AUTONOMOUS WHEELED ROBOT WITH PATH PLANNING

BY

ONG KOK TONG

A REPORT

SUBMITTED TO

University Tunku Abdul Rahman

In partial fulfilment of the requirement

for the degree of

BACHELOR OF COMPUTER SCIENCE (HONS)

Faculty of Information and Communication Technology

(Perak Campus)

MAY 2013

DECLARATION OF ORIGINALITY

I declare that this report entitled "**AUTONOMOUS WHEELED ROBOT WITH PATH PLANNING**" is my own work except as cited in the references. The report has not been accepted for any degree and is not being submitted concurrently in candidature for any degree or other award.

Signature : ________________________

Name : ________________________

Date : ________________________

ACKNOWLEDGEMENTS

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individual. I would like to extend my sincere thanks to all of them.

Firstly i would like to thank everyone who had contributed to the successful completion of this project. I would like to express my gratitude to my research supervisor, Mr. Liew Shiuh Deh for his invaluable advice, guidance and his enormous patience throughout the development of the research. In addition, I would also like to express my gratitude to my loving parent and friends who had helped and given me encouragement.

Finally, I take this chance to acknowledge the services of all UTAR staffs, lecturers, officers and everyone who collaborated in bringing this project to a successful end.

ABSTRACT

The purpose of this project is to design and construct a tunnel inspection robot which capable to moving around unfamiliar area with video feed. It can replace human high risk inspection job and used as a surveillance robot under insecure area such as radioactive zone or collapse structure. It able provides clear video feed and move in dark area with attached LED on the camera. This tunnel inspection robot is made up of a moving platform controlled by 2 dc motors. A mini camera is installed on the top of the robot and it is turntable by activating the servo to have multi-angle view. The whole platform is controlled by wireless system to avoid wire entanglements. The robot has auto and manual mode. In auto mode, the robotic car capable to travel around and avoid obstacles by the feedback from sensors. In manual mode user must use radio controller to control the movement of the robot.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS / ABBREVIATIONS

LED	Light Emitted Diode
IR	Infrared
US	Ultrasonic
MAEP	Mobile Arduino Experimental Platform
AI	Artificial Intelligent
EMI	Electromagnetic Interference
IDE	Integrated Development Enviroment
USB	Universal Serial Bus
PCB	printed circuit board

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

1.1 Background

In the 21st century, robot is widely developed in various of industries. Most of the sectors implement robotic systems in carrying out technical processes. Robotic systems have come a long way since their invention, and are getting more advanced. They can perform flawless work in short period of time and contributes in various factors such as quality, time, safety and much more.

Robots that encounter most frequently are robots that do work that are too dangerous or non-consistent job for human being. Most of them can be found in auto, manufacturing, medical and space industries. In fact, there are over a million of these types of robots working today in the world. For examples like the Mars Rover Sojourner and the upcoming Mars Exploration Rover, or the underwater man-size jellyfish robot Cyro helps human explore and collect environment data of the places that are too dangerous or impossible for human to discover.

This project proposes a wheeled robot with built in camera that would solve human problem in perform inspection under dangerous area or building. Tunnel inspection robotic car is a small designed remote control robotic car which can help tunnel inspector to perform better and convenient inspection job. There are many problems faced during inspection job such as narrow space which human are not capable to enter, damaged house which most of the structure are collapsed and radioactive area. It became a difficult and dangerous job for inspector to enter without having enough knowledge and background information of the area for inspection. The problems mentioned above might causes insecure situation for inspector while performing maintenance or testing work. Tunnel inspection robotic car is designed to replace human especially in high risk job. Besides, tunnel inspection robotic car is useful in determine the damaged structure whether is completely collapsed or partially collapsed instead of entering by human.

1.2 Problem Statements

Maintenance work in radioactive and collapsed structure area is high risk for human. The structure of damaged building might be unstable and might collapse anytime. It is not safe for inspectors perform inspection job and testing under those dangerous and hazardous areas. Moreover, without enough background information of the unfamiliar area might cause difficulties and insecure situation for inspectors. Narrow and small spaces of the tunnel cause inspector not capable to enter and perform maintenance job. Those problems might cause failure or inconsistent of data collect by inspector and low correctness of result.

1.3 Aims and Objectives

- To design and develop a robotic inspection car that can replace human in high risk and dangerous job.
    - The robotic inspection car is design in small size and easily control by inspector.

  - The video streaming from the robotic car must be clearly observed by the user or inspector especially in dark area.
- To develop a simple and user friendly interface
- To design a small size robot vehicle which able to enter through rough terrain and narrow spaces.
- To develop a robotic inspection car which capable move around and avoid obstacles automatically without human control and choose better path.

1.4 Project Scope

The scope of this project is to:

a. Design and program a tunnel inspection robotic car with Arduino board.
b. Design and develop an obstacle detector that can tell user the object is in front of it.
c. Mount a video camera on the top of the robotic car and transmit video.
d. Design and program the sensor attached on the robotic car which can detect the obstacle and avoid it by choosing another path.

1.5 Contribution

This project proposes and implement tunnel inspection robotic car which programmed to manoeuvre automatically without human control.

The lists below show the contributions of this project:

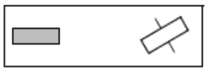
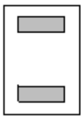
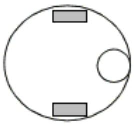
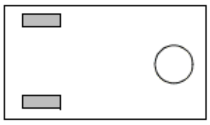
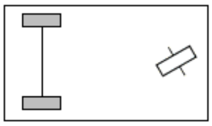
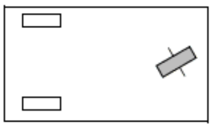
1. Robotic inspection car that can replace human in high risk and dangerous job such as performing inspection job in collapsed structure or radioactive area.
2. Robotic inspection car can replace human inspection job in certain area or places is narrow which human are not capable to enter. This vehicle is design in small size to enter rough terrain and narrow spaces.
3. Robotic inspection car is design to perform inspection job in unexplored structure or dark area.

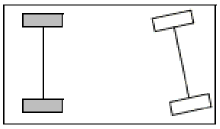
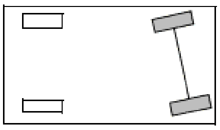
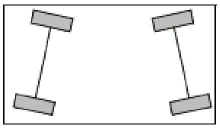
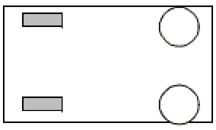
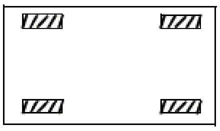
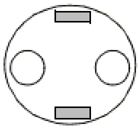
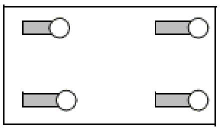
CHAPTER 2 LITERATURE REVIEW

2.1 Mechanical Design

Mobile robots define as robots which capable to move from place to place across the ground. Mobility provides a robot a much greater flexibility to perform new and complex task. With better mobility characteristic, robot can move anywhere needed with less power and energy consumption. Locomotion mechanism is needed which provide the capability for mobile robot to move unbounded throughout its environment. There are various ways to move from place to place which makes the selection of a robot's approach to locomotion an important aspect of mobile robot design. Many of these locomotion mechanism inspired by their biological counterparts which are adapted to different purpose and environments. For instance walk, crawl, hop, and slither.

Different of wheel configuration influence three fundamental characteristics: maneuverability, controllability and stability.

# of wheels	Arrangement	Description	Typical examples
2		One steering wheel in the front, one traction wheel in the rear	Bicycle, motorcycle
		Two-wheel differential drive with the center of mass (COM) below the axle	Cye personal robot
3		Two-wheel centered differen-tial drive with a third point of contact	Nomad Scout, smartRob EPFL
		Two independently driven wheels in the rear/front, 1 unpowered omnidirectional wheel in the front/rear	Many indoor robots, including the EPFL robots Pygmalion and Alice
		Two connected traction wheels (differential) in rear, 1 steered free wheel in front	Piaggio minitrucks
		Two free wheels in rear, 1 steered traction wheel in front	Neptune (Carnegie Mellon University), Hero-1

# of wheels	Arrangement	Description	Typical examples
4		Two motorized wheels in the rear, 2 steered wheels in the front; steering has to be different for the 2 wheels to avoid slipping/skidding.	Car with rear-wheel drive
		Two motorized and steered wheels in the front, 2 free wheels in the rear; steering has to be different for the 2 wheels to avoid slipping/skidding.	Car with front-wheel drive
		Four steered and motorized wheels	Four-wheel drive, four-wheel steering Hyperion (CMU)
		Two traction wheels (differential) in rear/front, 2 omnidirectional wheels in the front/rear	Charlie (DMT-EPFL)
		Four omnidirectional wheels	Carnegie Mellon Uranus
		Two-wheel differential drive with 2 additional points of contact	EPFL Khepera, Hyperbot Chip
		Four motorized and steered castor wheels	Nomad XR4000

Picture adapted from Roland Siegwarrt in introduction to Autonomous Mobile Robot © The MIT Press Cambridge, Massachusetts London, England

Table 2.1-1Wheel configuration of autonomous mobile robot

In this project, a low complexity and low power consumption wheeled robot will be the best choice to be developed. A four wheeled robot is preferable in order to moving on the uneven surface and better center of gravity. Each of two front bi-directional driven wheels spins in different direction makes it turn to right or left way while the rear two single direction driven wheels support forward or backward movement by spinning clockwise or anti-clockwise together.

2.2 Robotic Control

Robotic control can be explained as a combination of engineering and cognitive science. Perception algorithms process the sensor input meanwhile a program inside chipset decides the action should performed along with given goals and circumstances and command are sent to the motors to make the robot act. The process of robot control as shown below:

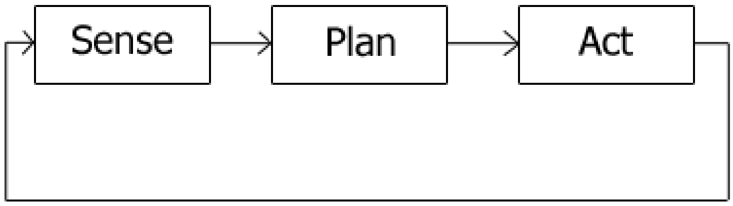
- Collects the information of the environment (Senses)
- Use and process the information (Process)
- Follow instructions to perform actions or works (Acts).

2.2.1 Hierarchical Paradigm

Hierarchical paradigm approach is mainly focus on the planning aspect of a robot’s behavioral cycle. This approach employs a top-down analysis of desired behavior of robot during the design phase and follows by implementation of sequence of modules. Firstly the robot senses its environment and collects the information then plans the next action based on these senses and follow by appropriate action taken by using available actuators attached. These robots are typically reflex agents which select specific action from rules based on the current ‘perceptions’ from sensory input.

After the sensor collect information from environments, the data will be passed to microcontroller to process those data.

The main characteristic of hierarchical paradigm is flow of control between the three primitive robotic functions defines in linear order is. Besides, explicitly representing knowledge about the environment is one of the characteristic of hierarchical paradigm. Lastly, the world model must contain everything that an agent needs to plan the sequence of action should be performed.



Picture adapted from Wikipedia.com

Figure 2.2-1Typical structure of hierarchical robot control system

2.2.2 Reactive Paradigm

Reactive robotic control system decomposes functionality into behaviors and become multiple instances of Sense-Act coupling. These coupling are run concurrently called behaviors. Behaviors implemented as low computational complexity algorithm (software) and behaviors are pure stimulus-response reflexes thus no memory required.

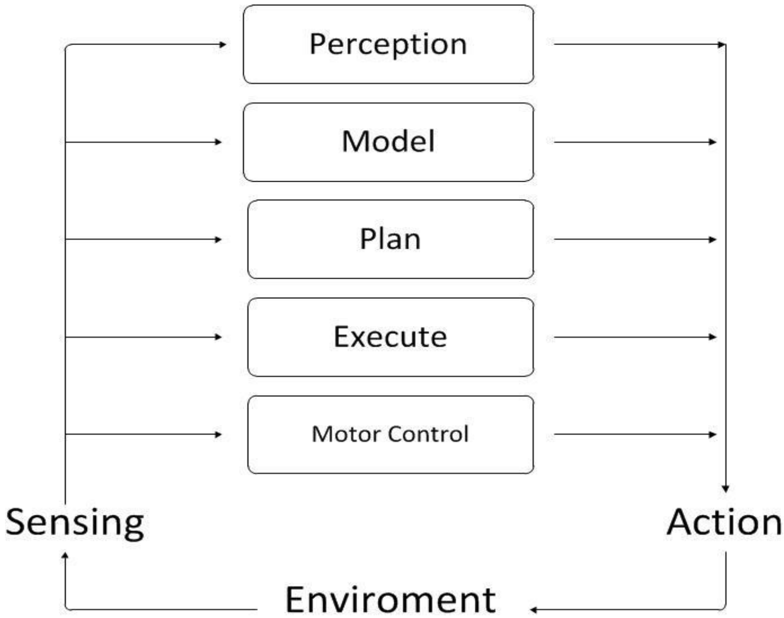


Figure 2.2-2Reactive paradigm as vertical decomposition

2.2.3 Subsumption Paradigm

The Subsumption architecture was developed Rodney Brooks in 1996. Behavior in this architecture is a network of sensing and acting modules which accomplish a task. Each module are grouped into layers of competence with lower layers capturing basic functions and more goal directed actions created by higher layers. Each additional layer builds upon and potentially relates with the existing input and output. Behavioral layers are concurrent process and only occur in either ways:

- If the output of higher layer is connected to the lower module which in “on” state then the output of lower module is stopped or blocked. (Inhibition)
- If the both higher and lower module is connected to input of another module and output of higher module is nonzero then the higher module will override the lower module

Task is accomplished by activation the appropriate layer. For instance lower layers will be run or process by reactivity to the environment.

2.3 Existing project studies

Project name	Protocol	Movement Control	Obstacle Detection	Live video feed
Wifi Robot [5]	Wifi	Manual (By VB program in PC)	No	Yes
Wifly Mini [4]	Wifi	Manual (By Android phone)	No	Yes
MAEP 2.0 [3]	No Wireless Module	Auto (Move by itself)	Yes	No

Table 2.3-1 Comparison of existing projects

Table 2.1 Summarize the existing homemade robotic car projects that use Arduino microcontroller platform in their project. Wifi Robot and Wifly Mini are wireless supported robot which can control by user within limited range while MAEP 2.0 only can move around itself without control by user. MAEP 2.0 has ultrasonic sensors which provide obstacle detection to avoid from collision. Wifi Robot and Wifly Mini capable to provide high quality live video streaming with camera attached on it.

Protocol: The way of communication between Arduino microcontroller and remote control to exchange message.

Movement Control: The Movement of the robot car can be control manually by remote control or move itself automatically without control by user.

Obstacles detection: Obstacles surrounded can be detected by sensor installed on the robot such as infrared or ultrasonic sensor. This can avoid collision of robot car.

Live video feed: Camera installed on robot can provide video feed to user in order to perform inspection in unfamiliar area or narrow tunnel.

2.3.1 Wi-Fi Robot

Wifi Robot is a remote control car that can be driven over the internet or with a laptop wirelessly from up to 500m away. It built in with a live-feed network camera so that it can be driven without line of sight. It consist of a remote control car, a Cisco Linksys router, Arduino (ATmega168) Freeduino MaxSerial, a customize power supply unit, a Panasonic BL-C1A web camera and a tiny horn. The remote control car is control by using the program of Visual Basic 6 language in PC via Wi-Fi and a C program runs inside the router which running OpenWRT WhiteRussian v0.9 (Linux). There are quite a numbers of large size accessories attach to the remote control car such as router and webcam which made it slightly bigger in size and not mobility. Besides, no LED light is attach to the car will produce low clarity of video streaming if it is moving in dark area.

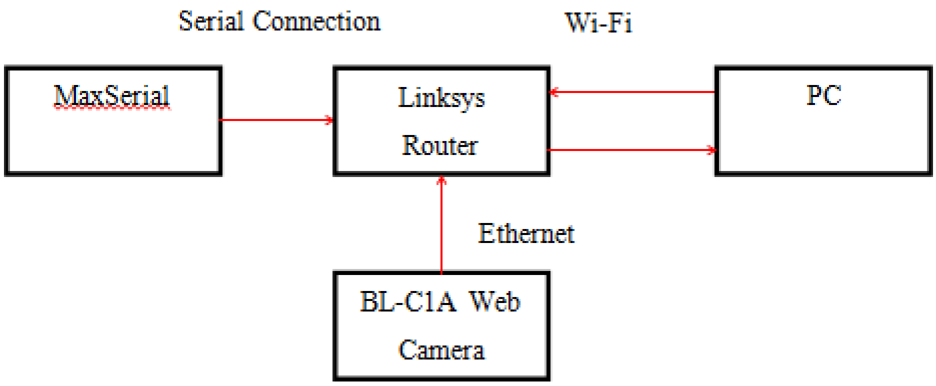
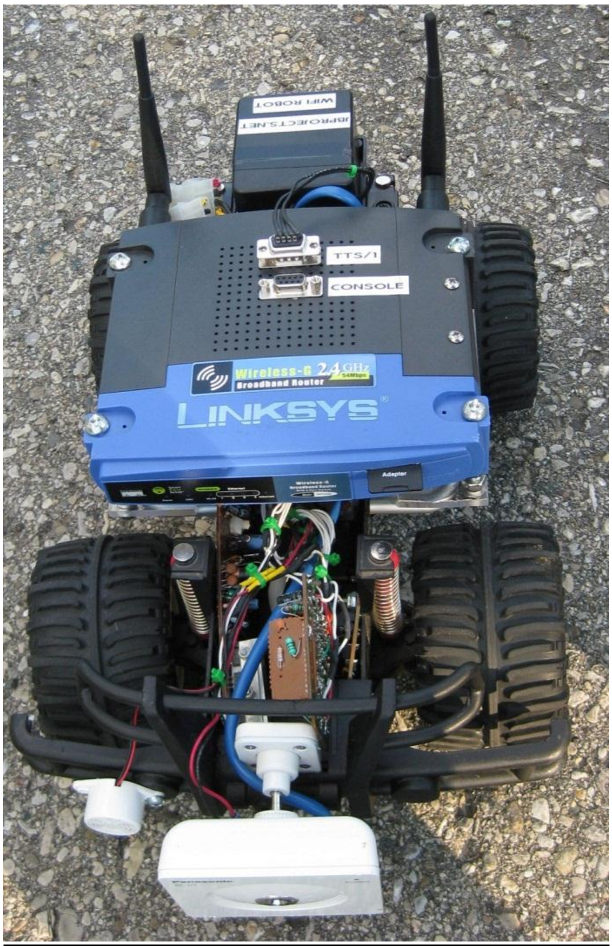


Figure 2.3-1 Flow chart of Wi-Fi Robot



Picture adapted from www.jbproject.com

Figure 2.3-2 Wifi Robot

2.3.2 Arduino Wifly Mini

Arduino Wifly Mini is control by using console game controller and it communicates over Wi-Fi. The Wifly module provides Arduino with a function to send and receive messages over a WiFi network. The Wifly module communicates with the WiFi access point which specified in programmed configuration. It mounted on top of an UNO board. Arduino Wifly Mini sends the video feed via a 1.3GHz 300mW video transmitter. The Arduino initiates a serial connection with the Wifly and these two devices communicate and exchange information.

The Arduino Wifly process the exchanged informations in UDP (User Datagram Protocol) packet form. Although UDP is not reliable as TCP but it response faster than TCP. A triple axis accelerometer is mounted at the front of the vehicle. This is use to detect the change in acceleration of robot speed.

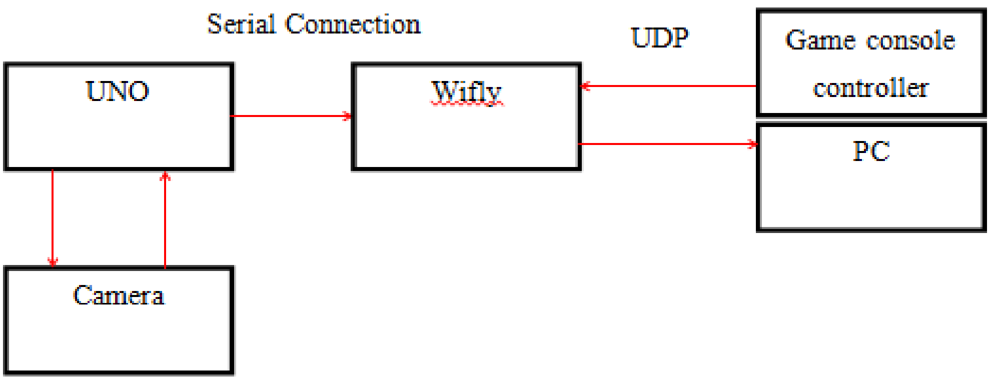


Figure 2.3-3Flow chart of Arduino Wifly Mini

Picture adapted from www.hackedgadget.com

Figure 2.3-4 Arduino Wifly Mini

2.3.3 Mobile Arduino Experimental Platform (MAEP) 2.0

Mobile Arduino Experimental Platform (MAEP) is a microcontroller robot which capable to avoid obstacle surrounded by detecting them with ultrasonic and infrared sensor built by a hobbyist with a nickname GeneralGeek. In this project, it choose the PING))) Ultrasonic Distance Sensor as the obstacle detection sensor while the infrared sensor is use for cliff detection. The PING))) Ultrasonic Distance Sensor mounted with the standard Parallax servo which provide rotation for obstacle detection. Arduino UNO microcontroller board is chosen in this project because it is easier to program than other microcontroller and lower cost. Besides, 2 parallax continuous servos is chosen for wheel spinning in order to move forward or left and right turning. For example, if the robot car wish to turn left, the left servo spin in clockwise direction while the right servo spin in anti-clockwise direction and vice versa.

Below listed the specification of the robotic car,

- Actuators / output devices: 2x parallax continuous rotation servos, 1x parallax standard servo
- CPU: Arduino Uno
- Power source: 4.8V 700 mAh NiCd, 9V alkaline battery
- Sensors / input devices: Sharp IR, PING))) sonar
- Target environment: indoors

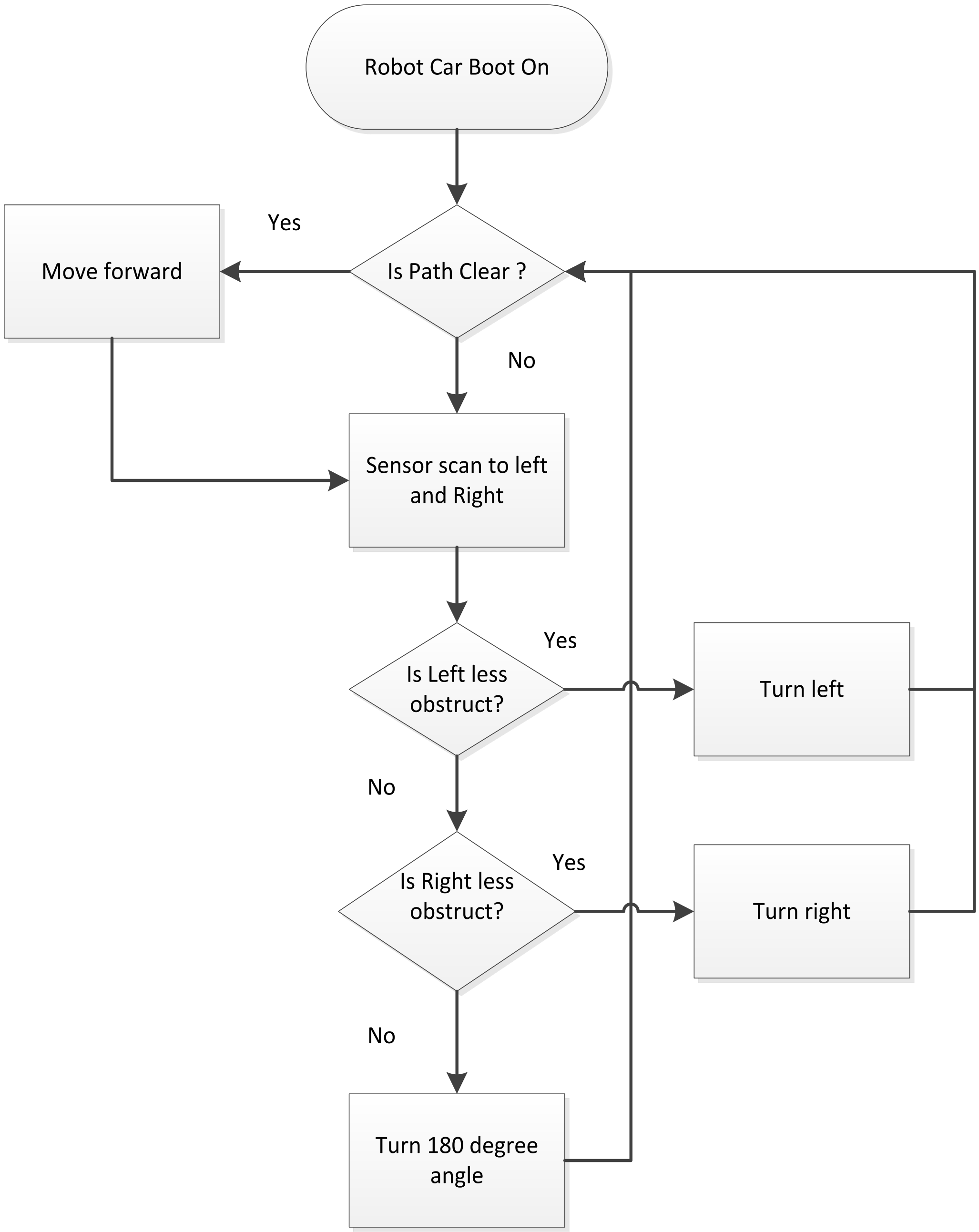
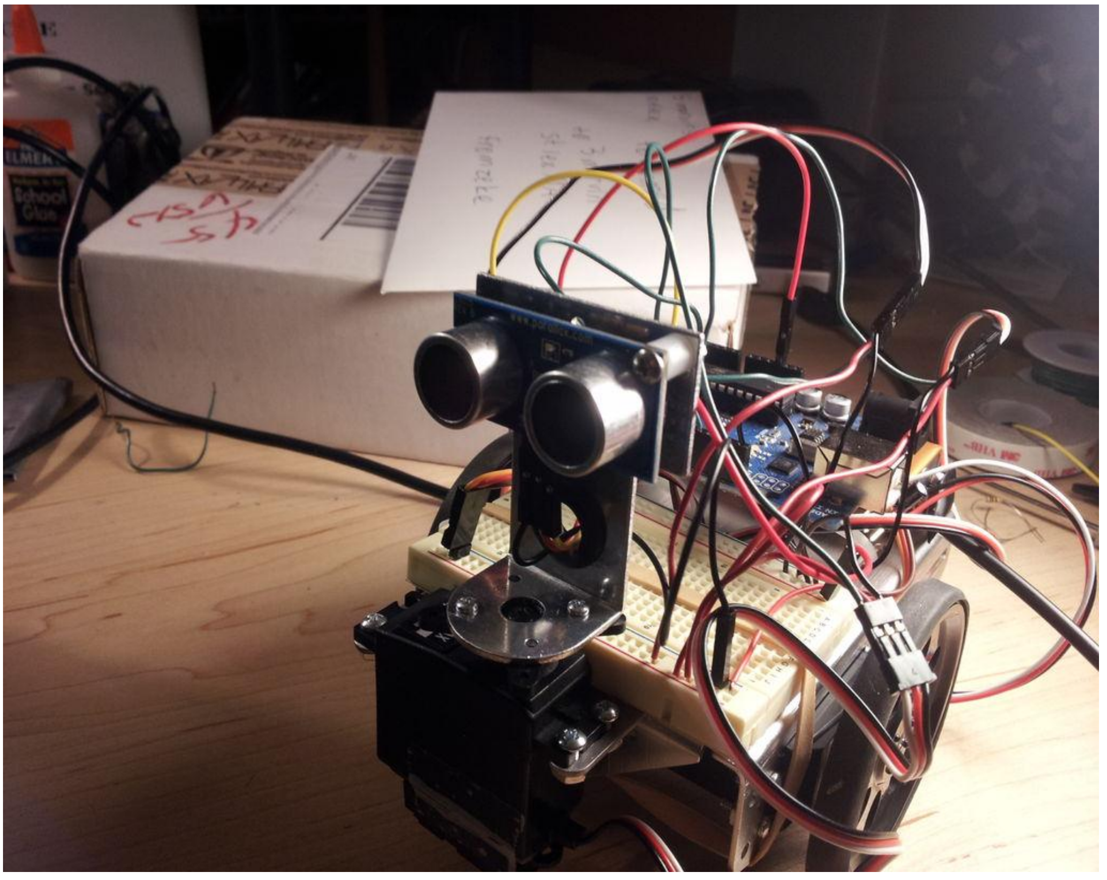


Figure 2.3-5Flow Chart of Mobile Arduino Experimental Platform (MAEP) 2.0



Picture adapted from www.instructable.com

Figure 2.3-6 Mobile Arduino Experimental Platform (MAEP) 2.0

CHAPTER 3 METHODOLOGY

3.1 Overview

In this project, several technologies are used in order implement the robotic car to increase its efficiency and mobility.

Research and study on:

- Obstruction sensor
- C program with AI component
- Arduino board programming technique
- Robot design concept
- Radio transmission protocol

3.2 Obstruction sensor

Sensors installed on the robotic car for detect the obstacle and objects in front. It is useful for users to choose another path when there is an obstacle or unstable terrain. There are 2 types of sensors can be found in the market which is suitable in this project.

3.2.1 Infrared sensors [1]

Infrared (IR) sensors are widely used as proximity sensors and for obstacle avoidance in robotics. Circuits inside IR sensor provide a binary output, and there are also others types of IR sensors which provide analogue output or multiple bit output. The sensors with binary output are good for detecting the proximity of an obstacle but not the range. IR sensor will estimate the range of obstacle within a certain distance. IR sensor uses the reflected light to estimate the distance of the surface. IR sensor only offer short range estimation and only applicable for short distances (< 25cm).

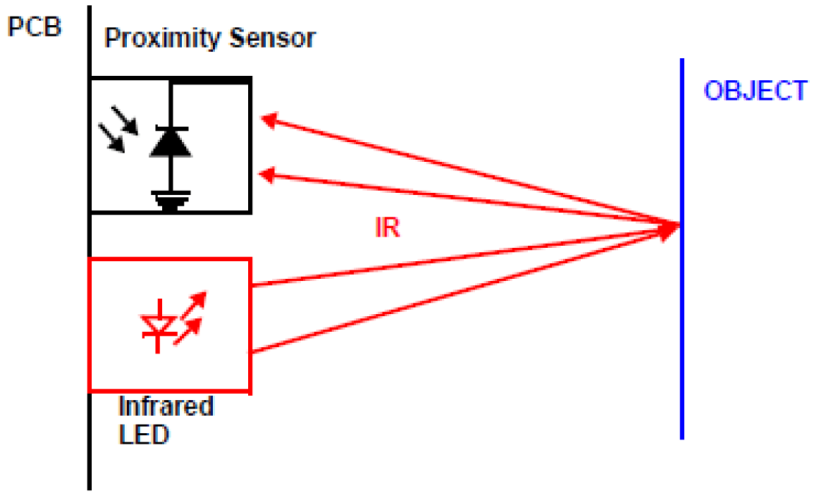
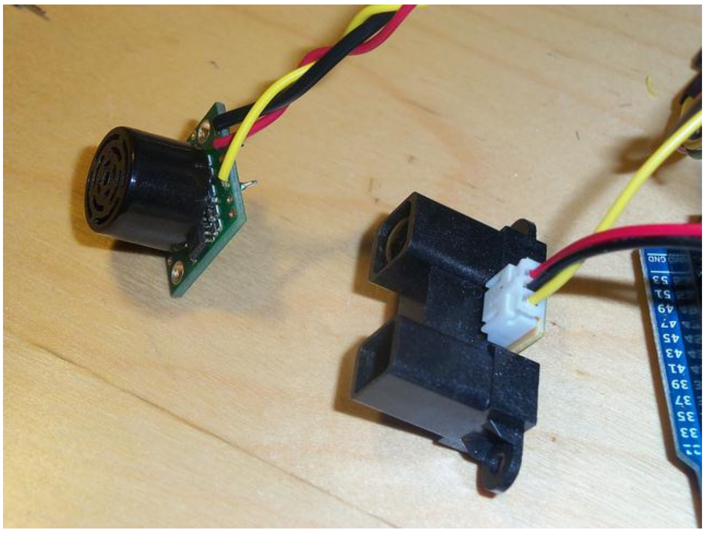


Figure 3.2-1Methodology of Infrared sensor

3.2.2 Ultrasonic sensors [1]

US sensors are widely used for proximity detection purpose. The concept of ultrasonic (US) sensor is to produce high frequency sound wave in cone shape by transducer and receive back after the reflected off the target. Sensor will calculate the time between sending and receiving echo to determine the accurate distance from the object or target. It can estimate with a precision of less than 1cm and up to 6m in distance measurement. US sensors capable to perform distance measurement on transparent, shiny and dark surface object. This ability allows ultrasonic sensors to detect material ranging from clear glass bottle to black rubber tyre. It can also works in tough environment such as noisy, dusty or dark area.



Picture adapted from www.flickr.com

Figure 3.2-2Ultrasonic and infrared sensors

Types of sensors	Infrared sensors (IR)	Ultrasonic sensors (US)
Method	Reflection of light emitted	Interpreting the echoes from radio or sound waves
Response time	Within 2ms	35ms
Angle of detection	Narrow	Wide
Range supported	Less than 25cm	Up to 6 metres
Cost	Low	High

Figure 3.2-3 Comparison of Infrared (IR) and Ultrasonic sensors (US)

Infrared sensor is cheaper and faster in response time compare with ultrasonic sensor. It has non-linear characteristic and they depend on the reflectance of the object surface. Ultrasonic sensor is resist to external disturbances such as vibration, infrared radiation, ambient noise, and EMI radiation. It is useful in poor lighting condition and transparent object. Besides, ultrasonic sensor is more reliable than infrared sensor and it is sensitive to specular surface but shorter range support. Infrared sensor is more accurate and faster in response time but it can only detect narrow width object.

3.3 Microcontroller

Arduino[2] is an open-source computing platform based on flexible design with combination of microcontroller and software development environments. It is built for designers, hobbyists to freely develop their ideas into a real objects and projects. The microcontroller is programmed by using the Arduino programming language and the Arduino development environment. Arduino projects can be stand-alone use in prototyping and projects. They also can communicate with software running on a computer such as Flash, Processing, MaxMSP. The Arduino board can be built or customize by ourselves or purchase preassembled in the market. There also much of projects samples and open source coding available online for reference or learning. Arduino is a microcontroller interface built with Atmel's ATMEGA8 and ATMEGA168 microcontrollers.

The Arduino integrated development environment (IDE) is a cross-platform application written in Java and most of the Arduino programs are written in C or C++. It can runs in all operating systems like Windows, Mac OS, and Linux. It is designed to introduce programming skills to artists and other newcomers which unfamiliar with software development or hardware programming. It includes a code editor with common developer software features like syntax highlighting, brace matching, and automatic indentation. It also provides a function that capable to compiling and uploading programs to the board via USB. It is a user-friendly interface for everyone from beginner to professional to learn or use. Users only need define two functions to make a runnable cyclic executive program:

- setup(): a function run once at the start of a program and initialize settings
- loop(): a function called repeatedly until the board powers off

Diagram 3.1.2-1 is an example of simple Arduino program written in C which turns on and off the LED repeatly for 1 second.

```
/*
  Blink
  Turns on an LED on for one second, then off for one second, repeatedly.

  This example code is in the public domain.
 */

// Pin 13 has an LED connected on most Arduino boards.
// give it a name:
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);               // wait for a second
  digitalWrite(led, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);               // wait for a second
}
```

Picture adapted from www.arduino.cc

Figure 3.3-1Sample Arduino program written in C [3]

The Arduino comes in a variety of different types and difficult to decide, but variety also allows for flexibility in choosing the perfect solution.

The differences of several versions of Arduino boards are listed as below. [2]

- Arduino Uno
- Arduino Nano
- Arduino Mega 2560
- Netduino
- Arduino Due (ARM-Based)

• Arduino board	Family	Clock	Digital	Analog	Operating Voltage
Uno	ATmega328	16 MHz	14	6	5V
Nano	ATmega168 or ATmega328	16 MHz	14	8	5V
Mega 2560	ATmega2560	16 MHz	54	16	5V
Netduino	ATmega328	16 MHz	14	6	5V
Due	AT91SAM3X8E	84MHz	54	12	3.3V

Table 3.3-1Comparison of Arduino boards [2]

Arduino Mega 2560 is the optimum choice and compatible to all devices in this project. It consists of 54 digital input/output pins, 16 analog inputs, a USB connection, 4 UARTs (hardware serial ports), an ICSP header, a 16 MHz crystal oscillator, a power jack, and a reset button. It has everything needed to support the microcontroller and just simply connect it to a computer via USB and install the Arduino IDE to get started. It also provide more PWM pins and I/O connections which provides more flexibility in project design.

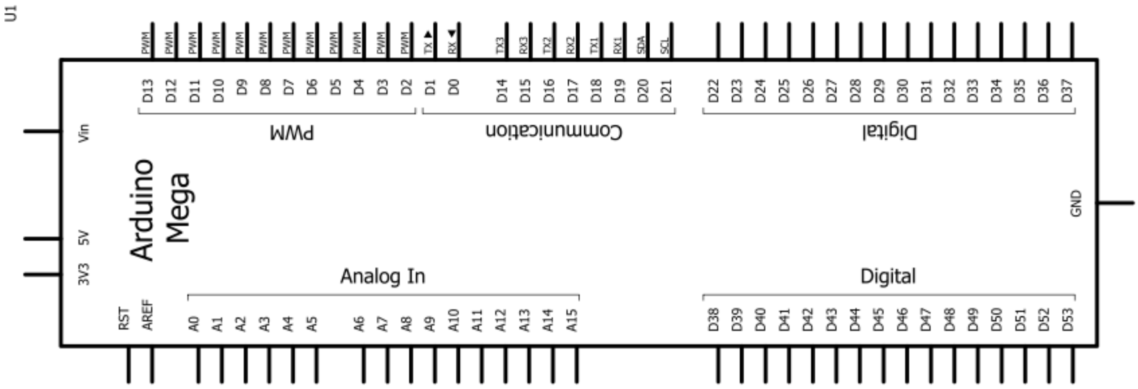


Figure 3.3-2Schematic design of Arduino Mega 2560

3.4 Ultrasonic Ranging Detector HC-SR04 Distance Sensor

Piezoelectric transducer inside ultrasonic sensor is used to produce and detect sound wave. High frequency of sound wave generated by transducer and evaluate the echo by receiver which the sound wave reflecting off the target surface. Distance from the sensor to target is determined by calculate the time interval between sending the signal and receiving the echo by sensor.

$$\text{Distance (cm)} = \frac{T \times Speed\ of\ sound}{2}$$

Speed of sound is approximately 341 m/s in air.

T = time between sound wave is emitted and received

It divided by 2 due to the sound wave travel to object and back to receiver.

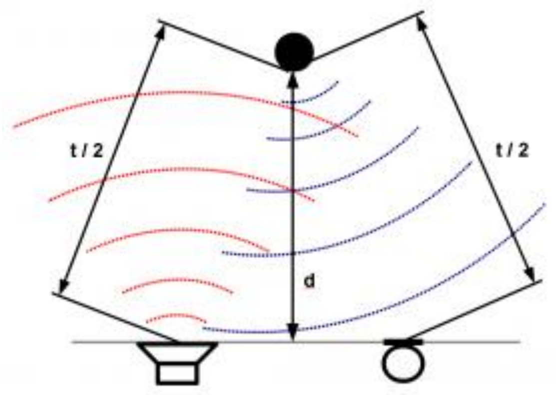
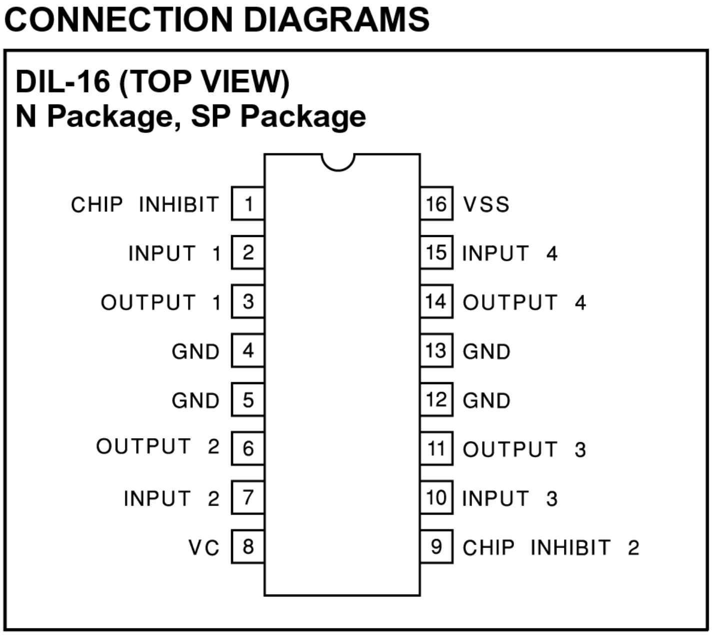


Figure 3.4-1 Sound wave transmitted and received

3.5 L293B Integrated Circuit Motor Driver (H-Bridge)

The L293 is an integrated circuit motor driver that can be used for simultaneous, bi-directional control of two small motors. The L293B is from ST microelectronic combines all the necessary connection for construct an H-bridge in one package.



ENABLE	DIRA	DIRB	Function
H	H	L	Turn right
H	L	H	Turn left
H	L/H	L/H	Fast stop
L	either	either	Slow stop

Adapted from L293B datasheet

Figure 3.5-1L293B and truth table

3.6 Robot Chassis

Picture adapted from hobbyking.com

Figure 3.6-1Aluminum 4WD Robot Chassis

3.7 Turnigy TGY-90S Servo

Servo is a small motor device that has an output shaft. As sending coded signal to the servo, this shaft can be positioned to specific angular positions. The angular position of the shaft can be change by manipulating the coded signal. Servo can be used in radio controlled cars, puppets, and also robot.

Picture adapted from hobbyking.com

Figure 3.7-1Turnigy TGY-90S Servo

The Pulse Coded Modulation is used to determine the angle of a servo motor to be turned to. The servo expects to see a pulse every 20 milliseconds (.02 seconds). The length of the pulse will determine how far the motor turns. A 1.5 millisecond pulse, for example, will make the motor turn to the 90 degree position (often called the neutral position). If the pulse is shorter than 1.5 millisecond, then the motor will turn the shaft to closer to 0 degrees. If the pulse is longer than 1.5ms, the shaft turns closer to 180 degrees as shown in diagram xxx below.

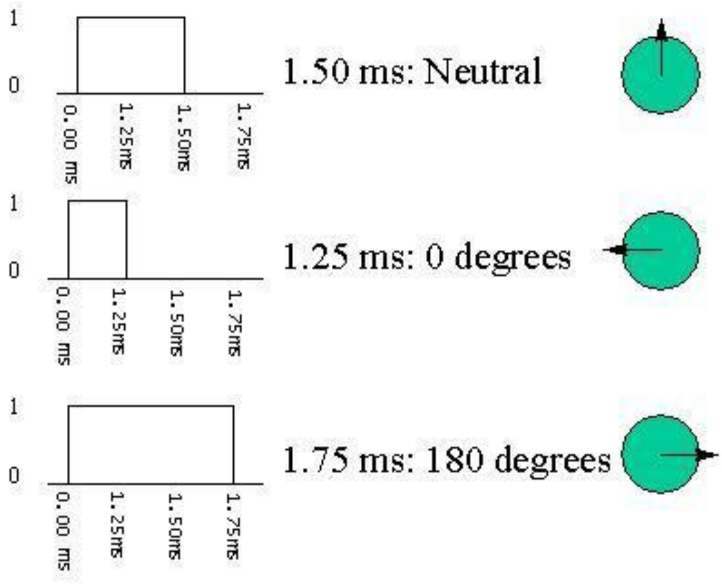
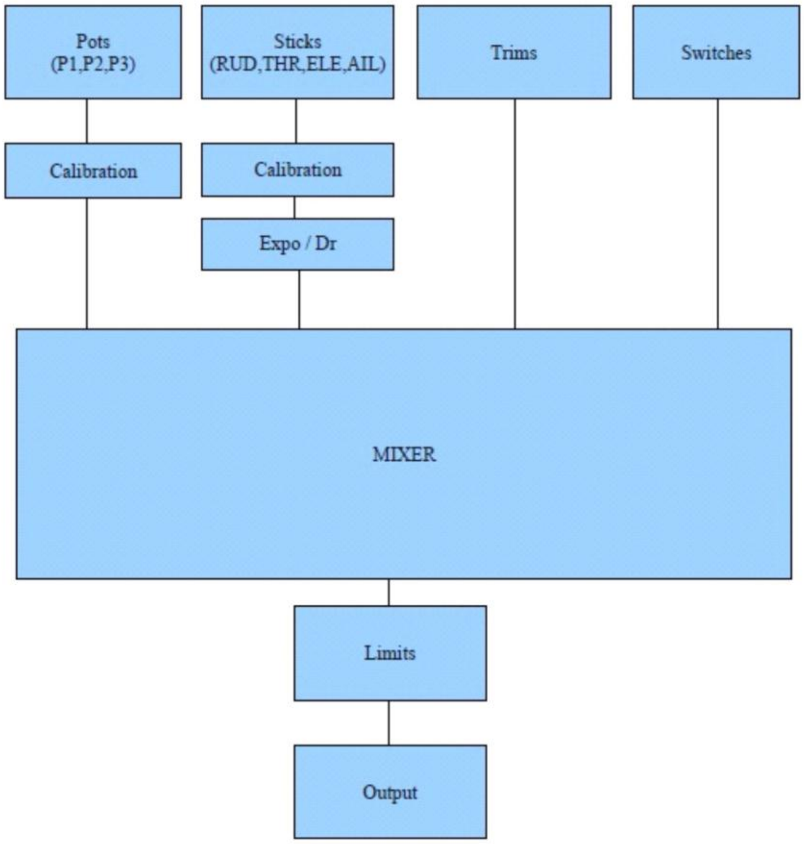


Figure 3.7-2 Pulse width modulation

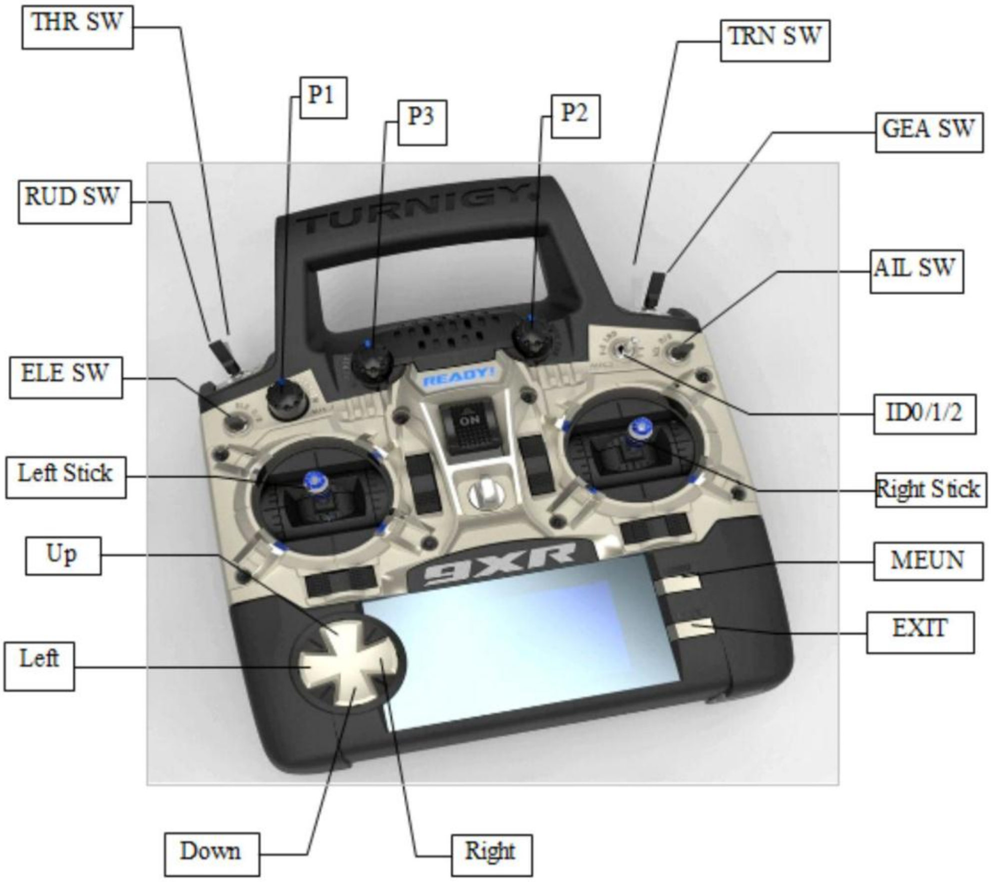
3.8 Turnigy 9XR transmitter

The 9XR transmitter is a computerized radio from Turnigy. It outfitted with a built in 128x64 pixel monochrome LCD, two axis gimbals, three variable potentiometers, six 2-position switches, one 3-position switch and Digital trims. The signal strength of this transmitter can support up to 1.5km.



Picture adapted from Turnigy9xr manual

Figure 3.8-1Flow chart of I/O processing of Turnigy 9XR



Picture adapted from Turnigy9xr manual

Figure 3.8-2 Layout of Turnigy 9XR

3.9 Communication

The Arduino microcontroller plays an important role in the whole system which handles the overall movement of the robotic car. The Arduino board is programmed in C language and act as a relay to control all the accessories installed on it. The live video feed will stream via the video transmitter on the robot to the receiver which installed with the small LCD monitor. The robotic car can run in either auto or manual mode. In auto mode, it will move around itself and avoid obstacles detected by sensors and choose another path while in manual mode user can control the robot movement by radio remote control. The camera on the robot can be turn in either horizontal or vertical way in order to have multi-angle view.

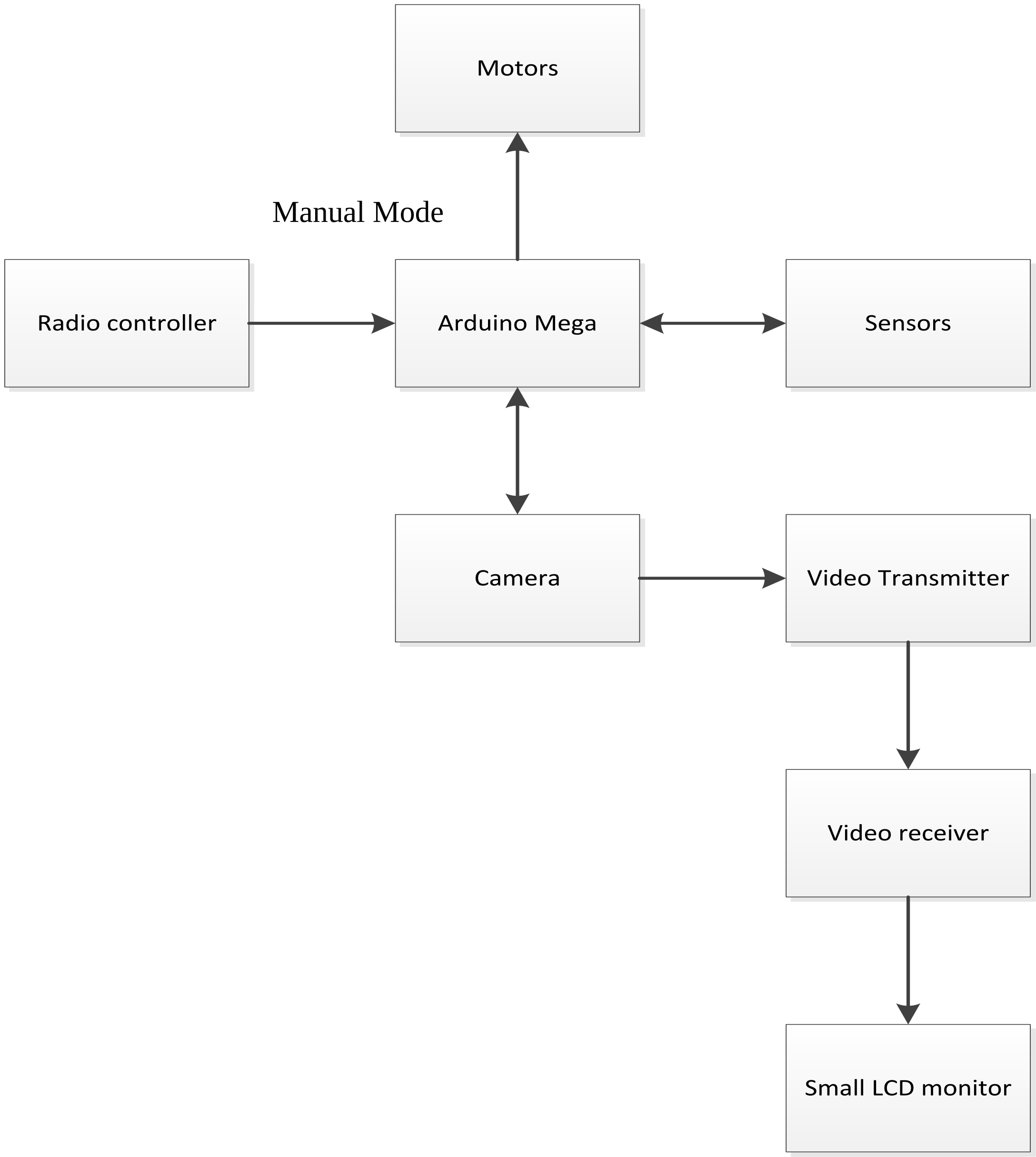


Figure 3.9-1Flow chart of Autonomous robot car with path planning

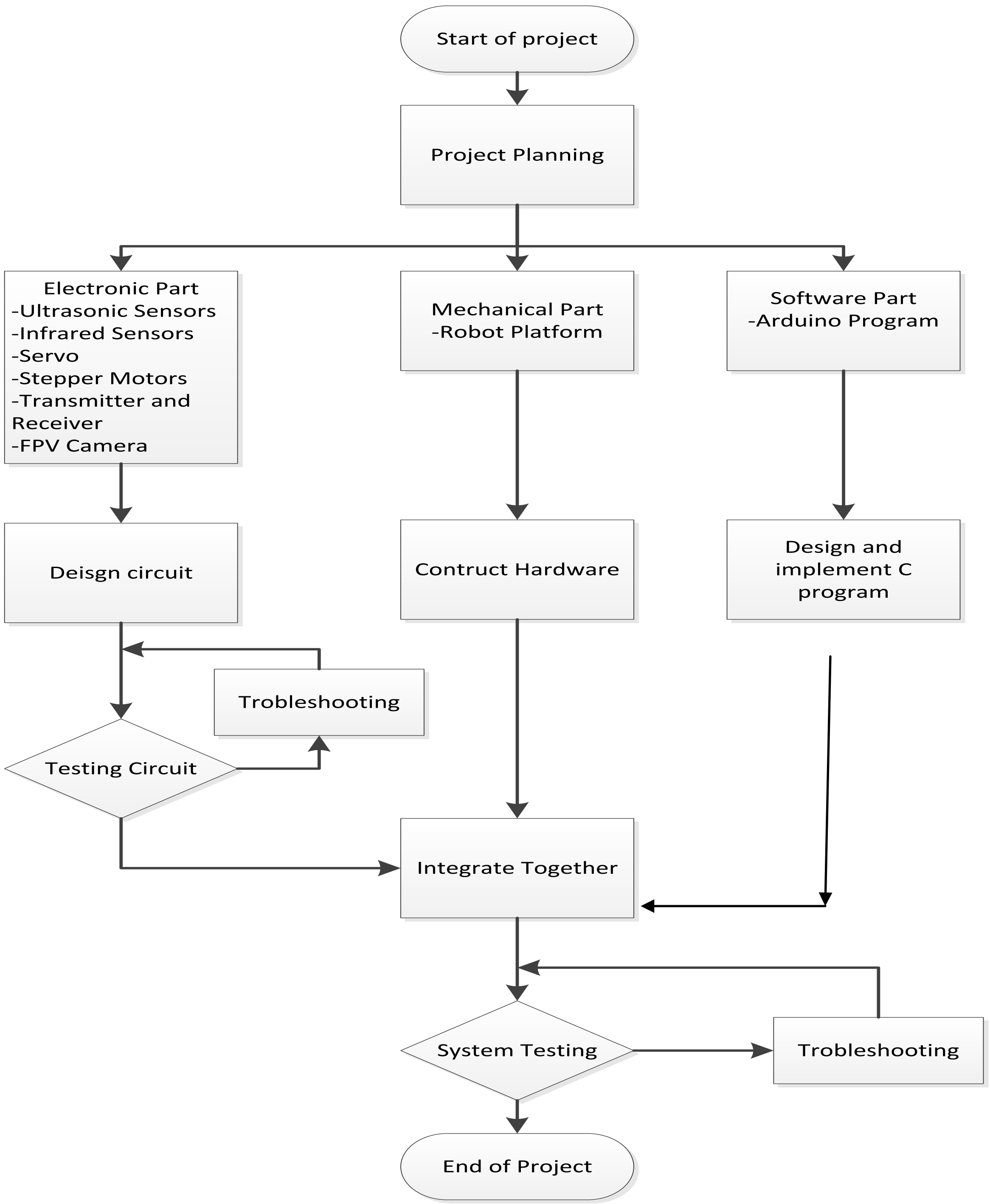


Figure 3.9-2 Flowchart of Project Methodology

CHAPTER 4 DEVELOPMENT ANALYSIS

4.1 Design of Autonomous Wheeled Robot

The autonomous robot (auto mode) has 3 statuses: move forward, stop and scan, and turn right or left. Since only 1 distance sensor used in this project, the sensor must have the capability to turn around to scan for alternative way to move so it must attach together with servo. The servo shaft can be turn in 180 degree to help the sensor scan the obstacle around and choose the clear path to move along. A homemade bracket used to attach the sensor to the servo.

In manual mode it can move forward or backward and also turn right or left by receiving signal from radio transmitter.

Picture adapted from http://www.yeggi.com/

Figure 4.1-1Distance sensor attached with servo motor by bracket

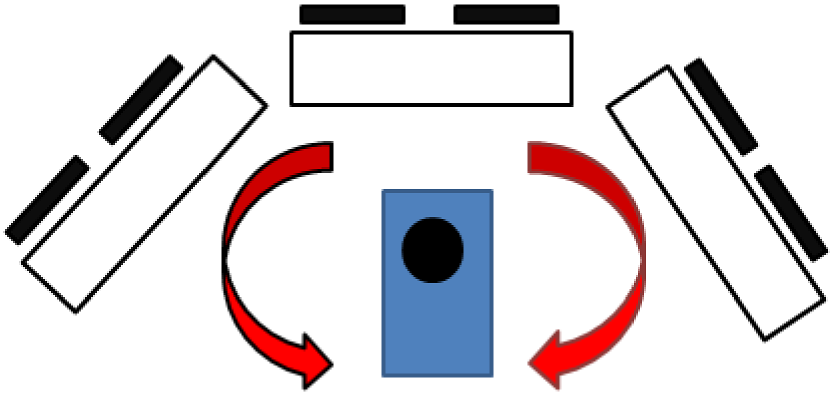


Figure 4.1-2 Turning function by servo motor to detect obstacles

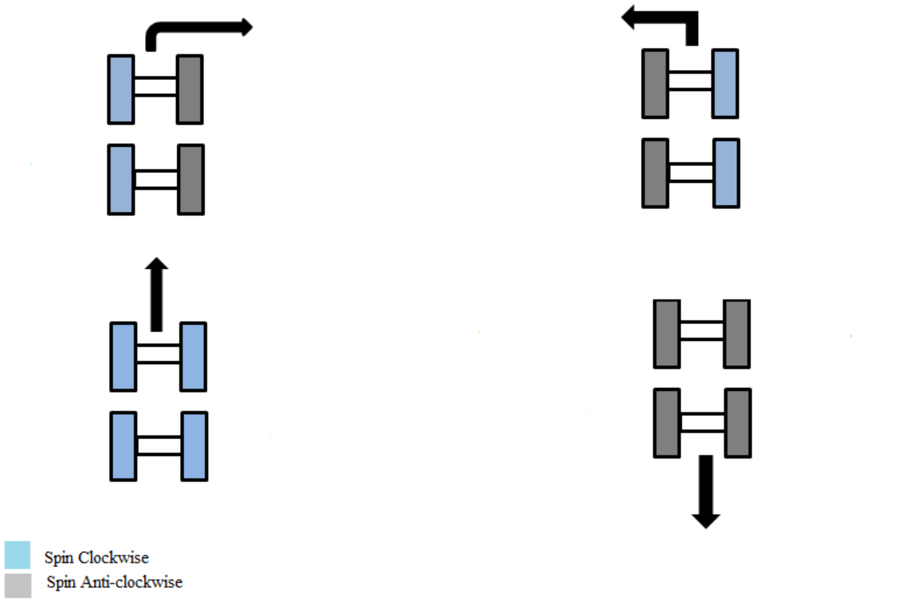


Figure 4.1-3 Motor Control on Wheel Turning using PWM

4.2 Completion measurement

The response and movement of the robot will be closely observed and monitored during the testing session. This project considers success by complete the criteria below:

- Move freely without human intervention (Suto mode)
- Avoid obstacles while travelling around (Auto mode)
- Able to make turn or change of direction of moving. (Both auto and manual mode)
- Provide auto and manual controlling mode (Control by radio transmitter)
- Provide live video feed to LCD monitor
- Precise control of motor moving when changing direction; increase the turning speed

4.3 Project testing and verification

In project testing module, the robot will be put into real environment and run in both autonomous and manual mode. In autonomous mode, we let it travel around the static and small environment for 5 to 10 minutes to observe the result of the obstacle avoidance system. Besides, we will use radio transmitter to control the movement of the wheel robot move within 80m to 150m range and observe the signal strength. Other function such as signal strength of video feed and the battery lifetime will be observed and optimized.

4.4 Project Cost

Item	**Cost per item (RM)**	**Quantity**	**Total cost (RM)**
Arduino Mega 2560	**165.00**	**1**	**165.00**
HC-SR04 Distance Sensor	**-**	**1**	**-**
Robot chasis (include wheels and dc motors)	**-**	**1**	**-**
Turnigy TGY-90S Servo	**-**	**2**	**-**
Wires (Packs)	**12.00**	**2**	**24.00**
Batteries (AA)	**1.20**	**6**	**7.20**

Table 4.4-1 Project cost

CHAPTER 5 IMPLEMENTATION AND DEVELOPMENT

5.1 Implementation

The figure and table below showed the pins connection of Arduino board and schematic and PCB diagram of custom made motor shield for the robot. The diodes (D1-D8) in the circuit are used to trap the voltage spikes and protect the components within the circuit. DC motors will produce high voltage than the original voltage supplied to resist the change in current due to change in polarity. The pin 1, 2, 7, 9, 10 and 15 are connect to the arduino board to receive signal for enable the input pins 2, 7, 10 and 15. The DC motor are ready to run when these pins are enabled. The motor speeds are adjustable since we are using the PWM signal to communicate with.

5.2 Pins connection and wiring

Pins on Arduino	Connection	Remarks
2 (PWM)	Pin 2 of L293B in PCB board 1	Control speed of DC motor
3 (PWM)	Pin 7 of L293B in PCB board 1	Control speed of DC motor
4 (PWM)	Pin 15 of L293B in PCB board 1	Control speed of DC motor
5 (PWM)	Pin 10 of L293B in PCB board 1	Control speed of DC motor
6 (PWM)	Pin 2 of L293B in PCB board 2	Control speed of DC motor
7 (PWM)	Pin 7 of L293B in PCB board 2	Control speed of DC motor

8 (PWM)	Pin 15 of L293B in PCB board 2	Control speed of DC motor
9 (PWM)	Pin 10 of L293B in PCB board 2	Control speed of DC motor
11 (PWM)	Channel 2 of radio receiver	Receive signal from x-axis of right stick
12 (PWM)	Channel 3 of radio receiver	Receive signal from y-axis of left stick
13 (PWM)	Channel 6 of radio receiver	Receive signal from
30	Pin 1 of L293B in PCB board 2	Enable pin
31	Pin 1 of L293B in PCB board 1	Enable pin
32	Pin 9 of L293B in PCB board 2	Enable pin
33	Pin 9 of L293B in PCB board 1	Enable pin
34	Echo pin of HC-SR04 Sonar Sensor	Echo output of Sensor
35	Trigger pin of HC-SR04 Sonar Sensor	Trigger input of Sensor
46	Signal pin of micro servo	Input pin for servo position
5v	VCC terminal	5v supply
GND	Ground terminal	Ground supply

Table 5.2-1 Pins assignment

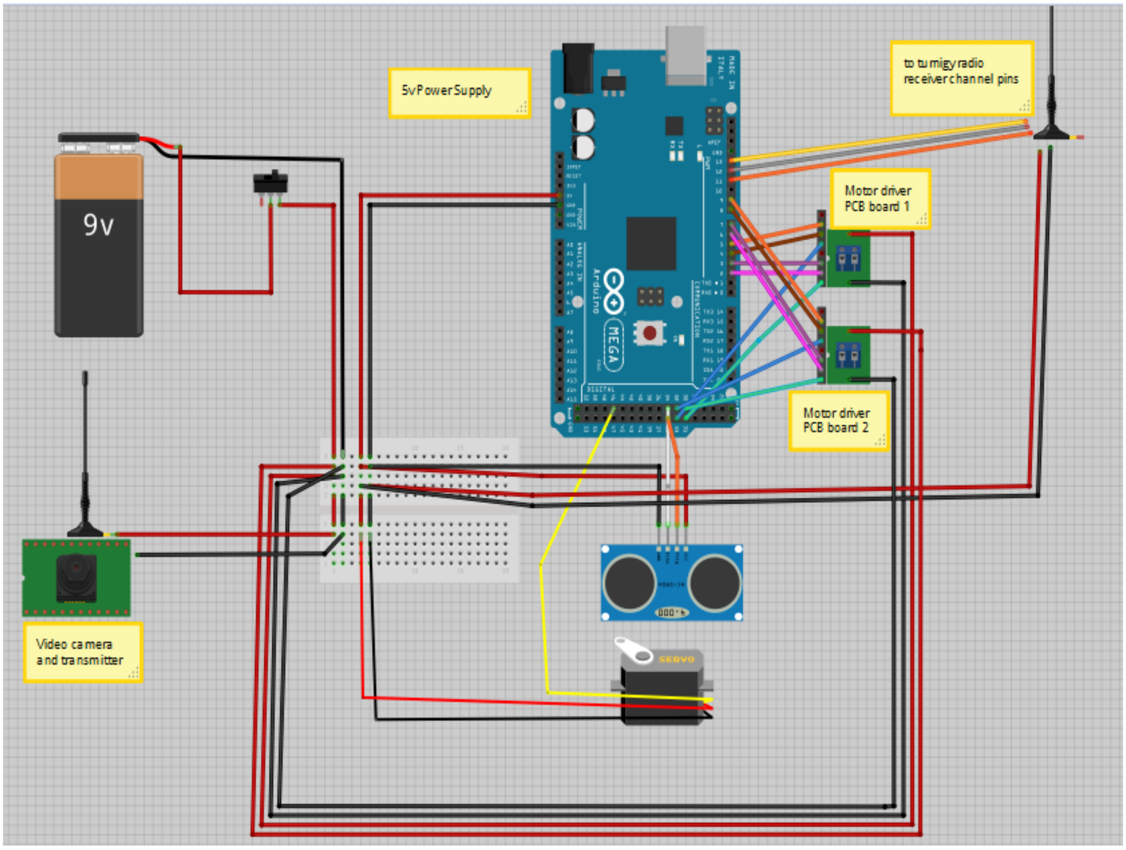
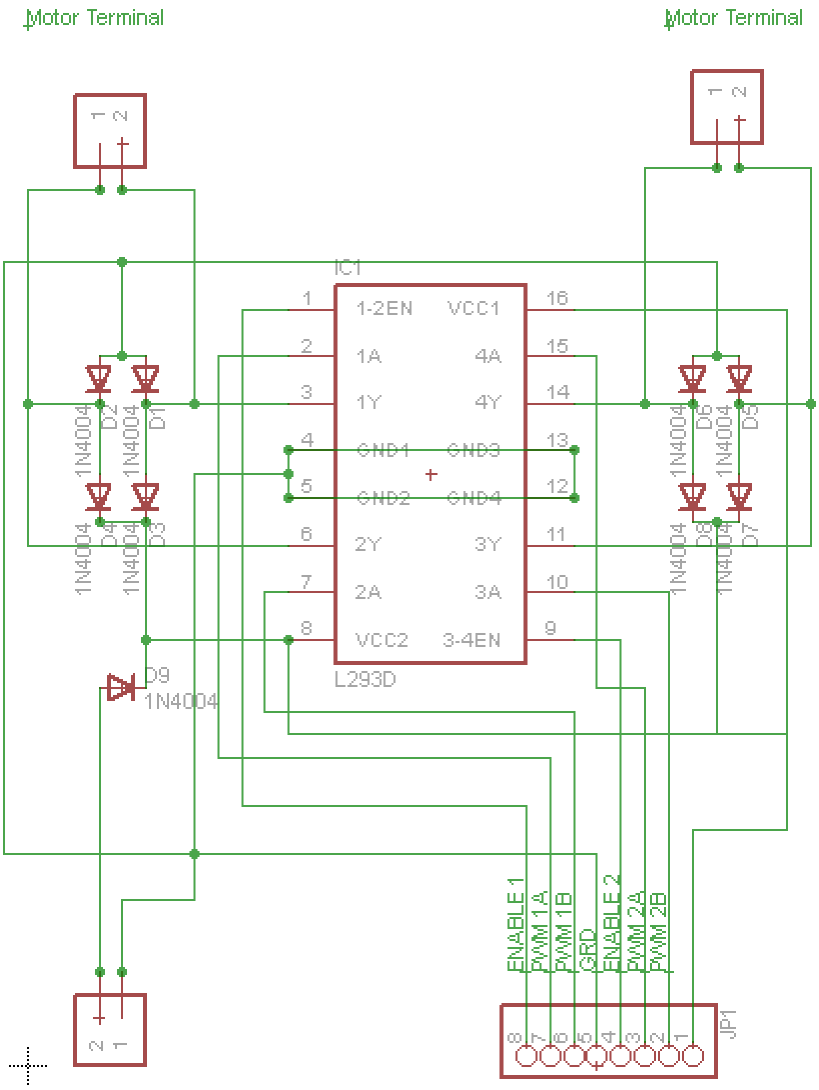


Figure created using Fritzing software

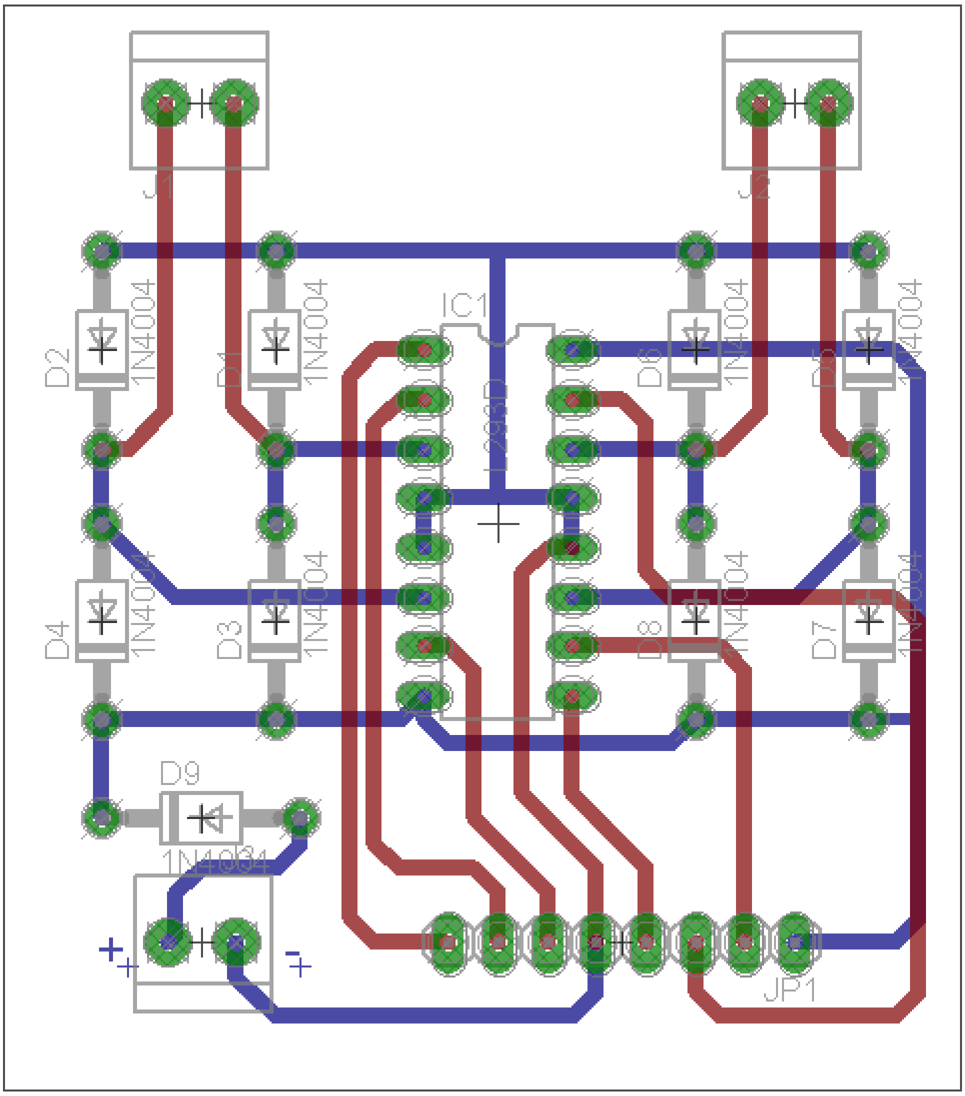
Figure 5.2-1 Project Wiring and Pins Connections

5.3 Schematic and PCB design



Schematic diagram created by Eagle software

Figure 5.3-1Schematic diagram of motor shield



Schematic diagram created by Eagle software

Figure 5.3-2 PCB (Printed Circuit Board) diagram of motor shield

When implementing the robot and circuit, there are some precautions should be notice to prevent short circuit or component damaged.

- Put the fingers onto the L293 IC when running to see if it getting too hot.

- The L293 ground connects to the battery and ground.
- If the L293 IC produces more heat means the drive needed to be change or replace in order to handle more current.

5.4 Processes and flow of program

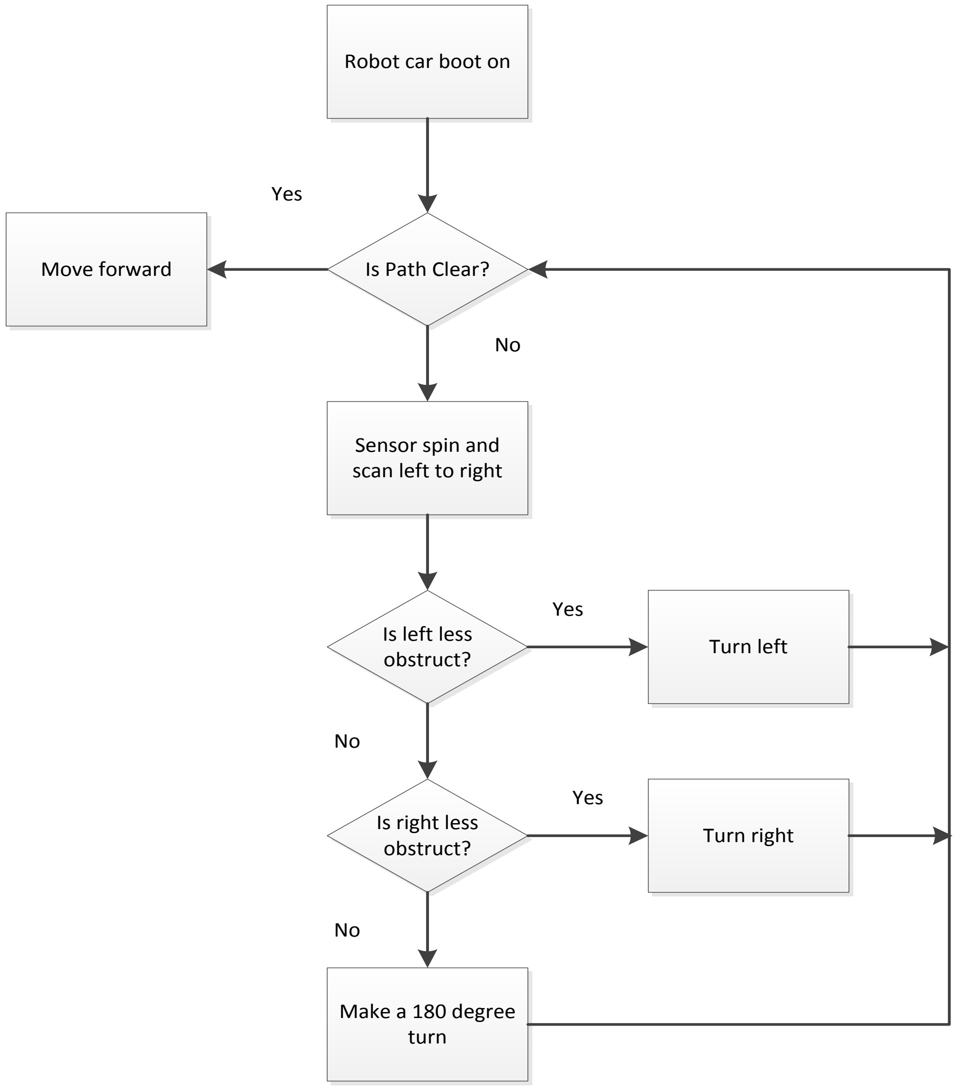


Figure 5.4-1 Flow chart of Autonomous Robot Car Program Sequence

CHAPTER 6 CONCLUSION

The proposed tunnel inspection robotic car has provided the solution for the problem laid out in chapter one. This very car is capable of replacing humans in undertaking inspection jobs in hazardous areas and even areas where it would be challenging dimension wise for humans to venture. Once switched on, the car is capable of manoeuvring its surroundings competently while simultaneously providing multi-angled video feedback to the user. The user also has the option of changing the car to manual control where its movements are tracked and controlled via remote control. The issue of weak and faltering signals has, without say, been addressed by a failsafe mechanism that will alert the user when the car has reached the limit of the signal radius. Besides that, the car will also rotate 180 degrees to allow it to embark on its return journey.

REFERENCES

American Psychiatric Association. (1994). Diagnostic and statistical manual of mental disorders (4th ed.). Washington, DC: Author.

American Psychological Association. (2003). Electronic references. Retrieved October 8, 2006, from http://www.apastyle.org/ref.html.

Christopher, P. (2006). Unleashing change: A study of organizational renewal in government. Administrative Science Quarterly, 51(2), 117-123. Retrieved October 8, 2006, from http://proquest.umi.com/ed32457.html

Mak, J. (2006, October 4). More public holidays for workers. The Star, p. N6.

Martin, S., Smith, L., Forehand, M. R., Mobbs, R., Lynch, T. F., Renfrew, E. J., et al. (2001). Use of collaborative technology in business, The Academy of Management Executive, 19, 130-138.

Playfair, J. H., & Bancroft, G. J. (2004). Infection and immunity. (2nd ed.). Oxford: Oxford University Press.

Reppel, F. G. (2003, February 13). Conflict and cognitive control. Natural Science, 33, 969-970.

Roeder, K., Howdeshell, J., Fulton, L., Lochhead, M., Craig, K., Peterson, R., et. al. (1967). Nerve cells and insect behavior. Cambridge, MA: Harvard University Press.

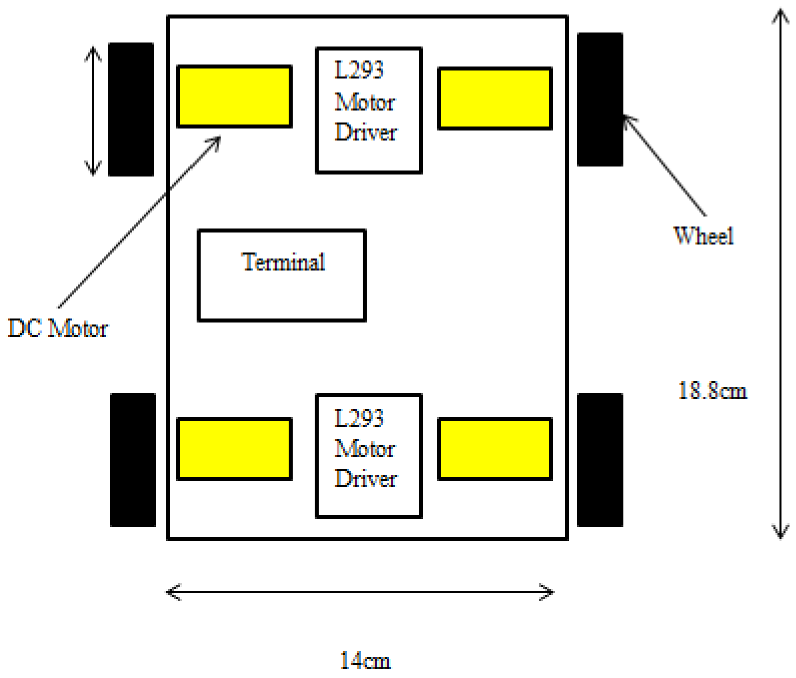
Serlin, R. C., & Lapsley, D. K. (1985). Rationality in psychological research: The good-enough principle. American Psychologist, 40, 73-83.

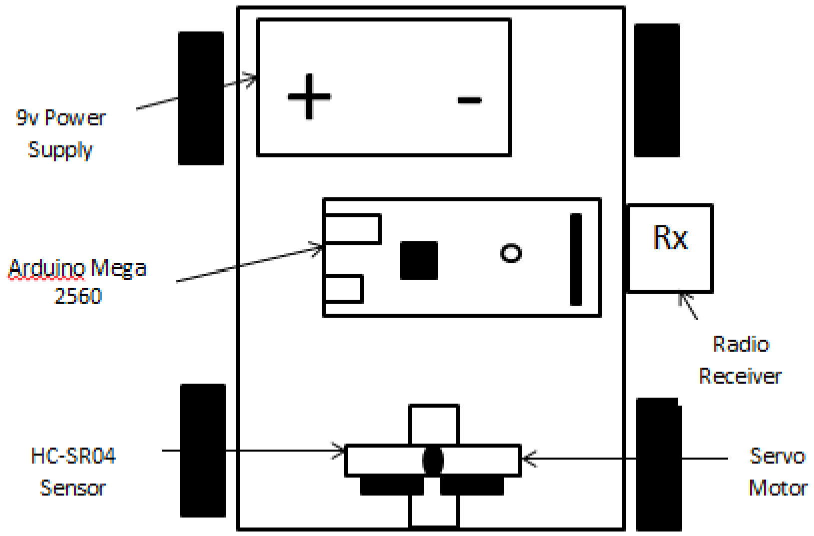
Skinner, M. E., Sun, K. F., & Liu, R. P. (2005). Small group learning, Psychological Bulletin, 26, 57-63.

Walker, M. M. (2001). Transfer of lexical information in adults with reading disorders. Perceptual and Motor Skills, 93 (1), 257-267.

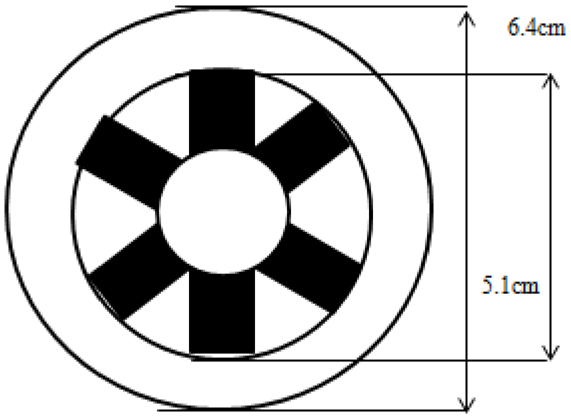
APPENDICES

APPENDIX A: Project Mechanical Structure





Measurement of project autonomous robot car



Measurement of robot wheels

APPENDIX B: Arduino Coding

```
#include <Servo.h>

//Motor Sheild 1 (Front)
#define fmotor1Pin 9   //H-bridge pin 10 to Arduino pin 9
#define fmotor2Pin 8  //H-bridge pin 15 to Arduino pin 8
#define f_enablePin 30 //H-bridge 1st enable pin
#define fmotor3Pin 7  //H-bridge pin 7 to Arduino pin 7
#define fmotor4Pin 6 //H-bridge pin 2 to Arduino pin 6
#define f_enable2Pin 32 //H-bridge 2nd enable pin

//Motor Sheild 2 (Back)
#define bmotor1Pin 5 //H-bridge pin 10 to Arduino pin 5
#define bmotor2Pin 4  //H-bridge pin 15 to Arduino pin 4
#define b_enablePin 31  //H-bridge 1st enable pin
#define bmotor3Pin 3  //H-bridge pin 7 to Arduino pin 3
#define bmotor4Pin 2  //H-bridge pin 2 to Arduino pin 2
#define b_enable2Pin 33  //H-bridge 2nd enable pin

#define echoPin 34 // Echo Pin
#define trigPin 35 // Trigger Pin

//Speed constant
#define nSpeed 100
#define fSpeed 80

long duration, distance;
```

```
int leftDistance, rightDistance, ch1, ch2, ch3, move, moveAbs,
turnAbs, swt, turn;
Servo irMotor;

void setup(){
  pinMode(fmotor1Pin, OUTPUT);  //set all H-bridge leg to ouput
pin
  pinMode(fmotor2Pin, OUTPUT);
  pinMode(f_enablePin, OUTPUT);
  pinMode(fmotor3Pin, OUTPUT);
  pinMode(fmotor4Pin, OUTPUT);
  pinMode(f_enable2Pin, OUTPUT);
  pinMode(bmotor1Pin, OUTPUT);
  pinMode(bmotor2Pin, OUTPUT);
  pinMode(b_enablePin, OUTPUT);
  pinMode(bmotor3Pin, OUTPUT);
  pinMode(bmotor4Pin, OUTPUT);
  pinMode(b_enable2Pin, OUTPUT);
  pinMode(11, INPUT); // Set channel pins as input pins
  pinMode(12, INPUT);
  pinMode(13, INPUT);
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin, INPUT);
  irMotor.attach(46); //attach motors to proper pins
  irMotor.write(90); //set PING))) pan to center
  Serial.begin(9600); //begin serial connection with computer
for debug purpose
}

void loop(){
```

```
    ch1 = pulseIn(11, HIGH, 25000); // Read the pulse width
of
    ch2 = pulseIn(13, HIGH, 25000); // each channel
    ch3 = pulseIn(12, HIGH, 25000);

    move = map(ch1,1000,2000,-500,500); //map the pulse
received between -500 and 500 to center zero
    move = constrain(move,-200,200); //set the limit of motor
moving speed
    turn = map(ch3,1000,2000,-500,500); //map the pulse
received between -500 and 500 to center zero
    turn = constrain(turn,-200,200); //set the limit of motor
turning speed
    swt = ch2;

    if (swt > 800 && swt < 1000){  //If the ID swtich is 0
(the pulse received is between 800 to 1000ms) then stop
        pinStat(false);
}

    if (swt > 1000 && swt < 1500 ){  //If the ID swtich is 1
(the pulse received is between 1000 to 1500ms) enter manual
mode
         manualMode();
}

    if (swt > 1600){    //If the ID swtich is 0 (the pulse
received is between 1500 to 2000ms) enter automode
        autoMode();
```

```
    }
}

void autoMode(){

    irMotor.write(90);
      //set servo position to center
    int  distanceFwd  =  ping();  //read  the  sensor  distance
value

    if(distanceFwd > 50){
        pinStat(true);  //set enable high
        moveForward(fSpeed);   //robot car move forward with
speed 80 which declared
}

    else{
        pinStat(false); //set enable pin low
            //Serial.println(distance);
        irMotor.write(0); //set the servo position to right
        delay(100);
        rightDistance = ping(); //read the sensor distance
value
        delay(500);
        irMotor.write(178); //set the servo position to left
        delay(700);
        leftDistance  =  ping();  //read  the  sensor  distance
value
        delay(500);
        irMotor.write(90); //return to center
```

```
            delay(100);
            compareDistance();
}

}

void manualMode(){

   pinStat(true); //set enable pin high

   if(move > 50){ //if the channel 2 radio receive the signal value more than 50 then move forward
      moveForward(move);
    }

    if(move < -50){ //if the channel 2 radio receive the signal value less than -50 then move forward
      moveAbs = abs(move);
      moveReverse(moveAbs);
    }

    if(turn > 50){  //if the channel 3 radio receive the signal value more than 50 then turn right
      moveRight(turn);
    }

    if(turn < -50){   //if the channel 3 radio receive the signal value less than -50 then turn left
      turnAbs=abs(turn);
      moveLeft(turnAbs);
```

```
    }

    if((turn >= -50 && turn <=50) && (move >= -50 && move
<=50) ){ //if signal of channel 2 and 3 received is between -50
to 50 then the robot stop (speed = 0)
      halt(0);
    }
}

long ping()
{
  digitalWrite(trigPin, HIGH); //trigger wave
  delayMicroseconds(100);

  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH); //read the wave received
duration

  //Calculate the distance (in cm) based on the speed of sound.
  distance = duration/58.2;

  return distance;
  delay(300);
}

void compareDistance()
{

  pinStat(true);
```

```
  if (rightDistance < 50 && leftDistance < 50) //if distance of
both direction are less than 50cm then turn 180 degree around
  {
    moveRight(nSpeed);
    delay(1300);
  }

  else if(leftDistance > rightDistance) //if left distance is
greater than right then turn left
  {
    //turn left
    moveLeft(nSpeed);
    delay(700);
  }
  else if (rightDistance > leftDistance) //if right distance is
greater than left then turn right
  {
    //turn right
    moveRight(nSpeed);
    delay(700);
  }
}

void moveForward(int speed){  //function of move forward

    analogWrite(fmotor1Pin, speed); //set leg 1 of the H-bridge
in board 1 to high
    analogWrite(fmotor2Pin, 0);  //set leg 2 of the H-bridge in
board 1 to low
```

```
    analogWrite(fmotor3Pin, speed);   //set leg 3 of the H-
bridge in board 1 to high
    analogWrite(fmotor4Pin, 0);  //set leg 4 of the H-bridge in
board 1 to low
    analogWrite(bmotor1Pin, speed);   //set leg 1 of the H-
bridge in board 2 to high
    analogWrite(bmotor2Pin, 0);  //set leg 2 of the H-bridge in
board 2 to low
    analogWrite(bmotor3Pin, speed);   //set leg 3 of the H-
bridge in board 2 to high
    analogWrite(bmotor4Pin, 0);  //set leg 4 of the H-bridge in
board 2 to low
}

void moveReverse(int speed){ //function of reverse

      analogWrite(fmotor1Pin, 0); //set leg 1 of the H-bridge
in board 1 to low
      analogWrite(fmotor2Pin, speed);   //set leg 2 of the H-
bridge in board 1 to high
      analogWrite(fmotor3Pin, 0);  //set leg 3 of the H-bridge
in board 1 to low
      analogWrite(fmotor4Pin, speed);   //set leg 4 of the H-
bridge in board 1 to high
      analogWrite(bmotor1Pin, 0);  //set leg 1 of the H-bridge
in board 2 to low
      analogWrite(bmotor2Pin, speed);   //set leg 2 of the H-
bridge in board 2 to high
      analogWrite(bmotor3Pin, 0);  //set leg 3 of the H-bridge
in board 2 to low
```

```
      analogWrite(bmotor4Pin, speed);  //set leg 4 of the H-
bridge in board 2 to high
}

void moveRight(int speed){ //function of moving right
      analogWrite(fmotor1Pin, 0); //set leg 1 of the H-bridge
in board 1 to low
      analogWrite(fmotor2Pin, speed);  //set leg 2 of the H-
bridge in board 1 to high
      analogWrite(fmotor3Pin, speed);  //set leg 3 of the H-
bridge in board 1 to high
      analogWrite(fmotor4Pin, 0);  //set leg 4 of the H-bridge
in board 1 to low
      analogWrite(bmotor1Pin, speed); //set leg 1 of the H-
bridge in board 2 to high
      analogWrite(bmotor2Pin, 0);  //set leg 2 of the H-bridge
in board 2 to low
      analogWrite(bmotor3Pin, 0);  //set leg 3 of the H-bridge
in board 2 to low
      analogWrite(bmotor4Pin, speed);  //set leg 4 of the H-
bridge in board 2 to high
}

void moveLeft(int speed){ //function of moving left
      analogWrite(fmotor1Pin, speed); //set leg 1 of the H-
bridge in board 1 to high
      analogWrite(fmotor2Pin, 0);  //set leg 2 of the H-bridge
in board 1 to low
      analogWrite(fmotor3Pin, 0);  //set leg 3 of the H-bridge
in board 1 to low
```

```
	analogWrite(fmotor4Pin, speed);  //set leg 4 of the H-bridge in board 1 to high
	analogWrite(bmotor1Pin, 0);  //set leg 1 of the H-bridge in board 2 to low
	analogWrite(bmotor2Pin, speed);  //set leg 2 of the H-bridge in board 2 to high
	analogWrite(bmotor3Pin, speed);  //set leg 3 of the H-bridge in board 2 to high
	analogWrite(bmotor4Pin, 0);  //set leg 4 of the H-bridge in board 2 to low
}

void halt(int speed){ //function of stop
	analogWrite(fmotor1Pin, 0); //set all H-bridge leg to low
	analogWrite(fmotor2Pin, 0);
	analogWrite(fmotor3Pin, 0);
	analogWrite(fmotor4Pin, 0);
	analogWrite(bmotor1Pin, 0);
	analogWrite(bmotor2Pin, 0);
	analogWrite(bmotor3Pin, 0);
	analogWrite(bmotor4Pin, 0);
}

void pinStat(boolean stat){ //function of enable pin setting
	digitalWrite(f_enablePin, stat);
	digitalWrite(f_enable2Pin, stat);
	digitalWrite(b_enablePin, stat);
	digitalWrite(b_enable2Pin, stat);
}
```

APPENDIX C: Datasheet