CHAPTER 1

INTRODUCTION

1.1 Background of Study

Demand of conventional vehicle such as long distance transportation bus, medium and large lorry carrier, and trucks are really high. The current system is using human as this conventional vehicle’s driver for the whole journey. Yet human has its own limitation when they need essentials rest during this long period of journey. The break time those taken by the drivers are actually a waste of time in term of efficiency of time consumption and delivery department. Productivity efficiency can be slightly increased if this rest time can be converted into a gain of distance if maneuvering on highway and rest time can be done simultaneously. Therefore the project ‘The Design of an AutoGuide System prototyped for Conventional Vehicle on Highway’ is to manipulate this rest time to a control system that are reliable for the task of maneuvering the vehicle with less need of monitoring by human. This system will able to improve the transportation efficiency and time management effectively. A sensor will be attached to a side of the vehicle and monitor its distance with the highway divider. This sensor then will communicate with the actuator to control the direction of the vehicle. Another sensor will also attach to the in-front of the vehicle. This is to monitor the distance of the car with the car in-front, and communicate with the actuator to initiate braking system. Upon switch to AutoGuide mode, the system will set the vehicle speed to a safe speed and maintain this speed along the time period. At the end of project, to prove its reliability, the similar system will be shown by create a prototyped that can show how this system work.

1.2 Problem Statement

Table 1 below shows the statistic of accident from year 1995-2007 in Malaysia

YEAR	TOTAL CASE
1995	6,802,375
1996	7,686,684
1997	8,550,469
1998	9,141,357
1999	9,929,951
2000	10,598,804
2001	11,302,545
2002	12,018,291
2003	12,819,248
2004	13,764,837
2005	14,816,407
2006	15,790,732
2007	16,812,440

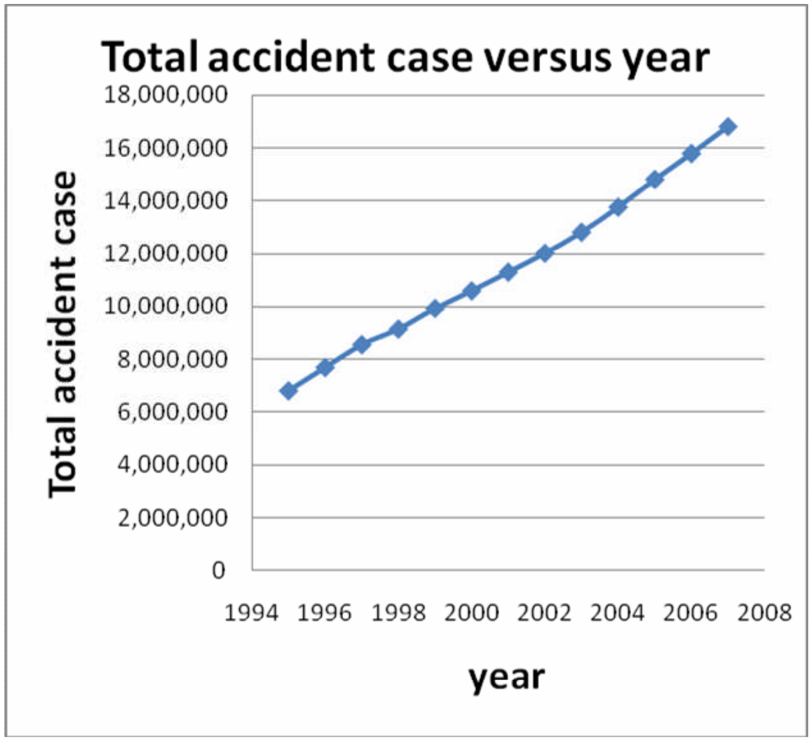


Table 1 shows the total accident case from year 1995 – 2007 in Malaysia. When convert this value to graph statistic total accident case versus year, its show the increasing accident cases for every year. After done some research, the causes of this accident are such as because of human attitude, condition of car or road, and the driver sleep when they drive, not concentrate when drive or feel tired because of long journey. Accident problem is a one of the big issue in this country. Government always concern about this increasing value of accident. They always organize campaign, talk and other thing to make sure the people more concern about this problem. Actually, this project is one of the ways to make the total of accident cases is decreasing for the coming years.

The other thing, conventional vehicle are requirement for most Manufacturing and Business Company. Hence a deficiency in this department of transportation means a loss in time management and reliability to deliver goods on time. In addition to this deficiency is the accident rate by this type of transportation method which increased by time. Not only lead to properties loss but also increase on overhead capital. All this deficiency is because of human errors that have limitations and imperfectness in doing their job. Normally, for the long journey the driver must take a rest. But this will waste the time.

So, the system that implemented here is maneuvering the vehicle without the driver. Actually, the system control the speed and steering of the vehicle when the driver takes a rest.

1.3 Objective

- To build a miniature prototyped of an AutoGuide System based on previous design
- To implement some modifications to improve the system

1.4 Scope of Study

- Development of control system
- Selection of suitable controller, sensor etc.
- To build the circuit of autoguide system
- Programming of the PIC by using the C-Compiler
- Prove the autoguide system with testing the prototype

CHAPTER 2

LITERATURE REVIEW/THEORY

2.1Literature Review

Conventional vehicle are basically that used for business purpose. The examples of conventional vehicles are bus, lorry, trailer and good truck. The usage of conventional vehicles to transfer items or service is because of the quantities that are not small for regular and not too large for massive transportation vehicle. Most conventional vehicles are expected to make a long journey and currently it is manually operated by humans. Since human are not perfect, no matter how good is the conventional vehicles are deficiency and error is not negligible. Accident and delay are most common problem need to be concern by the company.

So, the purpose of build this autoguide system is to decrease the probability of accident because of human error. This system is to control the steering system (direction), speed and breaking system of the vehicle.

2.1.1 Measurement of range

2.1.1.1 Reversing Radar Based on CAN Bus [1]

The objective to develop a sort of vehicle reverse sensor alarming system based on CAN bus is to reduce the traffic accidents when backing or parking. The reverse sensor that used is ultrasonic sensor to measure the distance between the rear of the vehicle and the obstacle. The control system takes the microprocessor as the control core to control the running of the whole system and control various interface circuits. It transmits pulses and detecting the echoes through controlling the multiple selection switches, implements data processing, sample the time difference, measure the time difference from transmitting the ultrasonic signal to accepting ultrasonic signal, and measure the distance. The display alarm system displays the minimum distance and alarm to remind the driver. The hardware design includes designs of the range acquisition system, the single chip microcomputer controller and the display alarm system.

The type of detector used in this application is UCM-T40KI . Each one sensor was installed at two ends of the front bumper to detect the front obstacles, the other four sensors are installed at the rear bumper. These six detectors adopt the scanning method and share one signal processing circuit. The ultrasonic is generated by the software generation method. The software design adopts C programming.

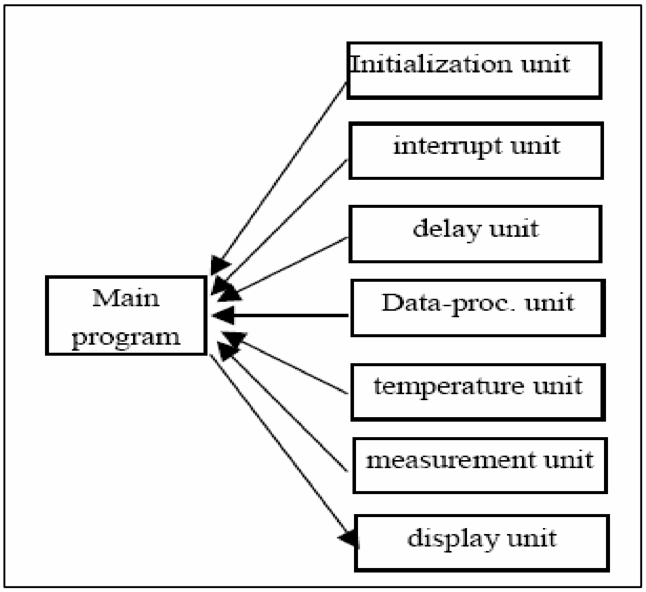


Figure 1: The software structure

2.1.1.2 Driver Assistance System (DAS) [2]

This system is to detect obstacles and warns the driver in advance of possible collision in such a congested traffic environment. The term DAS mean an autobrake system, a collision-warning system, a parking-assistance system.

Typical sensors used in DASs are laser sensor, millimeter-wave radar, charge-couple device (CCD) camera, ultrasonic sensor. Laser sensors and millimeter-wave radar are preferred for adaptive cruise control, and they are expensive compared with other sensor. CCD cameras are significantly affected by bad weather condition (snow, rain, dirt and dust). Ultrasonic sensors are less affected by adverse weather conditions and are more economical compared with all other sensor. So, ultrasonic emerge as one of the strong candidates for use in a DAS.

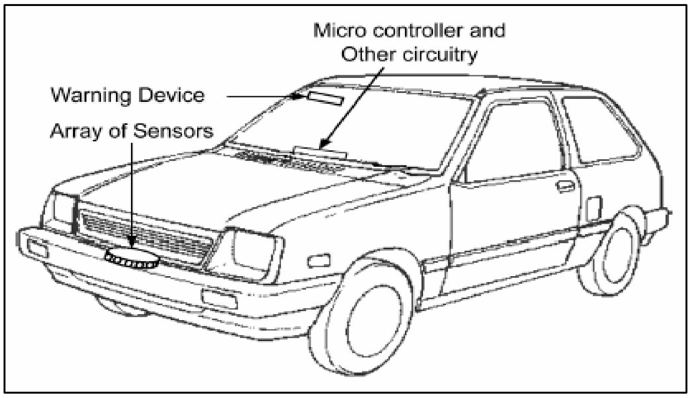


Figure 2: position of sensor and other
electronic on vehicle.

2.1.2.3 Object Tracking and QOS control Using Infrared Sensor and Video Cameras [3]

The object tracking scheme using infrared sensors to detect and locate moving objects, while video cameras are for tracking, producing trajectories and if possible, providing identity and details.

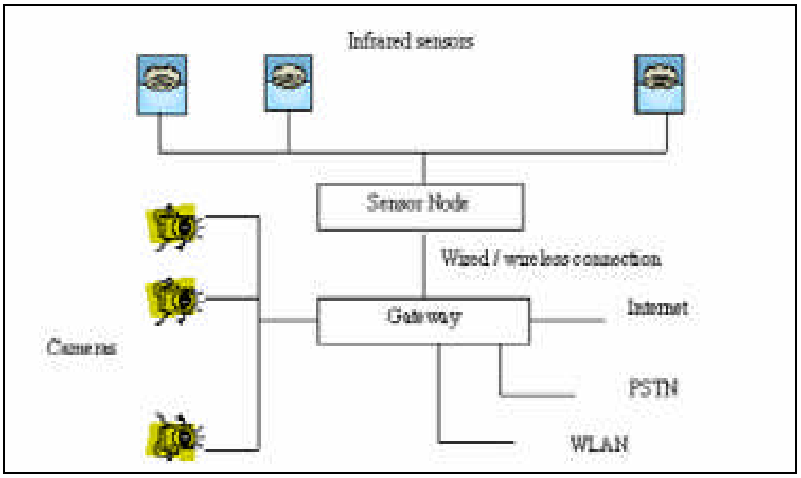


Figure 3: infrared sensor nodes, video cameras and gateways in a
sensor network tracking system.

When the object is detect using infrared sensor, its location and movement is calculated. After receiving a trigger signal, the gateway will select a camera, isu pan and zooming command, and then the video tracking of the detected object is launched. Because there are multiple, infrared sensors and video cameras are attached to a gateway, multiple object can be detected and tracked in real time.

2.1.2.4 An Autonomous Wall Following Robot[8]

This robot used a PIC 18F4520 as microcontroller as its brain. PIC 18F4520 receive input from Ultrasonic Distance Meter. Some computations are performed on this input and a control signal is generated to control the robot's position. This control signal is generated through a PD controller, which is implemented in the microcontroller. First, simulations were performed in Matlab to design a PD controller and to understand the overall picture of the robot's behavior. Afterwards, microcontroller was programmed in C language using a CCS compiler.

2.1.2Constant speed

2.1.3.1 Autonomous Intelligent Cruise Control [4]

Present generation cruise control systems only work well on clear roads and have to be disengaged when it is necessary to change speed because of other traffic. By introducing a sensing system which can determine the distance and relative speeds between a vehicle and the one in front it is possible to develop a cruise control system which will automatically maintain a safe distance between them. This system can give automated stop start driving in congested urban area, allowing the driver more time to concentrate on other tasks such as navigating through unfamiliar complex junctions and/or selecting the most appropriate lane. These have enormous potential to reduce the number of accidents currently occurring on Europe's road. It is possible to use the Autonomous Intelligent Cruise Control system to reduce the risk of running into the rear of other vehicles by automatically applying the brakes. Example, on the approaches to roundabouts. However it should only alert the driver in high speed situations because the correct avoiding action could be rapid acceleration or lateral deviation. If the brakes were automatically applied it could adversely affect the probability of safety completing such a maneuver.

In order to achieve the degree of longitudinal control required in an Autonomous Intelligent Cruise Control system it is necessary to enable the cruise controller to control the brakes and engine. For safety reason, the driver can always over-ride the cruise controller.

Sensing system is required in sensing the speed and distance to the vehicle being followed. It is also to determine various properties of the road between the two vehicles including the available adhesion and the path of the lane in which the vehicles are travelling. If the system is to avoid locking onto vehicles in other lane, roadside furniture etc, then it is necessary for the sensor system to be able recognize and follow not only the path of the road, but also the path of the relevant lane on the road. If have any vehicle move slow, then the cruise control equipped vehicle will also slow to a safe speed.

2.1.3.2 Intelligent Cruise Control and Roadside Information (AICC) [5]

Autonomous intelligent cruise control system controls a vehicle's speed according to the driver's desire and the speed of and distance to the preceding vehicle. This system offers a one directional short range system for vehicle-vehicle and roadside vehicle communication and considerations for recommended speed, limits, and traffic signals.

The Volvo's AICC system that developed and designed drivers in adapting their speeds with regard to

- The desired cruise speed
- The distance to and the velocity of the preceding vehicle
- Speed recommendations and limits
- Traffic signals and Green Wave system

The autonomous operation of the AICC system uses a target sensor made by Leica and consisting of five fixed, nonoverlapping infrared beams. Each beam has a range of 150 meters and the angular coverage 1.5 degrees, horizontally and vertically.

2.2Hardware

2.2.1 Sensor

Sensor is any of various devices designed to detect, measure, or record physical phenomena, as radiation, heat, or blood pressure, and to respond, as by transmitting information, initiating changes, or operating controls. In this project, the choosing of sensor is to determine the best method of guiding the system at a safe distance.

2.2.1.1 Ultrasonic Sensor [6]

Ultrasonic sensors are known for their robust performance in harsh and problematic environments, where there are a variety of reflective forms and where precise detection is essential. Additionally, they are unaffected by target color, ambient noise, or dusty atmospheric conditions; they provide noncontact distance measuring; and their longer and wider sensing ranges help solve some pretty tough applications. Ultrasonic sensors bridge the gap between proximity and photoelectric sensing. They are ideal for applications that require a longer sensing distance than inductive or capacitive proximity detection can provide, that are too dirty for photoelectric sensors, or that involve unusually shaped targets. They are especially effective in detecting and monitoring objects with a relatively high density and high acoustic reflectivity, such as solids, liquids, and granular materials.

Ultrasonic sensors operate by using sound waves to detect targets (see Figure 4). A sensor generates a short, very intense sound burst from a piezoelectric transducer, which is reflected back by the object. The sensor determines the distance to an object by measuring the time that elapses between the emission of an ultrasonic burst and the arrival of the echo reflected by the target. This sensing method ensures reliable operation regardless of the object's color or opacity.

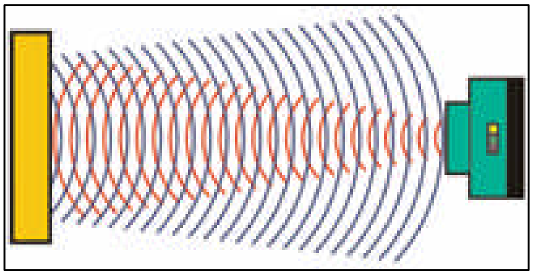


Figure 4: Ultrasonics determine distance with sound waves that are reflected back by the object being detected.

2.2.1.2 Infrared Sensor[14]

Infrared refers to that part of the electromagnetic spectrum between the visible and microwave regions. Electromagnetic spectrum refers to the seemingly diverse collection of radiant energy, from cosmic rays to X-rays to visible light to microwaves, each of which can be considered as a wave or particle traveling at the speed of light.

In measuring distance the device will emit an infra-red light periodically and this light will be bounce by the object to Charge-Couple Device (CCD) which included with the infrared sensor. CCD array will determine the angle of bounce and calculate the angle, thus this will determine the object distance from the sensor.

2.2.1.3 Infra-Red Reflectance Sensor[14]

This sensor contains a matched infrared transmitter and infrared receiver pair. These devices work by measuring the amount of light that is reflected into the receiver. Because the receiver also responds to ambient light, the device works best when well shielded from ambient light, and when the distance between the sensor and the reflective surface is small (less than 5mm). IR reflectance sensors are often used to detect white and black surfaces. White surfaces generally reflect well, while black surfaces reflect poorly. By calculating the voltage difference between the white region and black region hence the sensor is capable to constantly set to choose either to follow which region. Limitation of the capability to monitor it at one direction made this sensor should be installed more than a pair.

2.2.1.4 Laser Range Sensor [7]

This sensor is a phase-shift measurement device that compares the outgoing and returning wave signals to determine the distance to a target. The phase shift between the scent and the received modulated frequency waves are measured and the distance is calculated. This shift is adjusted based on ambient lighting conditions and temperature.

This non-contact measurement sensor reaches distances up to 500 m on reflective targets with an accuracy of ± 3mm.

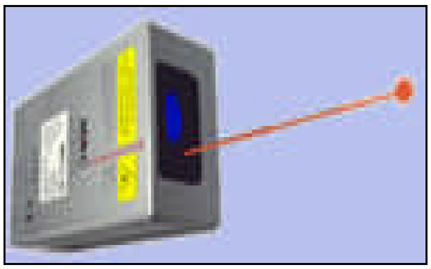


Figure 5: example of laser range sensor

2.2.1.5 Sensor Consideration

Determining of the sensor that will be choose for the guiding system is a vital part where the sensor must give the least limitation due to the condition of maneuvering on the roads. Among the condition this sensor must pass is the weather condition, imperfectness of the roads, unexpected foreign object in between the safe distance, and junction along the highway. Therefore the investigation of sensor should be conducted to determine the most suitable sensor in leading the Autoguide system.

2.2.2 Peripheral Interface Controller (PIC) [8]

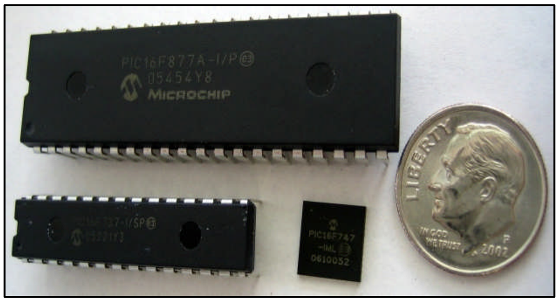


Figure 6: example of PIC

PIC (Peripheral Interface Controller) is the IC which was developed to control peripheral devices, alleviating the load from the main CPU. Compared to a human being, the brain is the main CPU and the PIC is equivalent to the autonomic nervous system. The PIC, like the CPU, has calculation functions and memory, and is controlled by the software. However, the throughput and the memory capacity are low.

The main function of PIC for this project is to control the movement of tire (steering system) and speed of vehicle after received signal from sensor. To make this PIC work as planned, programming software will be use to program it.

2.2.3 USB Programmer [9]

Figure 7: example of USB programmer

USB (universal serial bus) programmer used to download hex file into chip. Its mean that if the coding of chip in C-language, so used USB programmer to compile it become hex. Actually, hex file is a machine code.

2.3 Software

2.3.1 MPLab C Compiler for PIC18 MCUs [8]

MPLAB C Compiler is a cross-compiler that runs on a PC and produces code that can be executed by the Microchip PIC18XXXX family of microcontrollers. Like an assembler, the compiler translates human-understandable statements into ones and zeros for the microcontroller to execute. Unlike an assembler, the compiler does not do a one-to-one translation of machine mnemonics into machine code.

Code is written using standard ANSI C notation. Source text is compiled into blocks of program code and data which are then “linked” with other blocks of code and data, then placed into the various memory regions of the PIC18XXXX microcontroller. This process is called a “build,” and it is often executed many times in program development as code is written, tested and debugged. This process can be made more intelligent by using a “make” facility, which invokes the compiler only for those C source files that have changed since the last build, resulting in a faster build times.

2.3.2 Proteus VSM [11]

Proteus Virtual System Modelling (VSM) combines mixed mode SPICE circuit simulation, animated components and microprocessor models to facilitate co-simulation of complete microcontroller based designs. For the first time ever, it is possible to develop and test such designs before a physical prototype is constructed.

This is possible because you can interact with the design using on screen indicators such as LED and LCD displays and actuators such as switches and buttons. The simulation takes place in real time (or near enough to it): a 1GMHz Pentium III can simulate a basic 8051 system clocking at over 12MHz. Proteus VSM also provides extensive debugging facilities including breakpoints, single stepping and variable display for both assembly code and high level language source.

The Proteus Design Suite is wholly unique in offering the ability to co-simulate both high and low-level micro-controller code in the context of a mixed-mode SPICE circuit simulation. With this Virtual System Modelling facility, you can transform your product design cycle, reaping huge rewards in terms of reduced time to market and lower costs of development.

If one person designs both the hardware and the software then that person benefits as the hardware design may be changed just as easily as the software design. In larger organisations where the two roles are seperated, the software designers can begin work as soon as the schematic is completed; there is no need for them to wait until a physical prototype exists.

In short, Proteus VSM improves efficiency, quality and flexibility throughout the design process.

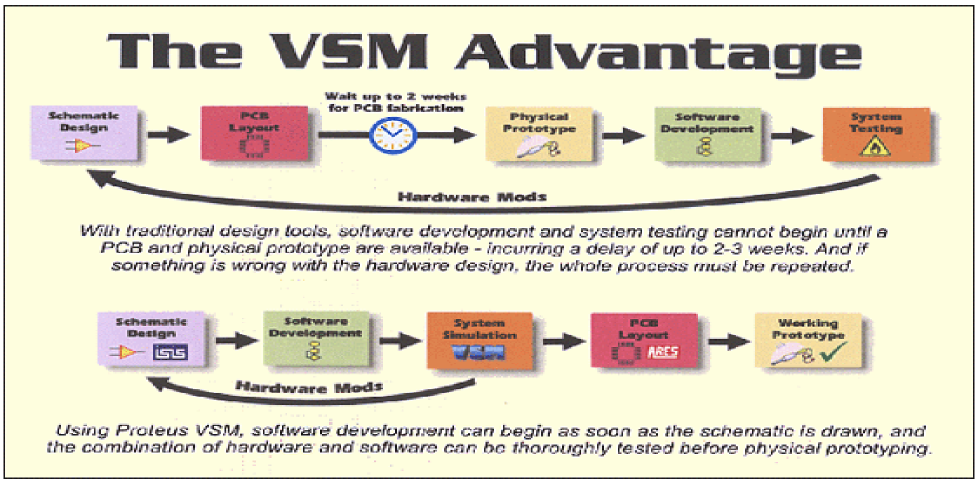


Figure 8: flow system of Proteus VSM software

2.3.3 KTechLab [12]

KTechLab is an Open Source Intergated Design Environment (IDE) for electronic and PIC microcontroller circuit design and simulation.Featuring an extensive circuit designer with autorouting and simulation of many common electronic components and logic elements, KTechLab is the ideal tool for educational or hobbyist use.

KTechLab features an easy to use, flowchart based PIC program designer – Flowcoder along with a BASIC like programming language called Microbe. Programs designed by these can be added to any circuit design as a 'virtual' PIC allowing complex microcontroller based circuits to be created. KTechLab supports a wide range of Open Source PIC programmers, enabling finalised PIC programs to be quickly and easily transfered to a real PIC microcontroller.

2.4 Existing similar System

2.4.1 Automated highway system [13]

An automated highway system (AHS) or Smart Road is a proposed intelligent transportation system technology designed to provide for driverless cars on specific rights-of-way. In one scheme, the roadway has magnetized stainless-steel spikes driven one meter apart in its center. The car senses the spikes to measure its speed and locate the center of the lane. Furthermore, the spikes can have either magnetic north or magnetic south facing up. The roadway thus has small amounts of digital data describing interchanges, recommended speeds

2.4.2 Single track vehicle [8]

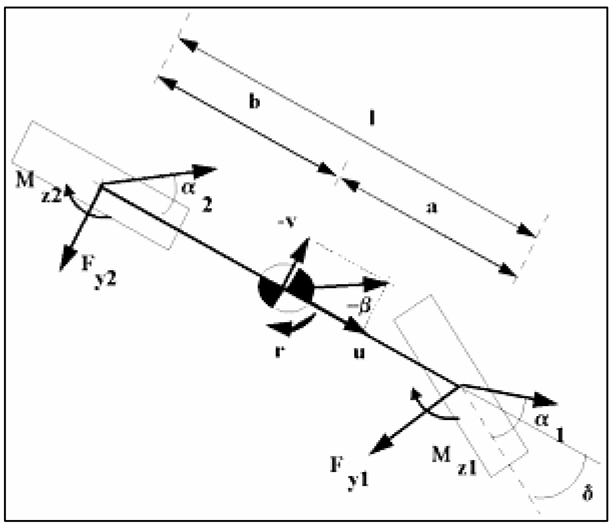


Figure 9: Simple one track vehicle

The steer and slip angles are restricted to relatively small values. The driving force required to keep the speed constant is assumed to remain small with respect to the lateral forces acting on the tire. Brakeforces will be neglected as well for the time being.

CHAPTER 3

FINDING

3.1 Autoguide system

3.1.1 Safe Distance [14]

To program the range sensor, the distance between vehicle and highway divider must fix it. Save distance must set because, vehicle will move constantly along the highway with highway divider as point reference.

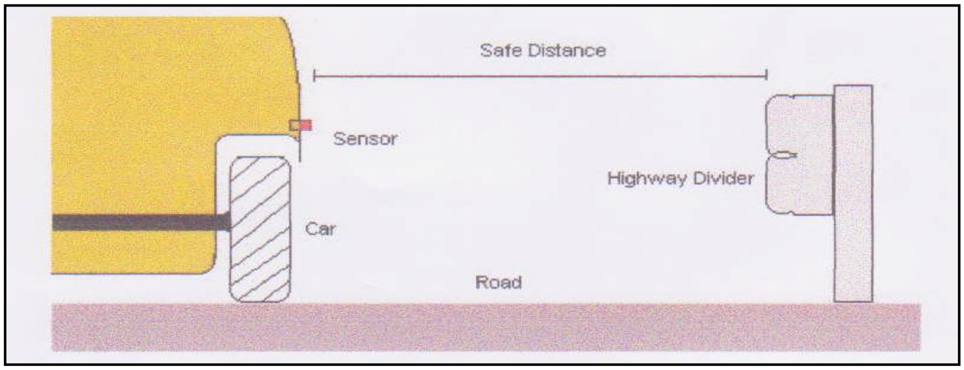


Figure 10: The save distance

The consideration to choose the save distance are:

(a) The vehicle must use the most left lane.

(b) The vehicle not on the emergency lane

(c) Safe distance must allow enough space for motorcycle.

So, after make this consideration, the safe distance suggested are 2.0 meter (200cm).

3.1.2 Installation of sensor

The sensor is planned to be install in between of 50cm to 100cm above the road at the side and front of vehicle. The side sensor will detect the highway divider and then communicate with the PIC to produce the actuation output. The actuation output was to control the maneuvering system which is by rotating the steering. So, this vehicle will move constantly distance from highway divider when the Autoguide system have activated.

The sensor at the front of the vehicle will detect any foreign object or other vehicle in front. So, this will void vehicle from accident.

3.1.3 System limitation

Firstly the problem of highway divider, when there is a maintenance work or unexpected foreign object in between of the safe distance. This will make error to the sensor detection. The other point is if the highway divider might change in shape because of the early condition or have accident before with the other vehicle.

The other limitation is about the condition of road, where the consideration of the direction change and not balance left-right condition might appear.

3.2 Road Simplification

This simplification made because to make sure the flow of this project smoother and also to almost eliminate the uncertainty that will be faced along the project.

About the road condition at highway, there is a lot of uncertainty involving road and among them not perfectly flat. This problem will effect to the vehicle steering when control the direction of vehicle.

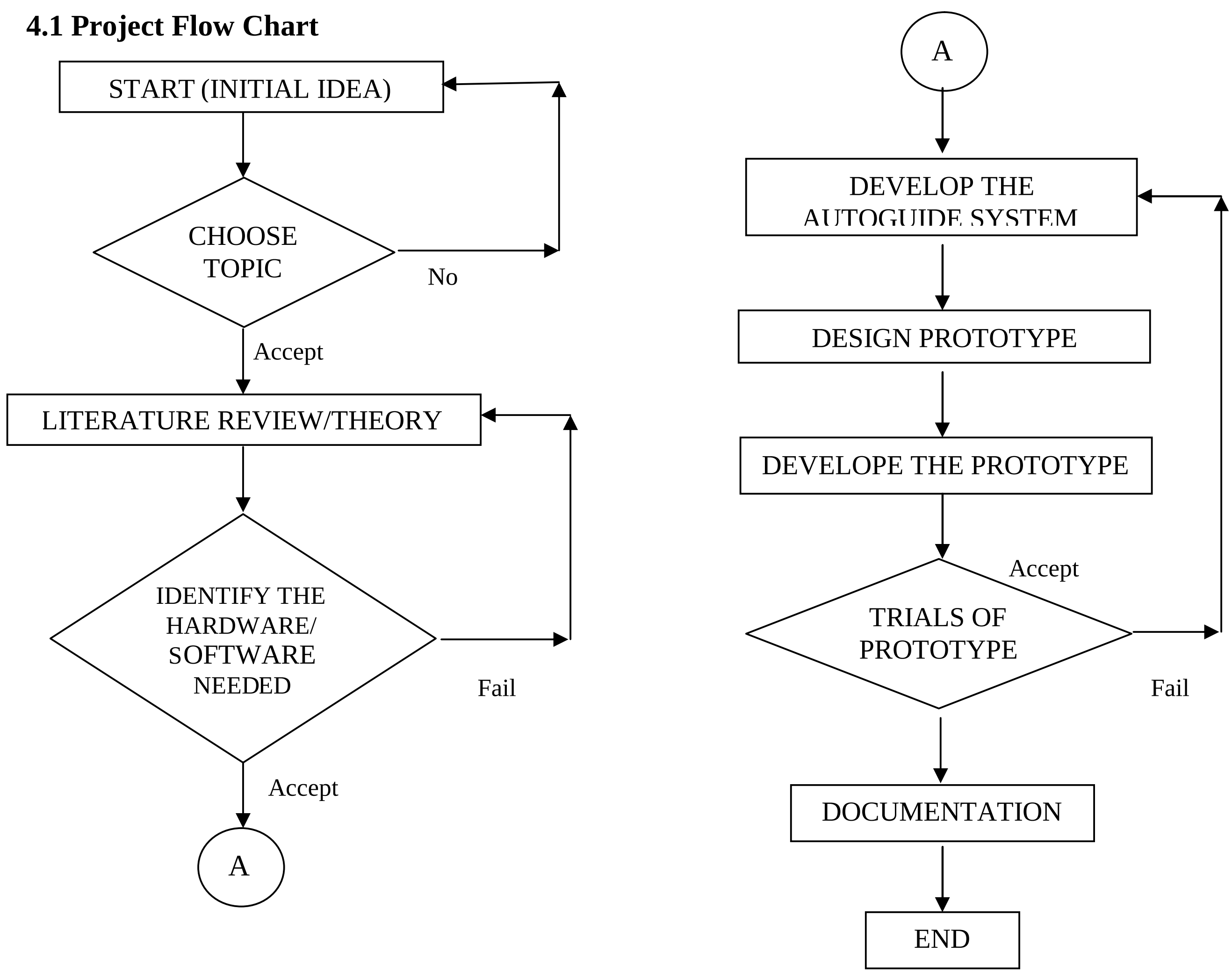
Therefore to be simplified, the road will considered as a flat surface on the ground. So, there is no problem when the Autoguide system controls the vehicle steering. The system will just follow the programmed that have setup before.

CHAPTER 4

METHODOLOGY

The project starts after finalizing and refining the proposal topic and final report that have done by final year student before. That title topic before are ‘AutoGuide System for Conventional Vehicle on Highway’. The project has done with successfully and the result has shown by using Mathlab simulation. This simulation will demonstrate how to control the steering and the vehicle velocity using logic operation and combination of condition. For my project, I’ll conduct to build a prototyped and some improvement about the system. After discussions with Supervisor, Puan Rosmawati, the title ‘The design of an AutoGuide System Prototyped for Conventional Vehicle on Highway have been selected.

4.1 Project Flow Chart



4.2 Project Gantt Chart

Final Year Semester 1 (FYP 1)											
week	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**Semester break**
Submission of preliminary report												
Submission of progress report												
Seminar												
Submission of interim report												
Oral presentation												
Study of literature review/theory based on: - Measurement range of sensor - Constant speed of vehicle												
Study about sensor (selection of sensor) - Ultrasonic - Infrared - Infrared reflectance - Laser Range												
Study about software - MP Lab - Circuit drawing Software - Catia												
Develop the Autoguide system - Flowchart of system - Drawing the circuit of system - Program the PIC												

Final Year Semester 2 (FYP 2)														
week	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
Design the prototype - Using Catia														
Find the Hardware needed - Sensor - PIC														
Build the prototype - Circuit - Programming of PIC														
Testing and repair the prototype - Build track for prototype														
Oral presentation														
Final report														

4.3 Selection of sensor

The selection of sensor based on a few factor. Firstly, function of sensor for this project. The sensor need for this project because to detected the highway divider and measuring the distance between vehicle and highway divider. So, the sensor choose must have the function to measuring the distance.

Secondly, the factor is range of sensor. Since the save distance between vehicle and highway divider are 2.0m, so the sensor selected must have ability to detect the obstacle at least 2.0 m. Higher range of detection, it is more better.

Other factor must consider in selection of sensor is about the characteristic/ properties to the environment. Since the sensor will be put outside of the vehicle, the sensor always expose to the bad condition of weather or environment such as rain, dust and dirt. The vehicle also must used for travel along a day or night, so the sensor must not dependent to weather/environments.

About the economics view, the selection of sensor must be the lowest in cost but the function of that sensor still the same or better compare to the other sensor. Lastly, the factor is about knowledge of that sensor.

Table 2 below shows the selection of sensor

	Ultrasonic	Infrared	Infra-Red Reflectance	Laser range
Function	*Measurement of range*	*Measurement of range*	*Measurement of range*	*Measurement of range*
Range of measurement	*2 meter and above*	*2 meter and above*	*2 meter and above*	*Longer range*
Effect with environments	*Less effected*	*Sensitive to ambient temperature*	*Sensitive to ambient temperature*	*Not effected by environment*
Cost	*Low*	*Low*	*Low*	*High*
Knowledge	*High*	*Medium*	*Low*	*Low*

Table 2

From the table above, by taking this entire factor into consideration, ultrasonic sensors emerge as one of the strongest candidates for used in this project. Ultrasonic is a sensor to measure the range. Qualification for the sensor about measuring the range is minimum 2 meters, so ultrasonic has met this requirement. Ultrasonic sensors are less affected by adverse weather conditions and are low in cost. Lastly about the knowledge on that sensor is quite high. If the knowledge about this sensor is high, so that the project will be done with more smooth and better producing of prototype.

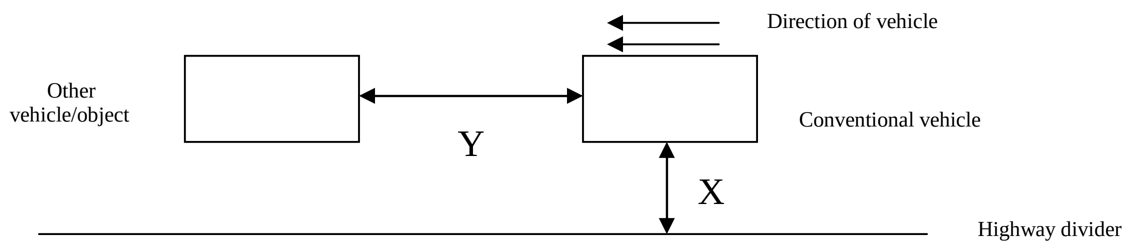
Infrared is fulfilling all the criteria needed except it is sensitive to ambient temperature. Since the vehicle use on the highway for a long journey and the sensor will expose with the bad environment/weather (snow, rain and dust), so this characteristic of sensor about effected to environment very important. This is same goes to infra-red reflectance sensor. In addition is the knowledge of author about this sensor is low.

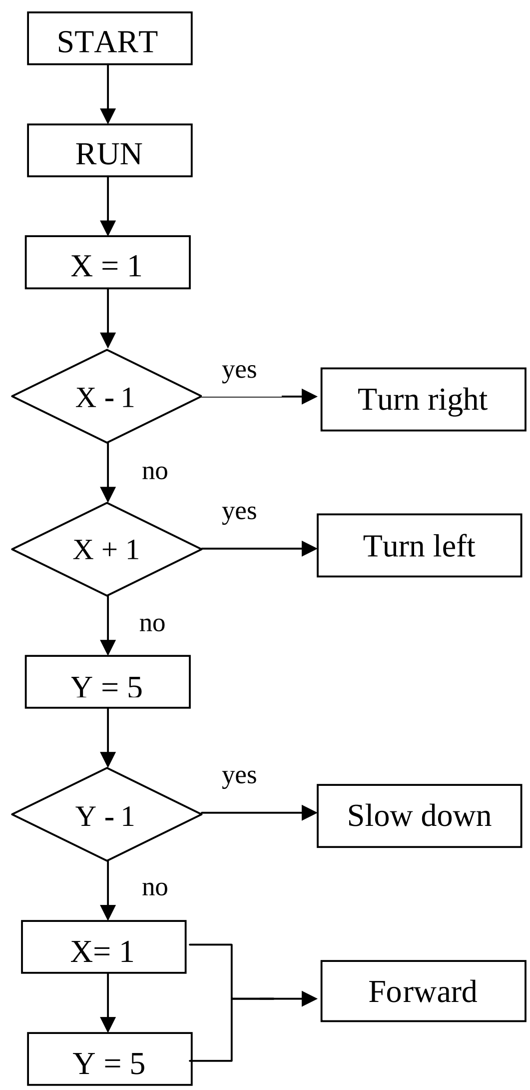
Laser range sensor is a one of the good sensor in measurement of range. But, the disadvantage of this sensor is high in cost and the low of knowledge.

4.4 Construction of Circuit

Before construct the circuit for this project, a few things must be done first. Which is flowchart of the of circuit system work, block diagram and schematic drawing. This flow must be follow to get the perfect circuit also to avoid the waste of time and money.

Flow chart of block diagram



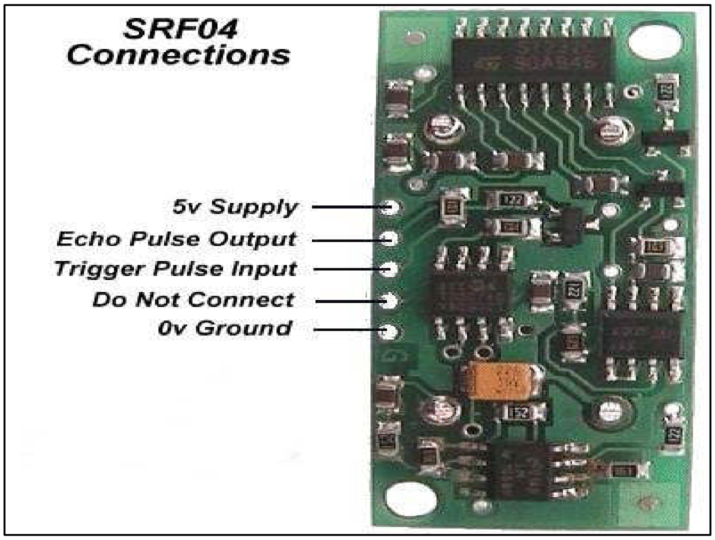


CHAPTER 5

DESIGN AND RESULT

5.1 Ultrasonic sensor – SRF04

The SRF04 was design to be just as easy to use as the Polaroid sonar, requiring a short trigger pulse and providing an echo pulse. The controller only has to time the length of this pulse to find the range. The connections to the SRF04 are shown below:



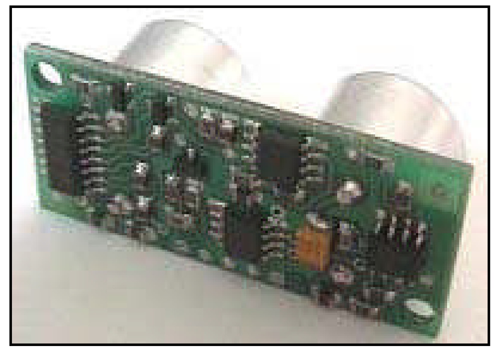


Figure 11: Example of ultrasonic sensor

This prototype for autoguide system has two ultrasonic sensors. One of the ultrasonic sensors is used to locate walls of track on its left side and detect obstacles in the front. The ultrasonic distance sensor provides precise, non-contact distance measurements. It transmits an ultrasonic burst to detect any obstacle and outputs a pulse. The output pulse width corresponds to the time required for the burst echo to return to the sensor. The distance between the prototyped and wall can be determined by multiplying the pulse width and the velocity of the ultrasonic burst. The ultrasonic sensor (SRF04) has a male four-pin header used for +5V supply (Vdd), ground (Vss), echo pulse output and trigger pulse input as shown in Figure .

5.2 Drawing of the track for Autoguide Sytem prototyped testing

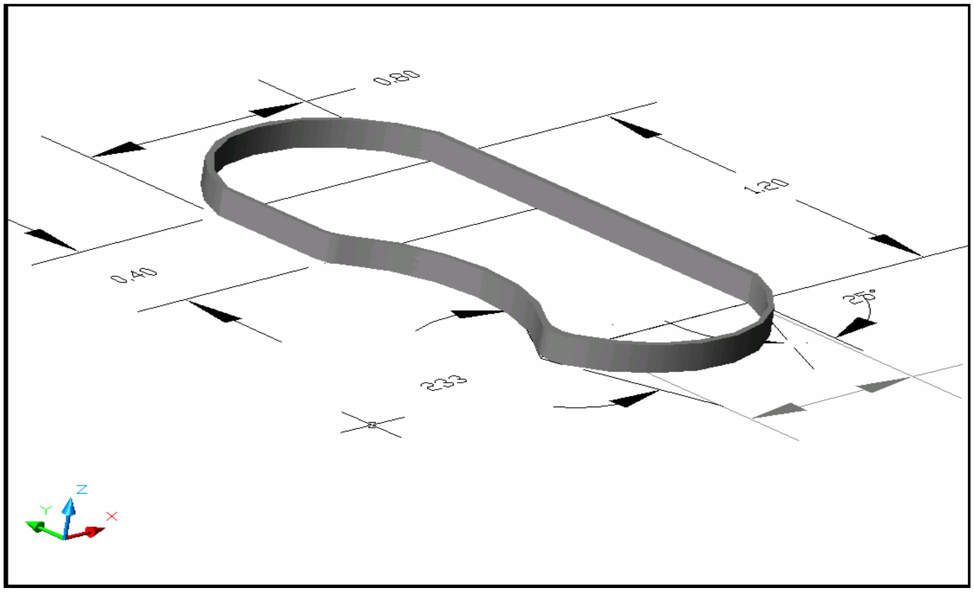


Figure 12: layout 3D of track for testing

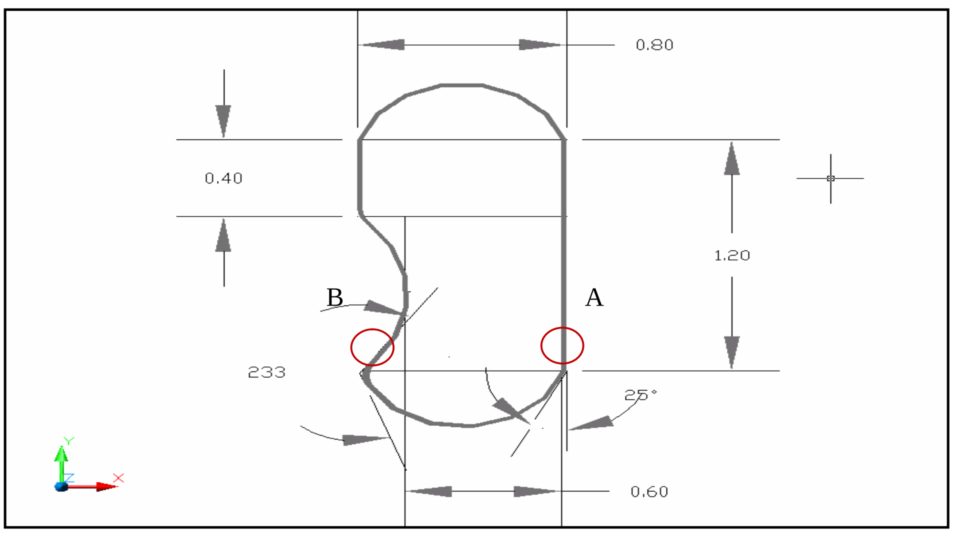
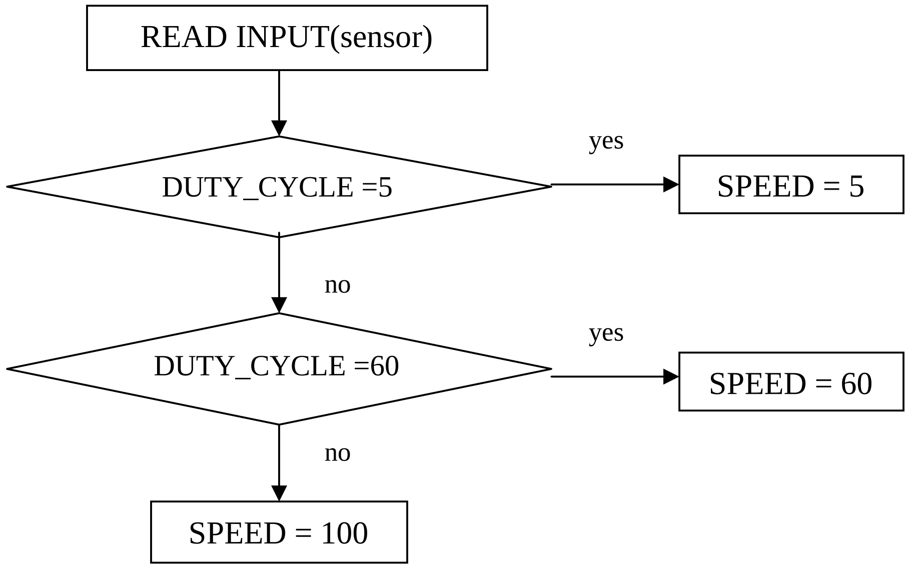


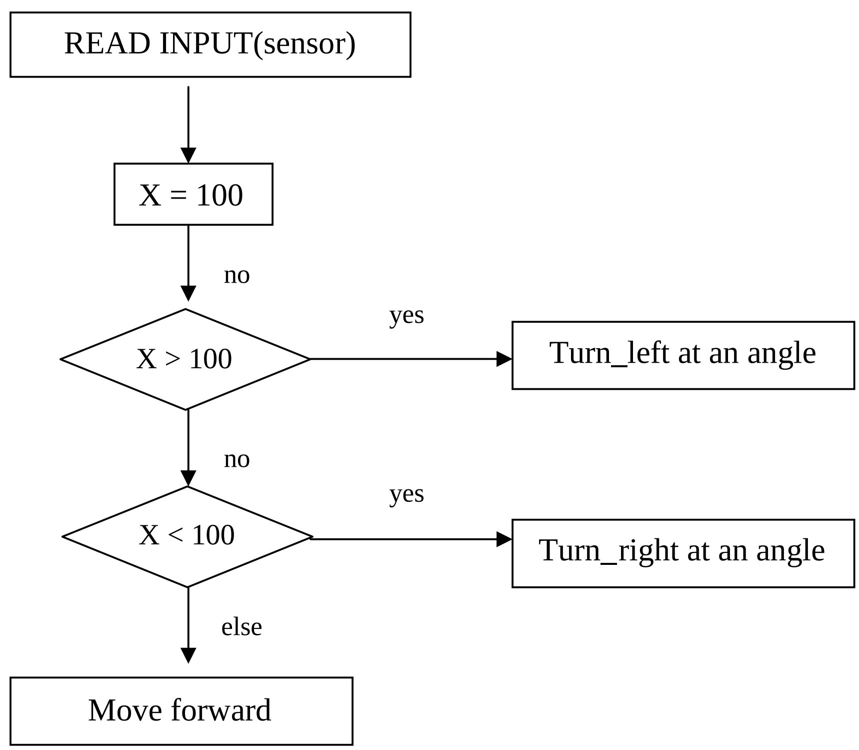
Figure 13: top view of track for testing

Total length of this track is 5.0m. for the straight line, the length is 1.2m. Angle at the corner A is 25 degree and at corner B is 233 degree. The track build based on the ability of prototyped to follow the track at the straight, acute angle, and bigger angle line. Height of the wall is 0.1m.

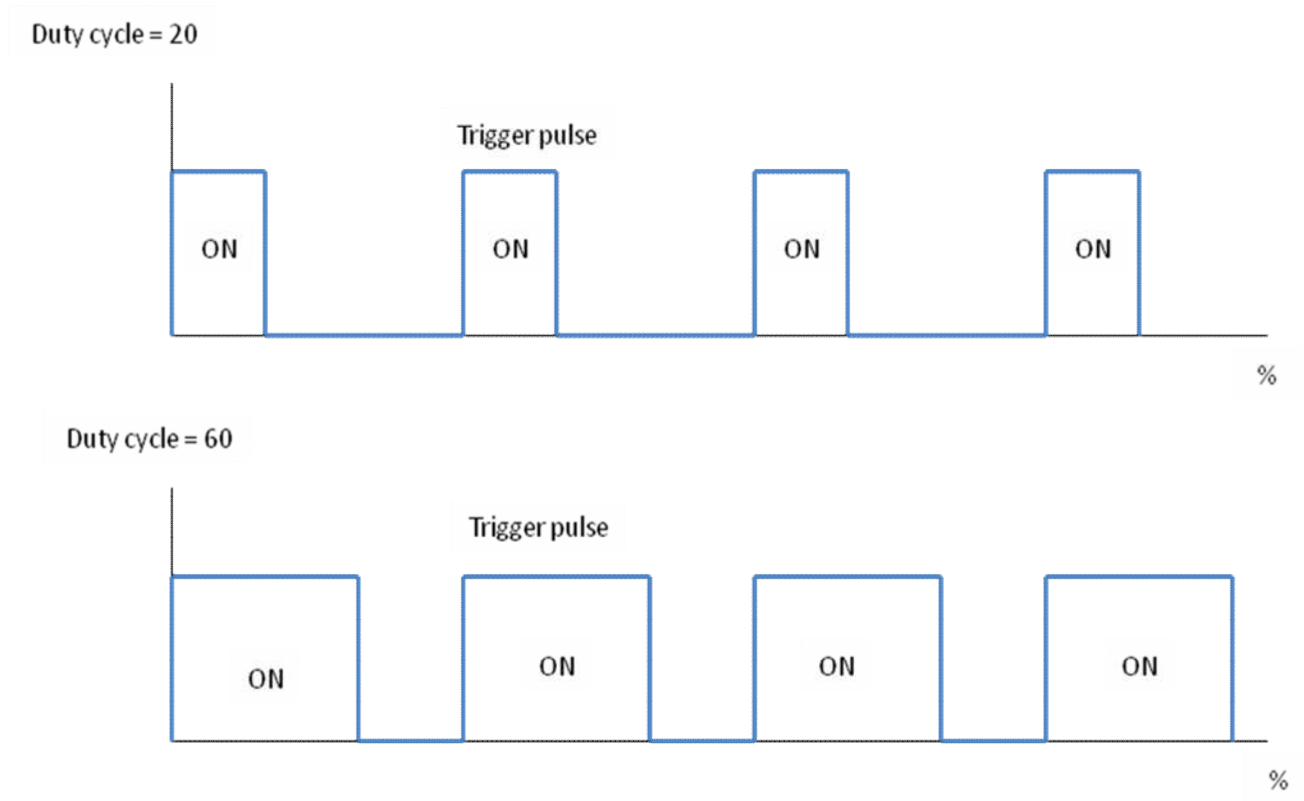
5.3 Flow chart of Autoguide System Prototype (front sensor)



5.4 Flow chart of Autoguide System Prototype (side sensor)



5.5 SRFO4 Timing Diagram



In this circuit, the definition of duty cycle is the ratio of off time to on time. Used of duty cycle is to control the speed of motor in autoguide system prototyped. For the higher duty cycle, the on time will increase and make the speed of prototyped increase.

5.6 Voltage Regulator

Voltage regulator is an electrical regulator design to automatically maintain a constant voltage level. It may use an electromechanical mechanism, or passive or active electronic components. Depending on the design, it may be used to regulate one or more AC or DC voltages.

In this project, the voltage regulator is to reduce and maintain the voltage for 5V before supply to PIC.

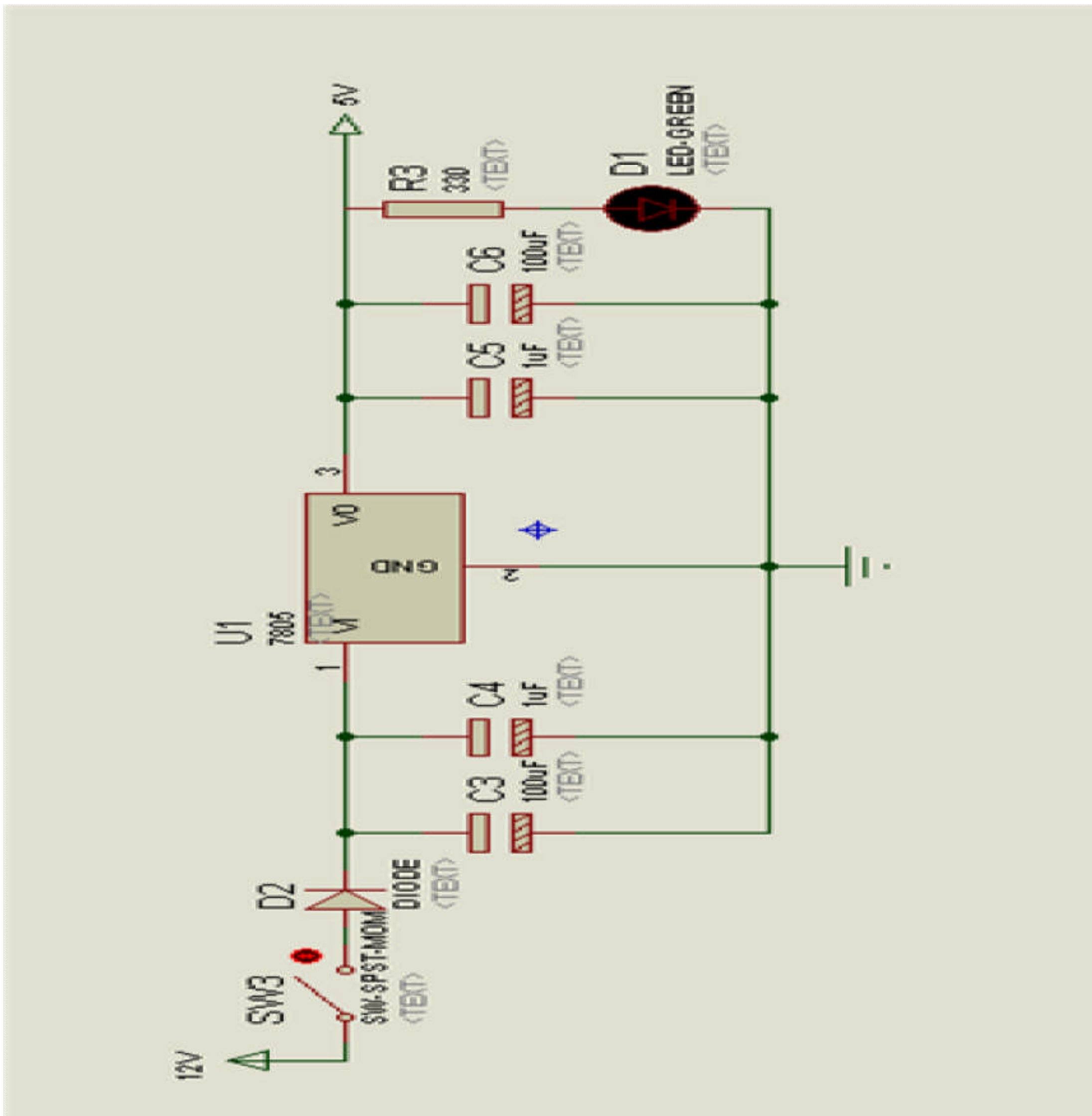


Figure 14: Drawing circuit of voltage regulator

5.7 Analog Input and Motor Driver

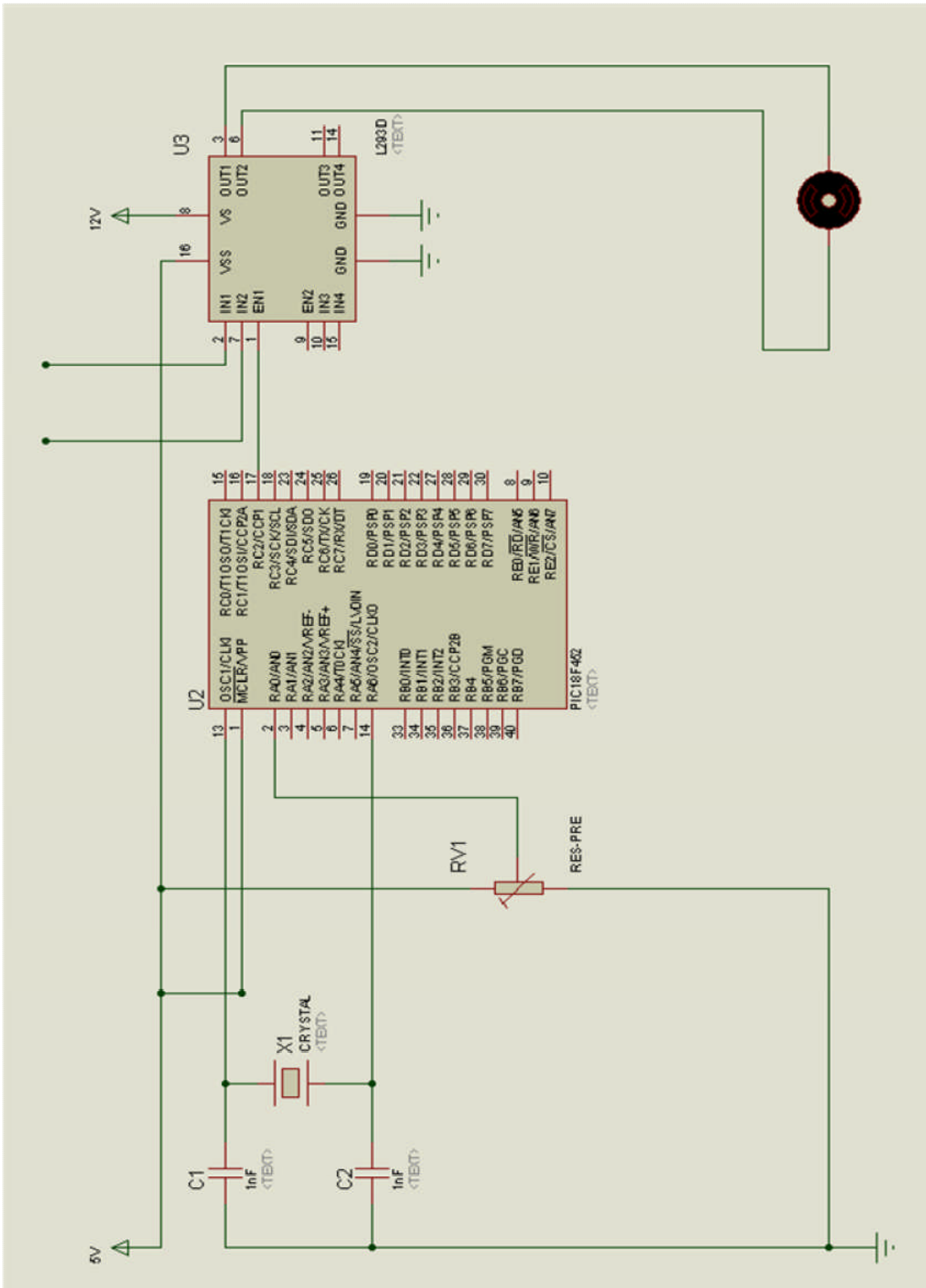


Figure 15: Drawing circuit of analog input and motor driver

5.8 Bill of Materials (BOM)

Table 3 below shows the bill of material (BOM) for this project

Unit	**Component**	**Model/value**	**No**
(1) volatge regulator	diode	1N4001	1
	electrolytic capacitor	100uF	2
		0.1uF	2
	voltage regulator	7805	1
	LED indicator	1	1
	resistor	360 Ohm	1
(2) Oscillator	oscillator	4MHZ	1
(3) Microcontroller	microchip	PIC16F877A	1
	socket	zip socket	1
(4) PWM generator	microchip	PIC16F877A	1
	socket	zip socket	1
(5) Motor driver	motor driver	L293D	1
	pin header		4
(6) Detector	ultrasonic sensor	SRF04	2
(7) Power supply	battery 1.5V		8

Table 3

5.9 Circuit for Autoguide System Prototyped

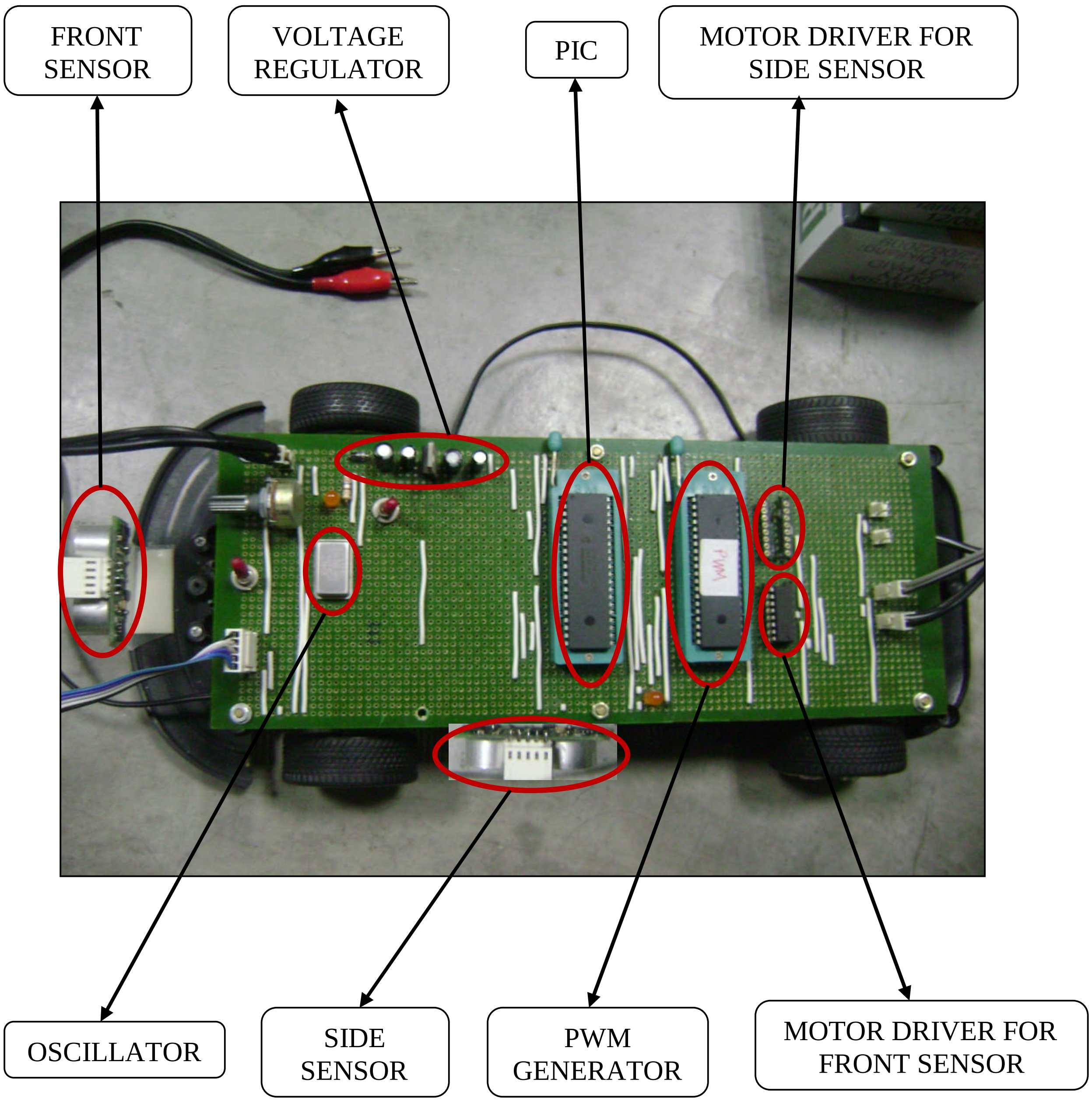


Figure 16: Circuit for Autoguide System Prototyped

5.10 Testing of the Autoguide System Prototype for side sensor (following wall track)

5.9.1 for the straight line

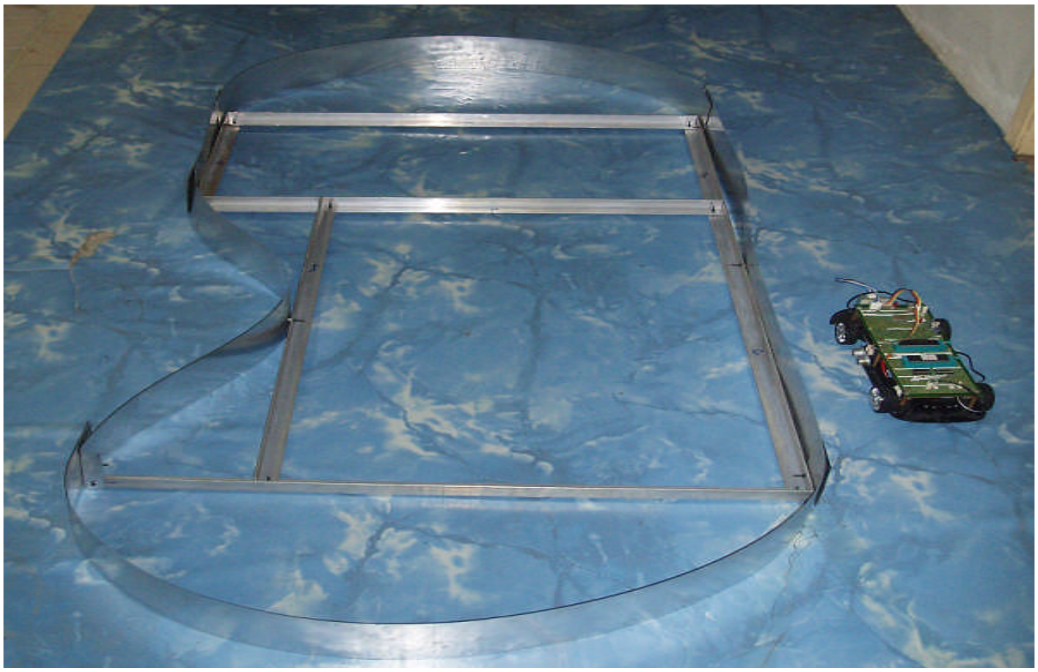


Figure 17: Testing for the straight line

5.9.2 for the angle 25˚

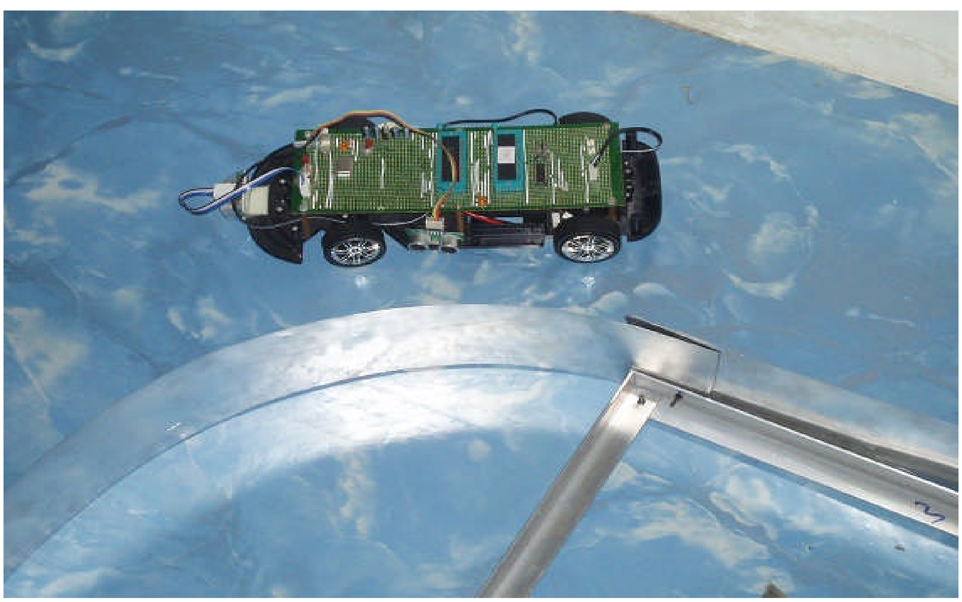


Figure 18: Testing for the angle 25˚

5.9.3 for the angle 35

Figure 19a: Testing for the angle 35˚

Figure 19b: Testing for the angle 35˚

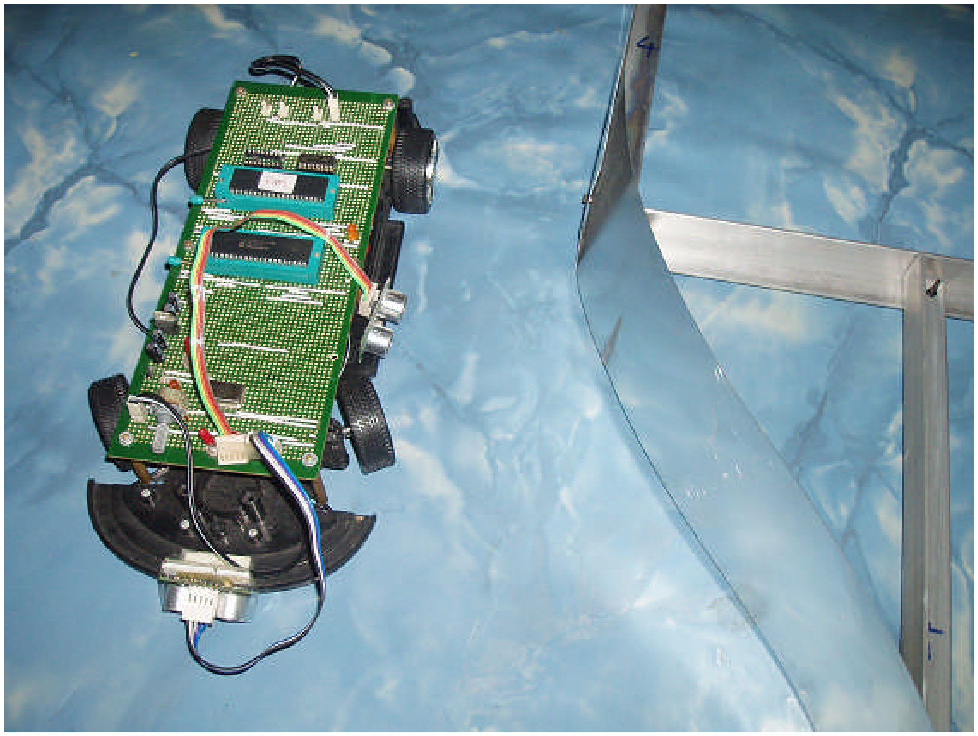


Figure 19c: Testing for the angle 35˚

The angles that consider in this testing were for the straight line (0˚), 25˚ and 35˚. These angles choose randomly based on the ability tire of the prototype to turn left or right at the specific angle. The distance between the prototyped and wall of track is 10cm. So, in the programming will fix the value of distance between prototyped and wall must always 10cm. This will make the prototyped move along the track by set wall as a reference point. The height of this wall is 10cm.

5.11 Testing of the Autoguide System Prototype for front sensor (detection wall in front the prototype)

Figure 20: Testing prototype for detection of object in front

The speeds of the prototype depend on the front sensor. If no obstacle in front, the prototype will move with the maximum speed that setup in programming. The speed decrease with the distance prototype with wall/obstacle more closure. The calculation made based on formula below:

$$\text{Duty cycle} = \frac{\text{speed} + 100}{2}$$

CHAPTER 6

CONCLUSION

‘The design of an AutoGuide System prototyped for conventional Vehicle on Highway’ is a project that is feasible and should develop in future for the main reason stated in problem statement. The implementation of this project into real application is highly recommended as it can improve the department of transportation efficiency. In addition it is also will enhance the highway user safety.

At the end of this project, the prototype will build to prove its reliability. The selection of material, hardware and software must do with carefully to make sure the prototype is work. Lastly, more research needed to make this project success and the function of this system can be shows to others. The improvement for every single thing must do from time to time.

Suggestion improvement:

- Improve the safety for driver and passenger
- Alarm activated when the vehicle stop because of system error or have the static vehicle at the front of vehicle
- Display the speed of vehicle prototype
- Use more accurate sensor
- Consider the type of surface for the track testing
- Consider the weight of prototype
- The conduction speed of the motor does not vary with the weight of the prototype. So, use the motor that have higher speed
- Test the prototyped with higher angle of corner

REFERENCES

[1] Xiao Chen, 2009 “International Conference on Industrial Mechatronics and Automation” *Reversing Radar Based on CAN Bus*

[2] N. Venkata Murali and C. Chandramouli, September 2009 *Driver Assistance System (DAS)*

[3] Can Zhang, Jiankang Wu, Guofang Tu, 2006 *Object Tracking and QOS control Using Infrared Sensor and Video Cameras*

[4] Ioannou, P.A. and Chien, C.C., 1993, *Autonomous Intelligence Cruise Control*, 42(4)

[5] Palmquist, 2002, *Intelligence Cruise Control and Roadside Information*, 13(1)

[6] Marjchrzak, J, Michalski, M and Winzynski, G, July 2009, *Distance Estimation with a Long-Range Ultrasonic Sensor System*, 9(7)

[7] Miyazaki, K. Hashimoto, M. Shimada, M., and Takahashi, K., 2009, *Guide following control using laser range sensor for a smart wheelchair*

[8] Mehta, Samarth, 2007, *An Autonomous Wall Following Robot*

[9] Thomas Fischl, 2007 <www.fischl.de/**usb**asp/>

[10] <http://www.microchip.com/stellent/>

[11] <http://www.labcenter.co.uk/products/vsm_overview.cfm>

[12] http://www.brothersoft.com/ktechlab-71211.html

[13] T.A. Lasky and B. Ravani, 1993, "*Review of Research Related to Automated Highway System (AHS)*" Transportation Research Part C 11 (2003) 1–28

[14] Farid Efanddy bin Hassan, Jan 2009, *Autoguide system for conventional vehicle on highway*, Degree Thesis, Universiti Teknologi Petronas, Malaysia

[15] S Selvaraj, S Dhanushkadi and P Jayaprakasan, 2006, *Automobile Technology'*

[16] John F. Kershaw and James D. Halderman, 2007, *Automotive Electrical and Electronic System*

[17] David G. Ullman, 2003, *The Mechanical Design Process*

[18] Nigel Gardner, 2002, *An Introduction to Programming The Microchip PIC in CCS C*

APPENDICES

Appendix 1: Programming for front sensor range

```
#include <16f877.h>
#device ADC=8
#USE DELAY(CLOCK=4000000) /* Using a 4 Mhz clock */
#FUSES XT,NOWDT,NOPROTECT,NOLVP
/* Use XT mode, No Watch Dog, No Code Protect, No Low Voltage Programming
*/unsigned int8 adcValue ,x;
int duty_cycle ,speed;
main()
{
setup_adc_ports( ALL_ANALOG );
setup_adc(ADC_CLOCK_INTERNAL); // Use internal ADC clock.
//set_tris_c(0xff);
setup_timer_2(T2_DIV_BY_1,120,16); //enable Timer2, PR2=99, prescaler=1
setup_ccp1(CCP_PWM); //enable PWM mode
set_tris_b(0x00);//set all pins at portB as input
set_tris_d(0x00);//set all pins at portD as output

while(1)
{
set_adc_channel(0);
delay_us(50); // Delay for sampling cap to charge
adcValue = read_adc(); // Get ADC reading
output_b(0x01);
duty_cycle = (adcValue/255.0)*100;
if(duty_cycle <= 5)
{
speed = 0;
output_d(0x01);
```

```
}
else if(duty_cycle <= 10)
{
speed = 10;
output_d(0x02);
}
else if(duty_cycle <= 20)
{
speed = 20;
output_d(0x02);
}
else if(duty_cycle <= 30)
{
speed = 30;
output_d(0x04);
}
else if(duty_cycle <= 40)
{
speed = 40;
output_d(0x02);
}

else if(duty_cycle <= 50)
{
speed = 50;
output_d(0x04);
}
else if(duty_cycle <= 60)
{
speed = 60;
output_d(0x08);
```

```
}
else if(duty_cycle <= 70)
{
speed = 70;
output_d(0x10);
}
else if(duty_cycle <= 80)
{
speed = 80;
output_d(0x20);
}
else if(duty_cycle <= 90)
{
speed = 90;
output_d(0x40);
}
else if(duty_cycle <= 100)
{
speed = 100;
output_d(0x80);
}
else
speed = 0;
CCP_1 = speed;
```

Appendix 2: Programming for side sensor range

```
#include <16f877.h>
#device ADC=8
#USE DELAY(CLOCK=4000000) /* Using a 4 Mhz clock */
#FUSES XT,NOWDT,NOPROTECT,NOLVP
// Use XT mode, No Watch Dog, No Code Protect, No Low Voltage Programming
// Unsigned int8 adcValue ,x;
#define velocity_cm 34442.4     // velocity of ultrasonic wave in cm/s

Int32 left_distance;              // from ultrasonic sensors, left_distance is side sensor
reading
Int16 overflow_count;            // overflow count for timer 1
Int value = 0;                   // PMW on time value in term of between 0 to 255
Signed int Speed = 0             //desired speed
Int leftturn, rightturn = 0      // number of left and right turns

void get_value ();               // get PWM on time value for speed
void ultrasonic ();              // read distance
void move_fwd ();                // move forward
void move ();                    // move forward in the beginning
void turn_right (int time);      // turn right for x amount of time (x is called time)
void turn_left;                  // turn left


void main ()
{
        Float x_ref = 30;               //reference value
        Output_low(PIN_D2);             //disable motor
        Delay_s(5);                     //5 second delay
```

```
        Setup_ccp1 (CCP_PWM;      //configure CCP1 as a PWM
        Setup_timer_2(T2_DIV_BY_16, 255, 16);           //configure time period for
PWM


        Speed = 50;                    // initiate the robot with 50% speed
        Get_value();                   // get PWM on time value for 50% speed
        move();                        //  moves forward with 50% speed
        ultrasonic ();                 // read distances
        Speed = 60;                    // increase speed to 60 %
        get_value();
        move();                         // moves forward with 60% speed

// controller code
while(True)
{
prev_error = error;                    // update previous error


// obstruction in front and too close to the wall and at an angle
if (error <= -2 && average <= 23 && front_distance < 100)
{
turn_right(500);                       //turn right for 500 ms

}
// vehicle moving away from reference value and is closer to the wall at an angle
else if (error <= -2 && error < prev_error && average <= 23 &&
average > 15)
{
decrease(15);

}
// vehicle is at the same position and is closer to the wall and at an angle
else if (error <= -2 && error == prev_error && average <= 23 &&
average > 15)
{
decrease(0);

}
// vehicle is moving closer to the reference value and is closer to the wall at an angle
else if (error <= -2 && error > prev_error && average <= 23 &&
average > 15)
{
decrease(5);
```

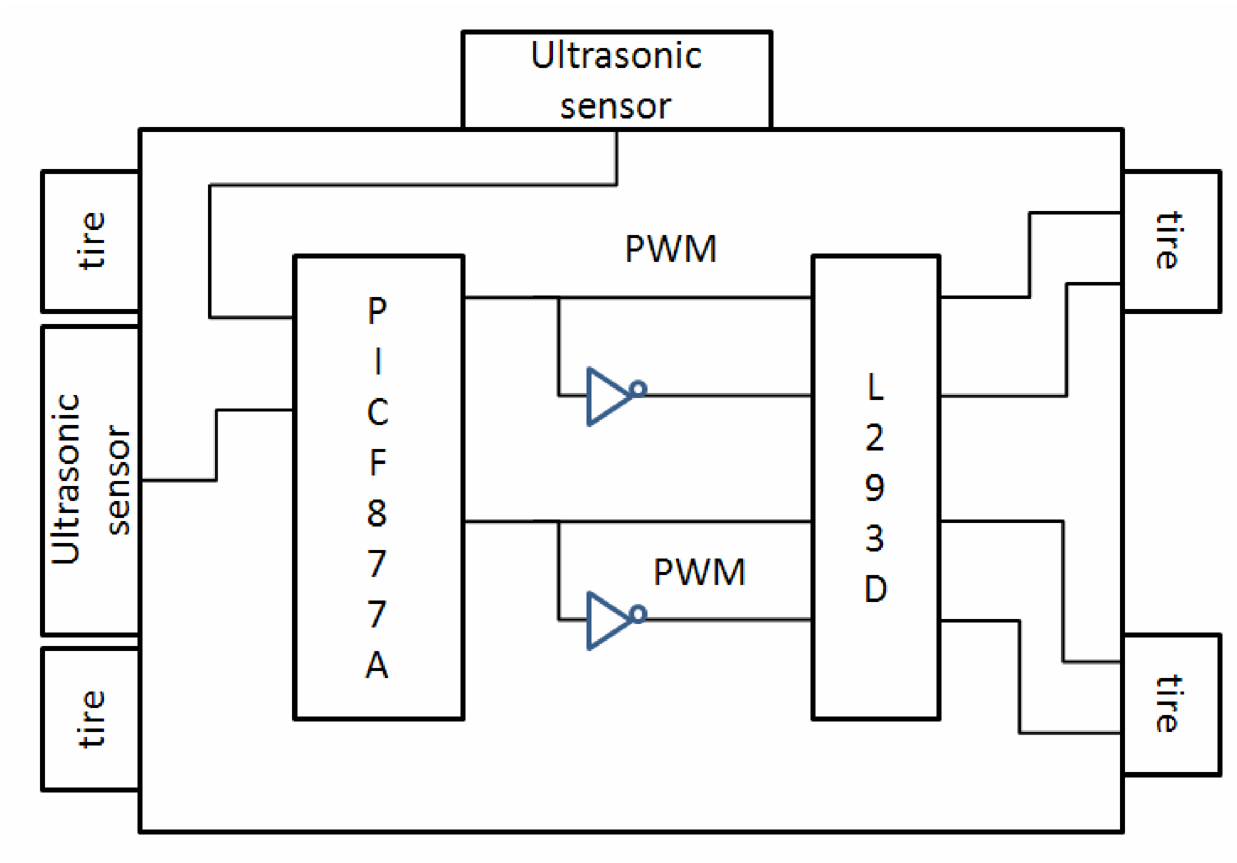
```
}
// vehicle is moving away from reference value and is closer to the wall
else if (error < prev_error && average < 28 && average > 23 )
{
decrease(10);
}
// vehicle is moving closer to reference value and is closer to the wall
else if (error > prev_error && average < 28 && average > 23 )
{
decrease(5);

}
// vehicle is at the same position and is closer to the wall
else if (error == prev_error && average < 28 && average > 23 )
{
decrease(0);

}
// obstruction in front and too away from the wall at an angle
else if ( error >= 2 && error > prev_error && average >= 37 &&
front_distance < 100 )
{
turn_left();

}
// vehicle  is moving away from the reference value and is away from the wall at an angle
else if (error >= 2 && error > prev_error && average > 37 &&
average < 45)
{
increase(10);

}
// vehicle is at the same position and is away from the wall at an angle
else if (error >= 2 && error == prev_error && average > 37 &&
average < 45)
{
increase(0);

}
// vehicle  is moving closer to the reference value and is away from the wall at an angle
else if (error >= 2 && error < prev_error && average > 37 &&
average < 45)
{
increase(5);

}
```

```
//vehicle is moving away from the reference value and is away from the wall
else if ( error > prev_error && average > 32 && average < 37 )
{
increase(10);


}
// vehicle is moving closer to the reference value and is away from the wall
else if ( error < prev_error && average > 32 && average < 37 )
{
increase(5);

}
// vehicle is at the same position and is away from the wall
else if ( error == prev_error && average > 32 && average < 37 )
{
increase(0);
}

// vehicle is at an angle less than 90 degrees with respect to x axis an closer to the wall
else if (error >= 2 && error > prev_error && average < 15)
{
increase(1);

}
// vehicle is at an angle greater than 90 degrees with respect to x axis and closer to the
wall
else if (error >= 2 && error < prev_error && average < 15 )
{
decrease(1);

}
// vehicle is at an angle greater than 90 degrees with respect to x axis and away from the
wall
else if (error >= 2 && error > prev_error && average > 45)
{
decrease(10);
}

// if vehicle is too close to the wall decrease right wheel speed by the largest amount,
//However, it's been checked again after this sequence and corrected by a turn.
else if (error >= 2 && error < prev_error && average < 15 )
{
decrease(20);
}
```

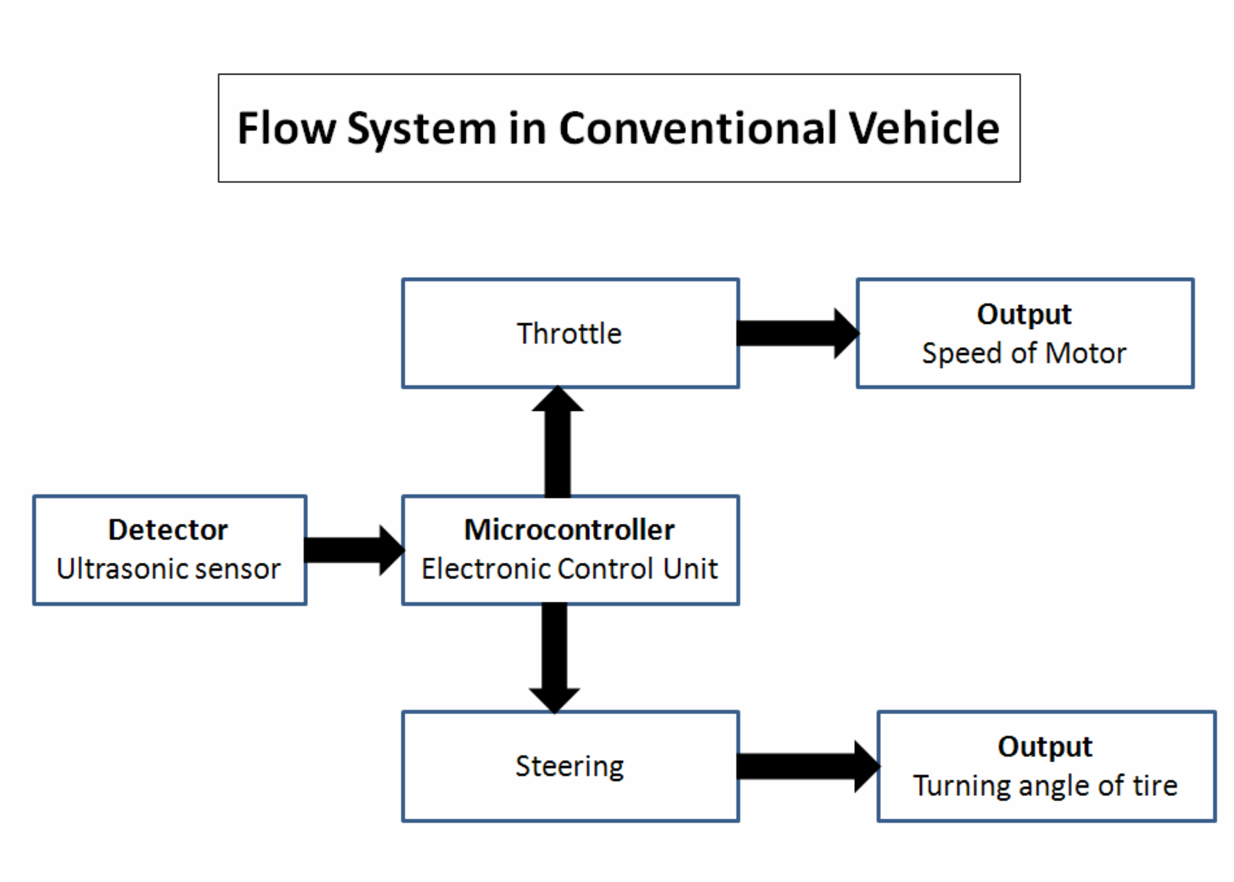
```
// for minor corrections in angle
else if (error<=-2)
{
decrease(1);
}

// for minor corrections in angle
else if (error>=2)
{
increase(1);
}

// vehicle is too close to the wall (for immediate correction)
if (average < 20 )
{
turn_right(250);
}
{
turn_right(500);
}
void turn_right(int time)
{
if (rightturn <= 2)
{
set_pwm1_duty(speed_65);              // This sets the on time for PIN #
output_high(PIN_D2);                   // enable motors
delay_ms(time);
output_low(PIN_D2);                   // disable motors
ultrasonic();                         // get UDM reading
rightturn++;                          // increment right turn count
}
else
{
rightturn = 0;                         // reset the right turn count
Speed = 65;                            // set nominal speed
get_value();
move_fwd();                           // move forward

{
int32 time;                           // travel time for sonar
int32 distance;                       // disctance between ultrasonic sensor and obstacle
int32 ultra[3];                        // right, center, left distance
setup_timer_1(T1_INTERNAL);            // using timer1
{
enable_interrupts(global);             // enabling global interrupt controls
// whether interrupts are serviced
```

```
/* input pulse of 10us is required
to turn-on ultrasonic range finder */

output_low(pin);                    // pin kept low for 1 ms
delay_ms(1);
output_high(pin);                    // pin kept high for 10us for pulse
delay_us(10);
output_low(pin);                    // pin brought back to low again to finish the pulse

/* Since same pin is used to transmit and recieve a signal,
PIN_C7 needs to be checked continuously for high */
while(!input(pin));                 // wait for signal to go high
set_timer1(0);                      // set timer1 to zero and turn it on as soon as signal
goes high
overflow_count = 0;                 // initialize overflow_count identifier
while(input(pin));                   // wait for signal to go low
disable_interrupts(global);         // disable global interrupts
time = get_timer1();                // get the pulse width

/* add 65535 when overflow_count is 1 to the original time */
time = time + ((int32)overflow_count<<16);
time -= 15;                         // subtract overhead
time /= 2;                          // remove return trip
distance = velocity_cm * time;       // distance between ultrasonic sensor and obstacle

/* clock tick for timer1 is 10MHz.
divide by 10MHz to get time in seconds.

division in the end gives less error, since
more significant numbers can give closer value */
distance /= instruction_time;       // get original distance in cm
ultra[x] = distance;
pin++;
}
left_distance = ultra[0];           // left_front distance
```

Appendix 3: Hardware block diagram for Autoguide system prototype



Appendix 4: Flow system in conventional vehicle



Appendix 5: Picture of prototype