Design and implementation of a robot to aid bedridden persons

Prepared By

Vayun Bhujan

A dissertation submitted in partial satisfaction of the

requirements for the degree of Bachelor

of Engineering in Mechatronics

Supervised By

Dr. Robert Ah King

University of Mauritius

Faculty of Engineering

2015

(Submitted March 31, 2015)

Table of Contents

List of Figures

List of Tables

Acknowledgment

I would like to express my profound thanks to many people who have contributed enormously in terms of their assistance and knowledge to the development of this project.

Firstly, I wish to thank sincerely my supervisor, Dr. Robert Ah King for his precious help by providing the right guidance, understanding and great expertise to allow me to complete my project. I am extremely grateful for his presence and support throughout the whole process of the final year project. His valuable insight enabled me to progress in each step of the project with ease.

In addition to that, my deepest thanks will go to Mr. Sujeebun, the lab technician who always available to provide a helping hand as well as Mr. Govinden who assisted and guided me in the construction of the prototype.

Finally, my heartfelt gratitude goes to my family who have endlessly supported me in every of my endeavors. Their warm support has allowed me to realize a project of this magnitude and their motivation increased my determination to produce a quality project. I extend further thanks to my friends who have been really supportive and been of immense help throughout the development this final year project.

UNIVERSITY OF MAURITIUS

PROJECT/DISSERTATION DECLARATION FORM

Name:
Student ID:
Programme of Studies:
Module Code/Name:
Title of Project/Dissertation:
Name of Supervisor(s):

Declaration:

In accordance with the appropriate regulations, I hereby submit the above dissertation for examination and I declare that:

(i) I have read and understood the sections on **Plagiarism and Fabrication and Falsification of Results** found in the University's "General Information to Students" Handbook (20…./20….) and certify that the dissertation embodies the results of my own work.

(ii) I have adhered to the 'Harvard system of referencing' or a system acceptable as per "The University of Mauritius Referencing Guide" for referencing, quotations and citations in my dissertation. Each contribution to, and quotation in my dissertation from the work of other people has been attributed, and has been cited and referenced.

(iii) I have not allowed and will not allow, anyone to copy my work with the intention of passing it off as his or her own work.

(iv) I am aware that I may have to forfeit the certificate/diploma/degree in the event that plagiarism has been detected after the award.

(v) Notwithstanding the supervision provided to me by the University of Mauritius, I warrant that any alleged act(s) of plagiarism during my stay as registered student of the University of Mauritius is entirely my own responsibility and the University of Mauritius and/or its employees shall under no circumstances whatsoever be under any liability of any kind in respect of the aforesaid act(s) of plagiarism.

Signature:	Date:

Abstract

A remote controlled system is one which is controlled wirelessly from two points of location. With the advent of the technology, several forms of remote control communication have been available on the market. The recent advancement in Wi-Fi technology has been beneficial as it has prompted the development in new concepts.

Nowadays, autonomous systems also are widely being favored as they provide ease of task completion. With the help of autonomous systems, tasks can be performed automatically without the need for human intervention.

In Mauritius, the number of bedridden persons is increasing at an exponential rate such that the need for caring of these people may become a social problem. Therefore, an assistance system to support them is required and implemented through this project. Several components are involved in the proposed assistance system for bedridden persons.

The main application of this project is to allow the bedridden persons to navigate through their house with the remote control help of the robot as well as be able to move to specific places autonomously whereby another person can place an object so that the robot brings it back to the patient.

The semi-automatic part of the robot will be based on an Android mobile phone and a Wi-Fi set up whereby commands can be given to the robot from the phone via the Wi-Fi network. Commands from the phone will be passed through a web server and ultimately to the WIFI shield on the robot which will further give the commands to the Arduino microcontroller to control the motors of the robot.

A line following system will be used for the autonomous part of the robot whereby it will follow a specific path of action. The robot will move along this track to perform a required task. Proportional Integral Derivative (PID) control will be integrated with this part of the system to ensure a smooth operation of the robot.

Chapter 1: Introduction

1.1. Overall Description

A bedridden person may be of any age and faces several difficulties in his everyday life. He may not be able to move to fetch an item in the kitchen depending on the type of illness. Due to motion troubles, the sight of events happening in the house may also be rare or even non-existent. Some bedridden persons suffer from psychological problems from the situation they live in daily as they often feel alone and incapable of conversing with other people in their houses with ease.

According to figures gathered from the Central Statistics Office of Mauritius, the number of persons who benefit from basic invalid pension is constantly increasing from the year 2009 to 2013. These people are mainly those who suffer from chronic ill health and in many cases proper assistance care should be provided to them. Among these individuals, many fall under the category of being bedridden as per the definition of an invalid person. The high number of beneficiaries of basic invalid pension of 30930 in 2013 as compared to 27169 in 2009 indicates the need to provide a better facility system to care for these persons.

Some of the difficulties encountered by these people can be solved by developing a robot to help them meet some of their needs and improve their lifestyle. The robot that will be designed will allow the bedridden person to give it a command to go to the kitchen where someone can load a glass of water or a plate of food on a tray on the robot. Furthermore, the person may control the robot through a mobile phone such that it can move in the house and he can see the surroundings.

1.2. Problem Description

Some of the problems encountered by the bedridden persons are:

- Inability or difficulty to move.
- Feeling of restlessness when disturbing family members repeatedly to fetch food.
- Feeling of boredom.
- Feeling of loneliness.
- Inability to see their surroundings.
- Inability to converse easily with people in the house.

1.3. Aims of project

- Design of the robot to help the bedridden persons.
- Incorporate a user friendly interface.
- Facilitate and improve the lifestyle of the bedridden persons.
- Design and implementation of remote control system via wifi
- Design and implementation of a line following system for autonomous performance of the robot.

1.4. Proposed Methodology

- Literature review
- Determine requirements and specifications of the system
- Conceptual designs and proposed solution
- Component selection and mechanical design
- Programming and electrical design
- Implementation of prototype
- Cost analysis and report writing

1.5. Gantt Chart

Table 1.1: Gantt Chart

	Aug 14	Sep	Oct	Nov	Dec	Jan	Feb	Mar 15
Literature Review								
Determine requirements and specifications of the system								
Conceptual designs and proposed solution								
Component selection and mechanical design								
Programming and electrical design								
Implementation of prototype								
Cost analysis and report writing								

Chapter 2: Literature Review

2.1. Introduction

In this section, key ideas and observations will be dealt with from the works already carried out by various individuals and the components present in different existing systems will be assessed. Consequently, an analysis will be done in order to have a knowledge of the elements that will be present in the proposed solution.

2.2. Existing Systems

Nowadays, different systems exist to help bedridden persons. These systems have been developed so as to allow the bedridden persons to feel more active in their everyday lives. They are mainly either automatic or semi-automatic systems.

I. Mobile robot system to aid the daily life for bedridden persons

This system has been designed by Shibaura Inst. of Technol. in Japan and consists of a mobile system, a manipulator and a visual sensor unit. A command can be given by the user via a monitor as a result of which the computer will measure the three dimensional distance in space. The manipulator will then grab the small object and bring it to the person.

II. Swing type obstacle sensing for remote internet controlled substitute robot

Kanagawa Institute of Technology in Japan came up with this system in which the robot consists of personal computer, a display, a Charge Coupled Device (CCD) camera, a microphone and a speaker. The operator can make use of the same set of accessories that is present on the robot. The two computers are connected using an internet connection. The presence of an ultrasonic sensor allows obstacle avoidance. This system’s main purpose is to help bedridden children to follow their school classes from distance.

III. Tele-care system by hoist-trolley robot arm from multi controllers and cellphone controller

Developed by the University of Tokyo in Japan, this robot carries light loads such as drinks or books to bedridden patients. The control of the robot is done with multiple remote controllers and cellphone control. The remote terminal is set at a remote place and the robot communicates voice and video signal whilst also being able to converse with the bedridden person. The mobile controller allows the care-giver to move around to different patients.

IV. Human Support Robot (HSR)

The design and implementation of the HSR was done by Toyota to help bedridden persons to pick up objects from the floor or a table and also to open curtains. The robot has both a Prosense (Microsoft Kinect) sensor and stereo cameras in its head which allows it to sense depth and visually identify people and objects. The Figure 2.1 shows the Human Support Robot designed by Toyota.

Figure 2.1: Human Support Robot by Toyota [https://encrypted-tbn3.gstatic.com/images?q=tbn:ANd9GcTAw-ADWaQkDHrA0gU9OI-oeBoTfv2WjM9k5LrWDFbJ4zT5jp4S]

V. Kompaï

French service robotics company Robosoft has introduced a robot called Kompaï designed to assist elderly and disabled people and others who need special care. Being able to talk, understand speech and navigate autonomously, the robot displays a wide variety of functions. In addition to that, it reminds people of meetings, keeps track of shopping lists, plays music and can be used for videoconferencing to talk with their family or doctors. Figure 2.2 shows the Kompaï robot.

Figure 2.2: Kompaï Robot by Robosoft company

[http://spectrum.ieee.org/image/1536260]

VI. Carebot

Developed by Gecko Systems, Carebot is a robot whose maneuvering is fully autonomous as it avoids static as well as dynamic obstacles and can also patrol. Interaction with the patients is also present in the form of question asking, setting of reminders and response to other verbal commands. Gecko also designed its robot to connect with third-party medical monitoring systems that measure blood pressure and sugar, pulse rate and other medical information which it sends to physicians via the Internet. Below in Figure 2.3, the robot is in action to help an old person.

Figure 2.3: Carebot designed by Gecko Systems

[http://www.geckosystems.com/images/CareBot_Grandma_09.jpg]

2.3. Summary

The table 2.1 below illustrates the advantages and disadvantages of the systems mentioned above and the systems are compared.

Table 2.1: Advantages and disadvantages of different systems

System	Adanvatages	Disadvantages
I.	• Presence of a robotic arm to grab objects • Visual sensor unit to obtain 3-D information	• Implementation tedious • Lack of communication between people
II.	• Camera vision • Connection to the internet • Obstacle avoidance sensors	• Designed mainly for bedridden school children to follow distance classes
III.	• Multiple remote controllers and cellphone control • Two-way communication • Carrier of small objects • Presence of video imaging	• Design of this robot takes time

IV.	• Identifies people and objects • Picks up objects from the floor • Draws curtains • Presence of stereo cameras to obtain information about background depth	• Design of robot complex
V.	• Ability to talk • Understands speech • Autonomous navigation of the robot • Allows the playing of music • Keeps reminders for people	• Does not have the ability to pick up objects
VI.	• Robot can run autonomously • Avoids static as well as dynamic obstacles • Interaction with patients • Communication with third party medical monitoring system • Medical information sent via internet to physicians for analysis	• Complex design of the robot

2.4. Analysis

The above systems developed are the currently available types of robots that have been constructed to aid bedridden persons. Refined robot models are still being designed and may be available on the market within two to three years. In the proposed project, I will be aiming to design and implement a friendlier and more effective robot that will be of help to these individuals to improve their daily lives. Actually, there is no robot which offers the option of being both autonomous and semi-automatic at a reasonable price. Thus, the design of the robot will be carried out to tackle existing problems and to provide for better facilities to help the bedridden persons. The designed robot would be controlled from a mobile phone as well as be able to navigate autonomously from one place to another. The incorporation of a robotic arm to the robot would be a huge bonus but would not be feasible due the limited lapse of time provided to construct the robot. The aspects that will be present in the system are as follows:

- Friendly design
- Low cost system
- Carrier for small loads such as drinks, books, plates.
- Easy to use and maintain
- User-friendly interfacing
- Easy control of robot
- Two-way communication
- Camera vision
- Cellphone control via Wi-Fi network
- Obstacle avoidance
- Automatic mode

Chapter 3: Concept Design

3.1. Introduction

In the concept design, appropriate components used in the system will be discussed. The robot to help bedridden persons will comprise of the following concepts (as shown in the overview in figure 3.1):

- Remote control system of robot with a mobile phone through WIFI network
- Autonomous control system of robot through a line following system
- Ultrasonic sensors
- Distance sensor
- IP camera
- Drive system
- WIFI system (Adafruit module)
- Microcontrollers

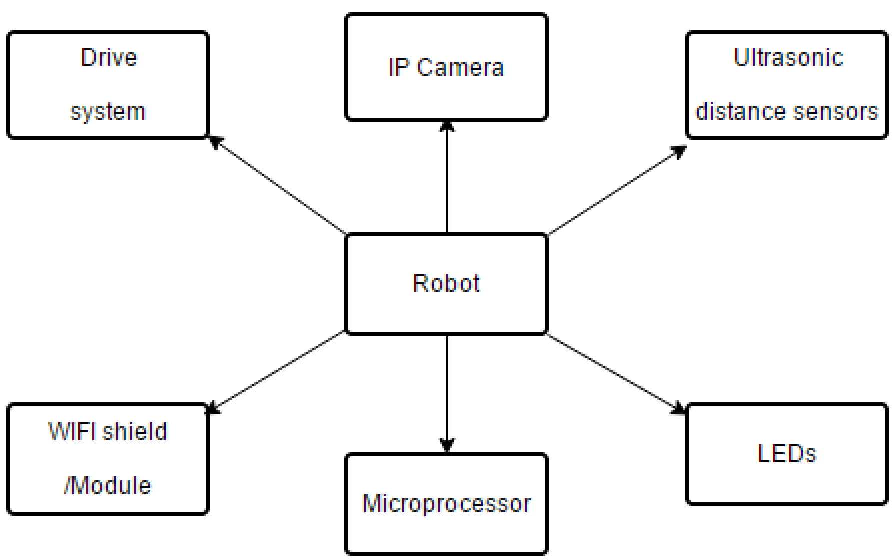


Figure 3.1: Overview of concept design

3.2. Remote control system

3.2.1. WIFI Bot Control

This remote control system is executed with an Android mobile phone and a microcontroller through WIFI network with a WIFI router. The Wifi Bot Control application needs to be downloaded on the Android phone. A webserver is used as a means of communication between the robot and the Wifi Bot Control program. The former interprets the commands sent by the application and provides a file by which the robot can read to fetch commands being sent to it. The webserver can form part of the robot or can be stand-alone.

The remote control system consists of the following elements represented in figure 3.2.

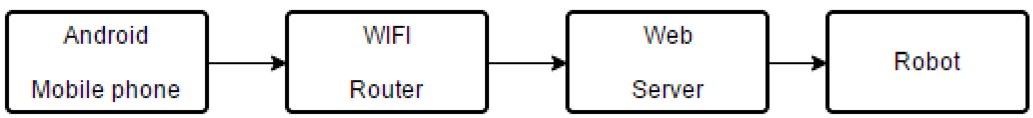


Figure 3.2: Block diagram of Wi-Fi control system

3.2.2. Radio Control System

The principle of a transmitter and receiver forms an integral part of the radio control system. The receiver decodes signal sent by the transmitter in order to take necessary actions. Different ways exist for transmission of radio signals and they include:

- Amplitude Modulation (AM)
- Frequency Modulation (FM)
- The 2.4 GHz Spread Spectrum

Most radio systems make use of amplitude modulation for the radio signal and the control positions are encoded with pulse width modulation. However, with the advent of the 2.4GHz technology, remote control systems make more use of this facility.

3.2.3. Summary

After considering the Wi-Fi system for control of the robot and that of the radio control system, it can be deduced that the Wi-Fi system is more suitable for this control purpose as it offers more advantages than the radio control system. In the case of the radio control system, the robot needs to be in the range of sight of the user whilst with the Wi-Fi system, the robot could be controlled in the Wi-Fi range and the user would be able to view the surrounding of the robot from his mobile phone with which the robot can also be controlled.

3.3 Autonomous control system

For a robot, the ability to navigate in its environment is important. Robot navigation is the capability of the robot to determine its own position in its frame of reference and to move towards a specific destination.

Various navigation systems exist for the automatic control of a robot. They comprise mainly of:

(i) Line following navigation system
(ii) Vision based navigation system
(iii) Beacons-guided navigation system
(iv) Global Positioning System (GPS)
(v) Map planning navigation system
(vi) Wall following navigation system

3.3.1. Line Following navigation system

Line following robot (figure 3.3) is a self-operating mobile machine designed to follow a line drawn track on the floor. The track can be a black line which is visible on a white surface or vice versa or it can also be a magnetic field which cannot be seen. The robot keeps moving along its track through the use of photosensors. Infrared (IR) or visible light is emitted from the emitter. The IR light is preferred to the visible one to avoid interference from the visible light present in the background. The former is also present in the atmosphere but has a lower intensity than the visible light, therefore making the IR light a more reliable output.

To improve the accuracy of the sensors, they should be shielded properly to prevent the background emissions. The emitted light hits the surface and gets reflected. In case of a white surface, the intensity of the reflected light is more compared to that when it comes from the reflection of a black surface.

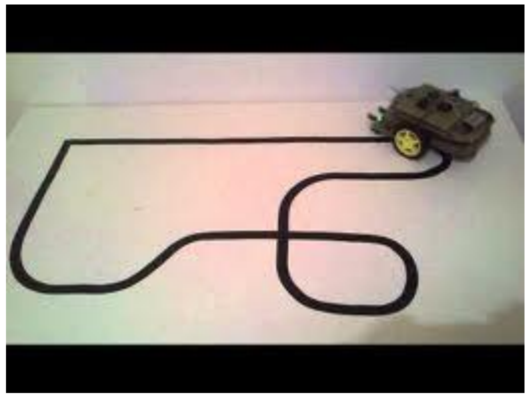


Figure 3.3: Line following robot

[https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTIHwyTgDiugS4HjPJu3EXsOlmpcGLPHMhGJihHbSAWl_ISsKC80w]

3.3.2. Vision based navigation system

The above mentioned navigation system makes use of optical sensors including laser-based range finder and photometric cameras using Charge-Coupled Device (CCD) arrays to extract the visual features necessary for the robot to localize its position in the surrounding environment.

The image gathered from this system can be analyzed and processed in different ways which makes use of representation of the environment, sensing models and localization algorithms. From the volume of information obtained with machine vision systems, extraction of visual features for positioning is not straightforward.

One of the associated problems with this system is the designing process for the algorithms to enable the robot to navigate autonomously. The robot may take a lot of time to reach its destination but also precise information of the robot's destination should be fed into the system for better efficiency of the robot.

3.3.3. Beacons-guided navigation system

The Pololu IR beacons work in pairs for a robot to locate positions and move towards its desired location. Each board has infrared emitters that send signals in all direction for each beacon to detect the presence of the other which is illustrated in Figure 3.4. In spite of the fact that the hardware implementation and programming is easy, the IR beacons have a limited detection range which is often impacted considerably with obstacles such as walls and fluorescent light as well. Moreover, the cost of each module is extremely high.

For the purpose of the robot being designed, this type of navigation system is not appropriate due to its limitations in terms of precise navigation.

Figure 3.5 shows the Pololu IR beacon.

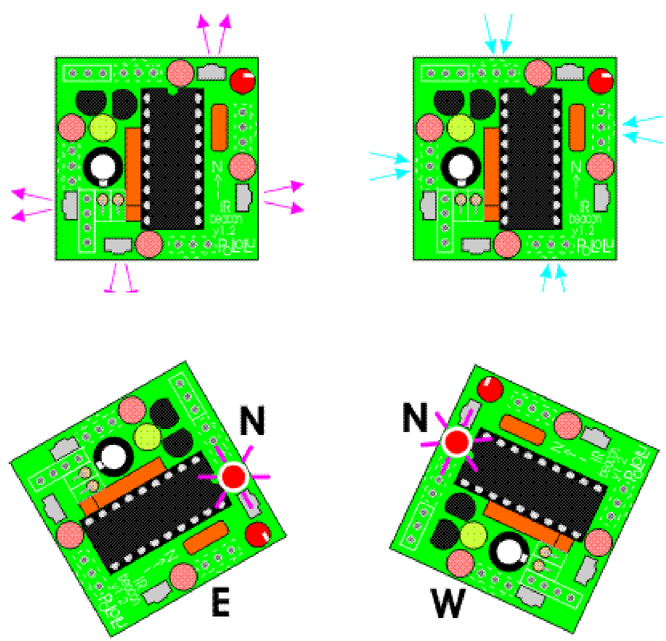


Figure 3.4: Working principle of IR beacons

[https://www.pololu.com/product/701]

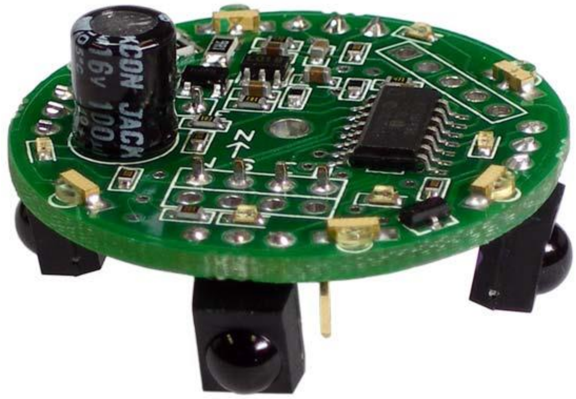


Figure 3.5: Pololu IR Beacon

[https://www.pololu.com/picture/view/0J90]

3.3.4. GPS

With the Global Positioning System (GPS), location and time of information can be provided with connections to four or more GPS satellites surrounding the Earth. Ublox NEO-6M GPS Module can be connected to the Arduino microcontroller to obtain points of location for the robot to navigate autonomously. However, the main drawback with this system is that the microwave signals generated by the satellites are vastly attenuated by roofs and walls which make it impracticable to implement this system for indoor purposes. For the case of our robot, this system is definitely not feasible as the robot will be piloted inside the house. Figure 3.6 illustrates the GPS module needed to implement this system.

Figure 3.6: GPS Module

[http://i.ebayimg.com/00/s/ODAwWDgwMA==/z/UmkAAOSwstxU2sid/$_57.JPG]

3.3.5. Map planning navigation system

Related to the study of plans and maps, this navigation system aims at autonomously guiding a robot to its final destination with reference to a designed map. With this method of navigation, dead reckoning is used which gathers previous data to determine current location. The revolution of the wheels can be tracked to obtain the absolute position of the robot and hence the robot can be programmed with this technique.

Map planning necessitates time and a good analysis of the plan of the building in which the robot will move. Given the fact that this method of navigation is tedious, it will be neglected in the design of the robot to aid bedridden persons.

3.3.6. Summary

For this type of robot, line following navigation system is better as it provides the precise movement of the robot to a specific location. The range of transmission of the beacons is low and is often compromised by interferences such as walls. The line following robot can move to and from a desired location efficiently without the need for human intervention. With this system of navigation, the implementation also is relatively easy and the program can be readily modified to improve the system which will be executed in this project with the introduction of obstacle avoidance to prevent the robot from collapsing with other objects in its path of navigation.

3.4. Drive system

The drive system deals with the ways in which the robot will move. The main driving systems are:

(i) Differential steering
(ii) Ackermann steering
(iii) Omni-directional steering

3.4.1. Differential Steering

Differential steering necessitates two motors connected to a wheel on either side of the robot. Speed difference is the key for the functioning of this type of steering. For example, when the speed of the right wheel is greater than the left wheel (speed slower or idle), robot turns left as illustrated in the figure below.

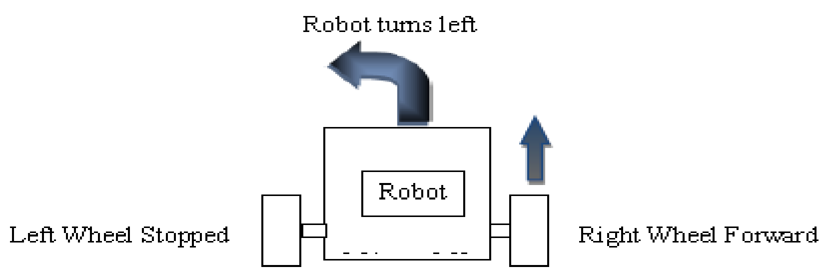


Figure 3.7: Robot with differential steering

The table below shows all possible directions of steering of the robot when left motor, M1 and right motor, M2 are on (1), off (0) or reversed (-1) respectively.

Table 3.1: Differential steering directions

Motor, M1	1	1	0	0	-1	-1	0
Motor, M2	1	0	1	-1	0	-1	0
Direction	Forward	Forward Right	Forward Left	Reverse Left	Reverse Right	Reverse	Idle

3.4.2. Ackermann Steering

Ackermann steering (figure 3.8) deals with the concept of making the inner wheel move at a greater angle than the outer wheel. The outer wheel has to travel a greater arc distance than the inner wheel.

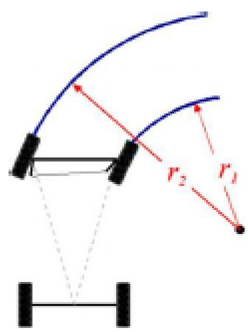


Figure 3.8: Ackermann Steering
[http://www.diracdelta.co.uk/science/source/a/c/ackermann%20steering/source.html#.VQCNMMKqqko]

Equation of arc length formula:

Since, r1 < r2,

$$\frac{N^0}{360^0} * 2\pi r_1 \quad < \quad \frac{N^0}{360^0} * 2\pi r_2$$

Calculations above deduce that the arc of inner wheel is lesser than that of outer wheel. This type of steering normally applies to the front wheel (rotates freely) and direction is changed by using either servo or stepper motor to provide the accurate rotation angle of both wheels.

3.4.3. Omni-directional Steering

Omni directional drive requires three or four wheels to be driven and they provide a high degree of freedom than many other systems. Each wheel requires a motor coupled to it. Generally, stepper and servo motor are used to drive three wheeled motors in order to provide the accurate angle of rotation of the wheel and dc motors are normally used for four wheel driving (stepper or servo can also be used but they are more expensive). Figure 3.9 demonstrates the wheels of an omni-directional steering robot.

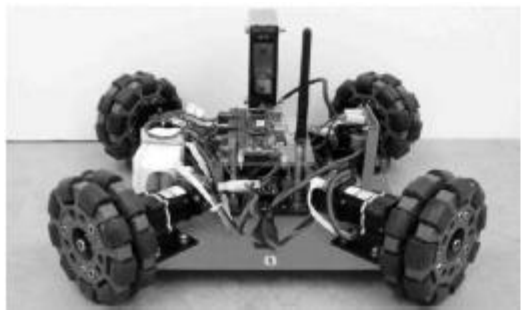


Figure 3.9: Omni-directional steering robot

[http://biorob.epfl.ch/files/content/users/175246/files/Public/Pictures/robot-firstpage2.jpg]

Table 3.2: Steering Systems with their advantages and disadvantages

Steering System	Advantages	Disadvantages
Differential Steering	• Low cost • High torque • Easy to implement • Easily controlled • Spinning possible • Easily programmed	• Prone to friction & wheel slippage
Ackermann Steering	• Excellent for high speed robots • Precise while turning	• Difficult to implement • High cost • Spinning not possible

Omni-directional Steering	• Greater stability • High degree of freedom	• Complex for building • Programming difficult • Expensive • Presence of additional motors & wheels

3.4.4. Summary

From the above table, it can be deduced that differential steering is the best solution for steering of the robot as it can turn in small areas easily. Furthermore, the steering is cheap and provides a high torque which is necessary for this robot. The implementation of the differential steering drive system is much easier as compared to the other driving systems proposed.

3.5. Proximity sensors

Detecting the presence of nearby objects without any physical contact is the main function of a proximity sensor. It can be in the form of an Infrared (IR) sensor which emits electromagnetic radiation and compares it with changes in the returned signal of in the form of an ultrasonic sensor. Proximity sensors have a high reliability and long lifetime due to the absence of mechanical parts and no physical contact between the sensor and the sensed object.

3.5.1. Ultrasonic Sensors

Ultrasonic sensors evaluate the presence of an obstacle by interpreting the echoes from radio or sound waves by measuring the time taken for the wave to come back. If the echo is back, through high level, the pulse width of the high output level is the time the ultrasound travels to the obstacle and returns. From information of the speed of sound and time taken, the distance between an obstacle and the robot can be calculated.

Test distance = (high level time× velocity of sound) / 2

The velocity of sound in air is approximately $340ms^{-1}$

Advantages:

- Lightweight and compact.
- Simple to use and implement.
- Adjustable ranges are achievable.
- Accurate distance measurement is feasible.
- It is a proactive sensor.
- Fast response time.
- Ultrasonic sensors can be used in all environments unlike IR sensors.

Disadvantages:

- Cost is relatively high compared to other sensors such as the bumper switch.

Figure 3.10: Ultrasonic sensor
[http://i.ebayimg.com/00/s/NTAwWDUwMA==/z/kLQAAOxyNyFS-xFw/$_12.JPG]

Module main technical parameters:

- Working Voltage : 5 V(DC)
- Static current : Less than 2 mA.
- Output signal : Electric frequency signal, high level 5V, low level 0 V.
- Sensor angle : Not more than 15 degrees.
- Detection distance : 2 cm to 450 cm.
- High precision : Up to 0.3cm
- Input trigger signal : 10us TTL impulse
- Echo signal : Output TTL PWL signal

Mode of connection:

(i) VCC
(ii) trig(T)
(iii)echo(R)
(iv)GND

3.5.2. Sharp Infrared Proximity Sensor

The Sharp Infrared Sensor is a distance measuring sensor unit comprising of an integrated combination of PSD (position sensitive detector), IRED (Infrared emitting diode) and signal processing circuit. It makes use of the triangulation method for detecting distance which is therefore not affected with parameters like the reflectivity of the object, the temperature of the environment and the operating duration. By knowing the angle between the emitted light and the sensor, the distance can be determined.

Advantages:

- Easy hardware implementation
- Quite easy programming
- Quite Compact

Disadvantages:

- Cannot detect obstacles from 0 cm to 20cm
- Short Range 20 cm to 150 cm

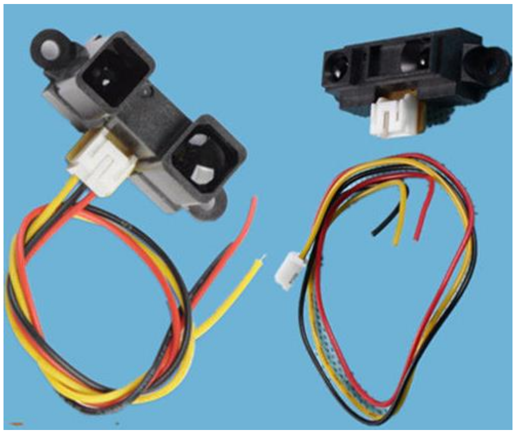
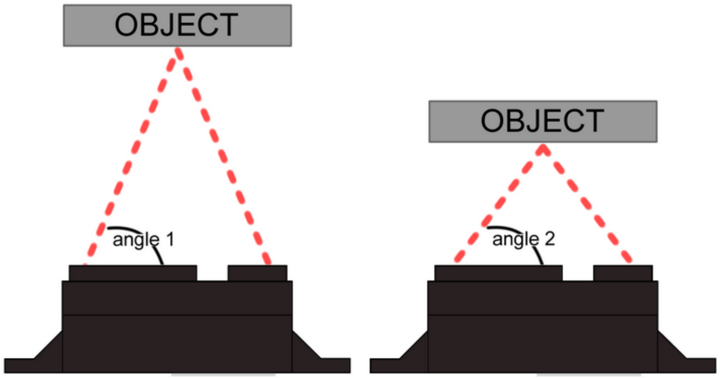


Figure 3.11: Sharp Distance Sensors

[http://i.ebayimg.com/00/s/NDQxWDUwMA==/z/G3UAAOSwajVUO7Q7/$_12.JPG
]



1Figure 3.12: Working principle of distance sensor

[http://www.robotc.net/blog/wp-content/uploads/2011/11/triangulation.png]

3.5.3 Summary

One of the main drawbacks with the IR sensor is that obstacles cannot be detected in the range of 0 to 20 cm which is not appropriate for the robot being designed. The ultrasonic is a more suitable for this application as the output it produces is more accurate and faster. Even though it is more costly, it is preferable to use a quality sensor for the functioning of the robot.

3.6 Microcontrollers

Nowadays, microcontrollers form part of many electrical applications ranging from simple to complex ones. The main brands of microcontrollers as shown in Figures 3.13, 3.14, 3.15 are the:

(i) Arduino microcontroller

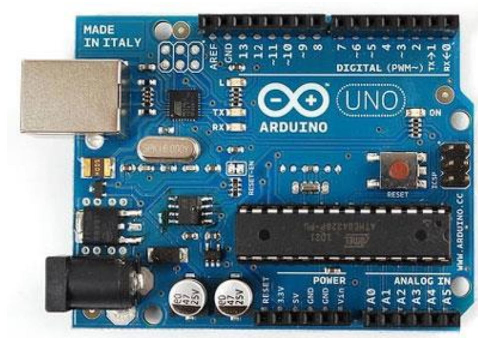


Figure 3.13: Arduino microcontroller

[http://gearguyd.com/blog/wp-content/uploads/2011/01/Arduino-Uno-Microcontroller-Board.jpg]

(ii) PIC microcontroller

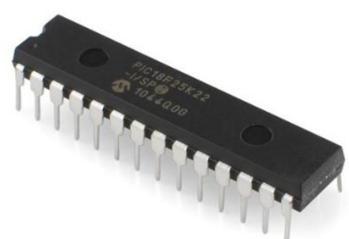


Figure 3.14: PIC microcontroller

[http://media.cdn-libelium.com/catalog/product/cache/1/image/9df78eab33525d08d6e5fb8d27136e95/p/i/picaxe_28x2_microcontroller__28_pin_.jpg] (iii)Raspberry Pi

Figure 3.15: Raspberry Pi board

[http://dlnmh9ip6v2uc.cloudfront.net/tutorialimages/RaspberryPi/Pi-board.jpg]

Table 3.3: Pros and Cons of the main types of microcontrollers

	Pros	Cons
Arduino Microcontroller	• Low power architectures • Easy to get started with great online support • Easy interfacing with sensors and data collection is very easy • Relatively cheaper • Windows based Operating System	• Memory limitations • Connectivity to the internet needs additional shields • Less powerful
Raspberry Pi Microcontroller	• Super powerful with lot of memory and processing capabilities. Expandable memory. • Presence of Ethernet shield	• Linux based Operating System

PIC Microcontroller	• Reliable and malfunctioning of PIC is less • Fast performance with RISC architecture • Low power consumption	• Lengthy programs because of RISC (35 instructions) • Program memory not accessible • Difficult to incorporate internet

3.6.1. Summary

With reference from Table 3.3, the Arduino microcontroller is best suited for use in the project due to its user friendly interfacing as well as being cost effective. Furthermore, shields can complement the microcontroller to provide for internet access which will be a major component in the robot. Proper services are also offered to troubleshoot problems associated with any type of programming related to the a

3.7. IP Camera

IP cameras are widely popular in video data gathering and processing. They connect directly to WIFI networks and provide for two-way audio communication. Some IP cameras work with the PoE-protocol which is Power over Ethernet protocol.

The IP camera utilized was the Sricam Wireless IP camera.

Figure 3.16: Sricam IP Camera
[http://i.ebayimg.com/00/s/MTAwMFgxMDAw/z/YpEAAOSwYGFUrJO4/$_57.JPG
]

Features of IP camera:

- MJPEG, CMOS 300k Pixel.
- 10 LED, IR distance 10M. Lens 3.6 mm.
- Two-way audio,Indoor use.
- Wi-Fi/802.11/b/g/n, DC 5 V 2 A.
- Motion detection, email alert, FTP upload.
- Phone View, support Android and iOS Smart Phone.
- Pan:355°, Tilt: 90°, Speed:5-50°/S
- P2P: Plug & play, supports view the camera via P2P ID on the Phone App and PC Client CMS software directly.
- PC client "DeviceViewer", supports centralize monitor, remote record , playback .

3.8. Wi-Fi module

The Wi-Fi module is an essential component in the project in the sense that it allows the microcontroller to communicate with the webserver such that data can be sent and received with ease. The mobile phone will need to gather information from the microcontroller which in turn requires command signals from the smartphone.

3.8.1. Adafruit CC3000 WIFI breakout module

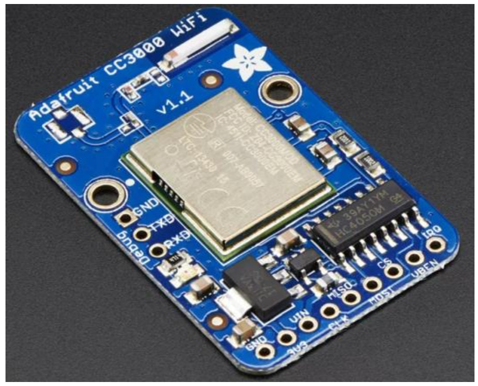
Satisfying the criteria of usability, price and capability, the Adafruit CC3000 is suitable option for this application. Additionally, it is compatible with the Arduino microcontroller through downloadable Adafruit libraries. With its use of Serial Peripheral Interface (SPI) for communication, data can be pushed as fast or as slow as you want. Asynchronous connections are possible with a proper interrupt system on the board with the IRQ pin. It supports 802.11b/g, open/WEP/WPA/WPA2 security. It does not support "AP" mode, it can connect to an access point but it cannot be an access point.

It has an onboard 3.3V regulator that can handle the 350mA peak current and a level shifter to allow 3 or 5V logic level. Even though the WIFI breakout module has an onboard antenna, its range is as good as a smartphone’s.

Technical parameters:

Dimensions : 26.22 mm x 40.45 mm x 2.95 mm

Weight : 3.46 g



2Figure 3.17: Adafruit Wi-Fi Breakout Module

[http://www.adafruit.com/product/1469]

3.8.2. Arduino Wi-Fi Shield

With the Arduino Wi-Fi Shield the Arduino can be connected to the internet wirelessly through a Wi-Fi router using the 802.11 wireless specification. Some of the features of the shield is given below:

- Operating voltage: 5 V (supplied from the Arduino Board)
- Connection via: 802.11 b/g networks
- Encryption types: WEP and WPA2 Personal
- Connection with Arduino on SPI port
- on-board micro SD slot
- ICSP headers
- Mini-USB for updating Wi-Fi shield firmware

An onboard micro-SD card slot can be used to store files for serving over the network. Furthermore, the shield is compatible with the Arduino Uno and Mega. Figure 3.18 shows an Arduino Wi-Fi shield which contains a micro-SD slot.

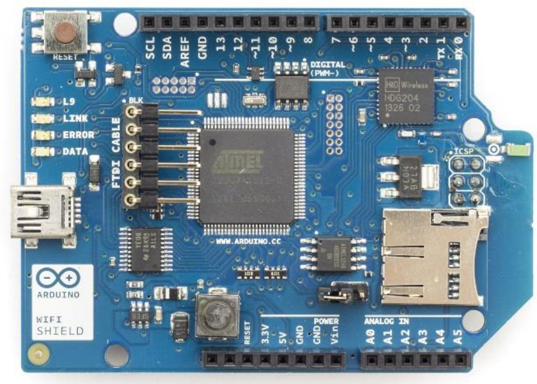


Figure 3.18: Arduino Wi-Fi Shield

[http://arduino.cc/en/uploads/Main/A000058_front.jpg]

3.8.3. Summary

Taking into consideration the Adafruit CC3000 Wi-Fi breakout module and the Arduino Wi-Fi shield, the former has been chosen due to its cost-effectiveness, availability and ability to be incorporated with the microcontroller. In addition to that, its library is available and can be downloaded.

3.9. Reflectance sensor

Most line following robots follow a path on a black tape on a white platform whereby a reflectance sensor is used to note the difference in colours. In the reflectance sensor, an infrared led and a photo-transistor pair is present. The latter is connected to a capacitor whose discharging time gives the data about the reflector. With high discharging of the capacitor, it can be deduced that the sensor is on a high reflection area. When the sensor is on a black area, the infrared light will not reflect to the photo-transistor. The detection distance of a reflectance sensor is usually in the range of 2mm to 9mm.

3.9.1. CNY-70 Reflectance sensor

Owing to the fact that it is cheap and readily available on the market, the CNY-70 reflectance sensor as shown in figure 3.19 is widely used. However, it gives only analog data of 0 and 1 as output and further circuit components needs to be connected to it for use.

Features:

Operating voltage : 5 V DC

Supply current : 30 mA (max)

Figure 3.19: CNY-70 Reflectance sensor

[http://1.bp.blogspot.com/_NK_fTn1bYR0/S7KAKq5GEXI/AAAAAAAAAa8/T5sE_0GcAro/s1600/cny-70.jpg]

3.9.2. Pololu QTR-8RC Reflectance Sensor Array

This sensor array shown in figure 3.20 has 8 Infrared LED/phototransistor pairs and a MOSFET can allow LEDs to turn off in order to save power. The purpose of the QTR-8RC reflectance sensor array is to be mainly used in a line following robot.

Specifications:

- Dimensions: 74.93 mm x 12.7 mm x 3.18 mm (without header pins installed)
- Operating voltage: 3.3-5.0 V
- Supply current: 100 mA

- Output format: 8 digital I/O-compatible signals that can be read as a timed high pulse
- Optimal sensing distance: 3 mm
- Maximum recommended sensing distance: 9.5 mm
- Weight without header pins: 3.09 g

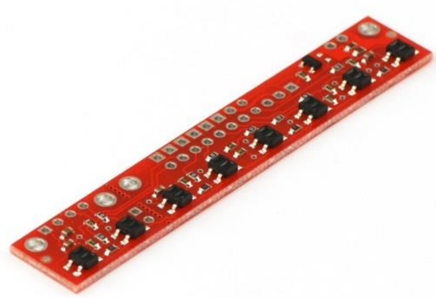


Figure 3.20: Pololu QTR-8RC Reflectance Sensor Array

[https://www.pololu.com/picture/view/0J615]

3.9.3. Summary

After checking the characteristics reflectance sensors, it can be deduced that the Pololu QTR-8RC reflectance sensor array is the best option to implement in the line following system even though of its relatively high cost. However, it worth the price as it is beneficial in terms of precision and effectiveness.

3.10. WiFi Router/Access Point/Repeater

The Wi-Fi Repeater function is to be used to connect to the IP camera. The functions of the Wi-Fi Network Bridge (figure 3.21) are listed below :

- Switching among wireless network router, wireless AP, Bridge, extender, wifi repeater, etc. up to 7 kinds of function.
- Support IEEE 802.11 b/g/n 2.4GHz.
- Advanced wireless-N technology delivers exceptional speed up to 150Mbps.
- One-button WPS encryption, easy to operate.

- Plug and play, instant access to wireless Internet
- With advanced features such as QOS management, MAC address filtering, web page filtering and NAT address switch.
- USB powered, it can be charged directly from laptop
- High-integrate and multifunctional designs, compact body, easy to carry

Figure 3.21 : Wireless Network Bridge

[http://i.ebayimg.com/images/g/gkAAAOSwYshUZc31/s-l1600/r.jpg]

More details about the setting up of the WiFi module is given in Appendix A.

Chapter 4: Mechanical Design

4.1. Introduction

The mechanical design of the robot is carried out based on the latter's stability, mobility as well as the aesthetics so that the robot is user friendly and is attractive to the persons using it. In figure 4.1 below, an overview of the different components in the mechanical design system is demonstrated.

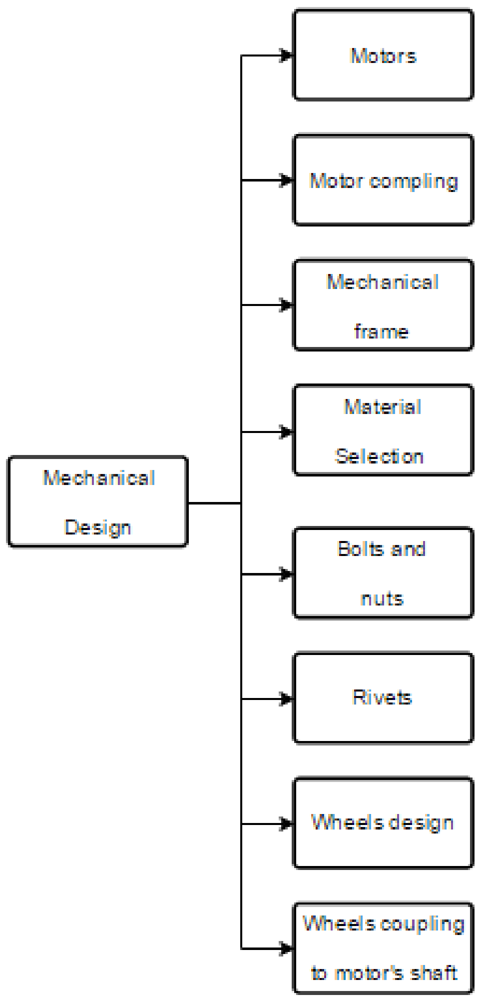


Figure 4.1: Overview of mechanical system

4.2. Prototype design

The figure 4.2 below shows the model of the robot designed in 3D on Autocad.

The prototype is a 235 mm × 235 mm × 460 mm robot. Further precise dimensioning is given in the Appendix B.

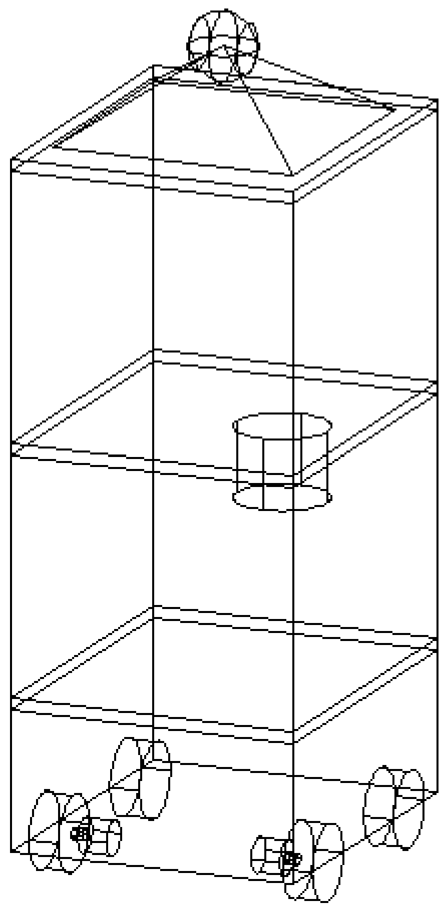


Figure 4.2: Front side view of prototype of robot

4.2. Selection of material for robot

The frame is the most important structure in the robot as it holds together all the different parts and components present in the robot together which includes the drive system, the electrical components, and the sensors, amongst others.

Requirements for selection of material for robot frame:

- Cost effectiveness
- Corrosion resistance
- Strength
- Machinability
- Lightness
- Recyclability
- Texture
- Availability
- Durability

Materials under consideration also represented in Figures 4.3, 4.4, 4.5, 4.6 are:

- Aluminium

Figure 4.3: Aluminium sheet [https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQfHS8zazhcnTF2gbuxD3oRmYNwig4tF1Re8_2eAPqa9V2FMBCDeA]

- Stainless steel

Figure 4.4: Stainless steel metal sheets

[http://aemforge.com/wp-content/uploads/2012/12/stainless-steel_plate1.jpg]

- Alucobond

Figure 4.5: Different colours of aluminium sheets

[http://i01.i.aliimg.com/img/pb/855/399/459/459399855_309.jpg]

- Wood

Figure 4.6: Wooden bar

[https://inspiredtelemarketing.files.wordpress.com/2014/09/483115_large.jpg]

Being the most widely used non-ferrous metal, aluminium has a vast range of advantages. However, stainless steel is an alloy with numerous benefits also. Alucobond, a composite of aluminium and polyethylene, is being taken into consideration too for the construction of the robot. A list of advantages and disadvantages for these three materials is given below.

Table 4.1: Advantages and disadvantages of different materials

	Advantages	Disadvantages
Aluminium	• Lightweight • Very high strength to weight ratio • Durable • Easy to work with (easily welded, riveted or bolted) • Resistant to corrosion • Reflective and attractive finish	• It oxidizes which then appears as a white residue • Expensive
Stainless Steel	• Highly resistant to corrosion • Heat resistant • Hygienic non-porous surface • Relatively easy to cut, machined and welded	• High cost • Tends to scratch and dent • Relatively heavy compared to aluminium
Alucobond	• Excellent vibration dampening characteristics • Easily folded and bent with conventional tools • Low weight	• Barely any disadvantage

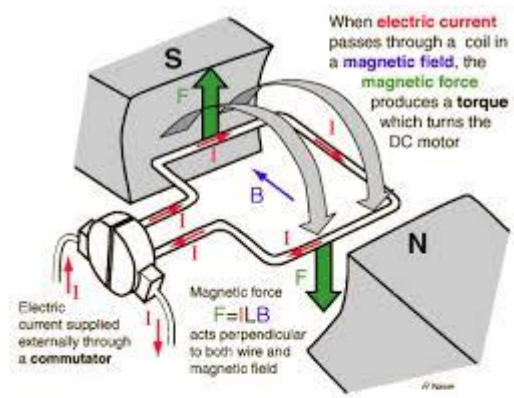
	• High rigidity • Recyclable • Shock resistant	
Wood	• Low cost • Environment friendly material • Low density, therefore light • Low thermal conductivity • Low electrical conductivity	• Strength decreases with water • Highly combustible • Susceptible to rot

The deduction obtained from the above table is that alucobond also is a proper alternative to using aluminium or stainless steel. Furthermore, stainless steel is heavier than aluminium or alucobond. The latter will be used to mount the surfaces of the robot whilst aluminium will be utilized as angle bars to support the alucobond plates. Even being lighter, cheaper and easily available, wood is not a suitable material for the construction of the robot due to its significant disadvantages that it demonstrates.

4.3. Selection of motor

4.3.1. DC Motors

DC (Direct Current) motors consist of two wires which provide current to the coil of the motors for continuous rotation of the motors. The speed of DC motors is controlled through the use of Pulse Width Modulation (PWM). The time period during which the power signal is on determines the speed of the motor. Figure 4.7 illustrates the working principle of a DC motor while Figure 4.8 shows a DC motor.



3Figure 4.7: Working principle of DC Motor
[http://t3.gstatic.com/images?q=tbn:ANd9GcQDqcdMPcbwEGdBtD93oWx_mcC7B87IeNr8XYHhbCSJjdtkj5Pkzg]

Figure 4.8: DC motor
[http://t1.gstatic.com/images?q=tbn:ANd9GcSyJJiUVTn1NQTenUAAkGZKF8_g9W0ThKJG3jDbmbAk9AvBDZ9HtA]

4.3.2. Servo Motors

Servo motors generally comprise of four components namely a DC motor, a gearing set, a position-sensor and a control circuit.

Position can be controlled more precisely with servo motors than with standard DC motors. They compose of three wires namely the power, control and ground wires and power to the motors is continuously applied. Servo motors have been designed for specific tasks where accurate position defining is necessary.

However, the rotation of servo motors is restricted unlike a standard DC motor as the angle through which they can rotate is 180 degrees. Determination of the correct position of the shaft of the servo motor is obtained from the control signal.

The control signal of the servo motors requires the PWM technique. However, the time for which the positive pulse lasts determines the position of the shaft rather than the speed as compared to standard DC motors. A servo motor when given a control signal will move to that position and maintain it despite the possible presence of an external force acting against it. Figure 4.9 shows the outlook of a servo motor.

Figure 4.9: Servo motor

[http://www.robotshop.com/en/hitec-hs-5585mh-servo-motor.html]

4.3.3. Stepper Motors

A stepper motor (figure 4.10) makes use of several electromagnets uses multiple toothed electromagnets set up around a central core to define position. A motor driver or microcontroller is necessitated to energize each electromagnet of the stepper motor for the motor shaft to turn.

With each rotation from one electromagnet to the next, the motor performs steps thus allowing rotation by steps through a 360 degree rotation.

Two types of stepper motors are the unipolar and bipolar ones. Since stepper motors have pre-defined locations of their magnets through which they move, positioning errors with them does not occur.

Figure 4.10: Stepper motor

[http://www.automationtechnologiesinc.com/wp-content/uploads/2011/11/KL23H286-20-8B-500x500.jpg]

4.3.4. Summary

To summarize the above analysis of the different motors, the DC motors are fast and continuous motors whilst the servo motors are fast high torque motors with a limited angle of rotation of its shaft and makes use of PWM. On the other hand, stepper motors are slow but allow precise rotation as well as having an easy set up and control. They have positional control through their incremental rotation and have an advantage over servos which require a feedback mechanism and support circuitry to drive positioning.

The drive system that the robot will make use of is the differential steering. Considering this aspect, we will have recourse to two DC motors to drive the robot. The rating of the DC motors used will be based basically on the motor torque calculation which is dependent on several factors including the size and mass of the robot.

4.4. Sizing of robot

The dimensions of the robot frame is 235 mm × 235 mm × 360 mm.

The thickness of the alucobond is 3mm. The volume of an alucobond sheet is given by the following equation:

Volume = Area of an alucobond plane x Thickness of alucobond

Total volume of alucobond = total surface area of the planes x thickness

$= (360\times235\times3\times3) + (235\times235\times4\times3) + (220\times235\times3)$

$= 1579200\ mm^3$.

The support of the alucobond sheets to form the structure of the robot was provided by aluminium angle plates having a thickness of 1mm and a width of 15mm.

Total volume of aluminium angle plates = total surface area of the angle plates x thickness $= (4\times2\times15\times360\times1) + (11\times2\times15\times235\times1)$

$= 120750\ mm^3$.

The density of alucobond is $1041 kg/m^3$ whilst that of the aluminium is $2600\ kg/m^3$.

Mass of an object can be calculated by the following equation.

Mass = Density x Volume

Mass of the alucobond used = $1579200 \times 10^{-9} \times 1041 = 1.6$ kg.

Mass of the aluminium angle plates used = $120750 \times 10^{-9} \times 2600 = 0.3$ kg.

Mass of the two motors = 0.096 × 2 = 0.192 kg.

Table 4.2: Robot components and mass	
Components	Mass/kg
Sheets of alucobond	1.6
Aluminium angle plates	0.3
2 DC Motors	0.192
Electrical Circuits	0.2
Other objects	1
Total	3.292

4.5. Wheel design

4.5.1. Wheel diameter

When designing wheels, motor parameters, which are torque and velocity, should also be taken into consideration. With large diameter of wheels, the robot will have low torque but high velocity. Therefore, wheels with larger diameters will be appropriate with a strong motor. However, if the motor is weak, smaller diameter wheels would be suitable. Additionally, a wheel with diameter close to or less than the motor diameter is poor design as only a small ground clearance will be available. The wheel diameter designed for the robot is 65 mm.

4.5.2. Wheel texture and width

The wheel texture is very terrain dependent and is a factor which should not be ignored. If the wheel is too smooth then it will have little friction, causing the robot to slide while accelerating or braking. However, a really rough wheel will have higher friction leading to inefficiency.

With too wide wheels, there will be increased resistance while rotating the wheel on a surface. The width of the wheel for the robot is 23 mm.

The wheels in Figure 4.11 depicts the ones used in the mounting of the robot.

Figure 4.11: Robot wheels

[http://i.ebayimg.com/00/s/MTAyNFg5OTQ=/z/E-wAAOSwAL9Uh7HN/$_12.JPG]

4.6. Motor Torque Calculation

Motor toque calculation is an essential aspect that must be taken into account when mounting a robot. The motor should have enough torque in order to provide enough force for the robot to move. The motor torque calculation is given in the Appendix C. Following the analysis for the motor torque calculation, the motor in figure 4.12 has been chosen because of the designed parameters and availability on the market.

Figure 4.12: DC motor for robot [http://i.ebayimg.com/00/s/NTMzWDgwMA==/z/n0oAAMXQ56ZSBt1P/$T2eC16d HJIQFHHTr)cLEBSBt1OwC,!~~60_57.JPG]

Motor specifications:

- Torque: 1.3 Nm
- Voltage rating: 12 V DC
- Revolution: 60 RPM
- Diameter: 25 mm
- Length: 72 mm
- Shaft diameter: 4 mm
- Weight: 96 g

Even though the torque of the motor is 1.3 Nm which is somehow higher than the designed torque, the robot will navigate properly and the speed of the motor can be controlled to prevent too fast movements, thus maintaining the stability of the system.

4.7. Drilling of alucobond

The drill tool of size 4 mm in diameter was used to drill holes in the alucobond and aluminium angle plates such that rivets and bolts can be inserted to hold the system tightly. A cordless drill machine with rechargeable batteries as shown in figure 4.13 was used due to availability of the machine. Additionally, the use of the cordless drill facilitates the machining of the materials as it is more practical and easy to carry as well as it provides the necessary and required torque to drill the materials.

4Figure 4.13: Metabo cordless drill

[https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcQws0j2FDpk4DnPVi3QNADOjJocLK5mVSG2qGBEkJp1mGjZ0_SW]

4.8. Fastening of prototype structure

Aluminium blind rivets as shown in figure 4.14 were used to fix the alucobond sheet and aluminium angle plates together such that a strong support is provided to the frame and structure of the prototype. Button head screws were also made use of to maintain a proper frame for the robot. However, the screwed alucobond sheets are meant to be removed to compensate for the maintenance of the system as represented in figure 4.15 below as electrical components will be placed inside the robot on the first shelf. Both the rivets and the screws were of diameter 4 mm.

Figure 4.14: Blind rivet (left) and button head screw (right) [

[https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcTLo6mfN9xZqnQJkST9_lU7RcGQNV53-9hp4cJBXXr1UZjpk4-FSw]

[https://encrypted-tbn3.gstatic.com/images?q=tbn:ANd9GcQOEcT45C_9S_OI4ySB-vOATQ_61WCWhUE1aDuIbladp3Ok9Lzc]

Figure 4.15: Front view of prototype with front cover removed

4.9. Prototype

Figure 4.16 shows a back view of the prototype whereby the aluminium angles have been riveted to the alucobond sheet.

Figure 4.16: Back view of prototype

In figure 4.17, the motor is fixed to the alucobond in such a way to provide adequate strong support to minimize vibrations.

Figure 4.17: Motor fixed on robot

Figure 4.18 below shows the motor shaft coupling which connects the motor shaft and the wheel.

Figure 4.18: Motor shaft coupling

In figure 4.19, the front view of the prototype is displayed.

Figure 4.19: Front view of prototype

Chapter 5: Electrical and Electronics Design

5.1. Introduction

The electrical and electronics design of the robot will be discussed in this chapter. The connection of the different components will be illustrated. Figure 5.1 demonstrates an overview of the electrical and electronics design implemented in this project.

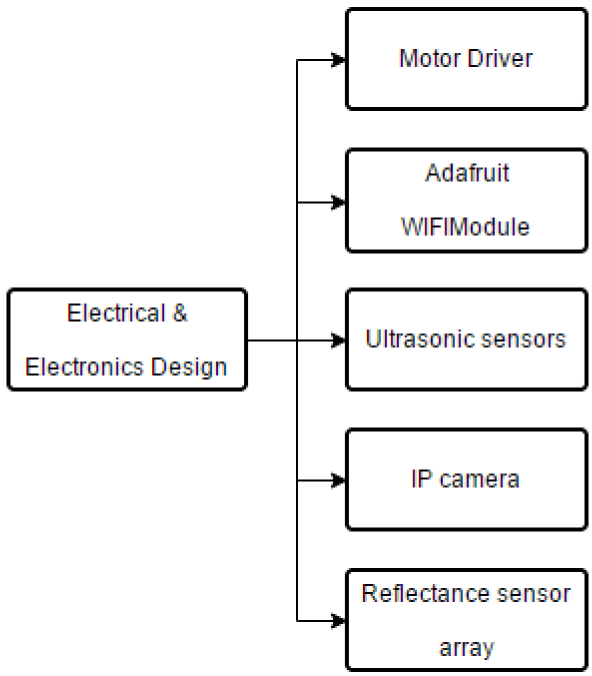


Figure 5.1: Overview of electrical and electronics system

5.2. DC Motor Driver

The requirement of motors in the project has prompted the need to include a motor driver. The driver chosen is the L298N Motor Driver (figure 5.2) which has the ability to drive one 2-phase stepper motor, one 4-phase stepper motor or two DC motors. To complement these characteristics, it has a large capacity filter capacitance, afterflow protection diode and is more stable and reliable.

Specifications:

- Double H bridge drive
- Chip: L298N

- Logical voltage: 5 V
- Drive voltage: 5V - 35 V
- Logical current: 2 A (max. single bridge)
- Maximum power: 25 W
- Size: 43 mm x 43 mm x 26 mm
- Net weight: 26 g

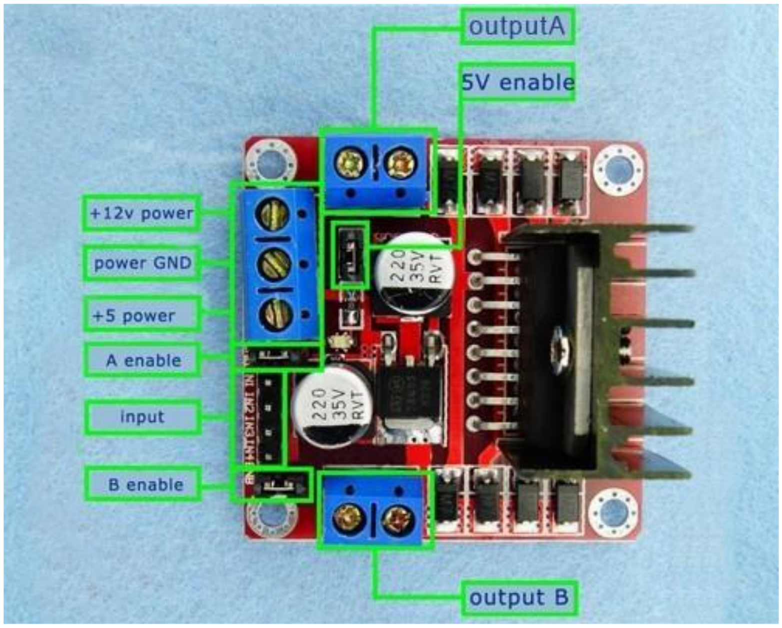


Figure 5.2: L298N Motor driver

[https://encrypted-tbn0.gstatic.com/images?q=tbn:ANd9GcThjcido1F7msaRC-6jlBSMmMoh4eY8m-ezfUv3aRqbEyGdmlmjsQ]

5.3. Selection of microcontroller

After having chosen the Arduino microcontroller for the development of the robot due to its ease of programming and availability on the market, the specific type of microcontroller to implement the robot’s system needs to be taken into consideration.

Following the fact that the robot will consist of two different processes, namely the autonomous and the semi-automatic ones, the Arduino Nano is best suited for the first mentioned process while the Arduino UNO is more appropriate for the other part of the project. The justification for choosing these specific microcontrollers for these processes is based on several aspects among which the prime reason is the amount of memory they possess as well as the number of pins they possess. The ATmega Arduino Nano has a flash memory of 32 kB while the Arduino UNO has also a flash memory of 32 kB.

5.4. Connection of components

5.4.1. Adafruit CC3000 breakout module with Arduino UNO microcontroller

The Figure 5.3 below shows the connection of the Adafruit CC3000 breakout module with the Arduino UNO microcontroller.

The GND pin of the module is connected to one of the Arduino GND pins and the Vin to Arduino +5V. Following this set up, VBEN is connected to Digital Pin 5, IRQ to Digital Pin 3. In addition to that the Serial Peripheral Interface (SPI) should be connected with CLK connected to Digital Pin 13, MISO to Digital Pin 12, MOSI to Digital Pin 11 and CS to Digital Pin 10.

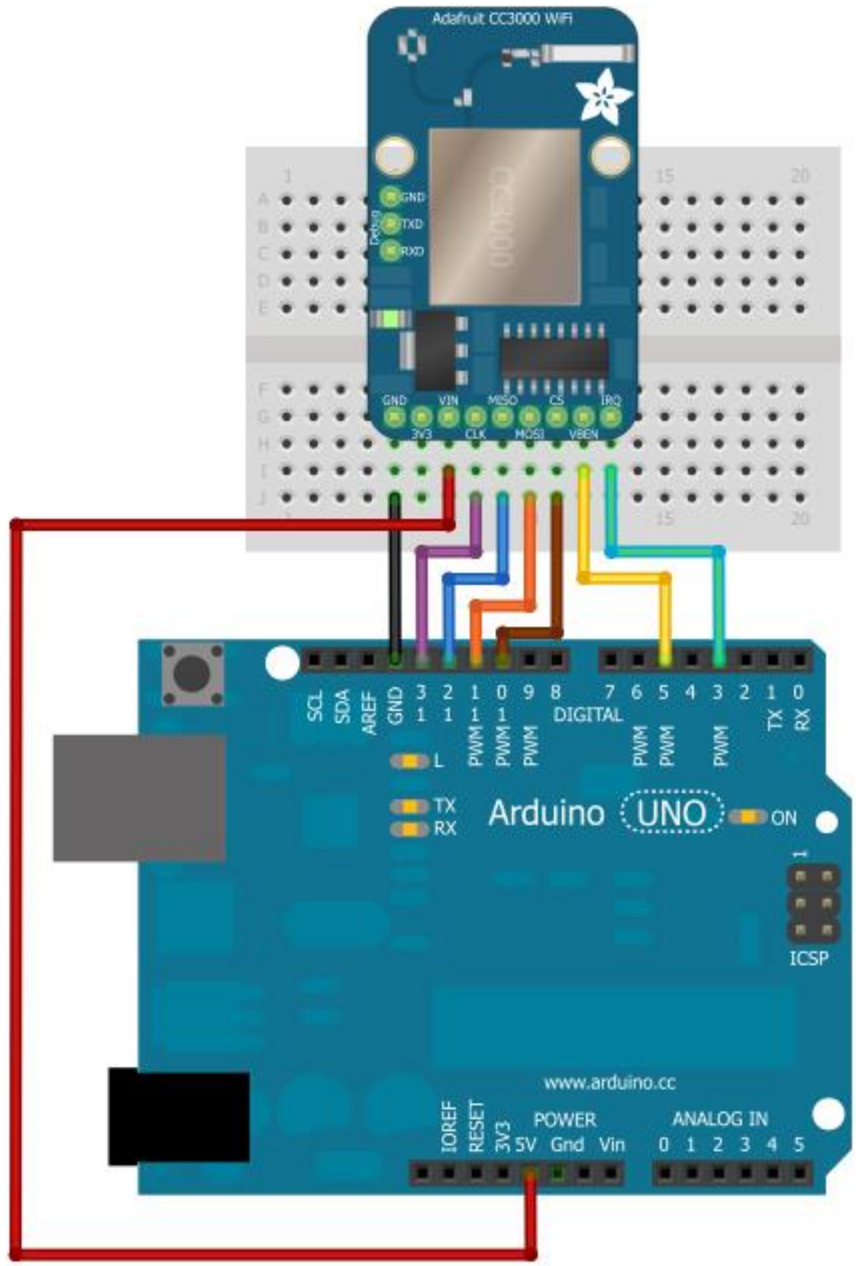


Figure 5.3: Connection of Adafruit CC3000 breakout module with Arduino UNO

5.4.2. Motor drive with Arduino UNO microcontroller

The left motor is connected to output A of the motor driver while the right motor is connected to output B. Just like the right motor, the left motor is controlled by 2 pins, one of which controls its speed and the other one controls the direction of the motor.

The pins D4 and D7 of the microcontroller are both connected to separate inverters such that their output can be sent to pins IN 4 and IN 2 respectively.

Pins D6 and D9 are PWM pins which are each connected to EN A and EN B of the motor driver. These connections can be viewed from figure 5.4.

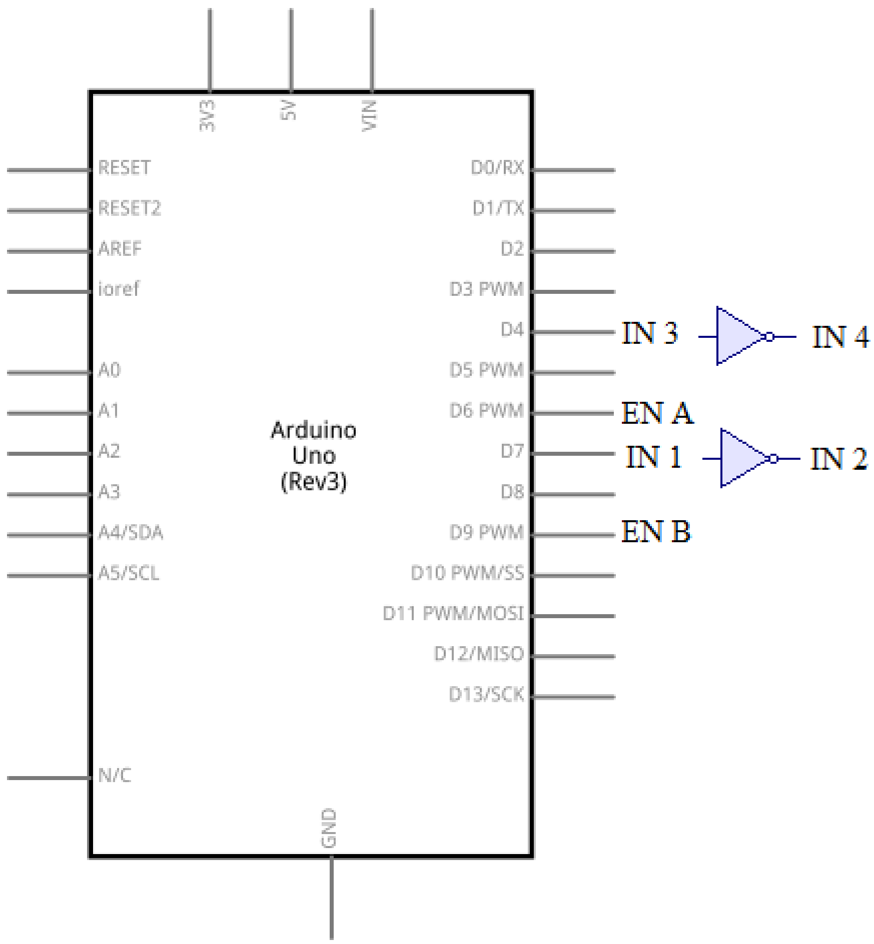


Figure 5.4: Motor drive connection with Arduino UNO

5.4.3. Motor driver and with Arduino Nano microcontroller

Being PWM pins, pins D3 and D11 are connected to EN A and EN B respectively. D9 and D1 are connected to IN 1 and IN 3 of the motor driver as shown in figure 5.5. Two inverters are also used to connect to pins IN 2 and IN 4 of the L298N motor driver and illustrated.

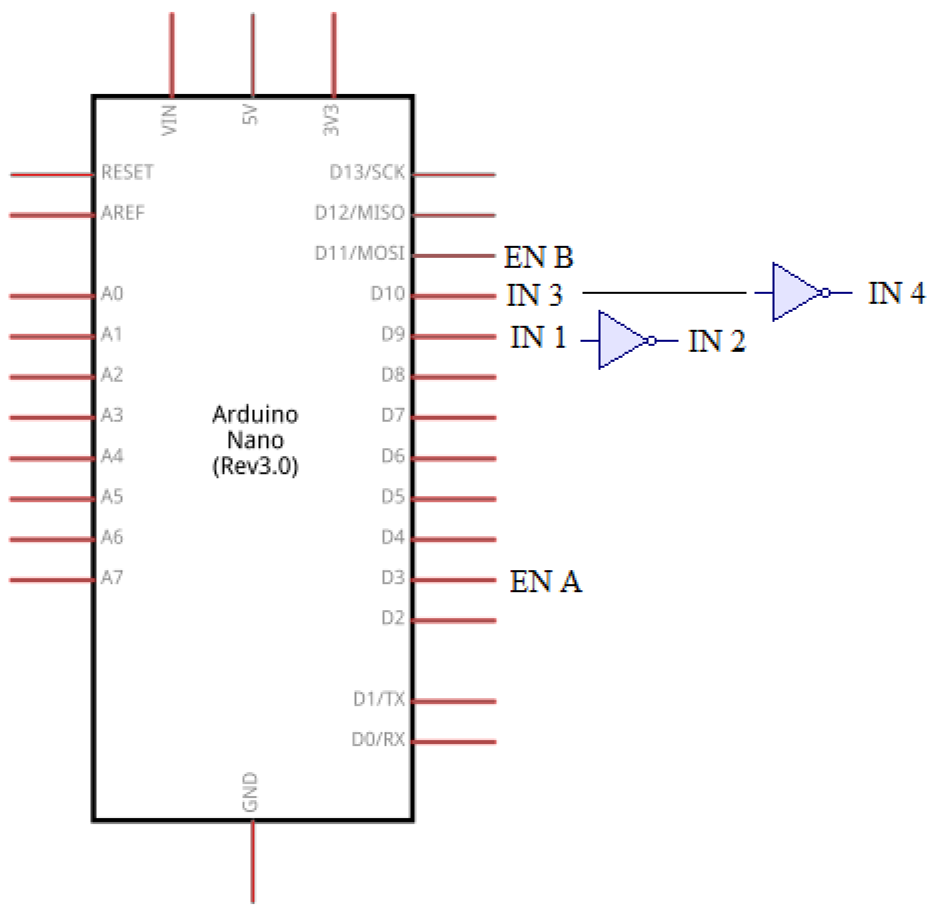


Figure 5.5: Motor driver connection with Arduino Nano

5.4.4. Ultrasonic sensors and Arduino Nano microcontroller

The triggering pin of the ultrasonic sensor is wired to the pin D13 of the Arduino Nano whilst its echo pin is connected to D12. This is illustrated in figure 5.6 below.

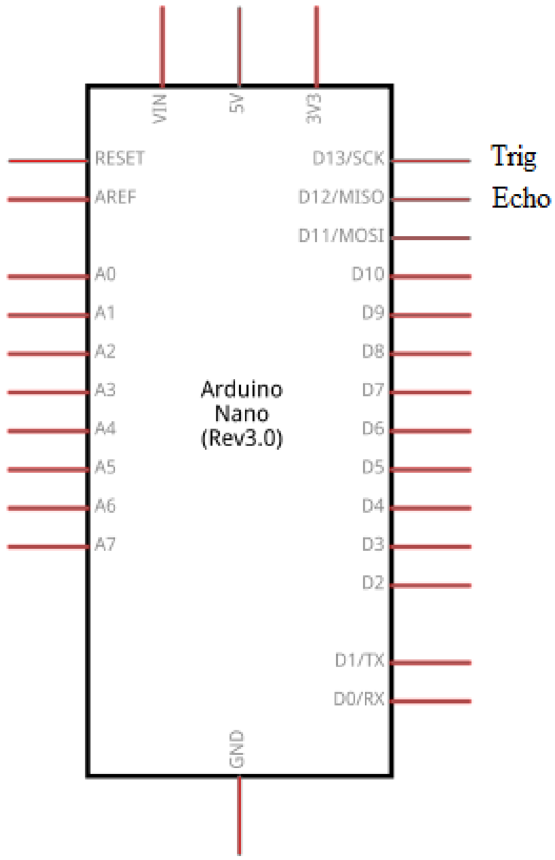
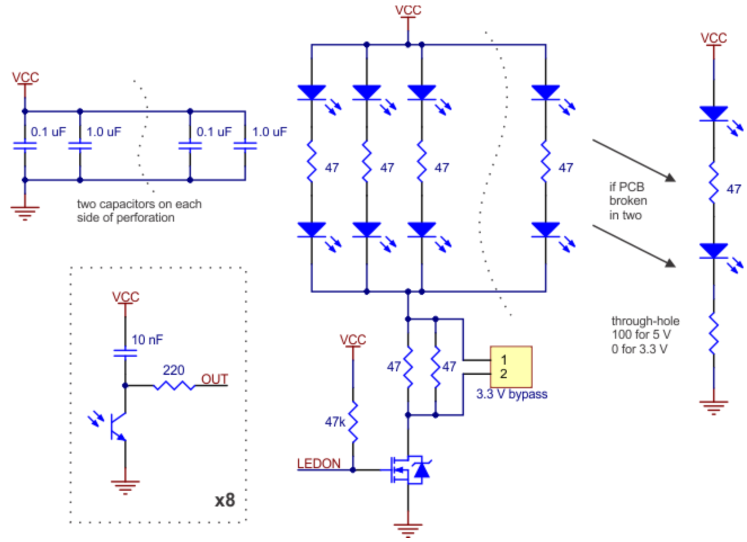


Figure 5.6: Ultrasonic sensor connection with Arduino Nano

5.4.5. Reflectance sensor and Arduino Nano microcontroller

The QTR-8RC reflectance sensor has infrared emitters and phototransistors which are separated equally by 9.525 mm. The amount of time for the discharging of the capacitor, which is shown in the schematic diagram in the figure below, indicates the degree of reflection that has occurred. A low reflection gives rise to a longer discharging time of the capacitor.



5Figure 5.7: Electrical connection of reflectance sensors in the array [https://a.pololu-files.com/picture/0J629.650.png?3eefac30beca66e1919bcfe3dfa78bf3]

6 sensors (sensors 1 to 6) are used in the project and connected to pins 2, 4, 5, 6, 7 and 8 respectively as shown in the figure below.

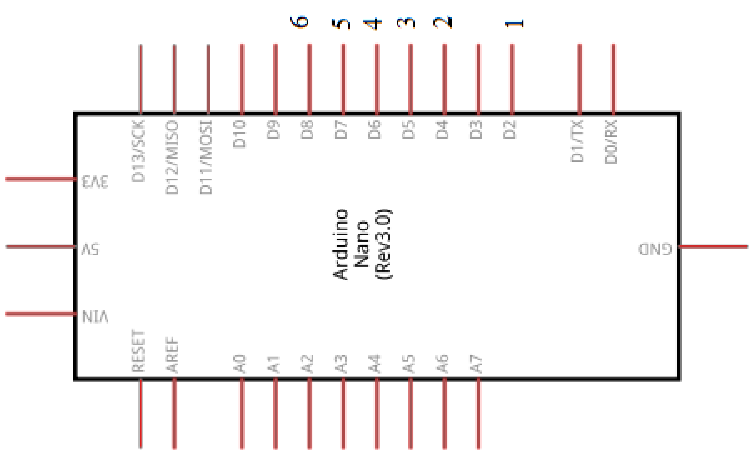


Figure 5.8: Reflectance sensor connection with Arduino Nano

Chapter 6: Software Design

6.1. Introduction

This chapter deals with the software design of the project which consists of two parts (as shown in figure 6.1):

(i) The semi-automatic system
(ii) The automatic system

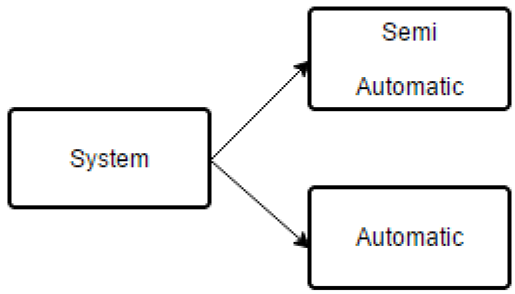


Figure 6.1: Overview of software design system

The different softwares that were made use of will be discussed as well as some working principles involving the Wi-Fi network.

6.2. Arduino IDE Software

The Arduino microcontroller chosen requires an interfacing with a personal computer so as to upload the written program to the microcontroller. To provide this interface, an open-source Arduino Software (figure 6.2) is available to make code writing and uploading easy. The software is compatible with Windows, Mac OS X and Linux. Any Arduino board is compatible with this software. Furthermore, regular updates of the software are available on the official site of Arduino. The latest version of 1.6.1 of the Arduino IDE Software was used. Programming with this software necessitates knowledge in C++. The Adafruit Wi-Fi module that has been chosen to integrate the robot system necessitates the uploading of certain libraries on the Arduino software. Furthermore, the Pololu reflectance sensor array also has a library which should be uploaded. This process of uploading a library to the Arduino software is illustrated in the Appendix D.

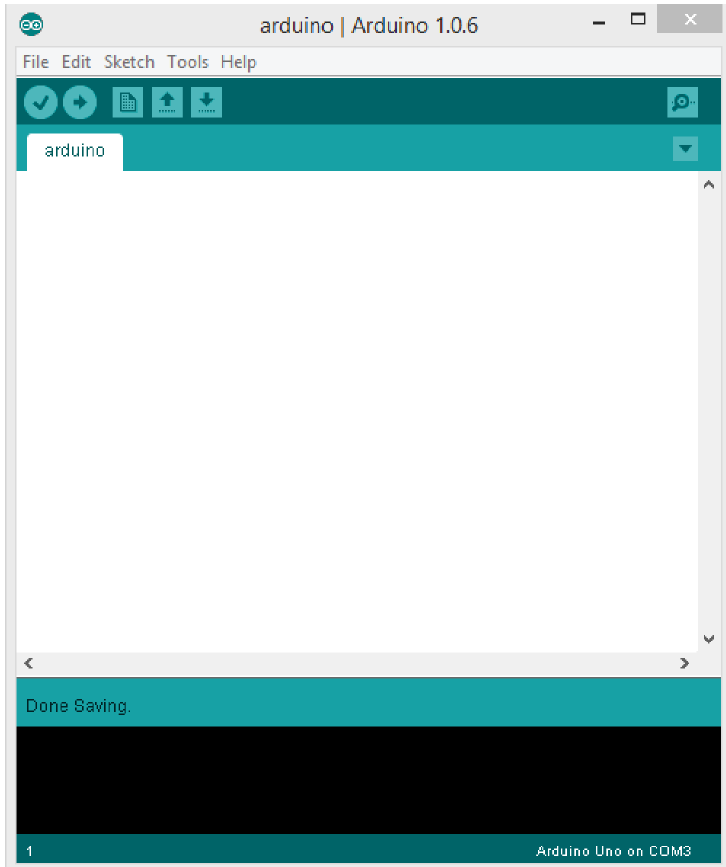


Figure 6.2: Arduino IDE software

6.3. Semi-automatic System

6.3.1. Flowchart

The process by which the semi-automatic system will function is illustrated in the flowchart below (figures 6.3 and 6.4).

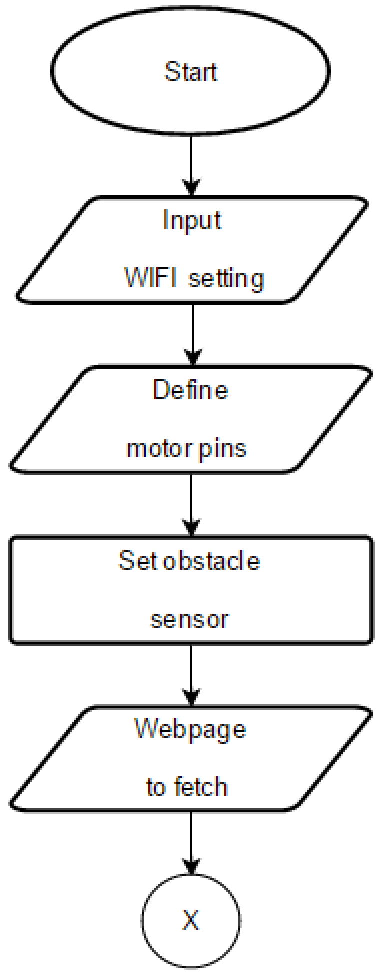


Figure 6.3: Semi-automatic system flowchart

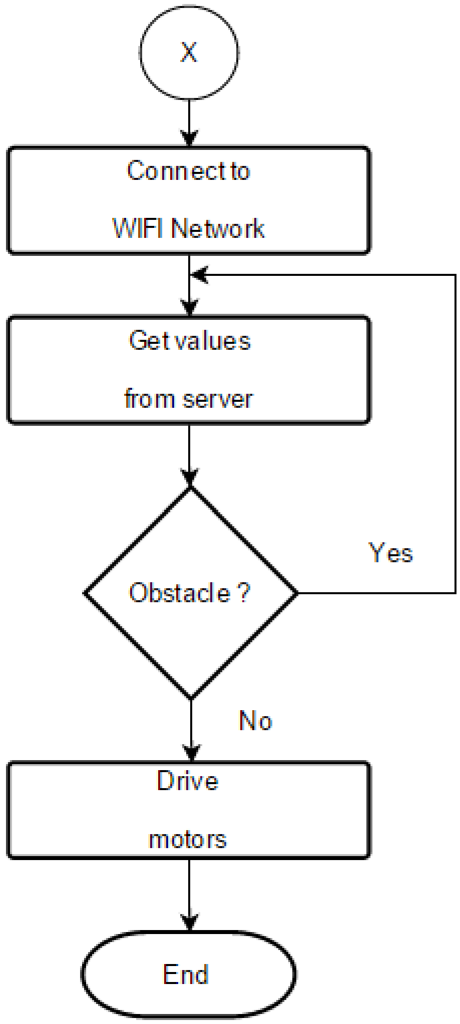


Figure 6.4: Semi-automatic system flowchart

6.3.2. WAMPServer

Being a local server package for Windows, WAMPServer allows the installation and hosting of web applications that use Apache, PHP and MySQL.

The functionalities of WAMPServer are complete and easy to use. Through this application, the user is able to:

- Manage Apache and MySQL services
- Switch from online and offline mode, that is, the option of allowing everyone to get access to files or only localhost
- Install and switch Apache, MySQL and PHP releases
- Manage servers settings

- Access logs
- Access settings files
- Create alias

After installation process of the application, files that need to be placed on the webserver should be inserted in “c:\wamp\www” directory which is automatically created by the application. In order to test whether the installation is working properly, http://localhost/ is entered in the web browser and a home page is displayed.

In case the WAMPServer homepage is not displayed, it should be checked that the hosts file has localhost mapped to 127.0.0.1. Moreover, other services such as another local server (XAMPP, DesktopServer), WebDAV or Skype, should not be running on port 80.

Verification of proper functioning of phpMyAdmin should be carried out by following the link http://localhost/phpmyadmin/ in the browser.

If Cannot connect: invalid settings error message appears, then, “C:\wamp\apps\phpmyadmin3.5.1\config.inc.php” file should be accessed in a plain text editor and the following option should be set to true:

$cfg['Servers'][$i]['AllowNoPassword'] = true;

Following the completion of all required steps, the files from the “c:\wamp\www” should be put online.

6.3.3. Notepad++

Notepad++ (figure 6.5) is a source code editor working in the Microsoft Windows environment. Contrary to Notepad, it supports tabbed editing which enables the opening of multiple files in a single window. Being able to support large files, Notepad++ is also faster than Notepad and Wordpad. In addition to that, it gives support to a variety of coding languages which ranges from C, C++, Java, amongst others.

Notepad++ has coloring text that provides information about the code written and it can distinguish and classify by color HTML tags, plain text, HTML attributes, and HTML comments thus making the code image clearer. Furthermore, missing characters can be easily identified and the numbers to all the lines become useful when PHP error occurs on a specific line number.

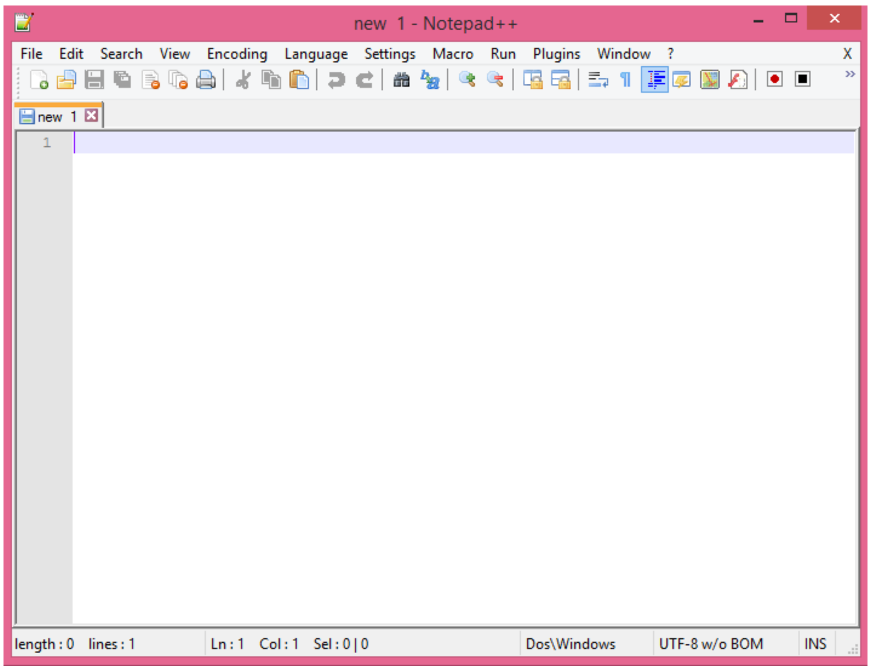


Figure 6.5: Notepad++

6.3.4. PHP

Also known as Hypertext Preprocessor, PHP is a programming language that is often used to in web development to create web pages. PHP code can be merged with HyperText Markup Language (HTML) code or used along with varying web frameworks. Normally, a PHP interpreter is used to process PHP codes. After the PHP code has been interpreted and executed, the web server sends the output to its client. Being fast and flexible, PHP allows the creation several kinds of websites.

6.3.5. JavaScript Object Notation (JSON)

JavaScript Object Notation (JSON) makes use of human-readable text to transmit data objects. It helps to store information in an organized way so that access to these details is easy. Its prime use is to transfer data between a server and a web application and provides an alternative to Extensible Markup Language (XML).

6.3.6. Port Forwarding

Each device connected to an internet network has an IP address through which communication to internet is possible. In the case of devices connected to a Wi-Fi router, the former have IP addresses linked to the router. The Wi-Fi router has an IP address which serves as reference for all the connected devices. However, the router also has an external IP address through which it can interact with devices outside the network. The devices connected to the Wi-Fi router each has a unique IP address and is illustrated in figure 6.6.

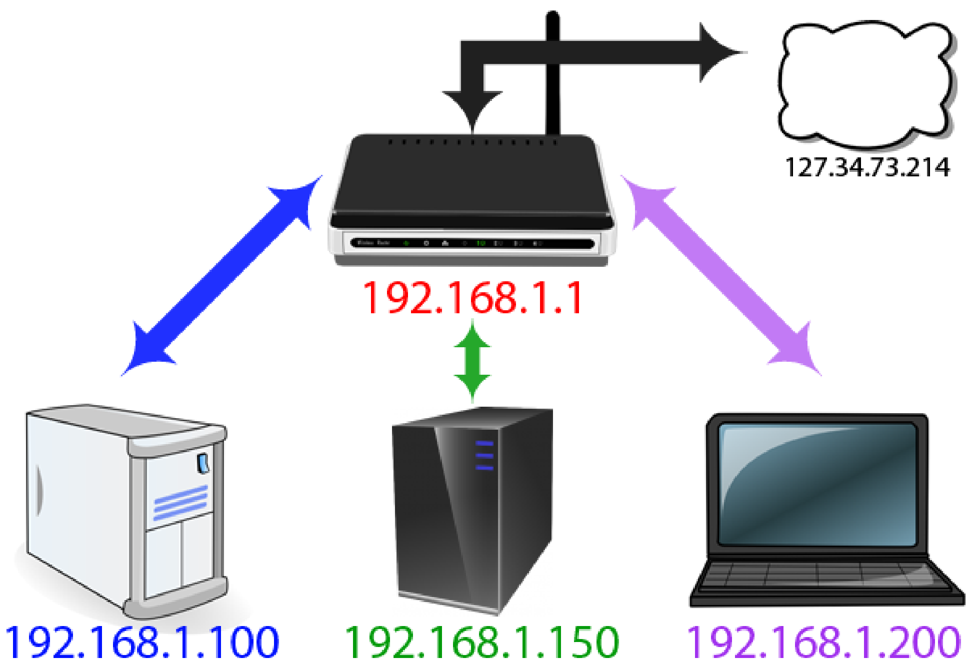


Figure 6.6: Port allocation

[http://cdn3.howtogeek.com/wp-content/uploads/2011/06/618x423xmap-intranet-fin-01.png.pagespeed.ic.vK8v5PwPG5.png]

Ports allow the communication between devices and each port makes use of Transmission Control Protocol (TCP) or User Datagram Protocol (UDP). Port 80 is mainly used to grab a web page.

Some important factors need to be taken into consideration before forwarding a port. If Dynamic Host Configuration Protocol (DHCP) is used, the IP address of each device connected to the network may vary resulting in changing the settings for port forwarding. Therefore, configuring a static IP address for a computer acting as a web server is useful. The setting up of the static IP address is given in the Appendix E.

Furthermore, when the network of the web server is accessed externally, the external IP address of the router needs to be known and can be obtained from www.whatismyip.com.

The Huawei HG8245 router forms part of the internet providing facility available for this project. The procedure to set up port forwarding on this router is referred to in the Appendix F.

6.3.7. WiFi Bot Control

WiFi Bot Control is an Android application which enables the remote control of a robot via Wi-Fi. A further option on the system is that it allows the viewing of a video stream from an IP camera mounted on the robot. The application provides up to eight customizable command buttons by which additional tasks can be performed on the robot. These commands can be used to initiate activities such as controlling sensors, servo motors, turning on/off a LED, amongst a list of other tasks. WiFi Bot Control can be downloaded on your smartphone on Google Play. Moreover, its compatibility with different versions of Android is remarkable and can be supported on most devices ranging from phones to tablets.

Features:

- Makes use of WiFi to control the robot and view the IP Camera video stream
- Configurable camera URLs (up to 3)

- 3 Joystick Modes: Default, Simple Mode and Orientation Sensor
- Supports up to 8 additional commands via Command Buttons.
- Configurable WiFi packet interval.
- Supports a number of microcontrollers having WiFi module/capabilities.
- Screen automatically re-sizes for smaller phones.

When operating in the default joystick mode, the values of the x coordinates and y coordinates are converted into a URL string which is transmitted to the web server in the form of a json file which is read by the robot from the Wi-Fi module and consequently the Arduino microcontroller.

In the simple mode of operation of the application, the directions of the joystick is given by four arrows with code U,D,L and R which represent Forward, Reverse, Left and Right respectively. Each direction command is converted into an URLudlr value and the URL string is sent to the webserver.

When commands are given to the robot through the Command Buttons, a value of 1-8 is saved to the URLcmdVal variable and the same communication process as above takes place.

In the WiFi Bot Control application on the smartphone, the URL, required to update values of the json file of the robot state, needs to input in the settings. The URL with the server name and the PHP file needs to be entered as in the example below:

http://192.168.100.2/updateStateNew.php?

Figure 6.7 below indicates where the URL has to be entered.

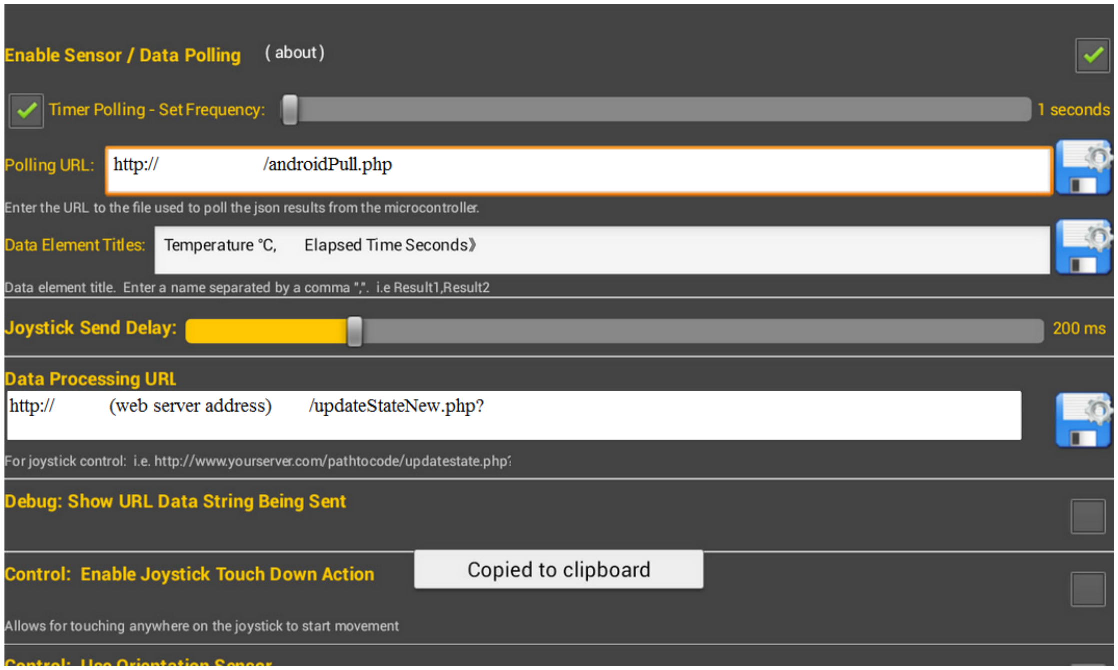


Figure 6.7: Wi-Fi Bot Control settings

The working principle behind the Wi-Fi Bot Control application is that it sends a URL request to a PHP file found on a server containing direction commands. As the file is requested, the robot constantly checks the server and calls a server.php file which parses the JSON file to obtain information about the data being sent from the application which are then used to run the motors and process other commands. This communication process is illustrated clearly in the figure below. An important aspect of the system that should be noted is that the computer, acting as a web server, should be continuously on such that the communication between the different devices can be maintained without any difficulty. The working principle is demonstrated in Figure 6.8.

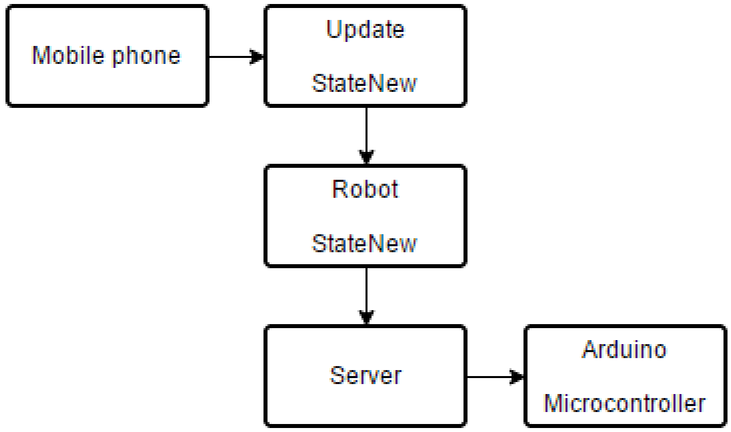


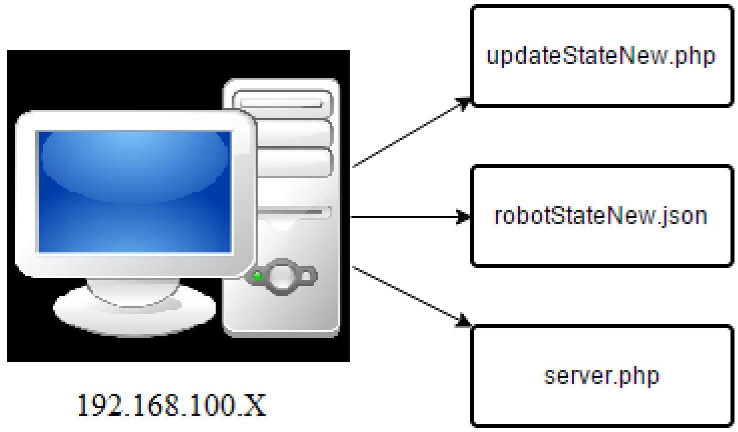
Figure 6.8: Working principle of Wi-Fi Bot Control system

6.3.8. IP Camera Software

The Sricam IP Camera is connected to a Wi-Fi Repeater which connects to the home Wi-Fi network such that the range of Wi-Fi access is expanded. When the robot has reached a place and the user wants to communicate with people in front of the robot, the person may switch from the Wi-Fi Bot Control application to another application known as the AP Camera App which can be downloaded on Google Play on the smartphone. The procedure to set up the IP camera in the application is provided in the Appendix G. Through this application, the IP camera can be rotated, videos can be recorded as well as a two-way communication can be established between the user of the mobile phone and the person in front of the camera. Additionally, the camera can be viewed via a web browser and this installation process is also explained in the Appendix G.

6.3.9. Programming

Figure 6.9 represents the web server and the files that it contains. From these files, the communication between the smartphone and the robot can be established. The programming in PHP and JSON of the files on the server can be obtained in the Appendix H.



6Figure 6.9: Files on web server

Furthermore, the Arduino has to be programmed to fetch data from the web server and drive the motors. The programming for this process is found in Appendix I.

6.4. Autonomous System

6.4.1. Flowchart

The process flowchart for the autonomous system is illustrated in figures 6.10 and 6.11 below.

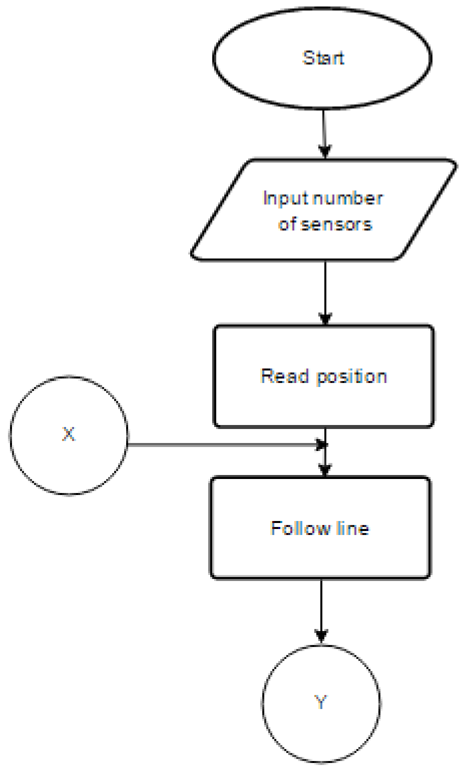


Figure 6.10: Autonomous system flowchart

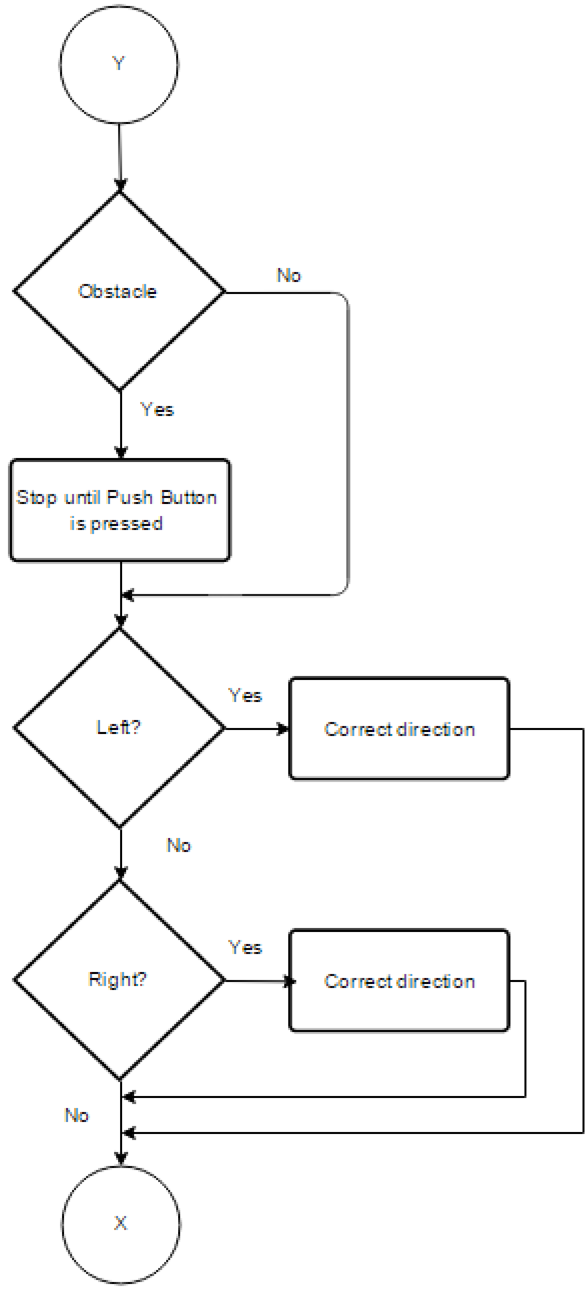


Figure 6.11: Autonomous system flowchart

6.4.2. Proportional Integral Derivative (PID) Control

A Proportional Integral Derivative controller (PID controller) is a feedback closed loop control system which determines the error which is the change between the process variable measured and a desired point that is set. The aim of this controller is to reduce the error by modifying the process by making use of a variable that can be changed.

The algorithm for the PID Controller involves three parameters that are constants. The first one is the proportional which depends on the current/present error, the second is the integral which is the accumulation of past errors and finally the third one is the derivative which is a prediction of future errors based on the actual rate of change.

The manipulated variable is also the output of the controller which is given by the sum of the proportional, integral and derivative terms. The equation for the output of the controller is given below.

$$MV(t) = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

where,

K_p = proportional gain

K_i = integral gain

K_d = derivative gain

e = error = Set point value – Process variable value

t = present time

τ = variable of integration taking values from time zero to present time t

The three constant parameters' effect on the output shall now be considered:

(i) Proportional term

The proportional term gives an output value which is proportional to the current error value. By multiplying the error by a constant $K_{p,}$ which is the proportional gain constant, the proportional response can be adjusted.

The proportional term, P is given by:

$P = K_p e(t)$

The graph in Figure 6.12 below gives the change in output when K_p is varied whilst K_i and K_d are held constant.

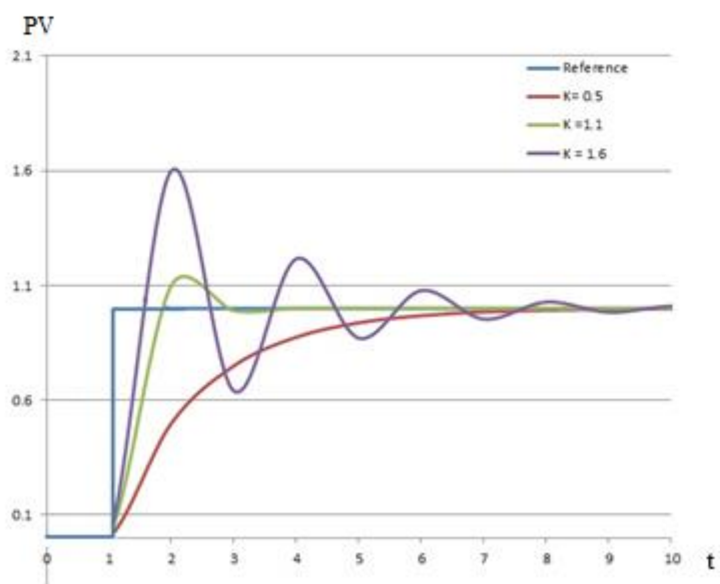


Figure 6.12: Graph of Process Variable (PV) against time t for different values of Kp

[http://upload.wikimedia.org/wikipedia/commons/a/a3/PID_varyingP.jpg]

As observed from the above graph, with a high proportional gain, the output experiences a large change for a given variation in the error. The system can become unstable with a proportional gain which is high. On the contrary, a small proportional gain results in an output with a small response to a large value of input error and therefore the controller is less responsive and less sensitive. With a low proportional gain, the control action may be too small when the system is subjected to disturbances.

(ii) Integral term

The integral term is directly related to the magnitude of the error and the time for which it lasts. Therefore, it is the sum of instantaneous error over time which gives the accumulated offset which was required to be corrected beforehand. The accumulated error when multiplied by the integral gain, Ki gives the following equation:

$$I = K_i \int_0^t e(\tau)d\tau$$

The graph in Figure 6.13 below shows the change in output when K_i is varied with K_p and K_d held constant.

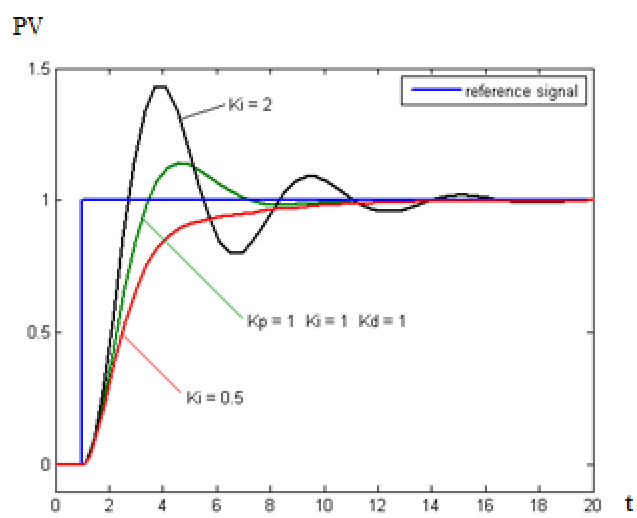


Figure 6.13: Graph of Process Variable (PV) against time t for different values of Ki

With the integral term, the flow of the process to reach the set point is accelerated and the remaining steady state error that occurs with a pure proportional controller is eliminated. However, as the integral term reacts to the errors accumulated from the past, the present value can make the set point value overshoot.

(iii) Derivative term

The derivative term is obtained by determining the slope of the error over time and multiplying it with the derivative gain, $K_{d,}$. The derivative term is given by the following equation.

$$D = K_d \frac{de(t)}{dt}$$

The graph in Figure 6.14 demonstrates how the output varies with time as K_d changes with K_p and K_i held constant.

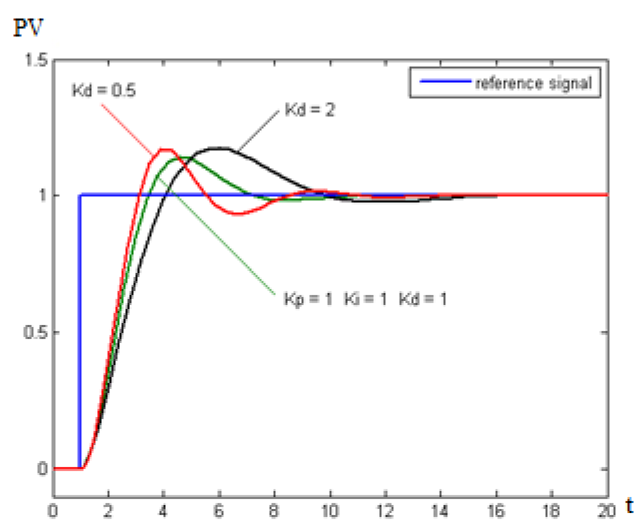


Figure 6.14: Graph of Process Variable (PV) against time t for different values of Kd

The impact of the derivative term is to predict system behaviour whilst improving settling time and stability of the system.

Following the analysis of the different parameters on the output of the system, the gains of the proportional, integral and derivative terms should be chosen correctly to avoid divergence in the output so as to ensure that the system remains stable.

6.4.3. Programming

The programming for the line following part of the robot aims at allowing the latter to move from the room along a path to a specific place such as the kitchen so that the robot can bring back food which is illustrated in the figure below.

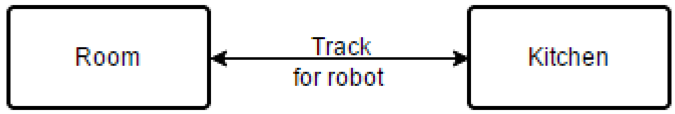


Figure 6.15: Track of line following robot

The line following robot's program can be found in the Appendix J.

Chapter 7: Implementation and Testing

7.1. Introduction

In this chapter, the implementation and testing of the robot consisting of mechanical, electrical, electronics and software designs will be carried out. Testing was carried out in parts, that is, every work performed has been tested before compilation for final testing.

7.2. Robot drive test

The robot was driven through the differential steering concept via the L298N motor driver. After testing was performed, the following results in the table below have been obtained.

Table 7.1: Robot drive test

Requirement	Status		Deduction
	Motor1	Motor2	
Forward	Forward	Forward	Successful
Left	Reversed	Forward	Successful
Right	Forward	Reversed	Successful
Backward	Reversed	Reversed	Successful

From the test performed on the motors, it can be concluded that the drive system was properly implemented with the robot performing all desired steering effectively.

7.3. Adafruit CC3000 Wi-Fi breakout module test

The Adafruit breakout module was connected to the Arduino UNO microcontroller as specified before in the electrical and electronics design chapter. Following the connection of the module just as in figure 7.1, the module is powered and data from it is sent to the microcontroller which is in turn sent to a serial port to a personal computer whereby the desired information is displayed showing that the Adafruit module has been connected appropriately to the network.

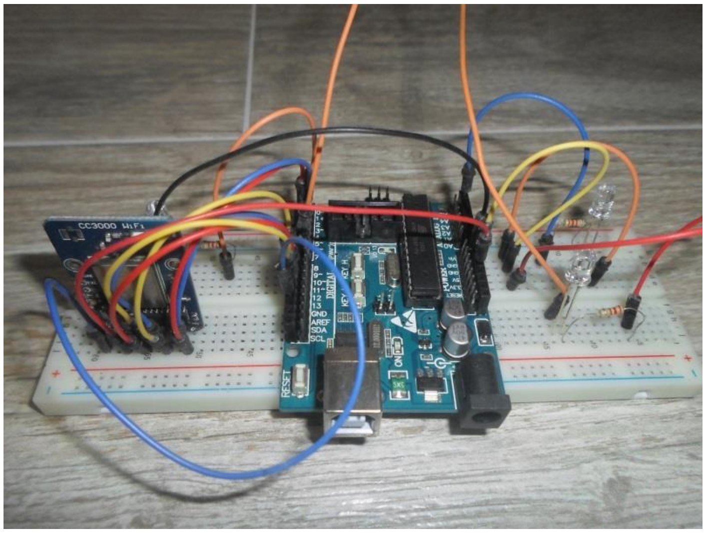


Figure 7.1: Connection of Adafruit CC3000 module and Arduino UNO

Figure 7.2 gives the information displayed on the Serial Monitor screen from the Arduino IDE Software which can confirm the fact that data was successfully sent to the web server and received from it.

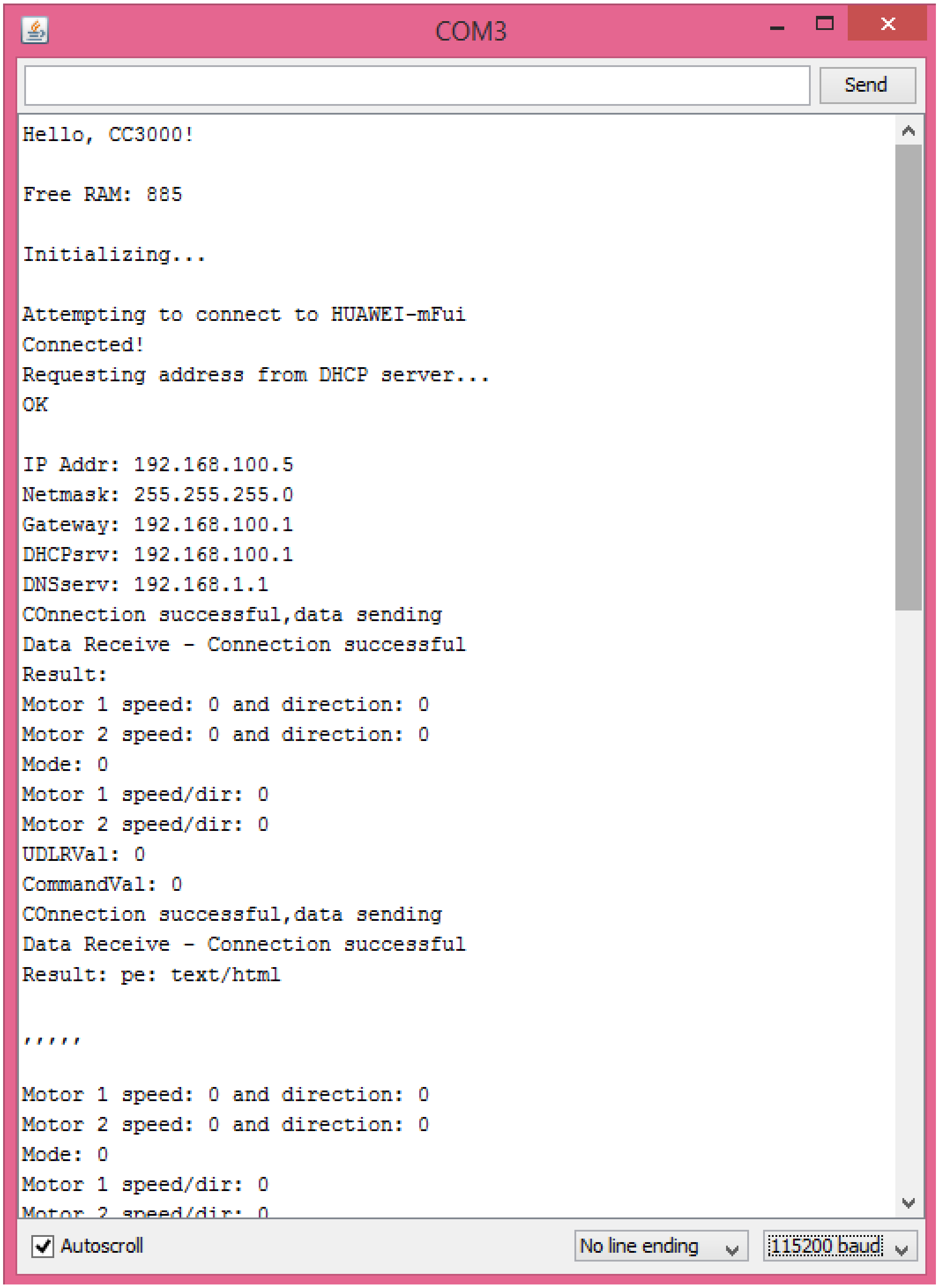


Figure 7.2: Connection of Adafruit module

7.4. Ultrasonic sensor test

The figure 7.3 below demonstrates how the ultrasonic sensor was connected to the Arduino Nano. The test on the ultrasonic sensor was successfully carried out as it gave precise distances. However, disturbances in the sound waves in the surrounding environment affected the performance of the sensor. Taking into consideration that the robot will be mainly running indoor, nearby turbulences from the ultrasonic sensor can be neglected.

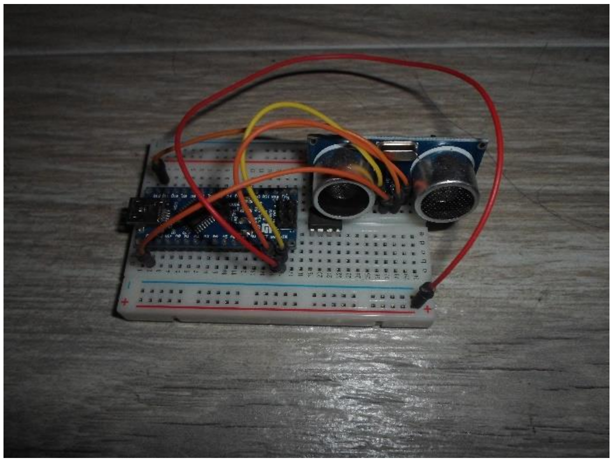


Figure 7.3: Connection of ultrasonic sensor and Arduino Nano

7.5. Motor driver test

The motor driver L298N was tested with the Arduino Nano to check whether the program was correctly written and ensure that the driver is in fine working condition. After testing was carried out, it could be confirmed that the motors A and B could be controlled in both directions and that the speed of rotation of both motors could be changed effectively. Figure 7.4 gives the representation of the connection of the motor driver to the Arduino Nano microcontroller.

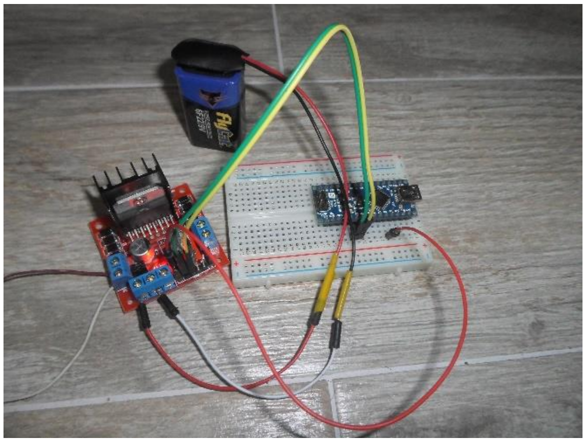


Figure 7.4: Connection of L298N motor driver with Arduino Nano

7.6. Pololu reflectance sensor array test

The Pololu reflectance sensor array was tested with the Arduino Nano microcontroller and positive results were obtained. The figure below shows the connection of the sensor array with the microcontroller which is in turn connected to a USB power bank.

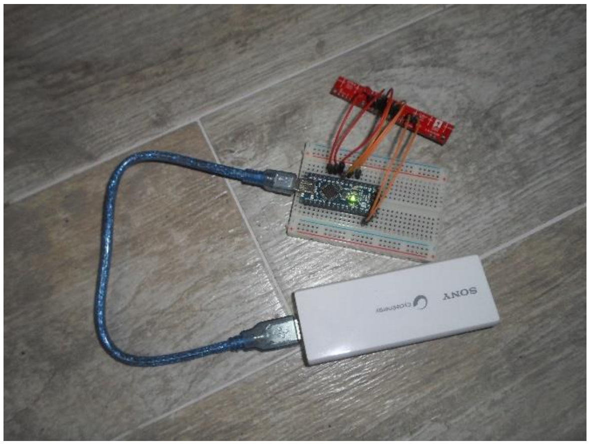


Figure 7.5: Pololu reflectance sensor array connection with Arduino Nano

7.7. Camera software test

The AP camera application software was used on a smartphone and the following results in Table 7.2 are obtained when controls were made on the movement of the camera.

4

Table 7.2: IP camera testing

Motion	Results
Up	Successful
Down	Successful
Right	Successful
Left	Successful

Furthermore, the image captured by the IP camera was clearly visible and a sample of the display of the IP camera is shown in the figure below.

Figure 7.6: Camera

[http://g04.s.alicdn.com/kf/HT1Ccz4FLNXXXagOFbXf/205766753/HT1Ccz4FLNXXXagOFbXf.jpg]

7.8. Problems encountered and solution

Table 7.3: Overall Analysis	
Problems Encountered	Solution
Robot's left wheel was spinning freely in air as well slipping at times as it was not touching ground due to negligible 1mm separating distance.	The metal support plate was drilled again to enlarge existing holes and therefore slide the motor, coupling and wheel downwards.
For the line following mode of the robot, its motion along the black track was unstable due to incorrect values of PID gains used	Testing was carried out with different values and a correct value has been set so that the robot moves along the path with controlled stability.

Chapter 8: Conclusion and Further Improvements

8.1. Conclusion

The aim of this project was to design and implement a robot to aid bedridden persons and which has various functions such as the ability to be remote controlled as well as autonomous. The robot was designed such that it is simple, reliable, efficient and also user-friendly.

Encompassing several concepts like differential steering, obstacle avoidance principles, Wi-Fi control and communication, camera vision as well as line following system, the robot was effectively implemented. The design of a quality robot was targeted and this objective has been achieved along with the vast knowledge gained from constructing this system from the very basic theories learned.

In spite of the achievement of the project goals, further works and adjustments can still be performed on the robot to maximize its functionality and produce a better system. Moreover, with a larger budget, more facilities can be evidently incorporated in the robot. A cost analysis of the robot is given in Appendix K.

8.2. Further improvements

The robot to aid bedridden persons can be improved in the several ways as discussed below:

- A robotic arm could be included in the robot so as it can pick up objects to place them inside it and bring them to the bedridden person.
- A communication system may be integrated to allow the robot to respond to certain voice commands using voice recognition techniques.
- Better support for the objects in the robot can be designed for the robot to be able to carry different types of objects.
- Machine vision can be used to detect certain specific objects which the robot may pick using its robotic arm.

References

Arduino WiFi Shield. Available from: http://www.hwkitchen.com/products/arduino-wifi-shield-integrated-antenna-/ [Accessed on 2 December 2014]

Cyber Care. Available from: http://www.scientificamerican.com/article/robot-elder-care/ [Accessed on 19 August 2014]

Dixon J., Henlich O., 1997, Mobile robot navigation. Available from:

http://www.doc.ic.ac.uk/~nd/surprise_97/journal/vol4/jmd/ [Accessed 11 November 2014]

Hannah J. & Hillier M.J., 1998, Applied Mechanics, 3rd Edition

Infared sensors. Available from: http://www.robotcraft.ca/index.php/main-robot-electronics-catalog/sensors-and-related-electronics/infrared-sensors.html?mode=list [24 December 2014]

Jeetah, N.K., 2005, Design and implementation of an ultrasonic obstacle avoidance robot, Dissertation, University of Mauritius

Limbrunner George F., Spiegel L., 1991, Applied Statics and Strength of Materials, 886 Third Avenue, New York, Macmillan Publishing Company

Line Follower Robot. Available from: http://www.engineersgarage.com/contribution/line-follower-robot [Accessed on 16 December 2014]

Komeda T., Matsuoka H., 1996, Mobile robot system to aid the daily life for bedridden persons. Available from: http://cirrie.buffalo.edu/database/17679/ [Accessed on 19 August 2014]

PID Controller. Available from: http://en.wikipedia.org/wiki/PID_controller [Accessed on 28 January 2015]

Richards, T.H, 1977, Energy Methods in Stress Analysis, Coll House, Westergate, Chichester, Sussex, England, Ellis Horwood Limited Publisher

Ramamruthan, S., 2008, Engineering Mechanics, 9th ed., Dhanpat Rai Publishing Company Ltd

Ramsohoy, G., 2010, Design and implementation of a remotely controlled mobile surveillance robot, Dissertation, University of Mauritius

Ramtohul, A., 2011, An indoor positioning system using WI-FI technology, Dissertation, University of Mauritius

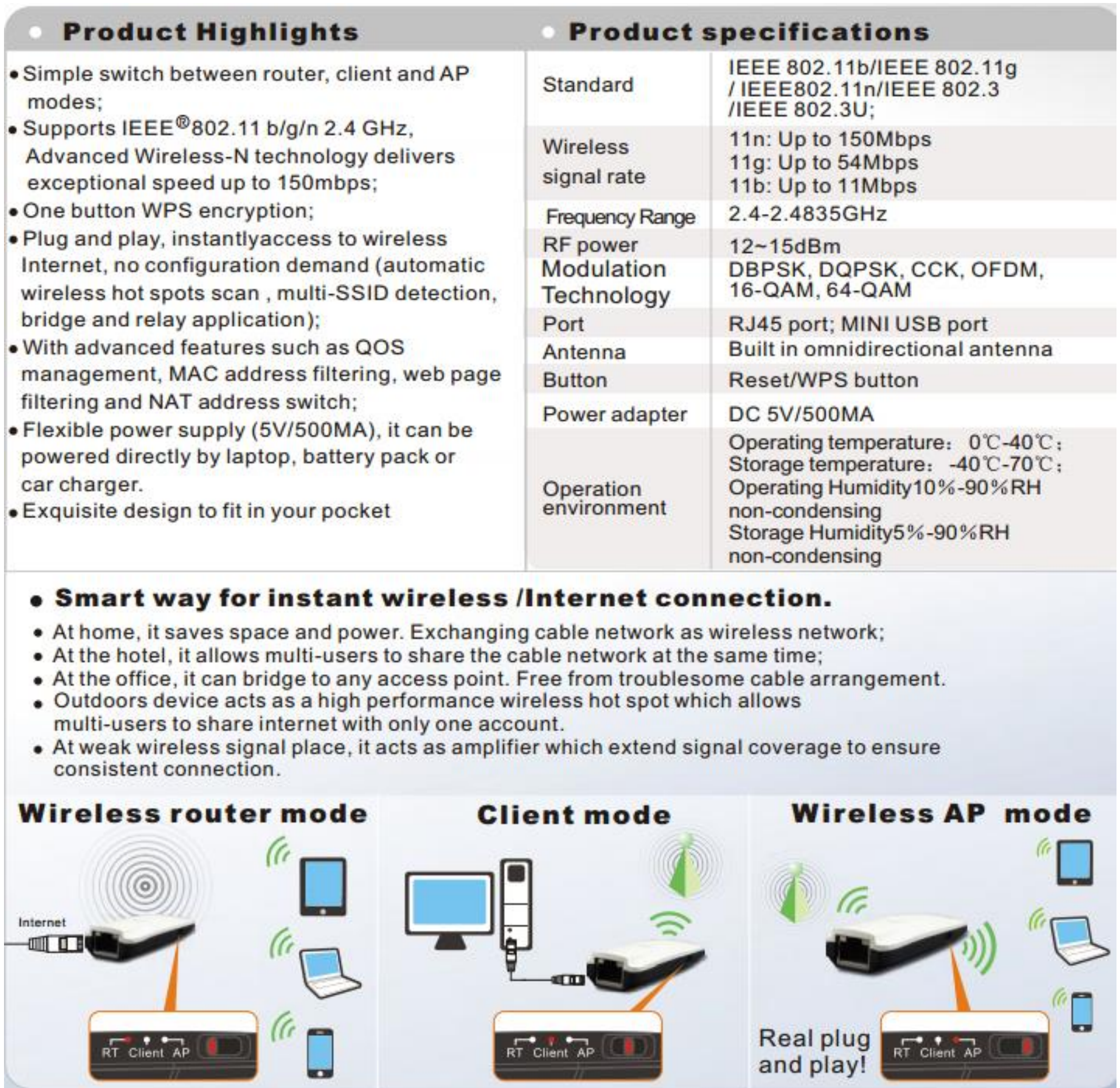
What's The Difference Between DC, Servo & Stepper Motors. Available from: http://www.modmypi.com/blog/whats-the-difference-between-dc-servo-stepper-motors [Accessed on 13 January 2015]

WiFi Robot. Available from: http://www.diyelectronics.co.uk/content/wifi-robot [Accessed on 30 July 2014]

Appendix A

Router mode of operation and specifications

Wireless Pocket Router



Setting up the Wireless-N Pocket Wi-Fi Repeater Using a Wireless Device

Step 1:

Position Switch - First position the switch on the side of the Pocket Repeater to the middle repeater position.

Step 2:

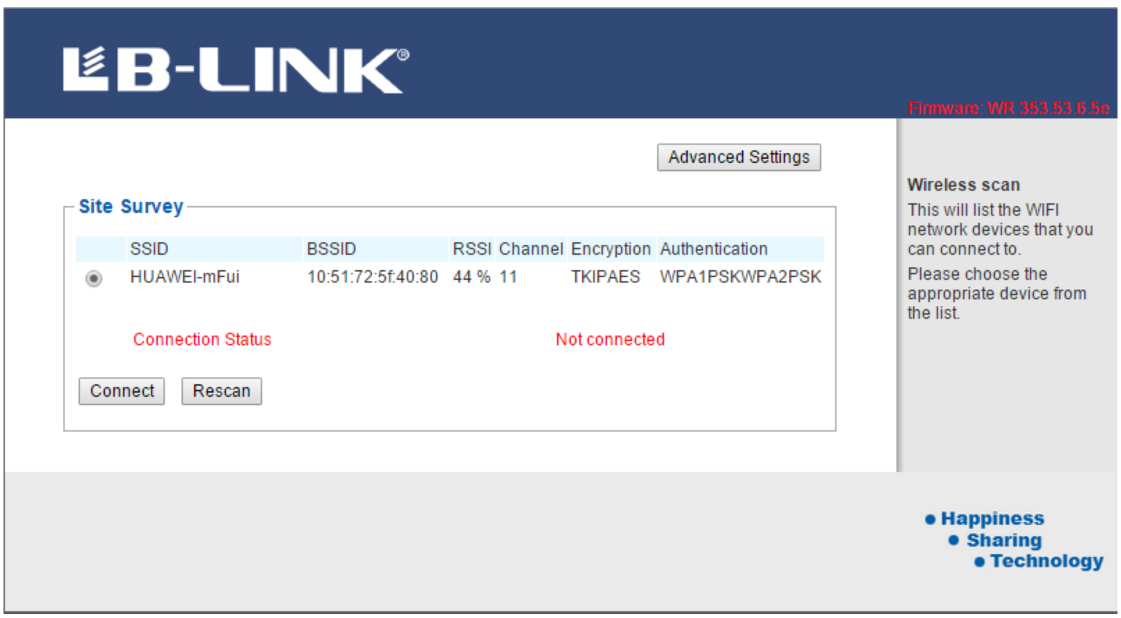
Power – Connect the pocket repeater to a USB socket to power it. This can be the USB socket on the device you intend to use it for, or the USB socket on your PC/Laptop. Once connected the RED power light will come on and the Wi-Fi light will start flashing.

Step 3:

Connect to Pocket Repeater - On your setup device, i.e. laptop, iPad, open your list of available wireless devices and connect to BL-MP01/2-xxxxxxxx, where xxxxxxx is a unique name for each Pocket Repeater. Just look for the device starting with BL-MP01/2. You will not be prompted for a security password when you connect.

Step 4:

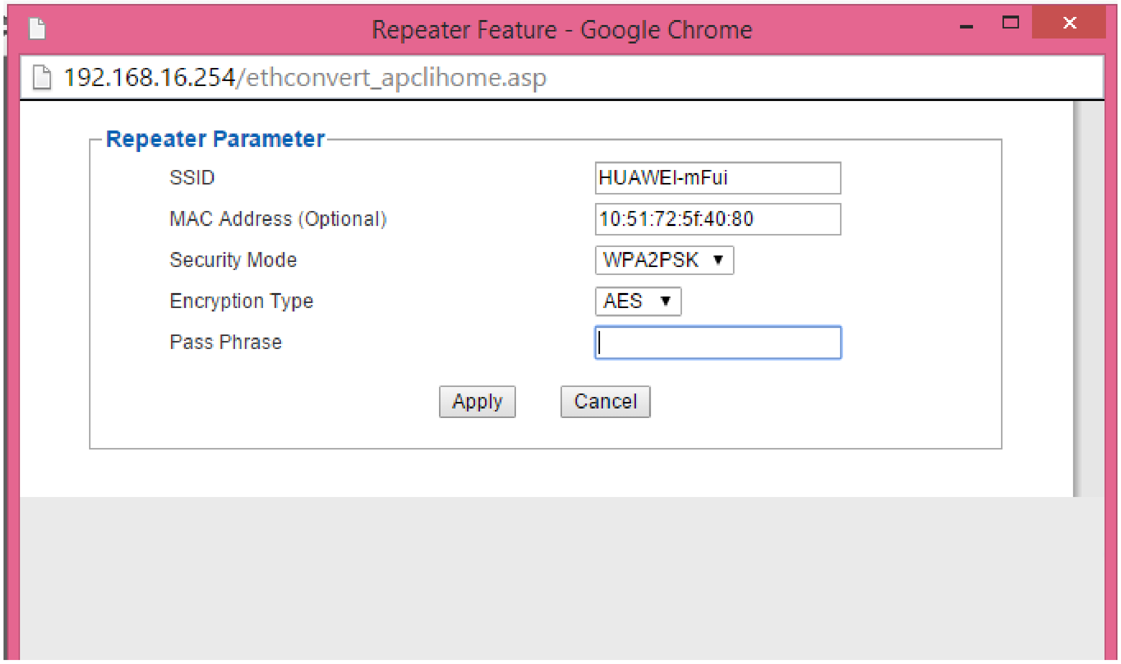
We now need to access the Pocket Repeater Management page using a web browser such as Internet Explorer. Make sure you are wirelessly connected to BL-MP01/2 device then open your web browser and type 192.168.16.254 in the address (not search box) bar > Press Enter. This will bring up the site survey screen, see below.



Select your home router SSID and click on connect.

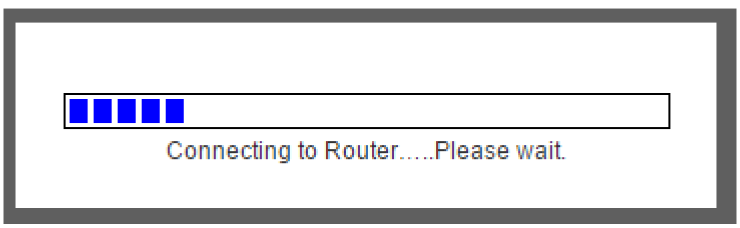
Step 5:

You should now see the Pocket Repeater screen asking for your home routers connection passphrase. Enter your routers Wi-Fi passphrase in the box and click on apply. See below.

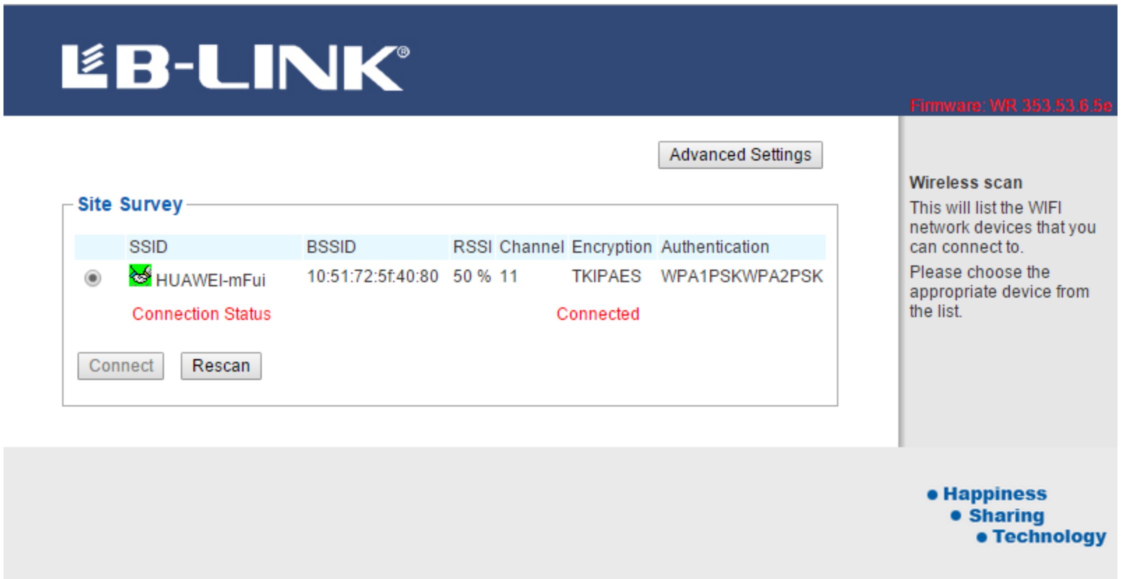


Step 6:

You will now see the connecting to router screen.

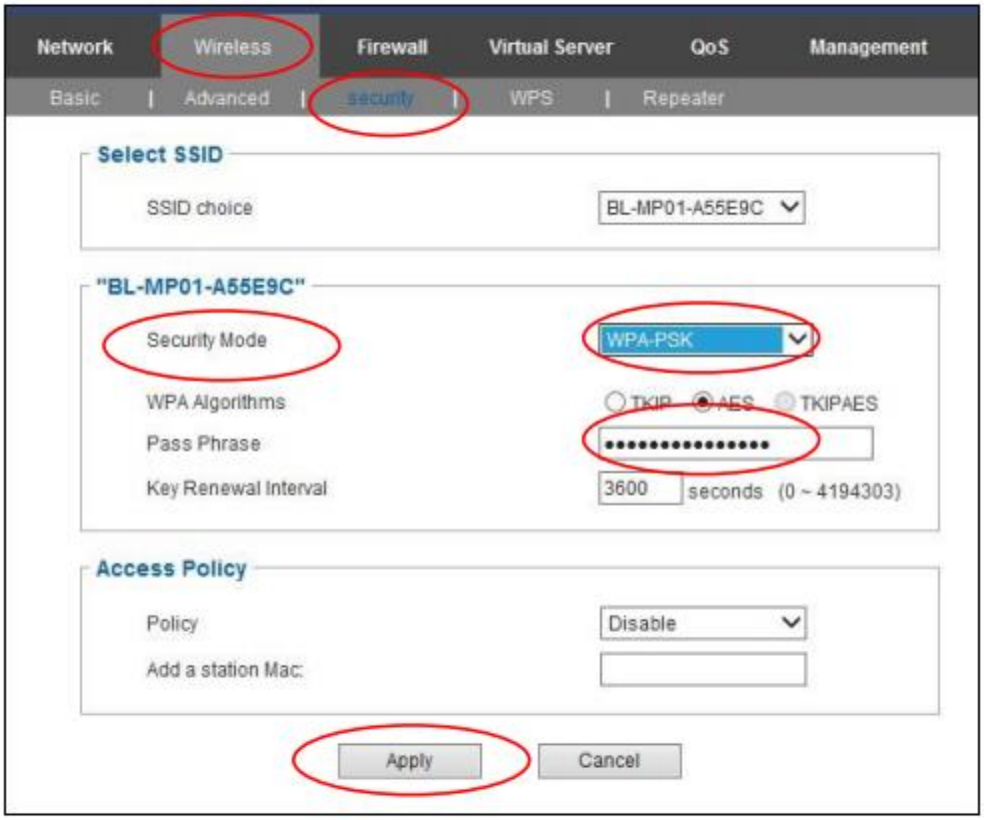


Let it finish connecting then go back to your wireless connection list and reconnect to the pocket repeater (step 3) as it will have most likely dropped the connection. You also need to refresh the 192.168.16.254 page in your browser to bring up the pocket repeater menu Site Survey menu. If you were successful in entering the correct passphrase for your router the pocket repeater will be married up with your home router. If this is the case you will see a green handshake icon next to your router as indicated in the screen below.



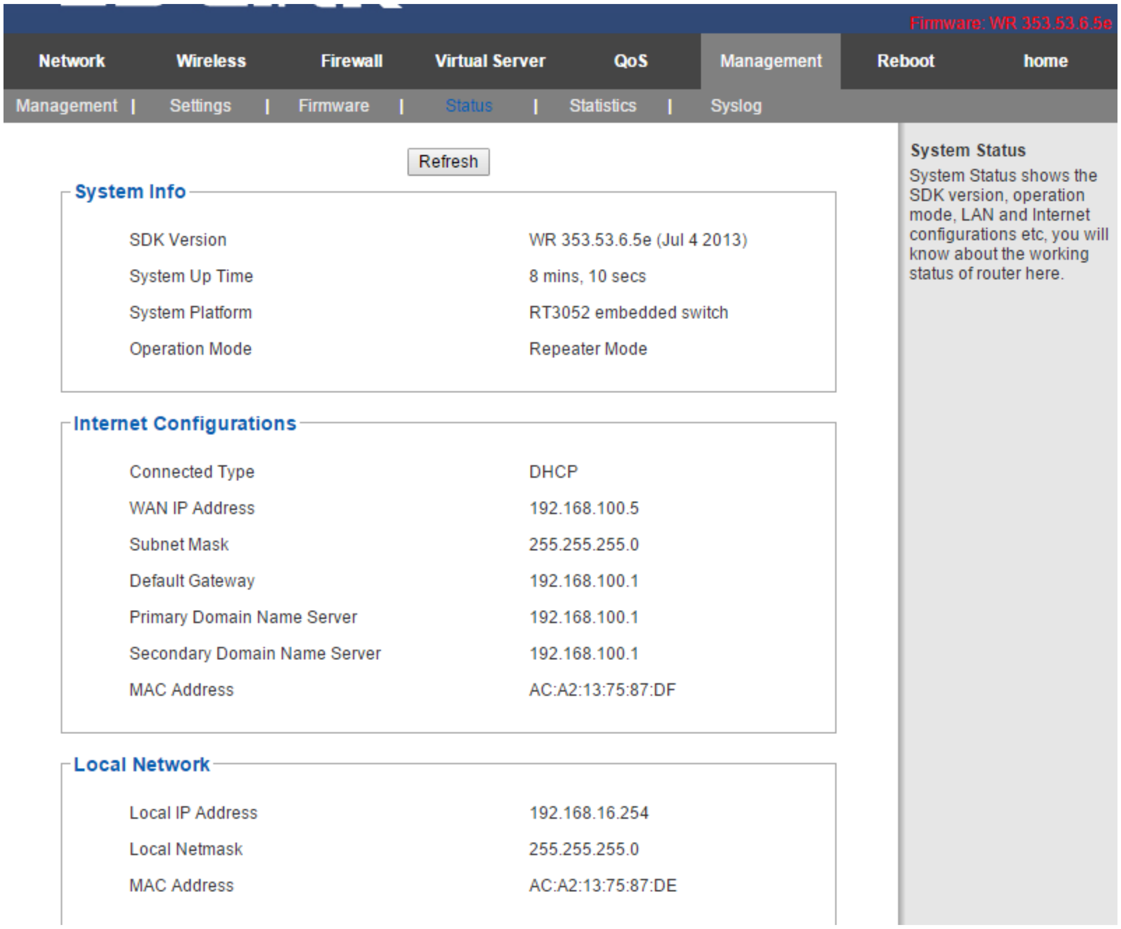
Step 7:

We now need to set some security on the Pocket Repeater to stop anybody else using you're internet connection. To do this you need to click on the Advanced Setting button (see above). This will bring up the Pocket Router management page where you need to click on Wireless and then security from the toolbar. You will then see the following screen.



From the security mode drop down box select WPA-PSK. In the passphrase box enter a passphrase of your choice but to keep it simple. I suggest you use the same passphrase as your home router, i.e. the same one you used in step 5. Click on Apply.

Below is the settings of the router.

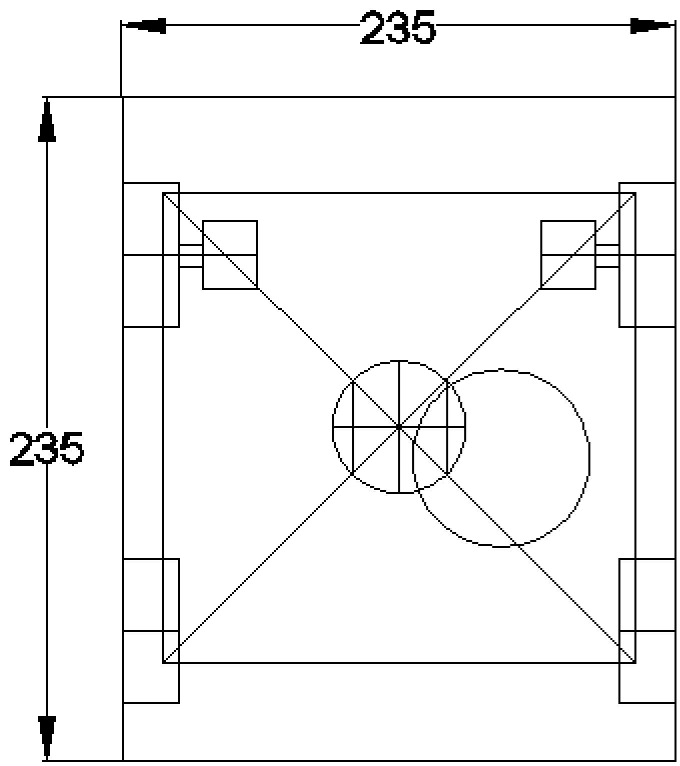

Firmware: WR 353.53.6.5e
Network
Wireless
Firewall
Virtual Server
QoS
Management
Reboot
home
Management | Settings | Firmware | Status | Statistics | Syslog
Refresh
System Info
SDK Version
WR 353.53.6.5e (Jul 4 2013)
System Up Time
8 mins, 10 secs
System Platform
RT3052 embedded switch
Operation Mode
Repeater Mode
Internet Configurations
Connected Type
DHCP
WAN IP Address
192.168.100.5
Subnet Mask
255.255.255.0
Default Gateway
192.168.100.1
Primary Domain Name Server
192.168.100.1
Secondary Domain Name Server
192.168.100.1
MAC Address
AC:A2:13:75:87:DF
Local Network
Local IP Address
192.168.16.254
Local Netmask
255.255.255.0
MAC Address
AC:A2:13:75:87:DE
System Status
System Status shows the SDK version, operation mode, LAN and Internet configurations etc, you will know about the working status of router here.

Appendix B

Mechanical Design Dimensions

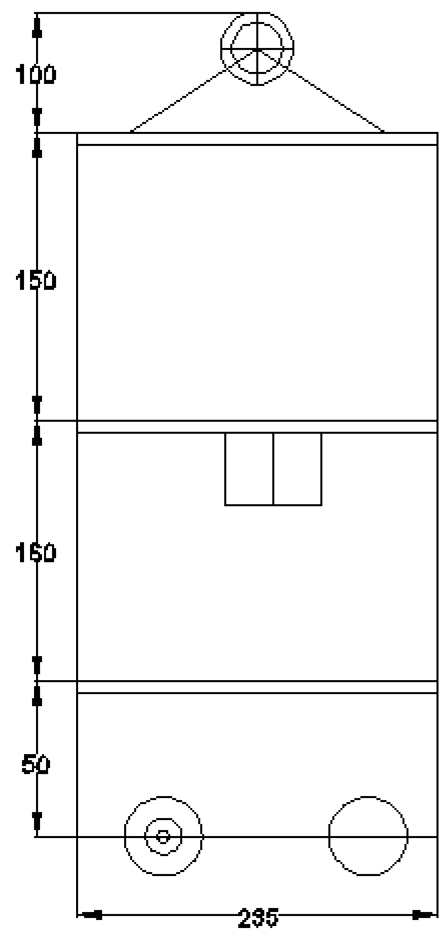
Top view of robot:

The top view of the robot shows the wheels, motors, IP camera and stand to support a glass.



Side view of robot:

The side view of the robot shows the different compartments of the robot, the wheels, the IP camera and the support for the glass.

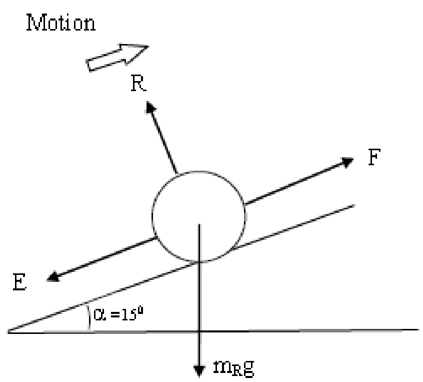


Appendix C

Motor Torque Calculation

Motor Torque Calculation

The motor torque calculation for the two DC motors that will run the robot is calculated below.



The accelerative force of the robot is given by:

$F = m_R g \sin\alpha$, where m_R is the mass of the robot (kg)

g is the acceleration due to gravity (ms^{-2})

The normal reactive force is given by:

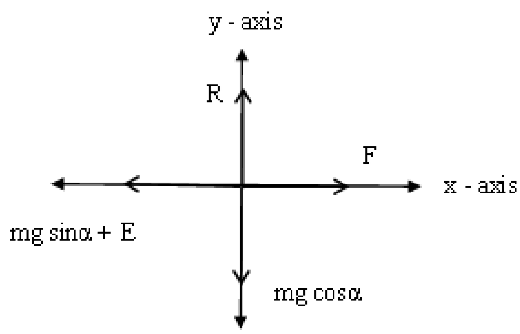
$R = m_R g \cos\alpha$, where m_R is the mass of the robot (kg)

g is the acceleration due to gravity (ms^{-2})

The resistive force due to friction is given by:

$E=\mu R$, where μ is the coefficient of friction

R is the normal reactive force



$\sum F_x = m_R a$, where a is the acceleration of the robot (ms^{-2})

$F - E - m_R g\sin\alpha = m_R a$

$F = m_R a + E + m_R g\sin\alpha$

$F = m_R a + \mu R + m_R g\sin\alpha$

F = (3.292×1) + (0.25×3.292×9.81×cos15) + (3.292×9.81×sin15)

= 19.4 N

Torque = Force x Radius

The radius of the wheels used is 0.0325 m.

Torque = F x r

= 19.4 × 0.0325

= 0.631 Nm

Since the robot is driven by two DC motors, the torque of each motor is given by:

$$\text{Torque of each motor} = \frac{\text{Total torque of the robot}}{2}$$

Torque of each motor = 0.316 Nm

With a safety factor of 1.5,

Torque of each motor = 0.474 Nm

Power of motor = Torque x Angular velocity ($rads^{-1}$)

Power = Torque x 2лN/60, where N is the revolution of the motor in rpm.

Power of motor = 0.474× (2л×60)/60 = 3.0 W

Appendix D

Installing Additional Arduino Libraries

Installing Additional Arduino Libraries

Once you are comfortable with the Arduino software and using the built-in functions, you may want to extend the ability of your Arduino with additional libraries.

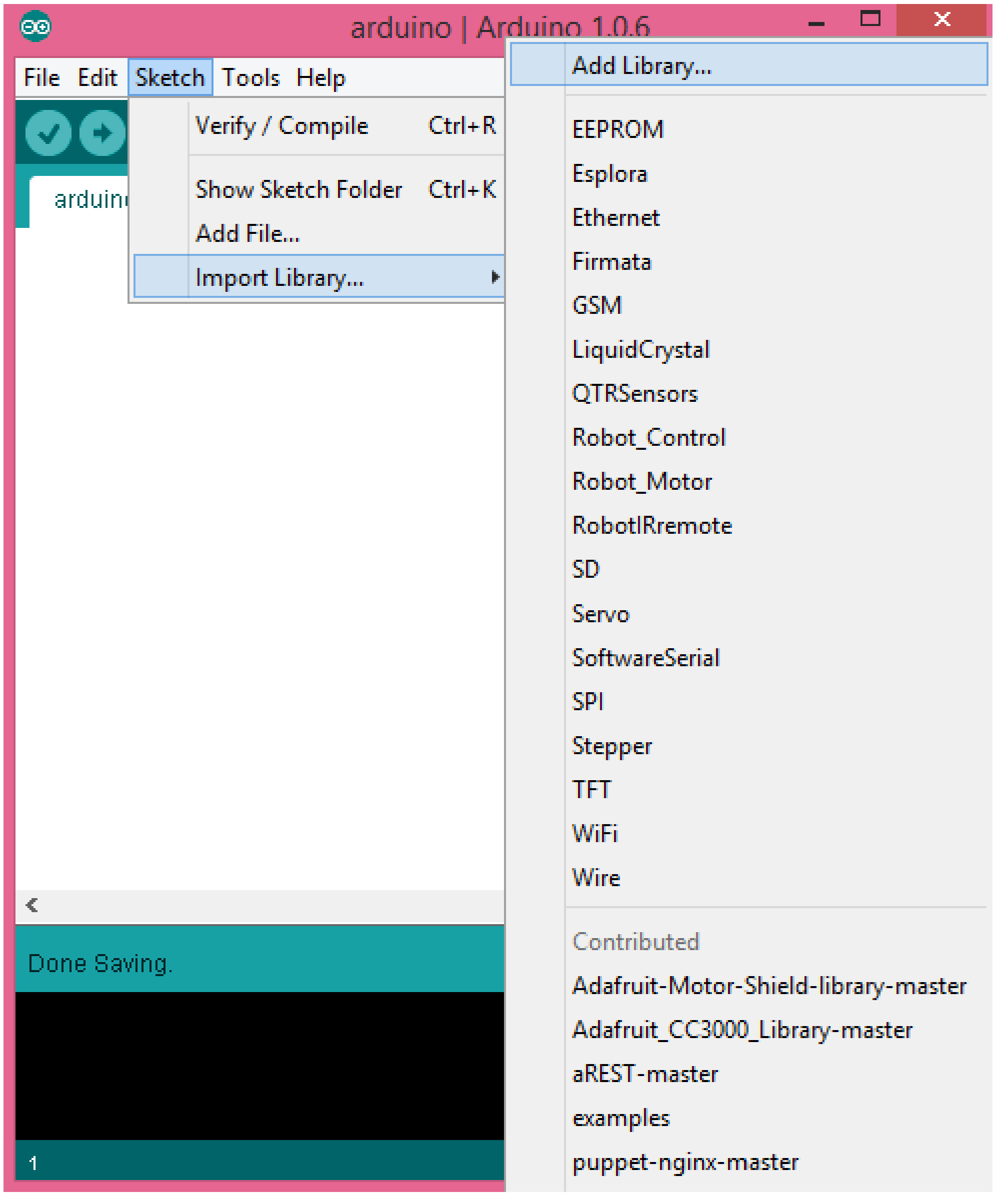
Libraries are a collection of code that makes it easy for you to connect to a sensor, display, module, etc.

Importing a .zip library

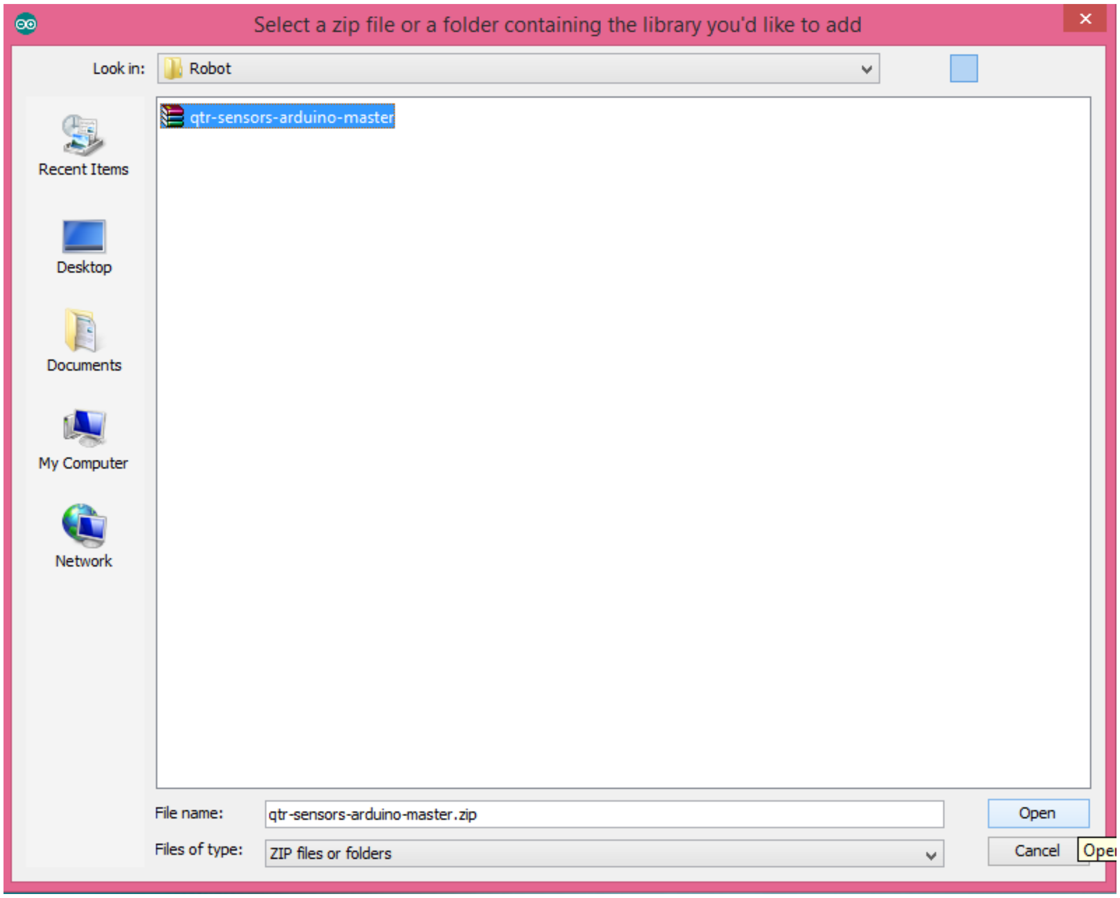
Libraries are often distributed as a ZIP file or folder. The name of the folder is the name of the library. Inside the folder will be a .cpp file, a .h file and often a keywords.txt file, examples folder, and other files required by the library.

Do not unzip the downloaded library, leave it as is.

In the Arduino IDE, navigate to Sketch > Include Library. At the top of the drop down list, select the option to "Add .ZIP Library".



You will be prompted to select the library you would like to add. Navigate to the .zip file's location and open it.



Return to the Sketch > Import Library menu. You should now see the library at the bottom of the drop-down menu. It is ready to be used in your sketch.

The zip file will have been expanded in the libraries folder in your Arduino sketches directory.

The library will be available to use in sketches, but examples for the library will not be exposed in the File > Examples until after the IDE has restarted.

Manual installation:

To install the library, first quit the Arduino application.

Then uncompress the ZIP file containing the library. For example, if you're installing a library called "ArduinoParty", uncompress ArduinoParty.zip. It should contain a folder called ArduinoParty, with files like ArduinoParty.cpp andArduinoParty.h inside. (If the .cpp and .h files aren't in a folder, you'll need to create one. In this case, you'd make a folder called "ArduinoParty" and move into it all the files that were in the ZIP file, like ArduinoParty.cpp and ArduinoParty.h.)

Drag the ArduinoParty folder into this folder (your libraries folder). Under Windows, it will likely be called "My Documents\Arduino\libraries". For Mac users, it will likely be called "Documents/Arduino/libraries". On Linux, it will be the "libraries" folder in your sketchbook.

Your Arduino library folder should now look like this (on Windows):

My Documents\Arduino\libraries\ArduinoParty\ArduinoParty.cpp

My Documents\Arduino\libraries\ArduinoParty\ArduinoParty.h

My Documents\Arduino\libraries\ArduinoParty\examples

Restart the Arduino application. Make sure the new library appears in the Sketch->Import Library menu item of the software.

Appendix E

Setting Static IP Address

Setting static IP address in Windows 8

It is very important to setup a static IP address, if you are going to use port forwarding. When you have port forwarding setup, your router forwards ports to an IP address that you specify. This will probably work when you initially set it up, but after restarting your computer it may get a different IP address. When this happens the ports will no longer be forwarded to your computer's IP address. So the port forwarding configuration will not work.

IP addresses

IP addresses are four sets of numbers separated by periods that allow computers to identify each other. Every computer has at least one IP address, and two computers should never have the same IP address. If they do, neither of them will be able to connect to the internet. There is a lot of information at the following link. You don't need all of it. But if you want to know more about how networks work, you'll find it there. For more information on IP addresses, subnets, and gateways go here

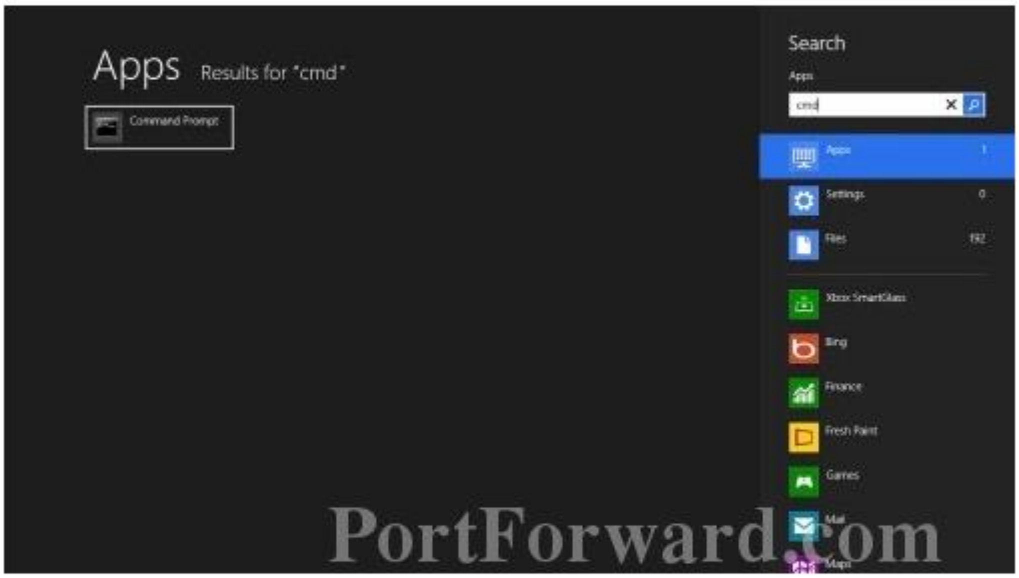
Dynamic vs Static IPs

Most routers assign dynamic IP addresses by default. They do this because dynamic IP address networks require no configuration. The end user can simply plug their computer in, and their network will work. When IP addresses are assigned dynamically, the router is the one that assigns them. Every time a computer reboots it asks the router for an IP address. The router then hands it an IP address that has not already been handed out to another computer. This is important to note. When you set your computer to a static IP address, the router does not know that a computer is using that IP address. So the very same IP address may be handed to another computer later, and that will prevent both computers from connecting to the internet. So when you assign a static IP addresses, it's important to assign an IP address that will not be handed out to other computers by the dynamic IP address server. The dynamic IP address server is generally referred to as the DHCP server.

Setting up a static IP for Windows 8.

Step 1:

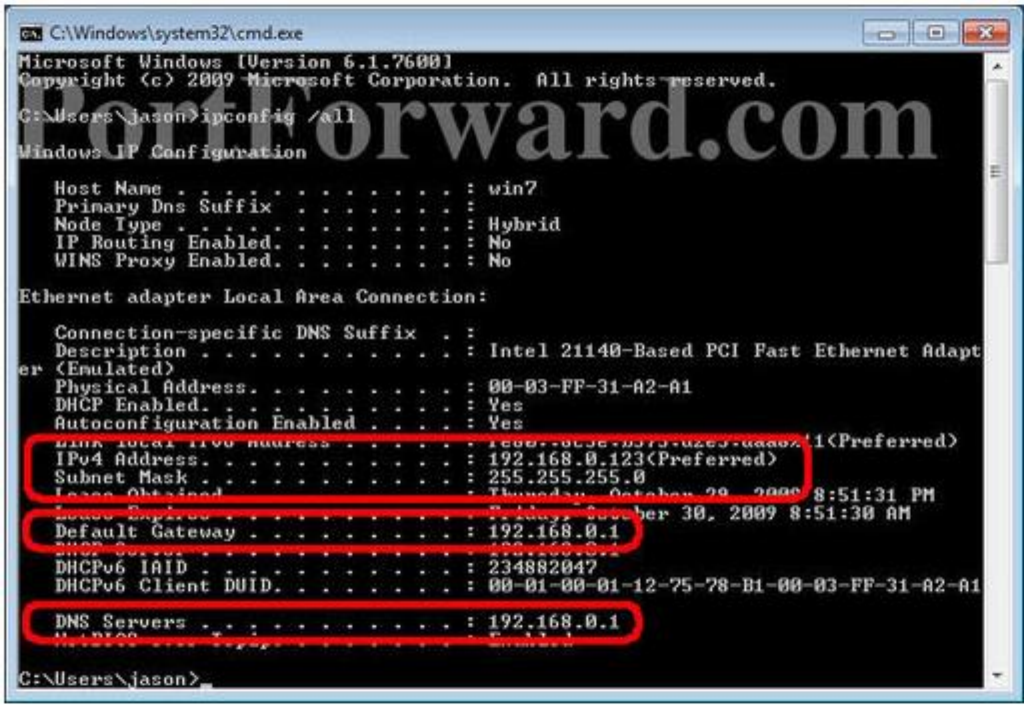
Open up the new windows 8 start screen by pressing the windows key on your keyboard.



Step 2:
After you see the start screen appear, simply type "**cmd**" then press the **enter** key on your keyboard to open up a command prompt.

Step 3:

The command prompt may look different on your screen, but it doesn't really matter. Type ipconfig /all in that window, and then press the enter key. This will display a lot of information. If it scrolls off the top you may need to enlarge the window.



Step 4:

I want you to write down some of the information in this window. Take down the IP address, Subnet Mask, Default Gateway, and Name Servers. Make sure to note which is which. We are going to use this information a little bit later. We are only concerned with IPv4 entries, you can ignore the IPv6 stuff.

The name server entries are a bit complicated. Name Server is just another name for DNS(domain name server) server. Some router's act as a proxy between the actual name servers and your computer. You will know when this is the case, because the Default Gateway will list the same ip address as the Name Servers entry. We need to have the correct Name Server IP addresses. If we do not, you will not be able to browse the web. There are a couple ways to get these. The first way is to log into your router's web interface, and look at your router's status page. On that page you should see an entry for DNS Servers, or Name Servers. Write down the ip adresses of your Name Servers. Another way to get the correct Name Servers to use, is to give your ISP a call. They should know the ip addresses of your Name Servers right off. If they ask you why you need them, you can tell them you are trying to setup a static IP address on your computer. If they try to sell you a static external ip address, don't buy it. That's an entirely different thing that what you are trying to setup.

Type exit in this window, then press the enter key to close it.

Step 5:

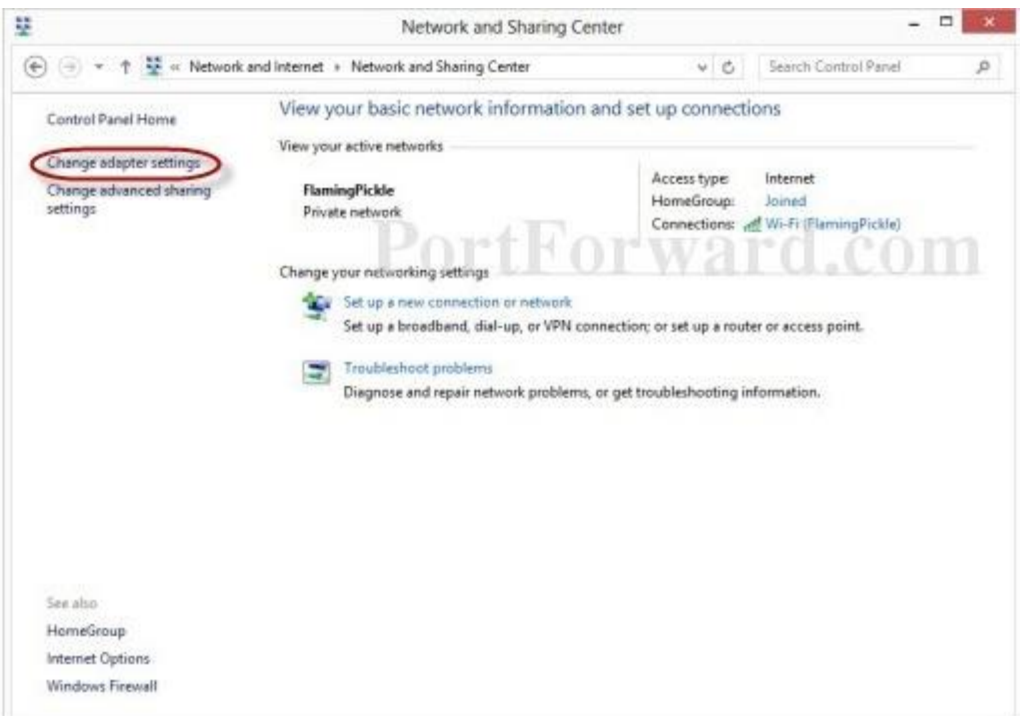
Once again open the windows 8 start screen. This time type Control Panel and press enter.

Step 6:

Click on View Network Status and Tasks.

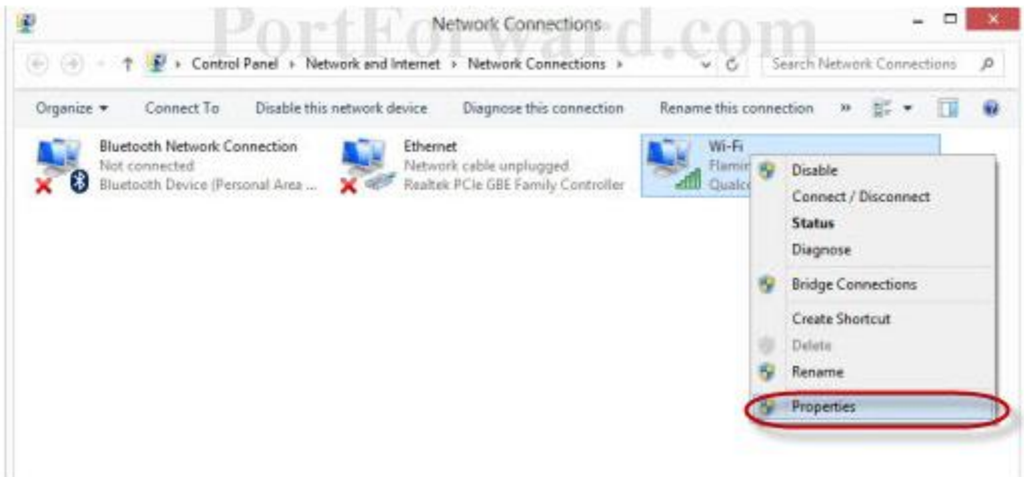
Step 7:

Single click Change adapter settings on the left side of your screen.



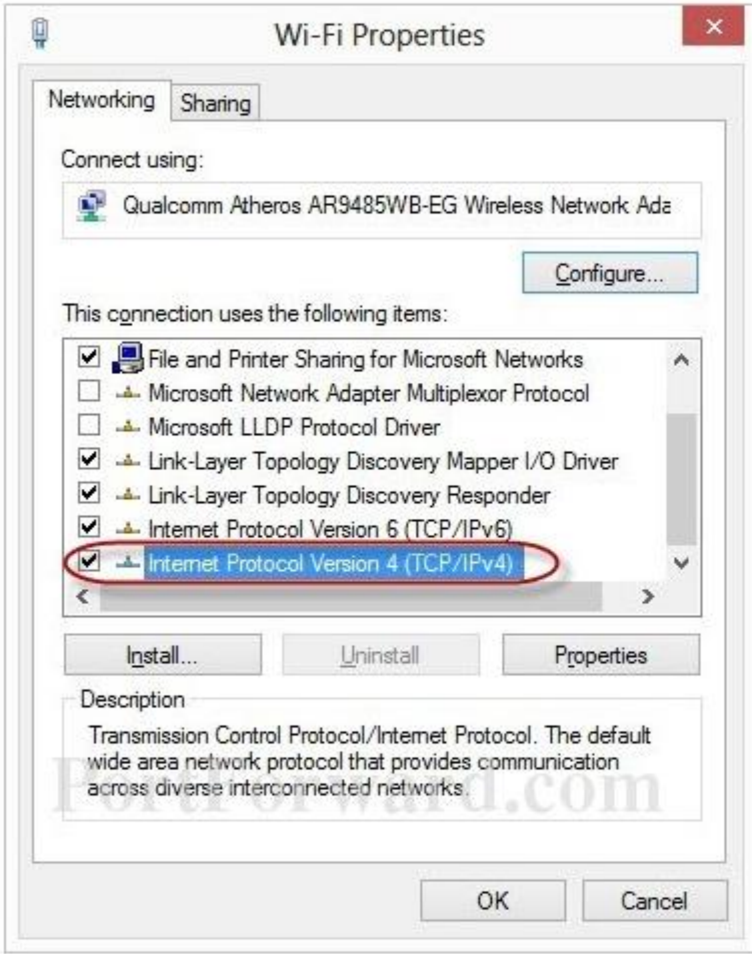
Step 8:

You might have more than one Internet connection listed here. You will need to determine which adapter is your connection to the Internet if this is the case. Right click on your network adapter and choose properties to open up the properties window of this internet connection.

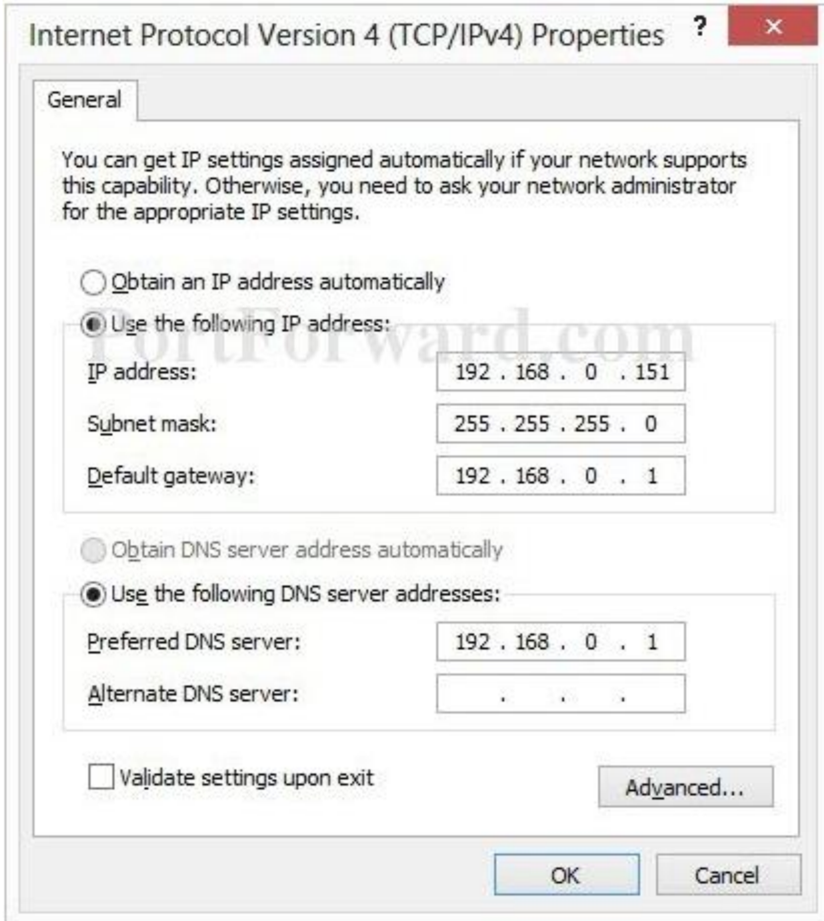


Step 9:

Click Internet Protocol Version 4(TCP/IPv4) and then the Properties button.



You will see the following screen:



Step 10:

Before you make any changes, write down the settings that you see on this page. If something goes wrong you can always change the settings back to what they were! You should see a dot in the Obtain an IP address automatically box. If you do not, your connection is already setup for a static IP. Just close all these windows and you are done.

Pick an IP address and enter it into the IP Address box. The IP address you choose should be very similar to the router's IP address. Only the last number of the IP address should be different. If the router's IP address is 192.168.1.1, I might choose 192.168.1.10. The IP address you choose should end with a number between 1 and 254, and should not be the same as the router's IP address. Every device that connects to your network needs to have it's own IP address.

Put the subnet mask we previously found in the subnet mask section. The default gateway should go into the Default gateway box. Enter the DNS servers we previously found into the two DNS Server boxes. Click okay all the way out of this menu.

Setting up a static IP address through software

The software can be downloaded on the following link:

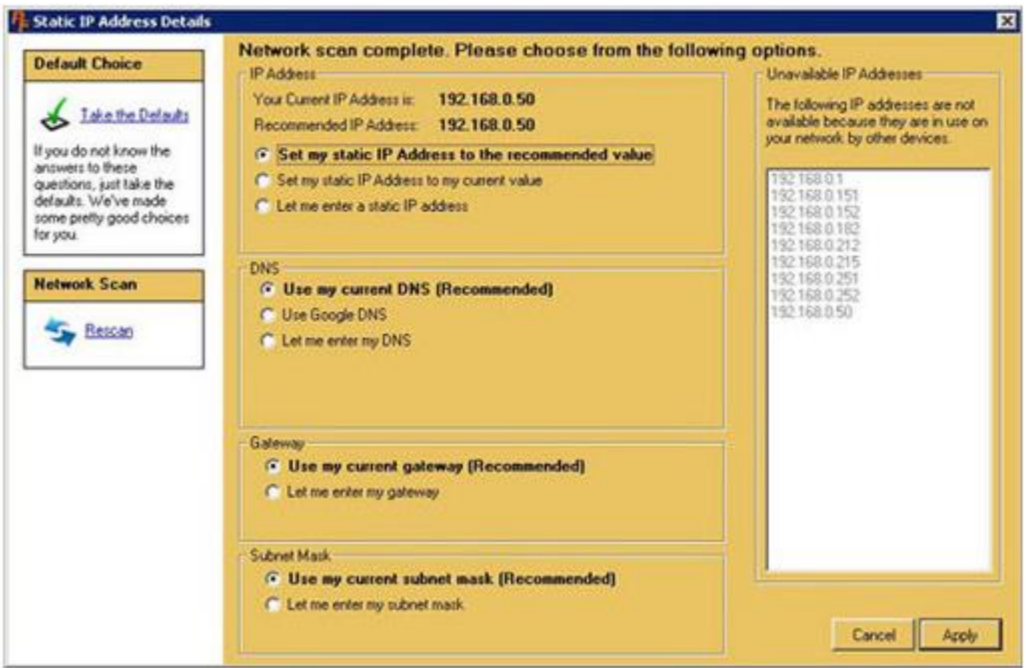
http://portforward.com/help/setup_static_ip_address.htm.

- Install Portforward Network Utilities and launch it with the icon on your desktop.
- Click **Static IP Address** button. Your network will be analyzed. You can then click **Make it Static** to change your IP address to static.

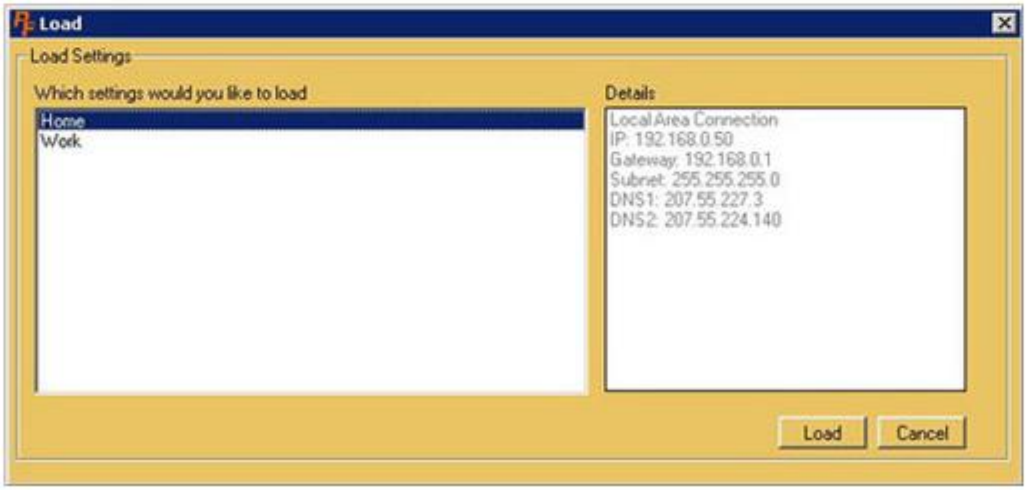
Features:

- Scan your network for available IP addresses
- Pick the best IP address and DNS server for you, so you don't have to do anything.
- Save your network settings so you can take your laptop with you and always have the network settings you want available.
- DHCP release / renew. In fact PFStaticIP makes an excellent free WinIPCfg replacement.

The figures below show some screenshots of the software.



You can save and load network configurations for quick changes.


Load
Load Settings
Which settings would you like to load
Home
Work
Details
Local Area Connection
IP: 192.168.0.50
Gateway: 192.168.0.1
Subnet: 255.255.255.0
DNS1: 207.55.227.3
DNS2: 207.55.224.140
Load
Cancel

Appendix F

Port Forwarding on Huawei HG8245 WiFi Router

Port Forwarding the Huawei HG8245 Router

Port forwarding is a method of making a computer on your network accessible to computers on the Internet, even though you are behind a router. It is commonly used for hosting game servers, peer to peer downloading, and voice over IP type applications.

Ports are virtual pathways on which information on the Internet travel. There are 65,536 ports to choose from. A good analogy is to think of ports like extensions on a phone system.

To setup port forwarding on the Huawei HG8245 router your device needs to have a static IP address.

Login to the Huawei HG8245 Router Using Your Web Browser.

Open a web browser like Internet Explorer or Firefox. Enter the internal IP address of your router in the address bar of your browser. By default the IP address should be set to 192.168.100.1.

You should see a box prompting you for your username and password. Enter your username and password now.

The Default Huawei HG8245 Router Username is: telecomadmin

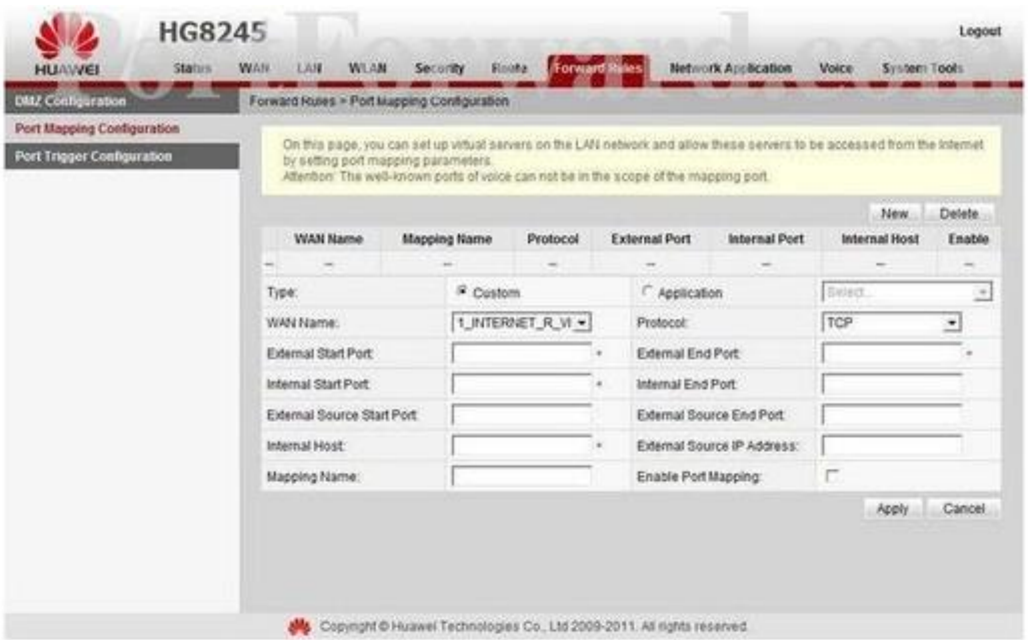
The Default Huawei HG8245 Router Password is: admintelecom

Click the Login button to log in to your Huawei HG8245 router.

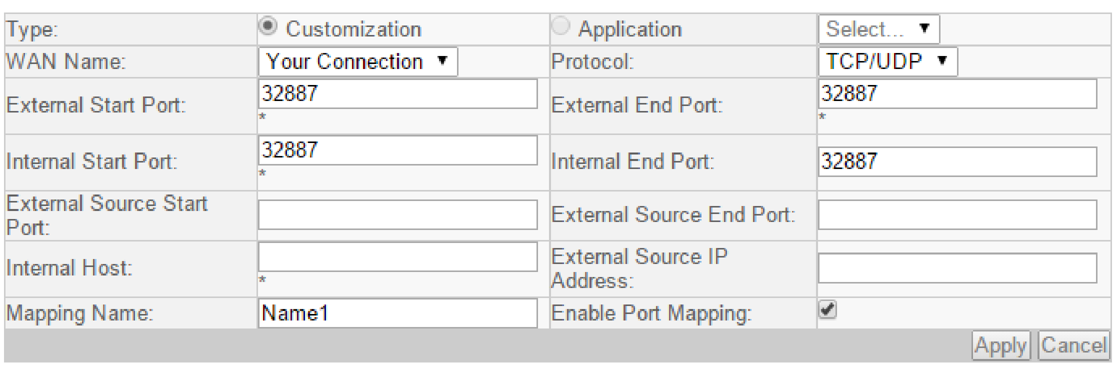
Huawei HG8245 Router Port Forwarding Screenshots

Click the Forward Rules link near the top of the page.

The menu on the left should change. In this new menu, click Port Mapping Configuration.



We will list a series of lines here that will show you exactly how to forward the ports you need to forward. In the figure below, an example of 32887 was used as port number. Go ahead and enter the settings shown above into the Port Mapping Configuration menu and then click Apply.



Type:	Customization	Application	Select...
WAN Name:	Your Connection	Protocol:	TCP/UDP
External Start Port:	32887 *	External End Port:	32887 *
Internal Start Port:	32887 *	Internal End Port:	32887
External Source Start Port:		External Source End Port:	
Internal Host:	*	External Source IP Address:	
Mapping Name:	Name1	Enable Port Mapping:	✔
			Apply Cancel

The port forwarding process has been completed.

Appendix G

IP camera set up

Sricam IP camera setup

Sricam IP Camera Brief Instructor

Thanks for choosing our products and service. Here are some questions hope can help you on installation. If you have any other questions, please feel free to contact with us and we will reply with 24 hours.

1. Software Download.
2. Connect the camera to network.
3. View the camera via smart phone.
4. View the camera via Browser.
5. WiFi setting.

1. Software download

http://www.sricctv.com/en/service/download

Notice: Please try to access this download website via the Internet Explorer Browser.

2. Connect the camera to network

First: Power the IP Camera and connect the camera to the router with a net-cable.

Second: Check Whether the power (yellow) pilot lamp and the network pilot lamp(green) are on, if not, check the net-cable and power adapter. After the camera connected to network, you need install the software.

3. How to view the camera via smart phone?

The camera support remote monitor via the smart phone (Android and iPhone) or PC。
First: Install APP to the Mobile Phone. The APP name is: APCamera.

A: APCamera APP Installation.

1. Phone with the QR Code Scanner. Scan bellow QR Code for download the Android or iPhone APP,then install it to your phone.

Available on the App Store | ANDROID APP ON Google play

2: Phone without the QR Code Scanner. Please download the APP from bellow website, Or search and download "APCamera" in google Play store or App Store of Apple.

For iPhone: https://itunes.apple.com/us/app/apcamera/id666250856?mt=8
For Andorid: https://play.google.com/store/apps/details?id=object.shazx1.client

B: Add Cameras to APCamera App.

Firstly, enable the WI-FI function of your mobile phone. (Note:Your phone and the camera must be in the same LAN.) Secondly, run the application.

After running the application, click to add cameras。

Add the camera to APCamera . (There are three ways to add camera into it.)
Method 1 : Input the Did,User, Pwd of your camera manually. Then click "Done" button.
Method 2 : Click "Search" button then you will get the camera list, the cameras in the same LAN with your phone. Choose the camera which you want to monitor. Finally input the Pwd, Then click "Done" button.
Method 3 : Scan the QRCode: Click the "Scan ID" button, at the button of the camera, there is a label with QRCode, then use your phone to scan the bar code, it will show the ID, then press "done".

View the camera via "APCamera" App. begin to view the real-time video. You will see following left pictures, Click the camera which you just added, then you will view the video show on the App.

Live video and operation interface.

①:Exit

②:Preset position

③:Mirror

④:Upside down

⑤:Autp pan&tilt

⑥:Full screen

⑦:Listen and speak

⑧:Photo

⑨:Record

⑩:Mode

⑪:Resolution

⑫:Frame rate

⑬:Brightness

⑭:Contrast

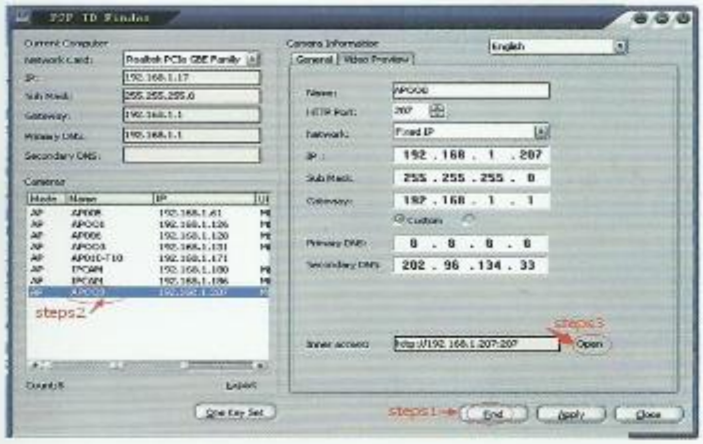
⑮:Default all

⑯IR control

Note: Sliding the screen can make the camera pan and tilt.

4. How to view the camera via Browser ?

Steps-1: Install the software " P2P ID Finder" software to your PC.
Steps-2: Install the "IPCam ActiveX",
Steps-3: Install the "oPlayer" Ocx software.
Steps-4: Run the search tool, Click [Find] button. All the IP camera in LAN will be listed on the left.Choose the camera which you want to view. Then click Inner access: http://192.168.1.207:207 [Open] Open then Login window pops up, Input the user name and password, the default username is admin, no password.



Input: user name and password.

Select the language and the browser.

The video shows as below

5. How to setting the WiFi ?

A. Login the camera, go to wireless LAN settings.

B.

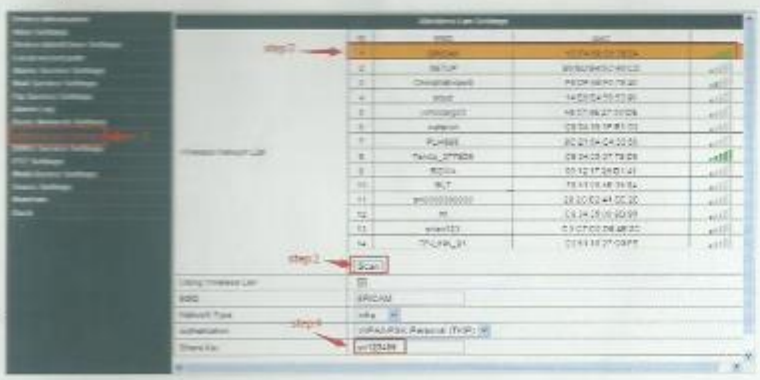
B-1. Go to wireless LAN settings.

B-2. Click"Scan" twice to find the wifi signals. (Scan for more times,if find no,try to powercycle your camera.)

B-3. choose the right SSID, in other words,the wifi signal you want to use.

B-4. Input the wifi key, it is the password of your wifi network.

B-5. Submit. (After submit, your camera will reboot automatically, then disconnect the cable, run the search tool " P2P ID Finder" to find out the camera IP address or refresh the webpage) .



PAY ATTENTION:

1.The key must be right(The Key only support numbers and letters, does not support the special Characters, such as: @, #, %,$ .....) It can be found in your router.
2.The distance from camera to the wireless router should be less than 20m.
3.Tighten the antenna.   4.Try to disconnect other wireless devices.
5.Change the encryption and channel.WPA or WPA2,AES or TKIP.Channel should be less than 11. 802.11b/g/n mode better.

Appendix H

Server files for semi-automatic system

Server files

1. **updateStateNew.php programming**

```
<?php


	// Load JSON state

   $string = file_get_contents("robotStateNew.json");

   $json_a= json_decode($string,true);



	//first part is the name of the value in the json text file that will be read by server.php when called from Arduino

	//second part is the URL value that is parsed.

   $json_a['mode'] = $_GET["URLmode"];

   $json_a['xval'] = $_GET["URLxval"];

   $json_a['yval'] = $_GET["URLyval"];

	$json_a['udlr'] = $_GET["URLudlr"];

	$json_a['cmdVal'] = $_GET["URLcmdVal"];



   $fp = fopen('robotStateNew.json', 'w');

   fwrite($fp, json_encode($json_a));

   fclose($fp);
```

```
// Create a TCP/IP socket & connect to the server

$socket = socket_create(AF_INET, SOCK_STREAM, SOL_TCP);

$result = socket_connect($socket, "192.168.100.5", "80");


// Request

$in = "HEAD / HTTP/1.1\r\n";

$in .= "Content-Type: text/html\r\n";

$in .= $json_a['mode'] . "," .

$json_a['xval'] . "," .

$json_a['yval'] . "," .

$json_a['udlr'] . "," .

$json_a['cmdVal'] . ",\r\n\r\n";

$out = '';


// Send request

socket_write($socket, $in, strlen($in));


// Read answer

while ($out = socket_read($socket, 4096)) {

  echo $out;

}
```

```
        // Close socket
        socket_close($socket);

?>
```

2. **robotStateNew.json programming**

{"mode":"","xval":"","yval":"","udlr":"","cmdVal":""}

3. **server.php programming**

```
<?php
   // Load JSON state
   $string = file_get_contents("robotStateNew.json");
   $json_a= json_decode($string,true);
   // Send command to robot
   foreach ($json_a as $key => $val){
      echo $val;
      echo ",";
   }
?>
```

Appendix I

Arduino programming for semi-automatic system

Semi-automatic programming

```
#include <Adafruit_CC3000.h>

#include <ccspi.h>

#include <SPI.h>

#include <string.h>

#include "utility/debug.h"

#include <stdlib.h>

#include <StopWatch.h>


StopWatch MySW;

StopWatch SWarray[5];


String result;

int motorCommand[5];

// will be used to parse values from the json file on the server:
{"mode":"1","xval":"0","yval":"0","udlr":"","cmdVal":""}

int resultLength;

float leftSpeed, rightSpeed;


int counter = 0;

int port = 80;
```

```
#define leftMtrSpdPin 6  //PWM pin

#define rightMtrSpdPin 9 //PWM pin

#define leftMtrDirPin 7

#define rightMtrDirPin 4

#define LEDStatPin 2

#define LEDDataReceivedPin A5


// These are the interrupt and control pins

#define ADAFRUIT_CC3000_IRQ   3  // MUST be an interrupt pin!

// These can be any two pins

#define ADAFRUIT_CC3000_VBAT  5

#define ADAFRUIT_CC3000_CS    10

// Use hardware SPI for the remaining pins

// On an UNO, SCK = 13, MISO = 12, and MOSI = 11

Adafruit_CC3000 cc3000 = Adafruit_CC3000(ADAFRUIT_CC3000_CS,
ADAFRUIT_CC3000_IRQ, ADAFRUIT_CC3000_VBAT,

                                         SPI_CLOCK_DIVIDER); // you can change this clock speed


#define WLAN_SSID       "HUAWEI-mFui"           // cannot be longer than 32
characters!
```

#define WLAN_PASS "xxxxxxx" // enter the password of the Wi-Fi router

// Security can be WLAN_SEC_UNSEC, WLAN_SEC_WEP, WLAN_SEC_WPA or WLAN_SEC_WPA2

#define WLAN_SECURITY WLAN_SEC_WPA2

#define IDLE_TIMEOUT_MS 3000 // Amount of time to wait (in milliseconds) with no data

// received before closing the connection. If you know the server

// you're accessing is quick to respond, you can reduce this value.

const unsigned long

dhcpTimeout = 60L * 1000L, // Max time to wait for address from DHCP

connectTimeout = 15L * 1000L, // Max time to wait for server connection

responseTimeout = 15L * 1000L; // Max time to wait for data from server

// Page to fetch

#define WEBSITE "192.168.100.2"

#define WEBPAGE "/server.php"

```
uint32_t t;

// Local server IP, port, and repository (change with web server settings)

uint32_t ip = cc3000.IP2U32(192,168,100,2);  //Looking to webserver

void setup(void)
{

  Serial.begin(115200);

  SWarray[0].start();
  result = "";

  pinMode(leftMtrSpdPin, OUTPUT);
  pinMode(leftMtrDirPin, OUTPUT);
  pinMode(rightMtrDirPin, OUTPUT);
  pinMode(rightMtrSpdPin, OUTPUT);
  pinMode(LEDStatPin, OUTPUT);
  pinMode(LEDDataReceivedPin, OUTPUT);
  pinMode(LEDMotortest, OUTPUT); //added
```

```
Serial.println(F("Hello, CC3000!\n"));

Serial.print("Free RAM: "); Serial.println(getFreeRam(), DEC);

/* Initialise the module */
Serial.println(F("\nInitializing..."));
if (!cc3000.begin())
{
  Serial.println(F("Couldn't begin()! Check your wiring?"));
  while(1);
}

// Optional SSID scan
// listSSIDResults();

Serial.print(F("\nAttempting to connect to ")); Serial.println(WLAN_SSID);
if (!cc3000.connectToAP(WLAN_SSID, WLAN_PASS, WLAN_SECURITY)) {
  Serial.println(F("Failed!"));
  while(1);
}
```

```
  Serial.println(F("Connected!"));


 // Check DHCP
  Serial.println(F("Requesting address from DHCP server..."));
  for(t=millis(); !cc3000.checkDHCP() && ((millis() - t) < dhcpTimeout);
delay(1000));
  if(cc3000.checkDHCP()) {
    Serial.println(F("OK"));
  } else {
    Serial.println(F("failed"));
    return;
  }


  /* Display the IP address DNS, Gateway, etc. */
  while (! displayConnectionDetails()) {
    delay(1000);
  }
}
void loop() {
```

```
 //Time of sending of date should match that of the polling frequency that is set for data display in WiFi Bot Control

 //If polling the data every 5 seconds, this information should be sent only every 5 seconds.


 //String timeSec = String((int) SWarray[0].elapsed()/1000);

 String timeSec = String((int) counter);

 String strDataToSend = "GET /arduinoPush.php?URLresult1=11&URLresult2=" + timeSec + " HTTP/1.1\r\nHost:" + WEBSITE + "\r\n";


  //START Sending data to WiFi Bot Control example

    Adafruit_CC3000_Client wwwSendData = cc3000.connectTCP(ip, port);

    if (wwwSendData.connected()) {

      wwwSendData.println(strDataToSend);

      wwwSendData.fastrprint(F("\r\n"));

        wwwSendData.println();

      Serial.println(F("COnnection successful,data sending"));

    }

    else

    {

      Serial.println(F("Data Send - Connection failed"));

      return;
```

```
  }
  wwwSendData.close();


  while (wwwSendData.available()) {
    wwwSendData.read();
  }
  //end send data out~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
 //END Sending data to WiFi Bot Control example



//Get data from WiFi Bot Control joystick movement.
 Adafruit_CC3000_Client www = cc3000.connectTCP(ip, port);
 if (www.connected()) {
  www.fastrprint(F("GET "));
  www.fastrprint(WEBPAGE);
  www.fastrprint(F(" HTTP/1.1\r\n"));
  www.fastrprint(F("Host: ")); www.fastrprint(WEBSITE); www.fastrprint(F("\r\n"));
  www.fastrprint(F("\r\n"));
  www.println();
  Serial.println(F("Data Receive - Connection successful"));
```

```
  } else {
    Serial.println(F("Data Receive - Connection failed"));
    return;
  }



  /* Read data until either the connection is closed, or the idle timeout is reached. */
  unsigned long lastRead = millis();
  while (www.connected() && (millis() - lastRead < IDLE_TIMEOUT_MS)) {
  //while (www.connected()) {
    while (www.available()) {
      char c = www.read();
      result = result + c;
      analogWrite(LEDDataReceivedPin, 150);  //flash LED to show data being
received.

      // Delete HTTP headers
      //if(result.endsWith("Content-Type: text/html"))
      result.toLowerCase();
      if(result.endsWith("content-type: text/html"))
```

```
      {
        //if true, it will essentially clear out all the header content that came before
        //the actual motor values x,x,x,x,  later this is also trimmed as there are spaces
around it
        result="";
      }
    }
    analogWrite(LEDDataReceivedPin, 0);
  }
  www.close();



 // Format result and extract the variables
 format_result(motorCommand,result);

 //blink LED
 digitalWrite(LEDStatPin, HIGH);
 delay(50);
 digitalWrite(LEDStatPin, LOW);
 delay(50);
```

```
 // Print received values
 Serial.println("Result: " + String(result));
 Serial.println("Motor 1 speed: " + String(motorCommand[0]) + " and direction: " + 
String(motorCommand[2]));
 Serial.println("Motor 2 speed: " + String(motorCommand[1]) + " and direction: " + 
String(motorCommand[3]));
 Serial.println("Mode: " + String(motorCommand[0]));
 Serial.println("Motor 1 speed/dir: " + String(motorCommand[1]));
 Serial.println("Motor 2 speed/dir: " + String(motorCommand[2]));
 Serial.println("UDLRVal: " + String(motorCommand[3]));
 Serial.println("CommandVal: " + String(motorCommand[4]));
 // Send motor commands
send_motor_command(speed_motor1,direction_motor1,motorCommand[0],motorCo
mmand[2]);
send_motor_command(speed_motor2,direction_motor2,motorCommand[1],motorCo
mmand[3]);
send_motor_command(leftMtrSpdPin,leftMtrDirPin,motorCommand[0]);
send_motor_command(rightMtrSpdPin,rightMtrDirPin,motorCommand[1]);
 //flip through the mode values and react accordingly
 switch (motorCommand[0]) {
   case 1:
     //driving via joystick x,y values
```

```
  if (motorCommand[1] != 0 && motorCommand[2] != 0)
   {
     driveMotors(motorCommand[1],motorCommand[2]);

   }
   else
   {
    analogWrite(leftMtrSpdPin, 0);   //drive the motor
    analogWrite(rightMtrSpdPin, 0);  //drive the motor
   }
  break;
case 2:
  //driving by simple joystick U D L R

  if (motorCommand[3] != 0)
   {
     //move motors according to command

   }
```

```
    break;
   case 3:
    //sending command values
    //write your own code here to react to command values 1 - 8
    break;
   default:
      analogWrite(leftMtrSpdPin, 0);   //Stop the motor
      analogWrite(rightMtrSpdPin, 0);  //Stop the motor
    break;
  }
 // Reset result variable
 result = "";
counter ++;
}
//######################################################################
######################################################

//parse the content and assign the mode, motor and command values
void format_result(int* array, String result) {

 result.trim();
```

```
//remove the last comma
//if (result.endsWith(","))
//{
//result = result.substring(0, result.length()-1);
//}
// result = result.substring(0, result.lastIndexOf(','));
resultLength = result.length();
//Serial.println(result);
//Serial.print(" result length = ");
//Serial.println(resultLength);


int commaPosition;
int i = 0;
do
 {
    commaPosition = result.indexOf(',');


    if(commaPosition != -1)
    {
       //Serial.println( result.substring(0,commaPosition));
```

```
        array[i] = result.substring(0,commaPosition).toInt();


        //Serial.print("CommaPos: " );
        //Serial.print (commaPosition);
        //Serial.print("  MotorVals: " );
        //Serial.print(array[i]);
        i = i+1;
        result = result.substring(commaPosition+1, result.length());
        //Serial.print("  NextResult: " );
        //Serial.println(result);
      }
      else
      {
        if(result.length() > 0) {
          Serial.println(result);
        }
      }
    }
    while(commaPosition >=0);
  }
```

```
void driveMotors(int xVal, int yVal)
{
  float xPct=1.0;
  int xAdj, yAdj;
  xAdj = map(abs(xVal), 0, 100, 100, 255);
  yAdj = map(abs(yVal), 0, 100, 100, 255);

  //approach -
  //if Y is positive, both motors are set to move forward, else reverse. Left and Right from center will determine the speed of each motor.  At extremes will reverse each motor for fast turns
  //the value for X will determine the relative speed of each motor.

  //first determine the direction
  if (yVal >= 0)
  {
    //both motors moving fwd
    digitalWrite(leftMtrDirPin, HIGH);
    digitalWrite(rightMtrDirPin, HIGH);
```

```
}
 else
{
 //both reversed
 digitalWrite(leftMtrDirPin, LOW);
 digitalWrite(rightMtrDirPin, LOW);
}


//now determine left / right

if (xVal <= 0)
{
  if (xVal < -70 && (yVal <= 40 || yVal >= -40))  //fast turn
  {
    digitalWrite(leftMtrDirPin, LOW);   //reverse the motor = faster turn
    leftSpeed = 150;  //something fast, but not too crazy
    rightSpeed = 150;
  }
  else
  {
```

```
    leftSpeed = (float)yAdj * ((float)(100 - abs(xVal)) / 100);
    rightSpeed = yAdj;
  }
}
else
{
  if (xVal > 70 && (yVal <= 40 || yVal >= -40))  //fast turn
  {
    digitalWrite(rightMtrDirPin, LOW); //reverse the motor = faster turn
    leftSpeed = 150;
    rightSpeed = 200; //something fast, but not too crazy
  }
  else
  {
    leftSpeed = yAdj;
    rightSpeed = (float)yAdj * ((float)(100 - xVal) / 100);
  }
}

 //Serial.print("yAdj: ");
 //Serial.print(yAdj);
```

```
 //Serial.print("  xVal: ");
 //Serial.print(xVal);
 //Serial.print("  leftspeed: ");
 //Serial.print(leftSpeed);
 //Serial.print("    rightspeed: ");
 //Serial.println(rightSpeed);
 //Serial.print("     100-lesvxal:");
 //Serial.print(100 - abs(xVal));
 //Serial.print("  100-lesvxal/100: ");
 //Serial.println((100 - abs(xVal))/100, DEC);


 //drive the motors
 analogWrite(leftMtrSpdPin, (int)leftSpeed);
 analogWrite(rightMtrSpdPin,(int)rightSpeed);


//set the motors to off
//analogWrite(leftMtrSpdPin, 0);    //drive the motor
//analogWrite(rightMtrSpdPin, 0);   //drive the motor
 //slight delay
delay(50);
leftSpeed=0, rightSpeed=0, xAdj=0, yAdj=0;
```

```
}

/*
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~ */

//use to map values to float.
float mapf (float x, float in_min, float in_max, float out_min, float out_max)
{
  return (x - in_min) * (out_max - out_min) / (in_max - in_min) + out_min;
}



bool displayConnectionDetails(void)
{
  uint32_t ipAddress, netmask, gateway, dhcpserv, dnsserv;

  if(!cc3000.getIPAddress(&ipAddress, &netmask, &gateway, &dhcpserv, &dnsserv))
  {
    Serial.println(F("Unable to retrieve the IP Address!\r\n"));
    return false;
  }
```

```
  else
  {
    Serial.print(F("\nIP Addr: ")); cc3000.printIPdotsRev(ipAddress);
    Serial.print(F("\nNetmask: ")); cc3000.printIPdotsRev(netmask);
    Serial.print(F("\nGateway: ")); cc3000.printIPdotsRev(gateway);
    Serial.print(F("\nDHCPsrv: ")); cc3000.printIPdotsRev(dhcpserv);
    Serial.print(F("\nDNSserv: ")); cc3000.printIPdotsRev(dnsserv);
    Serial.println();
    return true;
  }
}
```

Appendix J

Autonomous line following programming

Line following programming

```
#include <QTRSensors.h>  // Pololu QTR Library



//Defining line sensor

#define NUM_SENSORS   6     // number of sensors used

#define TIMEOUT       2500  // waits for 2500 microseconds for sensor outputs to go
low

#define EMITTER_PIN   2     // emitter is controlled by digital pin 2



//Line sensor declarations

//Sensors 1 through 6 are connected to digital pins 2 through 10, respectively (pin 3 is
skipped and used for motor control)

QTRSensorsRC qtrrc((unsigned char[]) {2, 4, 5, 6, 7, 8}, NUM_SENSORS,
TIMEOUT, EMITTER_PIN);

unsigned int sensorValues[NUM_SENSORS];


int pwm_a = 3;  //PWM control on digital pin 3

int pwm_b = 11;  //PWM control on digital pin 11

int dir_a = 9;  //direction control on digital pin 9

int dir_b = 10;  //direction control on digital pin 10
```

```
// PID loop variables

float error=0;

float lastError=0;

float PV =0 ;

float kp = 0;

float ki = 0;

float kd =0;

int m1Speed=0;

int m2Speed=0;

int line_position=0;

int motorspeed=0;




//Pin which triggers ultrasonic sound

const int pingPin = 13;


//Pin which delivers time to receive echo using pulseIn()

int inPin = 12;
```

```
//Range in cm to detect object
int safeZone = 5;


int redLed = 1;


void setup()
{
  pinMode(pwm_a, OUTPUT);  //Set control pins to be outputs
  pinMode(pwm_b, OUTPUT);
  pinMode(dir_a, OUTPUT);
  pinMode(dir_b, OUTPUT);

  analogWrite(pwm_a, 0);  //set both motors to run at (100/255 = 39)% duty cycle
(slow)
  analogWrite(pwm_b, 0);




  Serial.begin(115200);
```

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

  // Calibrate line sensor

  delay(500);

  pinMode(13, OUTPUT);

  digitalWrite(13, HIGH);    // turn on Arduino's LED to indicate we are in calibration
mode

  for (int i = 0; i < 400; i++)  // make the calibration take about 10 seconds

  {

    qtrrc.calibrate();       // reads all sensors 10 times at 2500 us per read (i.e. ~25 ms per
call)

  }

  digitalWrite(13, LOW);     // turn off Arduino's LED to indicate we are through with
calibration


  // print calibration results


  // print the calibration minimum values measured when emitters were on

  for (int i = 0; i < NUM_SENSORS; i++)

  {

    Serial.print(qtrrc.calibratedMinimumOn[i]);

    Serial.print(' ');
```

```
  }
  Serial.println();

  // print the calibration maximum values measured when emitters were on
  for (int i = 0; i < NUM_SENSORS; i++)
  {
    Serial.print(qtrrc.calibratedMaximumOn[i]);
    Serial.print(' ');
  }
  Serial.println();
  Serial.println();
  Serial.println("Calibration Complete");
  delay(1000);

}

void loop()
{

  // Line sensor values
```

```
  // Read calibrated sensor values and obtain a measure of the line position from 0 to
5000
  // To get raw sensor values, call:
  //  qtrrc.read(sensorValues);
  unsigned int line_position = qtrrc.readLine(sensorValues);


  // Print the sensor values as numbers from 0 to 1000, where 0 means maximum
reflectance and
  // 1000 means minimum reflectance, followed by the line position
  for (unsigned char i = 0; i < NUM_SENSORS; i++)
  {
 //   Serial.print(sensorValues[i]);
 //   Serial.print('\t');
  }
  //Serial.println(); // uncomment this line if you are using raw values
  Serial.println(line_position); // comment this line out if you are using raw values


  //delay(250);


//raw duration in milliseconds, cm is the converted amount into a distance
  long duration, cm;
```

```
//initializing the pin states
pinMode(pingPin, OUTPUT);
pinMode(redLed, OUTPUT);

//sending the signal, starting with LOW for a clean signal
digitalWrite(pingPin, LOW);
delayMicroseconds(2);
digitalWrite(pingPin, HIGH);
delayMicroseconds(5);
digitalWrite(pingPin, LOW);

//setting up the input pin, and receiving the duration in
//microseconds for the sound to bounce off the object infront
pinMode(inPin, INPUT);
duration = pulseIn(inPin, HIGH);

// convert the time into a distance
cm = microsecondsToCentimeters(duration);

if (cm > safeZone)
{
```

```
    follow_line(line_position);
  }
  else
  {
    analogWrite(pwm_a, 0);
    analogWrite(pwm_b, 0);
    digitalWrite(redLed, HIGH);

  }

  delay(10);

}  // end loop

//Line following subroutine
```

```
// PD Control

void follow_line(int line_position)  //follow the line

{

switch(line_position)

{

//Line has moved off the left edge of sensor

case 0:

       digitalWrite(dir_a, LOW);

       analogWrite(pwm_a, 200);

       digitalWrite(dir_b, HIGH);

       analogWrite(pwm_b, 200);

       Serial.println("Rotate Left\n");

break;
```

```
// Line had moved off the right edge of sensor

case 7000:

      digitalWrite(dir_a, HIGH);

       analogWrite(pwm_a, 200);

       digitalWrite(dir_b, LOW);

       analogWrite(pwm_b, 200);

       Serial.println("Rotate Right\n");

break;


 default:

   error = (float)line_position - 3500;


  // Set the motor speed based on proportional and derivative PID terms

  // kp is the a floating-point proportional constant (maybe start with a value around
0.5)

  // kd is the floating-point derivative constant (maybe start with a value around 1)

  // When doing PID, signs should be correct, else control loop is unstable


  kp=.5;

  kd=1;
```

```
PV = kp * error + kd * (error - lastError);
lastError = error;

//This code limits the PV (motor speed pwm value)
// limit PV to 55
if (PV > 55)
{
  PV = 55;
}

if (PV < -55)
{
  PV = -55;
}


m1Speed = 200 + PV;
m2Speed = 200 - PV;

   //Set motor speeds
```

```
      digitalWrite(dir_a, LOW);

      analogWrite(pwm_a, m2Speed);

      digitalWrite(dir_b, LOW);

      analogWrite(pwm_b, m1Speed);

    break;

}

} // end follow line



long microsecondsToCentimeters(long microseconds)

{

  // The speed of sound is 340 m/s or 29 microseconds per centimeter.

  // The ping travels out and back, so to find the distance of the

  // object we take half of the distance travelled.

  return microseconds / 29 / 2;

}
```

Appendix K

Cost Analysis

Cost Analysis

Item No.	Items	Unit price (Rs)	Quantity	Total price (Rs)
1	Ultrasonic sensor	45	1	45
2	2 pcs wheels	150	2	300
3	Adafruit CC3000 WiFi module	1095	1	1095
4	Arduino UNO	250	1	250
5	Arduino Nano	170	1	170
6	DC motors	330	2	660
7	Sricam IP camera	1300	1	1300
8	WiFi Router	400	1	400
9	L298N motor driver	90	2	180
10	SONY Power Bank	1000	1	1000
11	USB Hub	250	1	250
12	Alucobond sheets	200	1	200
13	Pololu reflectance sensor array	360	1	360
14	Aluminium angle plates	150	1	150
15	Li-Po Battery 11.1V	1200	1	1200

Appendix L

Record of Strategic Meeting with Supervisor

Strategic Meeting with supervisor

Student Name : Vayun Bhujan

Student ID : 1114714

Department : Department of Electrical and Electronics Engineering

Programme : BEng Mechatronics

Title of Dissertation : Design and implementation of a robot to aid bedridden persons

Supervisor : Dr. Robert Ah King

Project Coordinator : Mr. R. Jugurnath

- Your Progress Log serves as a record of your transferable skills and participation and attainment as a student for dissertation purposes.
- Its purpose is to help you to plan your own dissertation and to record the outcomes.
- As well as gaining valuable skills, you will find that the information accumulated in this Log will prove helpful during the write up of the dissertation.
- The document belongs to you and it is your responsibility to keep it up to date.
- It is your responsibility to ensure your supervisor is aware of the dissertation activities you have undertaken.

You should sign the appropriate statement below when you submit your Progress Log:

I confirm that the information I have given in this Log is a true and accurate record:

Signed: ……………………………………… Date: …………………..

Meetings	Date	Topics Discussed	Signature
1	19/08/2014	Project ideas	
2	05/09/2014	Literature Review	
3	08/10/2014	Literature Review +Concept Design +List of components	
4	02/12/2014	Mechanical Design +Programming Discussion	
5	08/12/2014	Mid-Year Report	
6	24/02/2015	Implementation of programming +Prototype discussion	
7	17/03/2015	Report +Programming	
8	25/03/2015	Report Discussion	

Supervisor ………………………… Signature …………………… Date…………….