WAYPOINT FOLLOWING INDOOR ROBOT

FHARIDATUN BINTI MOHAMMAD ZAIN

A project report submitted in partial fulfilment of the requirement for the award of the Degree of Master of Electrical Engineering

Faculty of Electrical and Electronic Engineering
University Tun Hussein Onn Malaysia

JANUARY 2015

ABSTRACT

In this project, a waypoint following indoor robot with predetermined set of waypoints is presented. For project purpose, a dead reckoning technique has been implemented to determine the current position of the robot. Optical encoders, which are mounted on both drive wheels count rotation of the wheels and give information for robot positioning. The results show the encoder ticks that have been calculated can determine angle of rotation and distance of movement for robot navigation. Besides that, the concept of ramp up and ramp down has been introduced for speed control in order to give a smooth movement for the robot. Moreover, since the waypoint following robot is specified in indoor environment, Wi-Fi is used to communicate between the Arduino Mega 2560 with the serial console, Putty for monitoring purpose. Hence, the robot can navigate through desired waypoint based on the method that has been proposed.

ABSTRAK

Projek ini membentangkan tentang sebuah robot yang berkebolehan untuk bergerak mengikut titik laluan yang telah ditentukan di dalam sebuah kawasan tertutup. Untuk mencapai objektif projek ini, satu teknik yang dipanggil "Dead Reckoning Technique" telah digunakan untuk menentukan posisi semasa robot tersebut. Sejenis alat pengesan yang bernama "enkoder optik" telah digunakan pada kedua-dua roda robot untuk mengesan kiraan putaran roda dan membantu menentukan kedudukan semasa robot dalam pergerakannya. Hasil kajian menunjukkan "kiraan enkoder" digunakan untuk menentukan sudut pergerakan dan jarak pergerakan robot ini. Selain itu, konsep tujahan menaik dan menurun diperkenalkan untuk mengawal kelajuan robot supaya memberi pergerakan yang lancar. Tambahan lagi, Wi-Fi juga digunakan sebagai medium penghubung antara robot dan komputer untuk tujuan pemantauan.

.

CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ABBREVIATIONS

d	Displacement
D	Separation distance between two wheels
R	Radius of the wheel
ω	Angular velocity
R_w	Radius of the wheels = 30mm
T	Number of encoder ticks in a full rotation of wheels = 333.33 ticks /revolution
T_1	Encoder ticks for right wheel
T_2	Encoder ticks for left wheel
$\Delta\theta$	Different between current angle and previous angle
CW	Clockwise direction
CCW	Counter clockwise direction
DIR	Direction
DOF	Degree of freedom
EKF	Extended Kalman filter
FOG	Fiber optic gyroscope
GIS	Global information system
GPS	Global positioning system
GUI	Graphical user interface
ICC	Instantaneous center of curvature
IDE	Integrated development environment
MAV	Micro air vehicles
PWM	Pulse width modulation
UKF	Unscented Kalman filter

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.0 Introduction

Navigation is a fundamental requirement of autonomous mobile robots. Referring to [1], navigation can be defined as a "scene of determining position, location, course, and distance travelled to a known destination". Navigation is used for the localization of the robot to do a required task. The localization is very important issue for the mobile robot to move autonomously to achieve the goal position in any environment. Mobile robot cannot execute all operations related with the navigation without the localization information. As the mobile robot moves around its environments, its actual position and orientation always differs from the position and orientation that it is commanded to hold. There are several types of method approach for the localization of the robot. For example trajectory planning, approaches in the presence of obstacle [2]. The goal of this method is to find the optimal of robot trajectory consisting the both path and the motion along the path which avoid the collision with moving obstacle. From previous research, some papers are widely used localization algorithm known as dead reckoning. Dead reckoning is a process of estimating the value of any variable quantity by using an earlier value and adding whatever changes have occurred in the meantime.

The study [3] shows waypoint following system were widely adopted as alternate mobile robot navigation technique after the used of advance, accurate and precision navigational system for example Global Positioning System (GPS) and

Global Information systems (GIS). Basically, the waypoints represent the sets of coordinates or checkpoints that provide the salient information for the identification of a point in any physical space. The waypoint had been decided by the users and through this set of co-ordinates provides the trajectory information to the robot to follow the route to reach the goal (destination) point. This system provides the route from starting point to the goal (destination) through waypoint depending on the destination (goal position) and trip plan. The authors in [3] claim that the route planning serves not only as a guide to reach the goal, but also serves as a route guide back to the starting point.

1.1 Statement of problems

Waypoint following indoor robot is one of the options for different types of robot navigation. The terms of localization refers to the process of determining a robot position by using information from external sensor (encoder). Recently, conventional method approaches using GPS for navigation. These methods however are not accurately giving best results for indoor environment due to barriers such as buildings or trees. Weak signal also affects the accuracy of the position [4]. Thus, this paper presents a low cost and efficient approach for estimating the position and orientation of the robot by using waypoint following system combining with dead reckoning algorithm for indoor environment. The main task of such mobile robot is to follow the specified set of waypoints to achieve the goal (destination) point.

1.2 Objectives

The objectives of this project are:

i. To develop a mobile robot equipped with encoders which is capable of following a specified set of waypoints.

ii. To determine the current location of the robot by using dead reckoning technique.

iii. To analyze the performance of the proposed technique in terms of robot position error.

1.3 Scope / limitation

This work presents the main scopes / limitation:

i) The robot is specifically to be used in indoor environment.

ii) The robot can be tested in a room with maximum size 100mx100m which depends on the specification of the router and Wi-Fi shield.

iii) In order to get high accuracy result, flat surface is used.

1.4 Organization of this report

The project report was organized into seven chapters. Introduction, overview on navigation and waypoint, problems of statement, research objectives, and scope are presented in Chapter 1.

Literature review on previous work on different aspects of the project for example the review of waypoint system and navigation of the robot are discussed in Chapter 2.

Chapter 3 presents the methodology includes the system descriptions, the overview of development of waypoint indoor robot. Besides that, the working principles for the systems were explained in this chapter and also provide the algorithm for the mobile robot navigation.

Chapter 4 presents the data collected through experiment that has been conducted and also discussion from the data obtained.

The last chapter 5 presents the conclusion and recommendation for future works.

CHAPTER 2

LITERATURE REVIEW

2.0 Introduction

This chapter start with the most important concept explaining the waypoint following system. Besides that, the discussion also focused on several methods used for navigation for mobile robot based on previous research paper and also its advantages and disadvantages. The related concept including the mathematical and theoretical knowledge will be reviewed here in order to be implemented in this project.

2.1 Waypoint following indoor robot using dead reckoning technique

Localization is one of the major tasks of autonomous robot navigation [5]. In indoor environment with the flat surface, localization becomes the issue of determining the coordinates (x, y) and orientation, θ of the robot [6]. To calculate the current position of the robot, the distance and the heading orientation from a known origin have to be measured. The set of waypoints that has been determined initially is used for the robot to follow. The dead reckoning method is applied here in order to accomplished this task. The dead reckoning basically is a simple method for mobile robot localization that integrates wheel translation and rotation to determine the robot's Cartesian location [7], [8]. While the encoder is used to count the number of rotation of the wheel where the information from that can be used to obtain the total distance travel by the robot. Since the current position can be calculated and updates using algorithm that will discuss later, the goal position can be achieve accordingly through the set of waypoints.

2.2 Related works

This section explained the previous works related with the project area.

2.2.1 Motion control for mobile robot navigation

Accurate position estimation is a key component to the successful operation of most autonomous mobile robots [9]. Generally, there are three phases comprises the motion of a mobile robot which are localization, path planning, and path execution. The position and orientation is determine using the external sensors. Different sensors such as GPS, ultrasonic and laser can be employed in absolute localization. The absolute localization is more complex and more accurate over long distance navigation [9]. The global path planning means finding a safe and short trajectory from the current location to a destination. The global path planning considers just fixed obstacles in the workspace map such as walls, but in the environment, the objects' locations may change. To achieve a collision-free navigation, the robot has to detect and avoid those obstacles. Mobile robot navigation with static obstacles is considered relatively simple because the robot needs to calculate only the distances to the obstacles [10]. The last phases is the execution of the planned path. This process is repeated so that the robot will remain on course towards the goal. Commonly, open-loop estimation (dead-reckoning) is used for intermediate estimation of position during path execution. Open-loop estimation is used because the encoders are available for motor control, which provides actual angular displacement of the wheels. However, due to errors in kinematic model parameters or wheel slip, poor position estimates may occur. Poor estimates in position during path execution require more frequent localizations to be made, incusing extra overhead and possibly slowing the movement of the robot. It is therefore important to minimize positional errors during the path execution phase.

2.2.2 Waypoint navigation system in indoor environment

Millington et al. [11] presents the navigation systems utilizes the waypoints information in a vehicle. The coordinates information used as a route while the waypoints and position of vehicle are displayed on the screen with the feedback systems for controlling the directions. Hamid et al.[12] presents another navigation system's implemented in a mobile robot in different way. GPS is used for navigation and sonar sensors for obstacles avoidance. Several researches used GPS for navigation and created the waypoints. The trajectory of robots are determined trough the series of waypoints and followed by navigating the next waypoints until destination position is reached. However, the used of GPS in indoor environment have some limitation due to the weak signal that reduces the accuracy of the navigation and also the barrier in the building [4]. A study by Amundson et al [13] shows another works for waypoint navigation. The system called as TripNav, where the robot sensor node navigate between position coordinates by following a path. In this works, the researchers have been developed a localization technique for positioning estimation by combining radio interferometric angle of arrival estimation with least square triangulation. Basically, triangulation is a process of determining the position of an object using bearings from known reference positions. Using least square triangulation method, a node can be determined its position with as little as two anchors with rapid result, low complexity, and still provides accurate position estimates from noisy bearing measurements. With the used of digital compass attached to the mobile robot platform, the heading of the robot can be estimated. The navigation is started with a selection waypoint range that specified how close the robot to a waypoint. The system has been designed to make a decision in order to reach waypoint by adjusting the waypoint range based on the speed and latency. The size of waypoint range will be reduced if the speed of the mobile robot node become slower and the size of waypoint range will be increased if the speed of mobile robot node become faster. The control speed is important in order to know the system reach the target point or not. Thus, this method give a rapid result and has sub meter accuracy for waypoint navigation system.

2.2.3 Correcting dead reckoning for mobile robot navigation

Jirawimut et al. [14] proposed a method for correcting dead reckoning parameters, which are heading and step size for a pedestrian navigation system. In this method, the compass bias error and the step size error can be estimated during the period that the global positioning system (GPS) signal is available. The errors are measured from the inputs of an extended Kalman filter. Then, the dead reckoning system has been implemented for correcting those parameters to improve the accuracy of position determination when the GPS signal is unavailable. The results show that the parameters can be estimated with reasonable accuracy. In addition, from the method proposed however helps to increase the positioning accuracy when the GPS signal is available.

Fiber optic gyroscopes (FOG's) is proposed by Chung, Ojeda & Borenstein [15] to improve the dead reckoning accuracy in mobile robot localization. The researchers have presented two methods which are calibration procedure for FOGs by compensating the static bias drift. This method however, required a series of test with the FOG mounted on precisely controlled rotary table. Hence, used the results to define the temperature error and produced third order polynomial calibration function. In addition, a linear error system model has been developed in order to implement indirect feedback Kalman Filter. From the results show that the gyro calibration is most efficient hence give an accurate result for improvement of dead reckoning.

2.2.4 Wireless Ethernet for indoor localization

Many mobile robots already used wireless for communication and has been deployed in indoor environment. Ladd et al [16] proposed another tool used for robot localization which is wireless Ethernet adapters. While communicating with IEEE 802.11b. wireless Ethernet, the robot could measure signal strength for the wireless and engage for localization depending on the range of wireless coverage. From this papers, the researchers have proven that this sensor can be used for location sensing and tracking about one meter precision in wireless enabled office building. However, there are limitations for this method which are the accuracy of localization attempts caused by poor initially the base station and behaviour of the signal in dynamic environments. This is caused by the absorption signal into human body and also heavy traffic light during daytime. Therefore, researchers recommended the performance of the localization methods can be increased by improving the number of base station hence efficiently increase the range of wireless coverage area.

2.3 Differential drive kinematics

A mobile robot or vehicle has 6 degrees of freedom (DOF) which composed into two parts: the position (*x, y, and z*) and attitude (*Roll, Pitch, and Yaw*). Roll basically can be defined as sidewise rotation, Pitch for the rotation forward or backwards while Yaw commonly denoted heading or orientation which refers to the direction in which the robot moves in the x-y plane. In order to determine the current position (*x', y' and θ'),* the forward kinematics equations can be applied as mention in [17]. Hellström, T. [18] explained the central concept for the derivation of the kinematics equation for robot position. Refer to the Fig. 2.1 below, when the robot with two wheels are rotating, each wheel must roll along its own y axis, thus a common center point for rotation must exist. This point is called as Instantaneous Center of Curvature (ICC) [18][19].

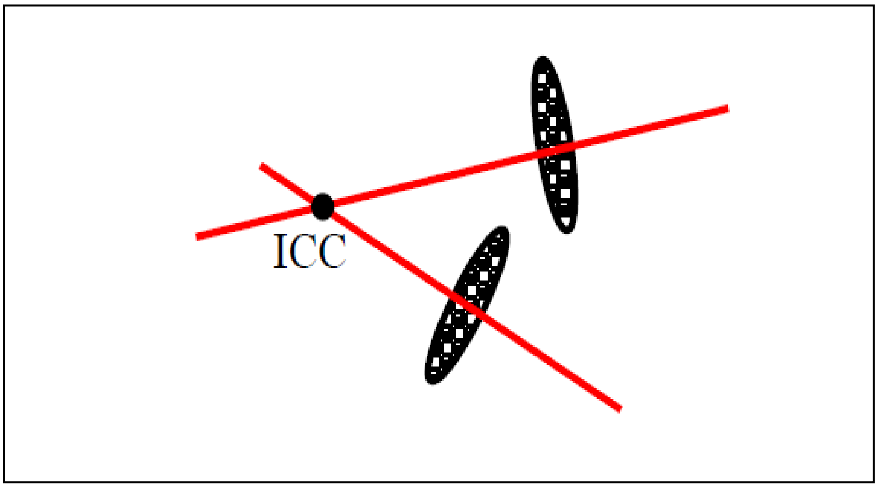

Figure 2.1: ICC (Instantaneous Center of Curvature) point (Hellström T.,2011)

For the concept of two wheels with the differential drive, a pair of wheel is mounted on a common axis. When the wheels are rotating on the ground by assuming there is no slipping, the point of ICC is around with both wheels. By varying the speed of both sides of wheels, the robot rotates around a common point denoted ICC and the trajectories of the robot will differ. Fig. 2.2 shows the details concepts.

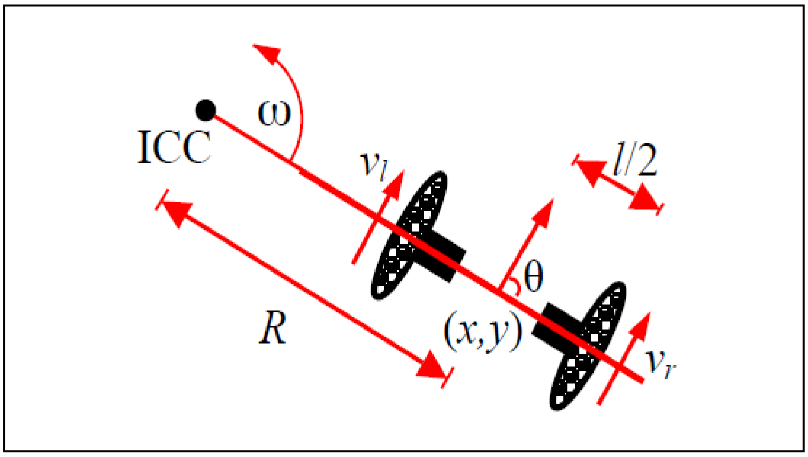

Figure 2.2: When left and right wheel rotate with different speeds, the robot rotates around a common point denoted ICC (Hellström T., 2011)

The expression radius (R) and angular velocity (ω) are summarizing below and this information is necessary in order to solve the forward kinematics problem.

$$R = \frac{l}{2}\left[\frac{(v_l +)v_r}{(vl - vr)}\right] \tag{2.1}$$

$$\omega = \frac{(vl - vr)}{l} \tag{2.2}$$

For situation when the robot is rotates around the ICC with angular velocity (ω) for certain time, the heading of angle will also change.

$$\theta' = \omega\delta t + \theta \tag{2.3}$$

Consequently, the center of rotation ICC also can be compute by using basic trigonometry equation:

$$ICC = [ICCx, ICCy] = [x - R\sin\theta, y + R\cos\theta] \tag{2.4}$$

The theory of differential drive commonly used for solving problem related to the angle of rotation and direction of movement. By understanding the theory of differential drive, this project come out with the mathematical model for driving the waypoint following indoor robot.

Table 2.1 below shows the summary from several previous papers related with the project title " Waypoint Following Indoor Robot".

Table 2.1 : Summary from previous research papers

No	Project Title	Objectives	Method Used	Advantages
1.	Yamauchi, B. (1996). Mobile Robot Localization In Dynamic Environments Using Dead Reckoning and Evidence Grids. *Proc. of the 1996 IEEE Int. Conf. on Robotics and Automation*. Minneapolis, Minnesota. pp. 1401 - 1406.	- To determine the amount of dead reckoning error accumulated in the time elapsed between the construction of the two grid.	- Used evidence grid method for a mobile robot's location	- More accurate results for robot position estimation.
2.	Briod, A. (2008). Waypoint Navigation With Air Vehicle (MAV). pp. 1-49.	- To implement a waypoint navigation using GPS as the position input for an outdor MAV.	- Used Dubin's Theory to generate intermediate homing waypoints for optimal path.	- Precise results for trajectories on MAV system by implying limited turning rates for GPS.
3.	Wang, Y., Mulvaney, D., Sillitoe, I. & Swere, E. (2008). Robot Navigation By Waypoints. *Journal Intell Robot Systems. vol.52*. pp. 175 -207.	- To generate waypoints for robot movement in specified environment.	- Novel waypoint based - robot navigation combines reactive and deliberative actions.	- The technique proposed reduces the time taken to produce a plan. - Able to reduce the search from the initially large physical are to a small number of representative points only for waypoint navigation.

Table 2.1: (continued)

4.	Ojeda, - Chung, H., Ojeda, L. & Borenstein, J. (2001). Accurate Mobile Robot Dead Reckoning With A Precision Calibrated Fibre Optic Gyroscope. *IEEE Transaction On Robotics and Automation. 17(1).* pp 80-84.	- To improve dead reckoning accuracy based on odometry and fiber- optic gyroscopes (FOGs) in mobile robot.	- Two methods approached: i) A meticulous calibration procedure for FOGs. ii) Indirect feedback Kalman Filter	- Reduced the ill effects of non-linearity of scale factor of FOG reading and temperature dependency. - Provide the large accuracy improvement when indirect Kalman Filter has been applied..
5.	Amundson, I., Sallai, J., Koutsoukos, X. , & Ledeczi, A. (2012). Mobile Sensor Waypoint Navigation Via RF-Angle of Arrival Localization. *International Journal of Distributed Sensor Networks*. pp. 1-14	- To implement the used of resource - constrained sensors for navigation system by using triangulation technique combining with radio interferometric angle of arrival estimation.	- Used triangulation method for robot localization and Kalman Filter to estimate the heading angle.	- Accurately perform waypoint navigation system by using a resource-constrained mobile sensor with an average position error of 0.95 m. - Provide a rapid and distributed results for navigation system.
6.	Khan, S., Ahmad, K., Murad, M., & Khan, I. (2013). Waypoint Navigation System Implementation via a Mobile Robot Using Global Positioning System (GPS) and Global System for Mobile Communications (GSM) Modems. *International Journal of Computational Engineering Research. 3(7).* pp. 49 - 54.	- To design a low cost and efficient approach for mobile robot vehicle system. - To navigate the robot using GPS by utilizes the used of Google Maps for visual system.	- Used GPS as geographic information and navigation system. - Sensors (LV-Maxsonar -EZ4) for obstacle avoidance system. - GSM modem for communication purpose.	- The used of GPS is more simple , highly accurate and weather proof for robot navigation based on outdoor environment.

Table 2.1: (continued)

7.	Zhang, Y., Wu, C., Cheng, L., & Chu, H. (2012). Localization and Tracking of Indoor Mobile Robot with Ultrasonic and Dead-reckoning Sensors. *Journal of Computational Information Systems.8(2).* pp. 531–539.	- To design algorithm for robot position estimation by using ultrasonic and dead reckoning sensors. - To implement the algorithm in indoor environment.	- Used ultrasonic sensor combining with dead reckoning sensors and applied unscented Kalman Filter (UKF) for position estimation algorithm. - Use trilateration method for position estimation algorithm.	- Used a low cost dead reckoning sensors which are encoder and compass for robot position. - The proposed method effectively improves the accuracy and flexibility of localization.
8.	Tsai, C.C. (1998). A Localization System of a Mobile Robot by Fusing Dead-Reckoning and Ultrasonic Measurements. *IEEE Instrumentation and Measurement Technology Conf.* Minnesota, USA. pp 144 - 149.	- To develops a novel system hardware structure and systematic digital signal processing algorithms for self-localization of an autonomous mobile robot.	- Used fusing dead-reckoning based on optimal filtering by fusing heading reading from a magnetic compass, a rate gyroscope and two encoders for computing the dead reckoned location estimation. - Used four ultrasonic timeof- flight (TOF) combined with dead reckoned location information fused to update the robot position by utilizing the extended Kalman Filter (EKF).	- The proposed algorithms are implemented by using a host PC 586 computer and standard C++ programming techniques. - The system prototype give a high accuracy result with strength magnetic interference immunity. - Provide simple and practical structure of the used and installation / calibration.

CHAPTER 3

METHODOLOGY

3.0 Introduction

The details of the system are identified before implementation of waypoint navigation can be done. Thus, three parts of the systems which are hardware, software and firmware have been specified and integrated before the system is completed. Hardware consists of wheel, motor, sensor, motor driver, and the most important is controller. The controller acts as a brain for the robot to be programmed and process the instructions and doing its task. Arduino IDE is a type of software which is compatible with Arduino Mega 2560 microcontroller. The Arduino IDE has its own serial monitor and thus can be used for monitoring the reading of encoder, robot movement and etc. Besides that, the language reference for this type of software is easier to learn especially for the beginner. Since this project focus on navigation system in indoor environment, the methodology for the system structured and implementation will become apparent throughout this chapter.

3.1 System description

Fig. 3.1 shows the flowchart for waypoint following indoor robot. The system is started with the predetermined waypoints by the user. Then, the microcontroller will calculate the angle of rotation and distance of movement to reach the desired waypoints. The angle and distance are represented by the encoder ticks for both wheels. The current position for the robot is updated trough monitoring PC. A mission is accomplished when the robot reaches at the target point.

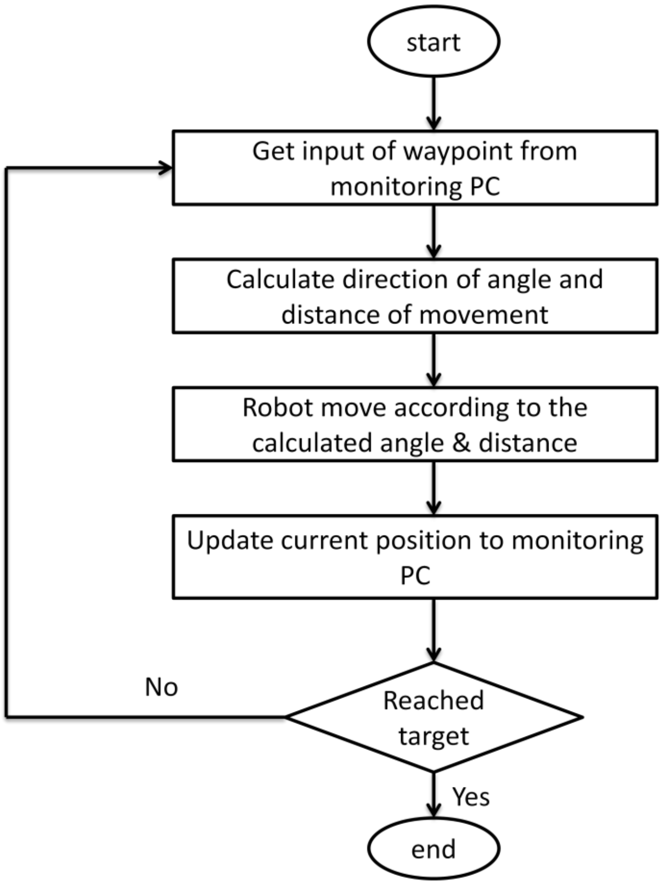

Figure 3.1 : Flowchart for waypoint following indoor robot

The control system block diagram, as shown in Fig. 3.2, is important in order to implement the controller into the robot. Arduino is used as the controller for the system in order to process the input signal and give an output signal as desired. For this project, the input signal is determined as goal position which consists of a set of waypoints while the output signal is determined as current position. Dc motor is used as an actuator. The waypoint following indoor robot need a sensor for giving the feedback to the microcontroller. The sensor which is an encoder is used to count the number of ticks. Then, the number of encoder ticks for both wheels is processed in the microcontroller before transmitted to the actuator. The number of ticks are actually used for the robot movement. Since the robot is moved from one waypoint to another according to the angle of rotation and distance of movement which is calculated from the encoder ticks, the current position can be determined and monitor from PC .

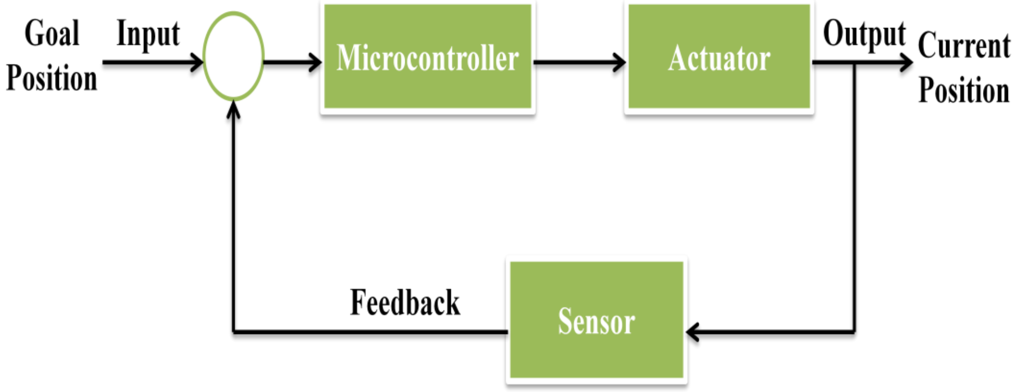

Figure 3.2 : Control system block diagram

3.2 Architecture system design

The architecture system design for this project can be divided into two parts, hardware design and software design. Hardware design which including selection for motors, wheels, sensor and also base for the robot. While software architecture system design including the selection software, language used for processing program and also algorithm for the robot to complete the task.

3.2.1 Hardware design

The hardware for this project consists of microcontroller board, encoder , motor, motor driver, router, and also Wi-Fi shield. The specification and working principle for each device are explained in this sub-section. Fig 3.3 shows the complete prototype for this project.

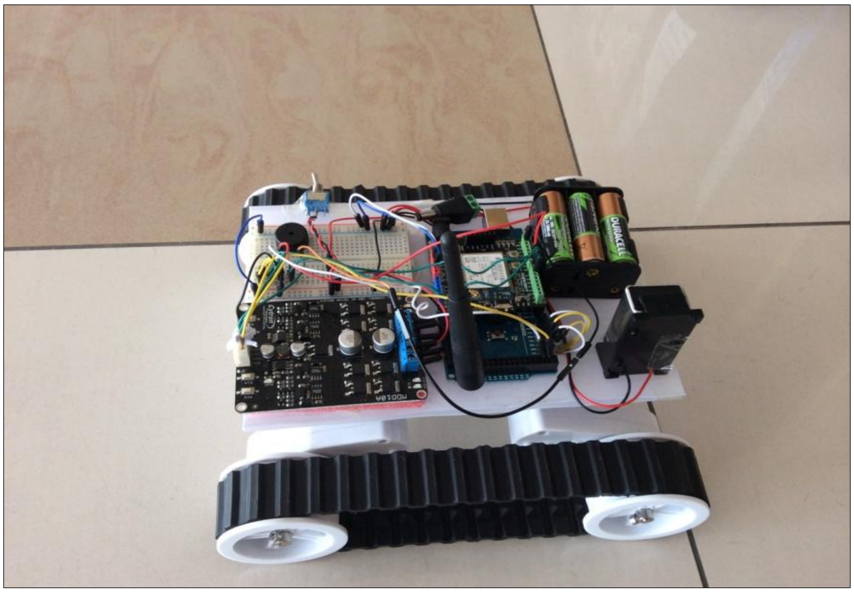

Figure 3.3: Prototype for the waypoint following indoor robot

3.2.1.1 Arduino Mega 2560 microcontroller

In this project, the Arduino Mega 2560 Microcontroller has been chosen as the "brain" for waypoint following indoor robot. As shown in Fig. 3.4, this type of microcontroller board is design based on the ATmega2560 as mention details the specifications in [20]. It has 54 for digital input/output pins. There are 15 pins can be used as PWM output, 16 analog input, 4 UARTs, a 16 MHz crystal oscillator, a USB connection, a power jack, and a reset button. It contains everything needed to support the microcontroller. The microcontroller can be connected to a computer via USB cable or power it with battery or AC-to-DC adaptor. This microcontroller can be operated with the external supply of 6 to 20 volt. However, the recommended range for voltage supply is 7 to 12 volts for stability. Besides that, it can be programmed as a USB to serial converter make the Arduino Mega 2560 is differs from preceding boards. The extra four pin for Interrrupt on Arduino Mega 2560 is another criteria for choosing this board rather than Arduino Uno. The Arduino microcontroller is pre-programmed with a boot loader that simplifies uploading the programs to the on chip flash memory compared with other devices that typically need an external programmer.

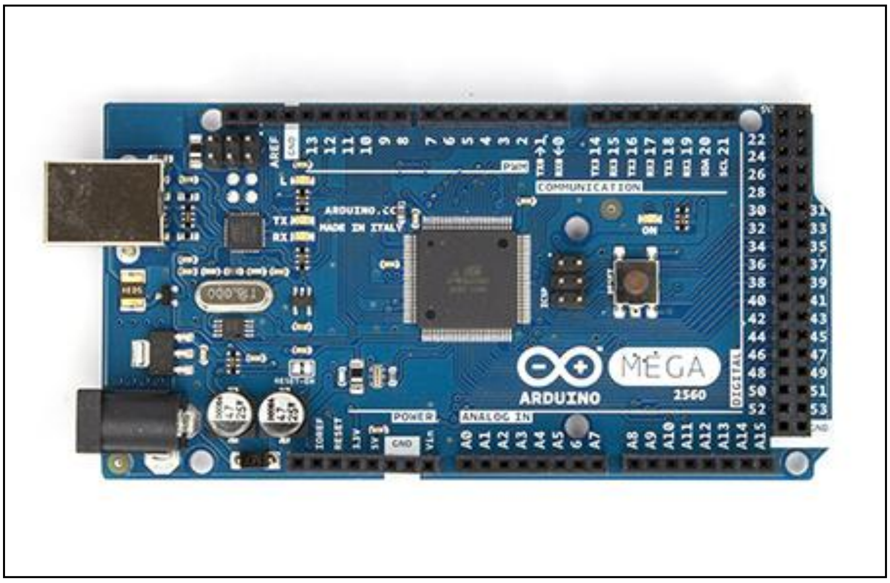

Figure 3.4: Arduino Mega 2560 microcontroller board

The detailed specifications of Arduino Mega 2560 microcontroller board are summarized in Table 3.1. Since Arduino Mega 2560 is used in this project, Table 3.2 shows the pin assignment and its function.

Table 3.1: Specification of Arduino Mega 2560 microcontroller board

Microcontroller	ATmega2560
Operating Voltage	5V
Input Voltage (Recommended)	7-12V
Input Voltage (Limit)	6-20V
Digital I/O Pins	54 (of which 15 provide PWM output)
Analog Input Pins	16
DC Current per I/O Pin	40 mA
DC Current for 3.3 pin	50mA
Flash Memory	256 KB of which 8KB used by boot loader
SRAM	8 KB
EEPROM	4 KB
Clock Speed	16 MHz

Table 3.2: Pin Assignment used on Arduino Mega 2560

Pin Assignment	Function
Pin 5 (PWM pin)	PWM Motor 1
Pin 7 (PWM pin)	PWM Motor 2
Pin 22 (Digital pin)	Direction Motor 1
Pin 24 (Digital pin)	Direction Motor 2
Pin 18 (TX1)	Communication
Pin 19 (RX2)	Communication
Pin 20 (SDA)	Communication
Pin 21 (SCL)	Communication
5V	Power Up Arduino
GND	Power Up Arduino

3.2.1.2 Encoder

Fig. 3.5 below shows an encoder inside of rover 5 gearbox. There are two IR sensors on the PCB that look at the black and white pattern on one of the gears. This quadrature encoder or known as an incremental rotary encoder measures the speed and direction of a rotating shaft. Borenstein et. al [21] stated that phase- quadrature incremental encoders can overcome the problem of single channel of tachometer encoder by adding a second channel, displaced from the first, so the resulting pulse trains are 90 degrees out of phase as shown in Fig 3.6.

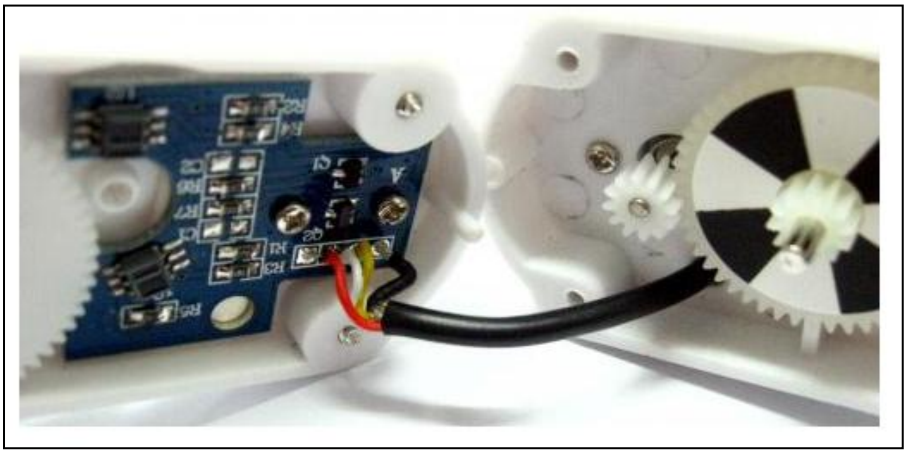

Figure 3.5 : Quadrature encoder

From this two output A and output B, the direction of motor rotation can be determined by looking the pattern of binary numbers generated by these two output depends on the direction of rotation either 0,1,3,2 or 0,1,2,3.

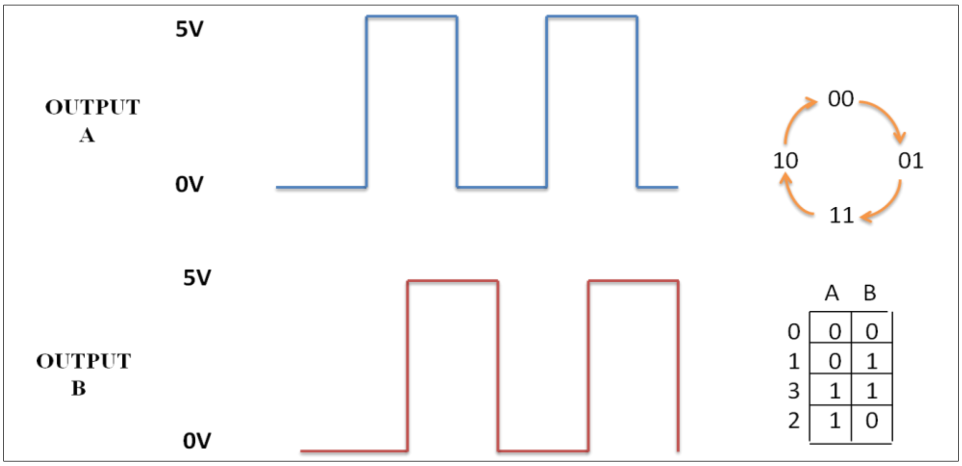

Figure 3.6: Encoder output waveform

The encoder ticks information however can be used to calculate the angle of rotation and distance of movement. In this project, the encoders used typically give 1000 pulses over 3 revolutions of the output shaft. It means, for one revolution, the encoder give the pulses around 333.33 ticks. Fig. 3.7 below shows the connection for both encoders to the Arduino Mega 2560 microcontroller board.

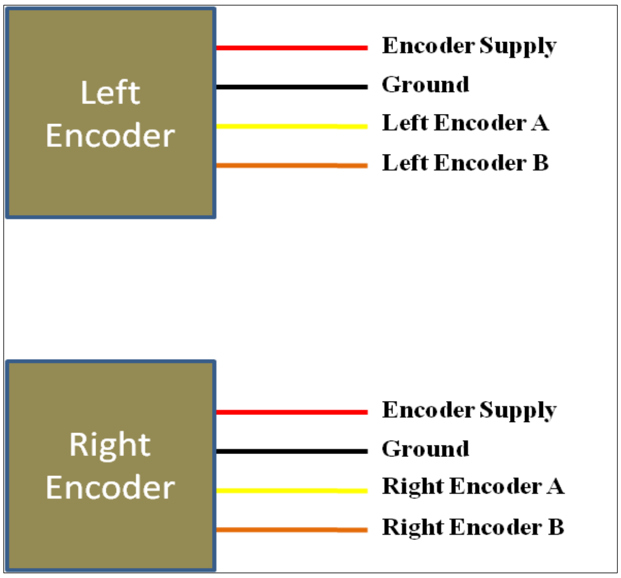

Figure 3.7: Connection for both encoder

3.2.1.3 Motor

The selection an appropriate motor for the project involves the analysis and the load for the performance of the mobile robot. The four important criteria needed to take note in the selection of the motor are torque, speed , power and energy. In order to drive the motor, a wheel of robot must produce enough torque to overcome any imperfection in the surface or wheel as well as the friction of the wheel. In this project, the dc motors are used. The dc motors are drive with the speed until 1km/hr. However, the adjustment of the speed can be made and controlled by PWM. In addition, the wheels connected to the motors are designed from rubber tread wheel. This type of wheel however suitable for any surface whether in indoor or outdoor environment. In order to drive the motor, the batteries 7.2V from NiMh is used since this batteries are last longer and have higher current output rather than alkaline batteries. The specifications for the dc motor are mention in the Table 3.3 below.

Table 3.3: Specification for the motor

Item	Specification
Motor rated voltage	7.2V
Stall current	2.5A
Output shaft stall torque	10kg/cm
Gear box ratio	86:8:1
Encoder type	Quadrature
Speed	1km/h
Encoder resolution	1000 state changes per 3 wheel rotations 1 rotation = 333.33 encoder ticks

3.2.1.4 MDD10A dual motor driver

Motor driver is used to drive both DC motors with its direction accordingly using Pulse Width Modulation (PWM). For given fixed load, a steady speed motor can be maintained by PWM. Thus, by changing the width of pulse applied to the DC motor, the amount of power provided to the motor can be controlled, thereby increasing or decreasing the motor speed.

PWM is widely used in the DC motor control. The Arduino microcontroller already comes with the PWM circuitry embedded in the chip. MDD10A is a type of motor driver that can drive 2 brushed DC motor with high current up to 10A continuously. This type of motor driver supports locked-anti phase and sign-magnitude PWM signal. For this project, PWM is operate in sign- magnitude PWM where two control signals are used to control the speed and direction of the motor. PWM is feed to the PWM pin to control the speed while DIR pin is used to control the direction of the motor. Table 3.4 shows the features for this motor driver.

Table 3.4: Features for MDD10A motor driver

MDD10A has been designed with the capabilities and features of :
✓ Bi- directional control for two brushed DC motors.
✓ Support motor voltage ranges from 5V to 25V.
✓ Maximum current up to 10A continuous and 30A peak (10 second) for each channel.
✓ Solid state components provide faster time and eliminate the wear and tear of mechanical relay.
✓ Fully NMOS H-Bridge for better efficiency and no heat sink is required.
✓ Speed control PWM frequency up to 20KHz.
✓ Support both locked -antiphase ad sign - magnitude PWM operation.
✓ Onboard push button to control the motor manually.

Each component on this motor driver board has their own function. Fig. 3.8 shows the board layout for MDD10A motor driver. In Table 3.5 shows the summary for each component and its function.

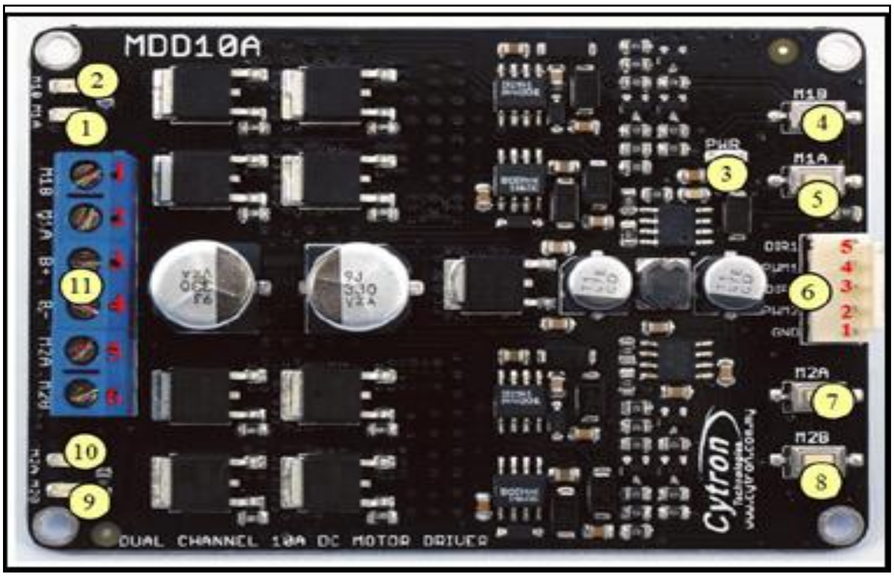

Figure 3.8: Board layout for MDD10A motor driver

Table 3.5: Summary for each components on MDD10A motor driver

No	Description
1	Red LED M1A: Turns on when the output M1A is high and output M1B is low. Indicates the current flows from output M1A to M1B.
2	Red LED MIB: Turns on when the output M1A is low and output M1B is high. Indicates the current flows from output M1B to M1A
3	Green LED: Power LED. Should be on when the board is powered on.
4	Test Button M1B: When this button is pressed, current flows from output M1B to M1A and motor will turn CCW (or CW depending on the connection).
5	Test Button M1A: When this button is pressed, current flows from output M1A to M1B and motor will turn CW (or CCW depending on the connection).
6	Input: Connect to PWM and DIR for both motor.
7	Test Button M2A: When this button is pressed, current flows from output M2A to M2B and motor will turn CW (or CCW depending on the connection).
8	Test Button M2B: When this button is pressed, current flows from output M2B to M2A and motor will turn CCW (or CW depending on the connection).
9	Red LED M2B: Turns on when the output M2A is low and output M2B is high. Indicates the current flows from output M2B to M2A.
10	Red LED M2A: Turns on when the output M2A is high and output M2B is low. Indicates the current flows from output M2A to M2B.
11	Terminal Block: Connect to motor and power source.

Fig. 3.9 indicates the connection for motor driver MDD10A with the microcontroller and motors. The input component is connected to the pin that has been assigned on Arduino Mega 256 microcontroller while the terminal block is connected to the both motors according to the pin assignment on the motor driver. In order to run this motor driver, the battery used is 7.2v. Basically, there are two wires for each motor indicates the signal for direction of motors. For example, to turn on motor1, PWM and DIR is in HIGH mode. Output A is in LOW position while output B is in HIGH position. As a result, motor1 will turn clockwise (CW) (depending on the connection) with high speed. Same condition for clock wise direction (CCW). Both motor1 and motor2 must be in HIGH mode for PWM and DIR as well as output A (LOW) and output B (HIGH). However, the motor is in LOW position (STOP) when PWM is LOW without concerning the DIR status. Table 3.6 shows the truth table for the control logic for the motor.

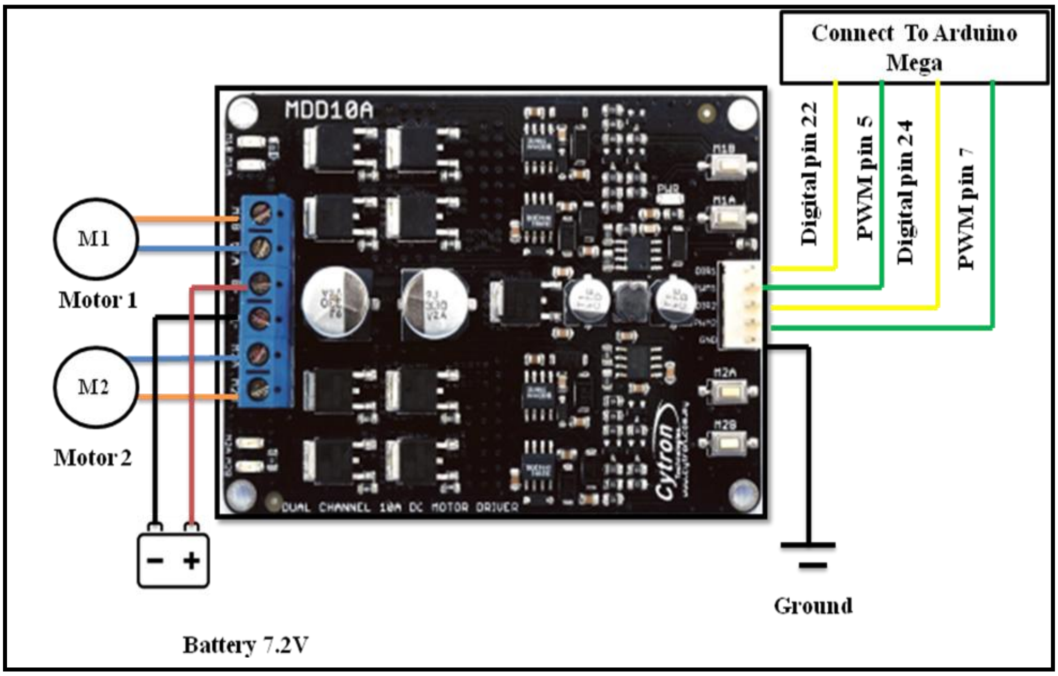

Figure 3.9: Connection for motor driver MDD10A

Table 3.6: Truth table for the control logic of motors

PWM	DIR	OUTPUT A	OUTPUT B
LOW	X (Don't care)	LOW	LOW
HIGH	LOW	HIGH	LOW
HIGH	HIGH	LOW	HIGH

3.2.1.5 Router

Router is a device for internet connection. This device is used as internet protocol based network. The router used is built- in Omni directional antenna. Besides that, the wireless router is used as access point for internet connection. Typically this type of router can cover a single -family dwelling only. This router determine the range for accessing 802.11 protocol. Besides that, the router operating on the range of 2.4 GHz and normally cover for 46m indoor environment and 92 m for outdoors. Fig. 3.10 shows the router used for this project.

Figure 3.10: Router

3.2.1.6 Wi-Fi shield (WizFi210)

The used of Wi-Fi shield (WizFi210) in Fig. 3.11 is to provide Wi-Fi communication interface by providing bridging from TTL serial port communication to IEEE802.11b/g/n wireless communication [20]. Since Arduino architecture have this protocol, so it easier for integration. In addition, this shield however is a low power

REFERENCES

[1] Asada, M. (1990). Map Building for A Mobile Robot From Sensory Data. *IEEE Transaction On Systems, Man and Cybernetics. 37(6).* pp. 1326-1336.

[2] Feng, L. & Milios, E. (1997). Robot Pose Estimation in Unknown Environments by Matching 2D Range Scans. *J. Intell. Robot. Syst*. 18(3). pp. 249-275.

[3] Ragavan, S.V., Ponnambalam, S.G. & Sumero, C. (2011). Waypoint-based Path Planner for Mobile Robot Navigation Using PSO and GA-AIS. pp 756-760.

[4] Mahiddin, N.A, Madi, E.N., S.D., Hasan, E.F., Safie, S. & Safie, N. (2013). User Position Detection In An Indoor Environment. *International Journal of Multimedia and Ubiquitous Engineering. 8(5).* pp 303-312.

[5] Wang, Y., Mulvaney, D., Sillitoe, I., & Swere, E. (2008). Robot Navigation by Waypoints. *Journal Intell Robot Systems. vol.52*. pp. 175–207.

[6] Kok, S.C. & Kleeman, L. (1997). Accurate Odometry and Error Modelling for a Mobile Robot. *Proc. of the 1997 IEEE Int. Conf. on Robotics an Automation*. Albuquerque, New Mexico. Monash Unversity. pp. 2783-2788.

[7] Yamauchi, B. (1996). Mobile Robot Localization in Dynamic Environments Using Dead Reckoning and Evidence Grids. *Proc. of the 1996 IEEE Int. Conf. on Robotics and Automation.* Minneapolis, Minnesota. Stanford University. pp 1401 - 1406.

[8] Lee, B.S., Cai, W., Stephen J.T., & Chen, L. (1999). Adaptive Dead Reckoning Algorithms For Distributed Interactive Simulation. *Proc. of 13th Workshop on Parallel And Distributed Simulation*. Atlanta, USA. Nanyang Technological University. 1(1-2). pp. 21 -34.

[9] Alsaab, A. & Bicker, R. (2014). Behavioral Strategy for Indoor Mobile Robot

Navigation in Dynamic Environments. *International Journal of Engineering Science and Innovative Technology (IJESIT). 3(1).* pp. 533-542.

[10] Millington, J.A., Palanisamy, C., Paprocki, R.M., et al. (2004). Vehicle Navigation System with Off-Road Navigation. United States Patent, Patent No.US 6,836,725 B2.

[11] Hamid, M.H.A., Adom, A.H., Rahim, N.A., Rahiman, M.H.F, (2009). Navigation of Mobile Robot Using GPS and Obstacle Avoidance System with commanded Loop Daisy Chaining Application Method. *Proc. on 5th Int. Colloquium on Signal Processing & Its Applications (CSPA).* Kuala Lumpur. University of Malaysia Perlis. pp. 176-181.

[12] Jirawimut, R., Ptasinski, P., Garaj, V., Cecelja, F. & Balachandran, W. (2003). A Method for Dead Reckoning Parameter Correction in Pedestrian Navigation System. *IEEE Transactions On Instrumentation And Measurement. 52(1).* pp. 209-215.

[13] Amundson, I., Sallai, J., Koutsoukos, X., & Ledeczi, A. (2012). Mobile Sensor Waypoint Navigation via RF-Based Angle of Arrival Localization. *International Journal of Distributed Sensor Networks.* pp. 1-14.

[14] Nuha.N (2005). H-Infinity Control Of An Autonomous Mobile Robot. pp. 1-85.

[15] Chung, H. , Ojeda, L., and Borenstein, J. (2001). Accurate Mobile Robot Dead Reckoning With A Precision-Calibrated Fibre Optic Gyroscope. *IEEE Transaction On Robotics and Automation. 17(1).* pp 80-84.

[16] Ladd, A.M., Bekris, K.E., Rudys, A.P., Wallach, D.S., & Kavraki, L.E. (2004). On the Feasibility of Using Wireless Ethernet for Indoor Localization. *IEEE Transactions On Robotics And Automation*. *20(3).* pp.555 - 559.

[17] Gomez, J. G., Victores, J.G., Gomez, A.V., & Abderrahim, M. Motion Control of Differential Wheeled Robots with Joint Limit Constraints.

[18] Hellström, T. (2011). Kinematics Equations for Differential Drive and Articulated Steering. Department of Computing Science Umeå University. pp. 1-12.

[19] Morales, Y., Carballo A., Takeuchi E., Aburadani A., & Tsubouchi T. (2009). Autonomous Robot Navigation in Outdoor Cluttered Pedestrian Walkways. *Journal of Field Robotics*. *26(8)*. pp. 609–635.

[20] ArduinoBoardMega2560. Retrieved October 6, 2014 from http://arduino.cc/ArduinoMega2560.

[21] Borenstein, J., Everett, H.R, & Feng, L. (1996). Where am I? Sensors and Methods for Mobile Robot Positioning. University of Michigan. pp 1-50.

[22] Jensen, T. M. (2011). *Waypoint-Following Guidance Based on Feasibility Algorithms*. Master Thesis. Norwegian University of Science and Technology.

[23] Briod, A. (2008). Waypoint Navigation With An MAV. Faculty Sciences Et Techniques De L'ingenieur. Institut D'ingenierie Des Systemes . pp. 1- 49.

[24] Khan, S., Ahmad, K., Murad, M., & Khan, I. (2013). Waypoint Navigation System Implementation via a Mobile Robot Using Global Positioning System (GPS) and Global System for Mobile Communications (GSM) Modems. *International Journal of Computational Engineering Research. 3(7).* pp. 49 - 54.

[25] Zhang, Y., Wu, C., Cheng, L., & Chu, H. (2012). Localization and Tracking of Indoor Mobile Robot with Ultrasonic and Dead-reckoning Sensors. *Journal of Computational Information Systems.8(2).* pp. 531–539.

[26] Tsai, C.C. (1998). A Localization System of a Mobile Robot by Fusing Dead-Reckoning and Ultrasonic Measurements. *IEEE Instrumentation and Measurement Technology Conf.* Minnesota, USA. pp 144 - 149.

[27] Miah, M.S, Gueaieb, W., Rahman, M.A, & El - Saddik, A. (2007). Autonomous Dead-Reckoning Mobile Robot Navigation System With Intelligent Precision Calibration. *IEEE Instrumentation and Measurement Technology Conf.* Warsaw, Poland. pp 1-5.

[28] Arvin, F., Samsudin, K. & Nasseri, M.A. Design of a Differential-Drive Wheeled Robot Controller with Pulse-Width Modulation. *Conference on Innovative Technologies in Intelligent Systems and Industrial Applications (CITISIA).* Monash University, Malaysia. pp 143 -147.

[29] Bruch, M.H., Gilbreath, G.A., Muelhauser, J.W., & Lum, J.Q. Accurate waypoint navigation using non-differential GPS. San Diego, CA. pp 1-15.

[30] Schwab, C., & Bushmais, A. Dead Reckoning Using Optical Flow for Mobile Device Location Tracking. University of Texas, USA. pp 1-5.

[31] Petereit, J., Emter, T. & Frey, C.W. Safe Mobile Robot Motion Planning for Waypoint Sequences in a Dynamic Environment. Institute of Optronics, System Technologies and Image Exploitation. Germany. pp 1-6.