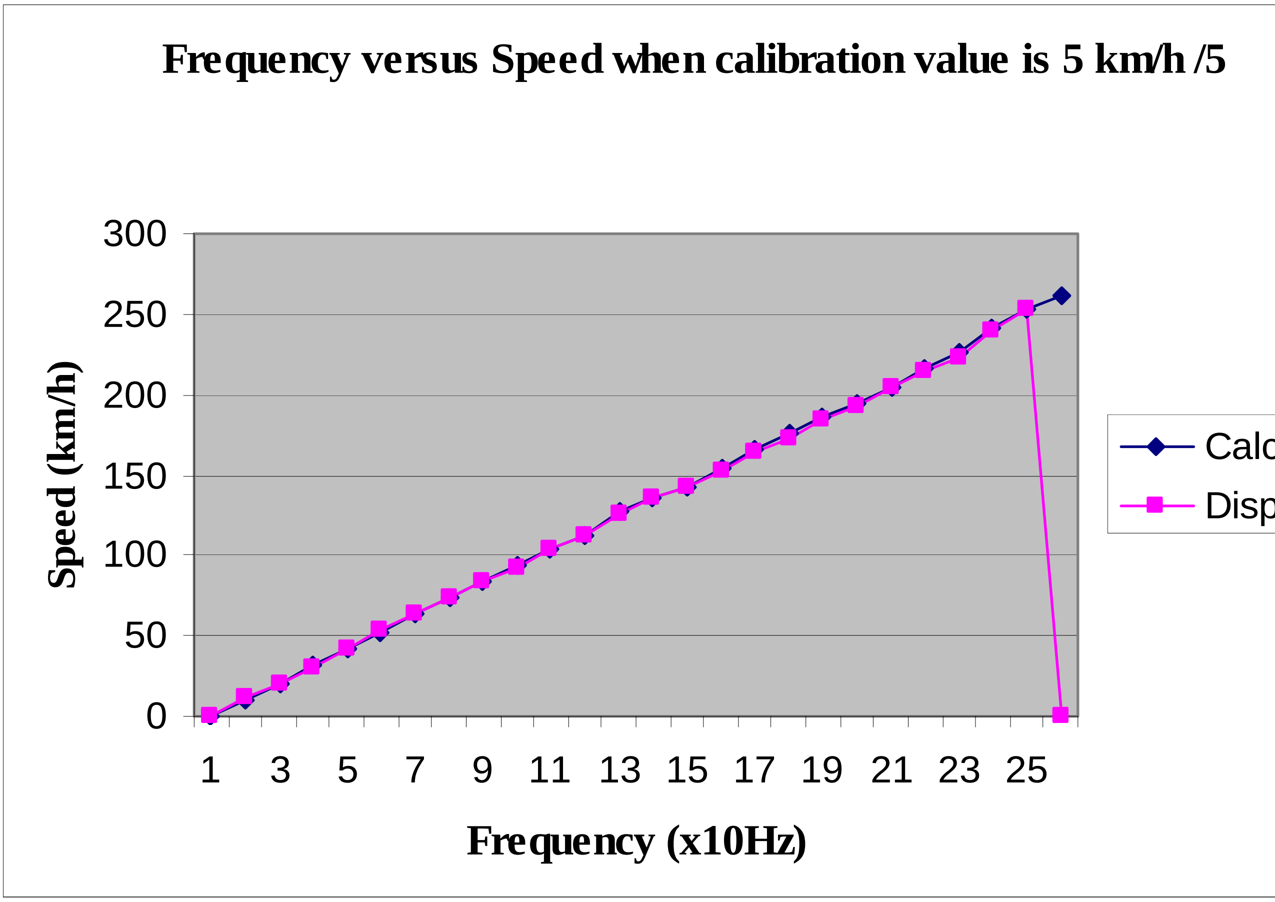

Figure 4.4: Input Frequency versus Speed when calibration value is 5km/h /5

4.2.1.2 Analysis 2: Calibration Value is 5km/h /8 where C=8

From Function Generator				Analysis of Data Being Feed Into Microcontroller		
Input Voltage (V_P)	Input Freq. (Hz)	T_P(ms)	Actual Freq.(Hz)	T_{PO} (Time/Pulse), ms	Number of Pulses/Sec	Calculated A (km
0.424	0	0.00	0.00	0.00	0	0
	10	99.80	10.02	99.80	11	7
	20	50.60	19.76	50.60	21	13
	30	33.80	29.59	33.80	31	19
	40	24.60	40.65	24.60	43	27
	50	19.90	50.25	19.90	53	33
	60	16.55	60.42	16.55	63	40
	70	14.30	69.93	14.30	73	46
	80	12.50	80.00	12.50	84	52
	90	11.08	90.25	11.08	95	59
	100	10.00	100.00	10.00	105	66
	110	9.28	107.76	9.28	113	71
	120	8.04	124.38	8.04	130	82
	130	7.66	130.55	7.66	137	86
	140	7.30	136.99	7.30	144	90
	150	6.78	147.49	6.78	155	97
	160	6.24	160.26	6.24	168	10
	170	5.94	168.35	5.94	177	11
	180	5.63	177.62	5.63	186	11

	190	5.41	184.84	5.41	194	12
	200	5.06	197.63	5.06	207	
	210	4.87	205.34	4.87	215	13
	220	4.59	217.86	4.59	228	14
	230	4.30	232.56	4.30	244	15
	240	4.09	244.50	4.09	256	16
	250	3.98	251.26	3.98	263	16

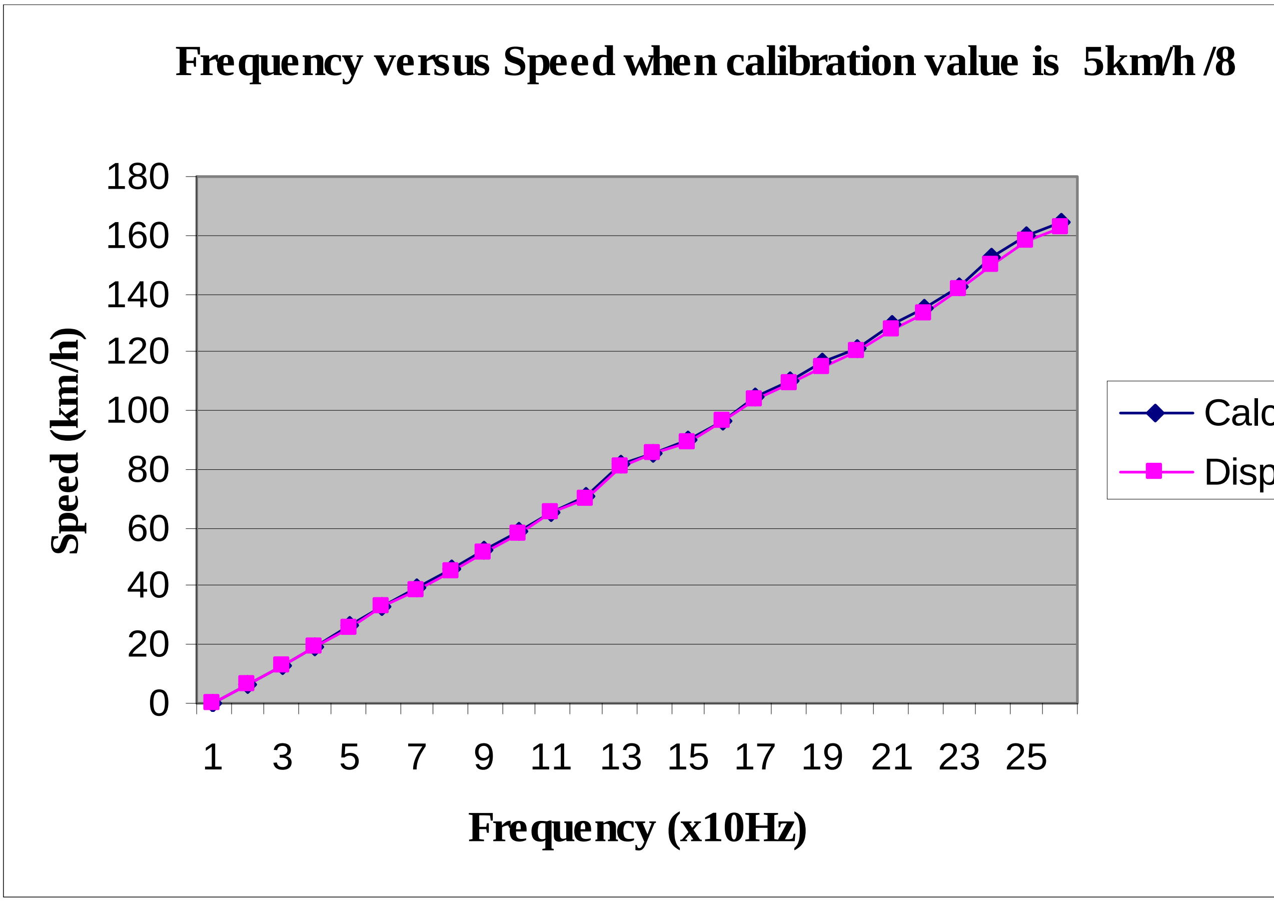

Figure 4.5: Input Frequency versus Speed when calibration value is 5km/h /8

4.2.1.3 Analysis 3: Calibration Value is 5km/h /10 where C=10

From Function Generator				Analysis of Data Being Feed Into Microcontroller		
Input Voltage (V_P)	Input Freq. (Hz)	T_P(ms)	Actual Freq.(Hz)	T_{PO} (Time/Pulse), ms	Number of Pulses/Sec	Calculated A… (km/…
0.424	0	0.00	0	0.00	0	0
	10	100.00	10.00	100.00	10	5
	20	50.90	19.65	50.90	21	10
	30	33.40	29.94	33.40	31	16
	40	24.85	40.24	24.85	42	21
	50	19.75	50.63	19.75	53	27
	60	16.45	60.79	16.45	64	32
	70	14.30	69.93	14.30	73	37

	80	12.50	80.00	12.50	84	42
	90	11.10	90.09	11.10	94	47
	100	10.00	100.00	10.00	105	52
	110	9.10	109.89	9.10	115	58
	120	8.30	120.48	8.30	126	63
	130	7.80	128.21	7.80	134	67
	140	7.05	141.84	7.05	149	74
	150	6.80	147.06	6.80	154	77
	160	6.35	157.48	6.35	165	83
	170	5.92	168.92	5.92	177	89
	180	5.62	177.94	5.62	187	93
	190	5.42	184.50	5.42	193	97
	200	5.06	197.63	5.06	207	10
	210	4.88	204.92	4.88	215	10
	220	4.64	215.52	4.64	226	11
	230	4.32	231.48	4.32	243	12
	240	4.10	243.90	4.10	256	12
	250	4.02	248.76	4.02	261	13

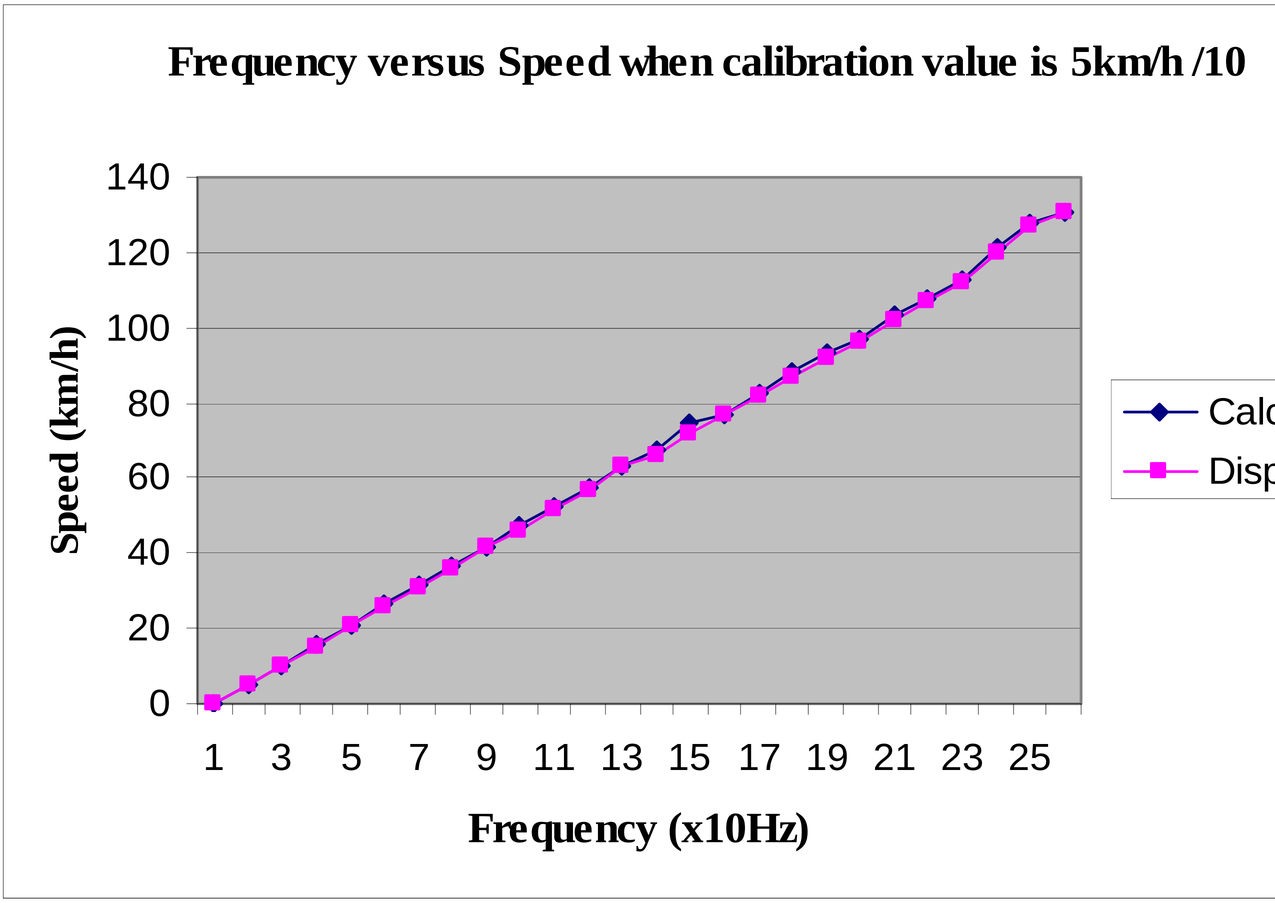

Figure 4.6: Input Frequency versus Speed when calibration value is 5km/h /10

4.2.1.4 Analysis 4: Calibration Value is 5km/h /16 where C=16

From Function Generator				Analysis of Data Being Feed Into Microcontroller		
Input Voltage	Input	T_P(ms)	Actual	T_{PO} (Time/Pulse),	Number of	Calculated A

(V_P)	Freq. (Hz)		Freq.(Hz)	ms	Pulses/Sec	(km
0.424	0	0.00	0.00	0.00	0	0
	10	100.00	10.00	100.00	10	3
	20	50.70	19.72	50.70	21	6
	30	33.70	29.67	33.70	31	10
	40	24.55	40.73	24.55	43	13
	50	19.60	51.02	19.60	53	17
	60	16.45	60.79	16.45	64	20
	70	14.20	70.42	14.20	74	23
	80	12.45	80.32	12.45	84	26
	90	11.12	89.93	11.12	94	29
	100	10.02	99.80	10.02	105	33
	110	9.28	107.76	9.28	113	35
	120	8.00	125.00	8.00	131	41
	130	7.88	126.90	7.88	133	42
	140	7.32	136.61	7.32	143	45
	150	6.76	147.93	6.76	155	48
	160	6.25	160.00	6.25	168	52
	170	5.97	167.50	5.97	176	55
	180	5.67	176.37	5.67	185	58
	190	5.38	185.87	5.38	195	61
	200	5.08	196.85	5.08	206	65
	210	4.85	206.19	4.85	216	68
	220	4.59	217.86	4.59	228	71
	230	4.33	230.95	4.33	242	76
	240	4.10	243.90	4.10	256	80
	250	3.98	251.26	3.98	263	82

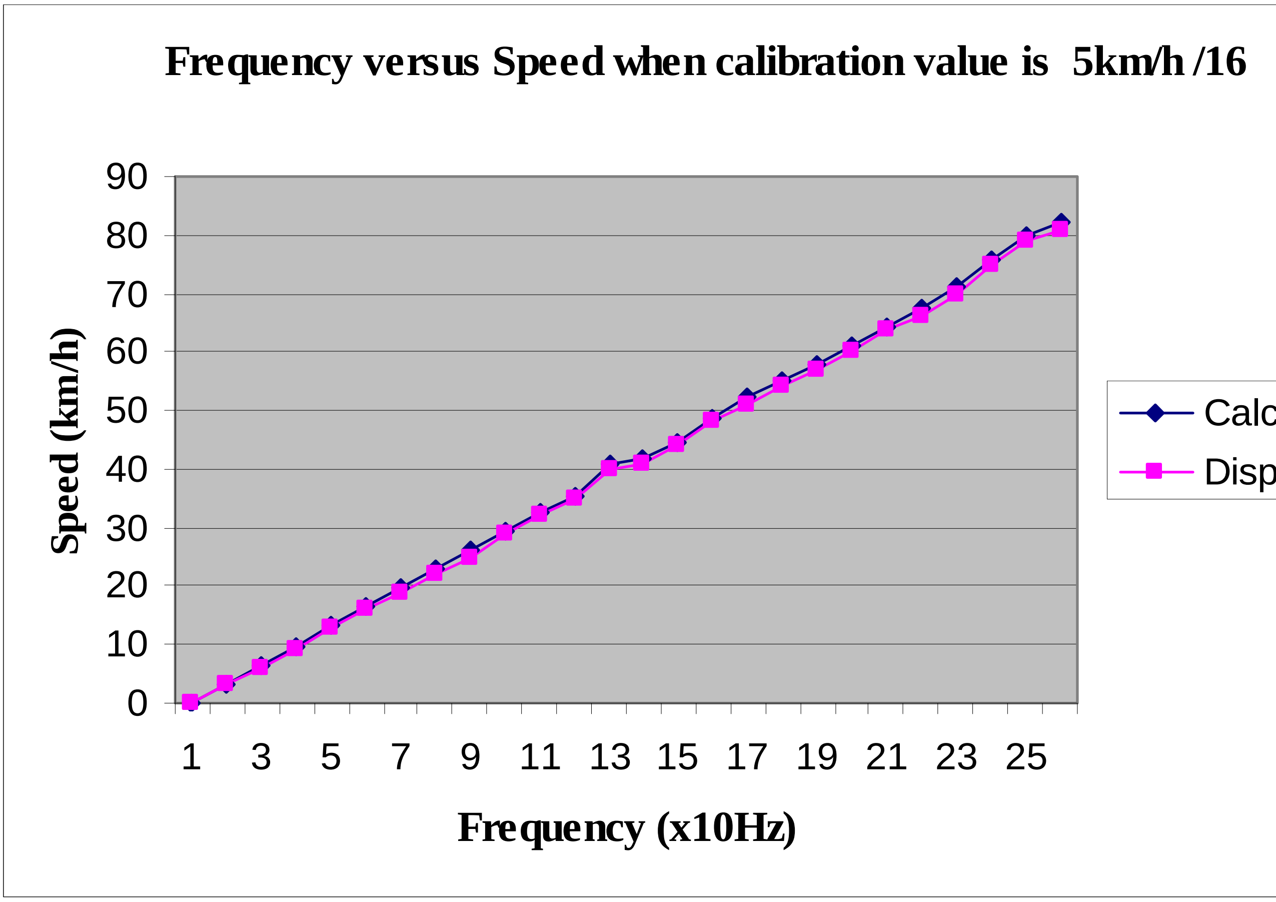

Figure 4.7: Input Frequency versus Speed when calibration value is 5km/h /16

4.2.1.5 Analysis 5: Calibration Value is 5km/h /25 where C=25

From Function Generator				Analysis of Data Being Feed Into Microcontroller		
Input Voltage (V_P)	Input Freq. (Hz)	T_P(ms)	Actual Freq.(Hz)	T_{PO} (Time/Pulse), ms	Number of Pulses/Sec	Calculated A (km/
0.424	0	0.00	0.00	0.00	0	0
	10	100.00	10.00	100.00	10	2
	20	50.90	19.65	50.90	21	4
	30	33.70	29.67	33.70	31	6
	40	24.75	40.40	24.75	42	8
	50	19.70	50.76	19.70	53	11
	60	16.45	60.79	16.45	64	13
	70	14.15	70.67	14.15	74	15
	80	12.50	80.00	12.50	84	17
	90	11.12	89.93	11.12	94	19
	100	9.96	100.40	9.96	105	21
	110	9.08	110.13	9.08	115	23
	120	8.26	121.07	8.26	127	25
	130	7.66	130.55	7.66	137	27
	140	7.32	136.61	7.32	143	29
	150	6.70	149.25	6.70	157	31
	160	6.42	155.76	6.42	163	33
	170	5.94	168.35	5.94	177	35

	180	5.62	177.94	5.62	187	37
	190	5.34	187.27	5.34	196	39
	200	5.10	196.08	5.10	206	41
	210	4.86	205.76	4.86	216	43
	220	4.68	213.68	4.68	224	45
	230	4.36	229.36	4.36	240	48
	240	4.17	239.81	4.17	251	50
	250	4.00	250.00	4.00	262	52

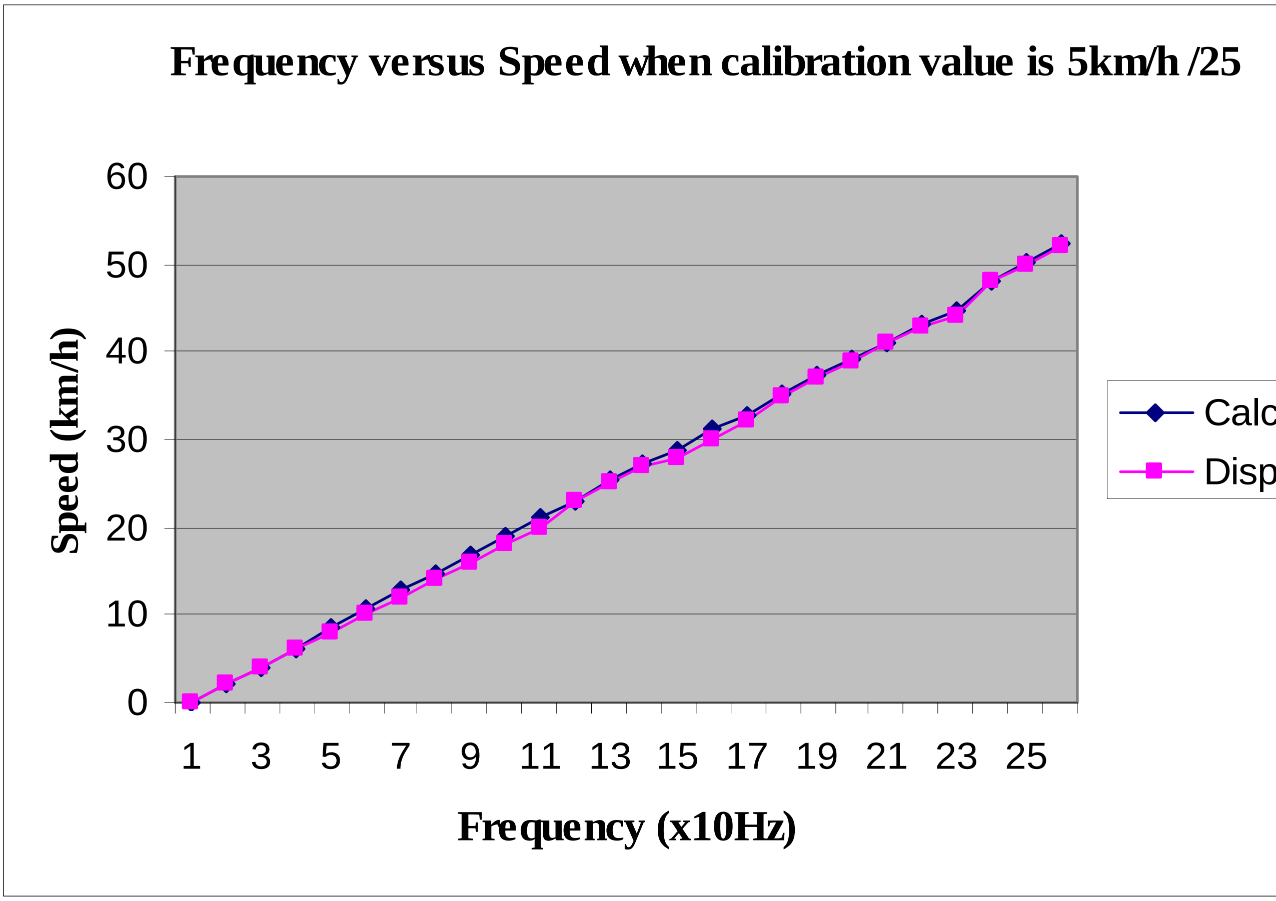

Figure 4.8: Input Frequency versus Speed when calibration value is 5km/h /25

CHAPTER V

DISCUSSIONS, CONCLUSIONS AND RECOMMENDATIONS

5.1 Discussions and Conclusions

We may notice that in our vehicle's analog speedometer is not very linear as compared to this Adjustable Speed Limit Digital Speedometer with Speed Alarm model that I developed here. These have been proved previously in Chapter IV where in all the experiments being done, although different calibration values are being used. The results obtained as being plotted concludes that this model being built here is indeed a very linear speedometer and in fact the actual speed display obtained is also very close to the calculated actual speed.

As this model of Adjustable Speed Limit Digital Speedometer with Speed Alarm is being developed and designed in the laboratory using Function Generator to generate the AC sine wave as an input replacement for the speed sensor and magnets which are use in practice. There might be flaw in displaying speed which is quite low. This is because in actual case, the magnets need to rotate at a reasonably fast rate before they induce voltage pulses of sufficient amplitude in the pickup coil for reliable processing by the Electronic Processing Circuit. This flaw probably can be eliminated by applying stronger magnets.

Besides that, one must also pay attention during installation of magnets to the driveshaft where the magnets must all install with the same polarity facing outwards. This can be checked by attaching the magnets together in a stack and it will either give a N-S-N-S stack or a S-N-S-N stack. This is important as it will practically generate the AC sine wave.

Apart from that, the behavior and performance of this Adjustable Speed Limit Digital Speedometer with Speed Alarm model here have been analyzed and discussed. The results obtained show that this built model here has the capability of:

i. Displaying the actual speed of the vehicle itself in the km/h reading.
ii. Displaying the desired speed limit entered by the user in digital mode.
iii. Having an adjustable speed limit controller as an input for the setting of the speed limit.
iv. Having a speed alarm that buzzes continuously when the speed limit is exceeded.

5.2 Recommendations

Although the model of Adjustable Speed Limit Digital Speedometer with Speed Alarm that I built here is successful and fulfilled all the scopes but there are quite some area of improvement which can be done in terms of either it is the software's development area or even to the hardware circuit. In term of software's development which is obviously the programming of the PIC microcontroller, this model can be improved to display speed which is more than 256km/h. Besides that, one can also try to use timing as the calibration method instead of the current method being used here where it is fixed to a second, which means the timing can be vary and thus can eliminate the calibration value being used in the current method.

Apart from that as for the hardware improvement, in order to made this model more advance is to upgrade this model in a way that it can also display the engine speed in the RPM mode as the reading in RPM is important especially during the fine tuning of the car engine. This can be only done by having another circuit to obtain the signal from the car ignition coil and also another circuit to process the signal received. As these will involve another input signal thus an extra input port to the microcontroller is needed in order to process the signal. This is because in reality, if the car is not moving then the speed of the car is 0km/h but the engine speed is not equal to 0 RPM and therefore conversions from km/h to RPM mode can't be apply. The same programming concept as to display the vehicle's speed in km/h can be applied here to display the engine speed in RPM mode.

Last but not least, this model here can be redesign and rebuild it in the form of PCB board instead of using strip board as being done here to make it look more attractive than the present one.

REFERENCES

1. Mohammad Asri Sulaiman (1992). “Digital Speedometer.” Universiti Teknologi Malaysia: Tesis PSM.

2. Rudy Timbang (2003). “Digital Speedometer.” Universiti Teknologi Malaysia: Tesis PSM.

3. Silicon Chip’s (2003). “Electronics Projects For Cars Vol. 2.” Silicon Chip Publication Pty Ltd.

4. Allan Bonnick (2001). “Automotive Computer Controlled Systems.” 1st. ed. Butterworth Heinemann.

5. Frank D. Petruzella (1991). “Automotive Electronic Fundamentals.” Macmillan/McGraw-Hill.

6. Rudolf F.Graf (1997). “Measuring Circuit.” Newnes.

7. Maplin Series (1995). “Auto Electronic Projects.” Newnes.

8. Tom Denton (2000). “Automobile Electrical and Electronic Systems.” 2nd. ed. Arnold Publisher.

9. William B.Ribbens (1998). “Understanding Automotive Electronics.” 5th . ed. Newnes.

10. Mel Sladdin and Alan Johnson (1995). “Elementary Electronics.” 2nd. ed. Hadder and Stroughton.

11. David G.Alciatore and Michael B.Histand (2003). “Introduction to Mechatronics and Measuring Systems.” 2nd. ed. McGraw-Hill.

12. Universiti Teknologi Malaysia (1999). “Panduan Menulis Tesis.” Edisi Ketiga. Universiti Teknologi Malaysia

13. Myke Predko (2002). “Programming and Customizing PICmicro® Microcontrollers.” 2nd. ed. McGraw-Hill.

14. Microchip (2001). “PIC16F87XA Data Sheet.” Microchip Technology Inc.

15. www.fke.utm.my

16. www.psz.utm.my

17. www.microchip.com

18. www.josepino.com

19. www.mikroelektronika.co.yu

UNIVERSITI TEKNOLOGI MALAYSIA
Research Management Centre

PRELIMINARY IP SCREENING & TECHNOLOGY ASSESSMENT FORM
(To be completed by Project Leader submission of Final Report to RMC or whenever IP protection arrangement is required)

1. **PROJECT TITLE IDENTIFICATION** :

Automatic Speed Limit Digital Speedometer With Speed Alarm

Vote No: 75225

2. **PROJECT LEADER** :

Name : **Anita binti Ahmad**

Address : **Faculty of Electrical Engineering, Universiti Teknologi Malaysia, 81310 Skudai Johor**

Tel : **07-5535326** Fax : **07-5566272** e-mail : **anita@fke.utm.my**

3. **DIRECT OUTPUT OF PROJECT** *(Please tick where applicable)*

Secientific Research	Applied Research	Product/Process Development
☐ Algorithm	☐ Method/Technique	☐ Product / Component
☐ Structure	☑ Demonstration / Prototype	☐ Process
☐ Data		☐ Software
☐ Other, please specify	☐ Other, please specify	☐ Other, please specify
________________	________________	________________
________________	________________	________________
________________	________________	________________

4. **INTELLECTUAL PROPERTY** *(Please tick where applicable)*

- ☐ Not patentable
- ☑ Patent search required
- ☐ Patent search completed and clean
- ☐ Invention remains confidential
- ☐ No publications pending
- ☐ No prior claims to the technology
- ☐ Technology protected by patents
- ☐ Patent pending
- ☐ Monograph available
- ☐ Inventor technology champion
- ☐ Inventor team player
- ☐ Industrial partner identified

5. LIST OF EQUIPMENT BOUGHT USING THIS VOT

-Nil

6. STATEMENT OF ACCOUNT

a)	APPROVED FUNDING	RM : ……-……………………
b)	TOTAL SPENDING	RM : ……-……………………
c)	BALANCE	RM : ……-……………………

7. TECHNICAL DESCRIPTION AND PERSPECTIVE

Please tick an executive summary of the new technology product, process, etc., describing how it works. Include brief analysis that compares it with competitive technology and signals the one that it may replace. Identify potential technology user group and the strategic means for exploitation.

a) Technology Description

A model of an Adjustable Speed Limit Digital Speedometer with Speed Alarm is developed in the laboratory using a PIC microcontroller which will be equipped with a 12-buttons keypad, a two dual digits 7-segment display and also a buzzer. A custom made programming was developed using assembly language for the PIC microcontroller which will be the key of the whole project. The built model of an Adjustable Speed Limit Digital Speedometer with Speed Alarm will able to display the actual running speed of the vehicle in km/h reading and also the desired speed limit entered by the user in digital mode while the secondary purpose is to have an adjustable speed limit controller as an input for the setting of the speed limit together with a speed alarm that buzzes continuously when the speed limit being set is exceeded by the current running vehicle speed.

b) Market Potential

For automotive industries

c) Commercialisation Strategies

We have contacted some automotive company and still discussing the potential of this research in their company.

8. RESEARCH PERFORMANCE EVALUATION

a) FACULTY RESEARCH COORDINATOR

	Excellent	Very Good	Good	Satisfactory	Fair	Weak
Research Status	( )	( )	( )	( )	( )	( )
Spending	( )	( )	( )	( )	( )	( )
Overall Status	( )	( )	( )	( )	( )	( )

Comment/Recommendations :

____

____

____

____

____

____

…………………………………………

Signature and stamp of
JKPP Chairman

Name : ……………………………………

Date : ……………………………………

b) RMC EVALUATION

	Excellent	Very Good	Good	Satisfactory	Fair	Weak
Research Status	( )	( )	( )	( )	( )	( )
Spending	( )	( )	( )	( )	( )	( )
Overall Status	( )	( )	( )	( )	( )	( )

Comments :-

____

____

____

____

____

____

Recommendations :

☐ Needs further research

☐ Patent application recommended

☐ Market without patent

☐ No tangible product. Report to be filed as reference

……………………………………….. Name : …………………………….………………

Signature and Stamp of Dean / Deputy Dean
Research Management Centre

Date : ……………………………………………

Lampiran 20

UTM/RMC/F/0024 (1998)

UNIVERSITI TEKNOLOGI MALAYSIA

BORANG PENGESAHAN
LAPORAN AKHIR PENYELIDIKAN

TAJUK PROJEK : AUTOMATIC SPEED LIMIT DIGITAL SPEEDOMETER WITH SPEED ALARM

Saya **ANITA BINTI AHMAD**

(HURUF BESAR)

Mengaku membenarkan **Laporan Akhir Penyelidikan** ini disimpan di Perpustakaan Universiti Teknologi Malaysia dengan syarat-syarat kegunaan seperti berikut :

1. Laporan Akhir Penyelidikan ini adalah hakmilik Universiti Teknologi Malaysia.
2. Perpustakaan Universiti Teknologi Malaysia dibenarkan membuat salinan untuk tujuan rujukan sahaja.
3. Perpustakaan dibenarkan membuat penjualan salinan Laporan Akhir Penyelidikan ini bagi kategori TIDAK TERHAD.
4. * Sila tandakan (/)

[ ]	SULIT	(Mengandungi maklumat yang berdarjah keselamatan atau Kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972).
[ ]	TERHAD	(Mengandungi maklumat TERHAD yang telah ditentukan oleh Organisasi/badan di mana penyelidikan dijalankan).
[/]	TIDAK TERHAD	

TANDATANGAN KETUA PENYELIDIK

Nama & Cop Ketua Penyelidik

Tarikh : 10 NOVEMBER 2006

CATATAN : ** Jika Laporan Akhir Penyelidikan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali sebab dan tempoh laporan ini perlu dikelaskan sebagai SULIT dan TERHAD.*