DEVELOPMENT OF MOTION CONTROLLER SYSTEM FOR AUTONOMOUS ROBOT

By

HANIS SYAFIQAH IZZATI BINTI MOHD HANAPI

FINAL PROJECT REPORT

Submitted to the Department of Electrical & Electronics Engineering
in Partial Fulfilment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Universiti Teknologi PETRONAS
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

© Copyright 2013
by
Hanis Syafiqah Izzati Binti Mohd Hanapi, 2013

CERTIFICATION OF APPROVAL

DEVELOPMENT OF MOTION CONTROLLER SYSTEM FOR AUTONOMOUS ROBOT

by

Hanis Syafiqah Izzati Binti Mohd Hanapi

A project dissertation submitted to the
Department of Electrical & Electronics Engineering
Universiti Teknologi PETRONAS
in partial fulfilment of the requirements for the
Bachelor of Engineering (Hons)
(Electrical & Electronics Engineering)

Approved:

____

Dr. Mohd Haris Bin Md Khir
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

September 2013

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.

____

Hanis Syafiqah Izzati Binti Mohd Hanapi

ABSTRACT

In current robotics technology, autonomous robots become the most favourable robot in order to perform the dangerous, risky and continuous tasks due to its capable in interpreting and controlling the situation by itself without any human guidance. This ability is actually depends on the competency of its controller system. Recent, lots of robotics projects are using the Interface Free Controller (IFC) board as their robots' brain. IFC board helps the user by storing lots of programming data in the library. However, not all of the data stored are used by an autonomous board. Only some data is use while the others are left without any function. IFC board also will just limit the movement of autonomous robot in small and complex area.

Throughout this project, the author will have an evaluation on current common controller system, which is IFC board. Then, the author will try to create a simple, compact and smaller motion controller board by using Programmable Interface Controller (PIC) Microcontroller, to be specified the PIC18F4550 due to its speciality of 35 Input / Output pins, as the robot brain. Microcontroller will just easily interpret and generate the digital input and digital output signal. However, for real-world analog input signal from sensors, PIC18F4550 needs the help of Analog to Digital Converter (ADC) so the input signals become more compatible to be processed. For controlling the speed of motor in robotics, Pulse Width Modulation (PWM) technique is used to varying the voltage with act as signal to ON and OFF the motor, periodically. At the end of this project, the author proves that this project's product is more practical and space-limited compared to current IFC board for simple autonomous robot.

ACKNOWLEDGEMENTS

In the Name of Allah, The Most Merciful and Compassionate, praise to Allah, He is the Almighty. Eternal blessings and peace upon the Glory of the Universe, our Beloved Prophet Muhammad (S.A.W), his family and companions. First and foremost, thank to the Almighty for the strength given to carry out the Final Year Project II under the title of Development of Motion Controller System for Autonomous Robot in September 2013 Semester.

Special thanks to Electrical & Electronics Engineering Department of Universiti Teknologi PETRONAS for giving opportunity to me to do this project and Dr. Mohd Haris Bin Md Khir, my supervisor for his endless guidance, patience and idea while doing this controller system's project. Being very supportive throughout the project, he patiently guided me from the beginning, while giving me the direction whenever I get stuck in any related problems towards the completion of the project.

Huge thanks to UTP PERTROBOTS team for their constant advises and for helping me out to build the controller board and conduct the experiments. Working with them, they have shared with me not only the engineering related knowledge but also their valuable experiences, which help me to successfully complete my final year project. Last but not least, I would like to thank all my friends and colleagues for the exchanging ideas and their tremendous support and motivation during the progress and completion of this project.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1: INTRODUCTION

1.1 Background of Study

Society began to develop align with the rapid growth of civilization. Therefore, the human labours demand is continuing to increase. With the level of education that keeps on increasing and lots of researches that have been done, human are able to come with a new technology of developing an autonomous robot in order to perform the dangerous, risky and continuous tasks and to slowly decrease the rely on human workforce.

In order to make the robot are fully autonomous, the controller system needs to be priorities. The circuit need to be designed with highest efficiency of microcontroller circuit to interpret the input and disturbance thus producing accurate feedback responds. Besides, the size of the robot must be appropriate with the function of the autonomous robot to ensure that it can move freely.

Throughout this project, the author conduct an evaluation on current commonly used of controller system in industry, which is Interface Free Controller (IFC) board. This controller seems to be favourable due to its effectiveness. However, because of its bulky size, it is impractical to be installed with small robot. Therefore, the author will try to create a simple, compact and smaller motion controller board by using Programmable Interface Controller (PIC) Microcontroller, to be specified the PIC18F4550 type, as the robot brain. At the end of this project, the author hopes to stimulate and prove that this new simple style of motion controller that is more convenient compared to the IFC board.

1.2 Problem Statements

For an autonomous robot to be well-functional, it needs a good controller system that aligned with advanced programming in order to have a friendly communication and networking with the other parts of system during operation whether in structured or unstructured environment.

Common autonomous robot nowadays used IFC board in their controller system. This stack of interface cards helps to develop the microcontroller embedded system via its software library. Thus, this board smooth the designation process as the IFC board will interpret and conduct the commands by itself.

However, by referring to the current controller system, there are several problems need to be solved by the end of the project, which are;

- IFC board limits the autonomous movement as it is too bulky to be attached with the robot that assigns to conduct activities in small and compact areas.
- Not all the programming stored in the software library is used by the robot. Plus, this large of storage just will reduce the efficiency of the robots in giving the feedback responds.

1.3 Project Objectives

The goals of this research are:

- To study the development of motion controller system of an autonomous robot.
- To study the characteristics of IFC boards, especially on its disadvantages and possible ways to overcome them.
- To design a simple, compact and smaller motion controller board, compared to IFC board.

- To prove that an autonomous robot can be operated without installing IFC board, but just by simple motion controller board.

1.4 Scope of Study

1.4.1 Time Constrain

The project will be carry out in two semesters, which is approximately around 8 months. With the guide from Dr Mohd Haris and also UTP PETROBOTS Team, this project seems possible to be carried out. Plus, by the help of well organized Gantt chart (refer methodology), this project hopefully will successfully being done on the right track within the time provided.

By the end of first semester, the author aims to at least done with the literature review and the critical analysis, getting familiar with the equipments and designing the motion controller board circuits.

Then, by the second semester, the author can focus more into fabrication, programming the motion controller circuits, synchronise them according to the environment of the autonomous robot and lastly, the final prototype will be tested and some improvement will be done in order to confirming that the controller board and robot can communicate well.

1.4.2 Relevance

This project seems to become relevant aligns with current situation where IFC board is widely used in autonomous robot. As this project will help to improve the mobility of autonomous robot to move unlimitedly especially in small and complex environment, this kind of project definitely become the most welcomed by the current technology.

In IFC board, not all of the data stored in its huge library are useful for the robot. This seems to be unpractical for the simple controlling process as we need to spend lots of money for the cost, but just using few of its function. Therefore, this proposed motion controller board is also aimed to reduce the cost of controller system in an autonomous robot.

1.4.3 Feasibility

For the needed tools, equipments and common electric stuff such as microcontrollers, sensors and actuators, all of this mostly available in UTP PETROBOTS lab. Besides, with the help of familiar software used by UTP PETROBOTS like MPLAB IDE software, the controller programming process will become easier.

Meanwhile, UTP Electrical & Electronics Engineering Department Store also helps in providing electronics components for the circuit board. Besides, the Printed Circuit Board (PCB) facilities also are provided in UTP lab. With the guide of UTP Technician, the author is hopefully being able to prepare a PCB board of the circuit by the end of the project.

CHAPTER 2: LITERATURE REVIEW AND THEORY

2.1 Robot and Robotics

According to the Oxford Dictionaries, robot is defined as 'a machine that able to carry out complex series of actions automatically according to what has been programmed earlier by the computer' [1]. Practically, a robot is an electro-mechanical machine with combination of electromagnetism, electrical engineering and mechanics, added with the guidance from computer and electronics programming [2].

Meanwhile, robotics is a terms to represent the branch of technology that deals with the robots designation, construction, operation, and application [1]. In general, robots are classified in various group related to their capabilities. Common standard robots classification is shown as chart below.

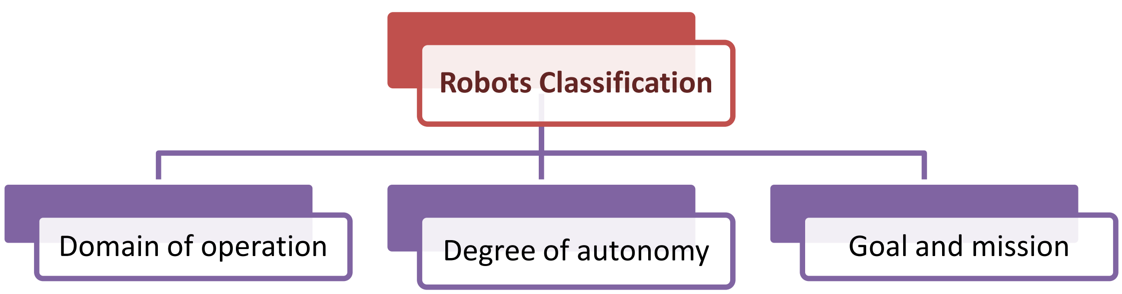


Figure 1: *Common robot classification*

i. **Domain of operation**

Robots can be designed and built for any environment imaginable. Plus, autonomous robots are usually installed with the exteroceptive system that helps in sensing things about the environment via its exteroceptive sensors [2]. Some of common environment classifications are stationary, ground, underwater, aerial or microgravity.

ii. **Degree of autonomy**

Robots also can be classified according to its autonomy, which is the quality of being self-controlled or how much human control needed in their operations. To date, robots are classified into autonomous, semi-autonomous and remotely controlled [2]. However, the industry is currently very welcoming the advent of autonomous robots as they are able to control by itself without any guidance from human.

iii. **Goal and mission**

The goal of designing the robot is also important to classify the robots. Some robots are designed for contests, personal enrichment, industrial and manufacturing or just for an entertainment purposes [2].

2.2 Autonomous Robot

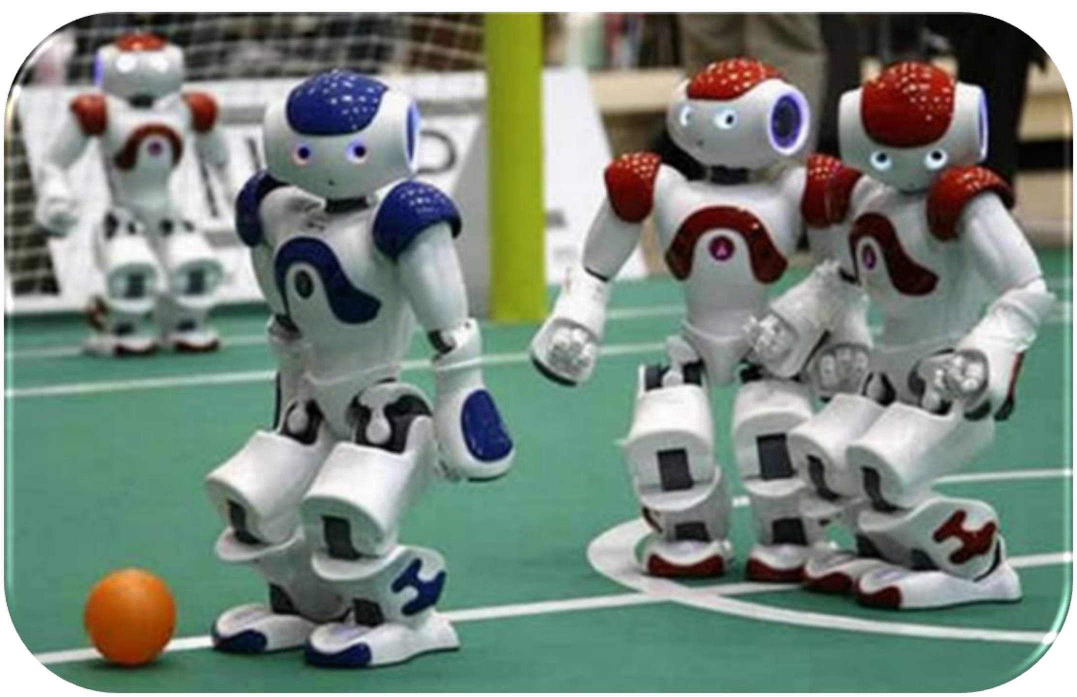


Figure 2: *Autonomous Soccer Robots*

Compared to semi-autonomous and remotely controlled robots, autonomous robots differ by having the ability to perform desired tasks in unstructured environments without continuous human guidance and control. This means that the autonomous robots can adapt themselves with the changes of the environment along their operations [2].

Autonomous robot seems to be aligned with the human mission of creating robot as it is independent from human control. Therefore, this is suitable to replacing human job in performing dangerous, risky and continuous task. However, the controller of the robot needs to be programmed in detail with its possible environments and disturbances.

2.3 General Parts of Autonomous Robot

To be considered as robot, a machine usually will consist of these five general working together parts of components [2].

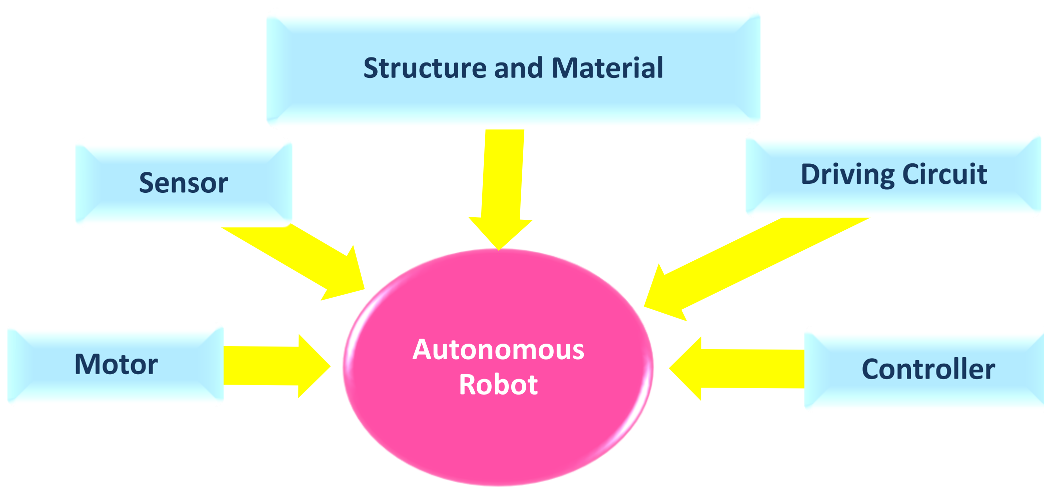


Figure 3: *General parts of Autonomous Robot*

i. Motor

Motor is a mechanism for robot movement. Several types of motor that are usually used for robot designation are DC motor, stepper motor, servo motor, brush motor and also brushless motor [2].

ii. Sensor

Sensor helps in giving the input to the robots movement by detecting the disturbance and sends the information in the form of electronics signal to the controller. The simplest form of sensor is the line sensor, where the sensor is able to detect the contrast between adjacent surfaces [2].

iii. Structure and Material

This section highlights the important of choosing materials suitable to the robot task. For example, robot mobility needs wheels to move forward and backward. Rather than the functionality, designer also needs to focus on the efficient and safety of the materials used [2].

iv. Driving circuit

Driving circuit regulates the current flowing through a circuit or used to control other components and devices in circuit. For example, H-bridge circuit able to control the standard electric DC motor of the robot with the help of transistors arrangement [2].

v. Controller

Every robot is connected to a computer, known as the controller that acts as the robot brain. It interprets the input of the disturbance and also will come with the solution to overcome it. Besides, the controller also allows the robot to be networked among the systems [3].

2.4 Controller System

Acting as the robot brain, the controller controls all the robot movement according to the stored data and executes the program for operation. The control system contains programs, data algorithms, logic analysis and also several of other processing activities to enable the robot to perform well.

The autonomous robots, which are also called as intelligent robots, are programmed to understand their environment so they can react independently. Moreover, some of current robots are able to learn from the past encounters. This means that they can identify a situation, process actions which have produced successful and unsuccessful results and also modify their behaviour to optimize the success [4].

The level of tasks assigned affected the processors used in the autonomous robot. For example, autonomous robot with high-level tasks, such as decision making, needs a standard PC or laptop mounted together with the robot to be operated. In contrast, the low-level tasks of robots just required some of simple, compact microcontrollers [5].

2.5 Interface Free Controller (IFC)

One of the most popular controllers used in an autonomous robot is Interface Free Controller (IFC). Interface means the point of interaction between the software, hardware and peripheral devices.

IFC board consists of up until 64 interfacing cards that being stack together that helps the process of determining the hardware interface and configuring peripheral in software. With the functions based on software library, IFC saves the time for software development as its only focus on algorithm development [6].

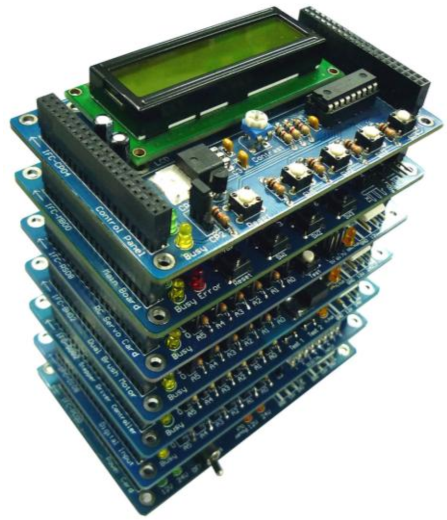


Figure 4: *IFC board [6]*

To complete the embedded system, IFC needs to fill in with several cards that being stacked together. The minimum card requires are Power Card and Main Board [6]. Power Card is used to supply DC power to the IFC boards, usually around 12V of battery. Meanwhile, Main Board contains the Peripheral Interface Controller (PIC) Microcontroller that will handle the interpreting and analysing of robot action and movement.

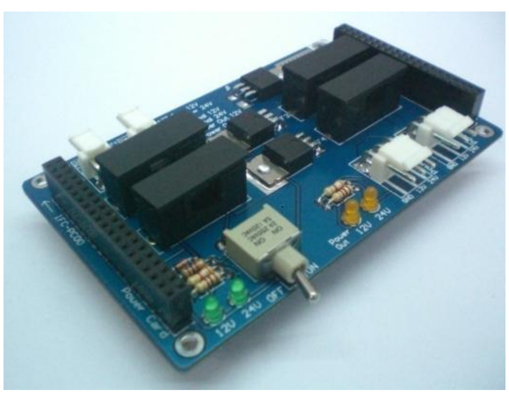


Figure 5: *Power Card of IFC [6]*

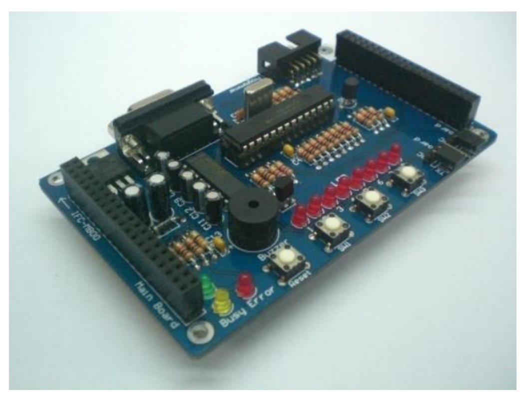


Figure 6: *Main board of IFC [6]*

Below are the characteristics of these two cards *[6]*:

Power Card	Main Board
User can use single or double 12V battery input depends on the application.	Act as the IFC main brain
Distribute power to all of IFC cards through side stack connector and external connector	PIC18F2685 MCU is used to provide the high processing speed (40MHz) and large programming memory (96K byte of program memory, 1K byte of EEPROM and 3K byte of RAM)
Provides protection to against wrong polarity and over current	Offer fast programming loading speed during the development by using external USB ICSP PIC Programmer

Table 1: *Power Card and Main Board characteristics [6]*

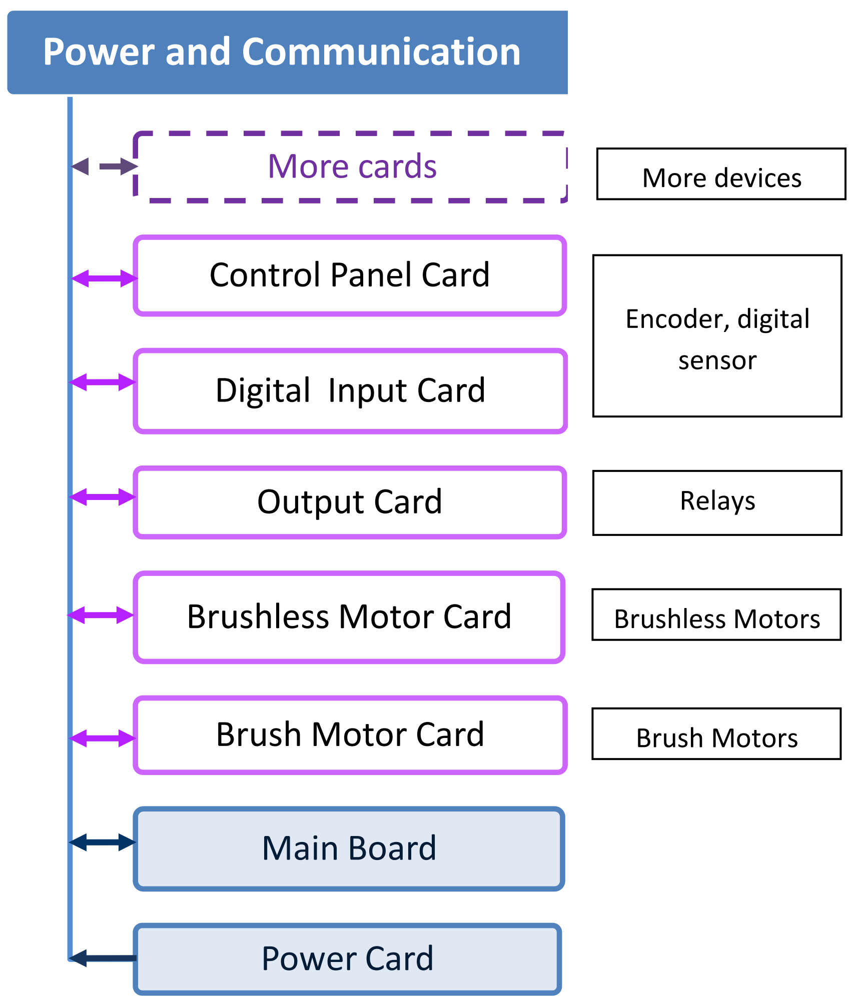


Figure 7: *Stacking card of IFC [6]*

It is true that IFC board smoothen the process of controlling and interpreting the input of the autonomous robot. However, IFC also limits the robot movements as it is too bulky for the robot, especially the one which designed to carry out task in small and complex environment. Plus, not all of the data stored in the library are useful for the robot. These then caused into inefficient way of controlling system as the huge library stored will just increased the noise between the connections. The figure below summarise the pro and cons of the usage of IFC board.

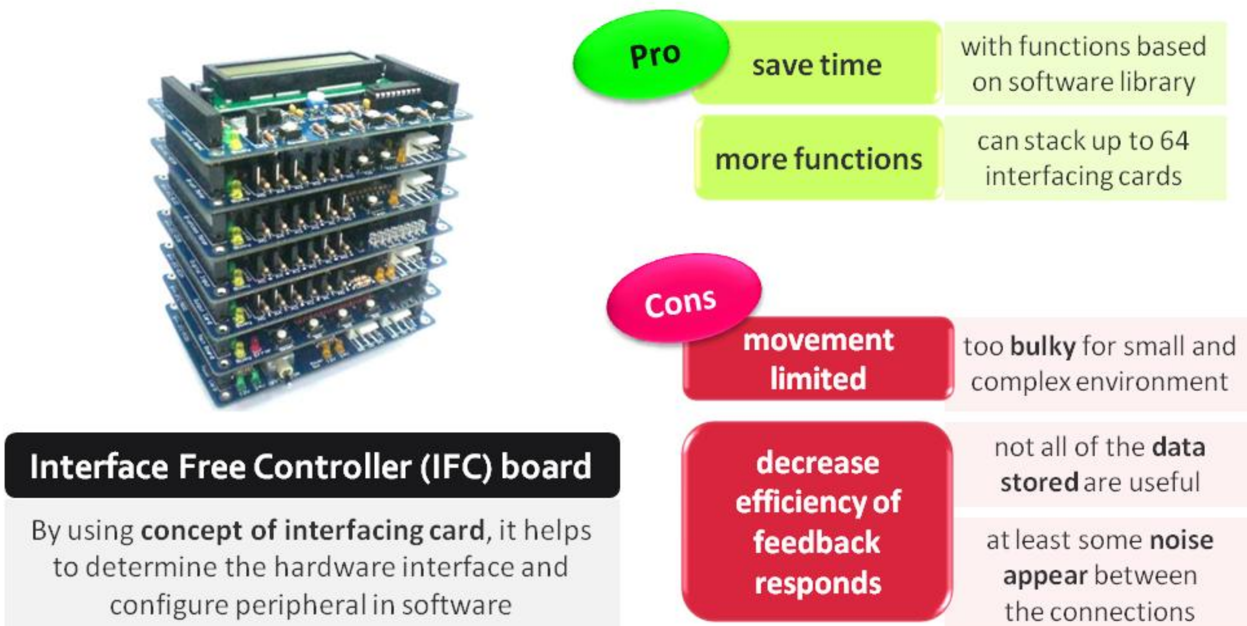


Figure 8: *Pro and Cons analysis of IFC board*

2.6 Microcontroller

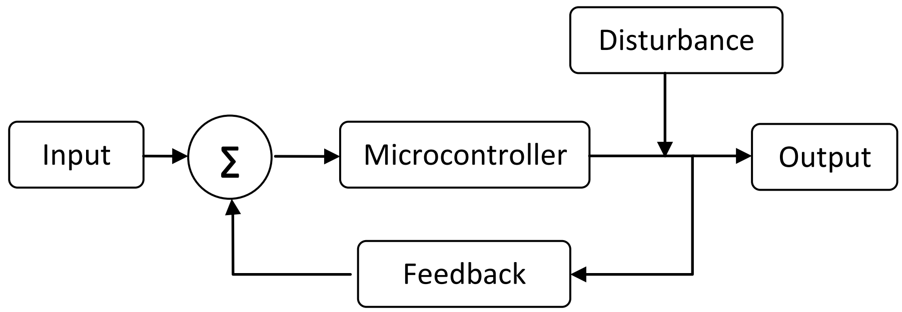


Figure 9: *Basic block diagram of the autonomous robots interpreting process*

As highlighted earlier, the Main Board Card plays the important role in interpreting the input and disturbance. The main component of the circuit is its microcontroller, which usually a Programmable Interface Controller (PIC) Microcontroller [6].

This microcontroller is well-known as the brain or control centre of robot that responsible for all computations, decision making and communications [7]. Microcontroller also allows the interfacing between sensors and required control electronics of the projects while also storing the overall logic of the robot. This logic can be programmed in many languages by beginner to the advanced programmer.

To interact with the outside world, a microcontroller possesses a series of pins that can be turned into two states through programming instruction, which are;

i. HIGH (1/ON)
ii. LOW (0/OFF)

Besides, the electrical signal from the input, such as sensors, can be interpreted via the microcontroller's pins. Nowadays, most of modern microcontrollers are already able to measure the analogue voltage signal by using Analogue to Digital Converter (ADC).

Also, with the help of ADC, a microcontroller can assign a numerical value to an analogue voltage; either it is HIGH or LOW [7].

However, in powering microcontroller up, there are some rules that need to be followed. One of the important things to be highlighted is that the output of microcontroller is only in a very small amount of electrical power, approximated around 5 Volts or below. Thus, a generic microcontroller will likely not be able to power up the electrical motors, solenoids or any other large load directly, as the power required is around 9 Volts to 12 Volts, or else the controller will be break down [7]. Therefore, some driving circuits need to be connected to the microcontroller so that the robot is able to react with the feedback actions.

CHAPTER 3: METHODOLOGY AND PROJECT WORK

In this chapter, the author will discuss the detail steps, methodology and project work that already planned to be implemented in order to complete the project.

3.1 Project Work

From the literature review process, the author therefore concludes that rather than using stack of interfacing card, it is better to rearrange the connections and fix them in just a board. With this design, the size of the controller will be decreases. At the same time, this board will be more practical as all of the connecting devices are being fully utilized.

First thing first, it is a must to know where the microcontroller is connected from and into, which means what is the type of sensors and actuators. This is important in order to design the connection between them. Besides, the functions of the circuit will later being programmed into the microcontroller. Figure 10 describes the main idea of the project circuit.

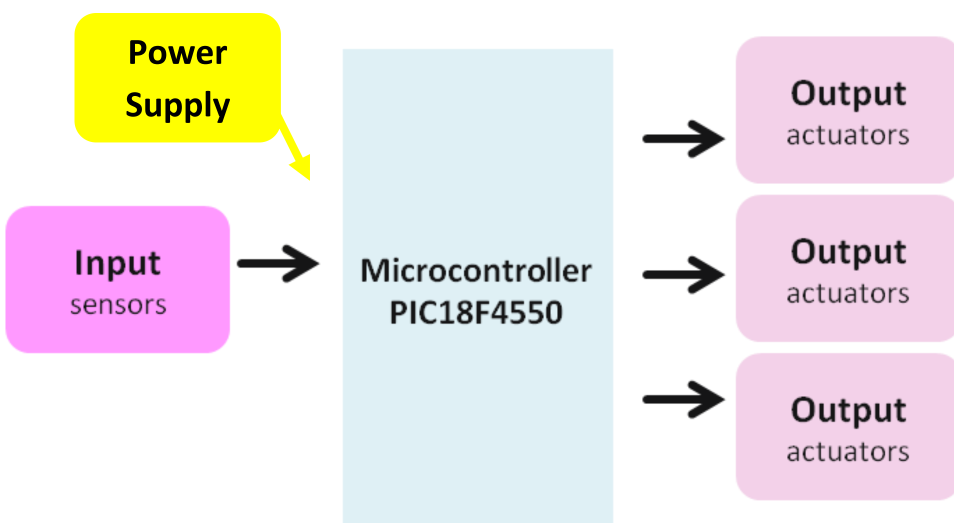


Figure 10: *Proposed controller board*

3.2 Study Plan

Step 1

Literature Review	
Collecting information	Collecting information and data on autonomous robot and its motion controller system. This scope involved knowledge on IFC boards, PIC microcontrollers
Critical analysis	Based on information and data obtained, the author tries to link them together, make some analysis and proposes a better solution to encounter the project's problem, which also helpful in the development of autonomous motion controller system.
In this step, the author is able to relate the project with current trends and demand, based on reliable resources.	

Step 2

Project Designation	
Circuits designing via software program	By using Altium Designer software, the circuits are designed according to the data and analysis gained in literature review.
Circuits stimulation	The circuits then will be run and all the errors will be corrected.
This step later assists the author in the real circuits' fabrication process.	

Step 3

Project Fabrication	
Circuits fabrication	By referring to the circuits designed in Altium Designer software, the motion controller board will be implemented.
The implementation of the designing circuits via software into prototype.	

Step 4

Algorithm and microcontroller programming	
Microcontrollers programming	With help of MPLAB Integrated Development Environment (IDE) software and PICkit 3, the microcontroller is programmed so that it will act as the desired robot function.
In order for the microcontroller to act as robot's brain, it needs to be programmed first.	

Step 5

Trouble shooting and modification	
Prototype testing	The motion controller board will be tested either it can communicate with the autonomous robot or not.
Trouble shooting and modification	If there is any error, the prototype will being fixed until fully working final product is achieved.
This step is important in order to make sure that the designed prototype is in function, which at the same time achieving the project's objectives.	

Full system testing

Prototype done

(Final Product)

3.3 Tools and Equipments Used

3.3.1 Hardware

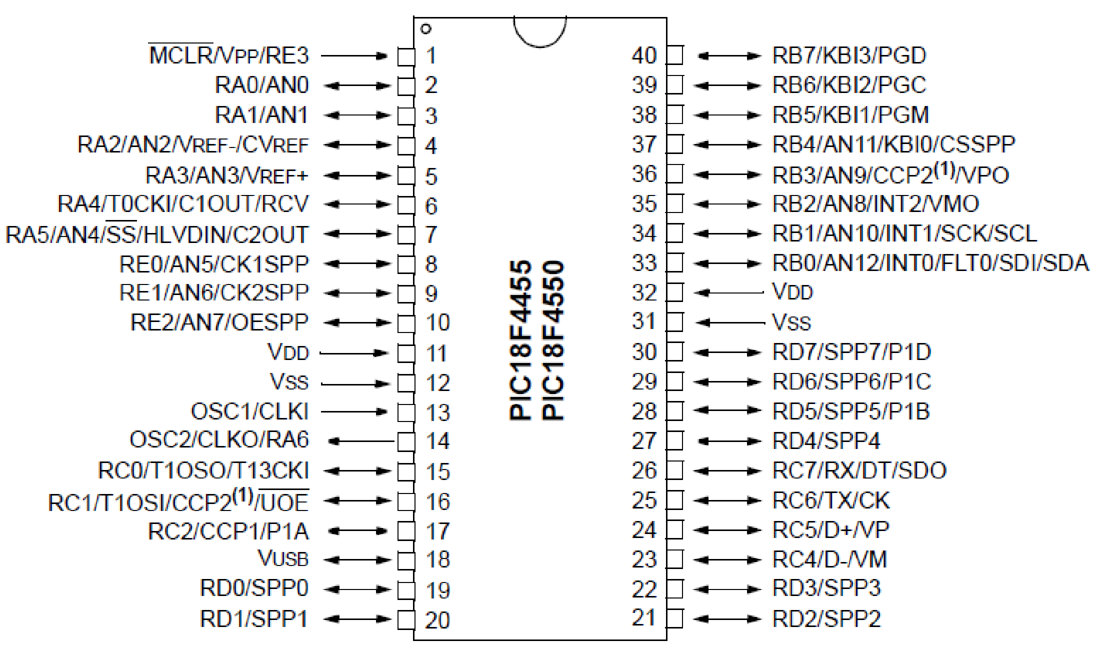


Figure 11: *40-Pin Plastic Dual In-Line Package (PDIP) [8]*

PIC18F4550 is an 8-bit microcontroller from PIC18 family and has 40 pins of PIC Microcontroller, which consists of 5 Input-Output (I/O) ports; PORTA, PORTB, PORTC, PORTD and PORTE. This is also an advanced microcontroller as it is equipped with enhanced communication protocols such as EUSART, SPI and USB. [8] Figure 12 explains the basic required connection of PIC18F4550 to make it functions.

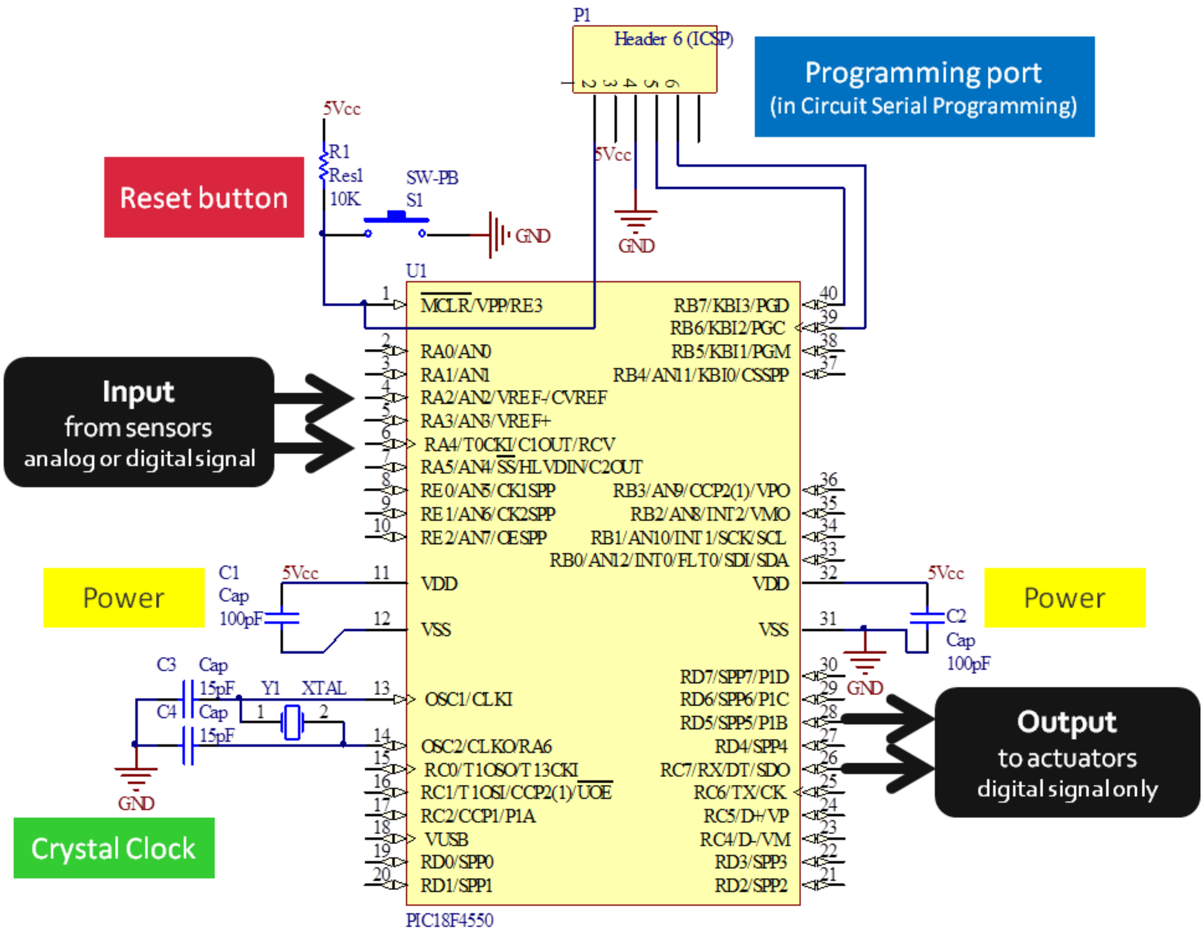


Figure 12: *Connection of working PIC18F4550*

Figure 13: *PICkit3*

PICkit3 acts as driver to allow the debugging and programming of the PIC Microcontrollers by using the powerful graphical user interface of the MPLAB Integrated Development Environment (IDE).

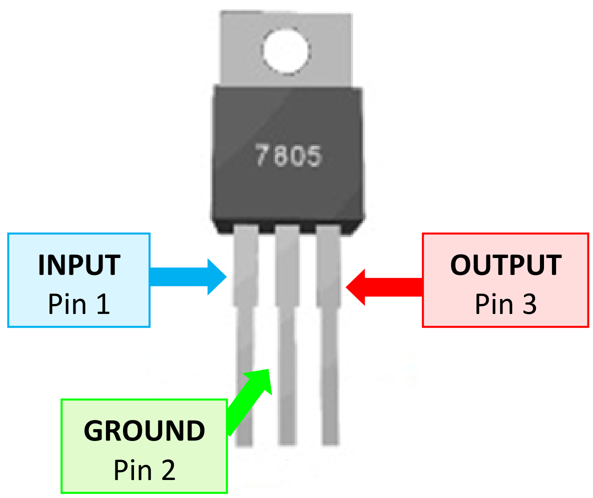


Figure 14: *LM7805 Voltage Regulator*

Voltage regulator is a device that maintains a relatively constant output voltage even though its input voltage may be highly variable. There are a variety of specific types of voltage regulators based on the particular method they used to control the voltage in a circuit. In general, a voltage regulator functions by comparing its output voltage to a fixed reference and minimizing this difference with a negative feedback loop.

The LM78XX Voltage Regulators are a popular kind for regulating and outputting positive voltage, while the LM79XX are a popular series of regulators for negative voltage. In this project, the positive voltage regulator with 5 Volt of output is used, which is the LM7805 Voltage Regulator.

Pin 1	**Pin 2**	**Pin 3**
Input Pin	Ground	Output Pin
This pin is connected to the original voltage source. For this project, it is connected to the 9 Volt of battery.	Ground connection is essential. Without ground, the circuit will not be complete as the voltage does not have any electric potential or a return path.	For LM780**5** Voltage Regulator, the 5 at the back shows that the output regulated voltage is 5 Volt.

Table 2: *LM7805 Voltage Regulator working principles*

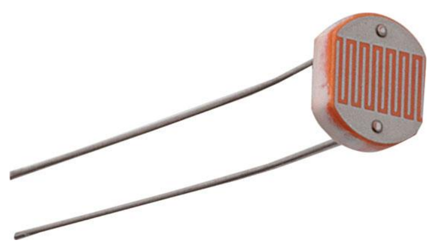


Figure 15: *Light Dependent Resistor, LDR*

Light Dependent Resistor (LDR) is very useful especially in light dark sensor circuit. Normally, the resistance of a LDR is very high, up to 1M ohms. However, due to the presence of light, this resistance will drop dramatically and then allows the current to flow pass through to produce the output.

3.3.2 Software

Figure 16: *EAGLE PCB software*

EAGLE PCB Software helps the author to sketch and design the PCB board with the schematic and layout editor. These designs are important as perfect arrangement of component will limit the size of PCB board.

Figure 17: *MPLAB Integrated Development Environment (IDE) software*

MPLAB Integrated Development Environment (IDE) exposed the author towards developing microcontrollers and digital signal controllers' applications. This software is called as IDE as it provides a single integrated environment to develop code for embedded microcontrollers.

MPLAB C Compiler for PIC18 Microcontroller also needed as a compiler that compiles the code written in the IDE into machine language that understandable by the microcontroller. After compilation, the output will be generated in .hex format. This .hex file is later used by PICkit3 to program the microcontroller. For future understanding, Figure 18 explains about the relationship between MPLAB IDE, MPLAB C Compiler and also the .hex file.

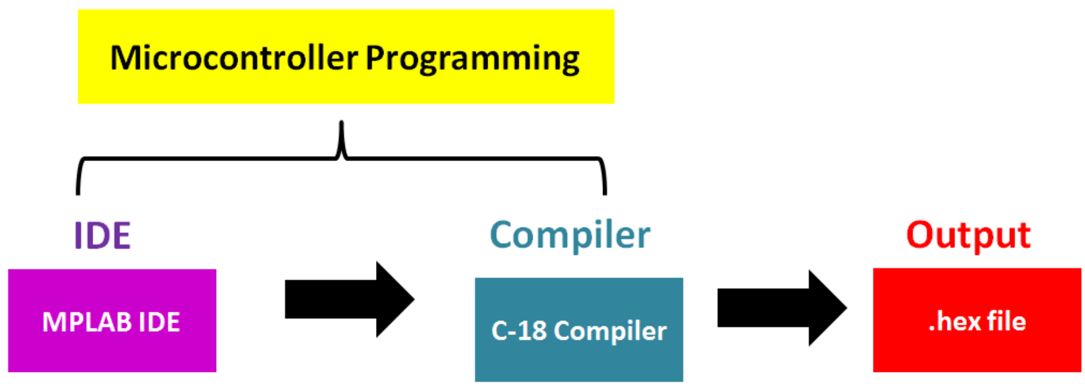


Figure 18: *Programming software needed to communicate with the microcontroller*

Note that different PIC series required different types of compiler. For PIC18F series, the most capable compiler is the C18 series. Besides, C18 also is easy language to understand to make the programming process become easier.

Another main thing to be considered in choosing the programming software is their version. For PICkit 3, the MPLAB IDE must be in version 8.20 or later with MPLAB C Compiler 18 version 3.00 or later.

User also can recheck the compatibility of these devices by using MPLAB IDE >> Configure >> Select Devices. For PIC18F4550, the recommended tools are explained in Figure 19. The green shows that the devices are compatible with PIC18F4550, while the red shows that it is not.

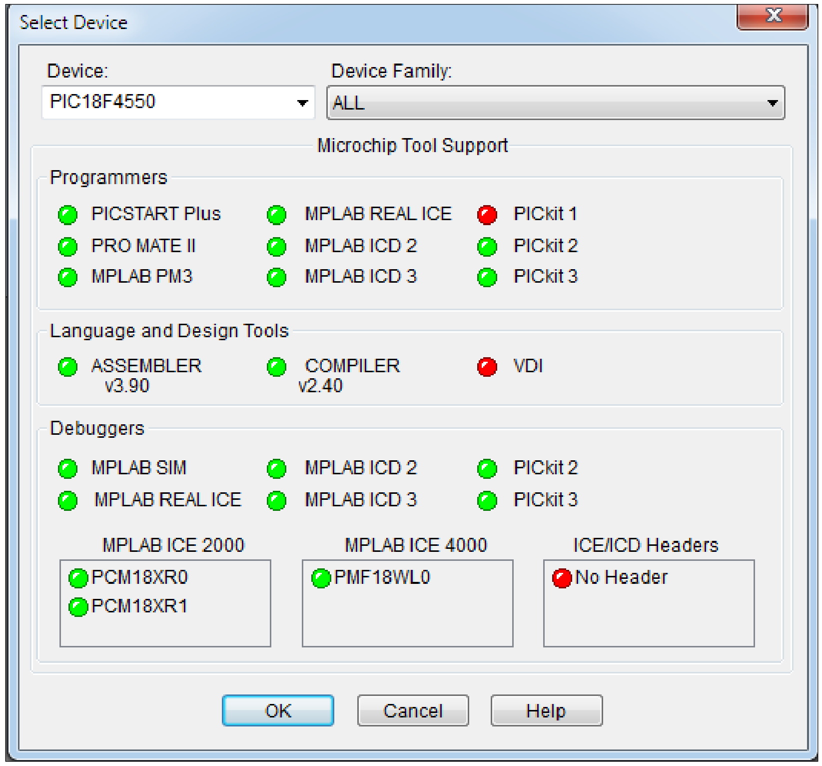


Figure 19: *Recommended tools for PIC18F4550*

3.4 Gantt Chart

Gantt chart helps the author to organize her project timeline in order to complete it within the allocated time. However, as this timeline is still in planning stage, it is more flexible and has the possibilities to be changed into more appropriate and efficient way.

No	Task	Durations	Week													
			1	2	3	4	5	6	7	8	9	10	11	12	13	14
Project selection																
1	Title released	2 weeks														
2	Title overview	2 weeks														
3	Project title selection and allocation	31st May														
Project development																
4	Project review	2 weeks														
5	Project brainstorming	4 weeks														
6	Info and data gathering	8 weeks														
7	Project drafting	8 weeks														
8	Project designation	5 weeks														
9	Fabrication of prototype	4 weeks														
10	Testing and improvement of prototype	4 weeks														
Extended Proposal																
11	Report preparation	4 weeks														
12	Draft submission	25th June														
13	Draft improvement	1 week														
14	Submission of Extended Proposal	28th June														
Proposal Defence and Progress Evaluation																
15	Improvement of Extended Proposal	3 weeks														
16	Preparing for Proposal Defence Viva	3 weeks														
17	Proposal Defence and Progress Evaluation	8th - 12th July														
Interim Report																
18	Report preparation	4 weeks														
19	Draft submission	15th Aug														
20	Draft improvement	1 week														
21	Submission of Interim Final Report	23rd Aug														

Table 3: *Gantt chart for first semester*

No	Task	Durations	Week														
			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Project development																	
1	Programmable activities	4 weeks															
2	Testing and improvement of prototype	8 weeks															
Progress Report																	
3	Report preparation	7 weeks															
4	Draft submission	8th Nov															
5	Draft improvement	1 week															
6	Submission of Progress Report	11th Nov															
Project Presentation																	
7	Preparation	7 weeks															
8	ELECTREX and Pre-EDX	4th Dec															
9	Oral presentation and viva session	30th Dec															
Final Report and Technical Paper																	
10	Reports preparation	5 weeks															
11	Drafts submission	16th Dec															
12	Drafts improvement	1 weeks															
13	Submission of Final Report (Softcopy)	23rd Dec															
14	Submission of Technical Paper	23rd Dec															
15	Submission of Final Report (Hardcopy)	6th Jan															

Table 4: *Gantt chart for second semester*

3.5 Relevancy of Objective

The designed Gantt Chart is hopefully can help the author in gaining the mission of preparing the more practical motion controller circuit board, especially in PCB board. At the same time, the project documentation also will always being update from time to time to time so both of project report and prototype are both in progress.

CHAPTER 4: RESULT AND DISCUSSION

This chapter will explain the results and discussions of the project. It is divided into some subtopics, which are power supply board, main board, programmer setup process, overall process of circuit board, Analog to Digital Converter (ADC) in making analog signal compatible with the microchip, and also Pulse Width Modulation (PWM) technique for varying the voltage for LED brightness and DC Brushless Motor shaft rotation.

4.1 Power Supply Board

For the controller board, PIC18F4550 needs 5 Volt of power supply. In this project, as common battery is 9 Volt, a voltage regulating circuit is made to tune the voltage down, from 9 Volt to 5 Volt. This is being done by connecting the 9 Volt of battery with LM7805 to produce 5 Volt of voltage output. Plus, with the help of 0.1 μF and 0.33 μF of ceramic capacitors, the input voltage is filter from any Alternating Current (AC) noise, to produce clean voltage output of DC.

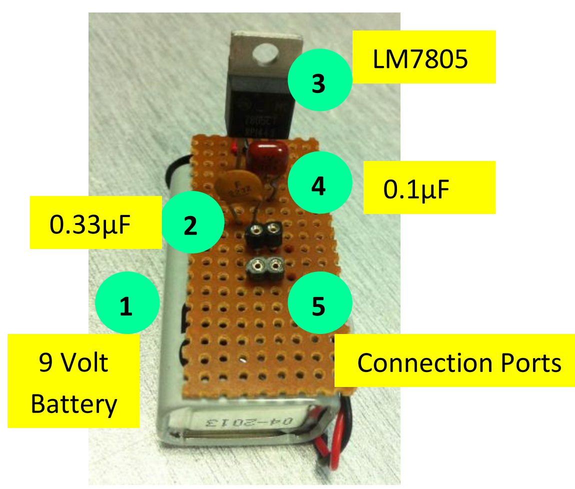


Figure 20: The connection of vol*tage regulator circuit*

The working principle of the voltage regulator circuit is explained in the table below.

1	9 Volt battery	Standard battery supply. As microcontroller needs 5 Volt for its power supply, voltage regulator is used to scale down the voltage.
2	0.33 µF ceramic capacitor	This capacitor acts to clean up the signal if any AC noise is present. It shorts this noise to ground and allows the pure DC signal into the regulator.
3	Voltage regulator	Voltage regulator regulates the 9 Volt down to 5 Volt.
4	0.1 µF ceramic capacitor	This capacitor acts to clean up any high-frequency or AC noise that may come out, again to produce a clean DC signal.
5	Connection ports	To connect the output DC voltage to the main board and others who need 5 Volt of power supply

Table 5: *Working principle of the voltage regulator circuit*

This circuit is then being measured by using multimeter to experimentally prove the concept. The reading of multimeter is recorded in the table below.

Expected Input Voltage, V_I	9 V
Measured Input Voltage, V_I	8.9887 V
Expected Output Voltage, V_o	5 V
Measured Output Voltage, V_o	5.000 V

Table 6: *Result of voltage regulator circuit*

4.2 Main Board

Main board is where the microcontroller, PIC18F4550 is located. It acts as the brain of the designed circuit. This chip will control the system to act as what have been programmed. The circuit is designed by referring to the Figure 12 shown earlier. By using EAGLE Software, the PCB board is designed. The figures of designed circuit are shown in Figure 21 to Figure 24.

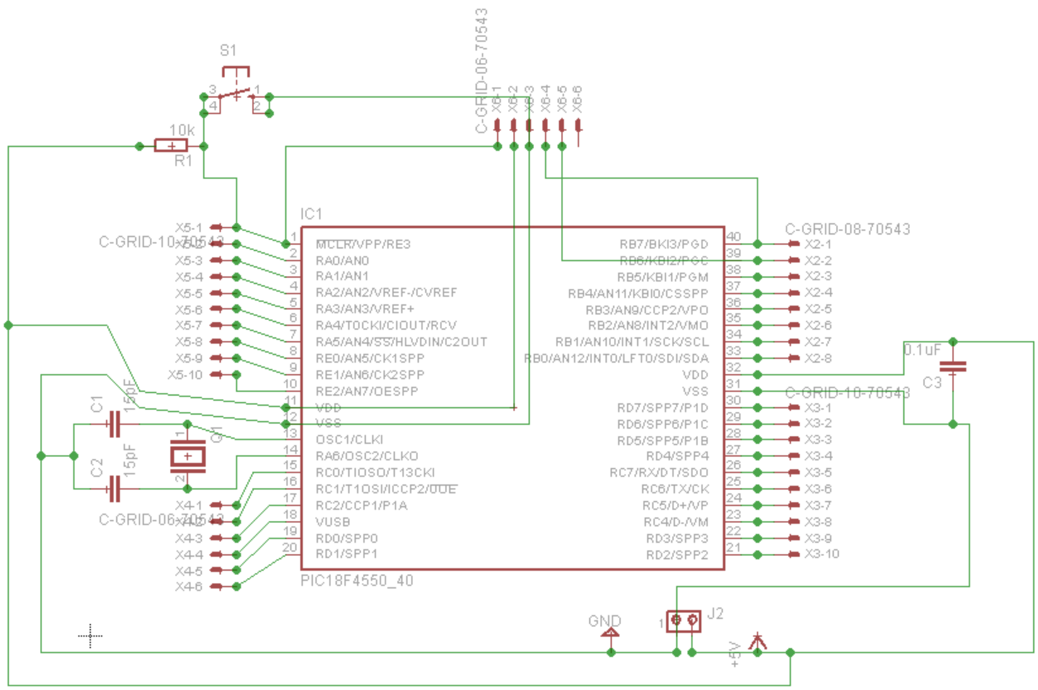


Figure 21: *Basic Circuit connection of target boards in EAGLE PCB Layout (Schematic Diagram)*

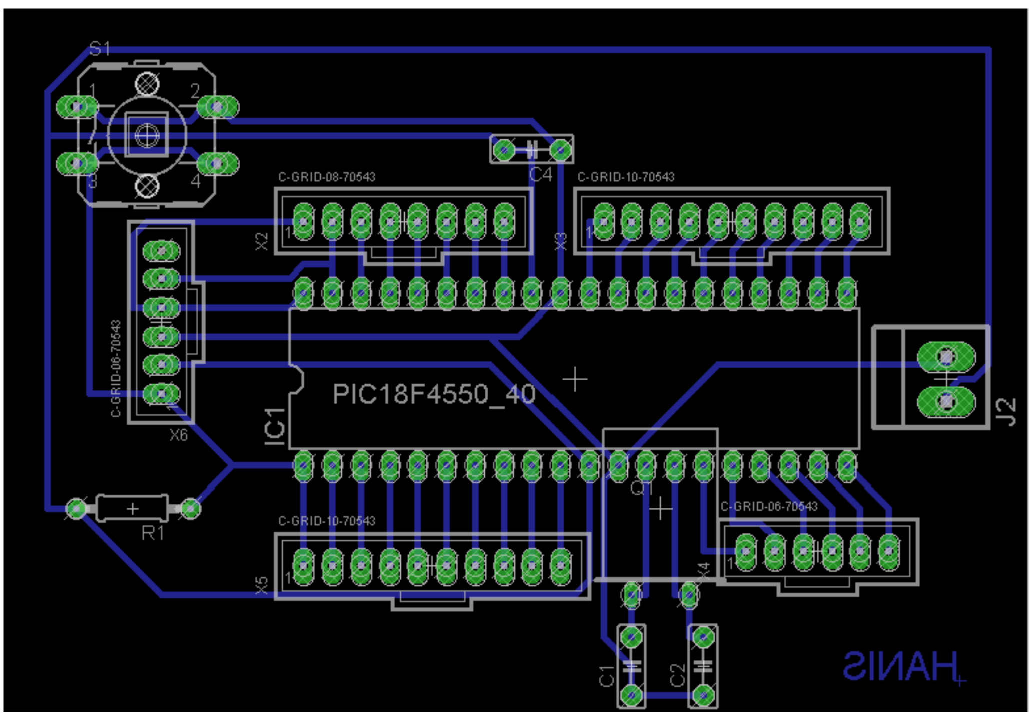


Figure 22: *Basic Circuit connection of target boards in EAGLE PCB Layout (Board Diagram)*

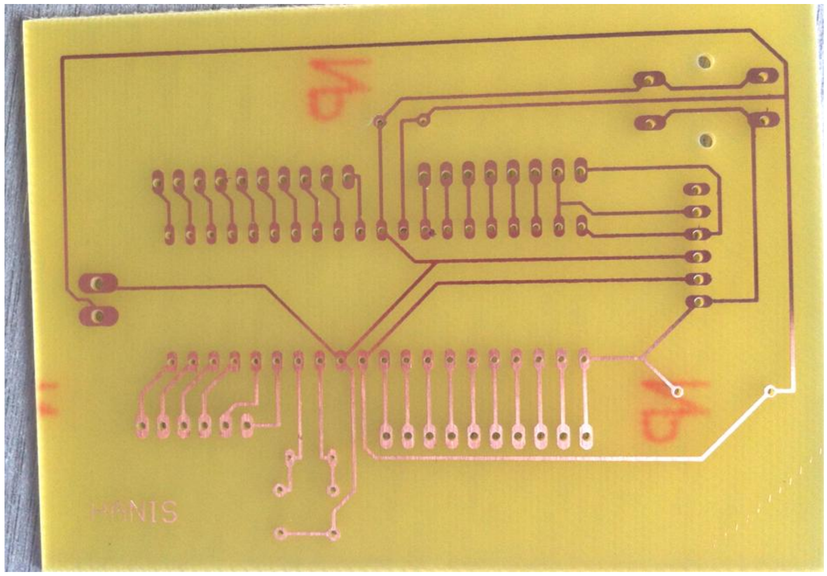


Figure 23: *PCB Board (Unsoldered)*

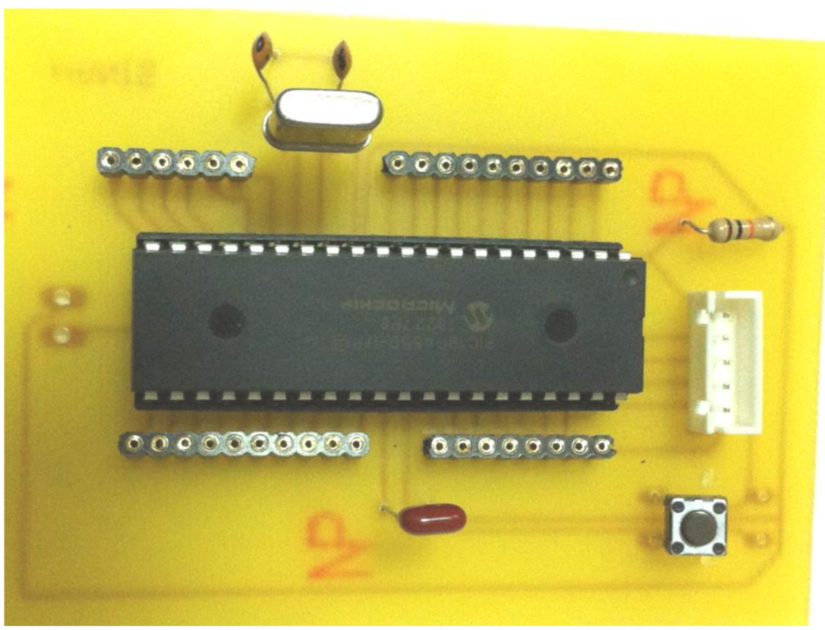


Figure 24: *PCB Board (With components)*

4.3 Programmer Setup

By referring to current technology by Microchip Inc, PICkit 3 is used to program the microcontroller due to its easiest and fastest responds with MPLAB IDE. As the writing is on the Flash Memory, the programming, erasing and reprogramming process become much easier and time practical.

Basic things needed to program the microcontroller are;

- **MPLAB IDE software** to control the programming activities.
- **PICkit3** to program the microcontroller.
- **Microcontroller circuit** or also known as **target board**.

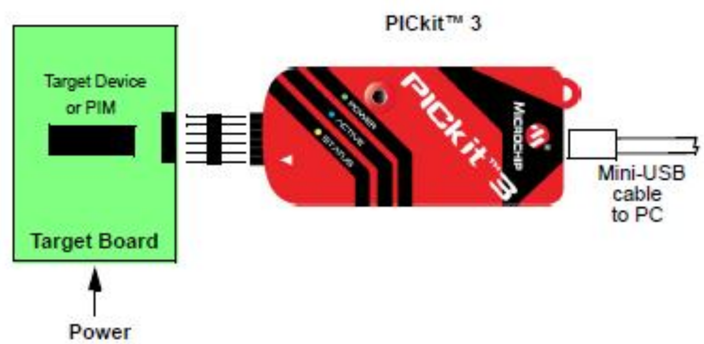


Figure 25: *Basic things to program the microcontroller*

Another important thing to be highlighted is the connection between the target board and the PICkit 3 pins. It is better to refer the PICkit 3 User Guide and PIC18F4550 datasheet first before connecting them together. Both Figure 19 and 20 show the recommended connection between the microcontroller target boards with the PICkit 3 programmer pinout.

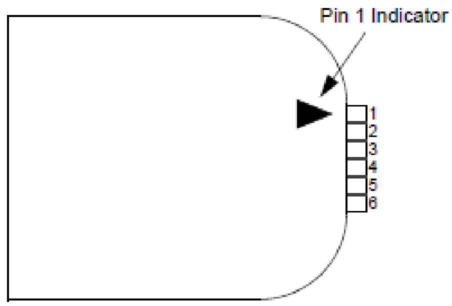


Figure 26: *Pickit3 Programmer Pinout*

Pin Description	
1	$\overline{MCLR}/v_{PP}$
2	v_{DD} Target
3	v_{SS} (Ground)
4	PGD (ICSPDAT)
5	PGC (ICSPCLK)
6	PGM (LVP)

Table 7: *Pin Description of PICkit3*

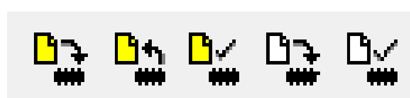


Figure 27: *MPLAB IDE Buttons to control the programming activities. From left is the program, reading, verifies, erase flash device and blank check all button.*

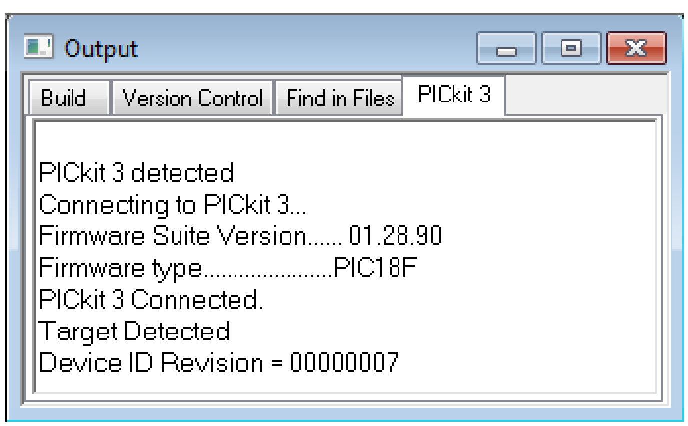


Figure 28: *Message shown that the PICkit 3 and the target board are ready for programming activities*

4.4 Circuit board

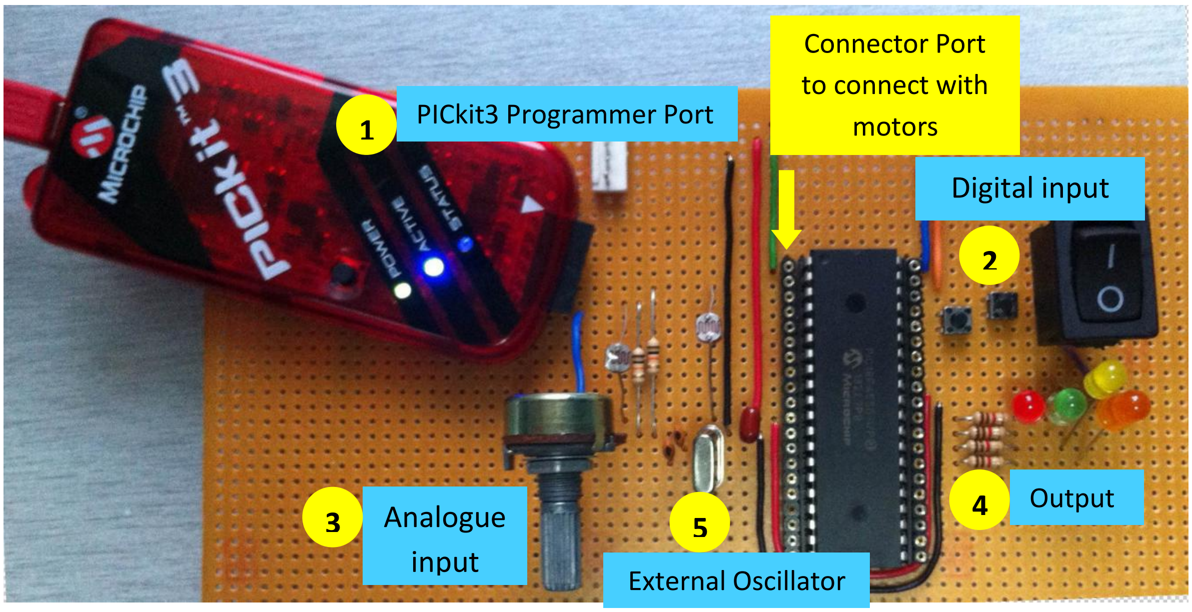


Figure 29: *The development of circuit board*

The working principle of the circuit is explained in the table below.

1	PICkit 3 able the circuit to be programmed or debugged in circuit-serial mode.
2	Pin will be defined as digital input port. Digital signal from ON OFF switch and push button will give either 0 or 1. In this project, the ON OFF switch used to control the open and close of the connection to the motor. Meanwhile, the push button is used to control the speed of shaft motor rotation.
3	Pin will be defined as analog input port. Analog signal input from LDR will be received due to the presence of light. LED (output) will start to bright up when there is light, and verse versa. For potentiometer, the resistance can be controlled by turn the head up and down. LED (output) will be bright if the resistance is low, and verse versa. Therefore, we can conclude here that the microcontroller can accept variable off analog input voltage range, that is from 2.0 V to 5.5 V. This seems reliable as PIC18F4550 is completed with an Analog to Digital Converter (ADC). This ADC helps the analog signal to be compatible with microcontroller.
4	Pin will be defined as output port. Connectors to the motor are connect here. Microcontroller will only produce digital signal of output. However, with the

	technique of Pulse Width Modulator (PWM), the output voltage can be varied so the output signal will behave almost likely to analog signal. One of the applications of using this PWM technique is controlling the speed of shaft motor rotation.
5	External Oscillator is just an optional and additional component in increasing the clock of the microcontroller. This clock is preferable when connecting the circuit board to the USB. Plus, by using this additional clock, the frequency of the PWM produced will also be increased.

Table 8: *Working principle of circuit board*

4.5 Analogue To Digital (ADC) Converter

Microcontroller can directly interpret the digital signal input as it is constantly on 0 or 1. However, for analog signal, microcontroller needs to interpret the data with the help of Analog to Digital Converter (ADC) as the signal supply is not stable. ADC allows the analog input to become compatible with the microcontroller.

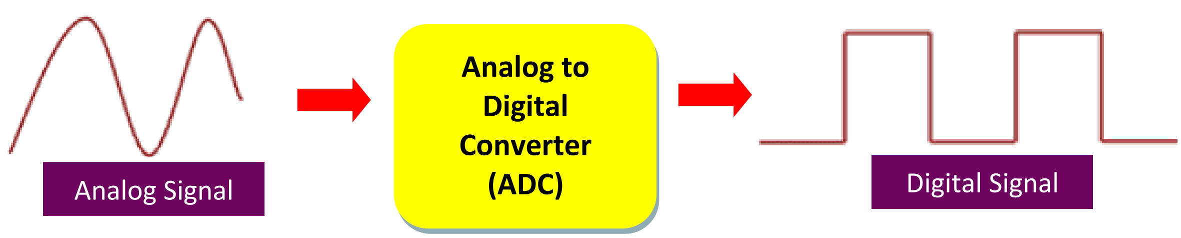


Figure 30: *The Analog to Digital Converter (ADC) concept*

The way an ADC works is fairly complex. There are a few different ways to achieve this feat, but one of the most common techniques uses the analog voltage to charge up an internal capacitor and then measure the time it takes to discharge across an internal resistor. The microcontroller monitors the number of clock cycles that pass before the capacitor is discharged. This number of cycles is the number that is returned once the ADC is complete.

One of the PIC18F4550 ADC characteristics is having a 10-bit resolution of output. In other words, an analog input will be converted into a corresponding of 10-bit digital output. This also means that PIC18F4550 has the ability to detect 1,024 discrete analog levels as;

$$\begin{aligned}\text{Discrete Analog Levels} &= 2^{\text{bit}} \\ &= 2^{10} \\ &= 1{,}024\end{aligned}$$

ADC has the ability to reports a ratiometric value of the signal. This means, the voltage is related to the ADC value.

$$\frac{\text{Resolution of the ADC}}{\text{System Voltage}} = \frac{\text{ADC Reading}}{\text{Analog Voltage Measured}}$$

Since PIC18F4550 is a 5 Volt system with 10-bit resolution, we can simplify that;

$$\frac{2^{10}-1}{5} = \frac{\text{ADC Reading}}{\text{Analog Voltage Measured}}$$

$$\frac{1023}{5} = \frac{\text{ADC Reading}}{\text{Analog Voltage Measured}}$$

Thus, this also means that ADC assumes that 5 Volt is equivalent to 1023 as $2^{10}-1 = 1023$. At the same time, any value that is less than 5 Volts will be ratio below the 1023. Some of the analogue input that being used is LDR and also potentiometer. Usually, for LDR, the voltage is considered as ON when the voltage is 2.8 Volt and above. Therefore, the ADC reading of 2.8 Volt is;

$$\text{ADC Reading} = 2.8\,\frac{1023}{5}$$

$$\text{ADC Reading} = 523$$

For PIC18F4550, it has 13 channels of ADC which means that 13 analog signal can be converted simultaneously into digital. Vref+ at point RA3 and Vref- at point RA2 pins are connected to the power supply as the external reference voltage. Shown is the sample of coding written in MPLAB IDE to program the PIC18F4550 to be operated in ADC function.

```
#include <adc.h>                    //include library ADC functions
#define ADCMAX 1023                 //max value the ADC result can have, 2^10 - 1.

int ADCresult;
float voltage;

void main (void)
{
 OSCCON = 0x70;                     //set it to use the 8MHz internal clock.
 ADCON1 = 0b01111101;               //set all ADC-capable pins to NOT be analog
                                    //inputs, just digital IO, except for AN1 (the
                                    //leftmost bit is meaningless)
 DDRB=0b11111111;                   //all inputs.
 DDRA=0b00000010;                   //all outputs except RA1, which is AN1 (the order
                                    //is RA7 - RA0)
 OpenADC (ADC_FOSC_32 &             //this is for clock speeds up to 10.65MHz
 ADC_RIGHT_JUST &                   //the result will be in the lowest 10 bits
 ADC_CH1  &                         //use pin AN1 i.e. RA1
 ADC_VREFPLUS_VDD &                 //set the reference maximum voltage to be 5V
 ADC_VREFMINUS_VSS &                //set the reference minimum voltage to be 0V
 ADC_INT_OFF, 0b01111101  );        //no interrupt is being set

while(1)
{
 ConvertADC();                      //capture analog volt & start ADC process
 while(BusyADC() )                  //until the conversion process completes
{                                   //do nothing and wait for conversion to happen
}

 ADCresult = ReadADC();             //get the result from the ADC peripheral.

 voltage = (double)ADCresult *(5.0)/ ADCMAX;
                                    //scale it into a voltage by multiplying by 5V/1023

 if(voltage > 2.8)                  //common LDR will light greater than 2.8 Volt
 LEDPIN = 1;
 Else
 LEDPIN = 0;
}
}
```

4.6 Pulse Width Modulation (PWM)

Digital signal only produces either 0 (low) or 1 (high) output. This disable the signal to control the speed of DC motor as it is only will ON or OFF the motor. However, there is a technique of controlling and varying the amount of power delivered to an electronics load using this ON and OFF digital signal, which is called as Pulse Width Modulation (PWM) technique. By this technique, the speed of DC motor, or any motor eventually, can be controlled.

On the other view, motor needs a lot more power at the starting up compared to when it is running. Motor also draws a lot more power when a robot goes uphill or pushes something. As the power requirements increase, it will quickly exceed the power rating on a resistor or potentiometer. The electronic component will get very hot and then will likely fail permanently. Furthermore, a resistor wastes excess power as heat. In a battery powered robot, it is preferred not to waste energy. Therefore, there are the reasons why PWM solution is needed in robotics controller

For this project, PIC18F4550 generates the PWM and send it to the connected output motor. PIC18F4550 itself has a built-in hardware, called Capture/Compare/PWM (CCP) module, to generate a PWM signal. Below is the sample of coding written in MPLAB IDE to program the PIC18F4550 to be operated in PWM function.

```
#include <pwm.h>                         //include library PWM functions

void main()
{
  PWM1_Init(5000);                        // PWM module initialization (5KHz)
  new_DC = 0;                             // Initial value of variable Duty Cycle
  current_DC = 0;
  PWM1_Start();                           // Start PWM1 module with Zero DC
```

```
  PWM1_Set_Duty(current_DC);
  do {
  if (!UP){                          // If the button connected to RB0 is
pressed
  debounce();
  if (new_DC < 250)                  // Don't go above 250
  new_DC = new_DC + 25 ;             // increment Duty Cycle by 25
 }

  if (!DOWN) {                       // If the button connected to RB1 is
pressed
  debounce();
  if (new_DC !=0)                    // Don't go below 0
  new_DC= new_DC - 25 ;              // Decrement Duty Cycle by 25
 }

  if (current_DC != new_DC) {
  current_DC = new_DC ;
  PWM1_Set_Duty(current_DC);         // Change the current DC to new value
 }
} while(1);
```

PWM works with the help of duty cycle in a period, which is the fraction of the signal that is ON for that period. The average DC value of the signal can be varied by varying the duty cycle. The duty cycle can be anywhere between 0 to 1. Suppose, if the signal has +5 Volt while it is ON and 0 Volt during OFF condition, then by changing the duty cycle of the signal, any voltage between 0 Volt to 5 Volt can be simulated.

To calculate the percentage of duty cycle, equation below has been referred.

$$Duty\ Cycle = \frac{t_{Duty}}{t_{Period}} x\ 100\%$$

Table 9 shows the percentage of the duty cycle with its illustrated signal graph.

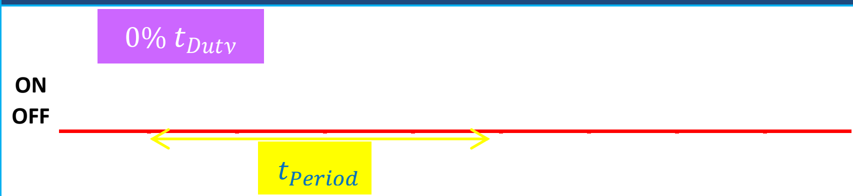
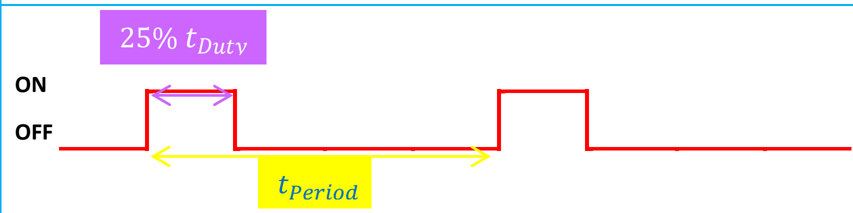
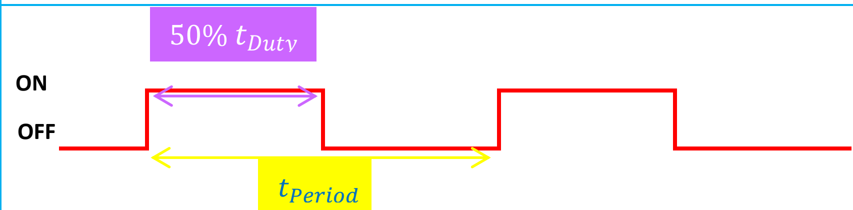
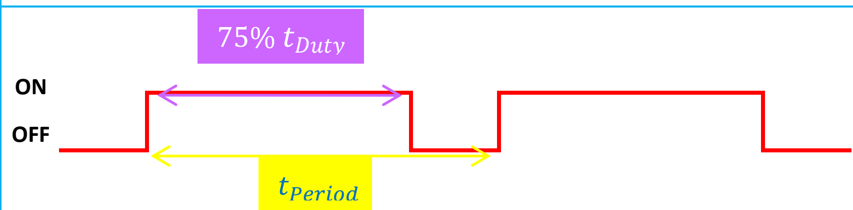
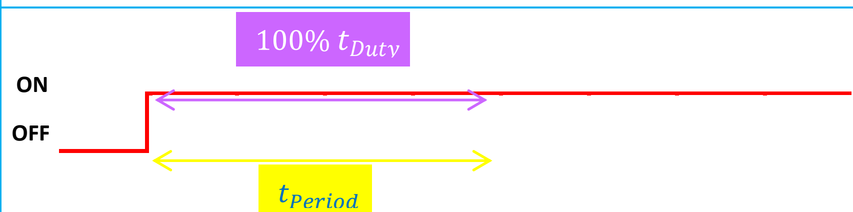
Percentage of duty cycle (%)	Graph illustrates the signal
0	
25	
50	
75	
100	

Table 9: *The PWM signal according to the percentage of the duty cycle*

These percentages of the PWM are later being tested with LED and DC Brushless Motor. The brightness of the LED and the speed of rotating DC Brushless Motor shaft are being recorded in the table 10.

Percentage of duty cycle (%)	Brightness of LED	Rotation of DC Brushless Motor Shaft
0	The LED does not light up. This means that it is in OFF mode.	The shaft does not move or rotate. This means that it is in OFF mode.
25	The LED starts to light up, but with the dim of light.	The shaft starts to rotate, but in the slow speed.
50	The LED lights up more bright compared to previous (25%).	The shaft rotates faster compared to previous (25%).
75	The brightness of the LED is increasing. Compared to 50%, 75% of duty cycle gives more bright of light.	The speed of shaft rotation is increasing. Compared to 50%, 75% of duty cycle gives more faster speed of rotation.
100	The LED brightens the most compare to others. This means, the output voltage is at the maximum which is approximately to +5V.	The speed of rotating shaft is the most fastest compared to others. This might because of the maximum

Table 10: *The result of applying PWM to the LED and DC Brushless Motor*

When using PWM, it is important to consider how slow the LED can be flashed so that the viewer does not perceive the oscillation. The eye's inability to see rapid oscillations of light is caused by our persistence of vision. This means, the light can be seen as ON even after it has turned OFF. This technique is how televisions display a seemingly moving picture which is actually made up of a number of different still frames displayed one after the other very rapidly. The minimum speed of an LED oscillating which can be seen by the human eye varies from person to person.

CHAPTER 5: CONCLUSION AND RECOMENDATION

Robotics is one of the vast, broad fast branches of technology which implement the current art of artificial intelligence. This technology will keeps on expanding toward more compatible and practical trend. Therefore, this project hopefully will contribute one step towards the mission.

In conclusion, the author is managing to achieve the objectives of the project in designing more practical controller board, compared to IFC board. The advantages of 35 Input/Output pins allow the board to be interface with many inputs and outputs. The optional of having external clock connection is to increasing the frequency of the circuit clock when being connected to the USB. Analog to Digital Converter (ADC) helps so the input analog signals become more compatible to be processed. For controlling the speed of motor in robotics, Pulse Width Modulation (PWM) technique is used to varying the voltage with act as signal to ON and OFF the motor, periodically.

Even the data storage in the designed circuit board does not as many as in the IFC board library, but it is enough for a simple robot to operate by itself. At the same time, the cost for preparing a simpler motion controller board is far more cheaply compared to buy the IFC board.

For further development of similar system, the author suggests to extend the system towards other robotics related motor, such as servo motor. Plus, rather than using the DC motor card, it is better to include the H-bridge circuit in the development board itself to reduce spaces on the working board of the autonomous robot.

REFERENCES

[1] *Oxford Dictionaries*

[2] M. N. Nik Nurshahirah, "Design & Fabrication of Autonomous Robot for Security Monitoring," Electrical & Electronics Engineering, Universiti Teknologi PETRONAS, Tronoh, Perak, Malaysia, 2012.

[3] N. A. Abdullah, "Design and Fabrication of Autonomous Vehicle for Letter/Parcel Delivery," Electrical and Electronics Engineering, Electrical and Electronics Engineering, Universiti Teknologi PETRONAS, Tronoh, Perak, Malaysia, 2012.

[4] M. A. Othman, "Design and Development of Autonomous Line Tracking Robot Using Microcontroller," Faculty of Manufacturing, Universiti Teknikal Malaysia Melaka, Melaka, Malaysia, May 2008.

[5] W. Mattias, *Introduction to Autonomous Robots* vol. -. Sweden: Chalmers University of Technology, 2012.

[6] "Interface Free Controller Main Board - User's Manual," C. T. S. Bhd, Ed., ed. Malaysia: Cytron Technologies, April 2008.

[7] RobotShop. *RobotShop Grand Tutorial*.

[8] "PIC18F2455/2550/4455/4550 Data Sheet," M. T. Inc., Ed., ed, 2009.

[9] "PICkit 3 Programmer/Debugger User's Guide," M.T. Inc., Ed., ed, 2010.

[10] "PICkit 3 Starter Kit User's Guide," M.T. Inc., Ed., ed, 2012.

[11] Microchip Technologies Website.

[12] Cytron Technologies Website.

APPENDICES

Coding for the circuit board

```
/************** Compiler Directives  (compdirectives.h)************************
All these settings are refer to
Help>>Topics>>LanguageTools>>PIC18ConfigSettings>>PIC18F4>>PIC18F4550*******/

#pragma config VREGEN = OFF             // USB voltage regulator disabled
#pragma config WDT = OFF                // Watchdog timer enabled
#pragma config PLLDIV = 5               // Divide by 5 (20 MHz oscillator input)
#pragma config MCLRE = OFF              // RE3 input pin enabled; MCLR pin disabled
#pragma config WDTPS = 32768            //
#pragma config CCP2MX = ON              // CCP2 input/output is multiplexed with RC1
#pragma config PBADEN = OFF             // PORTB<4:0> pins are configured as digital I/O
on Reset
#pragma config CPUDIV = OSC1_PLL2       // [Primary Oscillator Src: /1][96 MHz PLL Src:
/2]
#pragma config USBDIV = 2               // USB clock source comes from the 96 MHz PLL
divided by 2
#pragma config FOSC = INTOSCIO_EC       // Internal oscillator, port function on RA6, EC
used by USB (INTIO)
#pragma config FCMEN = OFF              // Fail-Safe Clock Monitor disabled
#pragma config IESO = OFF               // Oscillator Switchover mode disabled
#pragma config PWRT = OFF               // Power-up Timer Enable
#pragma config BOR = OFF                // Brown-out Reset enabled in hardware only
#pragma config BORV = 3                 // Brown-out Reset Voltage Minimum setting
#pragma config LPT1OSC = OFF            // Timer1 configured for higher power
operation
#pragma config STVREN = ON
#pragma config LVP = OFF
#pragma config ICPRT = OFF
#pragma config XINST = OFF
#pragma config DEBUG = OFF
#pragma config CP0 = OFF, CP1 = OFF, CP2 = OFF, CP3 = OFF
#pragma config CPB = OFF                         // CPB off
#pragma config CPD = OFF
#pragma config WRT0 = OFF, WRT1 = OFF, WRT2 = OFF, WRT3 = OFF
#pragma config WRTC = OFF
#pragma config WRTB = OFF
#pragma config WRTD = OFF
#pragma config EBTR0 = OFF, EBTR1 = OFF, EBTR2 = OFF, EBTR3 = OFF
#pragma config EBTRB = OFF
```

```
/********************** LED Setting ledsettings.h ***********************/
#define setpin() TRISB = 0;               // Setting all PORTB pins to output pin RB0-7

#define LD1     LATBbits.LATB0        // RB0
#define LD2     LATBbits.LATB1        // RB1
#define LD3     LATBbits.LATB2        // RB2
#define LD4     LATBbits.LATB3        // RB3

#define LED1_on()  LD1 = 1;           // High
#define LED2_on()  LD2 = 1;           // High
#define LED3_on()  LD3 = 1;           // High
#define LED4_on()  LD4 = 1;           // High

#define LED1_off()   LD1 = 0;         // Low
#define LED2_off()   LD2 = 0;         // Low
#define LED3_off()   LD3 = 0;         // Low
#define LED4_off()   LD4 = 0;         // Low


/**********************Define Include Header File**********************/
#include<p18f4550.h>             // Include the PIC18F550 Headers
#include "compdirectives.h"      // User defined header: Chip Config | Complier directivs
#include "ledsettings.h"         // Port settings

#define switchpin TRISAbits.TRISA0   // Define A0
#define ADCMAX 1023

int ADCresult;
float voltage;

/****Set Timer****/

void delay1(void)                 // 25%
	{	int i, j;
		for(i=0;i<100;i++)
	{
			for(j=0;j<1;j++)
			{
			}
		}
	}

void delay2(void)                 // 50%
	{	int i, j;
```

```
		for(i=0;i<66;i++)
	{
			for(j=0;j<1;j++)
			{
			}
		}
	}


void delay3(void)		// 75%
	{	int i, j;
		for(i=0;i<33;i++)
	{
			for(j=0;j<1;j++)
			{
			}
		}
	}

void delay4(void)		// 100%
	{	int i, j;
		for(i=0;i<1;i++)
	{
			for(j=0;j<1;j++)
			{
			}
		}
	}


void main(void)	// The main
{

switchpin();
setpin();
OSCCON = 0x70;
ADCON1 = 0b01111101;
DDRB=0b11111111;
DDRA=0b00000010;

OpenADC( ADC_FOSC_32    &
ADC_RIGHT_JUST &
ADC_4_TAD,
ADC_CH1  &
```

```
ADC_VREFPLUS_VDD &
ADC_VREFMINUS_VSS &
ADC_INT_OFF, 0b01111101  );

while (1)
{

ConvertADC();

                    while(BusyADC() )

                    {
                    }

                    ADCresult = ReadADC();

                    voltage = (double)ADCresult *(5.0)/ ADCMAX;

          if(voltage>2.0)
                              {
                              LED1_on();

                              }
                    else
                              {
                              LED1_off();
                              }


if(PORTBbits.RB2>>0)
          {
                                        LED1_on();                              // Off led 1
                                        delay1();
                                        LED1_off();
                                        delay1();
          }

else if(PORTBbits.RB3>>0)
          {
                                        LED1_on();                              // Off led 1
                                        delay2();
                                        LED1_off();
                                        delay2();
          }
```

```
else if(PORTBbits.RB4>>0)
	{
				LED1_on();			// Off led 1
				delay3();
				LED1_off();
				delay3();
	}

else if(PORTBbits.RB5>>0)
	{
				LED1_on();			// Off led 1
				delay4();
				LED1_off();
				delay4();
	}
	}
}
```