

DESIGN OF CONTROL  
TEC



DRIVE SYSTEM FOR  
R

MOHD HASNUL MUBAKIR BIN OSMAN

A report submitted in fulfilment of the requirements  
for the award of the degree of  
Bachelor of Mechatronics Engineering

Faculty of Manufacturing Engineering  
UNIVERSITI MALAYSIA PAHANG

JUNE 2013

**PERPUSTAKAAN  
UNIVERSITI MALAYSIA PAHANG**

<b>No. Perolehan</b> <b>080235</b>	<b>No. Panggilan</b> TK 4058 1407 2013 70 60.
<b>Tarikh</b> <b>08 NOV 2013</b>	

## **EXAMINER'S APPROVAL DOCUMENT**

We certify that the thesis entitled “Design of Control Signal for Electrical Drive System for Technology Demonstrator” by Mohd.Hasnul Mubakir Bin Osman. We have examined the final copy of this thesis and in our opinion; it is fully adequate in terms of scope and quality for the award of the degree of “Bachelor of Mechatronics Engineering”. We herewith recommend that it be accepted in fulfilment of the requirements for the degree of Bachelor of Mechatronics Engineering.

Name : ASSOC. PROF.DR AJISMAN APEN

Signature :

Position : LECTURER (FKP)

## **SUPERVISOR'S DECLARATION**

I hereby declare that I have checked this thesis and in my opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Engineering Manufacturing or Bachelor of Mechatronics Engineering.

### **SUPERVISOR**

Signature :

Name : ASSOC.PROF.DR ABDUL AZIZ BIN JAAFAR

Position : LECTURER (FKP)

Date : 19<sup>th</sup> JUNE 2013

## **STUDENT'S DECLARATION**

I declare that all parts of this report are the results of my own work and all references have been cited adequately as required by the university.

Signature :

Name : MOHD HASNUL MUBAKIR BIN OSMAN

Matrix No. : FB09029

Date : 19<sup>th</sup> JUNE 2013

## **ACKNOWLEDGEMENT**

The special thanks goes to my helpful supervisor, Assoc.Prof.Dr Abdul Aziz Bin Jaafar. The supervision and support that she gave truly help the progression and smoothness of the final year project. The co-operation is much indeed appreciated.

My grateful thanks also go to both of my parent Osman Bin Awang Samin and Norul Huda Binti Samsudin cause they give big contribution and supportive from start until the end of my final year project. The project cannot be done without the enthusiasm and imagination from them.

Last but not least, great appreciation go to all my beloved friends cause always help me from the time to time during project. This really brought us together to appreciate the true value of friendship and respect of each other.

## ABSTRACT

Drive System for Omni-directional 4 wheel mobile platform consists of combination between electrical mechanism and mechanical mechanism. The most important part for this Mobile Platform is to drive the DC Motor. The control system development from digital signal provided by transmitter and receiver (Remote Control) to control the Motor Speed is the main problem for this study.

This study also required the research on conversion of digital signal to digital signal (DC-to-DC) by using Programmable Integrated Circuit (ATMega 328P). This study is very important not only to drive the DC motor but it also help to read information from any sensor signal (analog and digital) and providing a readable data display.

The research on Digital Signal from receiver also has been made during this study. The advance Remote Control such as FUTABA has its own PWM signal with its own fix frequency and duty cycle. By using 555 timer, Receiver PWM Wave Generator has been made. This device is able to change its frequency and duty cycle freely. This generator helps this study to build up a receiver signal as same as other Advance Remote Control Receiver signal.

## ABSTRAK

Sistem panduan untuk kenderaan 4 tayar Omni-directional adalah kombinasi antara electrical mekanisma dan mekanikal mekanisma. Perkara yang paling penting dalam membina kenderaan ini ialah menghasilkan sistem pandu untuk DC Motor. Sistem kawalan yang perlu dibina adalah hasil daripada signal digital daripada transmitter dan juga receiver (Remote Control) digunakan untuk mengawal Kelajuan Motor merupakan masalah utama projek ini.

Kajian ini juga melibatkan pertukaran Signal Digital ke Signal Digital (DC-to-DC) dengan menggunakan ATmega 328P. Kajian ini penting bukan sahaja untuk mengawal DC motor tapi juga membantu untuk membaca data daripada Digital Signal dan mempamirkannya.

Kajian berkenaan dengan Digital Signal dari Receiver semasa experiment ini. Kebanyakan model alat kawalan jauh yang lebih canggih seperti FUTABA mempunyai signal PWMnya yang tersendiri dengan frekuensi yang tetap. Dengan menggunakan 555 timer, Receiver Wave Generator telah dibina. Alat ini berkeupayaan untuk menghasilkan Signal PWM malah boleh mengawal frekuensi dan juga duty cycle secara bebas. Alat ini juga mempunyai mana-mana Receiver PWM signal yang lain.



## TABLE OF CONTENT

	<b>Page</b>
<b>Examination Approval</b>	<b>ii</b>
<b>Supervisor's Declaration</b>	<b>iii</b>
<b>Student's Declaration</b>	<b>iv</b>
<b>Acknowledgement</b>	<b>v</b>
<b>Abstract</b>	<b>vi</b>
<b>Abstrak</b>	<b>vii</b>
<b>Table of Content</b>	<b>viii</b>

### CHAPTER 1

#### INTRODUCTION

1.1	Project Background	1
1.2	Problem Statement	2
1.3	Objectives of the Project	3
1.4	Scopes of the Project	3

### CHAPTER 2

#### LITERATURE REVIEW

2.1	Permanent Magnet DC Motor	4
2.2	Pulse Width Modulator (PWM)	5
2.3	555 Timer IC	7
2.4	555 Timer Astable Oscillators	8
2.5	DC Motor Drive System	9
2.6	16-bit Microcontroller (Arduino)	10

## **CHAPTER 3**

### **METHODOLOGY**

3.1 Design of RC Signal Generator	11
3.1.1 Signal Acquisition	11
3.1.2 PWM in Arduino	14
3.2 Study on Electrical Mechanism for Motor Control	15
3.2.1 Device with H-Bridge	15

## **CHAPTER 4**

### **RESULT AND DISCUSSIONS**

4.0 Introduction	18
4.1 RC Signal Generator	18
4.1.1 CAD Design	18
4.1.2 Development of Electrical and Electronic Fabrication	22
4.2 RC Equal Acquisition	24
4.2.1 Data comparison and discussion	26
4.3 Motor speed control	27
4.4 RC to Motor Speed Control Conversion	28
4.5 Implement of PWM generator on AGV	30

## **CHAPTER 5**

### **CONCLUSION**

5.1 Conclusion	31
5.2 Recommendation	31
5.3 Reference	32

### **APPENDIX**

# CHAPTER 1

## INTRODUCTION

### 1.1 Project Background

KANBAN System is an alternative scheduling system for just-in-time (JIT) and lean productions (Taiichi Ohno (1953)). Electronic KANBAN or e-KANBAN is a signalling system that uses a mix of technology to trigger the movement of materials within manufacturing or production facilities. E-KANBAN uses technology to replace traditional elements such as KANBAN cards and runners with barcode with Automation Guided Vehicle (AGV). In practice, an AGV is designed to sustain a prescribed product loading and vehicle components.

Automated Guided Vehicle (AGV) is widely used by manufacturing facilities or a warehouse to help the factory increasing their KANBAN efficiency and reducing cost. It been used in a wide variety of application to transport many different types of material including pallets, rolls, rack, cart, and container. AGV also become famous in healthcare industry for efficient transport, and been programmed fully integrated with automatic open door, elevator/lift, cart washer, trash dumper, etc.

Most common drive motor use for AGV in industries is brushless DC motor. This is because brushless offered more torque per weight and more torque per watt (increased efficiency). Brushless DC Motor also well known to have lots of advantages rather than using Brush DC Motor. Even in a RC (Remote Control) hobby, Brushless Motor is widely use and the controller signal already fully integrated with electronic speed controller (ESCs) where the signal from the Controller Receiver can directly controlling the motor speed. However, Brushless Motor has a higher cost than the brush DC Motor because it is required complex

ESCs to run, which are also very expensive, and some practical uses have not well developed in commercial sector.

Because of this factor, the developments and improvement of Brush DC Motor with high torque for AGV and RC purposes is limited. Fortunately, Brush DC Motor still widely use in other industrial application such as automotive and robotic because of its simplicity and affordability and this allowing the improvement in Brush DC Motor drive system and application. The equipment like a power window, DC Geared Motor, DC Motor Driver using H-MOS is an example of equipment development in Brush DC Motor field. The electronic speed controls (ESCs) that have been developed now for DC Motor seems much cheaper than Brushless DC motor. This is because Brush DC Motor do not required complex electronic component for speed control.

For this project, permanent magnet brush DC motor will be used as motor for AGV 4 wheel Omni directional type Vehicle for e-KANBAN purpose with RC integrated system will be synthesized.

## **1.2 Problem Statement**

The main problem for this project is to synthesize drive system for Automatic Guided Vehicle (AGV) with permanent magnet (PM) brush DC motor as the prime with radio control integrated system. This drive system requires mechanism to convert a signal from RC signal to speed controller of the DC Motor signal. The suitable mechanism for converting this signal is by developing DC-to-DC converter because both signals are Digital Signal with voltage equal or less than 5V.

The easiest and cheapest way for converting digital signal is by using microcontroller. The computer code that will be created must be able to read digital and produce signal for controlling motor speed. The coding also must be able to display data from RC control. Theoretically, RC signal and ESCs signal are Pulse Width Modulator (PWM). Creating suitable programming to produce PWM signal for motor speed control is the key to solve this problem. To reduce the cost, microcontroller with 16-bit processing will be suitable to use.

### **1.3 Objectives of the Project**

General objective for the study is to develop drive system for AGV. The specific objectives of the study as:

- i. To synthesizes drive system of mobile platform for model of automated guided vehicle (AGV).
- ii. To integrate RC (Remote Control) system into the automated guided vehicle (AGV) control system.
- iii. To design and implement DC-to-DC converter and data display in to AGV control system.

### **1.4 Scopes of the Project**

The study will be limited to the usage of permanent magnet brush motor drive system. The microcontroller Arduino IDE system will be use study to create suitable program implementation to the drive system. The experimented drive system will be used to the AGV model.

## CHAPTER 2

### LITERATURE REVIEW

#### 2.1 Permanent Magnet DC Motor

DC motor operation is consists of stator, commutator and rotor. The stator is always provided the magnetic field and fixed in the motor. Rotor is a removable part that creates torque while the motor operate. Commutator is a rotary switch that reverses the current supplied to the motor at each cycle so that the motor is rotating at the same direction. [11]

There are various types of DC motor. There are permanent magnet DC motors, separately excited DC motor and series DC motor. Permanent magnet DC motor is using permanent magnet at its stator. The magnet is installed into the motor to excite the magnetic field. The permanent magnet is more reliable, sturdy and compact because the field fluxes remain constant for all loads. It is also more linear to speed torque character. The disadvantage of permanent magnet DC motor is that its speed can't be controlled when above base speed.[11]

## 2.2 Pulse Width Modulator (PWM)

Pulse Width Modulator is a technique to control the width of the pulse signal, based on modulator signal information. The modulator technique can be used to encode information, transmission of data and controlling the power supplies to electrical device.

The average value of the out voltage ( $V_{avg}$ ) applies to the load is controlled by turning the switch between supply and load on and off at a selected frequency. The longer the switch is on compared to the off periods, the higher the power supplied to the load is. The switching mechanism produces two variables; the duty cycle and frequency of PWM. Both variables are use to manipulate PWM signal for communication signal that involve transmitting and reading data.

Duty cycle is the proportion of 'on' time to the regular interval or 'one period' of time; a low duty cycle corresponds to low power, because the power is off for most of the time. Duty cycle is expressed in percent, 100% being fully on.

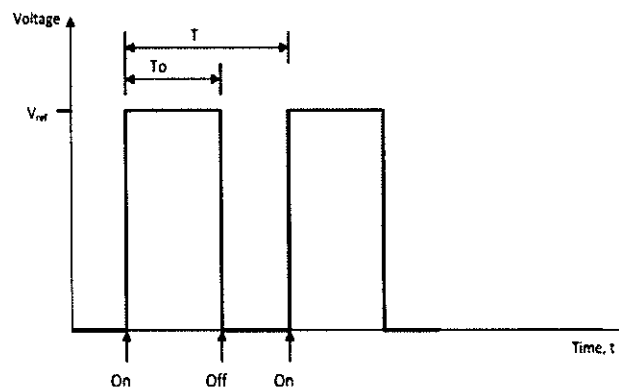


Figure 2.1: Duty cycle

The PWM switching frequency has to be much faster for engaging the power to the load. Typically switching for lamp dimmer is 120 Hz and from few kilohertz (kHz) to tens of kHz for a motor drive and well into the tens or hundreds of kHz in audio amplifiers and computer power supplies.

Equipment	Frequency
Lamp Dimmer	120Hz
Motor	1kHz – 10kHz
Audio Amplifiers	10kHz – 100kHz
Computer Power Supply	10kHz – 100kHz

Table 2.1: Frequency implementation in equipment

The main advantage of PWM is that power loss in the switching devices is very low. When a switch is off there is practically no current, and when it is on, there is almost no voltage drop across the switch. Power loss, being the product of voltage and current, is thus in both cases close to zero. PWM also works well with digital controls, which, because of their on/off nature, can easily set the needed duty cycle.



### 2.3 555 Timer IC

555 Timer IC is an integrated circuit use widely for timer, pulse generation and oscillator applications[10]. This IC can be used to generate pulse-width modulation (PWM) signal with few external components. There are three operating modes perform by 555 Timer. They are:

Modes	Description
Monostable mode	<ul style="list-style-type: none"> <li>➤ 555 Timer react as “one shot” pulse in generator.</li> <li>➤ Pulse begin when 555 timer receive a signal at trigger input that fall below a third of the voltage supply.</li> <li>➤ Output pulse width be determine by the time constant of RC network , consist with a capacitor and resistor</li> <li>➤ Pulse end when voltage on capacitor equals 2/3 of the supply voltage.</li> </ul>
Bistable mode	<ul style="list-style-type: none"> <li>➤ 555 Timer react as a basic flip-flop.</li> <li>➤ The trigger and reset inputs (pin 2 and 4) held high by pull-up resistor.</li> <li>➤ Threshold input (pin 6) grounded.</li> <li>➤ No capacitor required in bistable configuration.</li> </ul>
Astable mode	<ul style="list-style-type: none"> <li>➤ 555 Timer output able to produce PWM signal with duty cycle more than 50%.</li> <li>➤ Consist of external component.</li> <li>➤ Resistor 1 (R1) is connected between Vcc and the discharge pin.</li> <li>➤ Resistor 2 (R2) is connected between</li> </ul>

	<p>the discharge pin, the trigger pin and the threshold pin.</p> <p>➤ The PWM pulse frequency depends on the value of Resistor 1, Resistor 2 and Capacitor.</p>
--	---

Table 2.2: 555 Timer Modes

### 2.4 555 Timer Astable Oscillators

The astable oscillators with particular frequency and mark ratio can be design by calculating the periodic of time required from the required frequency and the discharge time and change time using formulae for t. Some external component will be needed. [10] The components are Resistor 1, Resistor 2, Capacitor 1 and Capacitor 2

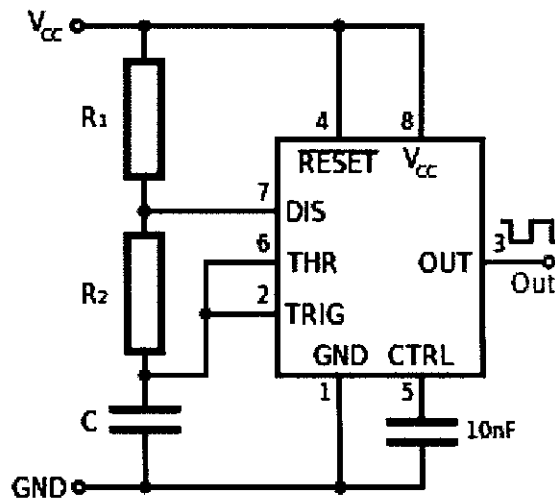


Figure 2.1: Astable Timer

To selecting suitable value of C1 component for period of time desired, a suitable value can be determined by using diagram in Figure 2.

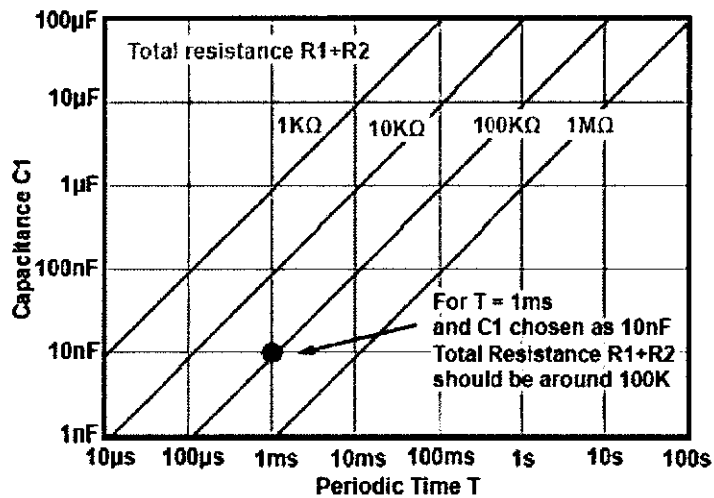


Figure 2.2: Varies capacitor 1 value

From the Figure 2, it also shows the total maximum value of resistor that may be can be used. The total value of resistor can be up to 10 or 20MΩ. However, using such high value can increase the error between calculated and actual frequency, so only 1MΩ maximum is recommended. The minimum total resistance value for combination of R1 and R2 still depends on the value of R1.[10]

## 2.5 DC Motor Drive System

To control the directional of the DC motor, H-bridge application is the right application to be done with. H-bridge is a link that binding electrical digital signal with mechanical action. H-bridge usually consisting with switching component such as BJT or MOSFET because it required switching around 1kHz to 10kHz for controlling a motor speed.

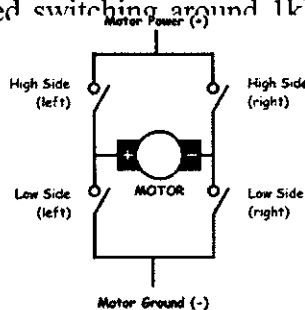


Figure 2.3: H-bridge

## 2.6 16-bit Microcontroller (Arduino)

Arduino is the one of the popular microcontroller board used for creating individual projects. This board been designed to be really easy to use. With free open source code, any project can be done without becoming an expert in programming.

Arduino are also consisting with Plug-In expansion such as Motor Driver, Wireless, Bluetooth and even telephone sim reader. This feature also call “Shields”. There are 4 basic functional elements in Arduino, which is:

- An Atmel “ATmega328P” AVR microcontroller
- A simple 5V power supply
- A USB-to-serial converter for loading new programs onto the board
- I/O headers for connecting sensors, actuators, expansion boards, etc

For Arduino UNO there are 3 internal timers is installed in it. They are timer0, timer1 and timer2. Timer1 is 16-bit and timer0 and 2 it is 8-bit. This timer is important to represent how much PWM channel it can produce in 1 operation[1].

For Arduino MEGA 2560 have an additional 3 more timer which is timer3, timer4 and timer5. All of them are 16-bit. [2]

## CHAPTER 3

### METHODOLOGY

#### 3.1 Design of RC Signal Generator

The design of RC signal generator is purposed to watch the signal produce by the receiver of commercial radio controller model KX-7 and other types of advance remote control. Figure 3.1 below show signal that can be produce by a KX-7. KX-7 has a fixed frequency at 50Hz and producing controllable duty cycle at 5% to 9% which is producing “ON” time around 1ms to 2ms with period of 20ms.

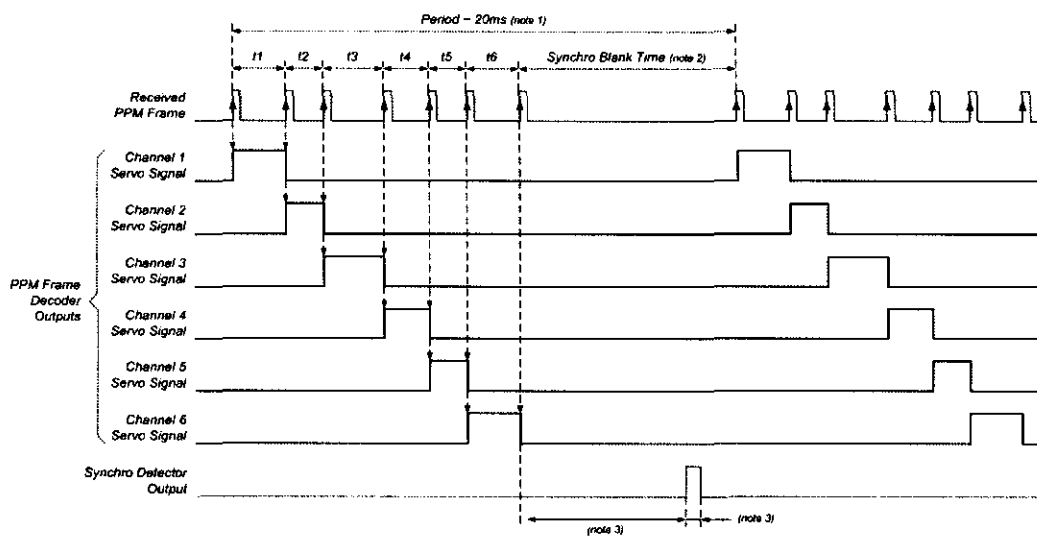


Figure 3.1: KX-7 receiver signal

By manipulating capability to produce PWM signal from astable mode circuit using 555 Timer, the output signal is use to synthesize the RC actual signal. However, this circuit unable to produce duty cycle less than 50%. Therefore, other signal modes need to be approach.

By combining astable mode circuit with monostable mode circuit, the RC signals can be synthesized. By transferring an output signal from astable mode circuit to the trigger pin monostable mode circuit, allowing output to produce PWM signal with same frequency as astable timer circuit. Duty cycle for RC is depends on the total resistance of monostable timer circuit.

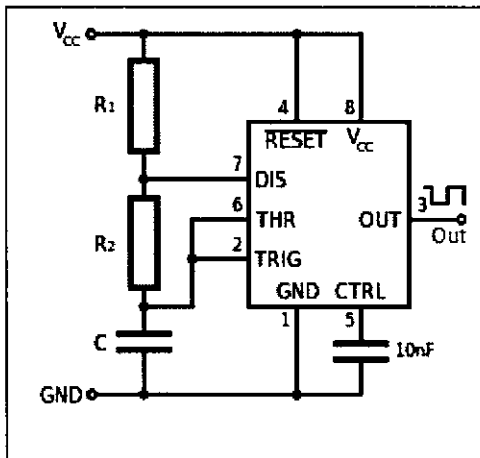


Figure 3.2: Astable Mode Circuit

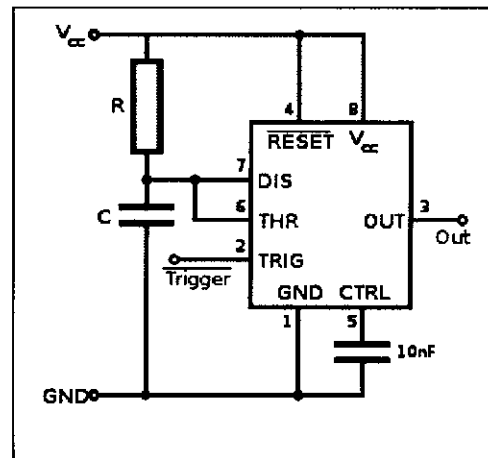


Figure 3.3: Monostable Mode Circuit

To manipulate the output signal frequency, the total resistance in the astable timer circuit must become unfixed.

By referring to the Figure 2, to produce 50Hz frequency, the total resistance and capacitor for astable mode circuit need to be determined. The arrangement for 1 cycle is 20ms. By referring to the Figure 2, it shows that 100nF capacitor require around 200k $\Omega$  total resistance. To produce duty cycle at 1ms to 2 ms, total resistance for monostable mode timer circuit is 20k $\Omega$ .

### 3.1.1 Signal Acquisition

Arduino has capabilities to read analog and digital signals. For analog signal, Arduino has built-in analog to digital converter (ADC) that allow Arduino change signal from 0V to 5V into 1024 different voltage levels or intervals. 0V is for 0 intervals and 5V is for 1024 interval. Analog signal can be read by analog pin. Arduino UNO is design with 6 analog outputs providing 6 analog channels [7]

Arduino UNO is capable to read a PWM signal. There is several function in Arduino IDE that can be used to read the signal. PulseIn function is able to read pulse, high or low on a pin. By selecting HIGH, pulseIn() will wait until the pin is HIGH, start timing, the waiting for the pin to go LOW and stops timing.[7] The length of pulse will be read in microsecond. The other approach is by using Timer and Interrupt. Timer/counter is hardware built in the Arduino controller. This timer can be programmed by some special registers. In Arduino UNO and MEGA 2560, 8-bit timer and 16-bit timer has been used. The timer is depends on the system clock. Interrupt is an external event that interrupts the running program. It is operates on special interrupt service routine (ISR). In Arduino, interrupt can generally enable and disable by using function interrupts() / noInterrupts(). The Arduino function attachInterrupt() and detachInterrupt() can only be use for external interrupt pins. To read a signal attachInterrupt() function can be use. [8]

### 3.1.2 PWM in Arduino

In Arduino, load PWM can be produced by using several ways using the Arduino IDE by simply call `analogWrite(output pin, duty cycle)`, producing PWM signal where “duty cycle” is a value from 0 to 255, and “output pin” is one of the PWM pins (3, 5, 6, 9, 10, or 11). The `analogWrite` function provides a simple interface to the hardware PWM, but does not provide any control over PWM frequency.[7]

PWM also can be generate by using `delay()` function. This can be done by user interface repeating the selected pin on and off. This technique has the advantage that it can use any digital output pin. It have full control the duty cycle and frequency. One major disadvantage is that any interrupts will affect the timing, which can cause considerable jitter unless interrupts. A second disadvantage is the output running cannot be left while the processor does something else. It is difficult to determine the appropriate constants for a particular duty cycle and frequency.[7]

Example of programming using manually `delay()` function:

```
void setup()
{
  pinMode(13, OUTPUT);
}

void loop()
{
  digitalWrite(13, HIGH);
  delayMicroseconds(100); // Approximately 10% duty cycle @
1KHz
  digitalWrite(13, LOW);
  delayMicroseconds(1000 - 100);
}
```



### 3.2 Study on Electrical Mechanism for Motor Control

#### 3.2.1 Device with H-Bridge

H-Bridge is the most important way to controlling the motor direction. There is lots of device and IC with the application of H-Bridge. L293D is a simple IC example that consisting of 2 H-Bridge in it. It is able to control 2 DC motor at one time. However, this type of only capable to running a motor at best rate with current up to 0.6A only.[4]

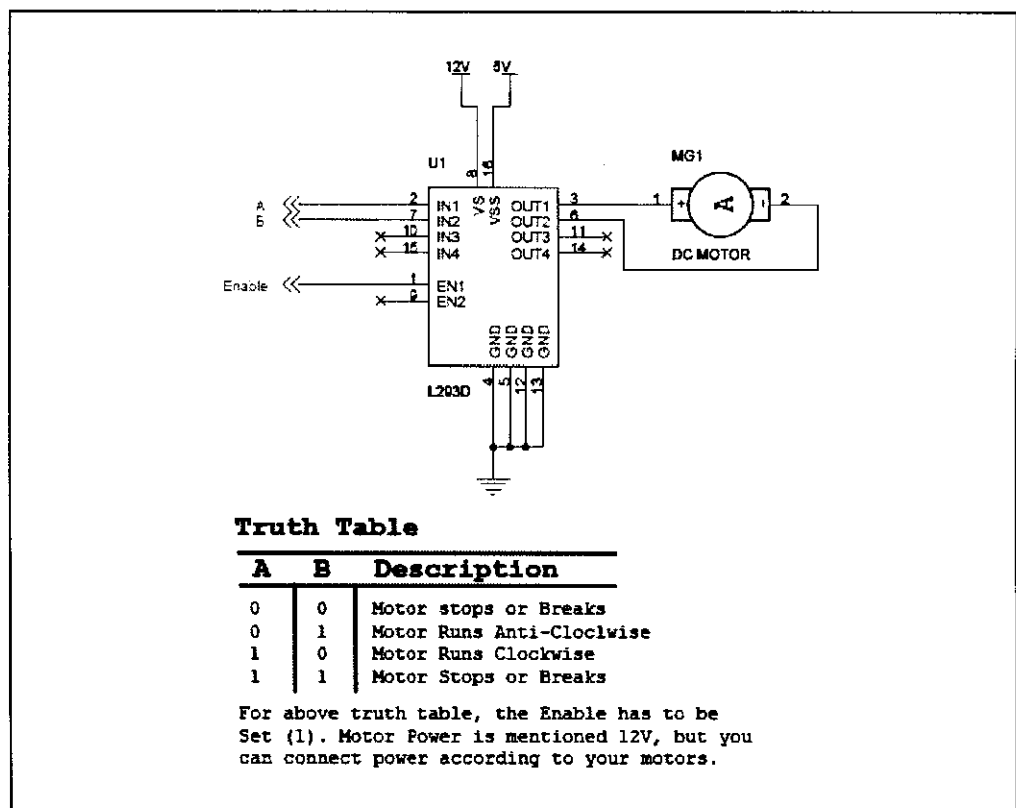


Figure 3.4: L293D pin and Table of Truth

By referring to the figure, L293D required a digital signal to switch on the direction at pin 2,7,10 and 15. This IC also required extra 5V source to make it enable by supplying it to pin 16 (Vcc), and also required supplying it to pin 1 to enable Motor1 and pin 9 to enable Motor2. This IC also required 12V or less power supply to power up the motor.[4]

To perform a motor with current up to 30A, the other type of motor driver is required. MD30B is a motor driver capable to run a single DC motor up to 30A. This device required PWM signal up to 10kHz to perform motor speed control. The PWM signal is usually can be generated from microcontroller. However, this device is an old version and have complex building component.[5]



Figure 3.5: MD30B Motor Driver

However, the connection to operating this device is less complicated than L293D. It only need 1 PWM signal up to 10kHz, 2 pin for direction control and 12 V power source to power up the motor. [5]



Figure 3.6: MD30B combining with microcontroller

SHIELD-MD10 is an accessories for Arduino. Compatible with Arduino IO pin, this driver capable to support 10A of current continuously. It supports power supply up to 25V. The H-Bridge is made by fully NMOS and it does not require any heat sink. This driver is capable of controlling only one motor.

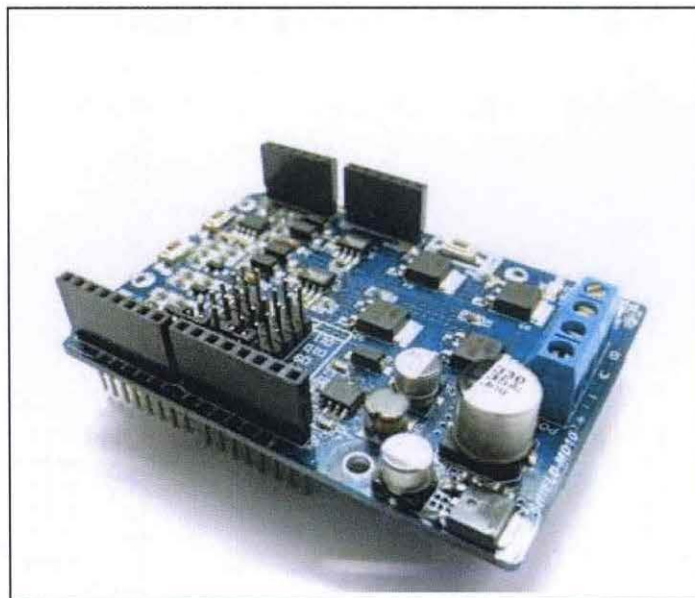


Figure 3.7: SHIELD-MD10B

However, the connection for motor control is much better than L293D and MD30B. It required only PWM signal and only one pin to control the direction. Users just have to make sure that the dir pin is intact with IO pin use.[6]

## CHAPTER 4

### RESULT AND DISCUSSIONS

#### 4.0 Introduction

In this chapter, data collect and the experimental result will be determined. The main contribution in this study is programming development by using 16-bit microcontroller, for converting the RC signal that able to control permanent brush DC motor speed.

#### 4.1 RC Signal Generator

##### 4.1.1 CAD Design

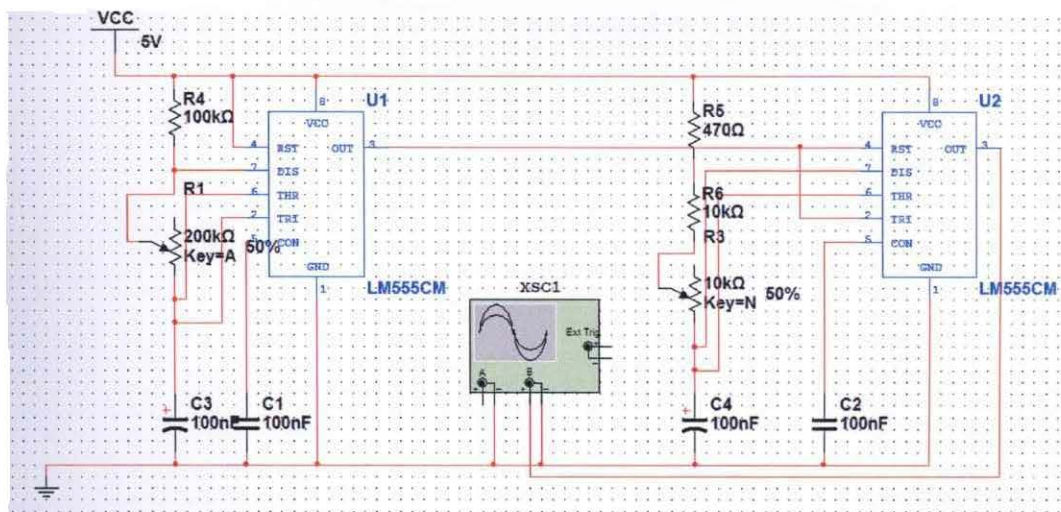


Figure 4.1: RC Signal Generator Simulation

Figure 4.1 shows that combination between astable mode timer circuit with monostable mode timer circuit. Potentiometer from astable timer circuit allows changing the PWM frequency freely and potentiometer at Monostable Timer allow to changing the duty cycle freely. Figure 4.2 show the Ideal signal produced by this simulation. Figure 4.3 show the simulation for minimum value for duty cycle. The simulation data shows that the equipment is capable to produce RC signal.

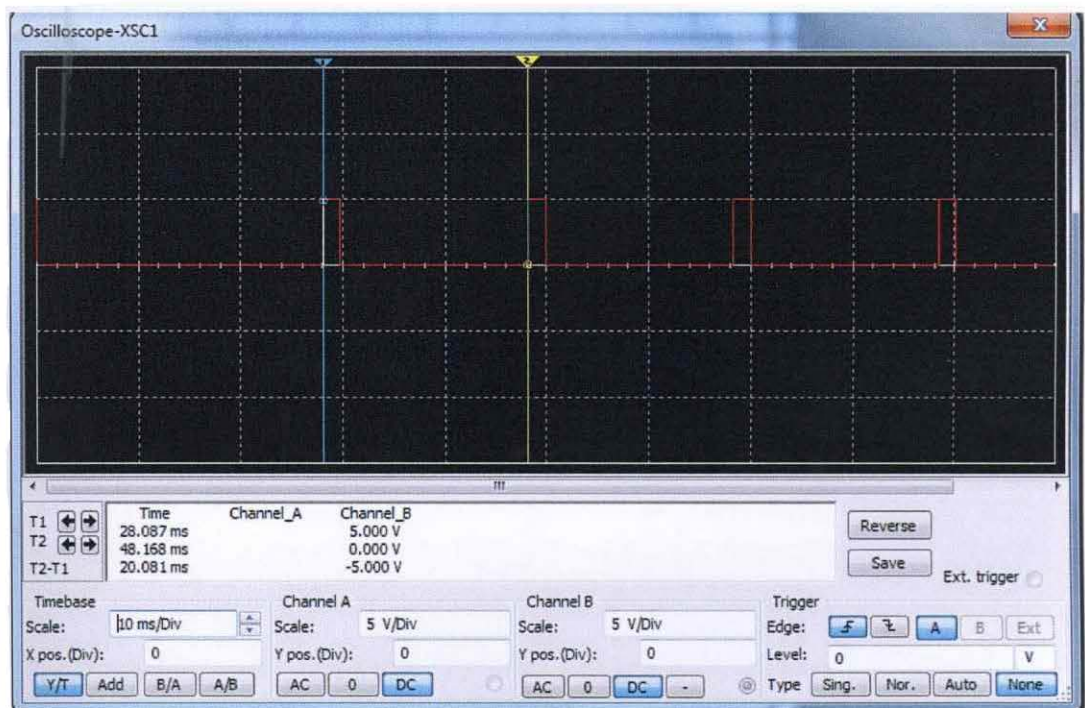


Figure 4.2: Period

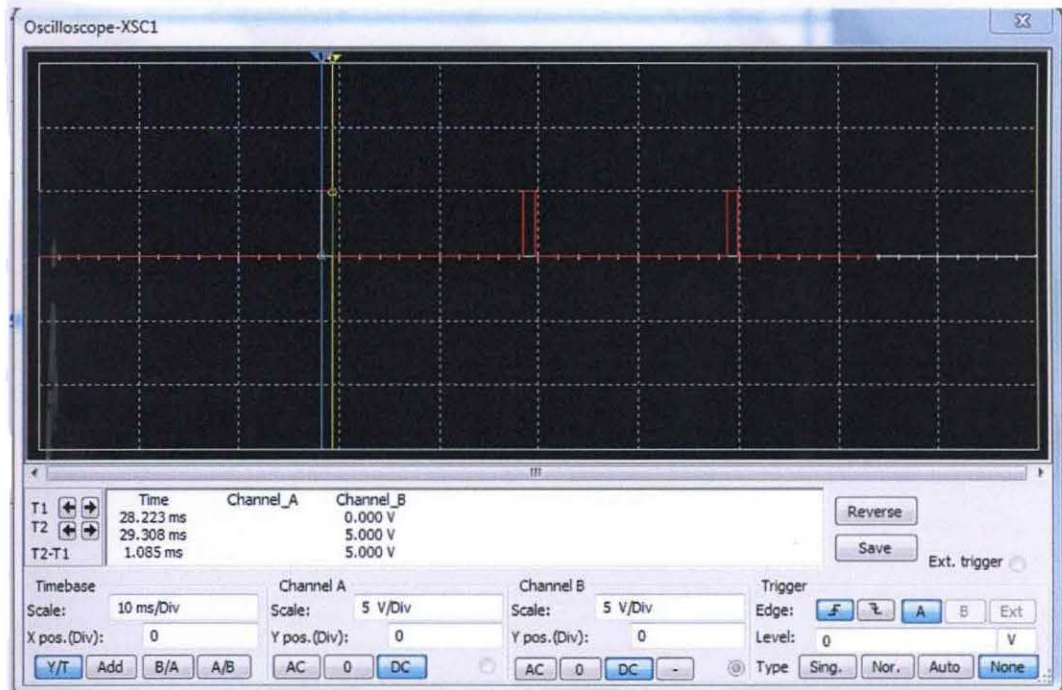


Figure 4.3: Lowest Duty Cycle

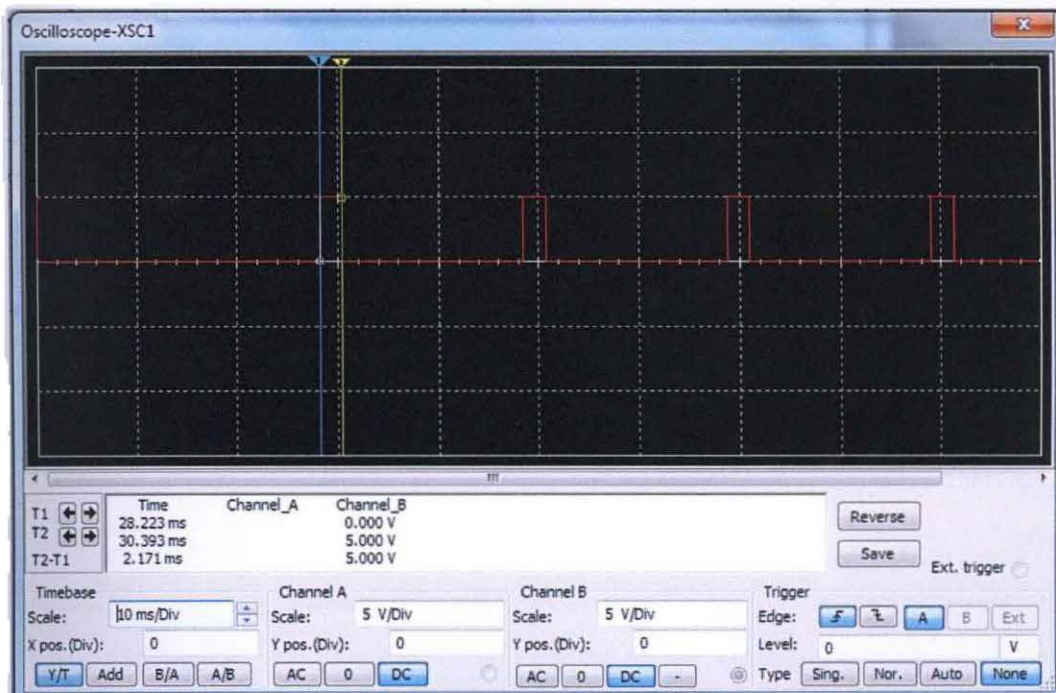


Figure 4.4: Highest Duty Cycle

This simulation shows that by changing the value of Period is 20 ms which is 50Hz frequency. However, from this experiment, it shows that the value of potentiometer needs to be higher than 100k $\Omega$  to producing much higher frequency.

The signal duty cycle can be change by varying the total resistance in the monostable timer circuit. To change the resistance value, 10k $\Omega$  potentiometer is used to vary the total resistance in the monostable timer circuit. For the monostable circuit, there are 3 resistance used to build up total resistance 20k $\Omega$ .

#### 4.1.2 Development of PWM Generator Board

All component first need to be assembles by using bread board. This is a typical way to test where the components are able to produce the same data as simulation or not. After the circuit successfully produce the data, it then undergoes itching process. A completed circuit then assemble with its housing shown in Figure 4.6.

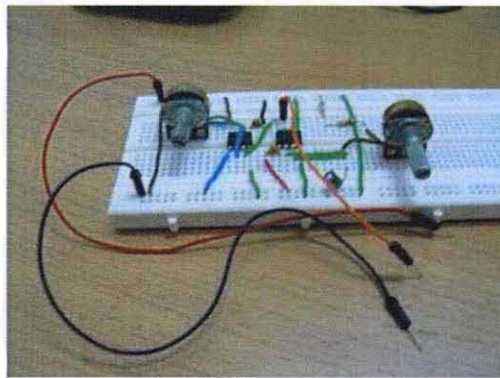


Figure 4.5: Circuit being built up by using Bread Board

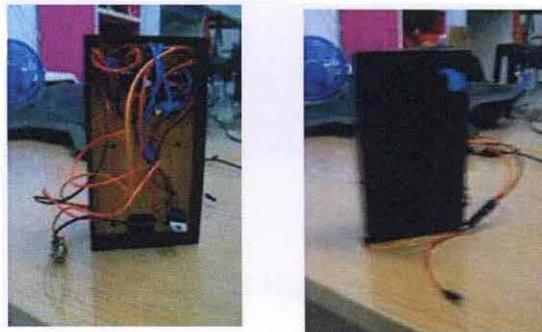


Figure 4.6: Circuit after Itching process and assemble with housing



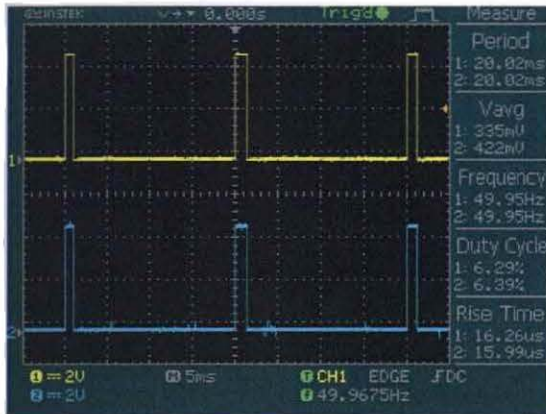


Figure 4.8: Comparison between KX-7 signal and RC wave generator

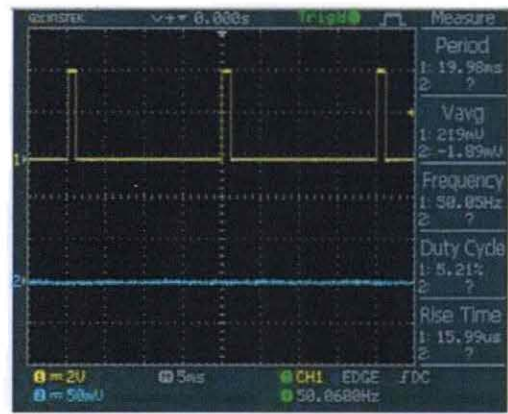


Figure 4.9: Lowest duty cycle

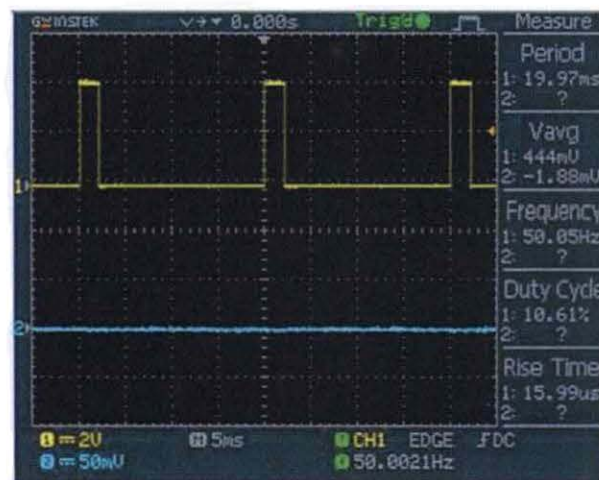
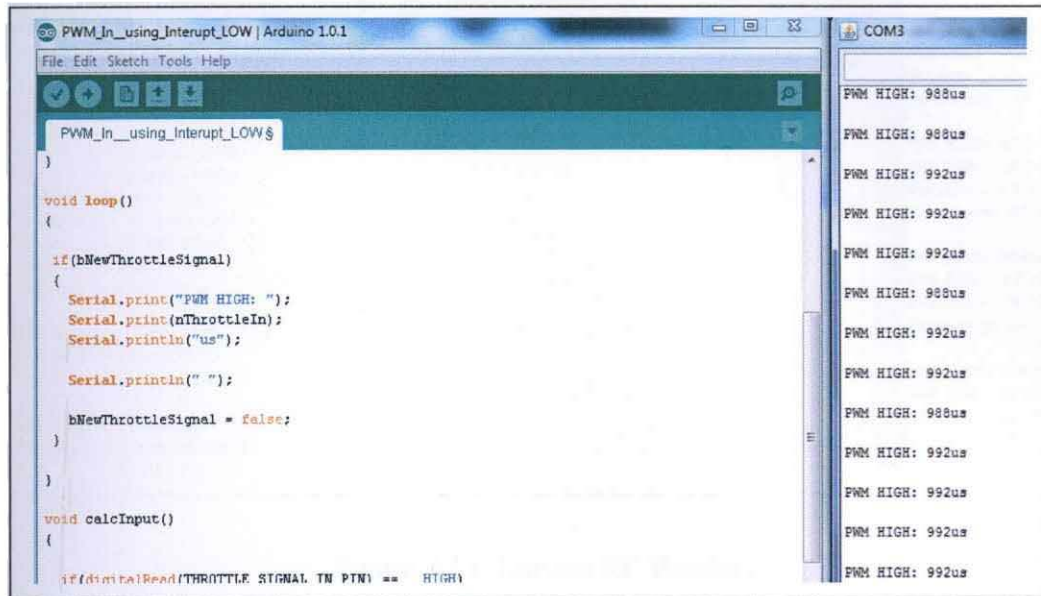


Figure 4.10: Highest duty cycle

The result show that this device successfully producing RC signal with fixed 50Hz and producing controllable duty cycle. This device is able to produce 5.21% to 10.61% duty cycle which mean that, it able to produce HIGH signal from 1ms to 2ms.

## 4.2 RC Equal Acquisition



```
PWM_In_using_Interupt_LOW$
}

void loop()
{
  if(bNewThrottleSignal)
  {
    Serial.print("PWM HIGH: ");
    Serial.print(nThrottleIn);
    Serial.println("us");

    Serial.println(" ");

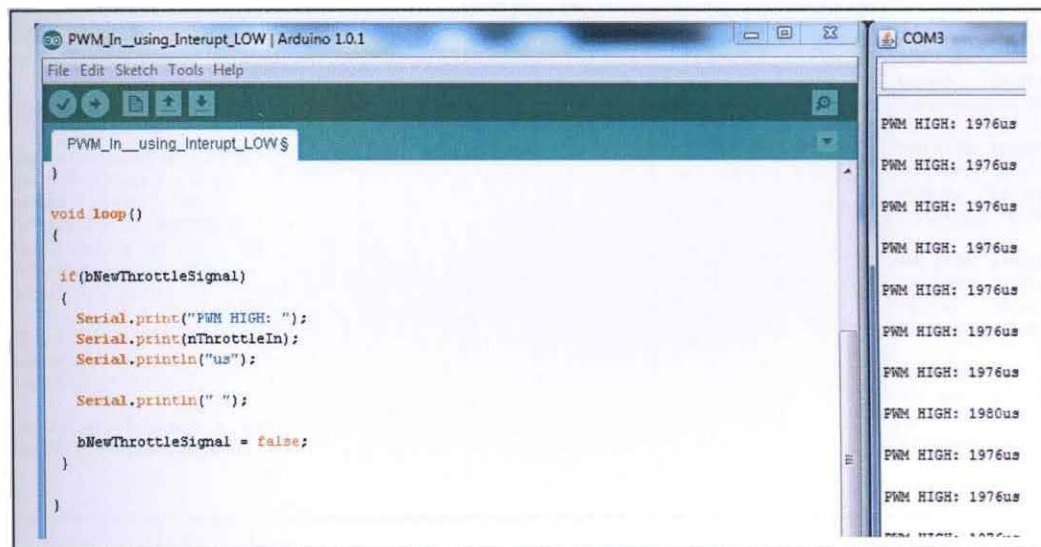
    bNewThrottleSignal = false;
  }
}

void calcInput()
{
  if(digitalRead(THRITTLE_SIGNAL_IN_PIN) == HIGH)
```

COM3

PWM HIGH: 988us  
PWM HIGH: 988us  
PWM HIGH: 992us  
PWM HIGH: 992us  
PWM HIGH: 992us  
PWM HIGH: 988us  
PWM HIGH: 992us  
PWM HIGH: 992us  
PWM HIGH: 988us  
PWM HIGH: 992us  
PWM HIGH: 992us  
PWM HIGH: 992us  
PWM HIGH: 992us

Figure 4.11: Lowest RC



```
PWM_In_using_Interupt_LOW$
}

void loop()
{
  if(bNewThrottleSignal)
  {
    Serial.print("PWM HIGH: ");
    Serial.print(nThrottleIn);
    Serial.println("us");

    Serial.println(" ");

    bNewThrottleSignal = false;
  }
}

void calcInput()
{
  if(digitalRead(THRITTLE_SIGNAL_IN_PIN) == HIGH)
```

COM3

PWM HIGH: 1976us  
PWM HIGH: 1976us  
PWM HIGH: 1976us  
PWM HIGH: 1976us  
PWM HIGH: 1976us  
PWM HIGH: 1976us  
PWM HIGH: 1976us  
PWM HIGH: 1980us  
PWM HIGH: 1976us  
PWM HIGH: 1976us  
PWM HIGH: 1976us

Figure 4.12: Highest RC

Figure 4.11 and Figure 4.12 show that the data read from the RC signal generator by using pulseIn() function. It shows that the lowest ON time is 992us and the highest ON time is 1980us. By using formula  $T = \frac{1}{f}$ , it shown that the lowest duty cycle is 4.96% and the highest duty cycle is 9.88%.



Figure 4.13: Lowest RC Reading

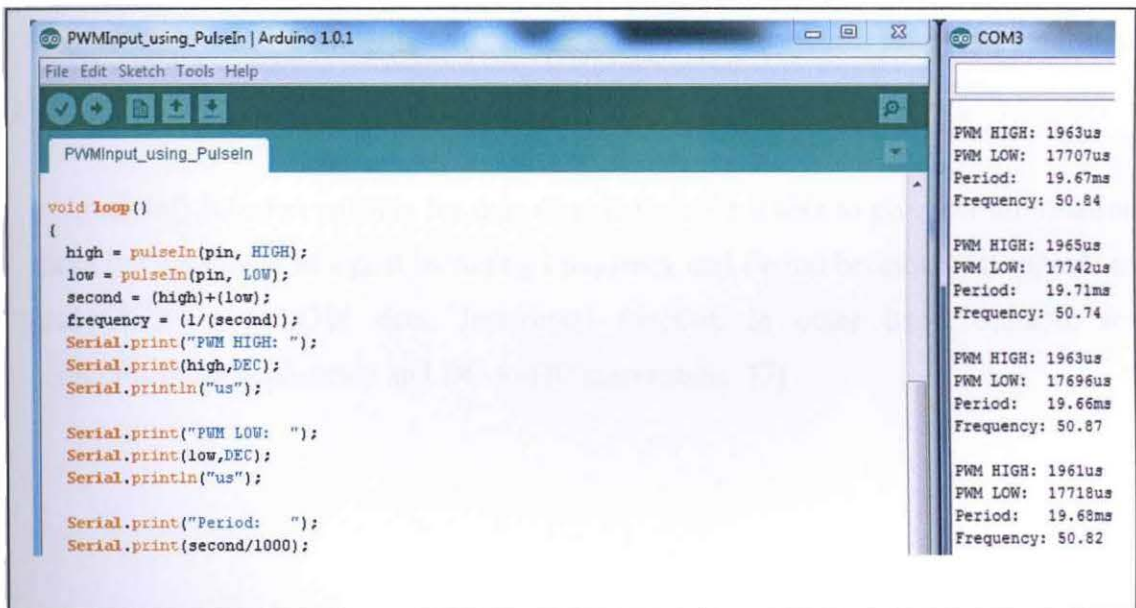


Figure 4.14: Highest RC Reading

Figure 4.13 and Figure 4.14 show that the data read from the RC signal generator by using `attachInterrupt()` function. It shows that the lowest ON time is 984us and the highest ON time is 1961us. It shows that the lowest duty cycle is 4.74% and the highest duty cycle is 9.815%.

#### 4.2.1 Data comparison and discussion

The output result by referring to the Figure 4.11, Figure 4.12, Figure 4.13 and Figure 4.14 show that the highest and lowest ON time value between Interrupt function and pulseIn function has a small different around 10us.

The different between interrupt() function and pulseIn() function is, the Interrupt() function only able to read 1 pulse condition which is HIGH pulse (depend on the programming) but pulseIn() able to read HIGH pulse and LOW pulse. These also show that Interrupt() function using less operation then pulseIn() function.

Arduino such as UNO is able to operate around 16 million operations per second. However, in 2ms Arduino will use around 32,000 operations. If it read whole 20ms by reading HIGH and LOW pulse, it will take around 320,000 operations. There are lots of operations wasted and it is not very efficient if the interested part is just only for HIGH pulse.[8]

pulseIn() function suitable for data display because it able to give full information about the whole PWM signal including Frequency and Period because of it capable to read HIGH and LOW data. Interrupt() function in other hand, suitable for communication application and DC-to-DC conversion. [7]

### 4.3 Motor speed control

The easiest way to produce PWM in Arduino is by using `analogWrite()` function. Therefore, this application became the main program to produce PWM signal for Motor Driver. Figure 4.15, Figure 4.16, and Figure 4.17 show that the PWM signal for Motor Driver.

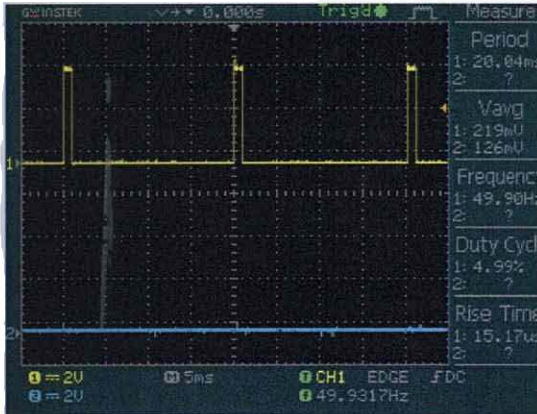


Figure 4.15: Stall Mode

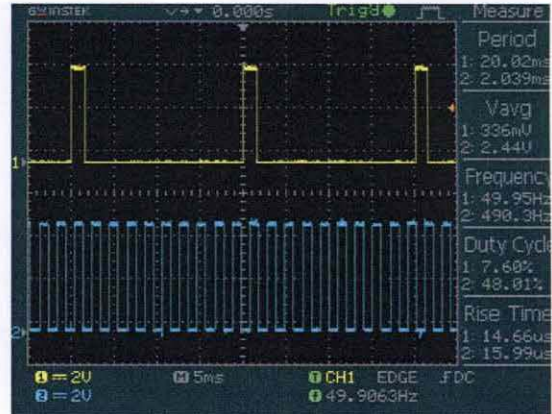


Figure 4.16: Slow Mode

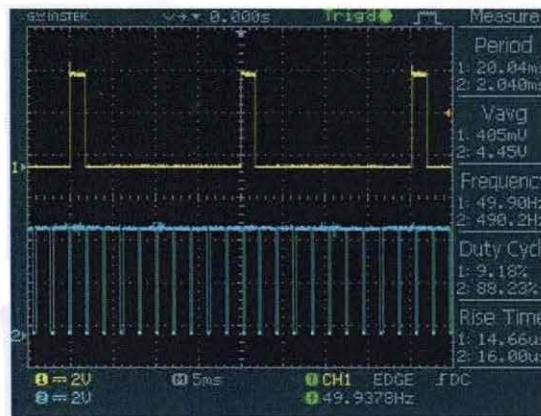


Figure 4.17: Fast Mode

From the Figure 4.17, it show that the duty cycle for PWM motor driver signal is increasing with an increase of duty cycle from RC signal. When the motor is in stall mode or in inverse mode, the RC duty cycle is as low as 4.99%.

When the motor is move in slow mode, the duty cycle for RC is 9.18% which the duty cycle for PWM motor driver is 88.23%. Given the period of the PWM motor driver is 2ms, it show that it HIGH pulse is 0.9602ms.

When the motor is move in fast mode, the duty cycle for RC is 7.60% which the duty cycle for PWM motor driver is 48.01%. Given the Period of the PWM motor driver is 2ms, it show that it HIGH pulse is 1.7646ms.

The result also shown that the frequency for the PWM motor driver already been fixed by 490Hz. This frequency is fixed by analogWrite() function program in Arduino. However, the frequency is depends on the timer bit when the data is convert.

Mode	Duty cycle motor	RC duty cycle signal	Frequency PWM motor	Frequency PWM RC
Fast Mode	88.23%	9.18%	490Hz	50Hz
Slow Mode	48.1%	7.6%		
Stall mode	0%	4.99%		

Table 4.1: Acquisition Data

#### 4.4 RC to Motor Speed Control Conversion

The technique to converting the RC signal to become PWM signal for motor driver is by combining Arduino IDE for data acquisition and PWM generator by using analogWrite() function. APPENDIX A and APPENDIX B show the full program for pulseIn() function and interrupt() function. Figure 4.4.1 and Figure 4.4.2 show the main structure how the data acquisition generating PWM.

Figure: 4.41: Data acquisition program provide value for PWM generator program

```

duration = pulseIn(PWM_in, HIGH);

if ((duration >=980)&&(duration <1470))
{
sensorValue = 0;
Serial.println("0% PWM");
digitalWrite(CCW, LOW);
digitalWrite(CW, LOW);;
}
if ((duration >=1471)&&(duration <1570))
{
sensorValue = 500;
Serial.println("50% PWM");
digitalWrite(CCW, LOW);
digitalWrite(CW, LOW);
}

```

Signal Reading Program using pulseIn

```

void digital_output()
{
if (( nThrottleIn >= 1800) && ( nThrottleIn<1999))
{
value = 1000;

Serial.println("Really Fast");

digitalWrite(CW,HIGH);

digitalWrite(CCW,LOW);
}
}

```

Signal Reading Program using Interrupt

Figure 4.4.2 : Data PWM generator

```

void loop()
{
outputValue = map(sensorValue, 0, 1023, 0, 255);

analogWrite(OutPin, outputValue);

digital_input();
}

```

PWM Signal Program (pulseIn)

```

void loop()
{outvalue = map(value, 0, 1023, 0, 255);

analogWrite(outpin,outvalue);

digital_output();

if(bNewThrottleSignal)
{ bNewThrottleSignal = false;
}
}

```

PWM Signal Program (Interrupt)

Referring to the pulseIn example, in the signal reading program, the value of HIGH pulse is determine by using pulseIn() function. In here, the HIGH pulse is declared as “duration”. When the “duration” is 980, us refer as the value for HIGH pulse is 980us.

PWM signal program use 10-bit signal and converting it to 8-bit signal. The 1024 is divided with 255 interval. The 10-bit signal is declared as “sensorValue” and 8-bit signal is declared as “outputValue”. As been explain in chapter 3, analogWrite(pin,PWM) is the reason why the PWM can be produce.

In this program, when the reading signal provides 980us to 1470us, the “sensorValue” is program as 0. “outputValue = map(sensorValue, 0, 1023, 0, 255);” will convert “0” value from 10-bit(sensorValue) to “0” value of 8-bit(outputValue).

When the reading signal provides 1471us to 1570us, the “sensorValue” is program as 0. “outputValue = map(sensorValue, 0, 1023, 0, 255);” will convert “500” value from 10-bit(sensorValue) to “128” value of 8-bit(outputValue).

Mode	Data acquisition value	
	Interrupt()	pulseIn()
Stall/ Stop	Less than1599us	980us-1470us
Slow	1600us-1799	1471us-1570us
Fast	1800us-1999us	1571us-1770us

Table 4.2: Data acqisation value use to set the PWM mode

#### 4.5 Implement of PWM generator on AGV

From the experiment, it seems that the permanent magnet DC brushed motor required voltage around 10V with current less than 10A to accomplish the experiment requirement. Therefore, the motor driver that has been selected is SHIELD-MD10.

The result shows that the motor speed can be control using a signal from RC. This experiment is also applied open loop actual system.



## **CHAPTER 5**

### **CONCLUSION**

#### **5.1 Conclusion**

From this project, it is clear that by using 16-bit microcontroller such as Arduino, capable to synthesize Drive System to control Permanent Magnet DC motor speed with integrated Remote Controller signal. This find shows that the Interrupt() function is more efficient to read a Digital Signal than pulseIn() function. pulseIn() is more suitable to use as data display.

To built a drive system that capable to carry a high load, MD30B is more suitable as the motor driver. For application that required only 10A current, SHEILD-MD10 is more appropriate. To control 4 motor in 1 operation the microcontroller with more than 4 timers. Arduino MEGA 2560 has about 6 internal timers allowing it to produce more PWM signal at one time.

#### **5.2 Recommendation**

There is lots of improvement can be done for this project. But it is better to focus on implement this system in close loop.

### 5.3 Reference

1. Arduino UNO Manual
2. Arduino MEGA 2560 Manual
3. L293D Manual
4. SHEILD-MD10 Manual
5. M30B Manual
6. [www.cytron.com.my](http://www.cytron.com.my)
7. [www.arduino.cc/tutorial/SerialInputout/](http://www.arduino.cc/tutorial/SerialInputout/)
8. <http://rcarduino.blogspot.com>
9. IC Timer Cookbook; 2<sup>nd</sup> Ed; Walter G Jung; Sams Publishing; 384 pages;1983
10. Engineer's Mini-Notebook – 555 timer IC circuit; Forrest M Mims III; Radio Shack; 32 pages;1989
11. Electrical Machinery and Power System Fundamentals; Stephan J. Chapman; MacGrawHill 355 pages;2002

## APPENDIX A

Motor Speed Control with Interrupt() function.

```
#define THROTTLE_SIGNAL_IN 0

#define THROTTLE_SIGNAL_IN_PIN 2

#define NEUTRAL_THROTTLE 1500

volatile int nThrottleIn = NEUTRAL_THROTTLE;

volatile unsigned long ulStartPeriod = 0;

volatile boolean bNewThrottleSignal = false;

int value = 0;

int outvalue = 0;

int outpin = 5;

int CW = 3;

int CCW = 4;

void digital_output()

{

  if(( nThrottleIn >= 1800) && ( nThrottleIn<1999))

  {

    value = 1000;

    Serial.println("Really Fast");

    digitalWrite(CW,HIGH);

    digitalWrite(CCW,LOW);

  }

  if((nThrottleIn >= 1600) && ( nThrottleIn <1799))

  {

    value = 900;

    Serial.println("Fast");

    digitalWrite(CW,HIGH);

    digitalWrite(CCW,LOW);

  }

}
```

```
}  
  
if(nThrottleIn < 1599)  
{  
    value = 0;  
    Serial.println("Break");  
}  
}  
  
void setup()  
{  
    pinMode(CW, OUTPUT);  
    pinMode(CCW, OUTPUT);  
    attachInterrupt(THROTTLE_SIGNAL_IN,calcInput,CHANGE);  
    Serial.begin(9600);  
}  
  
void loop()  
{  
    outvalue = map(value, 0, 1023, 0, 255);  
    analogWrite(outpin,outvalue);  
    digital_output();  
    if(bNewThrottleSignal)  
    {  
        Serial.println(nThrottleIn);  
        bNewThrottleSignal = false;  
    }  
}  
  
void calcInput()  
{  
    if(digitalRead(THROTTLE_SIGNAL_IN_PIN) == HIGH)
```

```
{
    ulStartPeriod = micros();
}
else
{
    if(ulStartPeriod && (bNewThrottleSignal == false))
    {
        nThrottleIn = (int)(micros() - ulStartPeriod);
        ulStartPeriod = 0;
        bNewThrottleSignal = true;
    }
}
}
```

## APPENDIX B

Motor Speed Control Using pulseIn() function

```
const int OutPin = 5;

int CCW = 4;

int CW = 3;

int sensorValue = 0;

int outputValue = 0;

int PWM_in = 2;

int duration;

void digital_input()

{

    duration = pulseIn(PWM_in, HIGH);

    Serial.print("Duration:");

    Serial.println(duration);

    if ((duration >=980)&&(duration <1470))

    {

        sensorValue = 0;

        Serial.println("0% PWM");

        digitalWrite(CCW, LOW);

        digitalWrite(CW, LOW);

    }

    if ((duration >=1471)&&(duration <1570))

    {

        sensorValue = 500;

        Serial.println("50% PWM");

        digitalWrite(CCW, LOW);

        digitalWrite(CW, LOW);

    }

}
```

```

}
if ((duration >=1571)&&(duration <1770))
{
  sensorValue = 700;
  Serial.println("70% PWM");
  digitalWrite(CCW, LOW);
  digitalWrite(CW, HIGH);
}
if ((duration >=1771)&&(duration <1970))
{
  sensorValue = 900;
  Serial.println("90% PWM");
  digitalWrite(CCW, LOW);
  digitalWrite(CW, HIGH);
}
}
void setup()
{
  pinMode(CCW, OUTPUT);
  pinMode(CW, OUTPUT);
  Serial.begin(9600);
}
void loop()
{
  outputValue = map(sensorValue, 0, 1023, 0, 255);
  analogWrite(OutPin, outputValue);
  digital_input() }

```

## APPENDIX C

### READING RC Signal using Interrupt

```
#define THROTTLE_SIGNAL_IN 0 // INTERRUPT 0 = DIGITAL PIN 2 - use the interrupt number in
attachInterrupt

#define THROTTLE_SIGNAL_IN_PIN 2 // INTERRUPT 0 = DIGITAL PIN 2 - use the PIN number in
digitalRead

#define NEUTRAL_THROTTLE 1500 // this is the duration in microseconds of neutral throttle on an electric
RC Car

volatile int nThrottleIn = NEUTRAL_THROTTLE; // volatile, we set this in the Interrupt and read it in loop so
it must be declared volatile

volatile unsigned long ulStartPeriod = 0; // set in the interrupt

volatile boolean bNewThrottleSignal = false; // set in the interrupt and read in the loop

void setup()
{
    attachInterrupt(THROTTLE_SIGNAL_IN,calcInput,CHANGE);

    Serial.begin(9600);
}

void loop()
{
    if(bNewThrottleSignal)
    {
        Serial.print("PWM HIGH: ");

        Serial.print(nThrottleIn);

        Serial.println("us");

        Serial.println(" ");

        bNewThrottleSignal = false;
    }
}

void calcInput()
{
    if(digitalRead(THROTTLE_SIGNAL_IN_PIN) == HIGH)
    {
```



```
    ulStartPeriod = micros();  
}  
else  
{  
    if(ulStartPeriod && (bNewThrottleSignal == false))  
    {  
        nThrottleIn = (int)(micros() - ulStartPeriod);  
        ulStartPeriod = 0;  
        bNewThrottleSignal = true;  
    }  
}  
}
```

## APPENDIX D

### READING RC Signal using PulseIn()

```
int pin = 2;

unsigned long high;
unsigned long low;
double second;
double frequency;

void setup()
{
    pinMode(pin, INPUT);
    Serial.begin(9600);
}

void loop()
{
    high = pulseIn(pin, HIGH);
    low = pulseIn(pin, LOW);
    second = (high)+(low);
    frequency = (1/(second));
    Serial.print("PWM HIGH: ");
    Serial.print(high,DEC);
    Serial.println("us");
    Serial.print("PWM LOW: ");
    Serial.print(low,DEC);
    Serial.println("us");
    Serial.print("Period: ");
    Serial.print(second/1000);
    Serial.println("ms");
```

```
Serial.print("Frequency: ");  
Serial.println(1/(second/1000000));
```

```
Serial.println(" ");
```

```
delay(1000);
```

```
}
```