RDU 0903115

TRAJECTORY TRACKING CONTROL OF TWO WHEELS MOBILE ROBOT

(TRAJEKTORI KAWALAN PENJEJAKAN SISTEM ROBOT DUA TAYAR)

NOR MANIHA BINTI ABDUL GHANI
RAJA MOHD TAUFIKA BIN RAJA ISMAIL
NURUL HAZLINA BINTI NOORDIN
MOHD ASHRAF BIN AHMAD

RESEARCH VOTE NO:
RDU 0903115

Fakulti Kejuruteraan Elektrik & Elektronik
Universiti Malaysia Pahang

2011

Universiti Malaysia PAHANG

Engineering • Technology • Creativity

PUSAT PENGURUSAN PENYELIDIKAN (RMC)

BORANG PENGESAHAN
LAPORAN AKHIR PENYELIDIKAN

TAJUK PROJEK : TRAJECTORY TRACKING CONTROL OF TWO WHEELS MOBILE ROBOT

Saya : NOR MANIHA BINTI ABDUL GHANI
(HURUF BESAR)

Mengaku membenarkan **Laporan Akhir Penyelidikan** ini disimpan di Perpustakaan Universiti Malaysia Pahang dengan syarat-syarat kegunaan seperti berikut :

1. Laporan Akhir Penyelidikan ini adalah hakmilik Universiti Malaysia Pahang
2. Perpustakaan Universiti Malaysia Pahang dibenarkan membuat salinan untuk tujuan rujukan sahaja.
3. Perpustakaan dibenarkan membuat penjualan salinan Laporan Akhir Penyelidikan ini bagi kategori TIDAK TERHAD.
4. * Sila tandakan (/)

[ ]	SULIT	(Mengandungi maklumat yang berdarjah keselamatan atau Kepentingan Malaysia seperti yang termaktub di dalam AKTA RAHSIA RASMI 1972).
[ ]	TERHAD	(Mengandungi maklumat TERHAD yang telah ditentukan oleh Organisasi/badan di mana penyelidikan dijalankan).
[/]	TIDAK TERHAD	

Tandatangan & Cop Ketua Penyelidik

NORMANIHA BINTI ABD GHANI
PENSYARAH
FAKULTI KEJURUTERAAN ELEKTRIK & ELEKTRONIK
UNIVERSITI MALAYSIA PAHANG
26600 PEKAN
PAHANG DARUL MAKMUR
TEL : 09-4242110 FAKS : 09-4242032

Tarikh : 11/10/2011

CATATAN : **Jika Laporan Akhir Penyelidikan ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali sebab dan tempoh laporan ini perlu dikelaskan sebagai SULIT dan TERHAD.*

ABSTRACT

TRAJECTORY TRACKING CONTROL OF TWO WHEELS MOBILE ROBOT

(Keywords: infra-red sensor, PID algorithm, line follower Balancing Robot)

This project focuses on the development of a line follower control algorithm for a Two Wheels Mobile Robot. In this project, ATMEGA32 is chosen as the brain board controller to react towards the data received from the Balance Processor Chip on the balance board which monitoring the changing of the environment through two infra-red distance sensor to solve the inclination angle problem. Hence, the system will immediately restore to the set point (balance position) through the implementation of internal PID algorithms at the balance board. Application of infra-red light sensors with the PID control is vital, in order to develop a smooth line follower robot. As a result of combination between line follower program and internal self balancing algorithms, we able to develop a dynamically stabilized Balbot with line follower function.

Key researchers: Nor Maniha Abdul Ghani, Raja Mohd Taufika Raja Ismail, Nurul Hazlina Noordin, Mohd Ashraf Ahmad

Email : **normaniha@ump.edu.my**
Tel No : **09-4242110**

Vote No. : **RDU 0903115**

ABSTRAK

TRAJEKTORI KAWALAN PENJEJAKAN SISTEM ROBOT DUA TAYAR

(Kata kunci : **pengesan infra-red, PID algoritma, pengikut garisan robot**)

Projek ini tertumpu kepada pembangunan pengendali pengikut garis dan algoritma untuk robot dua roda mudah alih. Dalam projek ini, ATMEGA32 dipilih sebagai pengendali papan utama untuk bertindak balas terhadap data yang diterima dari Cip Keseimbangan pada papan keseimbangan yang memantau perubahan persekitaran melalui dua pengesan jarak infra-merah untuk menyelesaikan masalah sudut cerun. Oleh kerana itu, sistem akan mengembalikan ke titik penentuan (kedudukan keseimbangan) melalui pelaksanaan algoritma PID dalaman di papan keseimbangan. Penerapan pengesan sinar infra-merah dengan kawalan PID sangat penting, dalam rangka untuk mengembangkan sebuah robot pengikut garis halus. Sebagai hasil dari kombinasi antara pengaturcara pengikut garisan dan dalaman algoritma menyeimbangkan diri, kita mampu mengembangkan Balbot stabil secara dinamik dengan fungsi pengikut garisan.

Penyelidik : Nor Maniha Abdul Ghani, Raja Mohd Taufika Raja Ismail,
Nurul Hazlina Noordin, Mohd Ashraf Ahmad

Email : **normaniha@ump.edu.my**
Tel No : **09-4242110**
Vote No. : **RDU 0903115**

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATION

TWIP	-	Two Wheeled Inverted Pendulum
Balbot	-	Balancing Robot
PID	-	Proportional Integral Derivative
BPC	-	Balance Processor Chip
BEMF	-	Back Electromotive Force
GUI	-	Graphical User Interface
IR	-	Infrared
I^2C	-	Inter Integrated IC
LED	-	Light Emitting Diode
CV	-	Control Variable
SP	-	Set Point
PV	-	Process Value
MIPS	-	Microprocessor without Interlocked Pipeline Stages
EEPROM	-	Electrically Erasable Programmable Read Only
SRAM	-	Static Random Access Memory
PWM	-	Pulse Width Modulation
DAC	-	Digital Analog Converter
ADC	-	Analog Digital Converter

CHAPTER 1

INTRODUCTION

1.1 Background of Project

Over the past decades, the research on two wheeled inverted pendulum mobile robot or commonly known as balancing robot have gain momentum in a number of robotic centers around the world due to natural unstable dynamics of the system. [1] Since a Two wheeled balancing robot need a good controller to maintain itself in upright position without the needs from external forces. Thus, providing a good platform for researcher to explore the efficiency of various controllers in control system based on the inverted pendulum model. Nowadays, various controllers were implemented on two wheeled balancing robot for example s Linear Quadratic Regulator, Pole-Placement Controller, Fuzzy Logic Controller, and Proportional Integrated Derivative Controller. [3]

A two wheeled balancing robot is categorized by the ability to balance on its two wheels and spin on the spot. As a result from this additional maneuverability allows easy navigation on the various terrains, turn sharp corner, traverse small step or curbs and ability on carry load. Two wheeled robots also have a small footprint than three or four wheeled robots thus enable it to travel around corridors and tight corners more easily. [4] These capabilities have the potential to solve many challenges in industry and society. As the two wheeled balancing robot has been investigated and developed to become human transport machine. The Segway, Pegasus, and iBot models are the example of the design of two wheeled balancing robot as a human transport machine. In addition, a motorized wheelchair utilizing this technology would give the operator greater maneuverability and thus access to places most able-bodied people take for granted. Small carts built utilizing this technology allows humans to travel short distances in a small area or factories as opposed to using cars or buggies which is more polluting. [2]

In this project, Balbot Advanced is an autonomous, active-balancing robot that is fully customizable been design and fabricate as two wheeled balancing robot platform that has ability to balance itself on a flat terrain with add-on line follower function. The robot chassis design is robust and symmetrical with high centre of gravity. These mobile robots not only solve the balancing problem but also can automatically move around and avoid basic obstacles through the implementation of Dual ground Sensor and looking forward infrared sensor respectively. The Dual ground sensor is located at the bottom of the Balbot which is the infrared IR distance sensor, where ground sensors are used to measure the tilting angle. Powerful ATMEGA32 processor is used to be the brain of the robot. BEMP motor velocity sensor is used to obtain the speed of the platform. The entire controller algorithm will be compute into C programming and store inside the microcontroller. Without an active control system, the robot would just fall over. Thus, the controller plays an important role in this project. Lastly, three pairs of infra-red sensor are used to guide the line follower task during balance state.

There are a few specific tasks included such as:

1. Integration of sensor to determine and updating the status of the platform.
2. Integration of hardware for data acquisition.
3. Integration and testing of the performance of the controllers for trajectory control.
4. Integration of hardware for line follower application.

1.2 Problem Statement

The problem statement for this project work is expressed as follows:

"To develop a stable line follower circuit and control algorithm for the purpose of line following of two wheels balancing robot"

1.3 Objective of Project

There are three main objectives in this project which are;

1) To control the Balbot with line following function using Atmel Microcontroller.

2) To design and construct a simple and functional IR sensors circuit for line detection.

3) To program the microcontroller to perform stable and smooth line following task during balanced state.

1.4 Scope of Project

In order to achieve the objective of the project, there are few scopes had been outlined which involve hardware and software:

1) A Balbot is to be considered able to conduct self-balancing and position control using Linear Controller via the existing balance board and default program.

2) Construct a line following hardware system using Infra-red sensor.

3) Develop a line follower algorithm to perform line following task.

1.5 Project Methodology

The research project are divided into chapters, each recorded a chronological step in the process of developing and structure the Line Follower Balbot. This approach was utilized in attempt to progress the project from one task to the next as it was undertaken. Each is clear so that it builds on the preceding task thus evolving the robot within the goals and requirements generated. This eventually led to the completion of the Balbot that met the objectives within the timeframe available.

Chapter 1 formed the first step where key points and objectives were established including the idea of actually what is a two wheeled balancing robot. Understanding about this project is critical in determining plans for conducting research and performing the design work.

Chapter 2 provides the second step in which a comprehensive understanding of previous projects and approaches is required. This established the foundations for making informed decision based on the past experiences and problem encountered. This can help set up an avoidance of problems, sufficient planning of resources and the helpful application of effort.

Chapter 3 entailed the software review, hardware review, and balancing with line following algorithm in Balbot. Detailed information about the basic operation of Balbot and hardware specification such as microcontroller Brain Board and Balance Board was listed and explained. The Balbot is considered to be balances on its body first then line following can be conducted. The line following feature for Balbot is based 3 infra-red sensors and the line follower programming algorithm that had designed. This step expanded to include a method to setup the Balbot and loading program on to the Balbot.

The subsequent step was to analyze the actual performance of the Balbot and ascertain its ability in achieving the objectives of balance and following line. This also provided the opportunity to calibrate and perform additional fine tuning of the output allowing it to become more effective and efficient in its performance.

The final component comprises of a complete assessment of each process undertaken, the choices made and achievements obtained during the project as well as evaluation of the final Balbot effectiveness. This expanded to include recommendations for future work that could be undertaken in an effort to improve areas of the process or design, addition of capabilities, or how to overcome problems that may have been encountered.

1.6 Flow of Project

The project implementation and works flow for Balbot are summarized into flow chart as shown in figure 1.1. The details of the works of the project that had been implemented are shown in the form of Gantt chart for both PSM 1 and PSM 2 as in Table 1.1 and Table 1.2.

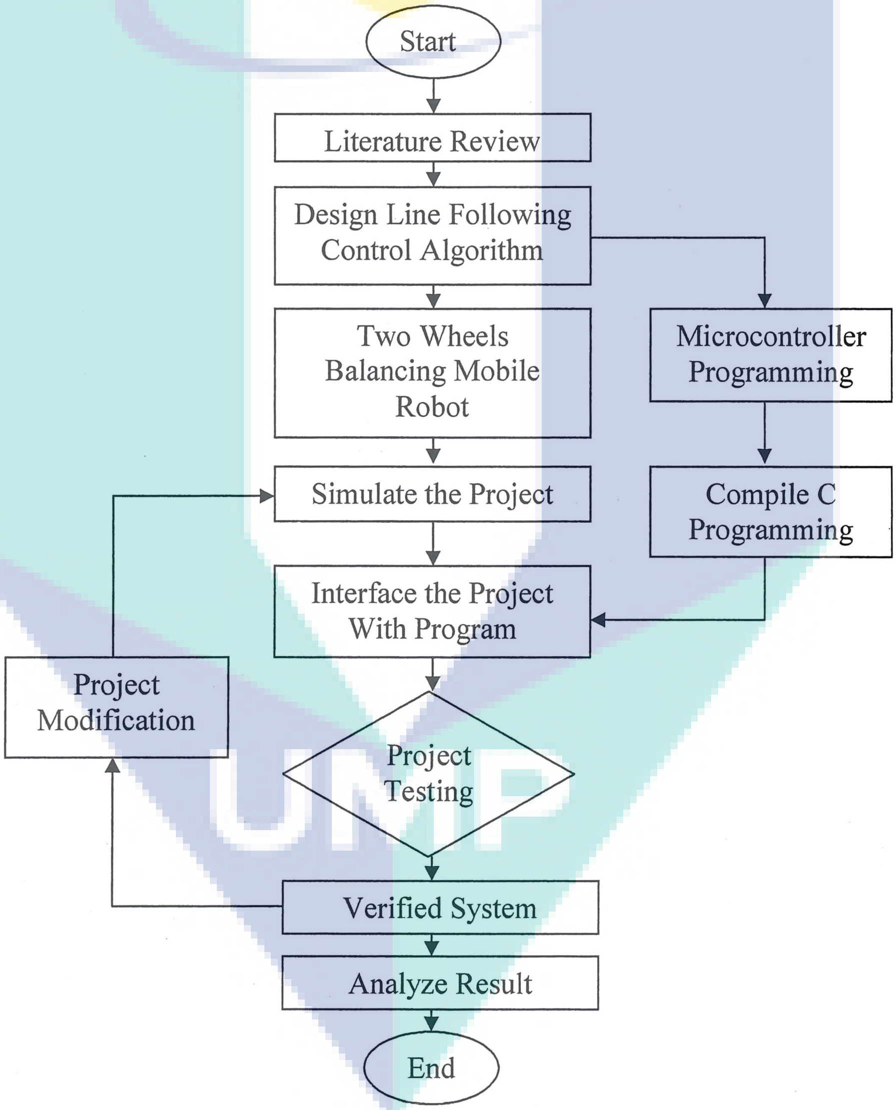


Figure 1.1: Project Work Flow

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

Conducting literature review prior to begin a research project is vital in understanding two wheels balancing robot control technique, as this will supply the researcher with much needed additional information on the methodologies and technologies available and used by other research counterparts around the world. This chapter provides a condensed summary of literature reviews on key topics related to balancing a two-wheeled robot. [2] Comparisons between the present project and the related topics of existing information will also be discussed.

The two wheels balancing concept has been a topic of high interest among the control engineering community. The uniqueness and complexity of the two wheels balancing concept has made it an ideal concept that can be used for both commercial and military (government) applications. In past few years, researcher and engineers have applied the two wheels balancing model to various fields, which include walking gait for humanoid robots, personal transport systems and robotic wheelchairs. As examples, the Segway is used for commercial purposes and the prototype of the VECNA B.E.A.R robot project can be used for government military operation. Humanoid like robots that gain the mobility through two wheels have become popular in the past few years in commercial and government.

The control problem can be simulated and implemented on a classroom setting to teach control engineering students the need for control in the unstable two wheeled robot. The following research literature abstracts summary of some popular balancing robot platforms and extra features technologies that are related and used by researchers and engineers around the world which are available in an attempt to gain an understanding and appreciation of two wheels balancing robot.

2.2 Previous Project Work

2.2.1 BallBot

A research professor at Carnegie Mellon University, Ralph Hollis who has developed a totally unique balancing robot that balances on top of a bowling ball. He calls his robot design "BallBot" [5, p. 72]. Mr. Hollis and his research associates believe that robots in the future will play a vital role in the daily lives of humans. He believes that in order for robots to be productive in our daily lives, some key problems need to be solved first. One the important problem he states in his article about mobile self balancing robots is the overall structure of the robot itself.

As stated by Ralph Hollis," Robots tall enough to interact effectively in human environments have a high centre of gravity and must accelerate and decelerate slowly, as well as avoid steep ramps, to keep from falling over. To counter this problem, statically stable robots tend to have broad bodies on wide a wheelbase, which greatly restricts their mobility through doorways and around furniture or people" [5, p. 74]. The size of the robots will ultimately affect its mobility of the robot. In order to solve the problem, Hollis came up with a new design that improved the robot's overall structure and mobility. Hollis and his associates have built a five foot tall, agile, and skinny robot. The robot's design is to balance itself on top of a spherical wheel. Hollis compares his robot structure design much like a giant ball pen or a circus clown trying balance on top of a ball. [6]

A side from acknowledging and solving the structure and mobility difficulty, Hollis also faced a major challenge when it came to keeping the self-balancing robot on a stable vertical position. Hollis found that the best way to solve this new issue was by implementing into his design advance sensors and control algorithms. The sensors that he incorporated included a gyroscope and an accelerometer. They were set and placed orthogonal to each other. Hollis implemented a Linear Quadratic Regulator (LQR); a control algorithm technique used to keep the BallBot in a stable vertical state. The LQR is based on optimal control theory.

The main objective when using optimal control techniques on a system is to minimize the effort to stabilize the BallBot in the vertical position. The Ballbot's incorporates optimal control algorithms. These control algorithms helped increase stability and system robustness. BallBot major strength is the inertial measurement units used to provide the tilt angle information. The Ballbot's size and inability to climb staircases were obvious weaknesses. Hollis's article "BallBot" is informative piece of literature that has presented a technological innovation in robotics. It has inspired parts of this master's project.

Just like Hollis BallBot, this master's project two wheeled balancing robot also incorporates the control techniques in order to achieve a vertical stability. After acquiring additional information on both the gyroscope and accelerometer sensors, it was determined that it was the best choice of sensors to be implemented in the self-balancing two wheel robot. As there are similarities between Hollis's BallBot and self-balancing two wheel robot, there are also differences. The structures and control algorithms used were a major difference. As stated before, the BallBot uses optimal control theory to minimize the robot's effort to stabilize. The self-balancing two wheel robot used for this master's project will use classical control theory.

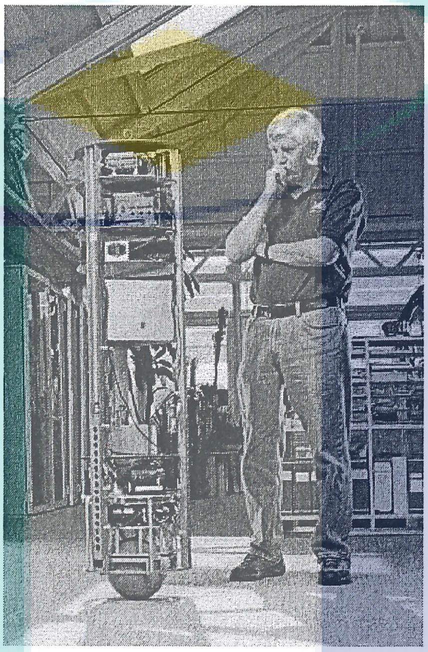


Figure 2.1: Ralph Hollis and his team's BallBot

2.2.2 nBot and Legway

Two wheel balancing robots have also gain popularity among hobbyists and engineering students. Examples of such popular two wheeled balancing include the nBot and the Legway. The two wheeled robot platforms have drawn high interest from the robot enthusiast communities.

nBot is a two-wheeled balancing robot built by David P. Anderson. This robot uses commercially available inertial sensors and motor encoders to balance the system. Such inertial sensors that are used on nBot are an accelerometer and a gyroscope. The basic idea for a two-wheeled dynamically balancing robot is pretty simple which is driving the wheels in the direction that the upper part of the robot is falling. If the wheels can be driven in such a way as to stay under the robot's center of gravity, the robot remains balanced. In practice this requires two feedback sensors: a tilt or angle sensor to measure the tilt of the robot with respect to gravity, and wheel encoders to measure the position of the base of the robot. Four terms are sufficient to define the motion and position of this "inverted pendulum" and thereby balance the robot. These are

1. The tilt angle and
2. Its first derivative, the angle velocity,
3. The platform position, and
4. Its first derivative, the platform velocity.

These four measurements are summed and fed back to the platform as a motor voltage, which is proportional to torque, to balance and drive the robot. (Anderson D.P. 2003)

A researcher, Steve Hassenplug has successfully constructed a balancing robot called Legway using the LEGO Mindstorms robotics kit. Two Electro-Optical Proximity Detector sensors from HiTechnic Sensors to provide the tilt angle information and detect lines. The controller is programmed in high level programming language specifically created for LEGO Mindstorms which was written in brickOS (LegOS) and uses EOPDs to maintain a constant distance from the ground. As the distance decreases, Legway moves forward. As the distance increases, Legway moves backward. Every 50 ms, Legway attempts to recalculate the balance point by measuring the current distance and motor speed. To move forward for line following, Legway actually sets the motors to run backward, causing a tilt, which it automatically corrects by moving forward. When one sensor is over the line, it stops that motor, and Legway balances using only the other motor, causing it to turn. To spin in place, both motors are shifted "off center" in opposite directions, the same amount, but they still correct for tilting. Legway uses its two optical proximity detectors to balance the two wheel LEGO robot. (Hassenplug S. 2003)

Major strengths of the both the nBot and the Legway are the accessibility and availability of parts and the lower building cost. The fact that these two designs use off the shelf parts with no custom parts make them easier to build and in turn bring down the price. With that in mind, the autonomous self-balancing two wheel robot was also designed to accommodate commercially available parts. The autonomous self balancing two wheel robot presented on this master's project report will have almost similar design structure as that of David P. Anderson, nBot. A weakness of both the Legway and nBot is the limited environment and terrain that both robots can travel.

The articles on the nBot and Legway briefly explained the control algorithms used to balance and keep the both robots in a stable state. The nBot created by David P. Anderson and the Legway created by Steve Hassenplug are both two wheeled balancing robots can be made from little control theory knowledge. Both robots are modelled after the inverted pendulum.

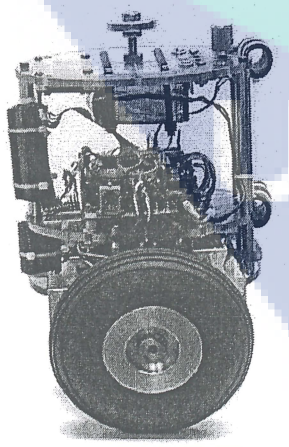


Figure 2.2: nBot

(Anderson D.P. 2003)

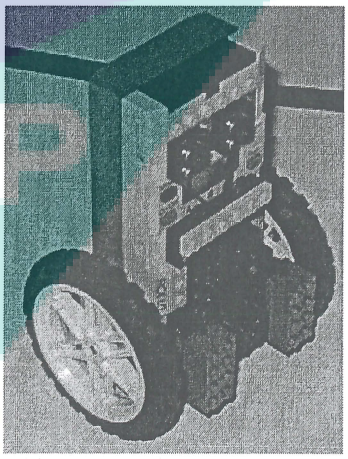


Figure 2.3: Legway

(Hassenplug S. 2003)

2.2.3 Segway

In past few years, the use of personal human transport vehicles has gained popularity. The Segway PT is a popular personal vehicle that is available to the public. Invented by Dean Kamen, the Segway PT's dynamics are identical to the inverted pendulum. For added mobility, the Segway is also based on the two wheel platform design. The advanced control algorithms behind the Segway transporter are a company trade secret. The basics of a Segway are computers that process the control algorithms, two tilt sensors, five gyroscopes, and two electric motors. Only three of the five gyroscopes are used to balance the Segway. The remaining two gyroscopes are used as backup. These critical components that make up a Segway are important to keep the vehicle in perfect balance. Current models of the Segway personal transporter can achieve top speeds of 12.5 mph. The Segway is able to navigate thru rough terrain, while successfully carrying a human onto of the platform. The Segway is typically found in urban settings; used for guided tours and city government officials.

The strengths of the Segway are that the personal transporter can be used in outdoor recreation. It is an alternative for people that are unable to walk long distances or ride a bike to enjoy the outdoors without the use of a vehicle. Since the Segway runs on rechargeable batteries, it is environmental friendly. A disadvantage of Segway is its cost which can run in the few thousands of dollars.

In contrast to this master's project on the autonomous self balancing two wheel robot, it has only a cost of few hundred dollars. The autonomous self balancing two wheeled robot presented on this master's project report can implement basic control algorithms similar to the Segway.

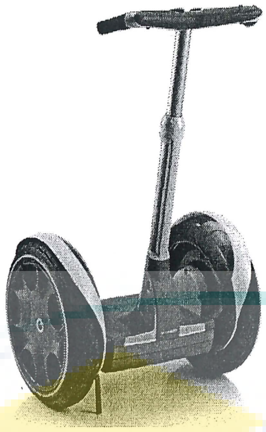


Figure 2.4: SEGWAY
(Dean Kamen 2001)

2.2.4 Introduction of Line Follower Robot

Line follower robot is one of the self operating robot that follows a line that drawn on the floor. Generally, the path is predefined and can be either visible like a black line on a white surface with a high contrasted colour or it can be invisible like a magnetic field. The basic operations of the line following are as follows:

1. Capture line position with optical sensors mounted at front end of the robot. Most are using several numbers of photo-reflectors, and some leading contestants are using an image sensor for image processing. The line sensing process requires high resolution and high robustness.

2. Steer robot to track the line with any steering mechanism. This is just a servo operation; any phase compensation will be required to stabilize tracking motion by applying digital PID filter or any other servo algorithm.

3. Control speed according to the lane condition. Running speed is limited during passing a curve due to friction of the tire and the floor.

The researchers have illustrated the process of design, implementation and testing a small line follower robot, TABAR which designed for the line follower robots competition.

2.2.5 AVR Line Follower Robot

A researcher, Priyank Patil has successfully design and constructed a Line Follower Robot which wills sense a line and maneuvering the robot to stay on course, while constantly correcting wrong moves using feedback mechanism forms a simple yet effective closed loop system. This robot is running on Atmel's AVR® microcontrollers which have a RISC core running single cycle instructions and a well-defined I/O structure that limits the need for external components. Internal oscillators, timers, UART, SPI, pull-up resistors, pulse width modulation, ADC, analog comparator and watch-dog timers are some of the features you will find in AVR devices. AVR instructions are tuned to decrease the size of the program whether the code is written in C or Assembly. With on-chip in-system programmable Flash and EEPROM, the AVR is a perfect choice in order to optimize cost and get product to the market quickly. [8]

The robot use 4 Infra-Red sensors to sense the line, The output of the sensors is an analog signal which depends on the amount of light reflected back, this analog signal is given to the comparator to produce 0s and 1s which are then fed to the microcontroller. Then, the microcontroller wills decides the next move according to the controller algorithm specified in microcontroller initially by calculation. L298 Motor Driver is used to control the motion of the motors and two enable inputs which are used for switching the motors on and off. Pulse Width Modulation (PWM) waveform with variable duty cycle is applied to control the speed of the motors.

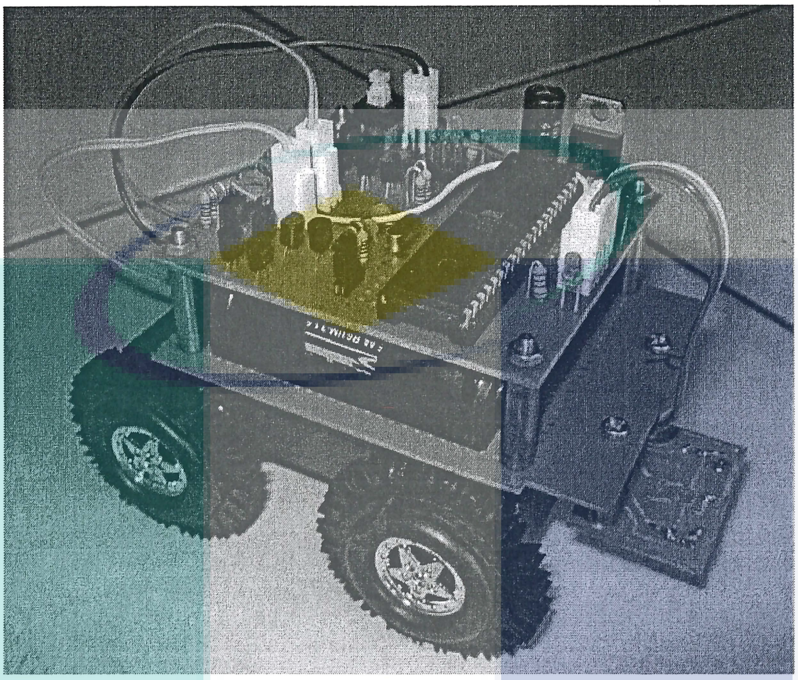


Figure 2.5: AVR Line Follower Robot

2.3 PID Control Systems

Control system development is an imperative process to guarantee the success of stabilizing the two wheeled robot. Since, there are varieties of control techniques that can be applied to stabilize the robot; the main objective is to control the robot system effectively and at a low cost without limiting the strength and performance of the controller. The elements that define how a balance control algorithm will be implemented depend on how the system will be modeled and how the tilt sensor data is obtained. A common approach that is often used by two wheeled robot designers is to separate the balancing and position control from the mobile robot.

Control techniques for a control system can be divided into two distinct categories: First technique being a linear control model of the system and second category being the nonlinear controller model.

A linear control method models the process about a desire operating point. The linear method is usually very sufficient in balancing the system and bringing it to a stable vertical position. On the other hand, a nonlinear controller uses the unrealistic dynamics model of the system in order to design a controller. Nonlinear controllers would provide a more robust system implementation.

The implementation and complexity difficulty associated with the nonlinear method causes most control researchers to utilize the linear controller approach. The method that will be used to control the self-balancing two wheeled robot will be a linear controller. It will be applied through a Proportional, Integral, and Derivative also refer to as the PID. The PID has proven to be popular among the control engineering community. As stated by the author of article Vance J. Van Doren, "For more than 60 years after the introduction of Proportional-Integral-Derivative controllers, remain the workhorse of industrial process control" [9].

CHAPTER 3

METHODOLOGY

3.1 Introduction

The main purpose of this project is to upgrade a Balbot to become a line follower Balbot. It involves the understanding of balancing system in the Balance Board which using PID controller as linear controller with Analog GP2D120 IR distance sensors as feedback input and the concept of line following system with the implementation of IR sensors as shown in Figure 3.1 about the Balbot General System Block Diagram. The line following mode can be establish based on the algorithm programmed in the Brain Board once the Balbot achieved balance state or condition as described in Figure 3.2 about the Balbot's Internal System Flow.

This project is dividing into two parts namely hardware and software implementation. Hardware part involves more in interfacing between Brain Board and Balance Board with Analog GP2D120 IR distance sensors and Infra Red Sensors and also brief introduction on the Balbot's components. While in the software part include programming software using Brain Board Code Editor and procedure in loading programs into the microcontroller on the Brain Board. Each part of the project will be discussed in the following section.

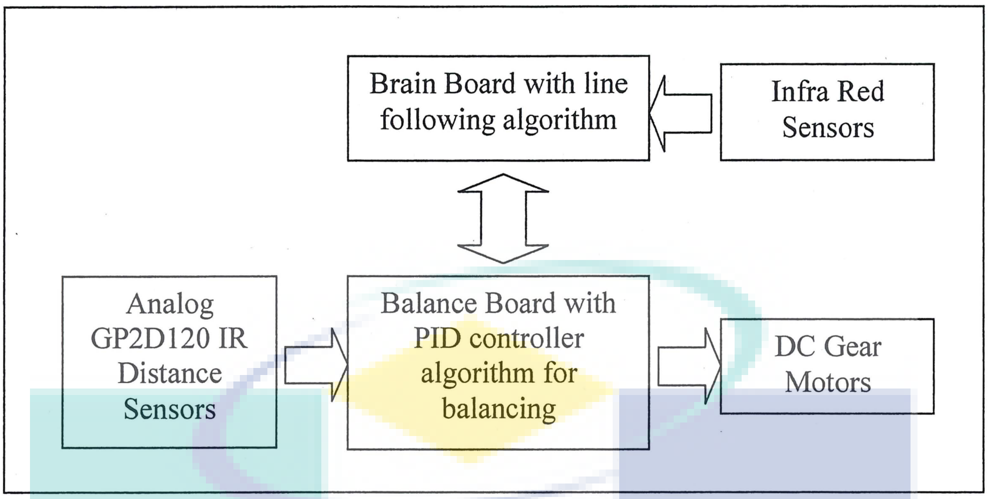


Figure 3.1: Balbot's General System Block Diagram

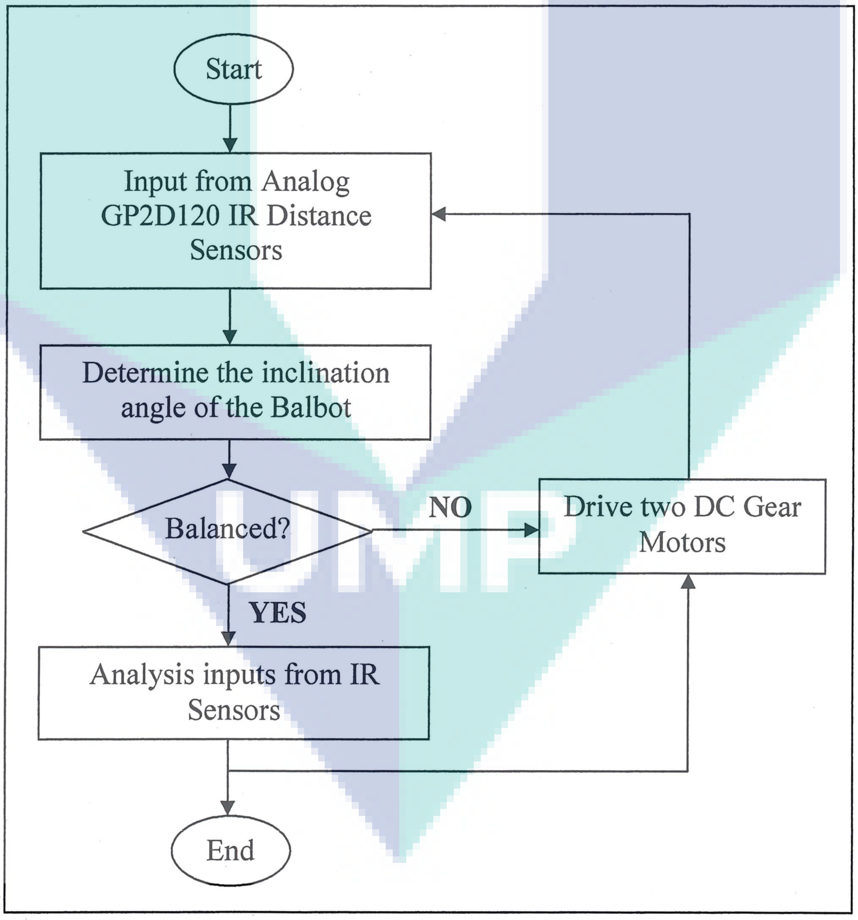


Figure 3.2: Balbot's Internal System Flow

In this chapter, three main topics will be discussed.

i. Balancing and Line Following Control System

- PID Controller
- Balancing System
- Line Following System

ii. Software Review

- Brain Board Code Editor
- Mega32 ISP Programmer
- Procedures in Program Loading

iii. Hardware Review

- Brain Board
- Mode of Operation
- Balance Board
- Analog GP2D120 IR Distance Sensors
- Infra-Red Sensors Module
- DC Gear Motors
- Full Gear Programmer
- RS232 Serial Cable

In the early stage, literature review and experiment are performed manually to understand the concept of the Balancing Robot and Line Following System. Interactive software will be program using Brain Board Code Editor. The program will be consists of instructions to ease any mistake when performing experiment. Finally, the program will integrate with the hardware to produce a stabilize Line Follower Balbot.

3.2 Balancing and Line Following control system

This subtopic discuss about the balancing mode of the Balbot and the designing of line following control system. This Balbot used PID controller as linear controller for balancing system. The balancing system have to achieve balance state then only the line following system is established based on the programming set for line follower in Balbot.

3.2.1 PID Controller

PID controller algorithm was used to maintain the balance state of the two wheeled balancing robot. The proportional, integral, and derivative (PID) controllers, is well known as a three term controller. The PID controller is the popular choice in industrial process control, due to the simplicity of the controller. Thus, it has become the basis for many advanced control algorithms and strategies.

As microprocessors and microcontrollers have become popular in control engineering, the PID controller has become a popular embedded software implementation. This PID controller being implemented in software has outperformed the analog and mechanical versions of the controller. The controller can now be programmed onto a single integrated circuit chip.

The PID control algorithm is a very straight forward algorithm that provides the necessary output system response to control a process. One unique feature of the PID controller is that it is capable of manipulating the process inputs based on the history and rate of change of the signal. The algorithm is best suited for linear system modeled process control which gives more accurate and stable control.

The PID controller consists of proportional, integral, and derivative terms. The terms of the PID controller are summed up to create a controller output signal. Each term performs a different task in the control process. The controller terms also have different effects on the system output response.

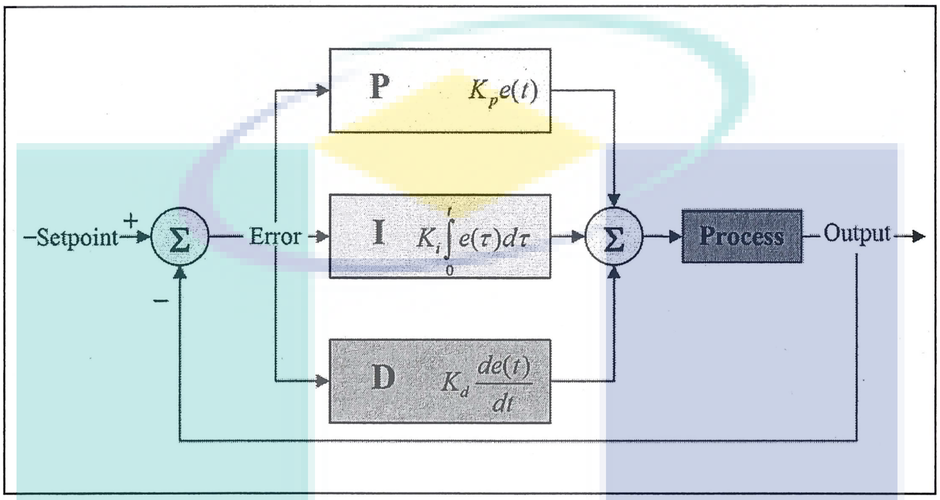


Figure 3.3: PID Controller Block Diagram

The input to the controller is the *error* from the system and the *u* is the output controller signal. The KP, KI, and KD are referred as the proportional, integral, and derivative constants. The PID controllers are widely used on closed loop control systems, where the process output measurement is fed back to the system and gets processed by the controller.

The closed loop control system is also referred to as a negative feedback system. The basic idea of a negative feedback system is that it measures the process output *y* from a sensor. The measured process output gets subtracted from the reference *set-point* value to produce an *error*. The error is then fed into the PID controller, where the error gets managed in three ways. The error will be used on the PID controller to execute the proportional term, integral term for reduction of steady state errors, and the derivative term to handle overshoots.

After the PID algorithm processes the error, the controller produces a control signal *u*. The PID control signal then gets fed into the process under control. The process under PID control is the two wheeled robot. The PID control signal will try to drive the process to the desired reference set point value. In the case of the two wheel robot, the desired set-point value is the zero degree vertical position.

3.2.2 Balancing System

The Balbot balances by using the infrared range sensors aimed at the ground to calculate the absolute angle and the angular velocity of the robot. The primary function of the Balance Processor Chip (BPC) is to perform real time balancing control by using data from the GP2D120 IR Distance Sensors to drive the two motors in a manner that will keep the center of gravity above the wheels at all times. Thus, frees the master processor in the Brain Board to perform other tasks. The PID controller in balancing system is a linear controller that control overall performance of Balbot. DC Gear motors are actuator for movement.

The Balance Board executes the standard Digital Cascaded Control Loop (built into Balance Processor Chip) algorithm to balance the robot. The control loop runs at a frequency of approximately 40.2Hz. The following Figure 3.4 gives a diagram of simplified overview on how this balancing control system works which is a closed loop system with controlled of the desired velocity as the input of the system. Its parameters "angle_offset", "K1", and "K3" can be modified by using an add-on microcontroller (the Brain Board) or it also can be by-passed altogether to allow user's completely custom control loop. [10]

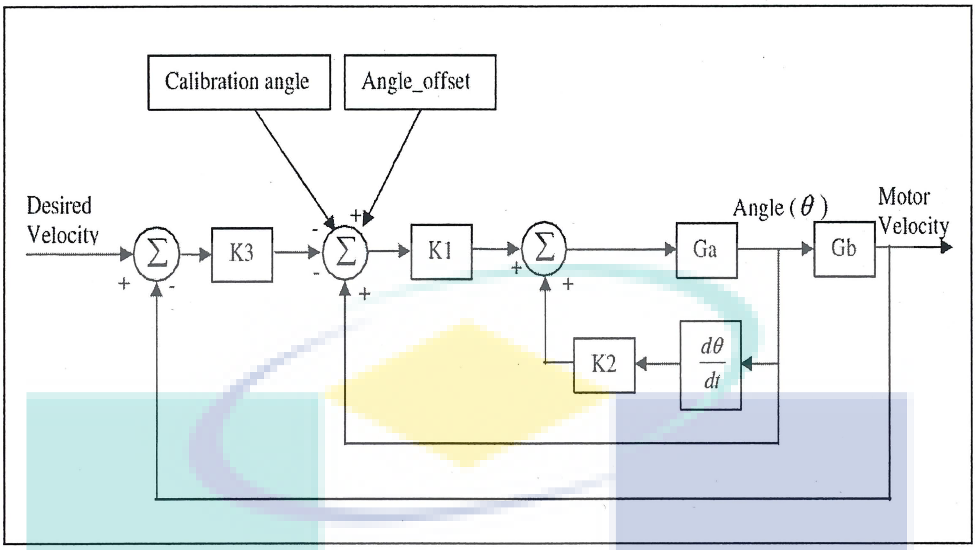


Figure 3.4: Balbot Balancing Control System's Block Diagram

The actual velocity of Balbot will be detected by BPC using Back-Electromotive Force (BEMP). Then, BPC will calculate the error between the desired velocities with the actual velocity by considering also the tilt angle as the feedback from GP2D120 IR Distance Sensors, thus the Balbot will moving toward a falling direction (left side) at a specific angular velocity calculated by the system corresponding with the tilt angle in order to maintained the Balbot in balance condition as shown in Figure 3.5. [10]

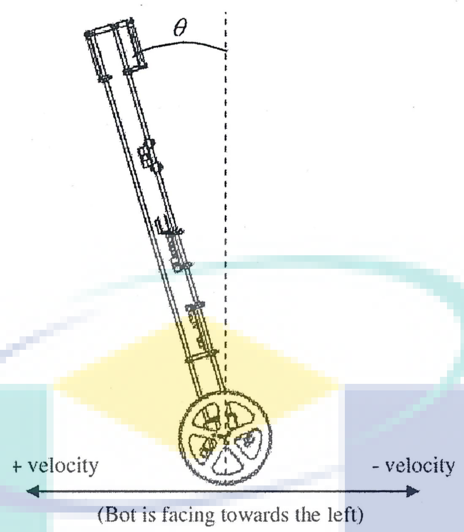


Figure 3.5: The movement of Balbot due to the tilt angle with specific velocity

3.2.3 Line Following System

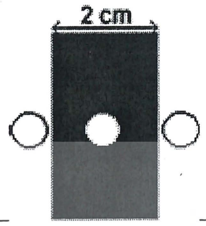
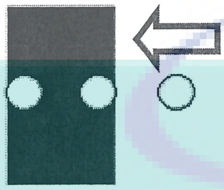
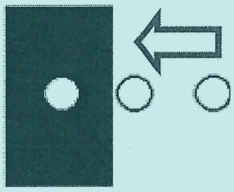
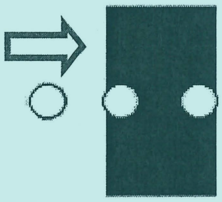
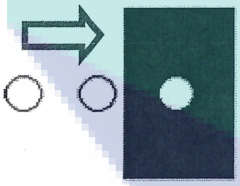
Line following task is performed through the implementation of specified algorithm with the data from the Infra-Red sensors which are mounted at the bottom of Balbot. The data or status receive from the sensor will be processed by the programmed microcontroller to perform the continuous movement according to the line. The direction and speed of rotation is control by the microcontroller through PWM manipulation.

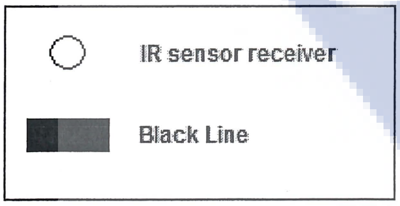
The Balbot will conduct line following function when a balancing state is achieved. If the robot is in unstable condition, balancing task will be performed first, then follow by the line follower. Table 3.1 shows that the designed line follower control algorithm for Balbot. There have five states for line following control in Balbot which are five possible condition might be occurs during line follower system execution in Balbot. All of the five states are expressed in five desired output direction in Balbot which are forward, shift to left, fast shift to left, shift to right, and fast shift to right.

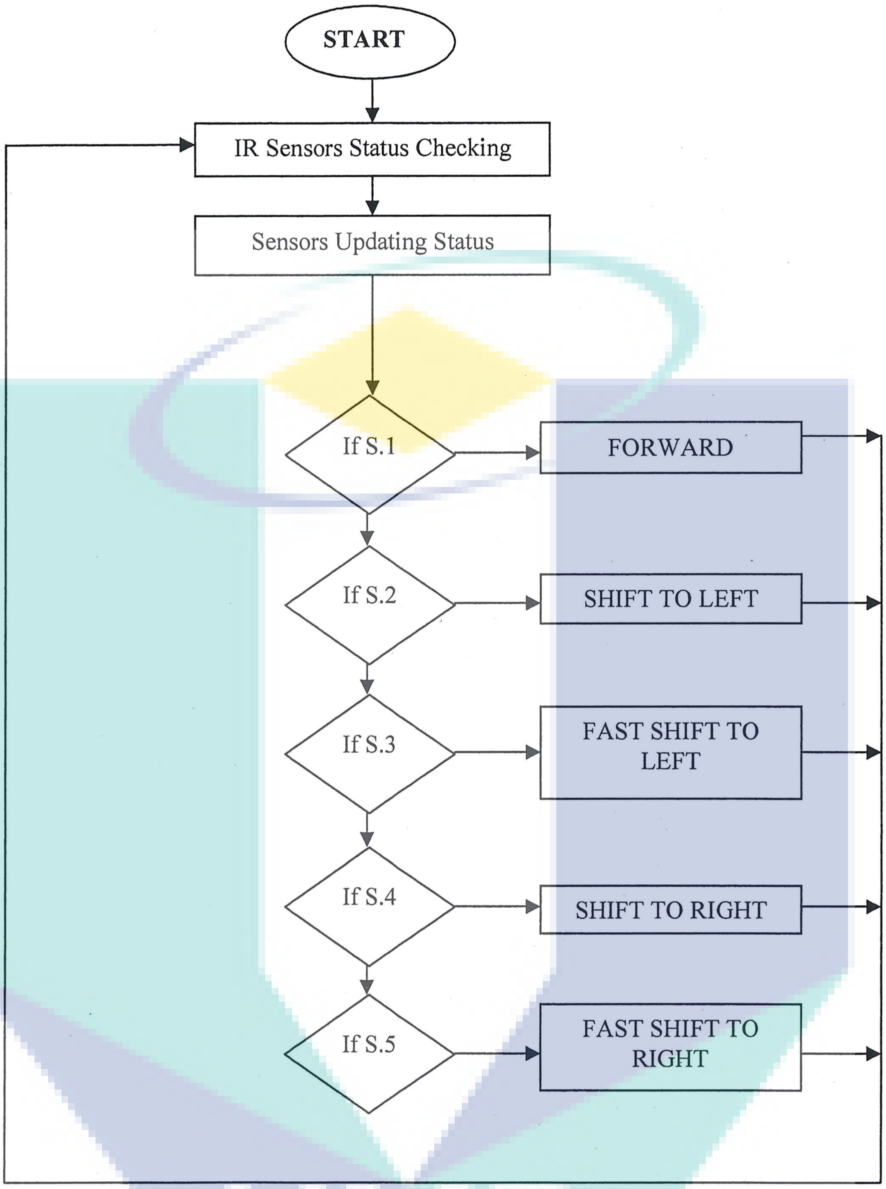
The inputs in this line follower are IR sensors which mounted at the bottom of Balbot. The logic "1" indicates no line detect while logic "0" indicates line detected. The algorithm can be adapted for situation in which the Balbot is continuously moving forwards. The line follower algorithm also can be expressed in flowchart for the decision of Balbot as shown in Figure 3.6. It is showed that the Balbot in this line follower algorithm will check on the state of the sensors first and based on the information from the three IR sensors, the desired motion of the Balbot will be determined.

The programming that loaded in the ATMEGA32 to defined function of a line follower control algorithm and decision of Balbot movement is shown in Figure 3.7. Firstly, all the input ports for the sensors are been defined and initialized in Brain Board Code Editor Software. Then all the input ports are enable pull-ups. There are two variables that will change the desired path and speed of Balbot which are tmp_fwd and tmp_steer. For the tmp_forward is function to move Balbot forward or reverse with specific speed, while tmp_steer function to determine the steering direction of Balbot to turn left or right at desired speed. The positive number in tmp_fwd and tmp_steer indicate that Balbot will move forward and turn right respectively and vice versa.

Table 3.1: Line Follower Control Algorithm

Status	Sensor Input L	Sensor Input M	Sensor Input R	Output
	0	1	0	FORWARD
	1	1	0	SHIFT TO LEFT
	1	0	0	FAST SHIFT TO LEFT
	0	1	1	SHIFT TO RIGHT
	0	0	1	FAST SHIFT TO RIGHT





S.1	L=0, M=1, R=0
S.2	L=1, M=1, R=0
S.3	L=1, M=0, R=0
S.4	L=0, M=1, R=1
S.5	L=0, M=0, R=1

Figure 3.6: Flow Chart for the Line Following System

```
if(explore) {              // If we're told to explore
DDRB & = 0X00;             // Configure B7:B0 as inputs (Momentary Switches)
PORTB |= 0XFF;             // Enable pull-ups on B7:B0
        if ((PINB&0X05)==0)                 //State 1
        {
                bal_proc_wr.fwd_rev =35;    // FORWARD
                bal_proc_wr.steer =0;
                }
        else if ((PINB&0X01)==0)    // State 2
        {
                bal_proc_wr.fwd_rev = 35;   // SHIFT TO LEFT
                bal_proc_wr.steer = -35;
                }
        else if ((PINB&0X03)==0)    //State 3
        {
                bal_proc_wr.fwd_rev = 35;   // FAST SHIFT TO LEFT
                bal_proc_wr.steer = -55;
                }
        else if ((PINB&0X04)==0)    //State 4
        {
                bal_proc_wr.fwd_rev = 35;   // SHIFT TO RIGHT
                bal_proc_wr.steer = 35;
                }
        else if ((PINB&0X06)==0)    // State 5
        {
                bal_proc_wr.fwd_rev = 35;   // FAST SHIFT TO RIGHT
                bal_proc_wr.steer = 55;
                }

} else {
        bal_proc_wr.fwd_rev = 0;
        bal_proc_wr.steer = 0;
}
```

Figure 3.7: Programming for Line Follower Algorithm

3.3 Software Review

This section basically is a discussion on programming software's that going to be used to write and program the ATMEGA 32 microcontroller. The step to compile the running program and procedures in program loading also discussed here.

3.3.1 Brain Board Code Editor

This Brain Board Code Editor is software specially designed and dedicated to Balbot. The C programming can be easily written and edited using this software. This software also responsible in compiling and running the C programming thus generate a functional hex file which is to be loaded into the microcontroller. The steps to write a C programming using this software is as followed:

1. "Brain Board Code Editor" icon is clicked as in Figure 3.8.

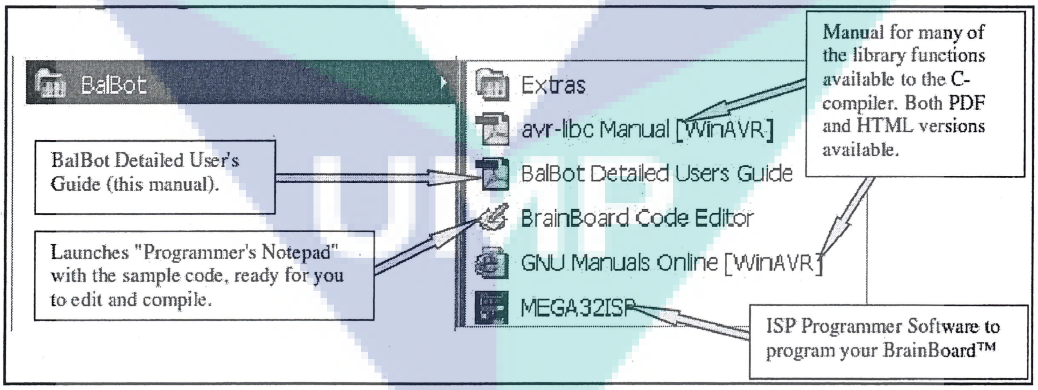


Figure 3.8: Icon for Brain Board Code Editor

2. Programmer's notepad is opened up with the sample of default project automatically loaded. This sample project includes a makefile, several *.c files, and several *.h header files.

3. The file balbot.c is been ensured that currently being displayed in the programmer's Notepad main window as shown in Figure 3.9.

4. The project is compiled by selecting "Tools", then "[WinAVR] Make All". In the window at the bottom of the screen, the project compiling by the AVR-GCC compiler shows that

Errors: none

-------end-------

>Process Exit Code: 0

5. A file called "balbot.hex" will be automatically created, when a successful compilation occurred,. This file contains the raw data to be programmed into the ATMEGA32 microcontroller.

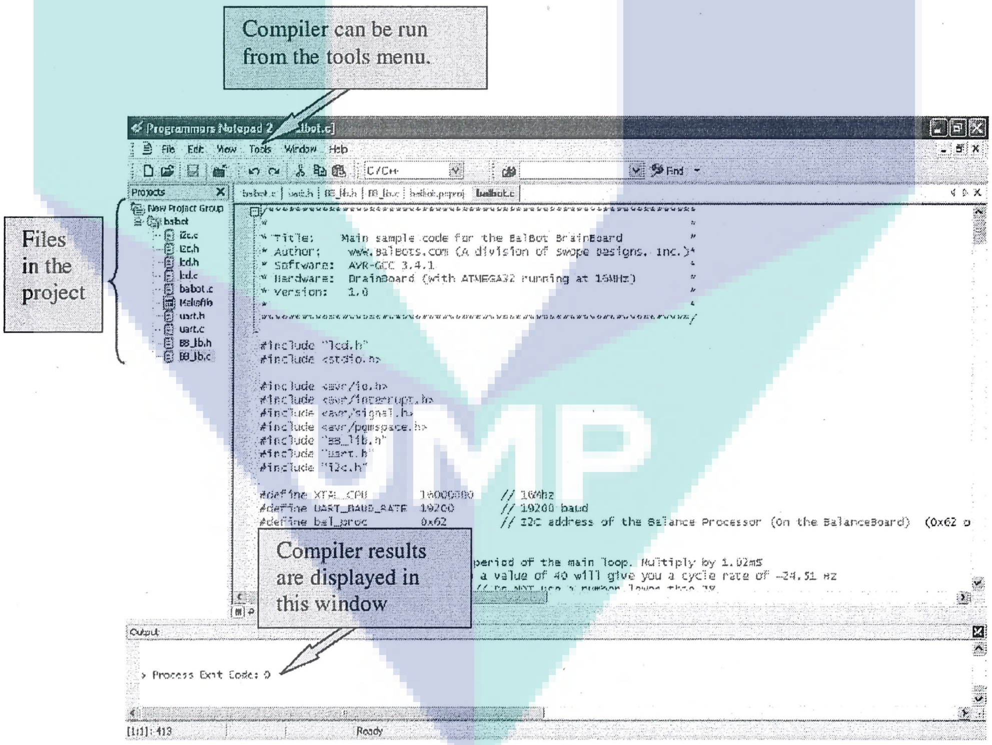


Figure 3.9: Description on Programmer's Notepad 2 Window

3.3.2 Mega32 ISP Programmer

The Mega32 ISP Programmer is convenient software that used to interface with the provided full featured programmer to program the microcontroller. This programmer will load the hex file generated by Brain Board Code Editor in to ATMEGA32 on the Brain Board through the connection between the parallel port on PC and the ISP connector on the Brain Board. The procedures to program the ATMEGA32 are showed in appendix A:

All the steps also shown in Figure 3.10 are to make it clear to loading a program into ATMEGA32 [10].

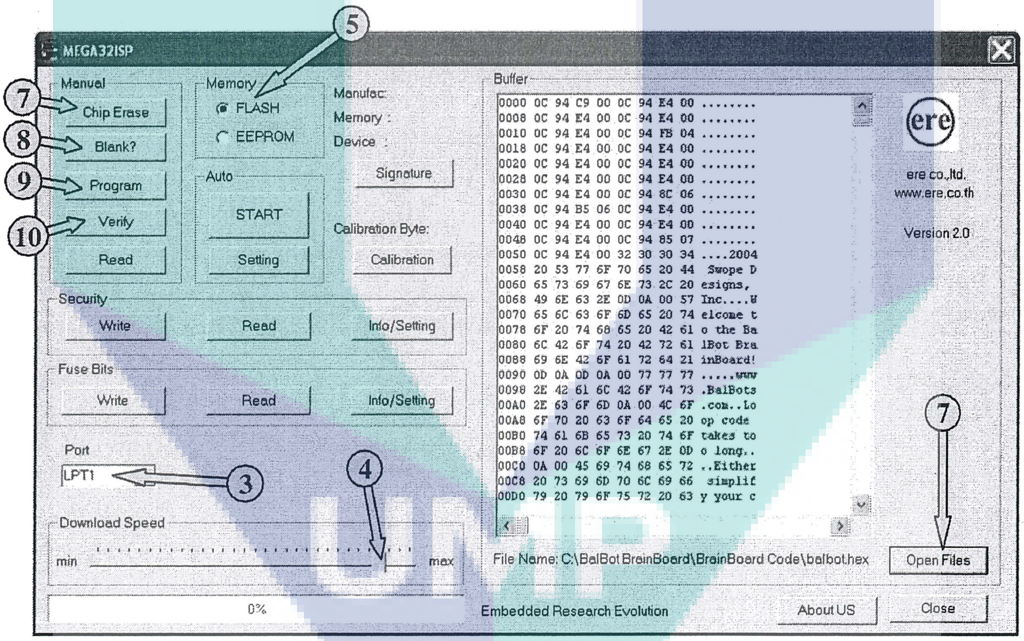


Figure 3.10: MEGA32ISP's GUI with Steps for Loading Program

3.4 Hardware Review

This sub topic will discuss about components part in the anatomy of Balbot Advanced as shown in Figure 3.11 that had been used included Expandable Frame System, Brain Board, Balance Board, DC Gear Motors, Analog GP2D120 IR Distance Sensors, Full Featured Programmer and RS232 Serial Cable for communication with PC. The specification of Balbot is been listed in Appendix A.

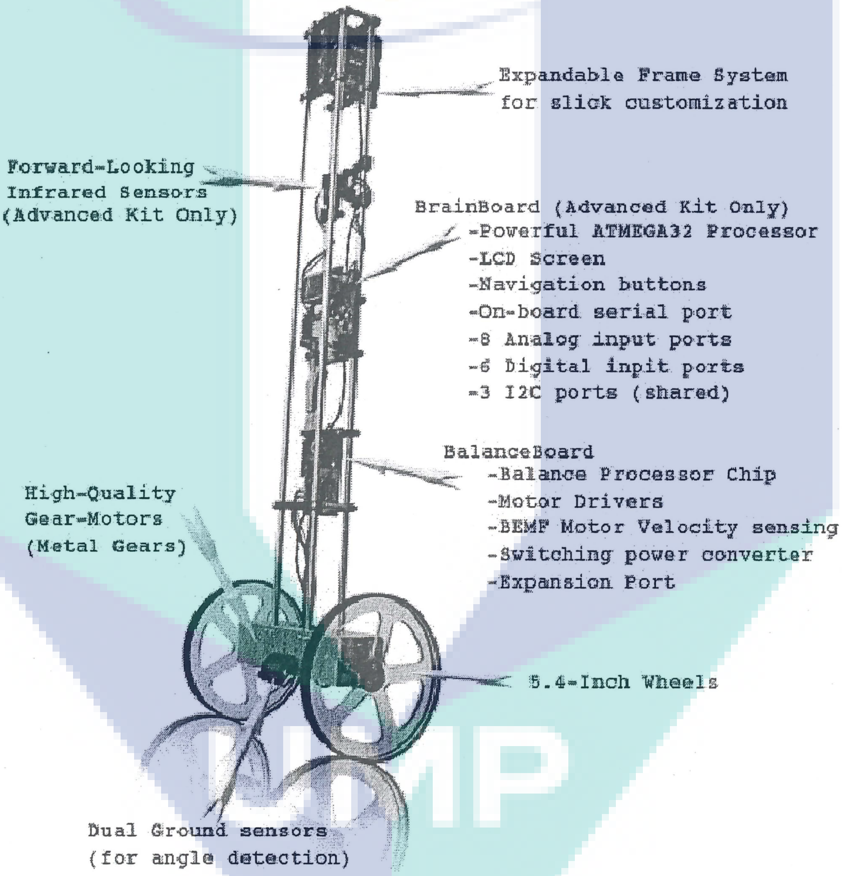


Figure 3.11: Anatomy of Balbot

3.4.1 Expandable Frame System

The Expandable Frame System (EFS), made primarily of aircraft aluminums, allows unlimited form factor changes, sensor additions, board additions with only a screw driver as shown in Figure 3.12. The robot comes equipped with 2 Powerful DC Brushed Gear motors, 2 Current-limiting motor controllers; 2 Forward-looking infrared sensors, 2 Infrared ground sensors and Back-EMF sensors (to detect wheel velocity).

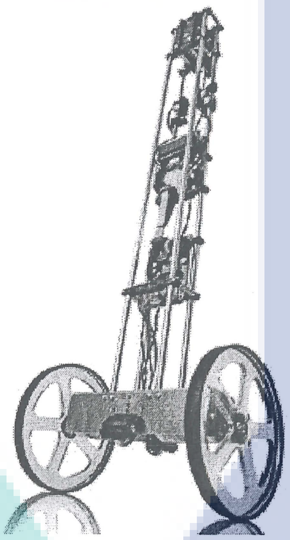


Figure 3.12: The Expandable Frame System (EFS)

3.4.2 Balance Board

Balance Board is a circuit board that plays an important role in balancing the Balbot. The Balance Board consists of Balance Processor Chip (BPC), motor drivers, BEMP motor velocity sensing, switching power converter and expansion port as shown in Figure 3.13 while Figure 3.14 is showing the side view of the Balance Board.

The power input terminals of the Balance Board required a 14.4V to 18V power source connected in order to operate well. There is a mode button which used for modes changing as well as to initiate calibration. The DC-DC converter on board is a switching regulator coverts battery input voltage to a regulated 5V. This high-efficiency converter significantly saves power over linear regulators. This helps extend battery life, and reduces heating of the conversion electronics. The on board motor drivers are sophisticated $\pm$ 3A (peak $\pm$ 6A) DMOS PWM motor controller. In addition to basic motor control functions, there is an analog circuitry on the BPC circuit board that is able to determine the velocity of each motor based on back-electromotive force (BEMP), without encoders or sensors. The Sharp GP2D120 Infrared Distance Sensors are connected to the ground sensor connectors where the BPC reads from these sensors to determine the angle of the Balbot relative to the ground.

The core of this board is the Balance Processor Chip (BPC) which is a programmable System-on-a-Chip contains analog and digital circuitry, as well as microprocessor, RAM, and FLASH memory to store calibration data. The BPC is responsible for determining each motor's velocity, each ground sensor's distance, and to accordingly perform real-time control of the motors to keep the robot balanced. This Chip can also accept commands from an external source (over an I2C bus) to move the robot forward, reverse, left or right. It will also report key information such as motor velocities, robot angle, and battery voltage.

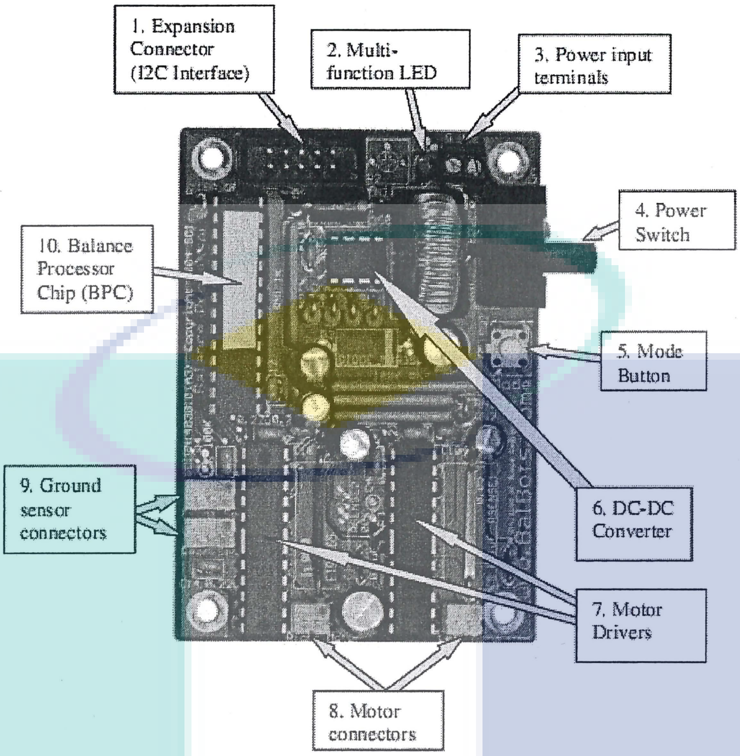


Figure 3.13: Balance Board's Components Overview

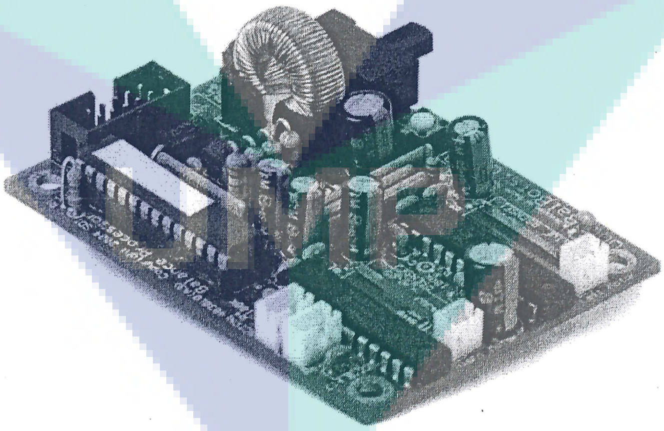


Figure 3.14: Balance Board Side View

3.4.3 Mode of Operation

The Balance Board has 7 basic modes that it can operate in but only mode 0-2 can access by using the mode button. Modes 0-2 can be easily used without programming. Modes 3-5 are similar to modes 0-2, except that they allow custom control variables in the control system (K1, K2, and K3) to be used. In all these modes (0-5), the BPC perform real-time control that allows the Balbot to successfully balance. Mode 6 is a very unique mode as the BPC does not try to balance the Balbot, it only reads and reports sensors information and also controls motors according to the data received over the I^2C interface. Hence, Balbot can only balance if existed external controller possessed its own real-time control system to balance the Balbot [3]. The modes descriptions are shown in Table 3.2.

In this project, mode 0 is used for the implementation of a line follower algorithm for line following systems. In this mode, two types of algorithm which are balancing or line following that can be chosen to be executed via the navigation button on the Brain Board. When the balancing function set, the LCD screen will display "Balancing…." and balance algorithm will be execute. The Balbot will balance itself and will attempt to stand still. On the other hands, when the setting changes to line following function, the LCD screen will display "Line following….". In the mean time, Balbot will continuously execute the Line follower algorithm in the ATMEGA32 to perform line following task once the Balbot achieved its balance state. In this way, the Balbot can easily drive according to line in a controlled manner.

In addition, Sleep Mode is a mode that motors are tri-stated (no power is sent to them, and electronic braking is not turned on). Thus Balbot will not balance and respond to commands from the Brain Board. The BPC chip will still be powered up and reading from the IR distance sensors. It will take BEMP reading as well. Besides that, there is also Nap Mode which motors will stop moving (tri-stated) for 3 seconds. After that, BPC will continually read from the ground sensors. If any significant changes detected, it will automatically switch to Balancing mode.

Table 3.2: Modes of Operation in Balance Board

Mode	Name	Control Variables Used	Description	LED Status
0	Normal Balancing	Default	Normal balance, free moving. (DEFAULT at power-on)	Solid
1	Sleep Mode	Default	Motors tri-stated	1 blink
2	Nap Mode	Default	Motors shut down until unit is bumped. Then switch to mode 0.	2 blinks
3	Normal Balancing	Custom	Custom control variables are used, free moving	3 blinks
4	Sleep Mode	Custom	Motors tri-stated (custom variables are loaded)	4 blinks
5	Nap Mode	Custom	Motors shut down until unit is bumped. Then switch to mode 3.	5 blinks
6	Custom control Algorithm mode	N/A	Dummy slave mode. Control algorithm is bypassed. I^2C definitions are different!	6 blinks

3.4.4 Brain Board

Brain Board is a microcontroller board that can be programmed in order to customize the Balbot as shown in Figure 3.15. The control of the system performance is powered by Atmel ATMEGA32 Microcontroller which has a lot of power to control the Balbot. The key features of this chip are listed as below:

1. 16 MIPS (with included 16 MHz Resonator)
2. 32K Bytes FLASH
3. 1K EEPROM
4. 2K internal SRAM

5. 8 Channel, 10-bit DAC
6. I^2C Interface
7. 4 PWM Channels
8. 3 Timer/Counters

This Brain Board consist of a 16x2 Alpha-Numeric LCD Display and serial port which used to connect the Brain Board to PC serial port by using a DB9 serial extension cable provided. There is four navigation buttons which can be used for anything that had been programmed to be navigate while the reset button will only reset the ATMEGA32 but not Balance Board. The ISP (In System Programming) Connector is used to program the ATMEGA32 without the need to remove the chip each time through the programmer and programming software provided.

The Brain Board had prepared 8 connectors for analog/digital inputs where each of these 8 connectors provide +5v, ground, and a direct connection to one of the ADC (Analog to Digital Convertor) channels on the ATMEGA32. There are also 6 specific connectors for digital inputs/outputs where each of these connectors provides +5V, ground, and 2 direct I/O connections to the ATMEGA32 as listed in Table 3.3.

Table 3.3: Digital I/O Ports Pin Description

Connector#	Pin 1	Pin 2	Pin 3	Pin 4
0	PB0 (XCK/T0)	Ground	PB1 (T1)	+ 5V
1	PB2 (INT2/AIN0)	Ground	PB3 (OC0/AIN1)	+ 5V
2	PB4 (/SS)	Ground	PB5(MISO)	+ 5V
3	PB6 (MISO)	Ground	PB7 (SCK)	+ 5V
4	PD4 (OC1B)	Ground	PD5 (OC1A)	+ 5V
5	PD2 (INT0)	Ground	PD3 (INT1)	+ 5V

Connectors #2 and #3 are sharing signals with the navigation buttons. Therefore, do not use these connectors if still wish to use the navigation buttons! Also note that connector #4 share signals with the servo connectors. Thus, do not use #4 if wish to directly connect servo!

The Brain Board is receiving power via system connector. In addition, the system connector also allowed I^2C connection to be established. This connector is specifically designed for communication with Balance Board for data exchanges. The pin out is been describe in Table 3.4 as follows:

Table 3.4: System Connector Pins Description

Pin No.	Name	In/Out	Description
1	/Reset	Out	When the BPC is power cycled, it will pull this line low momentarily. External processors may have this connected to their reset input to be reset automatically.
2	/I2C_EN	In	When this line is grounded, the I2C circuitry inside the BPC will become active. This line should be tied to ground on any external board that wishes to communicate with BPC over I2C.
3	Cycle	Out	The rising edge of this signal signifies the start of a new control loop cycle. This signal may be used by an external processor to synchronize its control activities with those of the BPC.
4	I2C_SCL		I2C Clock line
5	Vbatt	Out	Battery voltage. Warning this voltage is much higher than most digital circuitry can handle, so do not use it to directly power circuitry!
6	GND		Ground
7	+5V	Out	Regulated 5V power to run external electronic. Note: The power consumption of all total external electronic should be less than 800 mA.
8	I2C_SDA		I2C Data line
9	+5V	Out	Same as pin 7. Use both pins for more current capability.
10	GND		Ground

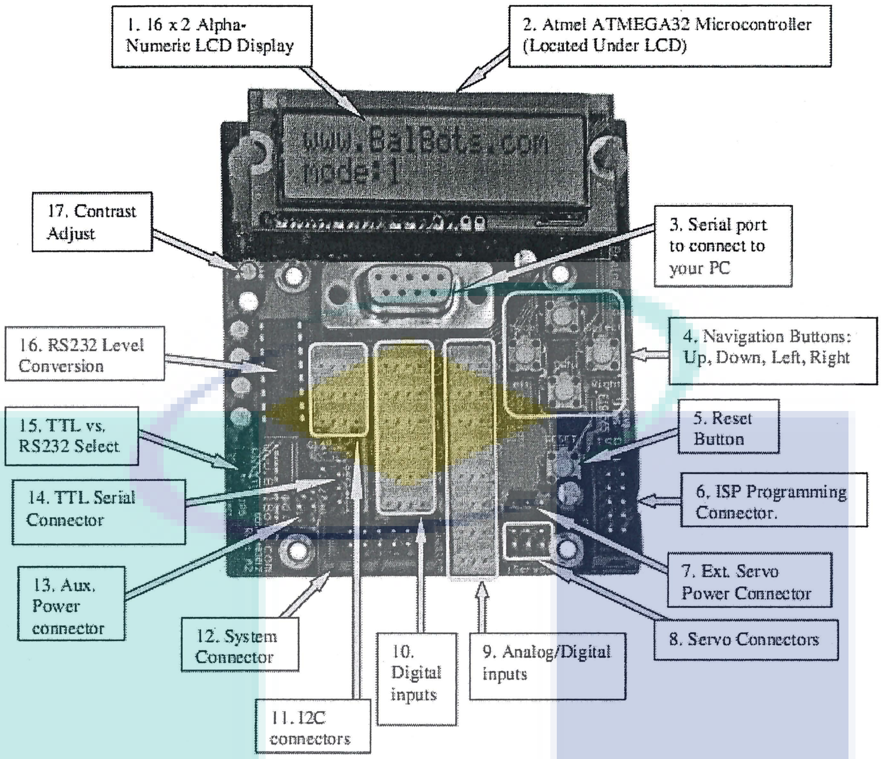


Figure 3.15: Brain Board's Components Overview

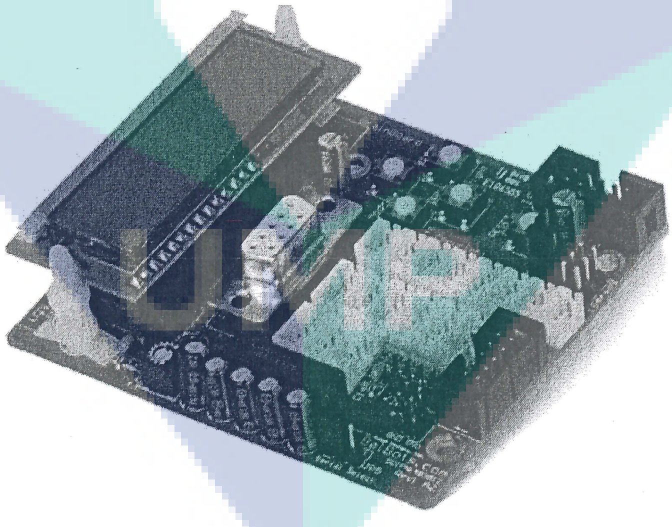


Figure 3.16: Brain Board Side View

3.4.5 Analog GP2D120 Infrared Distance Sensors

Analog GP2D120 IR Distance Sensors covered for a range 4 to 30 cm and these sensors provide non-linear analog output voltage versus distance to reflective object. The object that is detected by this sensor is distance from the sensors to the flat floor surface. The picture of this analog distance sensor is as shown in Figure 3.17.

The analog distance sensors also called as ground sensors which play a dual-role. First, they detect the position of the ground so that the balance board can compute the angle and angular velocity of the Balbot relative to the ground. The angle is used by the Brain Board's balance Processor Chip (BPC) to control the motors and balance the robot. The robot will balance on its own, respond to force from the environment and ensure it's always balances.

As long as the ground is a level surface, the angle relative to ground is also the angle relative to Earth's gravitational pull, and so BPC is able to resist this gravitational pull and keep the Balbot in balance. If the ground is not level, the BPS will be trying to control the Balbot relative to ground and not relative to gravity. This will result in the Balbot drifting off in the downhill direction.

The second role of the ground sensors is to detect whether or not the Balbot is on the ground and in a relatively upright position. The Brain board will automatically shut down the motors to save batteries, if one or both ground sensors do not detect an object (usually because the Balbot is lifted off the ground).

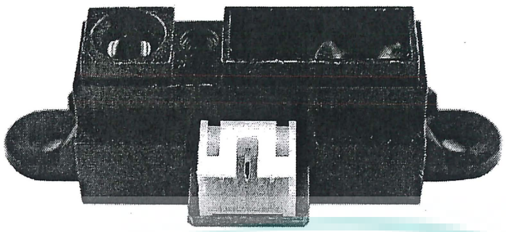


Figure 3.17: Analog GP2D120 Infrared Distance Sensor

3.4.6 Infra-Red Sensor Module

An infrared sensor is an electronic device that emits and/or detects infrared radiation in order to sense some aspect of its surroundings. This sensor works by illuminating a surface with infrared light via transmitter diode; then the receiver diode will pick up the reflected infrared light, based on its intensity determines the reflectivity of the surface in question. Light-coloured surfaces will reflect more light than dark surfaces. This allows the sensors to detect a dark line on a pale surface, or a pale line on a dark surface.

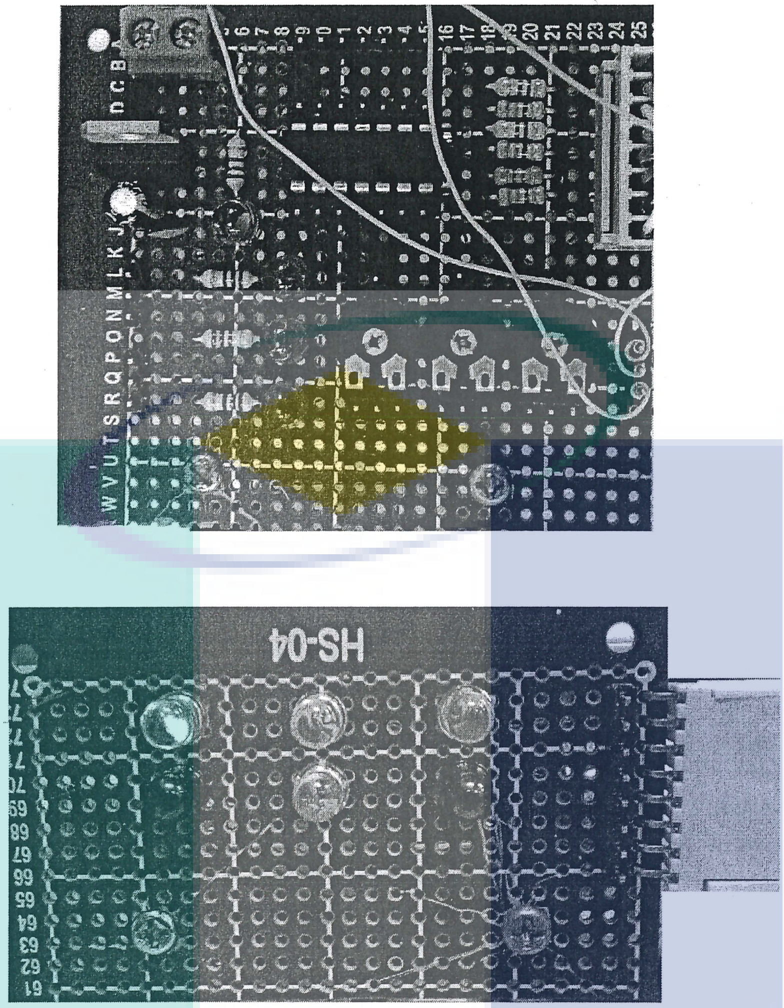


Figure 3.18: The Line Follower Controller and IR Sensors

3.4.7 DC Gear Motor

In order for the Balbot to remain in a vertical stable state, selection of good DC gear motor is vital. DC gear motor with high torque output and fast RPM is ideal to be used on two wheel robot system. The DC gear motor in Figure 3.20 is custom make by Balbot's company which possessed high torque and high RPM to conduct the movement of the Balbot.

Figure 3.19: DC Gear Motor

3.4.8 Full Featured Programmer

Full featured programmer used to load the program into ATMEGA32 on the Brain Board, by using the ISP programmer, which consists of a small circuit board and cable with a 25-pin parallel port connector at one end and a 10-pin connector on the other end [10].

The ISP programming port uses PB5, PB6, and PB7 pins on the ATMEGA32. These pins shared by the navigation buttons. Therefore, the navigation buttons cannot be pressed during programming. Otherwise, the programmer may hold down one or more port B pins (The Brain Board will act like a button is pressing by someone).

Figure 3.20: Full Featured Programmer

3.4.9 RS232 Serial Cable

This is a 6 ft. RS232 Extension Serial cable with DB9 Male / DB9 Female, 28 AWG, Fully Shielded, and Beige Outer PVC Jacket. Connectors are moulded and the contacts are gold plated. This cable can be used to connect between the Brain Board and a standard PC serial port and it is not used to program the board. This serial cable is used to get the output data from Hyper Terminal communication.

Figure 3.21: RS232 Serial Cable

CHAPTER 4

RESULT AND DISCUSSION

4.1 Introduction

In this section, result from the balancing and the line follower control algorithm obtained and observed from the Balbot will be describes. The data were collected and recorded through the observation. The discussion section will highlight about initial calibration, efficiency in balancing mode, and the performance of line follower control system hardware and also algorithm.

The analysis phase provides an opportunity to assess and evaluate the Balbot effectiveness and efficiency in maintaining stability and executing line following task. This allows a comparison to be undertaken between the actual system performance and the anticipated project objectives. The opportunity to calibrate and perform additional fine tuning of the design is also explored with the aim of achieving the best result possible in mind.

4.2 Balancing System

The Balancing System is considered stabilize when both an analog GP2D120 IR distance sensor obtaining a same value of reading. Both of these two values from each ground sensor are been subtracted with each other and it will result in zero reading in distance of cm for stabilize systems. In this circumstance, the Balbot balances on its own and always in place above the center of gravity. If the Balbot is been on uneven colored distribution surface or inflate platform, the analog reading between the two ground sensors will have some significant value. Thus the DC gear motor will function to drive on and keeps the Balbot back in normal balancing. As a result from balancing system, the Balbot will oscillate in a small angle which is considered as in stabilize system.

There are some weakness on the GP2D120 IR distance sensor where the sensor light reflection will be alter easily due to the colored of the surface especially glossy black colored surface and also high contrast. Once the sensor sensed the glossy black colored, the functionally of the sensor will be disturb and the output of the sensor is unstable. The distance sensors are too sensitive toward the colour changes on the background surface and concluded that the Balbot is under imbalance state thus it ordered the motor to react towards the error. As a result, the stability of the balancing system will be affected.

Corresponding towards the problems, background surface colour of the line following track should be taken into high consideration in order to perform a well balancing state thus proceed with the line follower task. The initial calibration for the ground sensors also plays an important role in providing high efficiency balancing performance.

4.3 Line Following System

Based on the line following system performance on the provided line track as shown in Figure 4.1, the line following algorithm has achieve the objective of line follower Balbot development. It is observed that, the movement of the Balbot is on the line track. The three infrared sensors will always guide the Balbot to be following the line track based on line follower control algorithm that programmed in the microcontroller.

There are two linear lines and two semi circle curves that had been purposely drawn to test the accuracy and performance of the interfacing between line follower control algorithm and the Balbot reaction towards the line track. The linear line is to test the basic performance of the algorithm. The Balbot have to keep on following the linear track in order to achieve the objective this project. The curve line will test the accuracy of the sensor and feedback respond toward the corner. The Balbot have to be slow down during the corner so that the Balbot will not out of the line. The Balbot seem to be not performing a smooth movement on the curve line due to the sufficient of sensors to provide more accurate data and also the control algorithm is lack of efficiency in giving good and effective respond toward the changes of state at the curve line.

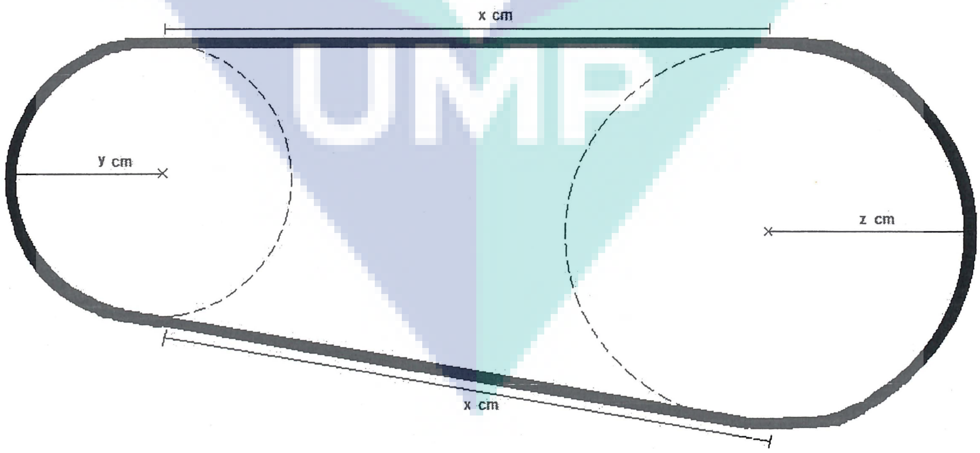


Figure 4.1: Line Track for the Line Follower Balbot

The difficulty in programming the Balbot has to be taking into consideration where syntax error easily occurred due to lake of understanding about the Balbot programming language. Hence, it's required some revision and knowledge in simple C programming to encounter such problem.

4.4 Initial Calibration

Before Balbot is used for the first time it must be calibrated, so that it reads the sensors correctly. Calibration may also be necessary if the center of the gravity changes (i.e. you add more accessories to the Balbot), or if the Balbot is used on a drastically different surface.

1. The Balbot is ensured on a flat, level surface, so that both ground sensors have an un-obstructed view of the ground.

Caution: The ground sensors are been ensured that never obstructed during operation! Otherwise the Balbot will get confused and think it's falling when it's not. This will result in quick motor reactions and the Balbot will likely fall down.

2. With the Balbot resting on a level surface, carefully balance it upright with one hand. You can judge whether you are balancing it or not by letting go of the Balbot. The Balbot stayed upright and motionless for a second or two, before it begins to fall over.

3. The mode button is pressed and held down with the other hand, while holding Balbot steady with one hand. Then the power is turn on while holding the mode button down. The mode button is kept held down and the Balbot is kept still for 5 seconds. It is important that the Balbot remains balanced and completely still during this time and the LED on the Balbot blinking slowly (mode 1).

4. The robot is turned off. The Balbot is now calibrated. The calibration settings are stored in FLASH memory and will not be lost by shutting down the Balbot or changing the batteries.

5. The Balbot is been turned back on by sliding the main power switch. Then, let go and allowing the Balbot to balance on its own. If it does not balance well, repeat the calibration procedure. The troubleshooting guide near the end of Balbot Assembly Manual can be consulted, once it still does not balance well.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Introduction

Conclusion and recommendations are essential to include comment on each aspect undertaken as part of the project. The analysis and evaluation stage provided the useful information that can be used to ascertain the successfulness of several key of performance. This is then expanded into several recommendations for improvements, addition of extra capabilities and future investigation. From the observations and discussions previously presented, this chapter will provide recommendations for future work, improvements and additional areas of investigation. This will provide an opportunity or platform to share an awareness of problems encountered that would complement the future work recommendations.

5.2 Conclusion

This project as a whole was very successful in driving a line follower two wheeled balancing robot with all the considerations necessary in ensuring it could meet the required capabilities and objectives. The software and programming parts is achieving satisfaction level but the hardware part was so disappointing with the majority of the difficulties encountered in these areas. This could be overcome with time but unfortunately, time was not plentiful. The following recommendations provide direction for future work and suggested areas of further investigation.

5.3 Recommendations

It is recommended that this approach be undertaken again but with an experience programming background. The difficulties encountered could have been easily overcome if significant time was available to further develop the programming to perform smooth line following task. The following sub-sections contain the suggested improvements, additions and possible areas for future investigation.

5.3.1 Improvement in Control Algorithm

Improvements to the current version of the Balbot can be achieved through the microcontrollers programming. This would provide the fundamental benefits faster control responses, effective and efficient computations and improved overall system performance. By improving the programming implementation a smooth line follower Balbot can be establish. The implementation of full PID control in the line following algorithm will result in high improve on the line following system and also the accuracy in motor speed control will increase.

5.3.2 Additional Features and Capabilities

Additional feature or capabilities that may improve the effectiveness, efficiency or operability of the Balbot include the addition of IR Sensors for line following and Analog IR Distance Sensors for obstacle avoidance through the optimum utilization of the extra digital and analog ports. This would allow Balbot to interact with its surrounding environment thus providing larger impact on the operability it may now achieve. This would increase the accuracy in line detection and better performance in line following task. The number of sensors implemented would increase the complexity of the control systems in place but enhanced capability may be beneficial.

A webcam vision could be implemented to the Balbot as to allow the robot to autonomously navigate, identify and avoid obstacles through the added vision capability. To better transverse uneven terrain, the wheels could be enhanced by increasing wheels diameters and width. This could allow Balbot to travel over small obstacle or uneven route while maintaining balance.

Designing and constructing an high quality chassis or cover for the Balbot is necessary. A chassis that allows the IR Distance sensors to easily see through would be ideal, allowing a smooth and nice covering for the outer side. This would provide high protection to Balbot against dust, water and other forms of intrusion into the sensitive circuits within. A cushion belt could be added to the Balbot to allow a softer collision in the event one may occur.

5.3.3 Future Investigation

The following recommended areas of investigation involve different approaches to the balancing control system. Further fine tuning of PID control algorithm could be implemented into the Balbot balancing system once the Balance Board Processor Chip is by-pass to achieve new output performance. Further research can be conducted to implement intelligent controllers on balancing system. Such intelligent controllers might be artificial neural network controller, fuzzy PID controller or self tuning PID controller.

REFERENCES

[1] S.W. Nawawi, M.N. Ahmad, J.H.S Osman (2008). "Real-Time Control of a Two Wheeled Inverted Pendulum Mobile Robot", Proceedings of World Academy of Science, Engineering and Technology Volume 29 May 2008.

[2] Ooi, Rich Chi (2003). "Balancing a Two-Wheeled Autonomous Robot". University of Western Australia. Thesis B. Mechatronics Engineering.

[3] Ong, Yin Chee; Abidin, M.S.B (2006). "Design and Development of Two Wheeled Autonomous Balancing Robot", Center for Artificial Intell. & Robotic (CAIRO), University Technology Malaysia, Kuala Lumpur. 4th Student Conference on Research and Development (SCOReD 2006), Shah Alam, Selangor, MALAYSIA, 27-28 June, 2006.

[4] Vivien Coelho; Stanley Liew; Karl Stol; Guangyu Liu (2008). "Development of a Mobile Two-Wheel Balancing Platform for Autonomous Applications". Department of Mechanical Engineering, the University of Auckland. 15th International conference on Mechatronics and Machine Vision in Practice (M2VIP08), 2-4 Dec 2008, Auckland, New-Zealand.

[5] José Miranda (2009), "Application of Kalman Filtering and PID Control for Direct Inverted Pendulum Control". Master Thesis, California State University, Chico Spring 2009.

[6] R. Hollis, "BallBots," *Scientific American*, October 2006. Retrieved February 4, 2009.
Available at the World Wide Web:
http://www.sciam.com/article.cfm?id=ballbots

[7] Pakdaman, M.; Sanaatiyan, M.M. (2009), "Design and Implementation of Line Follower Robot," *Computer and Electrical Engineering, 2009. ICCEE '09. Second International Conference on* , vol.2, no., pp.585-590, 28-30 Dec. 2009

[8] Priyank Patil (2010), "AVR Line Following Robot," Department of Information Technology K. J. Somaiya College of Engineering Mumbai, India. Retrieved Mar 5, 2010.
Available at: http://www.kmitl.ac.th/~kswichit/ROBOT/Follower.pdf

[9] Swope Design. Inc., "Balancing Robots Made Easy", Available at:
http:// www.Balbots.com, 2004.

[10] Miller Peter (2008), "Building a Two Wheeled Balancing Robot", University of Southern Queensland, Faculty of Engineering and Surveying. Retrieved Nov 18, 2009. Available at: http://eprints.usq.edu.au/6168/

Appendices

Dual Mode Navigation for Two-Wheeled Robot

N.M Abdul Ghani, L.K. Haur, T.P.Yon

Abstract**—** This project relates to a two-wheeled self balancing robot for transferring loads on different locations along a path. In more particularly, this robot apply dual navigation functions to navigate efficiently along a desired path. A plurality of distance sensors mounted at both sides of the body for collecting information on tilt angle of the body and a plurality of speed sensors mounted at the bottom of the body for collecting information of the velocity of the body in relative to the ground. A microcontroller for processing information collected from the sensors and configured to set the path and to balance the body automatically while a processor operatively coupled to the microcontroller and configured to compute change of the tilt and velocity of the body. A direct current motor operatively coupled to the microcontroller for controlling the wheels and characterized in that a remote control is operatively coupled to the microcontroller to operate the robot in dual navigation modes.

Keywords**—** Two-Wheeled Balancing Robot, Dual Mode Navigation, Remote Control, Desired Path.

I. Introduction

An inverted pendulum model can be described as a pendulum which has its mass above its pivot point. Thus, with the uniqueness and unstable nature of the system, it requires good control systems which capable to control itself in upright position dynamically balanced. Indeed, various type of controllers have been greatly implemented by the robotics researchers around the world currently such as Pole-placement, Linear Quadratic Regulator (LQR), Proportional-Integral-Derivative (PID), Fuzzy Logic and so forth.

In this project, this two-wheeled robot that equipped with self-balancing platform in order to keep balance while navigated on flat terrain. However, the navigation function is expanded with wireless PS2 controller due to its faster response and stable transmission. The receiver that plugged to the PSC28A PS2 I/O converter would receive the signal from the PS2 controller. From here, the microcontroller of the Brain Board which connected to the converter will collect the signal data, so that appropriate instruction will be sent to the Balance Board for further navigation with desired speed and direction.

N.M Abdul Ghani is with the Electrical & Electronics Engineering Faculty, Universiti Malaysia Pahang, 26600 Pekan Pahang (phone +609-4242110; e-mail: normaniha@ump.edu.my)

L.K Haur is with the Electrical & Electronics Engineering Faculty, Universiti Malaysia Pahang, 26600 Pekan Pahang (e-mail: jasonlkh_87@hotmail.com)

T.P.Yon is with the Electrical & Electronics Engineering Faculty, Universiti Malaysia Pahang, 26600 Pekan Pahang (e-mail: pokemon_tpy@hotmail.com)

Instead, the speed of the platform is acquired by the BEMP motor velocity sensor. The entire control algorithm that computed in C programming will react against the varies of velocity and tilt angle which detected by the ground sensors of GP2D120 infrared sensor of the two-wheeled mobile robot.

II. Literature Review

The researchers from the Industrial Electronics Laboratory at the Swiss Federal Institute of Technology, Lausanne, Switzerland had successfully implemented a scaled down prototype of a Digital Signal Processor controlled two-wheeled vehicle based on inverted pendulum model with weights attached to the system to simulate a human driver, namely JOE. In brief, the control system used to achieve the stability of the system is based on two state-space controllers, utilizing sensory information from a gyroscope and two incremental encoders mounted on each dc motor [1].

And yet another two-wheeled balancing robot similar to JOE developed by David P. Anderson which makes use of commercially available inertial sensor and the position information from the motor encoder to balance the system, called nBot. The mobile robot would remain balanced as the system could able to drive the two wheels in the direction that the upper part of the robot is falling. Four terms used to define the motion and position of the system which then summed and fed back to the platform as a motor voltage that is proportional to torque in order to balance and drive the robot [2]:

i. The tilt angle
ii. Its first derivative, the angle velocity
iii. The platform position
iv. Its first derivative, the platform velocity

Apart from that, Steven Hassenplug who had successfully built a two-wheeled robot called Legway using the LEGO Mindstorms robotics kit. However, two Electro-optical Proximity Detector sensors which based on the Infrared Proximity Detector circuit were used in order to provide the tilt angle of the robot to the controller which programmed in BrickOS, so that a constant distance to the ground can be maintained [3].

In the paper by Peter Miller on building a two-wheeled balancing robot stated that some important elements in designing a stabilize two-wheeled robot system such as inertial sensors, shaft encoders, control algorithm and microcontroller as the brain of the robot to complete computations and process data by executing instruction or programming [4]. However,

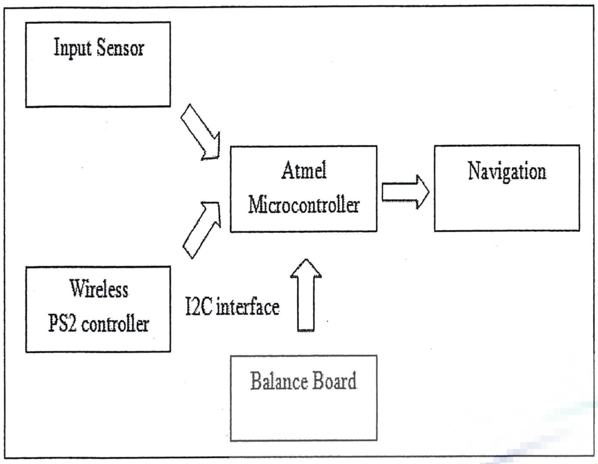


Fig. 2: Block Diagram of Robot System

An external system of microcontroller in the Brain Board will be connected to the Balance Board expansion connector in order to control the Balance Board and to receive sensory information from the Balance Board. However, the data is exchanged over the I2C bus. The BPC equipped in the Balance Board would act as an I2C slave. In order to communicate with it, a separate microcontroller on the other hand would act as an I2C master. The microcontroller will initiate I2C by reading commands to receive data, and I2C will then write commands to transmit data to the BPC. While writing data, the master should always write 10 bytes and it always read 18 bytes while reading data.

Fig. 3: Robot's Front View

A. Balancing System

The GP2D120 infrared distance sensors work as the dual ground sensor are utilized in the robot to acquire the necessary data of tilt angle and angular velocity crucially required to drive the two DC motors in a manner that keep the center of gravity above the wheels all the times. The Balance Board executes the standard Digital Cascaded Control Loop algorithm which built into Balance Processor Chip to keep the robot balance. The control loop runs at a frequency of approximately 40.2 Hz or delay of 0.025 s.

The control algorithm plays an important role in keeping the two-wheeled robot dynamically stable and able to stand upright on flat terrain. Since the robot worked under mode 0 and 1, therefore the gains of K1, K2 and K3 can only be read and modification is not allowed. The K3 gain is responsible to the varied of the velocity of the mobile robot. Meanwhile, the K1 and K2 gains will responsible to the changes of the tilt angle. The output angle will be measured and compared to the desired angle, Calibration angle. By then, appropriate Angle_offset or error will be added to ensure that the output angle similar to the calibration angle.

In a nutshell, the BPC will function to detect the changes of velocity via Back-Electromotive Force (BEMP). The error between the desired velocity and the measured output velocity will be calculated by the BPC. By considering the tilt angle that feedback from the infrared distance sensors, the mobile robot will accelerate in the direction that the upper part falling shown in the Fig. 3 In this condition, the two-wheeled robot will remain stable as both the values of two infrared distance sensors are the same

B. Navigation System

The navigation system of the two-wheeled robot is performed by the wireless PS2 controller as shown in the Fig. 4. However, the major problems dealing with this application may include of difficulties to obtain the suitable connector socket for PS2 which is unique and difficult to source as well as require well understanding in the protocol to communicate with it. Apart from that, the process of implementing the PS2 controller protocol to acquire the status (digital and analog) of each button and analog joystick are troublesome and time consuming.

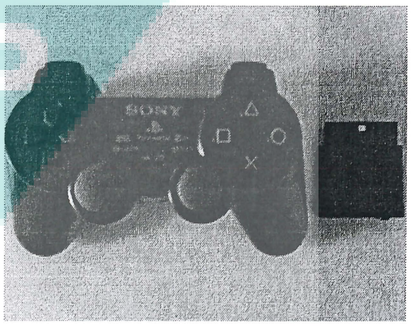


Fig. 4: Wireless PS2 Controller with Receiver

An addition circuit of PS2 I/O converter, PSC28A shown in the Fig. 5 is introduced to help to solve the problems mentioned above. More importantly, it offers a standard connector socket for PS2 controller to plug-in and we can opt to interfacing with or without the microcontroller. Briefly to say that it provides fast and simple way to use the joystick as the input of each button would be converted directly into an output and yet its design is compact, reliable and low cost.

Instead, PSC28A does not only convert the buttons and joystick information on PS2 controller into open collector output with 14 open collector output for 14 buttons, four PWM output for four joystick axis and eight open collector output for 8 joystick directions. Instead, there are also two inputs to control the vibrator motor on the wireless PS2 controller. However, the product specifications and limitations for PSC28A are shown in Table 1.

TABLE 1
ABSOLUTE MAXIMUM RATING FOR PSC28A

Parameter	Min	Typical	Max	Unit
Input voltage	7	12	15	V
PWR Out	-	Input voltage – 0.7	-	V
Current consumption	-	100	300	mA

The designing of position control or navigation system for the two-wheeled balancing robot using wireless PS2 controller is presented in Table 2. In brief, there are four status for the navigation system for the mobile robot which determine where its direction of movement either go forward, backward, turn left or right based on the button pressed on the wireless PS2 controller.

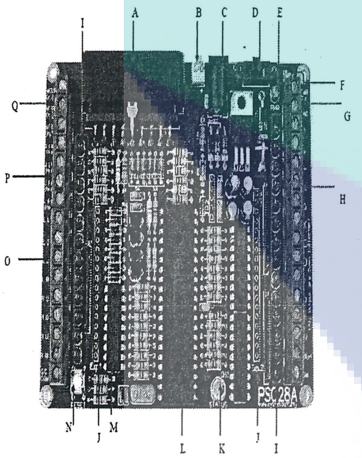


Fig. 5: Board Layout of PS28A

TABLE 2
NAVIGATION SYSTEM

Navigation Button	Output Desired Path
⇧	Forward
⇩	Backward
⇨	Turn Right
⇦	Turn Left

Once the navigation buttons on the wireless PS2 controller are pressed, the PSC28A will process the signal input obtained via the PS2 receiver plugged into it, hence will then be received by the ATMEGA32 microcontroller in the Brain Board. By then, the microcontroller will process the signal input received and identified which button is being pressed, so that appropriate action in the form of signal will be sent to the DC motors for further direction navigation of the mobile robot. However, the navigation system for the mobile robot can be expressed in flowchart which shown in the Fig. 6.

Fig. 7 shows the connection between the ATMEGA32 microcontroller in the Brain Board and PSC28A I/O converter in order to interface with the wireless PS2 controller. Therefore, port A is initially defined to work as the inputs for the PS2 controller. Instead, two out of the six modes is chosen to use: balancing (mode 0) and exploring (mode 1). In this project, the program via Brain Board Code Editor is written in such a way that the mobile robot only will be navigated while in the explore mode.

In addition, two important variables that determine the navigation system, namely bal_proc_wr.fwd_rev and bal_proc_wr.steer.The positive value for bal_proc_wr.fwd_rev indicates it will move forward and move backward if this variable is set to negative value. On the other hand, the positive value for bal_proc_wr.steer indicates it will turn right; else it will turn left if the value is set to be negative.

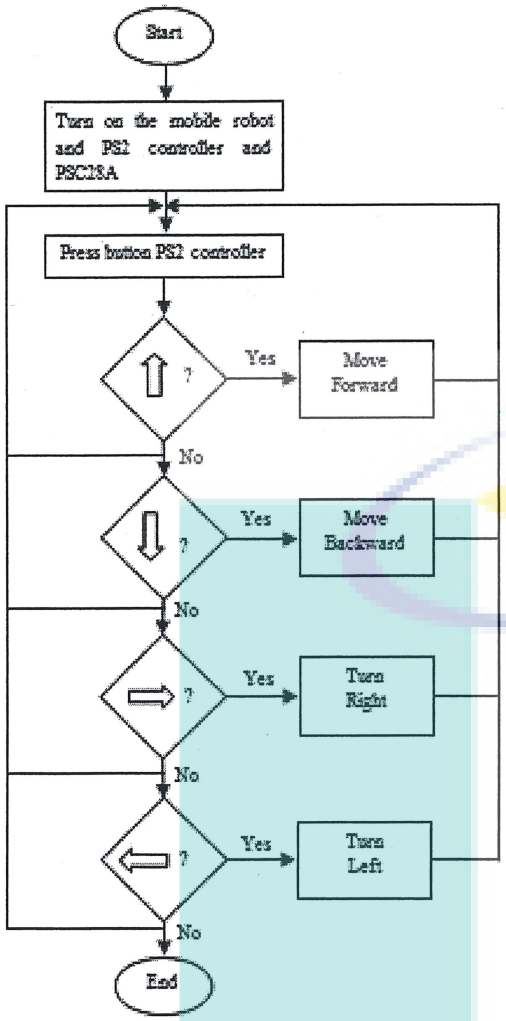


Fig. 6: Flow of Navigation System

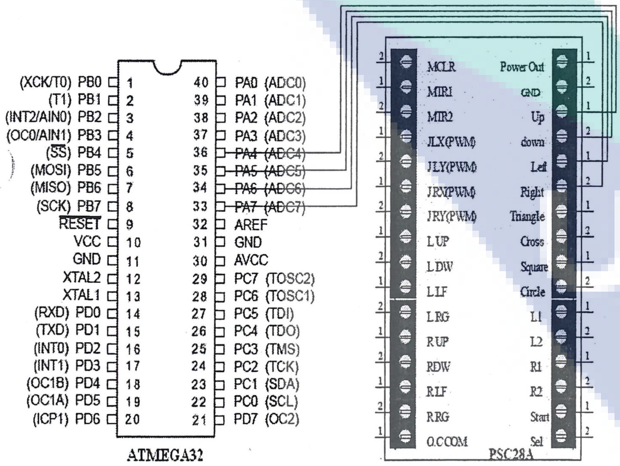


Fig. 7: Connection for Navigation System

IV. Result And Discussion

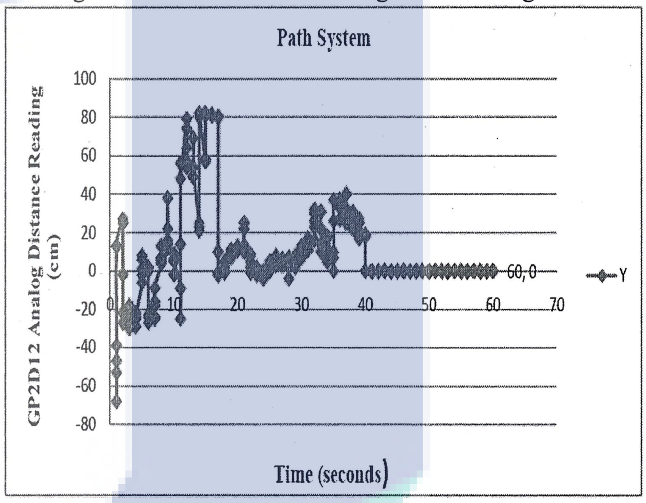
In this section, the result and observation on the robot in terms of auto and wireless modes will be discussed. Instead, this project is divided into two stages: testing stage and implementation stage which will be explained as well.

A. Auto Navigation Mode

The system is considered stable as both the two analog GP2D120 infrared distance sensors having the same values. As for that, the values from each distance sensor will be subtracted with each other. Once the subtracted value obtain is not zero, therefore the mobile robot will oscillate in a small angle performed by the Balance Board by sending signal to the DC gear motor to move in direction that the upper part body of robot is falling. The process will be repeated in order to achieve zero subtracted value, so that the mobile robot would always balance and capable to stand upright on the flat terrain.

The result for path navigation in auto mode is as shown in Fig. 8.

Fig. 8:GP2D120 Distance Reading for Path Navigation



B. Wireless Mode

The wireless PS2 controller which is considered an improved RF device with 2.4 GHz of transmission works greatly within the distance of four meters. However, the mobile robot would always try to balance itself first before performing the navigation instruction. The four buttons of the wireless PS2 controller that will be used for this project are Up, Down, Left and Right buttons. However, these pins of the PSC28A will be connected to the PORT A of PA4, PA5, PA6 and PA7 of the ATMEGA32 microcontroller via connect them to the Digital/Analog inputs of the Brain Board. However, bear in mind that both the PSC28A and Brain Board must be shared a common grounding. Below are the steps to communicate between the two-wheeled balancing robot and the wireless PS2 controller:

i. Before the two-wheeled mobile robot is started to navigate, calibration procedure must be carried out in order it will be stable when the power supply is on. It is done by fuzzy logic controller is implemented in order to ensure the stability of the system. Finally, the MATLAB is chosen as the simulation platform to work for stability and locomotion analysis throughout the system designed.

Similarly, Zatil Hanan from Universiti Teknologi Malaysia who taken the research project on position control of two-wheeled inverted pendulum mobile robot described that the mobile robot would balance on its own body and always place on the center of gravity as both the values of two analog reading GP2D120 IR distance sensor produce the same value. Instead, the GP2D12 is chosen to work as the input for the path control algorithm [5].

Moreover, under the research in [6] of modeling and control of a balancing robot using digital state space approach defined that the modeling is the crucial process of identifying the principal physical dynamic effects to be considered in analyzing a system, whereby the differential and algebraic equations are written based on the conservation laws and property laws of the relevant discipline, and then reduced them to a convenient differential equation model. The dynamic performance of a balancing robot depends greatly on the efficiency of the control algorithms and the dynamic model of the system. Despite, some assumptions and limitations had been mentioned to make the modeling of the mobile robot a reality which stated as below:

i. Motor inductance is neglected and the current through the winding is not considered in the equation of the motion of the motor
ii. The wheels of the robot will always stay in contact with the ground
iii. There is no slip at the wheels
iv. Cornering forces are negligible

The two-wheeled balancing robot requires a control algorithm in order to achieve the stability of dynamics of the system. Basically, it can be divided into two major types of control system: linear and non linear control system. The linear control system consists of proportional-integral-derivative (PID), linear quadratic regulator (LQR) and pole-placement. On the other hand, fuzzy logic and nonholonomic are categorized as non linear control systems.

Ong Yin Chee and Mohamad Shukri b. Zainal Abidin from Universiti Teknologi Malaysia who had undertake the research on design and development of two wheeled autonomous balancing robot .The information on current position and tilt angle of the robot are acquired from the distance measuring sensors (Sharp GP2D12) mounted on an aluminum strip placed at the front and back of the robot. Hence, the analog voltage output from the sensor will converted into digital value via A/D converter of microcontroller then inputs it into PID control algorithm to determine the speed and direction of the motors. Finally, the robot could stand it upright in stable condition and able to be navigated by RC remote control. In brief, these sensors are used to detect the current position and tilting angle of the mobile robot [7].

Rich Chi Ooi who undergoes research on balancing a two-wheeled autonomous robot described on how the PID controller working for the trajectory control. The PID controller is useful as it will ensure that the position errors between the wheels are minimized, so that the robot will move in a straight path. However, the PID controller used a set of tuning rules which is based on the Ziegler-Nichols method. The encoder reading difference is passed through the PID loop to obtain the required feedback to be added to the motor controller [8].

III. METHODOLOGY

The aim of this project is to apply dual mode navigation function on the robot via the application of wireless PS2 controller and auto navigation mode. In brief, the analog GP2D120 infrared distance sensors work as feedback input for the control algorithm illustrated in Fig 1 which responsible to achieve the stability of the entire system shown in Fig. 2 As such, the control algorithm programmed in the Brain Board will send a signal to the Balance Board for further action once changes on tilt angle and velocity are detected.

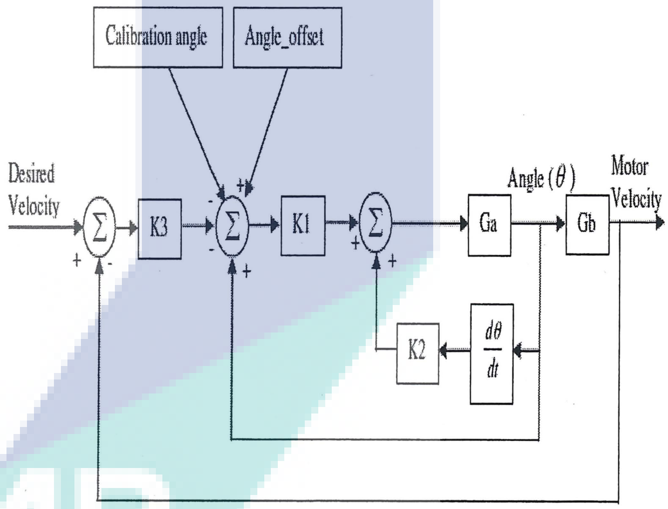


Fig. 1: Digital Cascade Control Loop for Robot

putting it on a flat terrain for a few seconds and calibrate under balancing mode.

ii. The PSC28A device is turned on and the Rx LED of the wireless PS2 receiver will start blinking indicates it started to search for the PS2 controller. While the power of the wireless PS2 controller is being turned on, the Rx LED of the receiver will become static indicates the PSC28A device is communicating with the wireless PS2 controller.

iii. Followed by turn on the power supply of the two-wheeled balancing robot. Switch the mode from balancing mode into explore mode for navigation purposes.

iv. The mobile robot can be wirelessly navigated through the wireless PS2 controller by pressing the Up, Down, Left and Right button.

V. CONCLUSION

In a conclusion, this project has been successfully applying dual mode navigation system over the two-wheeled mobile robot with having the capabilities to balance itself on the flat terrain. In brief, problems should be detected in the earlier stage, so that the interfacing between software and hardware would become much easy to implement. Indeed, the two-wheeled robot responds greatly on the wireless PS2 controller's button pressed. As such, this project can be furthered modified with utilizing vibrator motor and GP2D12 infrared distance sensor to notify the user not to move forward as obstacles are detected.

ACKNOWLEDGMENT

The authors gratefully acknowledged the contributions of S.W Nawawi and Z. Hannan for work on the original version of this document.

REFERENCES

[1] Grasser,F.., D'Arrigo, S.Colombi, and Rufer, *A. Joe: A mobile Inverted Pendulum.* Proceedings of the 2002 IEEE Trans. Electronics. Vol.49, no 1. February. Lausanne, Switzerland: IEEE, 2002. 107-104

[2] nBot Balancing Robot. Available at: http://www.geology.smu.edu/~dpa-www/robo/nbot/

[3] 17nd JAN2003,steve's LegwayURL http://www.teamhassenplug.org/robots/legway/

[4] Miller Peter (2008), "*Building a Two Wheeled Balancing Robot*", University of Southern Queensland, Faculty of Engineering and Surveying. Retrieved Nov 18, 2009. Available at: http://eprints.usq.edu.au/6168/

[5] Zatil Hanan Binti Ahmad Lutpi (2009). *Position Control of Two Wheeled Inverted Pendulum.* Universiti Teknologi Malaysia: Thesis B. Electrical Engineering (Control & Instrumentation)

[6] Herdawatie binti Abdul Kadir (Nov 2005). *Modelling and Control of a Balancing Robot using Digital State Space Approach*, Master Thesis, Universiti Teknologi Malaysia

[7] Ong, Yin Chee; Abidin, M.S.B (2006). "*Design and Development of Two Wheeled Autonomous Balancing Robot*", Center for Artificial Intell. & Robotic (CAIRO), University Technology Malaysia, Kuala Lumpur. 4th Student Conference on Research and Development (SCOReD 2006), Shah Alam, Selangor, MALAYSIA, 27-28 June, 2006.

[8] Ooi, Rich Chi (2003). "*Balancing a Two-Wheeled Autonomous Robot*". University of Western Australia. Thesis B. Mechatronics Engineering.

Two Wheels Balancing Robot with Line Following Capability

Nor Maniha Abdul Ghani, Faradila Naim
Faculty of Electrical & Electronics Enginnering
Universiti Malaysia Pahang
normaniha@ump.edu.my,faradilan@ump.edu.my

Tan Piow Yon
Faculty of Electrical & Electronics Enginnering
Universiti Malaysia Pahang
pokemon_tpy@hotmail.com

Abstract— **This project focuses on the development of a line follower algorithm for a Two Wheels Balancing Robot. In this project, ATMEGA32 is chosen as the brain board controller to react towards the data received from Balance Processor Chip on the balance board which monitoring the changing of the environment through two infra-red distance sensor to solve the inclination angle problem. Hence, the system will immediately restore to the set point (balance position) through the implementation of internal PID algorithms at the balance board. Application of infra-red light sensors with the PID control is vital, in order to develop a smooth line follower robot. As a result of combination between line follower program and internal self balancing algorithms, we able to develop a dynamically stabilized balancing robot with line follower function.**

Keywords— **infra-red sensor, PID algorithms, line follower Balancing robot.**

I. INTRODUCTION

Over the past decades, the research on two wheeled inverted pendulum mobile robot or commonly known as balancing robot have gain momentum in a number of robotic centers around the world due to natural unstable dynamics of the system. [1] Since a Two wheeled balancing robot need a good controller to maintain itself in upright position without the needs from external forces. Thus, providing a good platform for researcher to explore the efficiency of various controllers in control system based on the inverted pendulum model. Nowadays, various controllers are implemented on two wheeled balancing robot for example s Linear Quadratic Regulator, Pole-Placement Controller, Fuzzy Logic Controller, and Proportional Integrated Derivative Controller. [3]

A two wheeled balancing robot is categorized by the ability to balance on its two wheels and spin on the spot. As a result from this additional manoeuvrability allows easy navigation on the various terrains, turn sharp corner, traverse small step or curbs and ability on carry load. Two wheeled robots also have a small footprint than three or four wheeled robots thus enable it to travel around corridors and tight corners more easily. [4] These capabilities have the potential to solve many challenges in industry and society. As the two wheeled balancing robot has been investigated and developed to become human transport machine. The Segway, Pegasus, and iBot models are the example of the design of two wheeled balancing robot as a human transport machine. In addition, a motorised wheelchair utilising this technology would give the operator greater manoeuvrability and thus access to places most able-bodied people take for granted. Small carts built utilising this technology allows humans to travel short distances in a small area or factories as opposed to using cars or buggies which is more polluting. [2]

In this project, the two wheels balancing robot not only solve the balancing problem but also can automatically move around and avoid basic obstacles through the implementation of Dual ground Sensor and looking forward infrared sensor respectively. The Dual ground sensor is located at the bottom of the Balbot which is the infrared IR distance sensor, where ground sensors are used to measure the tilting angle. Powerful ATMEGA32 processor is used to be the brain of the robot. BEMP motor velocity sensor is used to obtain the speed of the platform. The entire controller algorithm will be compute into C programming and store inside the microcontroller. Without an active control system, the robot would just fall over. Thus, the controller plays an important role in this project. Lastly, three pairs of infra-red sensor are used to guide the line follower task during balance state.

II. LITERATURE REVIEW

Conducting literature review prior to begin a research project is vital in understanding two wheels balancing robot control technique, as this will supply the researcher with much needed additional information on the methodologies and technologies available and used by other research counterparts around the world. This chapter provides a condensed summary of literature reviews on key topics related to balancing a two-wheeled robot. [2] Comparisons between the present project and the related topics of existing information will also be discussed.

A. Previous Project Work

A research professor at Carnegie Mellon University, Ralph Hollis who has developed a totally unique balancing robot that balances on top of a bowling ball. He calls his robot design "BallBot" [6, p. 72]. Mr. Hollis and his research associates believe that robots in the future will play a vital role in the daily lives of humans. He believes that in order for robots to be productive in our daily lives, some key problems need to be solved first. One the important problem he states in his article about mobile self balancing robots is the overall structure of the robot itself. (R. Hollis 2009)

A researcher, Steve Hassenplug has successfully constructed a balancing robot called Legway using the LEGO Mindstorms robotics kit. Two Electro-Optical Proximity Detector sensors from HiTechnic Sensors to provide the tilt angle information and detect lines. The controller is programmed in high level programming language specifically created for LEGO Mindstorms which was written in brickOS (LegOS) and uses EOPDs to maintain a constant distance from the ground. As the distance decreases, Legway moves forward. As the distance increases, Legway moves backward. Every 50 ms, Legway attempts to recalculate the balance point by measuring the current distance and motor speed. To move forward for line following, Legway actually sets the motors to run backward, causing a tilt, which it automatically corrects by moving forward. When one sensor is over the line, it stops that motor, and Legway balances using only the other motor, causing it to turn. To spin in place, both motors are shifted "off center" in opposite directions, the same amount, but they still correct for tilting. Legway uses its two optical proximity detectors to balance the two wheel LEGO robot. (Hassenplug S. 2003)

nBot is a two-wheeled balancing robot built by David P. Anderson. This robot uses commercially available of the shelf inertial sensors and motor encoders to balance the system. Such inertial sensors that are used on nBot are an accelerometer and a gyroscope. The basic idea for a two-wheeled dynamically balancing robot is pretty simple which is driving the wheels in the direction that the upper part of the robot is falling. If the wheels can be driven in such a way as to stay under the robot's center of gravity, the robot remains balanced. In practice this requires two feedback sensors: a tilt or angle sensor to measure the tilt of the robot with respect to gravity, and wheel encoders to measure the position of the base of the robot. Four terms are sufficient to define the motion and position of this "inverted pendulum" and thereby balance the robot. (Anderson D.P. 2003)

As invented by Dean Kamen, the Segway PT's dynamics are identical to the inverted pendulum. For added mobility, the Segway is also based on the two wheel platform design. The advanced control algorithms behind the Segway transporter are a company trade secret. The basics of a Segway are computers that process the control algorithms, two tilt sensors, five gyroscopes, and two electric motors. Only three of the five gyroscopes are used to balance the Segway. The remaining two gyroscopes are used as backup. These critical components that make up a Segway are important to keep the vehicle in perfect balance. Current models of the Segway personal transporter can achieve top speeds of 12.5 mph. The Segway is able to navigate thru rough terrain, while successfully carrying a human onto of the platform. The Segway is typically found in urban settings; used for guided tours and city government officials. (Dean Kamen 2001)

A researcher, Priyank Patil has successfully design and constructed a Line Follower Robot which wills sense a line and manoeuvring the robot to stay on course, while constantly correcting wrong moves using feedback mechanism forms a simple yet effective closed loop system. This robot is running on Atmel's AVR® microcontrollers which have a RISC core running single cycle instructions and a well-defined I/O structure that limits the need for external components. Internal oscillators, timers, UART, SPI, pull-up resistors, pulse width modulation, ADC, analog comparator and watch-dog timers are some of the features you will find in AVR devices. AVR instructions are tuned to decrease the size of the program whether the code is written in C or Assembly. With on-chip in-system programmable Flash and EEPROM, the AVR is a perfect choice in order to optimize cost and get product to the market quickly. [8]

B. PID Control Systems

The implementation and complexity difficulty associated with the nonlinear method causes most control researchers to utilize the linear controller approach. The method that will be used to control the self-balancing two wheeled robot will be a linear controller. It will be applied through a Proportional, Integral, and Derivative also refer to as the PID. The PID has proven to be popular among the control engineering community. As stated by the author of article Vance J. Van Doren, "For more than 60 years after the introduction of Proportional-Integral-Derivative controllers, remain the workhorse of industrial process control" [9].

III. METHODOLOGY

The main purpose of this project is to upgrade a balancing robot to become a line follower Balbot. It involves the understanding of balancing system in the Balance Board which using PID controller as linear controller with Analog GP2D120 IR distance sensors as feedback input and the concept of line following system with the implementation of IR sensors as shown in Fig. 1 about the General System Block Diagram. The line following mode can be establish based on the algorithm programmed in the Brain Board once the robot achieved balance state or condition as described in Fig. 2 about the Robot's Internal System Flow.

This project is dividing into two parts namely hardware and software implementation. Hardware part involves more in interfacing between Brain Board and Balance Board with Analog GP2D120 IR distance sensors and Infra Red Sensors and also brief introduction on the robot's components. While in the software part include programming software using Brain Board Code Editor and procedure in loading programs into the microcontroller on the Brain Board. Fig. 3 illustrated the front view of the robot.

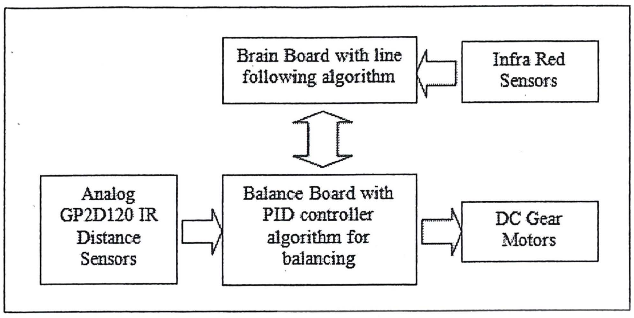


Figure 1: Robot's General System Block Diagram

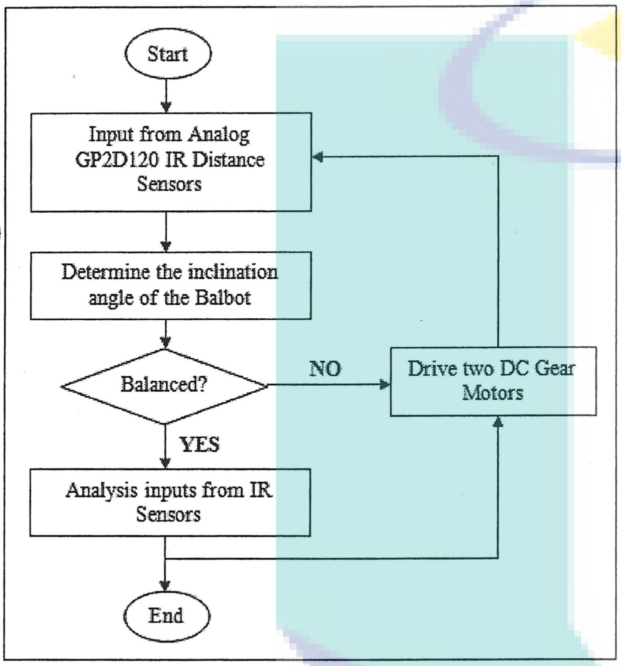


Figure 2: Robot's Internal System Flow

A. Balancing System

The Balbot balances by using the infrared range sensors ned at the ground to calculate the absolute angle and the angular velocity of the robot. The primary function of the Balance Processor Chip (BPC) is to perform real time balancing control by using data from the GP2D120 IR Distance Sensors to drive the two motors in a manner that will keep the center of gravity above the wheels at all times. Thus, frees the master processor in the Brain Board to perform other tasks. The PID controller in balancing system is a linear controller that controls overall performance of robot. DC Gear motors are actuators for movement. The Balance Board executes the standard Digital Cascaded Control Loop (built into Balance Processor Chip) algorithm to balance the robot. The control loop runs at a frequency of approximately 40.2Hz.

The actual velocity of robot will be detected by BPC using Back-Electromotive Force (BEMP). Then, BPC will calculate the error between the desired velocities with the actual velocity by considering also the tilt angle as the feedback from GP2D120 IR Distance Sensors, thus the robot will moving toward a falling direction (left side) at a specific angular velocity calculated by the system corresponding with the tilt angle in order to maintained the Balbot in balance condition as shown in Fig. 3. [10]

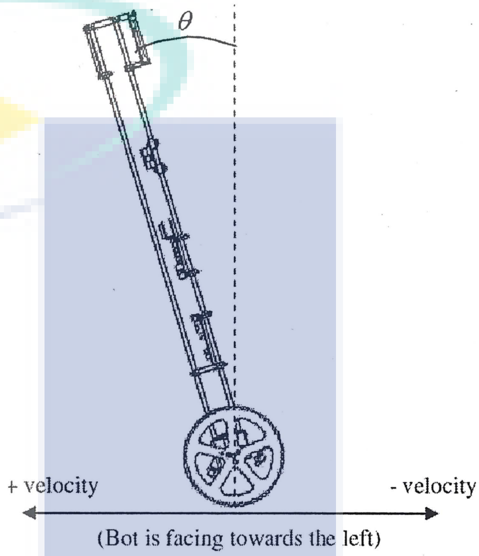


Figure 3: The movement of Balbot due to the tilt angle with specific velocity

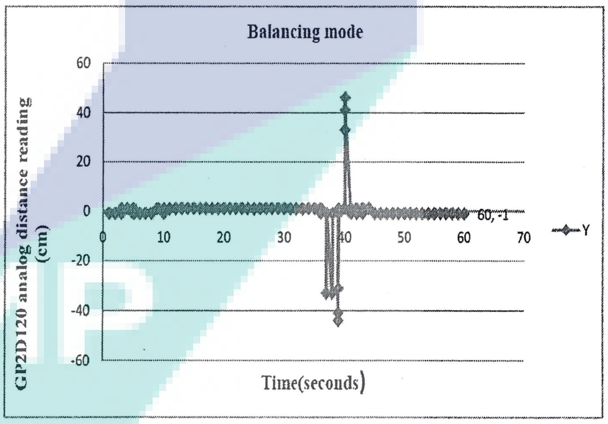


Figure 4: GP2D120 Distance Reading for Balancing Mode

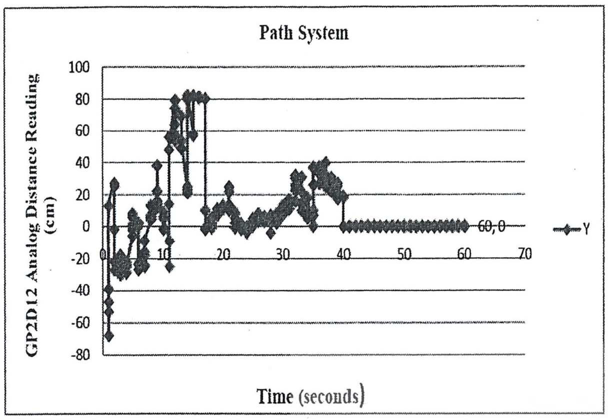


Figure 5:GP2D120 Distance Reading for Path Navigation

B. *Line Following System*

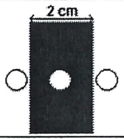
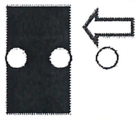
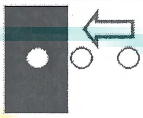
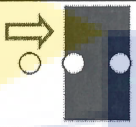
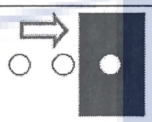
Line following task is performed through the implementation of specified algorithm with the data from the Infra-Red sensors which are mounted at the bottom of robot. The data or status receive from the sensor will be processed by the programmed microcontroller to perform the continuous movement according to the line. The direction and speed of rotation is control by the microcontroller through PWM manipulation.

The robot will conduct line following function when a balancing state is achieved. If the robot is in unstable condition, balancing task will be performed first, then follow by the line follower. Table 1 show that the designed line follower control algorithm for robot. There have five states for line following control in robot which are five possible condition might be occurs during line follower system execution in robot. All of the five states are expressed in five desired output direction in Balbot which are forward, shift to left, fast shift to left, shift to right, and fast shift to right.

The inputs in this line follower are IR sensors which mounted at the bottom of robot. The logic "1" indicates no line detect while logic "0" indicates line detected. The algorithm can be adapted for situation in which the robot is continuously moving forwards. The line follower algorithm also can be expressed in flowchart for the decision of robot as shown in Fig 6. It is showed that the robot in this line follower algorithm will check on the state of the sensors first and based on the information from the three IR sensors, the desired motion of the robot will be determined.

The programming that loaded in the ATMEGA32 is to defined function of a line follower control algorithm and decision of robot movement. First of all, the input ports for the sensors are been defined and initialized in Brain Board Code Editor Software.

Table 1: Line Follower Control Algorithm

Status	Sensor Input			Output
	L	M	R	
	0	1	0	FORWARD
	1	1	0	SHIFT TO LEFT
	1	0	0	FAST SHIFT TO LEFT
	0	1	1	SHIFT TO RIGHT
	0	0	1	FAST SHIFT TO RIGHT

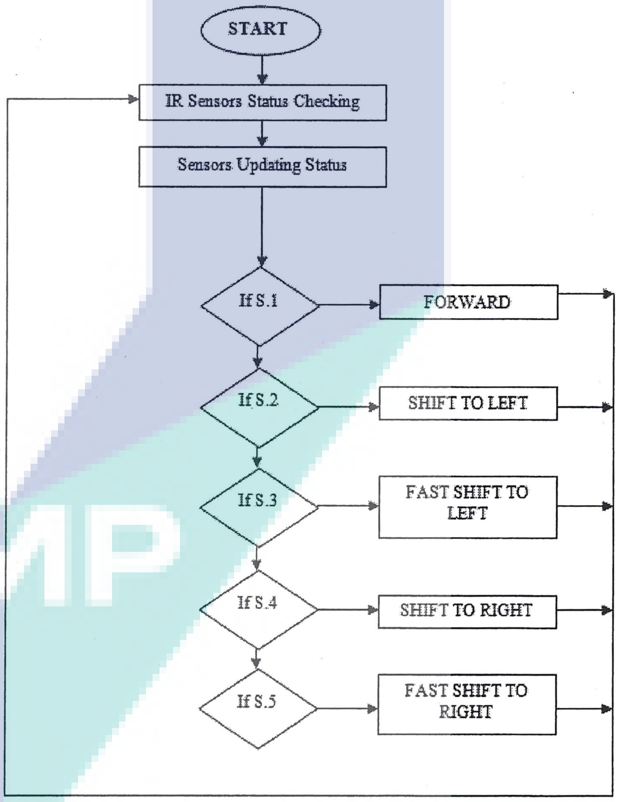


Figure 6: Flow Chart for the Line Following System

Then all the input ports are enable pull-ups. There are two variables that will change the desired path and speed of robot which are tmp_fwd and tmp_steer. For the tmp_forward is function to move robot forward or reverse with specific speed, while tmp_steer function to determine the steering direction of robot to turn left or right at desired speed. The positive number in tmp_fwd and tmp_steer indicate that robot will move forward and turn right respectively and vice versa. The picture of the IR sensors and Controller Circuit Board are shown in Fig.7.

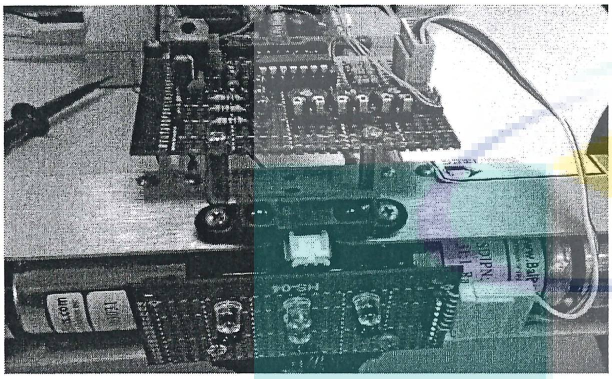


Figure 7: IR Sensors and Controller Circuit Board

IV. RESULT AND DISCUSSION

In this section, result from the balancing and the line follower control algorithm obtained and observed from the robot has been described. The data were collected and recorded through the observation. The discussion section will highlight about initial calibration, efficiency in balancing mode, and the performance of line follower control system hardware and also algorithm.

The analysis phase provides an opportunity to assess and evaluate the robot effectiveness and efficiency in maintaining stability and executing line following task. This allows a comparison to be undertaken between the actual system performance and the anticipated project objectives. The opportunity to calibrate and perform additional fine tuning of the design is also explored with the aim of achieving the best result possible in mind.

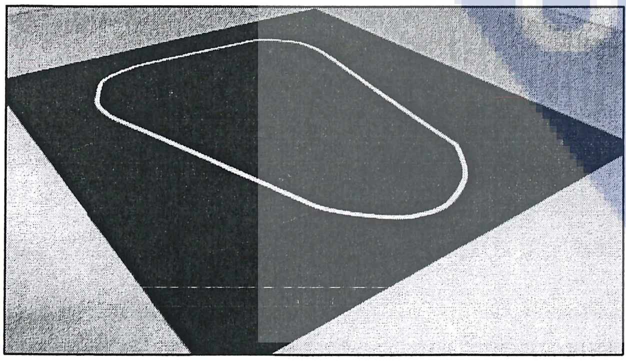


Figure 7: Line Follower Track

V. CONCLUSION

This project as a whole was very successful in driving a line follower two wheeled balancing robot with all the considerations necessary in ensuring it could meet the required capabilities and objectives. The software and programming parts is achieving satisfaction level. For future work and suggested areas of further investigation, it is recommended that using a gyrosensor instead of IR sensor to make the robot balance on all types of surface including uneven surface. Thus, it will be more robust in terms of controllability and functionality.

REFERENCES

[1] S.W. Nawawi, M.N. Ahmad, J.H.S Osman (2008). “Real-Time Control of a Two Wheeled Inverted Pendulum Mobile Robot”, Proceedings of World Academy of Science, Engineering and Technology Volume 29 May 2008.

[2] Ooi, Rich Chi (2003). “Balancing a Two-Wheeled Autonomous Robot”. University of Western Australia. Thesis B. Mechatronics Engineering.

[3] Ong, Yin Chee; Abidin, M.S.B (2006). “Design and Development of Two Wheeled Autonomous Balancing Robot”, Center for Artificial Intell. & Robotic (CAIRO), University Technology Malaysia, Kuala Lumpur. 4th Student Conference on Research and Development (SCOReD 2006), Shah Alam, Selangor, MALAYSIA, 27-28 June, 2006.

[4] Vivien Coelho; Stanley Liew; Karl Stol; Guangyu Liu (2008). “Development of a Mobile Two-Wheel Balancing Platform for Autonomous Applications”. Department of Mechanical Engineering, the University of Auckland. 15th International conference on Mechatronics and Machine Vision in Practice (M2VIP08), 2-4 Dec 2008, Auckland, New-Zealand.

[5] José Miranda (2009), “Application of Kalman Filtering and PID Control for Direct Inverted Pendulum Control”. Master Thesis, California State University, Chico Spring 2009.

[6] R. Hollis, “BallBots,” Scientific American, October 2006. Retrieved February 4, 2009. Available at the World Wide Web: http://www.sciam.com/article.cfm?id=ballbots

[7] Pakdaman, M.; Sanaatiyan, M.M. (2009), "Design and Implementation of Line Follower Robot," Computer and Electrical Engineering, 2009. ICCEE '09. Second International Conference on , vol.2, no., pp.585-590, 28-30 Dec. 2009

[8] Priyank Patil (2010), “AVR Line Following Robot,” Department of Information Technology K. J. Somaiya College of Engineering Mumbai, India. Retrieved Mar 5, 2010. Available at: http://www.kmitl.ac.th/~kswichit/ROBOT/Follower.pdf

[9] Swope Design. Inc., “Balancing Robots Made Easy”, Available at: http:// www.Balbots.com, 2004.

[10] Miller Peter (2008), “Building a Two Wheeled Balancing Robot”, University of Southern Queensland, Faculty of Engineering and Surveying. Retrieved Nov 18, 2009. Available at: http://eprints.usq.edu.au/6168/