

UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS♦

JUDUL: CONTACTLESS TACHOMETER

SESI PENGAJIAN: 2009/2010

Saya MUHAMMAD IZZAT B ZAKARIAH (860626-33-5209)
(HURUF BESAR)

mengaku membenarkan tesis (Sarjana Muda/~~Sarjana~~ /~~Doktor Falsafah~~)* ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Universiti Malaysia Pahang (UMP).
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
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

Disahkan oleh:

(TANDATANGAN PENULIS)

(TANDATANGAN PENYELIA)

Alamat Tetap:

**5/983 KAMPUNG PELINDUNG TASIK,
26100, BESERAH KUANTAN,
PAHANG**

RUHAIZAD BIN ISHAK
(Nama Penyelia)

Tarikh: **30 NOVEMBER 2010**

Tarikh: : **30 NOVEMBER 2010**

- CATATAN:
- * Potong yang tidak berkenaan.
 - ** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai atau TERHAD.
 - ♦ Tesis dimaksudkan sebagai tesis bagi Ijazah doktor Falsafah dan Sarjana secara Penyelidikan, atau disertasi bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (PSM).

“I hereby acknowledge that the scope and quality of this thesis is qualified for the award of the
Bachelor Degree of Electrical Engineering (Electronics)”

Signature : _____

Name : RUHAIZAD BIN ISHAK

Date : 30 NOVEMBER 2010

CONTACTLESS TACHOMETER

MUHAMMAD IZZAT BIN ZAKARIAH

This thesis is submitted as partial fulfillment of the requirements for the award of the
Bachelor of Electrical Engineering (Electronics)

Faculty of Electrical & Electronics Engineering
Universiti Malaysia Pahang

NOVEMBER, 2010

“All the trademark and copyrights use herein are property of their respective owner. References of information from other sources are quoted accordingly; otherwise the information presented in this report is solely work of the author.”

Signature : _____

Author : MUHAMMAD IZZAT BIN ZAKARIAH

Date : 30 NOVEMBER 2010

To my beloved family and friends,
My father and my mother,
Zakariah Bin Hashim
And
Siti Hadzrah Binti Yusoff

ACKNOWLEDGEMENT

First of all, I would like to thanks to Allah SWT because give me strength to finish this project. And my project supervisor, En. Ruhaizad Bin Ishak, for the guidance and enthusiasm given throughout the progress of this project. He has always assisted me when I handling my project. Besides, I would like to express my sincere appreciation for his valuable advices, guidance and encouragement. This has inspired me to be more confident in trying new things.

My appreciation also goes to my family who has been so tolerant and supports me all these years. Thanks for their encouragement, love and emotional supports that they had given to me.

Special thanks to staff FKEE, who have given me a great help in accomplishing this project.

Last but not least, I would like to say millions of thanks to all my course mates especially my best friend Jeyna, Amrik, Syazwan, Fauzi, Yazid, Daim and those who has lending me their helping hand directly or indirectly with this project.

Thank you.

ABSTRACT

A tachometer is a device that measures the rotation speed of a shaft or disk, as in a motor or other machine. In automotive use, it is used as a gauge showing the speed (RPM) of the engine shaft that is driving the transmission, usually in thousands of rotations per minute. What makes this device special is that it can very accurately measure the rotational speed of a shaft without even touching it. This is very interesting when making direct contact with the rotating shaft is not an option or will reduce the velocity of the shaft, giving faulty readings. This device is built on a microcontroller, an alpha-numeric LCD module, a battery and a proximity sensor or an infrared to detect the rotation of the shaft whose speed is being measured. If we were using proximity sensor, the counted pulses will detect any reflective element passing in front of it, and thus, will give an output pulse for each and every rotation of the shaft. But if we were using infrared, we will put the infrared on both shaft and the tachometer. Those pulses which we get from every rotation of the shaft will be fed to the microcontroller and counted.

ABSTRAK

Takometer adalah alat yang mengukur kelajuan pusingan suatu poros atau disk, seperti pada motor mesin yang lain. Dalam penggunaan otomotif, ia digunakan sebagai ukuran yang menunjukkan kelajuan dari aci mesin yang mendorong penghantaran, biasanya dalam ribuan putaran per menit atau rotation per speed (RPM). Apa yang membuatkan peranti ini khas ialah ia dapat mengukur kelajuan putaran aci bahkan tanpa menyentuhnya dengan sangat tepat. Hal ini sangat menarik ketika melakukan kontak langsung dengan poros berputar bukanlah suatu pilihan atau akan mengurangkan kelajuan dari aci, mahkan ia akan memberikan pembacaan yang salah. Peranti ini dibina di atas mikro kawalan, sebuah modul LCD alfa-numerik, bateri dan sensor jarak atau infra merah untuk mengesan putaran aci kelajuan yang sedang diukur. Jika kita menggunakan sensor jarak, pulsa yang dikira akan mengesan setiap elemen reflektif lalu di depannya, dan dengan demikian, akan memberikan keluaran pulsa untuk masing-masing dan setiap pusingan aci. Tapi kalau kita menggunakan infra merah, kami akan meletakkan infra merah pada kedua-dua aci dan takometer tersebut. Pulsa yang kita dapatkan dari setiap pusingan aci akan dimasukkan ke mikro kawalan dan dikira.

TABLE OF CONTENTS

CHAPTER	TITLE	PAGE
	DECLARATION	ii
	DEDICATION	iii
	ACKNOWLEDGEMENTS	iv
	ABSTRACT	v
	ABSTRAK	vi
	TABLE OF CONTENTS	vii
	LIST OF FIGURES	x
	LIST OF TABLES	xi
	LIST OF APPENDICES	xii
1.0	INTRODUCTION	
1.1	Background of study	1
1.2	Problem Statement	2
1.3	Objective	2
1.4	Scope of Project	3
1.5	Thesis Overview	3

2.0	LITERATURE REVIEW	
2.1	Introduction	5
2.2	Tachometer	5
	2.2.1 Digital Tachometer	5
	2.2.2 Contactless Analogue Tachometer	7
2.3	Sensor	8
	2.3.1 Infrared Sensor	9
2.4	Microcontroller	10
2.5	MicroCode Studio	11
2.6	Proteus 7 Professional	12
3.0	METHODOLOGY	
3.1	Introduction	14
3.2	Instantaneous Measurement Algorithm	15
3.3	The Proximity Sensor	16
3.4	PIC 16F877 Microcontroller	17
3.5	The Electronic Circuit	18
3.6	Software Implementation	21
3.7	The Flow Chart	30
4.0	RESULTS AND DISCUSSION	
4.1	Introduction	33
4.2	Table of Results	35
4.3	Discussion	36
5.0	CONCLUSION	
5.1	Conclusion	37
5.2	Future Recommendation	38
REFERENCES		39

LIST OF FIGURE

FIGURE	TITLE	PAGE
2.1	A Digital Tachometer	6
2.2	Schematic Non Contact Analogue Tachometer	8
2.3	Infrared sensor	9
2.4	An example of MicroCode Studio	12
2.5	An example of Proteus 7 Professional	13
3.1	Block diagram of contactless tachometer	14
3.2	Software algorithm for instantaneous measurement	15
3.3	IR LEDs positioning	17
3.4	PIC16F877 microcontroller pin configuration	18
3.5	Data of 16F877 microcontroller	18
3.6	The Microcontroller Board	19
3.7	IR proximity sensor	20
3.8	Connection of the sensor to the main board	21
3.9	Flow chart of the project progress	31
3.10	Flow chart of the contactless tachometer	32
4.1	Contactless Tachometer Diagram by Using Proteus	33
4.2	Contactless tachometer	34
4.3	Fan with reflecting marking tape	34
4.4	Contactless tachometer with fan	34

LIST OF TABLES

TABLE	TITLE	PAGE
4.1	Speed VS Time	35
4.2	Average RPM for each fan speed	36

LIST OF APPENDICES

APPENDIX	TITLE	PAGE
A	PIC Pin Diagram	41
B	PIC Block Diagram	42
C	PIC Pinout Description	43

CHAPTER 1

INTRODUCTION

1.1 BACKGROUND OF STUDY

A contact-less tachometer will let you know how quickly something spins and is frequently used for buses, trains, tractors, trucks, cars and planes. This non contact tachometer version uses a sensor that will sense revolutions through pulses. A contact-less tachometer consists of a shaft encoder and electronic circuits. The output of the shaft encoder provides electric pulses. The frequency of these pulses is proportional to the rotational speed. A speed signal is obtained by processing the pulses from the encoder using an additional electronic circuit. When the wheel or shaft rotates, it has a mirror or a tab that obstructs the path of the light every time it revolves. Then, there is simply a chip to count the number of obstructions per minute. This one is extremely accurate and can handle some of the highest speeds.

This paper proposes a new solution for the processing of the pulses from the encoder to derive the speed signal. The solution takes advantages of new single chip low cost programmable microcontrollers. It is shown how to use hardware and software combined with a suitable method for the speed evaluation to design a high performance tachometer. These can work like an optical sensory as you point it like a laser at what you want to measure.

1.2 PROBLEM STATEMENT

A tachometer typically use a rotating target attached to a wheel, gearbox or motor. This target may contain magnets, or it may be a toothed wheel. The teeth on the wheel vary the flux density of a magnet inside the sensor head. The probe is mounted with its head a precise distance from the target wheel and detects the teeth or magnets passing its face. One problem with this system is that the necessary air gap between the target wheel and the sensor allows ferrous dust from the vehicle's under frame to build up on the probe or target, inhibiting its function.

In the other words, the normal tachometer requires physical contact between the instrument and the device being measured. In applications where this is not feasible for technical or safety reasons, it may be possible to use a contactless tachometer to take measurements from a distance. This contactless tachometer is not only useful in terms of safety, but it is also very efficient. The efficiency depends on both the proximity sensor and the reflective element passing in front of it. Thus, it is clear that the method of contactless tachometer is a technique that worthy of being developed.

1.3 OBJECTIVE

- i. To prove that this contactless tachometer is more efficient than normal tachometer.
- ii. To develop and create a new reliable contactless tachometer.

1.4 SCOPE OF PROJECT

In order to achieve the objective of the project, there are several scope had been outlined. The scope of this project includes using MicroCode Studio to program microcontroller PIC 16F877A, design the circuit by using the Proteus software and build hardware for the system. The main goal of this project is to determine the revolutions per minute of the motor speed by using this non contact tachometer. The scope of this project is:

- i. To show that contactless tachometer is more efficiency when using the proximity sensor because of there is no contact between the motor and the tachometer. Besides, there will be no harm to the people who were using this contactless tachometer because of the safety that was proven.
- ii. The 16F877A microcontroller, proximity sensor and alpha-numeric LCD module that been used on this project is to detect the rotation of the shaft whose speed which is being measured.

1.5 THESIS OVERVIEW

This thesis is primarily concerned with the analysis and simulation of the contact-less tachometer. All the work done in this project is presented in 5 chapters:

Chapter 1 it discuss about the objective and scope of this project as long as summary of works outlines

Chapter 2 will discuss more on theory and literature reviews that have been done. It well discuss about types of tachometer, the sensor that been used, and the software that can be used to get the speed reading of the motor.

Chapter 3 the discussion will be on the methodology hardware and software implementation of this project. This chapter includes the flow of the project

development and the flow of the programming used in the project. This is one of the most essential part of the project as it determines the whether the flow of the project is smooth or otherwise.

Chapter 4 outlines the results of the speed motor reading by using this non contact tachometer. This is important to determine whether the objective of this project is achieved or not

Chapter 5 discuss the conclusion of this project, summarizes the overall project design and future work that can be done.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

This chapter includes the study of different types of tachometer, sensor, microcontroller PIC 16F877, MicroCode Studio and Proteus 7 Professional.

2.2 TACHOMETER

2.2.1 DIGITAL TACHOMETER

A tachometer is an essential part in the design of the feedback loop in the speed control of AC and DC drives. DC tachometers are spread used due to their good dynamic performances. However, the reasons listed below encourage the use of digital tachometers:

- i. a better accuracy,
- ii. in the case of a digital controller, no A/D conversion is needed.
- iii. no maintenance is needed, as digital tachometers are brushless,
- iv. noise immunity, which avoids filtering

The electronic circuit of a digital tachometer is composed of two parts: the encoder interface and the speed measure block which is shown in Figure 2.1. The encoder interface can be programmed in conventional logic. The speed is measured from the pulse train coming from the encoder, which has m marks equally spaced at the circumference. Two methods are normally employed to measure the speed: counting the pulses in a fixed period of time, and measuring the time elapsed between successive pulses.

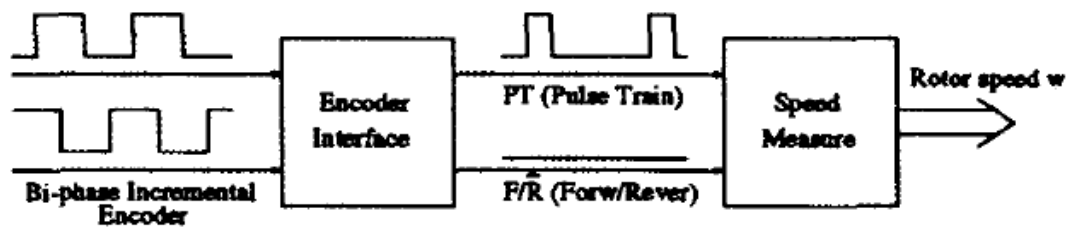


Figure 2.1: A Digital Tachometer

The analysis of these two methods leads to the following conclusions. The pulse counting method is suitable for medium and high speeds, but the relative error dramatically increases with lower speeds. On the other hand, measuring the elapsed time between two (or more) pulses exhibits a high accuracy in the low speed range, at the cost of a poor response in higher speeds, or a poor dynamic response in lower speeds. In reference [1], the author proposed the so called *constant elapsed time method*. In essential, it measures the elapsed time between k successive pulses, and dynamically adjusts the value of k to obtain a near constant response time. This method was implemented on a microprocessor. A new method is proposed that provides high accuracy in a wide speed range with good dynamic performances. The circuit is implemented in hardware using only one low-cost FPGA and one EPROM device. Human beings are faced with oil and coal depletion of fossil fuels such as a serious threat that these fossil fuels is a one-time non-renewable resources, limited reserves and a large amount of combustion of carbon dioxide, causing the Earth's warming, deterioration of the ecological environment. With the development of society, energy saving and environmental protection has become a topical issue [2].

2.2.2 CONTACTLESS ANALOGUE TACHOMETER

The tachometer employs a standard cathode-ray tube as a voltage to light-spot position actuator, a black white contrast edge on a diameter of the shaft end, and a photomultiplier light detector. The light spot is projected onto the shaft end and switched rapidly along the contrast edge about the shaft centre. Any detector output (at the switching frequency) is amplified, and by using a sampling function circle generator, the system is able to position the light-spot so as to minimize the detector output. Thus, this closed-loop system operates to maintain the light-spot on the contrast edge and the input to the circle generator is the tachometer output [3].

The instrument must be focused on the shaft end; no special calibration is required and the light spot readily locks onto a rotating contrast edge. An experimental instrument has been used on shafts down to 3 mm in diameter, which has a range of at least 20000rev/min in either direction, a bandwidth of about 1 kHz with a resolution of about 400rev/min limited by noise, owing to the effect of background light on the photomultiplier. Greater resolution may be obtained at the expense of bandwidth [3].

A simple schematic of the system is shown in Figure.2.2. The light spot is controlled to sit on the contrast edge along the radius away from the optical target centre of rotation. The c.r.t.(cathode-ray-tube) is driven by a circle generator so that the two in series act as a voltage to light-spot angular displacement actuator (an 'optoelectronic shaft'). Movement of the real shaft in either direction results in a change in level of light reflected from the optical target onto the photo detector. The resultant signal is amplified to drive the circle generator and so move the light spot back onto the contrast edge. The voltage driving the circle generator gives a measure of the angular displacement (and direction) of the c.r.t. light spot and so that of the shaft. The system will operate with the light spot anywhere on the contrast edge except the optical target centre of rotation, so that special calibration is unnecessary. The c.r.t. light spot is a very low inertia moveable light source and so the system has the possibility of high bandwidth.

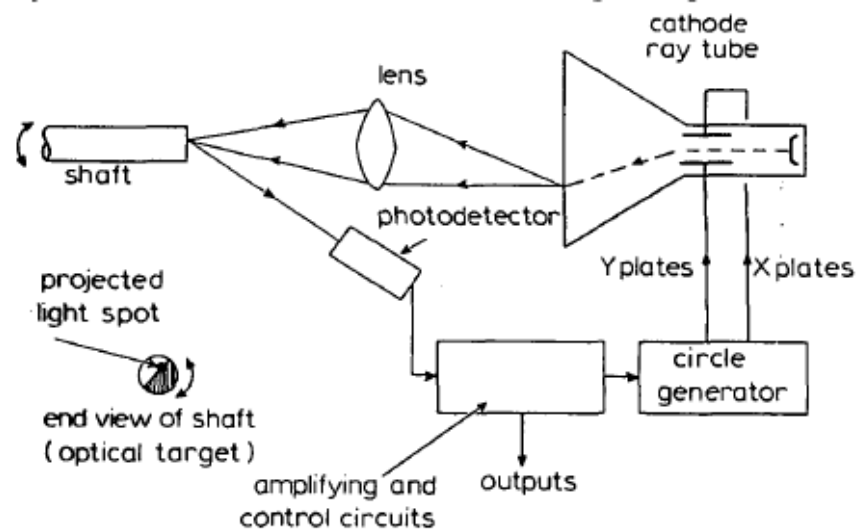


Figure 2.2: Schematic Non Contact Analogue Tachometer

2.3 SENSOR

A sensor; is a device that measures a physical quantity and converts it into a signal which can be read by an observer or by an instrument. A sensor is a device which receives and responds to a signal. A sensor's sensitivity indicates how much the sensor's output changes when the measured quantity changes. Sensors that measure very small changes must have very high sensitivities. Sensors also have an impact on what they measure. Sensors need to be designed to have a small effect on what is measured; making the sensor smaller often improves this and may introduce other advantages. A good sensor obeys the following rules:

- i. Is sensitive to the measured property
- ii. Is insensitive to any other property likely to be encountered in its application
- iii. Does not influence the measured property

Ideal sensors are designed to be linear or linear to some simple mathematical function of the measurement, typically logarithmic. The output signal of such a sensor is linearly proportional to the value or simple function of the measured property. The sensitivity is then defined as the ratio between output signal and measured property

2.3.1 INFRARED SENSOR

An infrared (IR) sensor is an electronic device that emits and/or detects infrared radiation in order to sense some aspect of its surroundings. Infrared sensors can measure the heat of an object, as well as detect motion. Infrared sensor is electromagnetic radiation with a wavelength between 0.7 and 300 micrometres, which equates to a frequency range between approximately 1 and 430 THz. IR wavelengths are longer than that of visible light, but shorter than that of terahertz radiation microwaves. Bright sunlight provides an irradiance of just over 1 kilowatt per square meter at sea level. Of this energy, 527 watts is infrared radiation, 445 watts is visible light, and 32 watts is ultraviolet radiation. But infrared sensors are usually designed only to collect radiation within a specific bandwidth. As a result, the infrared band is often subdivided into smaller sections.



Figure 2.3: Infrared sensor (Left: Receiver. Right: Transmitter)

2.4 MICROCONTROLLER

Microcontrollers must contain at least two primary components – random access memory (RAM), and an instruction set. RAM is a type of internal logic unit that stores information temporarily. RAM contents disappear when the power is turned off. While RAM is used to hold any kind of data, some RAM is specialized, referred to as registers. The instruction set is a list of all commands and their corresponding functions. During operation, the microcontroller will step through a program (the firmware). Each valid instruction set and the matching internal hardware that differentiate one microcontroller from another [4].

Most microcontrollers also contain read-only memory (ROM), programmable read-only memory (PROM), or erasable programmable read-only memory (EPROM). All of these memories are permanent: they retain what is programmed into them even during loss of power. They are used to store the firmware that tells the microcontroller how to operate. They are also used to store permanent lookup tables. Often these memories do not reside in the microcontroller; instead, they are contained in external ICs, and the instructions are fetched as the microcontroller runs. This enables quick and low-cost updates to the firmware by replacing the ROM.

Where would a microcontroller be without some way of communicating with the outside world? This job is left to input/output (I/O) port pins. The number of I/O pins per controllers varies greatly, plus each I/O pin can be programmed as an input or output (or even switch during the running of a program). The load (current draw) that each pin can drive is usually low. If the output is expected to be a heavy load, then it is essential to use a driver chip or transistor buffer.

Most microcontrollers contain circuitry to generate the system clock. This square wave is the heartbeat of the microcontroller and all operations are synchronized to it. Obviously, it controls the speed at which the microcontroller functions. All that needed to complete the clock circuit would be the crystal or RC

components. We can, therefore precisely select the operating speed critical to many applications.

To summarize, a microcontroller contains (in one chip) two or more of the following elements in order of importance [5]:

- i. Instruction set
- ii. RAM
- iii. ROM, PROM or EPROM
- iv. I/O ports
- v. Clock generator
- vi. Reset function
- vii. Watchdog timer
- viii. Serial port
- ix. Interrupts
- x. Timers
- xi. Analog-to-digital converters
- xii. Digital-to-analog converters

2.5 MICROCODE STUDIO

MicroCode Studio is a visual Integrated Development Environment (IDE) with In Circuit Debugging (ICD) capability designed specifically for microEngineering Labs PICBASIC™ and PICBASIC PRO™ compiler. The main editor provides full syntax highlighting of the code with context sensitive keyword help and syntax hints. The code explorer allows us to automatically jump to include files, defines, constants, variables, aliases and modifiers, symbols and labels that are contained within our source code. We just full cut, copy, paste and undo is provided, together with search and replace features [6]. In the MicroCode Studio, we can;

- i. Full syntax highlighting of the source code
- ii. Quickly jump to include files, symbols, defines, variables and labels using the code explorer window
- iii. Identify and correct compilation and assembler errors

- iv. View serial output from our microcontroller
- v. Keyword based context sensitive help
- vi. Support for MPASM

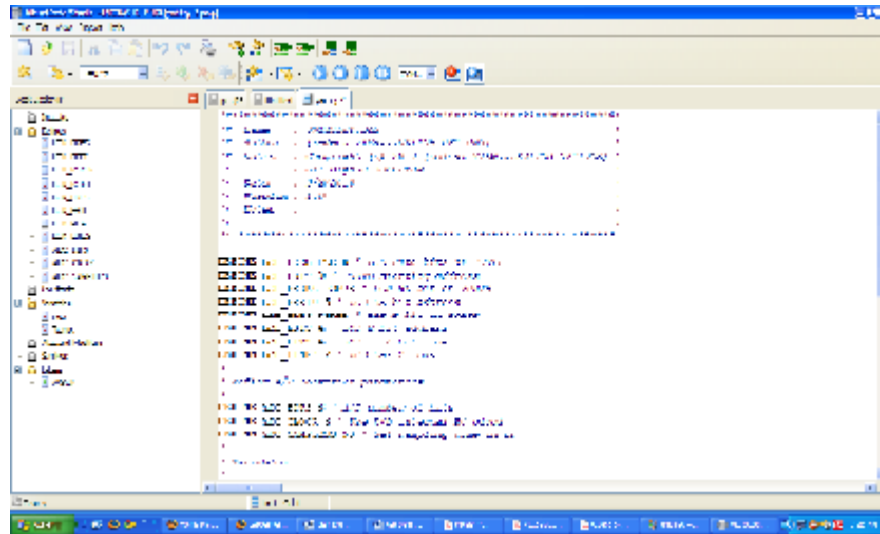


Figure 2.4: An example of MicroCode Studio

2.6 PROTEUS 7 PROFESSIONAL

Proteus PCB design combines the ISIS schematic capture and ARES PCB layout programs to provide a powerful, integrated and easy to use suite of tools for professional PCB Design.

All Proteus PCB design products include an integrated shape based auto router and a basic SPICE simulation capability as standard. More advanced routing modes are included in Proteus PCB Design Level 2 and higher whilst simulation capabilities can be enhanced by purchasing the Advanced Simulation option and/or micro-controller simulation capabilities [7]. In the Proteus 7 Professional, we can;

- i. Professional Schematic Capture module
- ii. Professional PCB Layout module
- iii. Hardware Accelerated Display Technology
- iv. Basic Simulation
- v. Max. Number of Pins In Netlist
- vi. Shape-based Power Planes

- vii. Global Shape Based Autorouting
- viii. External Autorouter Interface
- ix. Custom Scripted Autorouting
 - x. Command Driven Interactive Autorouting
- xi. 3D Board Visualisation
- xii. ODB++ Manufacturing Output
- xiii. Gate-Swap Optimizer
- xiv. Board Autoplacement

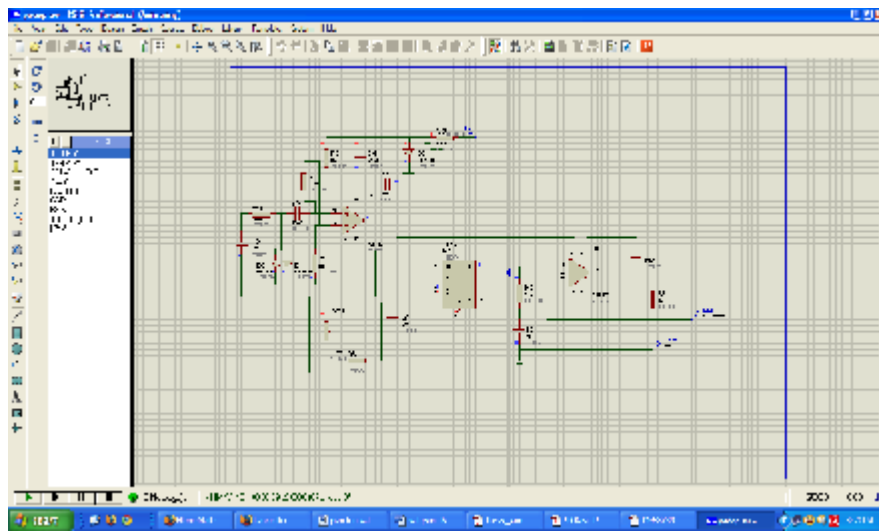


Figure 2.5: An example of Proteus 7 Professional

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

This chapter will cover the process involved in the development of the contactless tachometer study project. The processes involved are under constant changes due to unexpected changes or complications. The processes of the development of this project will be divided into several parts: block diagram of the process flow, flow chart of the project progress and the flow chart of contactless tachometer. The block diagram of the system is shown in Figure 3.1. It is a closed-loop with real time control system.

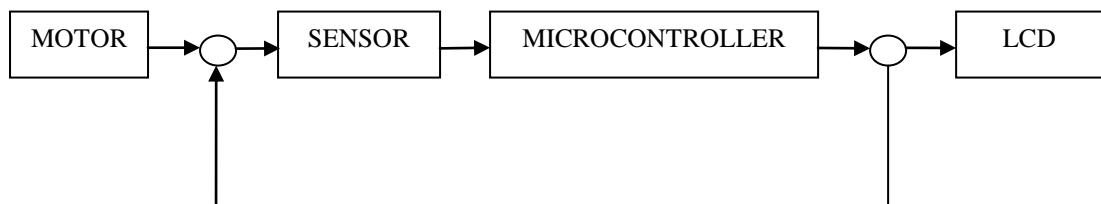


Figure 3.1: Block diagram of contactless tachometer

The actual speed of the motor will be measured by using the infrared sensor. In microcontroller, it will calculate the pulses and deduce the frequency of those pulses. Then, the microprocessor will process them and display the result on the LCD display. This process is happen again and again because there is a feedback after the microcontroller and it go back to the sensor. The process will only stop when we were not pushing the ON button again.

3.2 INSTANTANEOUS MEASUREMENT ALGORITHM

To be able to deduce an RPM reading in less than second, while constantly refining the reading's accuracy, a simple algorithm have been developed, where a counter and a timer are used. Counter and timers are part of the internal features of a micro-controller, (like the PIC 16F877 used in this project) and they can be easily configured through programming.

The figure below shows how the timer and the counter are used for this task [8-9]. The counter is connected in such a way to count pulses coming from the proximity sensor, while the timer is used to precisely feed the counted value to the microcontroller every fifth of a second, and reset the counter to 0 [10]. The microcontroller can now take an average of the last 3 readings (saved in C1, C2 and C3) and calculate the average numbers of pulses per fifth second, then multiply this value by 5, to get the number of pulses per second, then multiply this value by 60 to get the number of pulses per minute, which represents the measured RPM. The only purpose of calculating an average reading is that it will allow getting more stable reading and preventing display flickering [11].

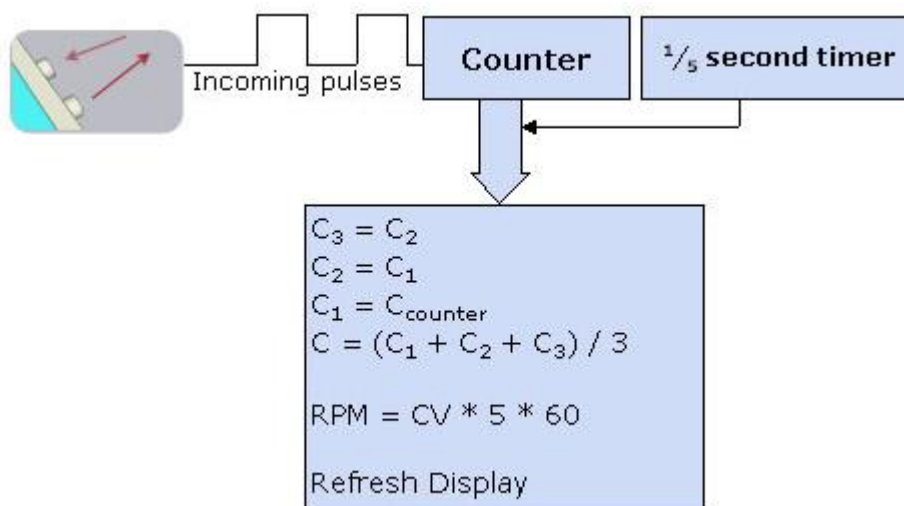


Figure 3.2: Software algorithm for instantaneous measurement. C1, C2 and C3 are used to store the last 3 reading

3.3 THE PROXIMITY SENSOR

The basic idea is to send infra red light through IR-LEDs, which is then reflected by any object in front of the sensor. Then all we have to do is to pick-up the reflected IR light. For detecting the reflected IR light, we are going to use a very original technique. We are going to use another IR-LED, to detect the IR light that was emitted from another led which is the exact same type [12]. This is an electrical property of Light Emitting Diodes (LEDs) which is the fact that a led produces a voltage difference across its leads when it is subjected to light. As if it was a photo-cell, but with much lower output current [14].

In other words, the voltage generated by the LED's can't be - in any way - used to generate electrical power from light, it can barely be detected. The correct positioning of the sender LED, the receiver LED with regard to each other and to the Op-Amp can also increase the performance of the sensor [18].

First, we need to adjust the position of the sender LED with respect to the receiver LED, in such a way they are as near as possible to each others , while preventing any IR light to be picked up by the receiver LED before it hit and object and returns back. The easiest way to do that is to put the sender(s) LED(s) from one side of the PCB, and the receiver LED from the other side, as shown in the 3D model on figure 3.3 below [13]. This 3D model shows the position of the LEDs. The plate between those LED is the PCB holding the electronic components of the sensor [10]. Notice that the receiver LED is positioned under the PCB, this way, there won't be ambient light falling directly on it, as ambient light usually comes from the top.

It is also clear that this way of positioning the LEDs prevent the emitted IR light to be detected before hitting an eventual obstacle [15]. Another important issue about components positioning, is the distance between the receiver LED and the Op-Amp which should be as small as possible. Generally speaking, the length of wires or PCB tracks before an amplifier should be reduced; otherwise, the amplifier will amplify - along with the original signal - a lot of noise picked up from the electromagnetic waves traveling the surrounding.

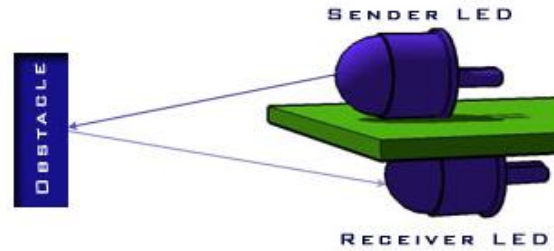


Figure 3.3 : IR LEDs positioning

3.4 PIC 16F877 MICROCONTROLLER

This microcontroller is a 40-pin device and is one of the popular microcontrollers used in complex applications. The device offers 8192 _14 flash program memory, 368 bytes of RAM, 256 bytes of non-volatile EEPROM memory, 33 I/O pins, 8 multiplexed A/D converters with 10-bits resolution, PWM generator, 3 timers, analogue capture and comparator circuit, USART, and internal and external interrupt facilities. The microcontroller acts like the brain of the DC motor speed control system. The microcontroller chip that has been selected for the purpose of controlling the speed of DC motor is PIC16F877A manufactured by Microchip. This chip is selected based on several reasons [16]:

- i. Its size is small and equipped with sufficient output ports without having to use a decoder or multiplexer.
- ii. Its portability and low current consumption.
- iii. It has PWM inside the chip itself which allow us to vary the duty cycle of DC motor drive.
- iv. It is a very simple but powerful microcontroller. Users would only need to learn 35 single word instructions in order to program the chip.
- v. It can be programmed and reprogrammed easily (up to 10,000,000 cycles) using the universal programmer in robotics lab.

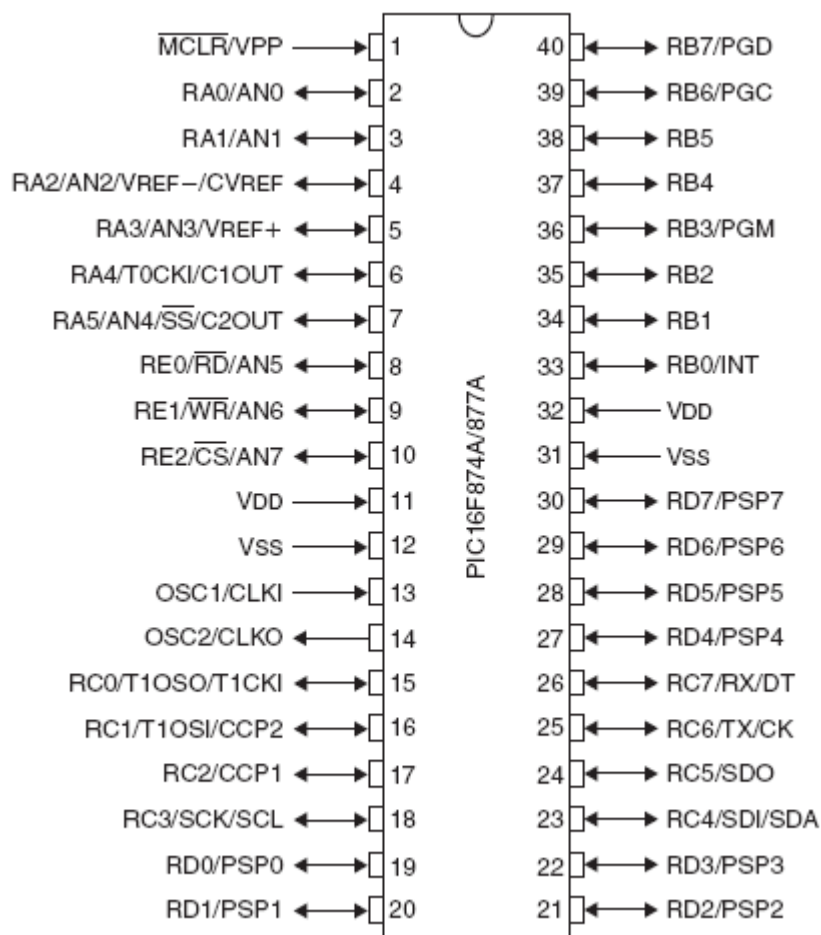


Figure 3.4: PIC16F877 microcontroller pin configuration

Microcontroller	Program Memory	Data RAM	Max Speed (MHz)	I/O Ports	A/D Converter
16F877	8192 x 14	368	20	33	8

Figure 3.5: Data of 16F877 microcontroller

3.5 THE ELECTRONICS CIRCUIT

This device is composed of two electronic circuits; the sensor, which is a slightly modified proximity sensor, and the microcontroller board, which analyses pulses coming from the sensor, process them and display the result on the LCD display. The proximity sensor has been described in the previous statement and now I am going to tell about the microcontroller board and the coding that been used

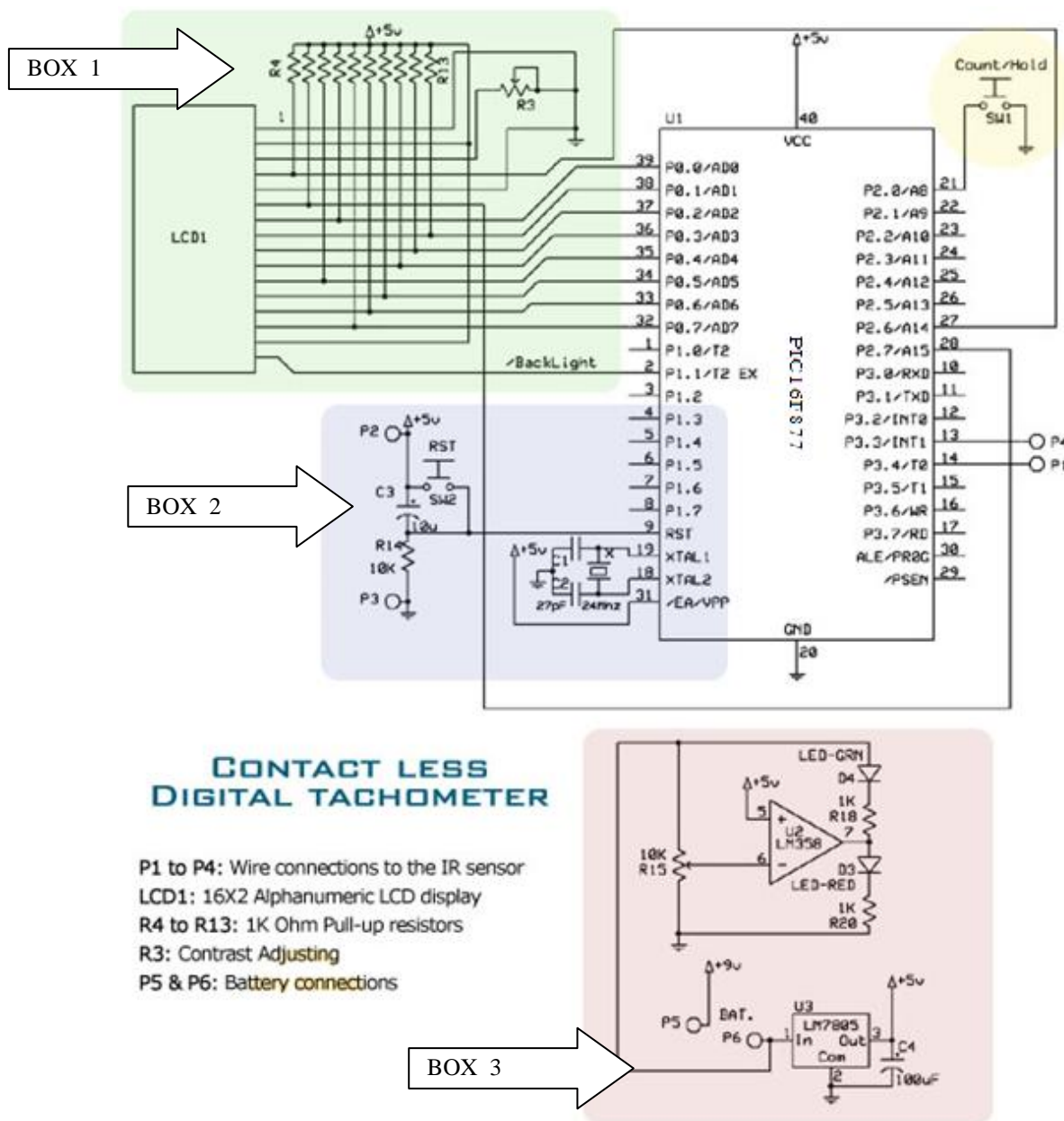


Figure 3.6: The microcontroller board

Circuit explanation:

- i. The LCD connections in the box number 1 is a standard for most of alpha numeric LCDs, the only feature I added is to be able to control the back light via the PIC 16F877 microcontroller.
- ii. The part in the box number 2 is also standard in any 16F877 microcontroller circuit, which includes the reset circuitry along with the crystal resonator that generates the clock pulses required.

- iii. The power supply, which is in box number 3 is regulates a 9V rechargeable Ni-CD battery and also provides a very simple battery monitor, with a green and a red LED, showing whether the battery need to be recharged or not.
- iv. The switch SW1, shown in the upper circle, is used to enable/disable the measurement or the counting process. When the switch is pressed, the device measures the RPM of the shaft under test, and constantly updates the reading on the LCD, when the switch is released, the last reading is held unchanged on the display, as long as the device stays on. When the switch is pressed again the old reading is replaced by the new one [5].

The wire connection P1, which is connected to the output of the sensor, is connected to the pin 3.4 of the microcontroller, this pin has a dual function which is to count incoming pulses and increment a 8, 13, or 16 bit register according to the configuration of the timer T0.

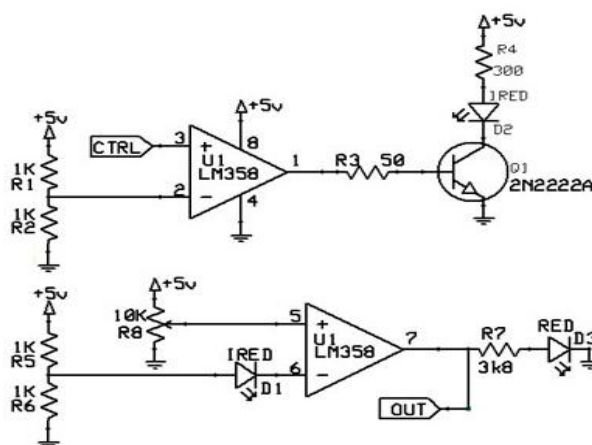


Figure 3.7: IR proximity sensor

The CTRL line in the figure 4, is an input coming from the microcontroller (*at the wire connection: P4*), turning the IR emissions ON and OFF, and the OUT line, is the output of the sensor, which is fed to the microcontroller (*at the wire connection:*

P1) [2-3]. After analyzing both the main board holding the microcontroller and the sensor, the figure 5 below show the simple diagram how they are connected together. We can refer to the figure 3 to see where P1, P2, P3 and P4 go in the main board, as well as the other lines concerning the sensor.

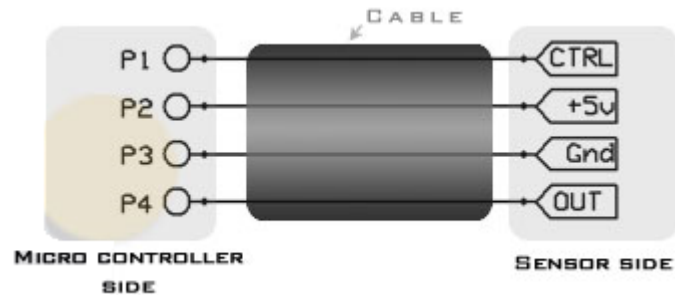


Figure 3.8: Connection of the sensor to the main board

3.6 SOFTWARE IMPLEMENTATION

For software implementation, MicroCode Studio is used to program microcontroller in assembly language. Besides, Proteus is used for user interface purpose and for monitoring the speed response of the system.

Here is the full C program, which was loaded into the microcontroller after being compiled to a HEX file. Those parts of the code were selected as the ones that emphasize the main purpose of a microcontroller in such an application [9].

```
#include <REGX51.h>
#include <math.h>

unsigned int
clk_tmp,clk_tmp2,clk_sec,clk_sec2,ex_pulses,rps,rps_tmp,temp,rps_avg,rps_ma
x;
unsigned int rps_his[5];
char a,b,c,d,e,ani;
unsigned char count1,count2,scale;
```

```

unsigned char scale = 4;
unsigned char bcd[10];
delay(y){
unsigned int i;
    for(i=0;i<y;i++){;}
}
ena(){
P2_7 = 1;
delay (5);
P2_7 = 0;
}
ini_lcd(){
P1_1 = 0;
P2_7 = 0; //transferring instructions
P2_6 = 0;
P0 = 56; //function set
ena();
P0 = 56; //function set
ena();
P0 = 12; // display on, cursor on, blink on
ena();
P0 = 1;    //clr display
ena();
P0 = 6;    //entry mode set
ena();
//enable on 2_6
//RS on 2_7
P2_6 = 1; // ready to transfere diplay data
}
locate(pos){
P2_6 = 0; //transferring instructions
P0 = pos; // change DDRAM adress
ena();
P2_6 = 1; // ready to transfere diplay data

```

```
}  
line_2(){//go to botom line  
locate(192);  
}  
line_1(){//go to botom line  
locate(128);  
}  
lcd_send(chr){  
P0 = chr;  
ena();  
}  
lcd_send_slow(chr){  
delay(1500);  
P0 = chr;  
ena();  
}  
lcd_send_num(num){  
if (num == 0){  
lcd_send (48);  
}else if(num == 1){  
lcd_send (49);  
}else if(num == 2){  
lcd_send (50);  
}else if(num == 3){  
lcd_send (51);  
}else if(num == 4){  
lcd_send (52);  
}else if(num == 5){  
lcd_send (53);  
}else if(num == 6){  
lcd_send (54);  
}else if(num == 7){  
lcd_send (55);  
}else if(num == 8){
```

```

lcd_send (56);
}else if(num == 9){
lcd_send (57);
}
}
}
setup_interrupts(){
EA = 1;
EX0 = 0;
EX1 = 0;
ET0 = 1; //set the Timer/counter 0
TR0 = 1; //Enable Timer/counter 0 to count
TMOD = 0X25; //counter 0 in mode 1 (16 bit counter) , timer 1 in mode 2 (auto
reload from TH1
TL0 = 0; //empty the counting registers
TH0      = 0; //empty the counting registers
TH1 = 0; //start counter from 0
ET1 = 1; //enable timer 1
TR1 = 1; //Enable Timer/counter 1 to count
PT0 = 1;
PT1 = 0;
}
void int_to_digits(unsigned int number){ //store the 5 digits of an integer number
in the variable a,b,c,d,e
float itd_a,itd_b;
itd_a = number / 10.0;
e = floor((modf(itd_a,&itd_b)* 10)+0.5);
itd_a = itd_b / 10.0;
d = floor((modf(itd_a,&itd_b)* 10)+0.5);
itd_a = itd_b / 10.0;
c = floor((modf(itd_a,&itd_b)* 10)+0.5);
itd_a = itd_b / 10.0;
b = floor((modf(itd_a,&itd_b)* 10)+0.5);
itd_a = itd_b / 10.0;
a = floor((modf(itd_a,&itd_b)* 10)+0.5);

```

```

}
clk() interrupt 3 //timer 1 interrupt
{
clk_tmp++;
clk_tmp2++;
if (clk_tmp2 > (1236)){ // update display
clk_tmp2 = 0;
rps_avg = floor(((rps_his[0] + rps_his[1] + rps_his[2] + rps_his[3] +
rps_his[4])/5)*60);
line_1();
int_to_digits(rps_avg);
// to lcd: AVG:
lcd_send (65);
lcd_send (86);
lcd_send (71);
lcd_send (58);
lcd_send_num(a);
lcd_send_num(b);
lcd_send_num(c);
lcd_send_num(d);
lcd_send_num(e);
// to lcd: rpm
lcd_send (32);
lcd_send (114);
lcd_send (112);
lcd_send (109);

if (P2_0 == 0){
    if (ani == 0){
        ani = 1;
        lcd_send (32);
        lcd_send (32);
        lcd_send (32);
    }else if(ani == 1){

```

```
    ani = 2;
    lcd_send (46);
    lcd_send (32);
    lcd_send (32);
    }else if(ani == 2){
    ani = 3;
    lcd_send (46);
    lcd_send (111);
    lcd_send (32);
    }else if(ani == 3){
    ani = 0;
    lcd_send (46);
    lcd_send (111);
    lcd_send (79);;
    }
}
}else{
lcd_send (32);
lcd_send (32);
lcd_send (88);
}
rps_max = rps_his[0];
if (rps_his[1] > rps_max){
rps_max = rps_his[1];
}
if(rps_his[2] > rps_max){
rps_max = rps_his[2];
}
if(rps_his[4] > rps_max){
rps_max = rps_his[3];
}
if(rps_his[4] > rps_max){
rps_max = rps_his[4];
}
int_to_digits(rps_max*60);
```

```

line_2();
// to lcd: MAX:
lcd_send (77);
lcd_send (65);
lcd_send (88);
lcd_send (58);
lcd_send_num(a);
lcd_send_num(b);
lcd_send_num(c);
lcd_send_num(d);
lcd_send_num(e);
// to lcd: rpm
lcd_send (32);
lcd_send (114);
lcd_send (112);
lcd_send (109);
lcd_send (32);
lcd_send (32);
lcd_send (32);
}
if (clk_tmp > (6584/scale)){ // update data
    clk_tmp = 0;
    if (P2_0 == 0){
        rps = TL0;
        temp = TH0;
        temp = temp * 256;
        rps = (rps + temp)* scale;
        rps_his[4] = rps_his[3];
        rps_his[3] = rps_his[2];
        rps_his[2] = rps_his[1];
        rps_his[1] = rps_his[0];
        rps_his[0] = rps;
    }
    TL0 = 0;

```



```
    TH0 = 0;
}
}
count_pulses() interrupt 1 //counter 0 interrupt
{
if (scale < 10)
scale++;
//ex_pulses++;
}
void main(){
scale = 10 ;
P3_3 = 0; // ini proximity sensor, OFF
P3_4 = 1; // ini sensor input
P1_1 = 0; //turn LCD backlight ON
P2_0 = 1; //ini count/hold button
ini_lcd(); // ini the LCD
lcd_send_slow (32);
lcd_send_slow (32);
lcd_send_slow (32);
lcd_send_slow (32);
lcd_send_slow (32);
lcd_send_slow (68);
lcd_send_slow (45);
lcd_send_slow (84);
lcd_send_slow (65);
lcd_send_slow (67);
lcd_send_slow (72);
lcd_send_slow (32);
lcd_send_slow (32);
lcd_send_slow (32);
lcd_send_slow (32);
lcd_send_slow (32);
delay(30000);
delay(30000);
```

```
delay(30000);  
delay(30000);  
line_2();  
lcd_send_slow (105);  
lcd_send_slow (107);  
lcd_send_slow (97);  
lcd_send_slow (64);  
lcd_send_slow (105);  
lcd_send_slow (107);  
lcd_send_slow (97);  
lcd_send_slow (108);  
lcd_send_slow (111);  
lcd_send_slow (103);  
lcd_send_slow (105);  
lcd_send_slow (99);  
lcd_send_slow (46);  
lcd_send_slow (99);  
lcd_send_slow (111);  
lcd_send_slow (109);  
delay(30000);  
delay(30000);  
delay(30000);  
delay(30000);  
delay(30000);  
delay(30000);  
delay(30000);  
delay(30000);  
delay(30000);  
delay(30000);  
delay(30000);  
delay(30000);  
delay(30000);  
setup_interrupts();  
while(1){  
  P3_3 = ~P2_0;  
  if (P2_0 == 1){  
    scale= 4;  
  }  
}
```

```
//e = fmod(5,10);  
}  
}
```

3.7 THE FLOW CHART

In figure below is shown the flow chart of the project progress and the flow chart of contactless tachometer. These entire figures shown below are the rules that should be taken to make this project successful implementation.

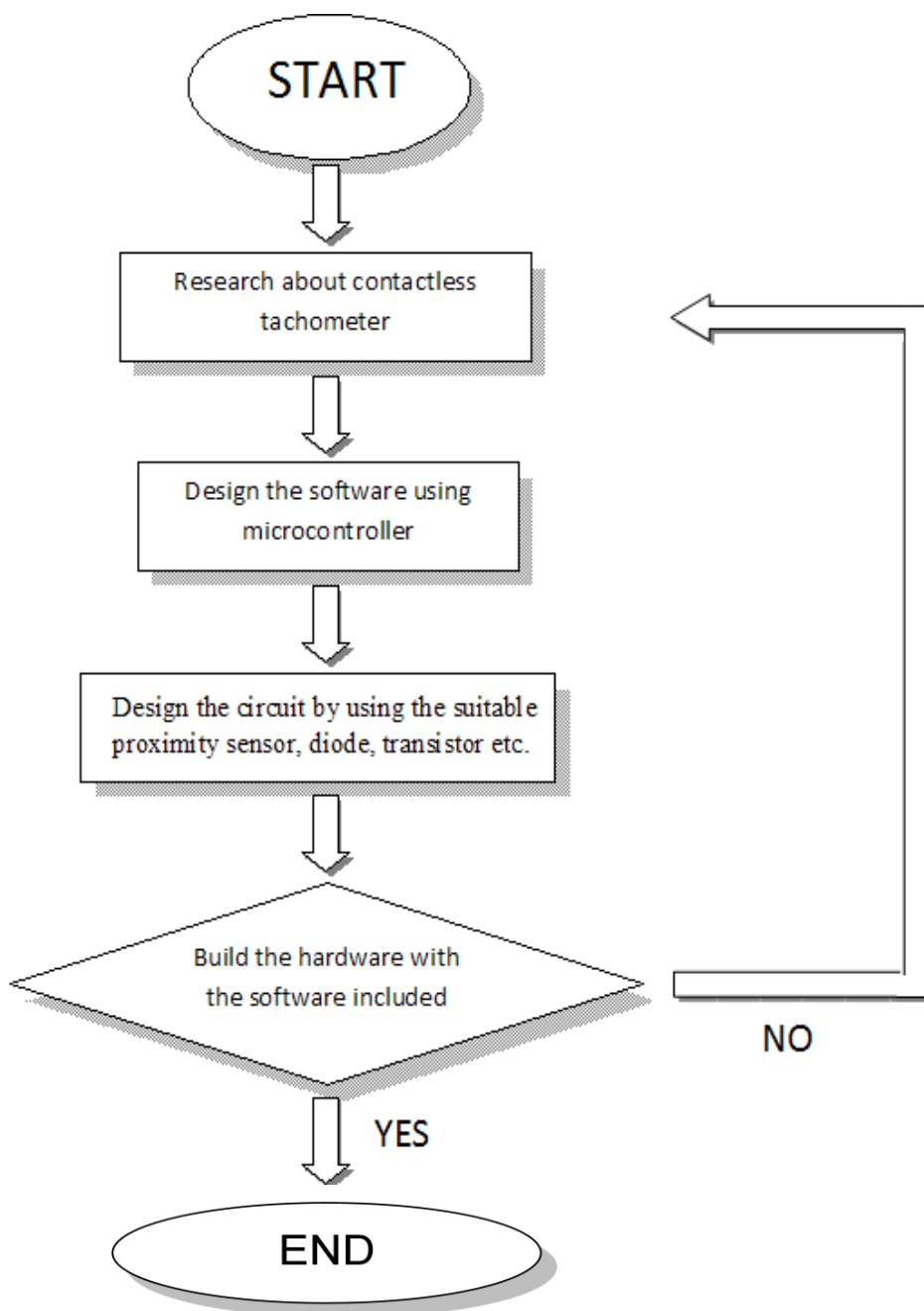


Figure 3.9: Flow chart of the project progress

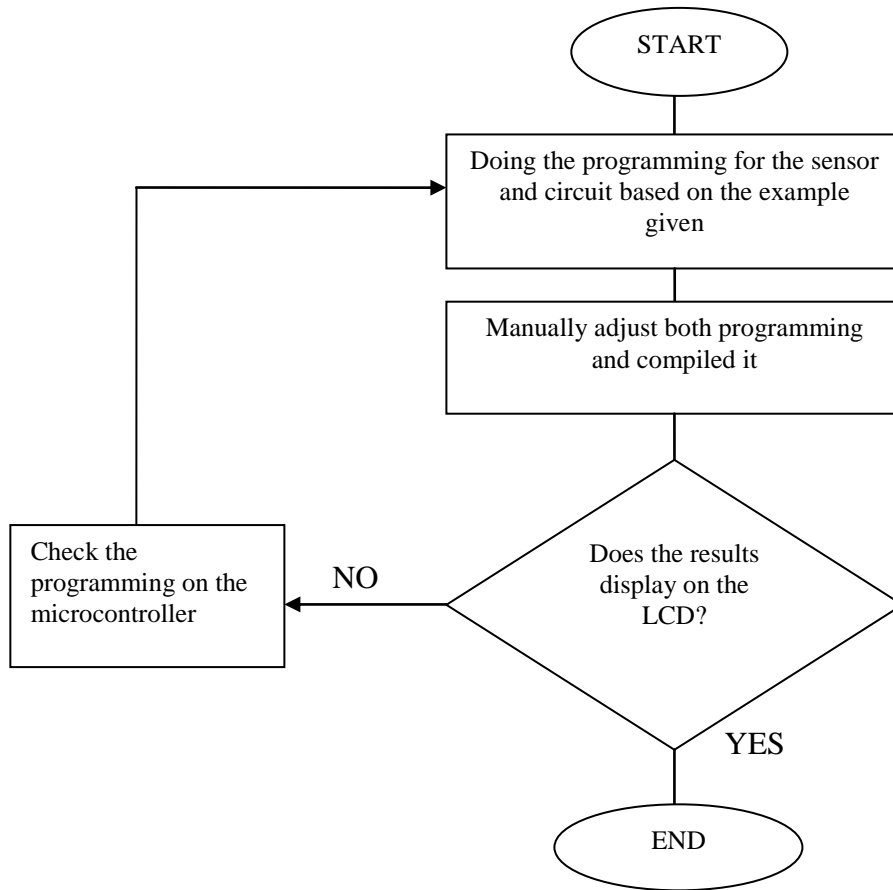


Figure 3.10: Flow chart of the contactless tachometer

CHAPTER 4

RESULTS AND DISCUSSION

4.1 INTRODUCTION

The results for this project are divided into two sections: software implementation - simulation of contactless tachometer using MicroCode Studio and contactless tachometer.

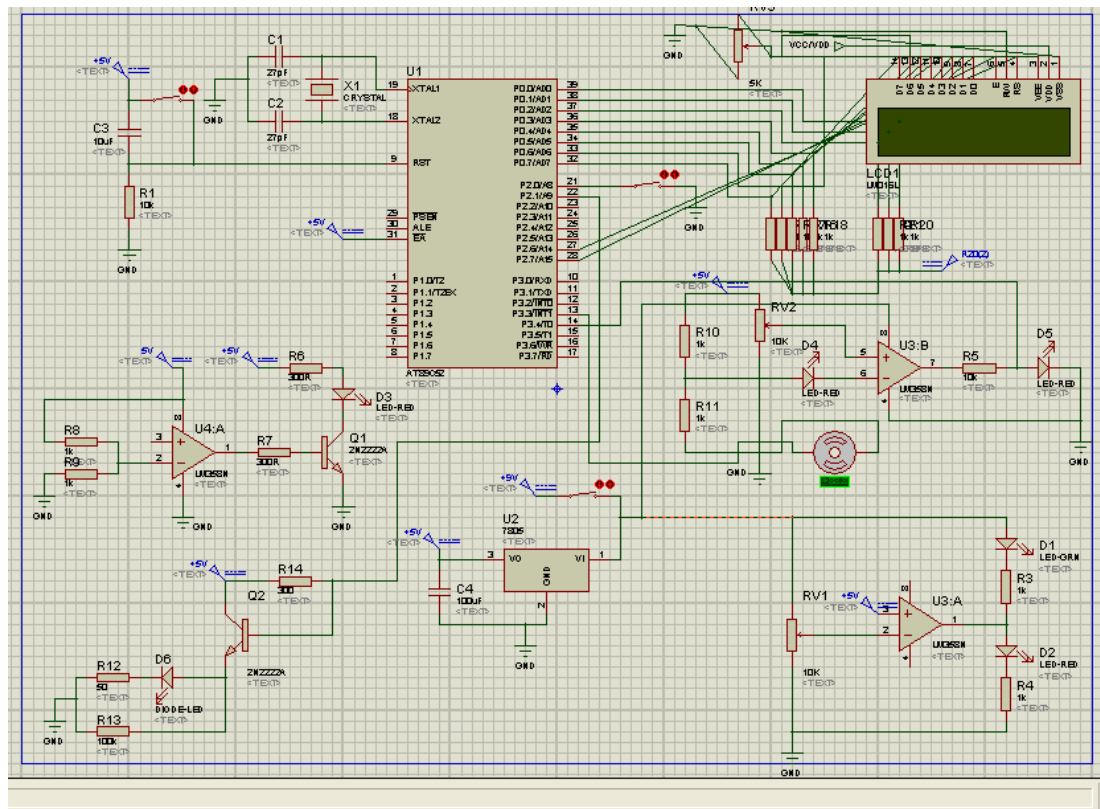


Figure 4.1: Contactless Tachometer Diagram by Using Proteus

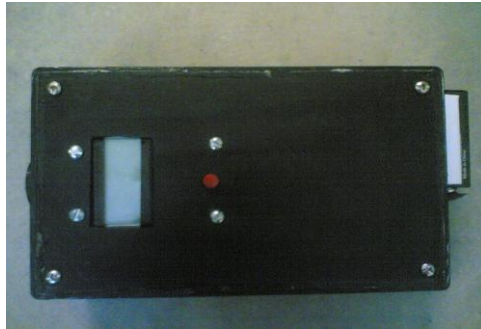


Figure 4.2: Contactless Tachometer

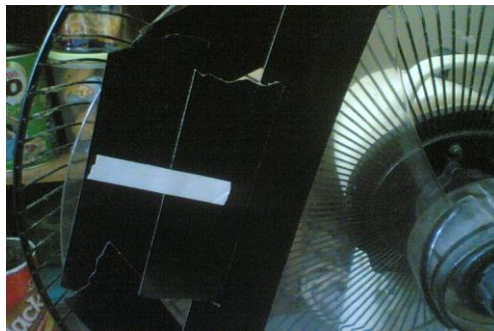


Figure 4.3: Fan with reflecting marking tape



Figure 4.4: Contactless tachometer with fan

4.2 TABLE OF RESULTS

Fan Speed	Time (s)	RPM
1	5	499.3
	10	822.3
	15	888.4
	20	915.2
2	5	761.2
	10	934.5
	15	1038
	20	1047
3	5	1062
	10	1174
	15	1179
	20	1183

Table 4.1: Speed vs Time

Fan Speed	Time (s)	RPM	Average RPM
1	20	925.2	921.4
		918.5	
		920.6	
2	20	1067	1064
		1064	
		1061	
3	20	1183	1190.3
		1193	
		1195	

Table 4.2: Average RPM for each fan speed

The calculation for average speed is as below;

$$\frac{1^{\text{st}} \text{ reading} + 2^{\text{nd}} \text{ reading} + 3^{\text{rd}} \text{ reading}}{3}$$

4.3 DISCUSSION

Those results show that the contactless tachometer is very accurate than the normal tachometer. This is because when we were using contactless tachometer, there will be no contact between the motor and the tachometer. So, the results that we got will be more precise than the normal tachometer.

CHAPTER 5

CONCLUSION

5.1 CONCLUSION

The 16F877 that I used to count the revolutions per minute of a rotating shaft is a low-power, high-performance microcontroller. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the PIC 16F877 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

As a conclusion, I can say that tachometers are widely used in industrial applications to measure the angular speed of rotating machinery. In most industrial settings, a simple quadrature encoder is not feasible to integrate into a heavy industrial application – a tachometer is a very good alternative. Typical sensors that are used are proximity, magnetic or inductive sensors. These sensors are relatively inexpensive and rugged and can be easily mounted and used as a part of a tachometer system, to get the rpm of any rotating object. The sensors output a voltage whose frequency will increase proportionally with speed. By the end of this project, it is expected that this contactless tachometer can be used at any big company or factory which use a lot of motor and also can reduced injury during time work.

5.2 FUTURE RECOMMENDATION

The performance of the system is slightly sluggish. For future works, some recommendations have been listed based on the problems in order to improve the performance.

i. Mathematical modeling of motor response

Mathematical model can be obtained from the graph of motor speed response. Then, from the mathematical model, it can be simulated using software such as Matlab to get the improved motor speed response by using controller packages such as PID controller, Fuzzy Logic Controller and others. Besides, it will reduce the total hardware complexity and cost at the same time.

ii. Hardware Improvement

Use the more precise infra red (IR) as transmitter and receiver for optical encoder for more narrow light. So, these will allow us to increase the resolution of optical encoder by increasing the number of slots. Thus, it will reduce the steady state error. Besides, time for getting data and for controller to take action also can be reduced. So, the motor speed response will become better.

iii. Software Improvement

Use fuzzy logic microcontroller which combine the idea of fuzzy logic in microcontroller to obtain a revolutions per minutes of the motor speed with excellent regulation and high robustness.

REFERENCE

- [1] R. Bonert, "Digital Tachometer with Fast Dynamic Response Implemented by Microprocessor". *ZEEE Trans, on Industrial Applications*, Vol. 19, pp.1052-1056, November-December 1983.
- [2] "A Simple Digital Tachometer with High Precision in a Wide Speed Range" by E. Galviin, A.Torralba and L.G.Franquelo, IEEE MEMBER Dpto de Ingenieria de Sistemas y Antomhtica, Univ. de Sevilla, Avda Reina Mercedes sln, 41012-Sevilla (SPAIN)
- [3] "Cathode-ray-tube actuator in a noncontact analogue tachometer" by B.E. Jones, M.Sc, Ph.D., M. Inst. M.C., C.Eng., M.I.E.E., and H.M. McNaughton, M.Sc, Ph.D.
- [4] Iovine John. *PIC Microcontroller Project Book*. 2nd Edition. Singapore: Mc Graw-Hill. 121-123; 2000.
- [5] Lawrence A. Duarte. *The Microcontroller Beginner's Handbook*. 2nd Edition. United States of America: Prompt Publication. 3-5; 1998.
- [6] <http://www.mecanique.co.uk/code-studio/>
- [7] http://www.labcenter.com/products/pcb_overview.cfm
- [8] "Extremely wide-range speed measurement using a double-buffered method," *IEEE Trans. Ind. Electron.*, vol. 41, pp. 550–559, Oct. 1994.
- [9] R. Nutt, "Digital time intervals meter," *Rev. Sci. Instrum.*, vol. 39, pp. 1342–1345, 1968.

- [10] M. Prokin, "Double buffered wide-range frequency measurement method for digital tachometers," *IEEE Trans. Instrum. Meas.*, vol. 40, pp. 606–610, June 1991.
- [11] K. Fujita and K. Sado, "Instantaneous speed detection with parameter identification for ac servo systems," *IEEE Trans. Ind. Applicat.*, vol. 28, pp. 864–872, July/Aug. 1992.
- [12] N. Chaudhuri, S. Ghosh, and A. M. Ghosh, "Wide-range precision speed measurement with adaptive optimization using a microcomputer," *IEEE Trans. Ind. Electron.*, vol. IE-30, pp. 369–373, Nov. 1983.
- [13] N.-J. Kim and D.-S. Hyan, "Very low speed control of induction machine by instantaneous speed and inertia estimation," in *Proc. IEEE IECON'94*, Bologna, Italy, Sept. 1994, pp. 605–610.
- [14] S. I. Sakai and Y. Hori, "Ultra-low speed control of servomotor using low resolution rotary encoder," in *Proc. IEEE IECON'95*, Nov. 1995, pp. 615–620.
- [15] W. Schumacher, P. Rojek, and H.-H. Letas, "Hochauflösende lageund drehzahlerfassung optischer geber für schnelle stellantriebe," *Elektronik*, vol. 10, pp. 65–68, May 17, 1996.
- [16] Lawrence A. Duarte. *The Microcontroller Beginner's Handbook*. 2nd Edition. United States of America: Prompt Publication. 3-5; 1998.
- [17] E. Galván, A. Torralba, and L. G. Franquelo, "Simple digital tachometer with high precision in a wide speed range," in *Proc. IEEE IECON'94*, Bologna, Italy, Sept. 1994, pp. 920–923.