

**DEVELOPMENT OF CONTROL SYSTEM FOR
AUTOMATED GUIDED VEHICLE (AGV)**

SYED MOHD SAFWAN BIN SAYED MD SAIFUDDIN

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS♦

JUDUL: **DEVELOPMENT OF CONTROL SYSTEM FOR
AUTOMATED GUIDED VEHICLE (AGV)**

SESI PENGAJIAN: 2009/2010

Saya SYED MOHD SAFWAN BIN SAYED MD SAIFUDDIN
(HURUF BESAR)

mengaku membenarkan tesis (Sarjana Muda/~~Sarjana~~ /~~Doktor Falsafah~~)* ini disimpan di Perpustakaan dengan syarat-syarat kegunaan seperti berikut:

1. Tesis adalah hakmilik Universiti Malaysia Pahang (UMP).
2. Perpustakaan dibenarkan membuat salinan untuk tujuan pengajian sahaja.
3. Perpustakaan dibenarkan membuat salinan tesis ini sebagai bahan pertukaran antara institusi pengajian tinggi.
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

Disahkan oleh:

(TANDATANGAN PENULIS)

(TANDATANGAN PENYELIA)

Alamat Tetap:

**90 TAMAN SERI SETIA
26600 PEKAN
PAHANG**

WAN SHARUZI BIN WAN HARUN
(Nama Penyelia)

Tarikh: **20 NOVEMBER 2009**

Tarikh: : **20 NOVEMBER 2009**

- CATATAN:
- * Potong yang tidak berkenaan.
 - ** Jika tesis ini SULIT atau TERHAD, sila lampirkan surat daripada pihak berkuasa/organisasi berkenaan dengan menyatakan sekali tempoh tesis ini perlu dikelaskan sebagai atau TERHAD.
 - ♦ Tesis dimaksudkan sebagai tesis bagi Ijazah doktor Falsafah dan Sarjana secara Penyelidikan, atau disertasi bagi pengajian secara kerja kursus dan penyelidikan, atau Laporan Projek Sarjana Muda (PSM).

DEVELOPMENT OF CONTROL SYSTEM FOR AUTOMATED GUIDED
VEHICLE (AGV)

SYED MOHD SAFWAN BIN SAYED MD SAIFUDDIN

Report submitted in partial fulfillment of the
requirements for the award of the
Bachelor of Mechanical Engineering with Automotive

Faculty of Mechanical Engineering
University Malaysia Pahang

NOVEMBER 2009

SUPERVISOR'S DECLARATION

We hereby declare that we have checked this project and in our opinion this project is satisfactory in terms of scope and quality for the award of the degree of Bachelor of Mechanical Engineering

Signature :
Name of Supervisor : WAN SHARUZI BIN HJ. WAN HARUN
Position : LECTURER
Date : 20 NOVEMBER 2009

Signature :
Name of Panel :
Position : LECTURER
Date : 20 NOVEMBER 2009

STUDENT'S DECLARATION

I hereby declare that the work in this report is my own except for quotations and summaries which have been duly acknowledged. The report has not been accepted for any degree and is not concurrently submitted for award of other degree.

Signature :

Name : SYED MOHD SAFWAN BIN SAYED MD SAIFUDDIN

ID Number : MH06067

Date : 20 NOVEMBER 2009

To my beloved father mother
Mr Sayed Md Saifuddin bt S Abu Bakar
Madam Noridah bt Daud

ACKNOWLEDGEMENT

I am grateful and would like to express my sincere gratitude to my supervisor En.Wan Sharuzi bin Hj. Wan Harun for his brilliant ideas, invaluable guidance, continuous encouragement and constant support in making this research possible. He has always impressed me with his outstanding professional conduct, his strong conviction for science, and his belief that a Bachelor program is only a start of a life-long learning experience. Also thanks to my co-supervisor, En. Nafriuzan bin Mat Yahya and Pn. Azlyna bt Senawi. I appreciate their consistent support from the first day I applied to PSM course to these concluding moments. I am truly grateful for their progressive vision about my work progressing, their tolerance of my naive mistakes, and their commitment to my future career. I also would like to express very special thanks again to my supervisor for his suggestions and co-operation throughout the study. I also sincerely thanks for the time spent proofreading and correcting my many mistakes.

My sincere thanks go to all staff of the Faculty of Mechanical Engineering and Faculty of Manufacturing Engineering ,UMP, who were helped me in many ways and made my stay at UMP pleasant and unforgettable. Many special thanks go to my fellow friends for their excellent cooperation, inspirations and supports during this study.

I acknowledge my sincere indebtedness and gratitude to my parents for their love, dream and sacrifice throughout my life. I cannot find the appropriate words that could properly describe my appreciation for their devotion, support and faith in my ability to attain my goals. Special thanks should be given to my fellow members. I would like to acknowledge their comments and suggestions, which was crucial for the successful completion of this study.

ABSTRACT

This thesis is focused on development of the control system for Automated Guided Vehicle (AGV). This thesis concentrates on developing the control system for AGV involve about how the AGV will operate, involve of the movement and loading & unloading mechanism. The objectives of this project are to develop the control system parts involving develop the electronic circuit system and computer program of the system. The wired guided navigation used to communicate the computer to the AGV to ensure the AGV work accordingly. Subsequently, this project needs to be fabricating each of the electronic components to become one complete circuit. All the computer programmings are building by using Code Blocks software, the compiler which is compatible with electronic components and Visual Basic software. Afterwards, Visual Basic 6 is used in this project to create user friendly interface which is better than C interface (Command Prompt). This thesis also includes the test result of control system involving test program and test circuit for the AGV before proceed to the real AGV control system.

ABSTRAK

Tesis ini memfokuskan tentang pembangunan sistem kawalan Kenderaan Panduan Automatik atau *Automated Guided Vehicle* (AGV). Tesis ini menumpukan tentang pembangunan system kawalan untuk AGV melibatkan bagaimana AGV ini beroperasi, melibatkan pergerakan dan mekanisma mengangkat dan menurunkan barang. Objektif projek ini ialah untuk membangunkan bahagian system kawalan melibatkan pembangunan litar elektronik dan system program computer. Konsep panduan wayar untuk kawalan digunakan untuk memastikan AGV ini beroperasi mengikut arahan yang betul dan tepat. Di samping itu, projek ini memerlukan pemasangan untuk setiap komponen elektronik untuk membentuk sebuah litar yang lengkap. Semua program computer untuk kawalan menggunakan perisian Code Block, perisian yang sesuai digunakan untuk komponen elektronik dan perisian Visual Basic. Tambahan, perisian Visual Basic 6 digunakan di dalam projek ini untuk membuat antara muka yang lebih mesra pengguna berbanding dengan menggunakan antara muka C. Tesis ini juga merangkumi keputusan ujian yang telah di jalankan melibatkan keputusan percubaan pertama dan litar ujian untuk AGV sebelum diteruskan kepada keputusan system kawalan AGV yang sebenar.

TABLE OF CONTENTS

		Page
SUPERVISOR’S DECLARATION		ii
STUDENT’S DECLARATION		iii
ACKNOWLEDGEMENTS		iv
ABSTRACT		v
ABSTRAK		vi
TABLE OF CONTENTS		vii
LIST OF TABLES		x
LIST OF FIGURES		xi
LIST OF SYMBOLS		xii
LIST OF ABBREVIATIONS		xiv
CHAPTER 1	INTRODUCTION	
1.1	Project Background	1
1.2	Automated Guided Vehicle (AGV)	1
1.3	Problem Statement	2
1.4	Objectives	3
1.5	Scope of Project	3

CHAPTER 2 LITERATURE REVIEW

2.1	Introduction	5
2.2	Material Handling Equipment (MHE)	5
	2.2.1 History of the Material Handling Equipment Technology	5
	2.2.2 AGV System	6
	2.2.3 Description of an AGV System	7
	2.2.4 Components of an AGV System	8
	2.2.5 AGV System Design	9
	2.2.6 Flow Path Types	11
	2.2.7 Language	15

CHAPTER 3 METHODOLOGY

3.1	Introduction	16
3.2	Project Flow Chart	16
3.3	Development of Control System for AGV	19
	3.3.1 Development of AGV Circuit	19
	3.3.2 Development of AGV Program	24

CHAPTER 4 RESULTS AND DISCUSSIONS

4.1	Introduction	29
4.2	AGV Wiring System	30
	4.2.1 AGV Main Circuit	30
	4.2.2 AGV Test Circuit	33
4.3	AGV Main Program	36
	4.3.1 AGV Flow Chart	36
	4.3.2 AGV Program Codes	38
	4.3.3 Calculation for Bits Value	41
	4.3.4 C Program Interface	44
	4.3.5 Visual Basic (VB6) Interface	46
4.4	Limitations	47

CHAPTER 5 CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion 48

5.2 Further Study Recommendations 49

REFERENCES 50

APPENDICES

A Gantt chart 51

B Code Blocks Setup 54

C Visual Basic 6 (VB 6) Setup 59

LIST OF TABLES

Table No.		Page
3.1	L298N pin functions	22
3.2	Parallel port bit value	25
4.1	Parallel port pin division	32
4.2	Absolute Maximum Rating	35

LIST OF FIGURE

Figure No		Page
1.1	Laser Cutting Machine Room	3
1.2	AGV working flow	4
2.1	AGV Basic Components	9
2.2	Traditional AGV flow path design	11
3.1	L298N (Multiwatt)	19
3.2	L298N block diagram	20
3.3	L298N pin connections	21
3.4	100 nF Capacitor	22
3.5	Parallel port Male and Female	23
3.6	Parallel port cable	23
3.7	AGV block diagram	24
3.8	Parallel port background	26
3.9	Code Blocks interface	27
3.10	Visual Basic interface	28
4.1	AGV circuit (Isometric view)	30
4.2	AGV circuit (Top view)	31
4.3	AGV circuit (Parallel port connections)	32
4.4	AGV test circuit	33
4.5	L298N circuit drawing	34
4.6	Power supply	35
4.7	AGV sample codes 1	39
4.8	AGV sample codes 2	40

4.9	C interface 1	44
4.10	C interface 2	44
4.11	C interface 3	45
4.12	C interface 4	45
4.13	VB interface 1	46

LIST OF SYMBOL

V_{ss}	Logic Supply Voltage
V_s	Power Supply Voltage
V_{ref}	Voltage Reference
V_{en}	Enable Voltage
V_i	Voltage Input
R_{SA}	External Resistor A
R_{SB}	External Resistor B
P_1	Input 1
P_2	Input 2
P_3	Input 3
M	Motor
I_o	Peak Output Current

LIST OF ABBREVIATIONS

AGV	Automated Guided Vehicle
MHE	Material Handling Equipment
MSDS	Material Safety Data Sheet
VB	Visual Basic
JIT	Just In Time
AS	Automatic Storage
RS	Retrieval System
IC	Integrated Circuit
AI	Artificial Intelligent
GND	Ground
CW	Clockwise
CCW	Counter-Clockwise

CHAPTER 1

INTRODUCTION

1.1 Project Background

This project is about to create and develop the Automated Guided Vehicle (AGV) based on three main parts which are design & prototype of AGV, development of control system for AGV and design & prototype of loading & unloading mechanism for AGV. This project involves of parts from sketching, drawing, measuring each dimension to the control system part which involves computing wiring system and software application and also the mechanism of loading & unloading system to ensure the AGV can run perfectly.

1.2 Automated Guided Vehicle (AGV)

AGV is one type of Material Handling Equipment (MHE) like conveyors, cranes & hoists, elevator & lifts, automatic storage & retrieval system and so on which are focuses on process of transferring something from one place to another places especially in industrial sector or industrial warehouse. Actually, the goals to maintain or improve product quality, reduce damage and provide protection of materials, promote safety and improve working condition, promote productivity, control inventory and so on.

For further information, AGV is a driverless vehicle capable of moving along predetermined paths and performing certain prescribed duties. Moreover, it's relatively non intelligent which can go where they are sent and perform very limited and simple activities.

Principally, the project is divided into major three areas as follows:

- I. **Design and prototype of AGV**
–Designing the AGV model then from selecting material process until fabricating process.
- II. **Development of Control System for AGV**
–Developing the control system involving wiring system and programming data as a command to run the AGV
- III. **Design and prototype of Loading Unloading Mechanism for AGV**
– Designing the loading and unloading parts and also planning the loading and unloading mechanism.

1.3 Problem Statement

This AGV is designed to operate in the isolated room in UMP FKM lab. Currently, the FKM laser cutting machine has located inside the room (**Figure 1.1**). The principle of laser cutting is to cut materials by melting. Therefore, when cutting an Acrylic using laser, it also will melt the Acrylic and produce fume that contain hazardous substances which can give bad effect to human health. According to the Material Safety Data Sheet (MSDS) of Acrylic, when this fume is being continuously exposed to human, it will cause irritation to skin, eye and respiratory system.



Figure 1.1: Laser Cutting Machine Room

1.4 Objectives

- I. To develop the control system for Automated Guided Vehicle (AGV).
- II. To develop AGV circuit and wiring system.
- III. To test the overall AGV system

1.5 Scope of Project

This project based on the application of the AGV control system where the flow of how the AGV will operate (movement and loading & unloading activities). To operate this AGV, there are consisting of two main parts which must achieve perfectly to ensure the AGV can work well. The first part is the AGV wiring system and the second part is the command software to guide the AGV movement and loading & unloading activities. Figure 1 below showed how the AGV is working.

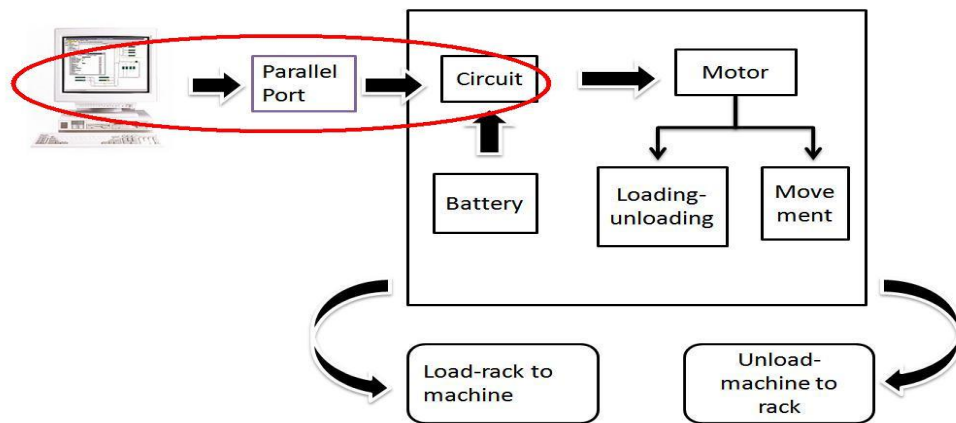


Figure 1.2: AGV working flow

The control systems are divided by 2 parts which are:

I. Wiring system

- a) Create the wiring system which involving electric and electronic components to ensure the AGV are working accordingly.
- b) To transfer control system from computer to the on-board driver by using parallel port.

II. Software application

There are two different soft ware that being used which are:

a) Code Blocks

- To create the body program of the whole operation.
- To control the tires movement and loading & unloading activities
- Code Blocks can be integrated with VB

b) Visual Basic (VB) 6.0

- To obtain user friendly interface.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

The aim of this chapter is to give the overview information about the Automated Guided Vehicle (AGV) which is part of the Material Handling Equipment (MHE) in the subject of its control system. In this chapter, the explanations and some of the MHE and the AGV control system histories, the previous research and findings, the theories are included. With reference from various sources as journal, thesis, references book, literature review has been carried out to collect information related to this project.

2.2 Material Handling Equipment-Automated Guided Vehicle

2.2.1 History of the Material Handling Equipment Technology

Material handling is defined by the Materials Handling Institute as the movement of bulk packaged and individual goods, as well as their in process and post process storage, by means of manual labor or machines within the boundaries of a facility. Although this field of study includes the handling of bulk (solid- or liquid-phase) material and individual goods, this chapter will only focus on the latter (i.e., “unit loads”), with a primary emphasis on material handling equipment, as opposed to facility planning and movement control.

Material handling does not add value to the product but only cost. Thus the objective of material handling is the efficient movement of goods for the on-time delivery of correct parts in exact quantities to desired locations in order to minimize associated handling costs. It is not uncommon to have parts/subassemblies moving around a plant several kilometers prior to their shipment. Manufacturing plants must therefore eliminate all unnecessary part movements, as well as in process inventories, for just-in-time (JIT) production. Material handling equipment can be classified according to the movement mode: above-floor transportation (e.g., belt conveyors, trucks, etc.), on-floor transportation (e.g., chain conveyors), and overhead transportation (e.g., cranes). In the following sections, we will review industrial trucks (including automated guided vehicles), conveyors, and industrial robots as the primary mechanized/automated material handling equipment. We will also briefly review the automated storage and retrieval of goods in high density warehouses, as well as the important issue of automatic part identification (including bar codes). The chapter will be concluded with a discussion on automobile assembly.

2.2.2 Automated Guided Vehicle (AGV) System

Material handling is an important aspect of any production system. Material handling system have been prevalent since the beginning of mass production, either as manual system, mechanical system (forklifts, conveyors), or in more recent years as fully automated system (automated guided vehicle, (AGV), automatic storage and retrieval system (AS/ RS) etc). Technological advances and the need for flexibility and reliability have increased focused on automated material handling systems. The use of AS/RS and AGV systems are becoming commonplace in today's industries. AGVs have become increasingly popular as a means of horizontal material handling transportation system. They are used wherever there is a need for an autonomous transportation system. AGVs are particularly useful where products need to be handled carefully or the environment is potentially dangerous to humans. Examples include handling of telecommunication products, IC chips, voltage cables and radioactive materials. In the automotive manufacturing industry, AGVs have been combined with robots to perform welding and painting operations.

2.2.3 Description of an AGV System

An AGV system is an advanced material handling system that involves one or more driverless vehicles each following a guide path and controlled by an off-board computer or microprocessor. AGV are typically used to carry unit loads in production and assembly operations. The advantages of AGVs include reliable, automatic operation, flexibility in adapting to changes in material flow, improved positioning accuracy, reduced handling damage, easily expandable layout and system capacity, and automated interfaces with other system. An AGV system allows automation of a certain portion of material handling and thereby, a reduction in the labor force. It also results in an increase in the efficiency of the material handling operation, resulting in better utilization of the work force and processing equipment. An AGV based material handling system also supports various tires of production systems and improves productivity.

The efficient material handling system also helps in reducing mistakes and improving quality. The improved system results in increased worker satisfaction as it is possible to change the material arrival rate to suit the workers pace. The main advantage that a discrete material handling system such as an AGV system offers is real-time control of material handling. This helps in identification of the parts, the routes they travel and the vehicles they travel in, resulting in a lower WIP inventory, reduced tardiness, lower inventory costs and better response to demands (Hammond (1986)). An AGV system also offers other benefits such as reduction in space requirements. Unlike conveyors or other material handling systems, AGVs are small in size and only move along the aisles. They minimize product damage and help in housekeeping. Changes in the layout and relocation of the material handling system are also much easier to accomplish when using AGVs. They also are combined with other existing material handling systems and offer flexibility.

The main disadvantage of an AGV based material handling system is its expense. The high cost of the control software used and the number of vehicles required in a system curtail the wide usage of AGVs as material handling systems. A trade-off analysis between the initial setup cost of an AGV system and the savings involved is necessary before installation. Other limitations of an AGV include necessity for polished floor surfaces for smooth operation of the AGVs, guide path bed stability problems and restrictions such as height of metal floors that must be crossed and weather conditions that it can operate under when used outside the manufacturing plant. Obstructions in the facility layout and ramp gradients are other obstacles that need to be overcome when designing the guide path for the AGVs. Other issues that need to be considered when installing an AGV system are management support, worker attitudes towards the new system, maintenance problems and requirements. It can be seen from above that though the AGV has a number of benefits it also has its share of disadvantages and may not be applicable in all cases.

2.2.4 Components of an AGV System

The different components of an AGV system are listed as below

- 1) **Vehicles** – The vehicle or the AGV consists of the frame, batteries, on-board charging unit, and electrical system, drive unit, steering, precision stop unit, communication unit, safety system and work platform. The components mentioned above can each be further classified into different categories based on their capabilities and features. The application for which the AGV is used dictates the type of component that is to be used.

- 2) **Guide path and guidance systems** – Most AGVs need a guide path to follow. The guide path techniques used are known as passive or active tracking. Passive tracking occurs when optical or metal detection principles (wireless) are used for vehicle guidance whereas active tracking involves inductive principles (for example, guide wire is used to help tracking).

- 3) **Floor and system controls** – The controller is the brain of the whole system, tying the vehicle and the guide path together and integrating the system. The AGVs contains three levels of control architecture: vehicle control system, floor control unit and vehicle on-board processor. These control systems take care of the different tasks such as lane selection, carrier selection, guide path frequency generation, blocking between vehicles, automatic routing, controlling speeds, displaying job information, monitoring floor equipment status, tracking loads and so on.

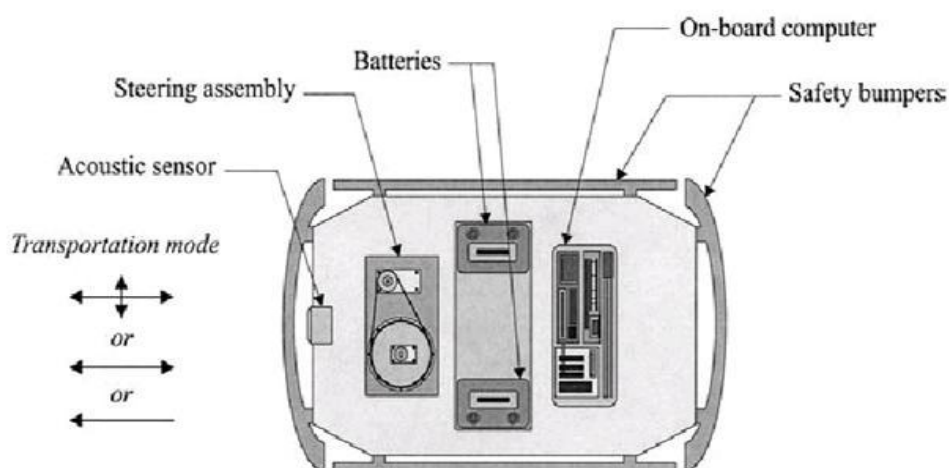


Figure 2.1: AGV Basic Components

Source: James M, 1977.

2.2.5 AGV System Design

The only current literature review on the Automated Guided Vehicle (AGV) flow path design problem is by Sinriech (1995). The general network design models for discrete material flow systems are reviewed. It concentrates more on the aspects of AGV flow path and reviewed individual papers very briefly. It also does not offer a classification scheme for the AGV flow path literature. However, the paper discusses various parameters involved in a material handling flow system and the different approaches that have been used to solve this problem. Peter, et al (1995) presents a control classification scheme for AGVs.

They present a nice classification scheme for an AGV system in general, but concentrate more on the control aspects of the problem. They present the classification scheme with three basic level namely – guide path determination, vehicle capacity and vehicle addressing mechanism.

The paper does not concentrate as much on each of the individual levels, but gives a brief overview of each level and sublevel. It presents a cubic structure (based on the levels and sub levels) which partitions the AGV system into 12 different classes. Depending on the functionality requirement and the sublevel chosen, this structure helps identify the relative complexity involved in designing the required AGV system.

Vosniakos and Mamalis (1990) discussed the issues involved in an AGV system design with respect to flexible manufacturing system applications. An overview of the different aspects of an AGV is presented with emphasis on route control and collision avoidance. Docking, load transfer, traffic control, communication between the controller and the vehicle, AGV management policies, evaluation of the control policies and various other aspects that have to be considered before setting up an AGV based system are also discussed.

One of the more important areas in an AGV design is the guide path and guidance system. The area of interest in the guidance system and guide path is the guide path layout. There has been little research in the area of guide path layout. The research done in area of guide path layout in AGV systems can be explained better using the classification scheme presented in Rajagopalan and Heragu (1997).

2.2.6 Flow Path Type

The type of flow path in an AGV system is its most important characteristic. The flow path for any AGV system dictates how the AGV will travel between the different pickup/drop-off (p/d) points. The classification is based on the type of flow path used.

2.2.6.1 Traditional Layout

Maxwell and Muckstadt (1982) first recognized the importance of AGV based material handling system design. They developed a model that determines the maximum number of AGVs needed to efficiently transfer material from one facility to another. The problem was solved assuming the guide path was already installed and the best route had to be determined. The objective under consideration was to minimize total travel time. Figure 2.1 shows a traditional AGV flow path design.

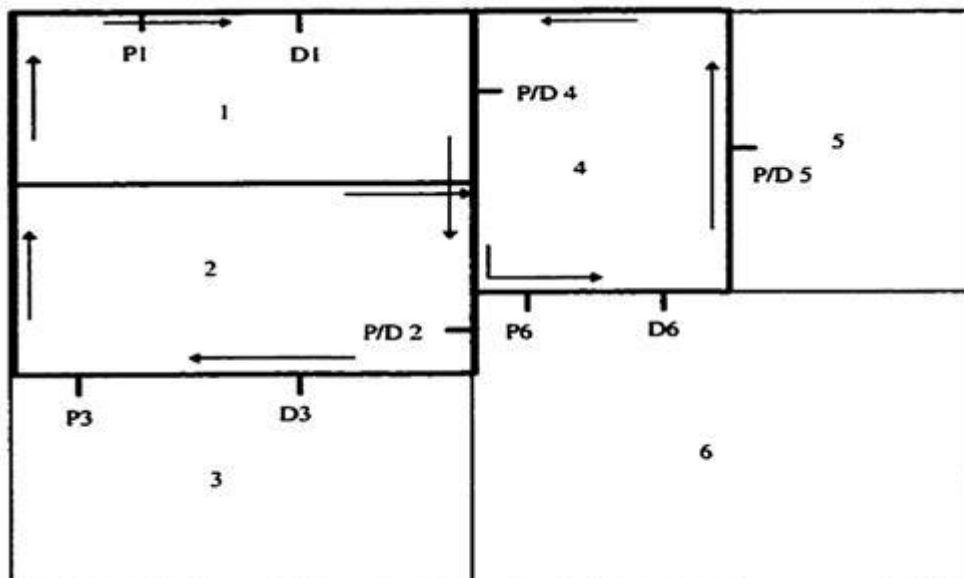


Figure 2.2: Traditional AGV flow path design

Source: James M, 1977.

Maxwell and Wilson (1981) had developed a dynamic network flow model earlier to analyze the effects of blocking in a fixed path system. A traditional flow path design involved determining the best path that connected all the given p/d points.

Gaskin and Tanchoco (1987) presented a binary integer model to determine the optimal flow path for an AGV system. They only considered the movement of loaded vehicles unlike Maxwell and Muckstadt (1982). They did not make any assumptions about the flow path except that movement was restricted to certain areas (such as the aisles), nor did they discuss any generic solution method to the model. An example is solved to illustrate the approach used, but this approach cannot be generalized.

Kaspi and Tanchoco (1990) considered a unidirectional AGV system and solved the optimal flow path design problem using a branch and bound technique. The formulation is the same as that presented by Gaskins and Tanchoco (1987). They formulate the problem as a graph network (node – arc network) with the pick-up/drop-off points as nodes and feasible guide paths as arcs. They also addressed the reach ability problem which tackles the situation where a group of nodes might end up as sink nodes.

Sinriech and Tanchoco (1991) developed a formulation to solve the traditional AGV flow path problem using a graph theory approach. They use the same node – arc formulation as Kaspi and Tanchoco (1990) and use the model to give directions to the undirected graph network. They then use an improved branch and bound technique to solve the problem. They consider both loaded and unloaded travel time in this formulation. Gaskins and Tanchoco (1989) developed a model to solve the virtual flow path problem. The virtual flow path arises in cases where the AGV guide path does not exist in reality. The AGV is guided by the controller without the need for a physical guide path.

They formulate the problem as a multi commodity flow problem where the material to be transferred is substituted for people and unit loads. This results in an integer model based on the multi-commodity flow problem. Instead of Euclidean or rectilinear distance, they consider a path distance which takes into account the fact that rectilinear distances and Euclidean distances may not always be applicable when considering distance between two facilities.

A path distance is the actual distance taken by an AGV to travel from one point to another. Goetz and Tanchoco (1989) and came up with an algorithm to solve layout design problem. The objective of this model is to minimize the total distance traveled. They reduce the problem size to be solved using a heuristic and the new reduced problem is used to determine the p/d points. Their heuristic determines the major flows into and out of each department and uses this as a base to prioritize the departments. It assumes that the flow data between departments is already known. Rectilinear distances are used and the flow is assumed to be between the departments centroids. It also uses the fact that in case of unidirectional AGV flow path design only four paths need be examined when considering the route between any pick up and drop off point. A linear programming model obtained after simplifying certain nonlinear terms is presented in this paper. For larger problem sizes this model may be difficult to solve.

Architecture's modules will contain sensor and actuator systems of a type relevant to the function of the module. Example modules may be Navigators, Manipulators, Vision Processing, etc. and each may contain any number of processor, sensor and actuator systems. The definition of a module is such that it must contain the following:

- a) An interface to the bus.
- b) The necessary logic to calculate all tasks assigned to that module.
- c) The hardware for controlling all the input and output devices the module needs to perform its calculated tasks.
- d) All the actual input and output devices needed by the module.

To illustrate each of these points, figure two shows an example of a navigator module. It satisfies the above conditions in the following ways:

- I. A bus connection and its control via a master processor.
- II. A master processor to decode bus master instructions and to handle the path planning for the robot.
- III. Two less powerful processors (slaved to the master processor) and their support devices, one controlling the module's range sensors for path planning information, and the other for controlling the path following of the robot by controlling the speed of the robots wheels and compensating for any motor drift that may take the robot off course. The master processor will instruct both these processors as required in order to execute an instruction passed to the module from the bus master.
- IV. The actual wheels and sensors themselves (Possibly already attached to some form of chassis or trolley). Two special forms of module are the Standard Input/output module and the Reflex module. The standard input/output module has two distinct functions in a robot's architecture. Firstly, it will provide the bus master with a program for the robot to follow, and secondly, it will handle all user input functions and display all system output (if any). Only in very limited architectures (i.e. those whose program will never change and require no other forms of input) will the standard input/output be absent and the bus master default to an internal program. The form of a standard input/output module may be varied. For example it may be a small portable terminal on a mobile robot or an actual terminal mounted next to a stationary one (or perhaps a large terminal radio linked to a mobile bus on an A.G.V.). There is no reason why a standard input/output module may not be an A.I. module that monitors the actions of the robot and generates its own program for the robot to follow thus making the robot an independent being.

Reflex modules can be added to a robots system to monitor the robots internal conditions and external environment in order to warn the bus master of impending danger (e.g. the robot is moving towards something very hot, the ground in front of the robot falls rapidly away, or both the robot's wheels go to full speed despite no (commanded) activity from the navigator module. The reflex modules have exclusive use of specific interrupt lines on the system bus and may instruct the bus master directly if they are of a complex enough form.

2.2.7 Language

The language provided for the control of the bus master (and therefore the robot) is quite unusual as it provides low, high and very high level commands. The language definition itself only commons the low and high level commands while the very high level commands have to be added to the language by means of include and import files. When a module is acquired or constructed these files must be proved as well, containing all the commands that the module understands in the form of a command-code list for the include files and in the form of a procedure (probably containing bus access commands) written in the language in the case of import files. This feature of the language will allow its command set to increase as new developments are made and also provide ease of use for a purchased module as the operator will only have to know the command list the module understands rather than any complex codes that it needs to be passed. Therefore to use a module on a new system it would only be necessary to plug it in and add the relevant commands to any program at the standard input/output terminal.

CHAPTER 3

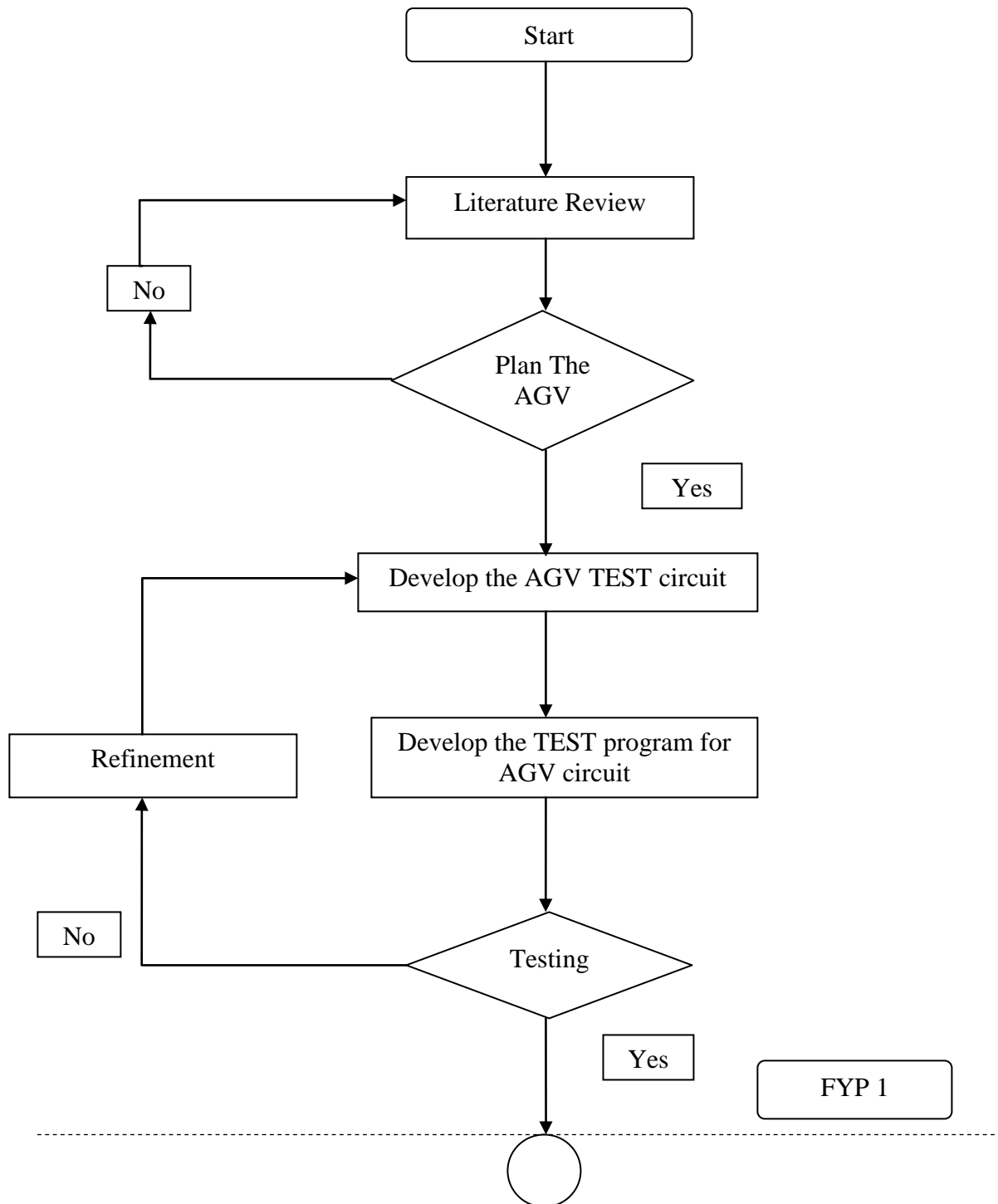
METHODOLOGY

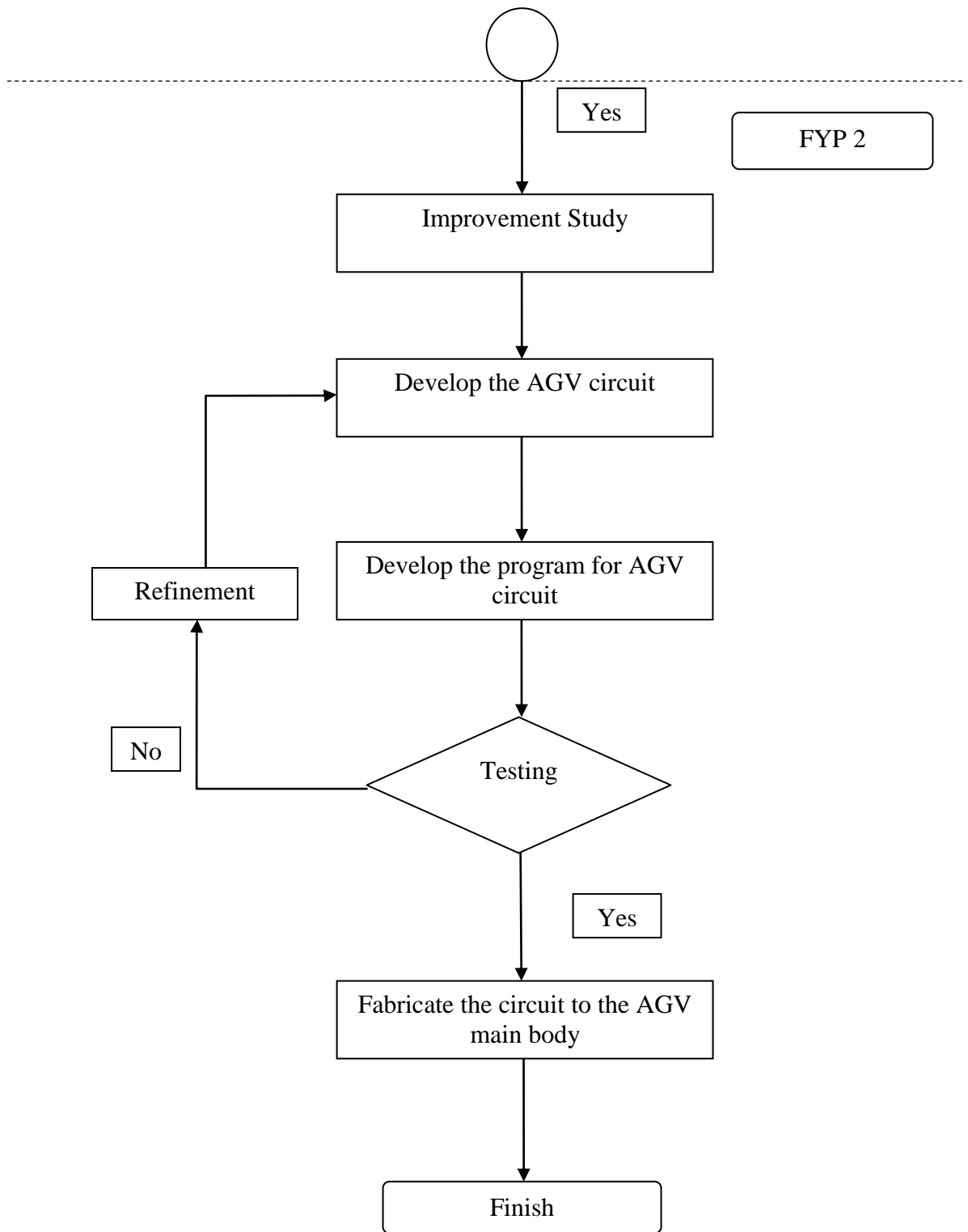
3.1 Introduction

Generally, this project involved in development of the program, creating the circuit for AGV where actually the main important point is to ensure the AGV work accordingly. This chapter will explain the overall flow for this project involves the flow for the AGV operation. The part of controls system will determine whether the AGV can operate without any problem and how to control the AGV.

3.2 Flow Chart

To achieve the objectives of project, a methodology were constructs base on the scopes of projects as a guiding principle to formulate this project successfully. The important of this project is to develop the circuit and the program. Therefore to achieve the objectives of this project where is to develop the control system for Automated Guided Vehicle (AGV). A terminology of works and plans show in the flow chart. This is very important because to make sure the goal or what purpose this project is not wrong.

PROJECT FLOW CHART



3.3 Development of Control System

3.3.1 Development of AGV Circuit

3.3.1.1 Electronic Components

In this project, the driver used is Dual Full-Bridge Driver type L298N (Multiwatt) which is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver. Two enable inputs are provided to enable or disable the device independently of the input signals. **Figure 3.1** shows the real Dual Full-Bridge Driver type L298N (Multiwatt)



Figure 3.1 : L298N (Multiwatt)

Source : Thomson Microelectronics, 2009

BLOCK DIAGRAM

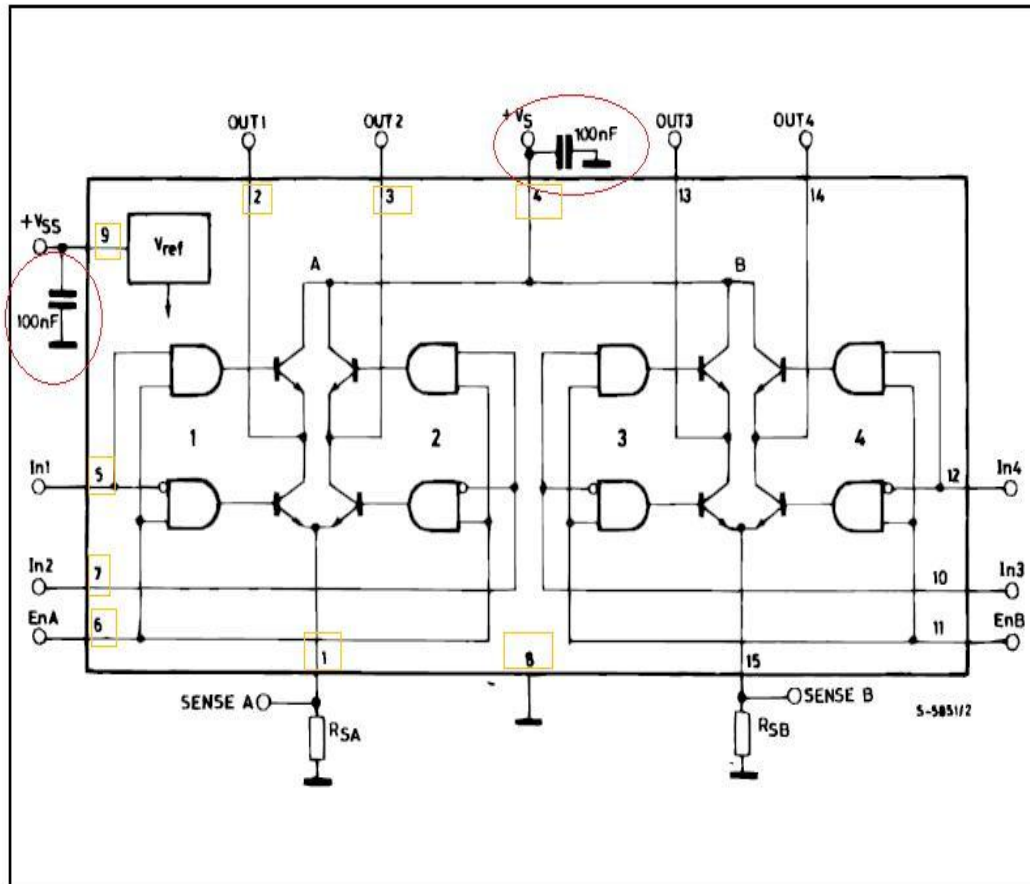


Figure 3.2 : L298N block diagram

Source : Thomson Microelectronics, 2009

The L298 integrates two power output stages (A;B). The power output stage is a bridge configuration and its outputs can drive an inductive load in common or differential mode, depending on the state of the inputs. The current that flows through the load comes out from the bridge at the sense output: an external resistor (R_{SA} ; R_{SB}) allows to detect the intensity of this current. (Refer to **Figure 3.2**)

Figure 3.3 shows the pin connections (top view) for each pins. This pin connection show how this driver operates. In this project, there were only nine pins used as shown highlighted in the **Figure 3.3**.

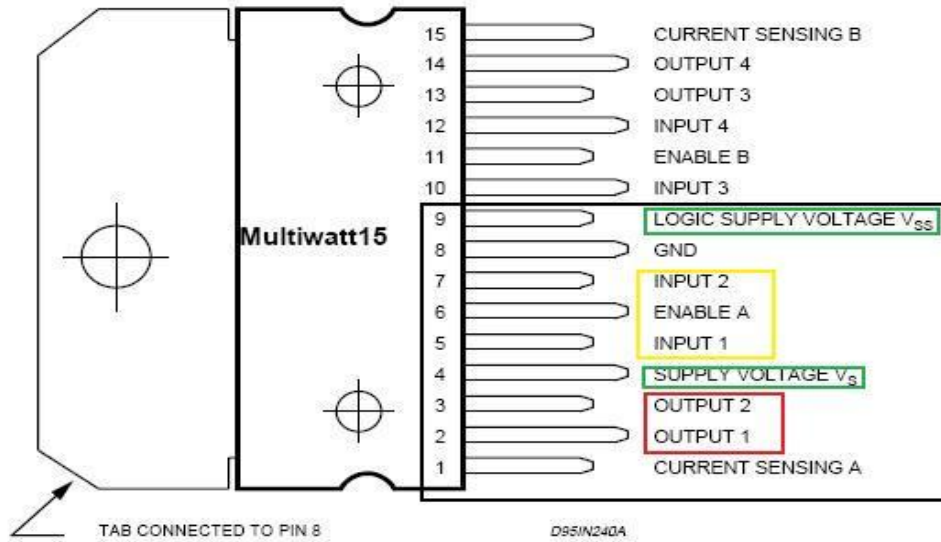


Figure 3.3 : L298N pin connections

Source : Thomson Microelectronics, 2009

Table 3.1 shows the overall function for each pin for this driver. The pin number 5, 6 and 7 are the pins that connected to the parallel port pins while the output pins 2 and 3 are the pins connected to the motor. This driver will interpret the data sent from the computer to transfer the data become the movement of motor.

Table 3.1 : L298N pin functions

PIN FUNCTIONS (refer to the block diagram)

MW.15	Name	Function
1;15	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	Out 1; Out 2	Outputs of the Bridge A; the current that flows through the load connected between these two pins is monitored at pin 1.
4	V _S	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	Enable A; Enable B	TTL Compatible Enable Input: the L state disables the bridge A (enable A) and/or the bridge B (enable B).
8	GND	Ground.
9	V _{SS}	Supply Voltage for the Logic Blocks. A 100nF capacitor must be connected between this pin and ground.
10; 12	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	Out 3; Out 4	Outputs of the Bridge B. The current that flows through the load connected between these two pins is monitored at pin 15.
-	N.C.	Not Connected

Source : Thomson Microelectronics, 2009

Figure 3.4 shown the 100nF capacitor. Usually, a non inductive capacitor used and must be foreseen between both V_S and V_{SS} to ground, as near as possible to GND pin. When the larger capacitor of the power supply is too far from the Integrated Circuit (IC), a second smaller one must be near the L298.

**Figure 3.4 : 100 nF Capacitor**

3.3.1.2 Parallel Port Cable

Parallel port is used in this project to connect the computer to the circuit. The programming for parallel will discussed in next subchapter. **Figure 3.5** shown the parallel port (Male & Female) and **Figure 3.6** shown the parallel port cable used in this project.



Figure 3.5 : Parallel port Male and Female



Figure 3.6 : Parallel port cable

3.3.2 Development of AGV Program

The overall AGV control system which involves the computer connected to the on-board circuit involving driver to conduct the movement of motor (for tires and loading & unloading part).

Figure 3.7 shows the block diagram for the overall AGV control systems.

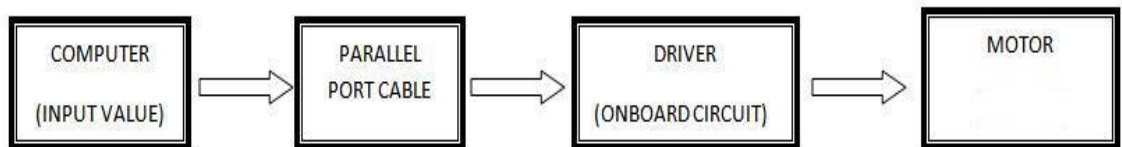


Figure 3.7 : AGV block diagram

In this project, the program for AGV is created by using Borland C++ software which is using the c language to give command for entire AGV system. This because the c language can integrated well with driver used and will avoid the driver dysfunction (if the driver still not damage). The body program is very important in this part because it will represent all the AGV activities. It is the command for entire AGV activities start from the movement of tires until the loading & unloading activities. The program will control the driver in on-board circuit which will decide to rotate the motor whether to rotate clockwise or anticlockwise (represent the forward and backward movement for tires and up & down for loading & unloading). The program starts when user/operator chooses to run the AGV by the following steps below.

3.3.2.1 Parallel Port Programming

Parallel port cable was used to connect the computer to the on-board circuit. It used the application of 'bit value' as the input which will represent each pin used in parallel port.

Table 3.2 below shows the bit number for each pin in parallel port.

Table 3.2 : Parallel port bit value

Pin Number	Pin9	Pin8	Pin7	Pin6	Pin5	Pin4	Pin3	Pin2
Bit Designation	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Bit Value	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Decimal Equivalent value	128	64	32	16	8	4	2	1

Source : Dr Wan Azhar Wan Yusoff, 2006.

Each pin in parallel port has different bit number and the sum of certain bit number will never equal with others. All these value represent for one pin. These bit value were used in this project as the input value to the L298N driven then controlled the rotation of the motors.

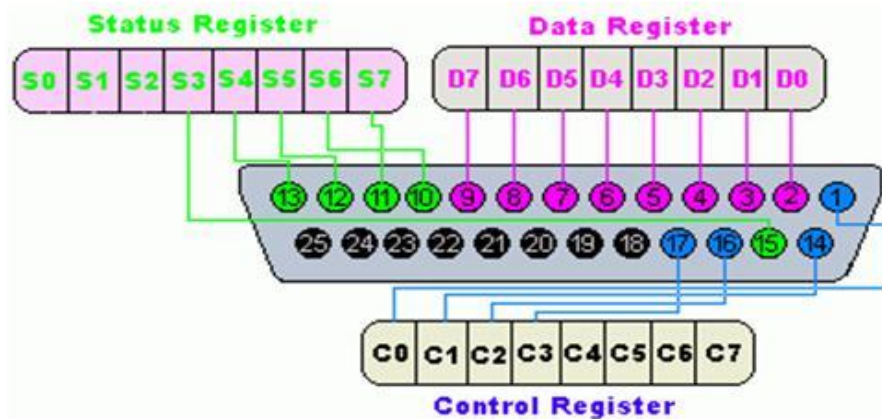


Figure 3.8 : Parallel port background

Source : Dr Wan Azhar Wan Yusoff, 2006.

Figure 3.8 shows the schematic diagram of the PC parallel port connector. There were 25 pins all together. The 25-pin represents four main functions:

1. **First Output Port** – the pins from Pin2 to Pin 9. It is called **data register**. It was used for sending data on the parallel port's data lines. The memory address for this port is 888.
2. **Input Port** – the pins are Pin10, Pin11, Pin12, Pin13 and Pin15. It is called **status register**. It was used to read the status of signals given by peripheral device to parallel port. The memory address for this port is 889.
3. **Second Output Port** – Pin1, Pin14, Pin16 and Pin17. These pins are also output port and are called control register. This is an extra port in case controlling through our first port is not sufficient. It was used to give control signals to peripheral device.. The memory address for this port is 890.
4. Ground pins which are Pin18 to Pin 25. These are PC ground.

But, only used the Data Register port because this port used for sending data on the parallel port data lines while the Status Register port used to read the status of signals given by peripheral device to parallel port or called ‘read only register’ and for Control Register port used to give control signals to peripheral device or called ‘write only register’.

3.3.2.2 Code Blocks Software

Code Blocks software has been used in this project as the compiler for the AGV program. It has been used to create the program for AGV consists of movement of AGV tires and also loading and unloading mechanism. This compiler can integrate well with L298N driver (Also other drivers and microcontrollers) also compatible with Visual Basic 6.0 (VB).

Figure 3.9 shows the Code Blocks interface. Steps for setup the Code Blocks compiler refer **Appendix B**.

```

switchc - Code::Blocks 8.02
File Edit View Search Project Build Debug wxSmith Tools Plugins Settings Help

1 #include <iopp.h>
2 #include <stdio.h>
3 #include <conio.h>
4
5
6 int main()
7 {
8     HANDLE StdOut = GetStdHandle(STD_OUTPUT_HANDLE);
9     LoadIODLL(); /* Open parallel port */
10
11     int type;
12     w: printf("\nOPTION\n\n");
13     printf("\n1-Movement\n");
14     printf("\n2-Loading/Unloading\n");
15     printf("\n3-Suction\n");
16     printf("\n4-Exit\n");
17
18     printf("\nplease enter your option\n");
19     scanf("%d",&type);
20
21     switch (type)
22     {
23     {
24         case 1:goto x;
25         case 2:goto y;
26         case 3:goto z;
27         case 4:goto m;
28         default: printf("\nInvalid selection. Please try again! \n");
29     }
30 }
31
32
33 int movement;

```

Figure 3.9: Code Blocks interface

3.3.2.3 Visual Basic 6.0 (VB 6)

Visual Basic has been used in this project to obtain user friendly interface compared to Code Blocks interface. Using VB6 interface was made the command to control each mechanism (Movement and Loading and Unloading) look attractive and easy to operate. **Figure 3.10** shows VB6 interface. Steps for setup the VB6 interface refer **Appendix C**.

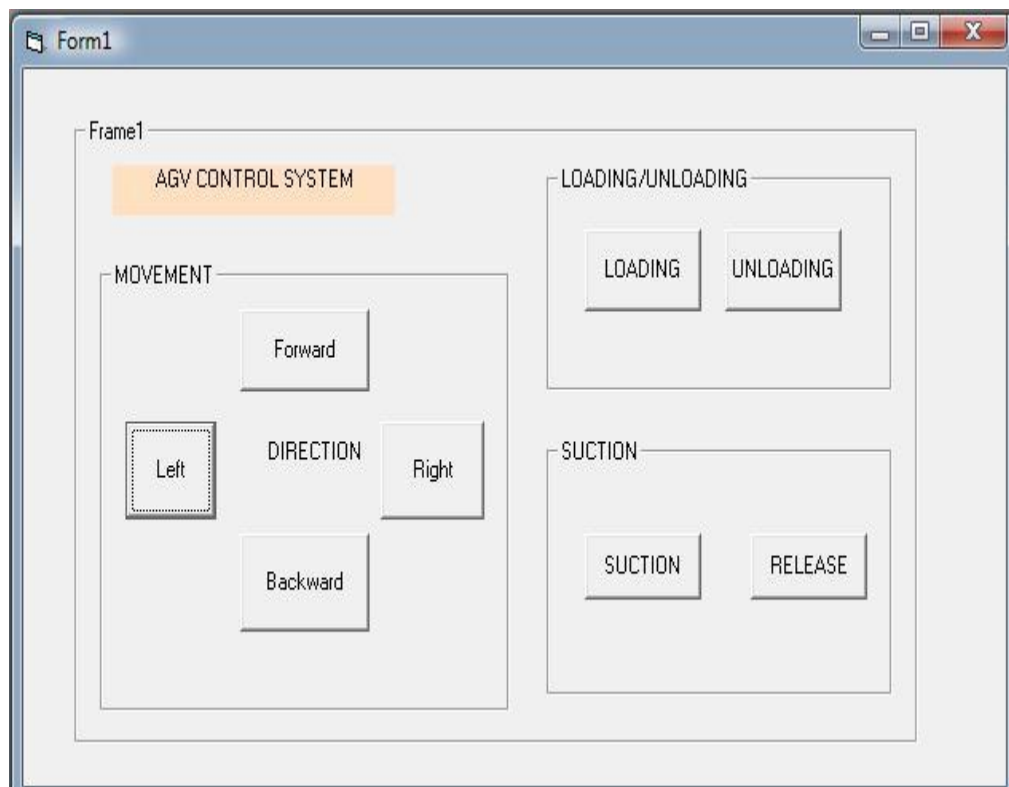


Figure 3.10: Visual Basic interface

CHAPTER 4

RESULTS AND DISCUSSIONS

4.1 Introduction

This chapter discusses about the overall result from the project which contains about Automated Guided Vehicle (AGV) circuit development, program development and testing for control system of the AGV. From this project, the circuit development consists of combination usage of parallel port application and the function of Dual Full Bridge Driver l298N to ensure the AGV can be operated accordingly.

For the program development, there were consisted about AGV test program. This shows about how the bit value has been applied in the program to control the rotation of motor for tires and loading and unloading parts. The real program has been developed to ensure all the motors for tires and loading and unloading parts which are adopted from the test program. It was contained the overall step to setting up the movement of tires and the movement of loading and unloading parts. In this part also discusses about the testing for the AGV from the beginning of loading acrylics to movement until it was unload the acrylics.

4.2 AGV Wiring System

4.2.1 AGV Main Circuit

Figure 4.1 below shows the AGV main circuit which were consisted all combinations of all L298N drivers to became one complete circuit. This AGV complete circuit was consisted of four L298N drivers, two 100nF non inductive capacitor, connecting wires and the important parts were about the parallel port pins connections. There were nine pins port connections have been used to control these four L298N drivers. In nine pins port connections, there was one pin port connected to Ground (GND), and others eight were connected to the input of L298N drivers.

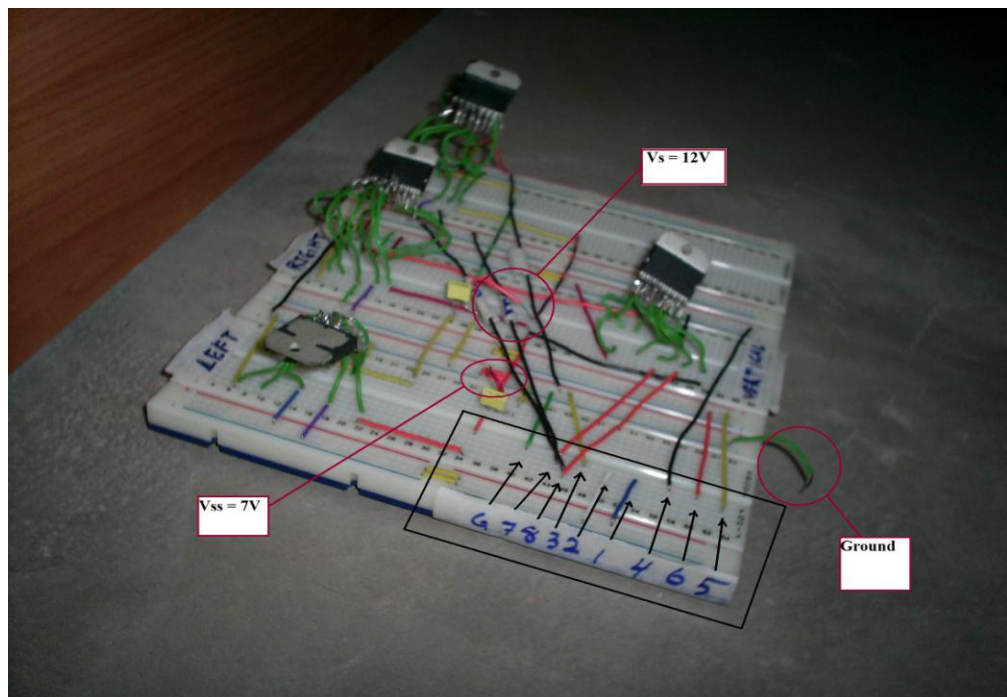


Figure 4.1 : AGV circuit (Isometric view)

There were four actions represent the usages of four L298N drivers. It was divided into four specific tasks for each driver which were to control the AGV motors to turn left, turn right (For movement of tires), vertical bar movement for loading and unloading motor and diaphragm pump for suction of acrylics. In **Figure 4.2** below shows division of each task for each L298N drivers. The two yellow lines shows the input voltage for Logic Supply Voltage, V_{SS} and Supply Voltage, V_S (Refer **Table 4b**)

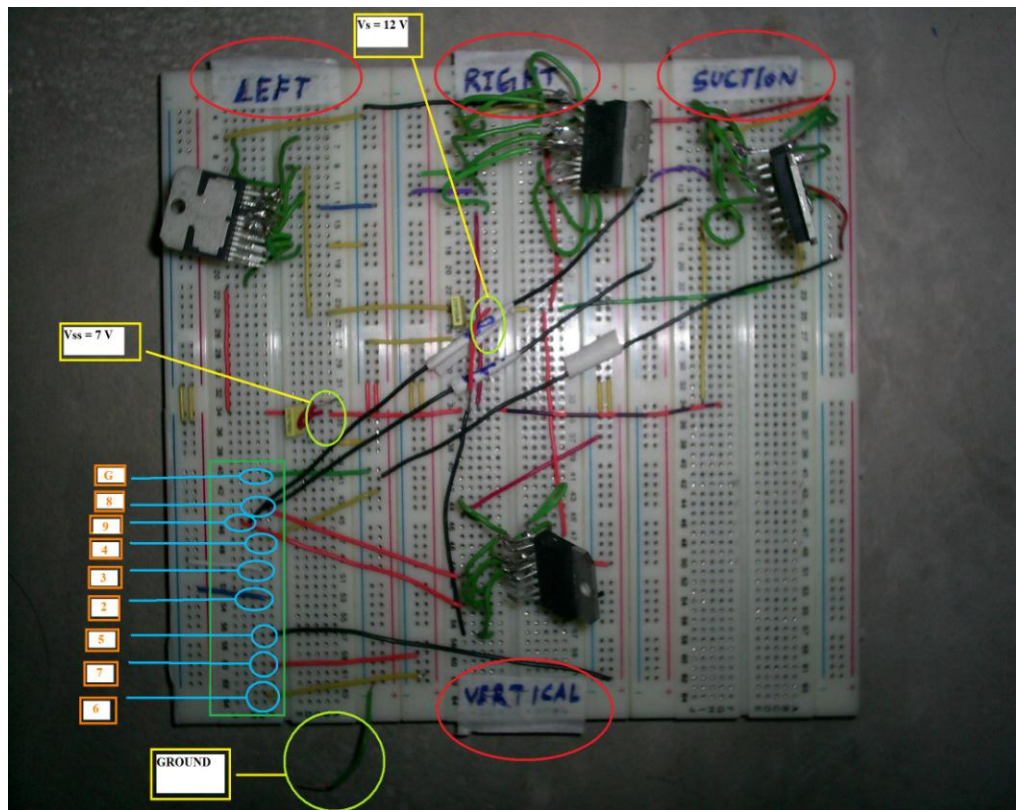


Figure 4.2: AGV circuit (Top view)

Figure 4.3 below shows the parallel port pins connection from the computer to the AGV main circuit. The numbers in the figure represent the port that has been used. This number was very important because each number represents bit designation and has each bit value. These bit values were used in the AGV program to control whether motor rotation clockwise or counter-clockwise. From this project, to control one L298N driver needs three pins of parallel port to operate it.

Table 4.1 below shows division of parallel port pins to each L298N drivers.

Table 4.1: Parallel port pin division

PARALLEL PORT PIN	TASK FOR DRIVER (L298N)
2,3,4	Right Tire Motor
5,6,7	Left Tire Motor
5,8,9	Vertical Motor(Loading & Unloading)
4,8,9	Diaphragm pump (Suction Process)

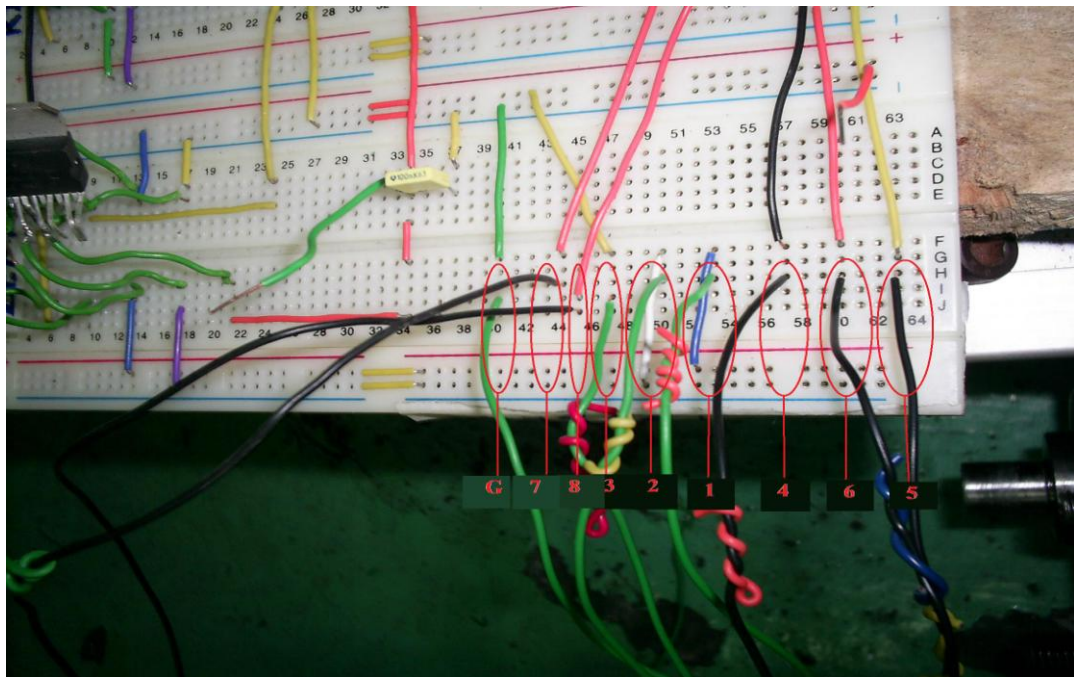


Figure 4.3: AGV circuit (Parallel port connections)

4.2.2 AGV Test Circuit

The test circuit was built by combine the circuit with the motor. In this project, power window motors are used to test whether the input from the computer to the L298N driver will give the output, rotation of the power window motor.

Figure 4.4 shows the test circuit to ensure the driver and motor can work well. After created the test circuit, it has been developed to became the real main circuit which were consisted all drivers for all tasks. Voltage cables red and black are the input voltage for V_{SS} and V_S . It were connected to the power supply by using two channel, one for input voltage for V_{SS} and other one for V_S .



Figure 4.4: AGV test circuit

Figure 4.5 - L298 Circuit Drawing

P1 = Input 1

P2 = Input2

P3 = Enable Input

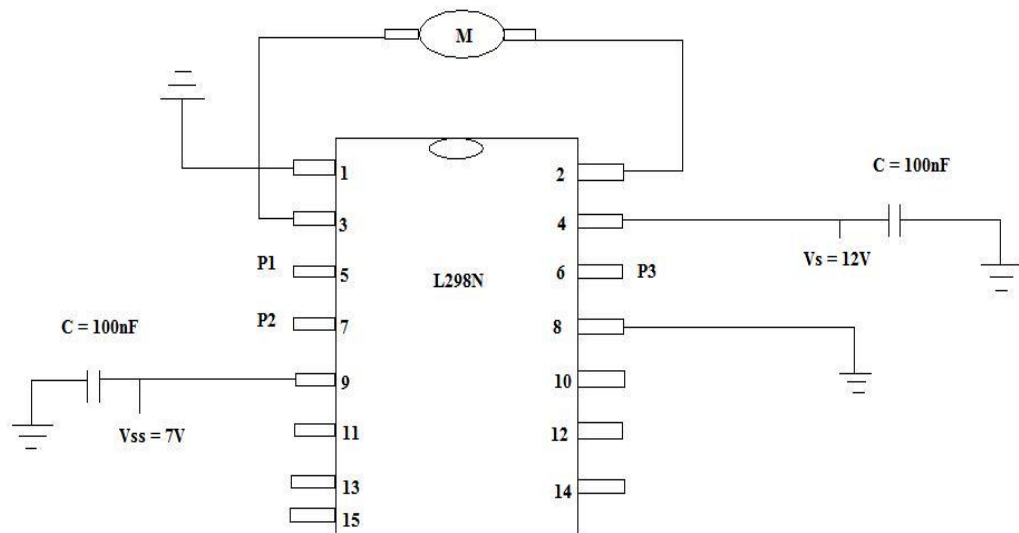


Figure 4.5: L298N circuit drawing

Figure 4.6-Power supply used to give input for V_{SS} and V_S . Value of voltage for V_S used was 12V because of the ideal voltage for power window voltage. This voltage can be increased up to 50 V (Refer **Table 4.2**). Value for V_{SS} was fixed, only 7V necessarily (Refer **Table 4.2**)

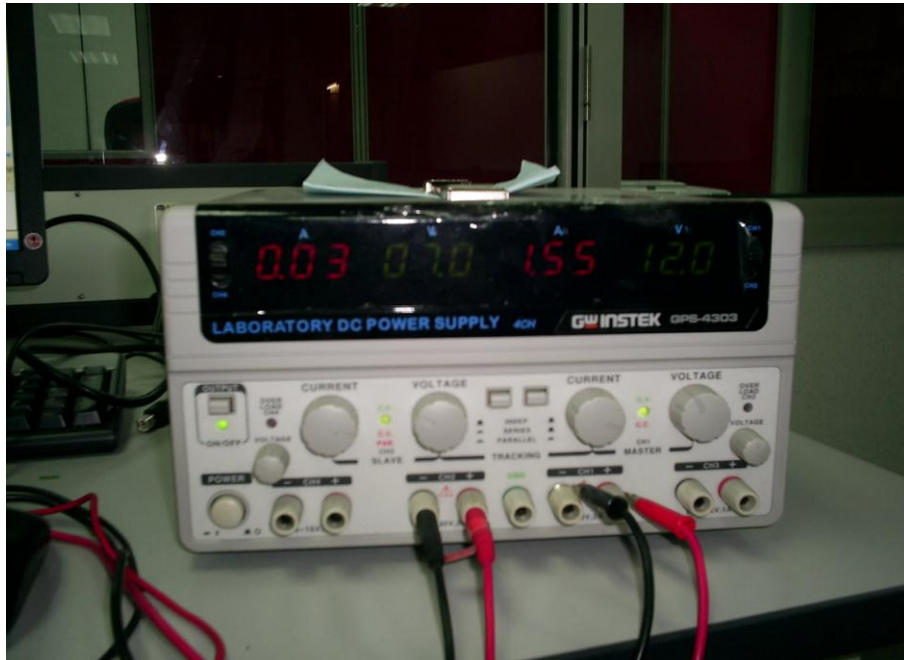


Figure 4.6: Power supply

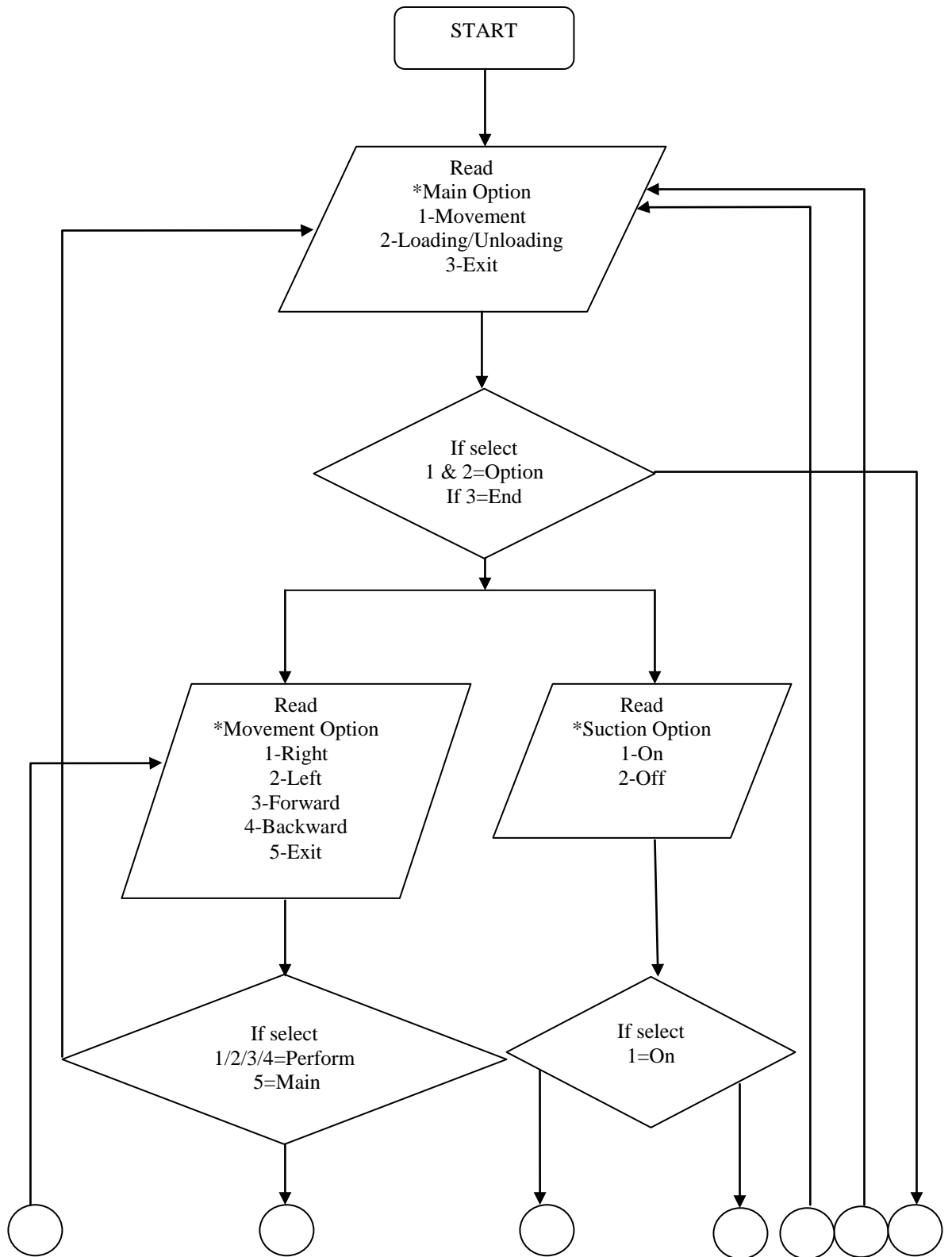
Table 4.2: Absolute Maximum Rating

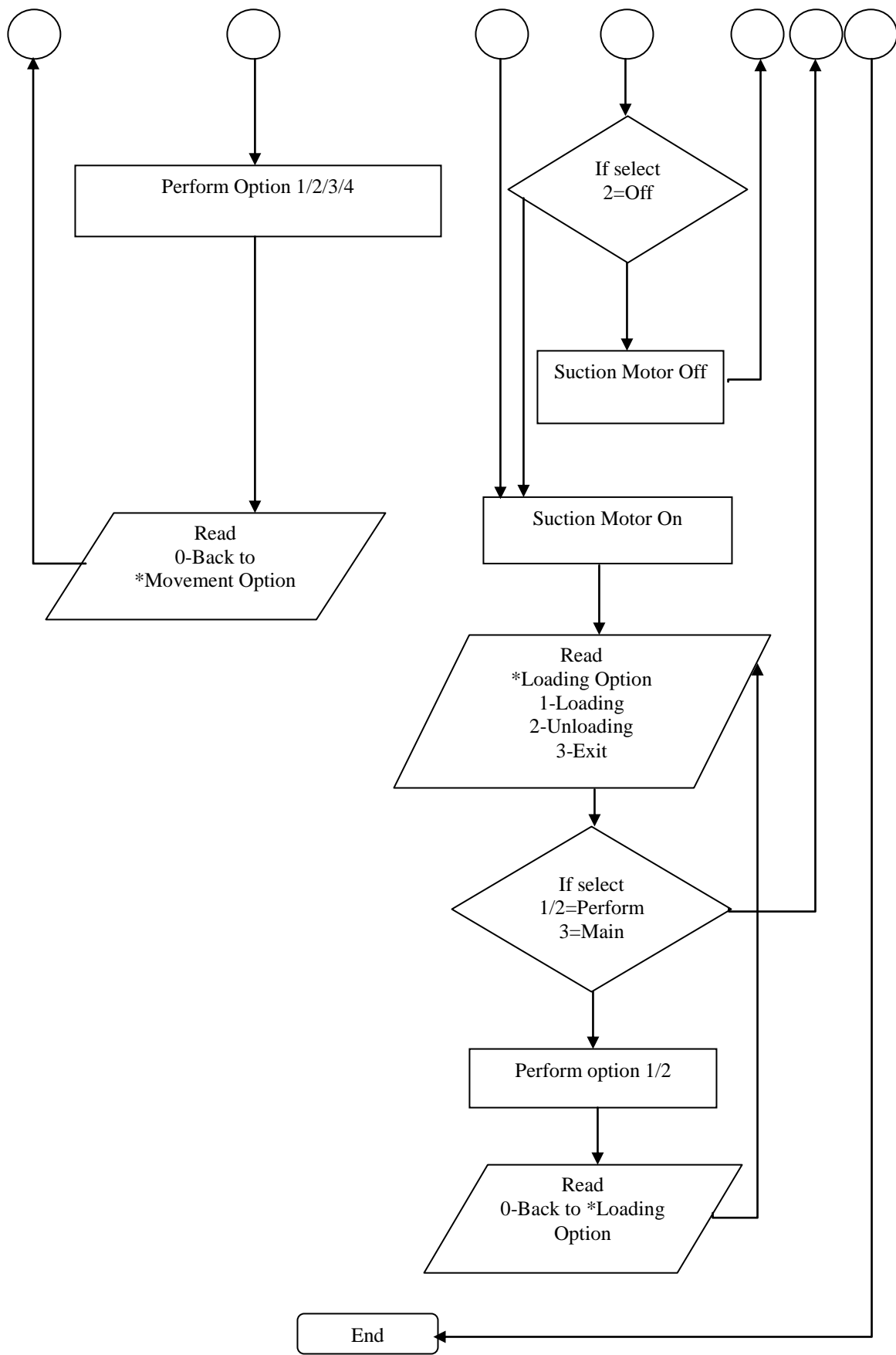
Symbol	Parameter	Value	Unit
V_S	Power Supply	50	V
V_{SS}	Logic Supply Voltage	7	V
V_i, V_{en}	Input and Enable Voltage	-0.3 to 7	V
I_O	Peak Output Current (each Channel)		
	- Non Repetitive ($t = 100\mu s$)	3	A
	- Repetitive (80% on -20% off, $t_{on} = 10ms$)	2.5	A
	-DC Operation	2	A

Source: Thomson Microelectronics, 2009

4.3 AGV Main Program

4.3.1 AGV Flow Chart





The AGV flow chart shows how the flow start from user selects the main options to choose whether to select movement of the AGV or to select loading and unloading part. Then, after user has selected the options, the program will run follow the steps like in the flow charts. This is how this AGV work.

4.3.2 AGV Program Codes

These were the result for the AGV program codes by using Code Blocks compiler. By using this compiler, it will interpret the C language in this program codes to run the motor (give output to motor). The main program was developed based on parallel port programming. Parallel port application has been used in this project.

The way to program the codes was based on bit value that will discuss after this. The parallel port programming was consisted of usages of header file ‘#include “iocpp.h”’ which allowed user to use ‘PortOut’ and ‘Delay’ function. These were specific commands for using parallel port.

Other commands used for parallel port programming were ‘LoadIODLL’ and ‘UnLoadIODLL’. These terms were used to open and close parallel port. ‘IODLL’ means Input Output Dynamics Link Library which was used to send the input through parallel port to the driver in types of bits. Other header file used was ‘#include <stdio.h>’ which was used for ‘Printf’ and ‘Scanf’ statement. **Figure 4.7** shows result for the program code.

Sample Program Codes

```

q.cpp*
#include "iocpp.h"
#include <stdio.h>
#include <conio.h>

int main()
{
LoadIODLL(); /* Open parallel port */

int type;

w: printf("\nOPTION\n\n");
printf("\n1-Movement\n");
printf("\n2-Loading/Unloading\n");
printf("\n3-Suction\n");
printf("\n4-Exit\n");

printf("\nplease enter your option\n");
scanf("%d",&type);

switch (type)
{
case 1:goto x;
case 2:goto y;
case 3:goto z;
case 4:goto m;
default: printf("\nInvalid selection. Please try again! \n");
}

int movement;
x: printf("\nOPTION\n\n");
printf("\n1-RIGHT\n");
printf("\n2-LEFT\n");

```

USE PORTOUT & DELAY FUNCTION

USE FOR "PRINTF" & "SCANF" STATEMENT

USE FOR "KBHIT" STATEMENT

OPEN PARALLEL PORT

Figure 4.7: AGV sample codes 1

```

q.cpp*
int movement;
x: printf("\nOPTION\n\n");
printf("\n1-RIGHT\n");
printf("\n2-LEFT\n");
printf("\n3-FORWARD\n");
printf("\n4-BACKWARD\n");
printf("\n5-Back To Main Menu\n");
printf("\nplease enter your option\n");
scanf("%d", &movement);

switch (movement)
{
    case 1: while (!kbhit())
            {
                printf("\nRight\n");
                PortOut(888, 5);
                Delay(1000);
                PortOut(888, 0);
                Delay(0);
            }
            break;
    case 2: while (!kbhit())
            {
                printf("\nLeft\n");
                PortOut(888, 24);
                Delay(1000);
                PortOut(888, 0);
                Delay(0);
            }
            break;
    case 3: while (!kbhit())
            {
                printf("\nForward\n");
                PortOut(888, 5+24);
                Delay(1000);
                PortOut(888, 0);
            }
            break;
}

```

RUN THE LOOP UNTIL THE KEYBOARD IS HIT
 C PROGRAM FUNCTION-SEND VALUE TO ADDRESS LOCATION
 CAUSE THE PROGRAM TO WAIT (IN MILISECOND)

Figure 4.8: AGV sample codes 2

4.3.3 Calculation for Bits Value

This part will show how the result to get bit value to be as the input in the AGV program codes. For movement options there were divided into four directions, Right, Left, Forward, Backward. In this part, the parallel port involved were port 1, 2, 3, 4, 5, 6. Therefore, the bits value for this port:

$$\text{Port 1} = \text{Pin 2} = 2^0 = 1$$

The bit value for this port was 1

$$\text{Port 2} = \text{Pin 3} = 2^1 = 2$$

The bit value for this port was 2

$$\text{Port 3} = \text{Pin 4} = 2^2 = 4 \text{ (Enable)}$$

The bit value for this port was 4

To rotate the motor Clockwise, CW direction (Based on test circuit)

$$\begin{aligned} \text{CW} &= \text{Port 2} + \text{Port 3} \\ &= 2 + 4 \\ &= 6 \end{aligned}$$

To rotate the motor Counter Clockwise, CCW direction (Based on test circuit)

$$\begin{aligned} \text{CCW} &= \text{Port 1} + \text{Port 3} \\ &= 1 + 4 \\ &= 5 \end{aligned}$$

$$\text{Port 4} = \text{Pin 5} = 2^3 = 8 \text{ (Enable)}$$

The bit value for this port was 8

$$\text{Port 5} = \text{Pin 6} = 2^4 = 16$$

The bit value for this port was 16

$$\text{Port 6} = \text{Pin 7} = 2^5 = 32$$

The bit value for this port was 32

To rotate the motor Clockwise, CW direction (Based on test circuit)

$$\begin{aligned} \text{CW} &= \text{Port 6} + \text{Port 4} \\ &= 32 + 8 \\ &= 40 \end{aligned}$$

To rotate the motor Counter Clockwise, CCW direction (Based on test circuit)

$$\begin{aligned} \text{CCW} &= \text{Port 5} + \text{Port 4} \\ &= 16 + 8 \\ &= 24 \end{aligned}$$

Thus, the bits values for each direction were listed below;

$$\begin{aligned} \text{Right} &= 5 \\ \text{Left} &= 24 \\ \text{Forward} &= 5+24 \\ \text{Backward} &= 6+40 \end{aligned}$$

For loading & unloading options there were divided into two tasks, loading and unloading. In this part, the parallel port involved were port 4, 7, 8. Therefore, the bits value for this port:

$$\text{Port 4} = \text{Pin 5} = 2^5 = 8 \text{ (Enable)}$$

The bit value for this port was 8

$$\text{Port 7} = \text{Pin 8} = 2^6 = 64$$

The bit value for this port was 64

$$\text{Port 8} = \text{Pin 9} = 2^7 = 128$$

The bit value for this port was 128

To rotate the motor Clockwise, CW direction (Based on test circuit)

$$\begin{aligned} \text{CW} &= \text{Port 7} + \text{Port 4} \\ &= 64 + 8 \\ &= 72 \end{aligned}$$

To rotate the motor Counter Clockwise, CCW direction (Based on test circuit)

$$\begin{aligned} \text{CCW} &= \text{Port 8} + \text{Port 4} \\ &= 128 + 8 \\ &= 136 \end{aligned}$$

Thus, the bits values for each task were listed below;

$$\begin{aligned} \text{Loading} &= 72 \\ \text{Unloading} &= 136 \end{aligned}$$

For suction options there were divided into two tasks, On & Off pump. In this part, the parallel port involved were port 3, 7, 8. Therefore, the bits value for this port:

$$\begin{aligned} \text{Port 3} &= \text{Pin 4} = 2^2 = 4 \text{ (Enable)} \\ \text{The bit value for this port was } &4 \\ \text{Port 7} &= \text{Pin 8} = 2^6 = 64 \\ \text{The bit value for this port was } &64 \\ \text{Port 8} &= \text{Pin 9} = 2^7 = 128 \\ \text{The bit value for this port was } &128 \end{aligned}$$

To rotate the motor Clockwise, CW direction (Based on test circuit)

$$\begin{aligned} \text{CW} &= \text{Port 7} + \text{Port 3} \\ &= 64 + 4 \\ &= 68 \end{aligned}$$

To rotate the motor Counter Clockwise, CCW direction (Based on test circuit)

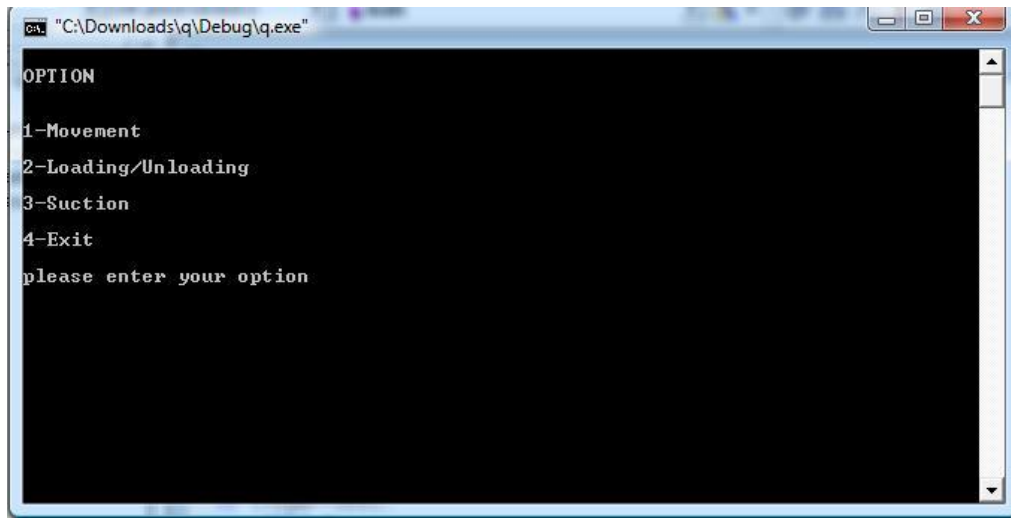
$$\begin{aligned} \text{CCW} &= \text{Port 8} + \text{Port 3} \\ &= 128 + 4 \\ &= 132 \end{aligned}$$

Thus, the bits values for each task were listed below;

$$\begin{aligned} \text{On} &= 68 \\ \text{Off} &= 132 \end{aligned}$$

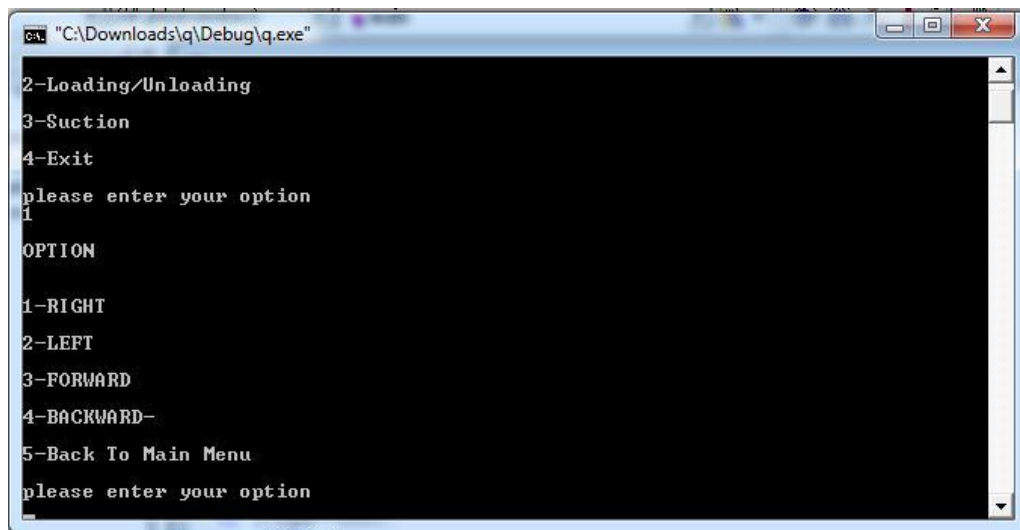
4.3.4 C Program Interface

There was the result after compiling by using Code Blocks compilers where the command prompt interface will appear. This interface needs user to insert value to choose the options. **Figure 4.9** until **Figure 4.12** shows the steps to choose the options for this AGV program.



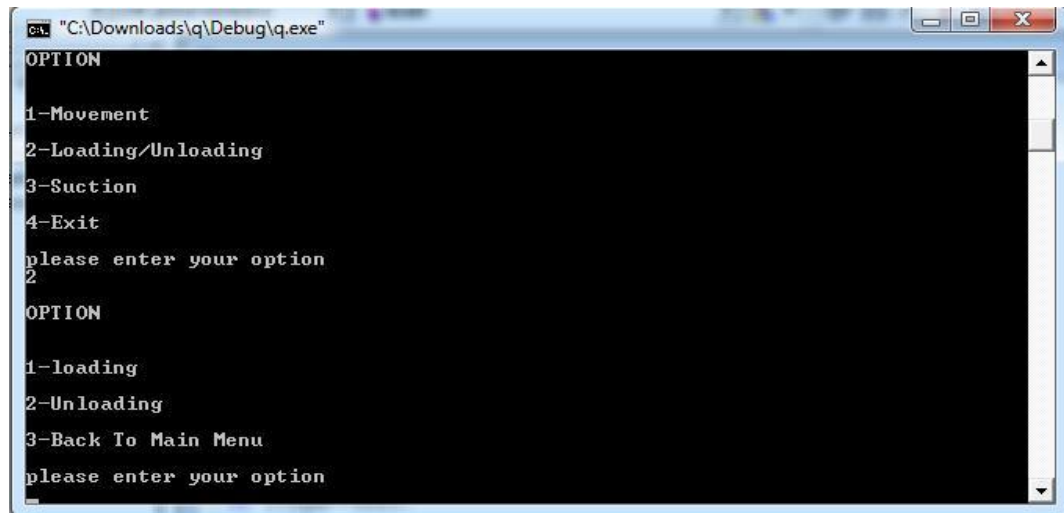
```
ca: "C:\Downloads\q\Debug\q.exe"
OPTION
1-Movement
2-Loading/Unloading
3-Suction
4-Exit
please enter your option
```

Figure 4.9: C interface 1



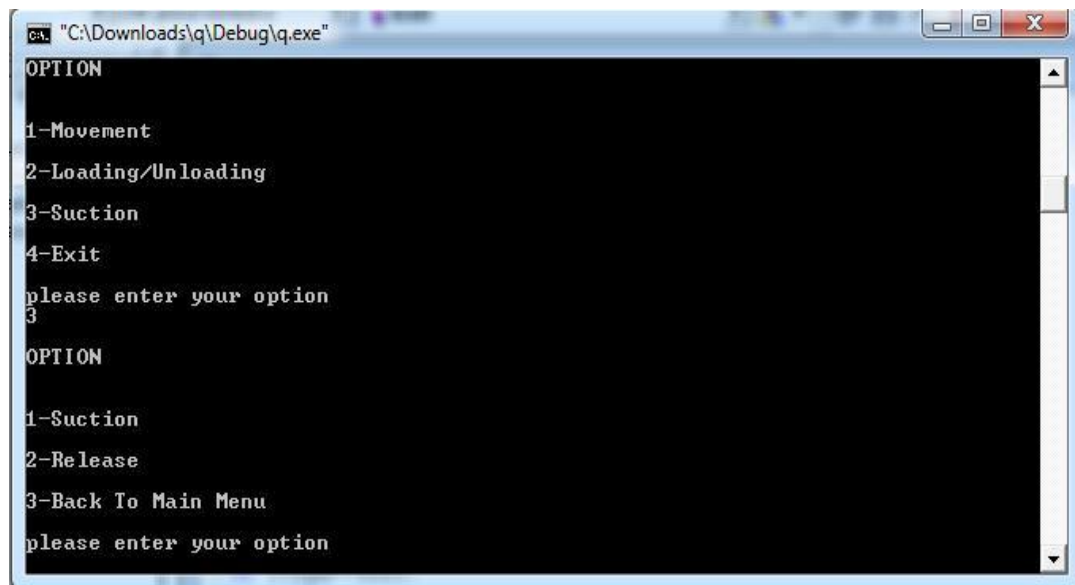
```
ca: "C:\Downloads\q\Debug\q.exe"
2-Loading/Unloading
3-Suction
4-Exit
please enter your option
1
OPTION
1-RIGHT
2-LEFT
3-FORWARD
4-BACKWARD-
5-Back To Main Menu
please enter your option
```

Figure 4.10: C interface 2



```
ca: "C:\Downloads\q\Debug\q.exe"
OPTION
1-Movement
2-Loading/Unloading
3-Suction
4-Exit
please enter your option
2
OPTION
1-loading
2-Unloading
3-Back To Main Menu
please enter your option
```

Figure 4.11: C interface 3



```
ca: "C:\Downloads\q\Debug\q.exe"
OPTION
1-Movement
2-Loading/Unloading
3-Suction
4-Exit
please enter your option
3
OPTION
1-Suction
2-Release
3-Back To Main Menu
please enter your option
```

Figure 4.12: C interface 4

4.3.5 Visual Basic (VB) Interface

This was the result for Visual Basic (VB) interface. Compared to command prompt interface, this interface was more users friendly because it didn't need user to insert any value. This interface just needs users to press each command button to execute each task. **Figure 4.13** below shows the AGV control system interface using VB.

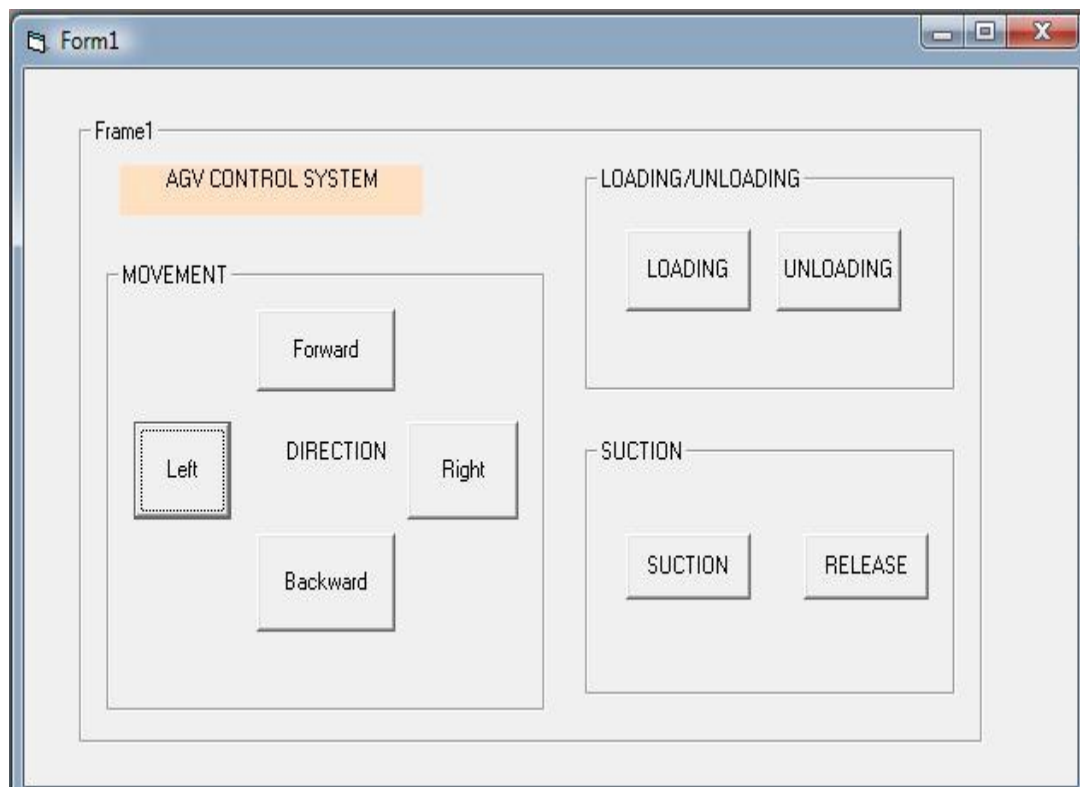


Figure 4.13: VB interface 1

4.4 Limitations

There were several limitations in this AGV control system which needs further improvement to get the perfect control system. Since this project still in development stage, there was some aspect that needs to be upgraded. These were limitations in this project;

- I. This AGV control system used wired (parallel port cables) to connect between computer to the on-board circuit.
- II. This AGV control system used power supply to give input voltage to L298N driver.

To connect between computers to the on-board circuit, it needs to provide long parallel port cable to ensure it can move in wide area. Since the parallel port cable will disturb the flow direction of the AGV, it will disturb the movement of AGV.

There was also to provide a power supply to supply voltage to the circuit. This needs to provide long cable to connect the power supply to the circuit. This also will disturb the movement of this AGV.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

After all the tasks were done, as the conclusion from this project, the objective in this project was done successfully achieved which to develop the control system for Automated Guided Vehicle (AGV). The second objective also successfully achieved that is to develop Automated Guided Vehicle (AGV) circuit and wiring system. The third objective also was done successfully achieved which to test the overall AGV system

From the result, the AGV main circuit was consisted of four Dual Full – Bridge Driver (L298N). Each driver has its own task which to control each motor at tires, vertical bar and diaphragm pump. This motor will control the movement of the AGV whether to turn right, left, move forward and move backward. It also control the motor for loading and unloading part and suction process for diaphragm pump.

From the AGV program, this program will send the input from computer to the AGV circuit through by using parallel port cable in bits value. This program was consisted all commands to each driver to do its own task. The output from this program were controlled the rotation of each motor.

5.2 Further Study Recommendation

After several improvement study and analysis in this project, there were some recommendations and further development listed below to improve this AGV control system to give better performance. The recommendations are:

- I. Use battery to switch the power supply to ensure this AGV is more independent and have better performance.
- II. For further development, use wireless AGV control to connect between computers to the AGV main body. This will make this AGV can move freely without disturbed by wired (Parallel port cable). This will make the AGV have wide range for its movement.
- III. Add other component like mobile camera to supervise its movement and also acoustic sensor to detect any incoming vibration on the surface.

REFERENCES

1. Apple, James M.1977. Plant Layout and Materials Handling. New York: Ronald Press.
2. Dr Wan Azhar Wan Yusoff.2006. BMM 2413 Electronic Lab04 Parallel Port Programming. Fakulti Kejuruteraan Mekanikal, Universiti Malaysia Pahang, 2006.
3. Hammond, Gary.1986. AGVS at Work: Automated Guided Vehicle Systems. Bedford:UK: IFS.
Oct 2008; 164, 10; ABI/INFORM Trade & Industry pg. 759.
4. Peters, B.A., Smith,J.S., and Venkatesh,S., A control classification of automated guided vehicle system., Accepted in the International Journal of Industrial Engineering, 1995.
5. Seo, Y.,and Egbelu, PJ., Flexible guide path design for automated guided vehicle system., International Journal of Production Research 33, n4,1995,pp 1135-1156
6. Seo, Yoon-Ho, Ph.D.1993, Integrated manufacturing system design through process plan and AGV guide path selection, The Pennsylvania State University.
7. Singh,N., Design of cellular manufacturing system, European Journal of Operations Research 69,n 3, 1993,pp 284-291.
8. Sinreich,D., and Tanchoco,J.M.A., Intersectiongraph method for AGV flow path design., International Journal of Production Research, v 29,n 9,1991, pp 1725-1732.
9. Srinivasan Rjagopalan.1999, Flow path Design & Analysis of AGV based Material Handling System, Rensselaer Polytechnic Institute
10. Stephen, Cameron, Penelope, Probert, eds. (1994). Advanced Guided Vehicles. River Edge, NJ: World Scientific.
11. Thomson Microelectronics, 2009 - Datasheet Catalog com.

APPENDIX A
PROJECT GANTT CHART

Appendix A1: Gantt chart for FYP 1

Project Activities	Week														
	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15
Identify title	■														
Identify Objectives and scopes		■	■												
Literature review		■	■	■	■	■									
Methodology planning			■	■	■	■	■	■							
Concept development				■	■	■	■	■							
System level design					■	■	■	■	■	■	■				
Development of circuit & wiring system								■	■	■	■	■			
Development of program									■	■	■	■			
Testing & Refinement										■	■	■	■		
Preparation of PSM 1 presentation											■	■	■	■	

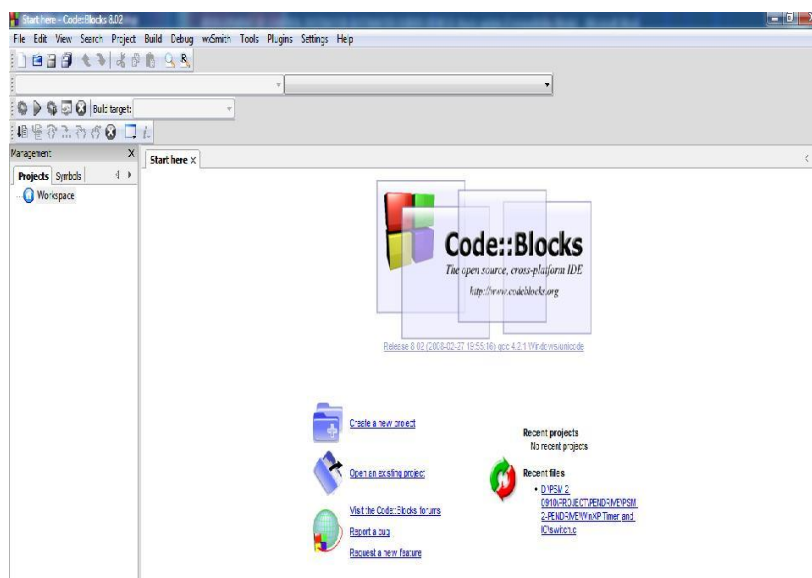
APPENDIX B

CODE BLOCKS SETUP

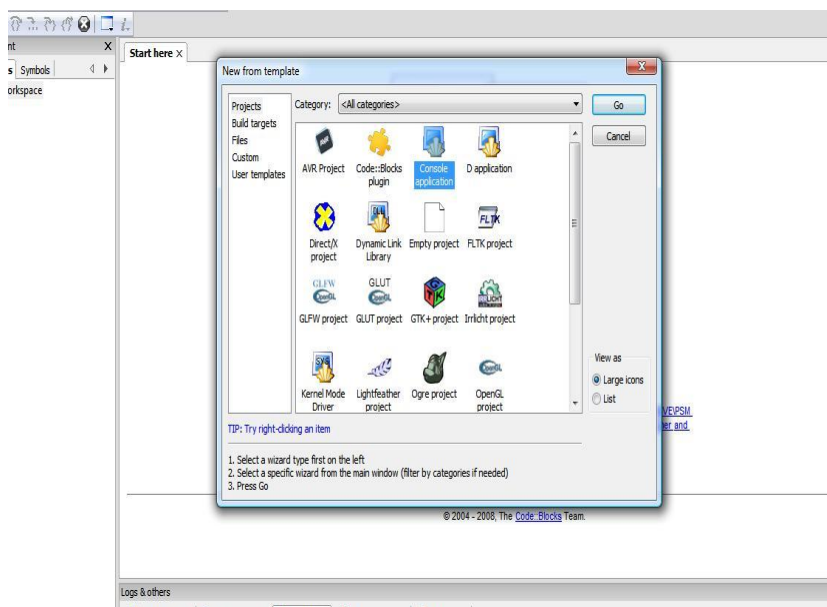
Appendix B

Create a Code Blocks Work Project

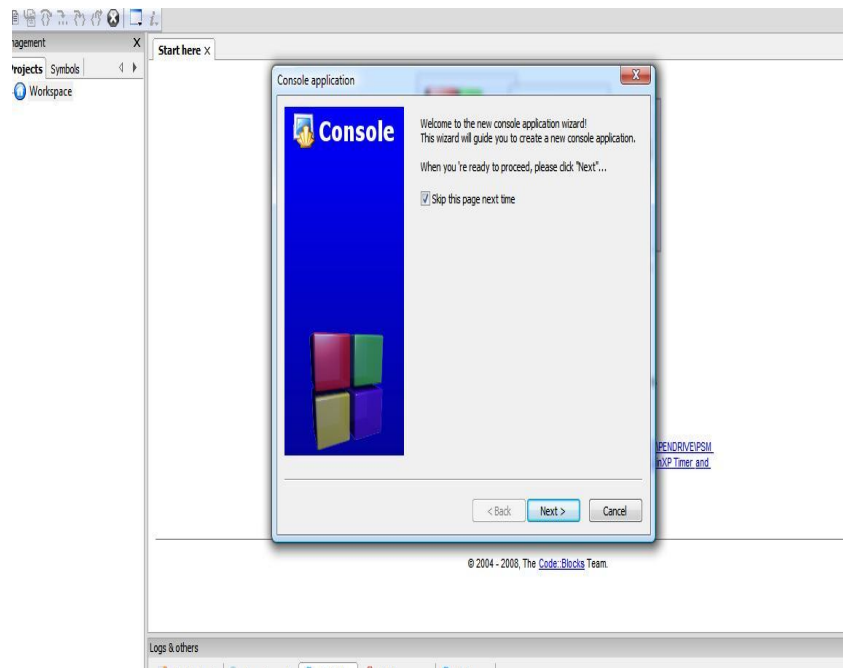
1. Click Code Blocks
2. Choose 'Create New Project.



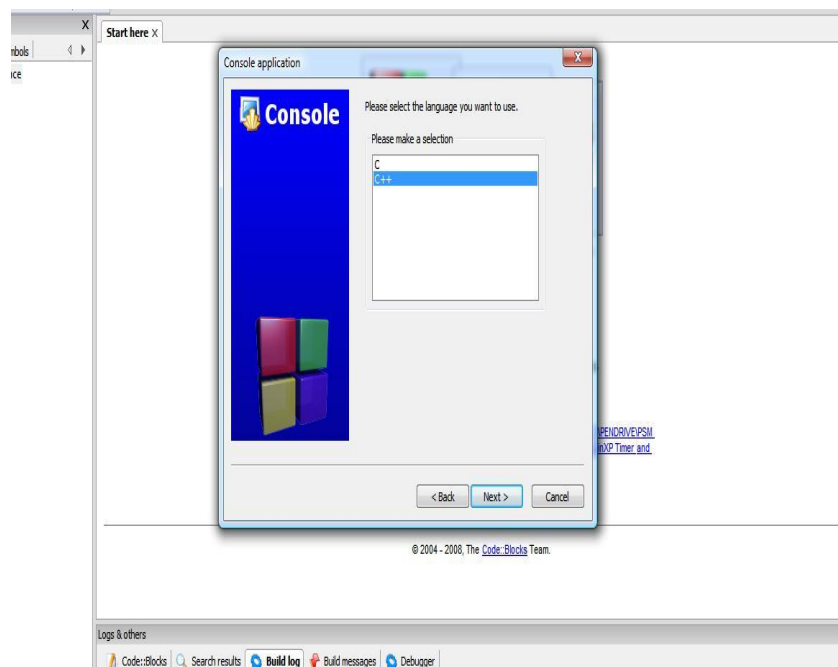
3. Choose Console Application



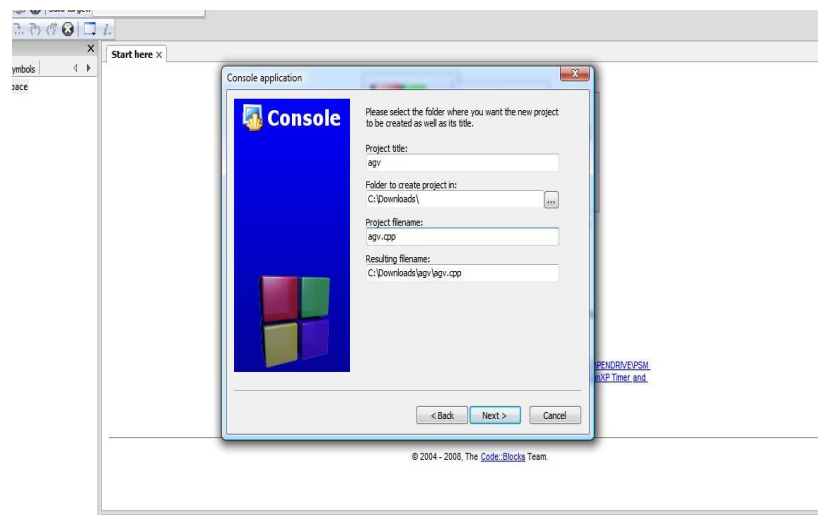
4. Mark the check board 'skip the page next time' then click next.



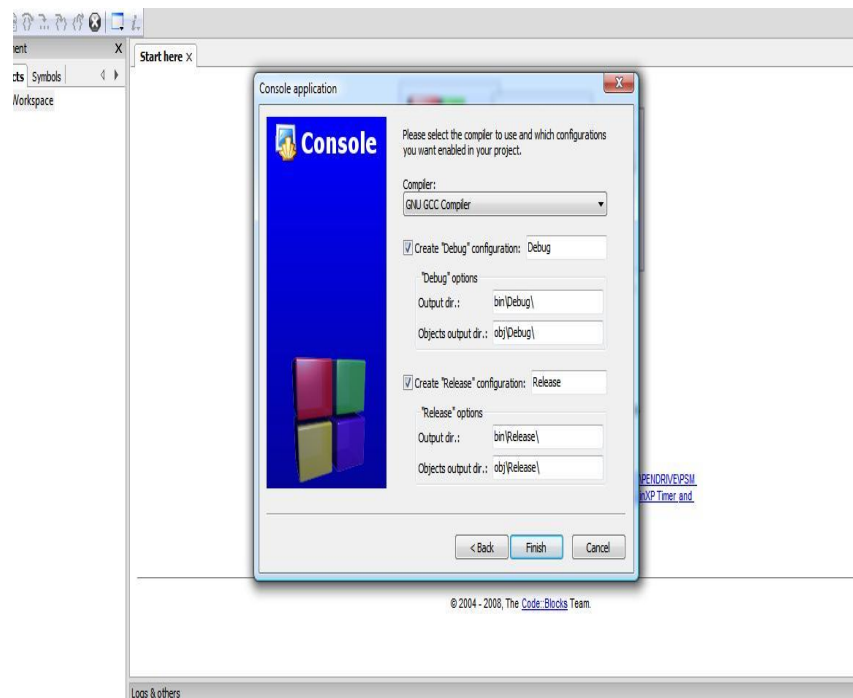
5. Choose C++.



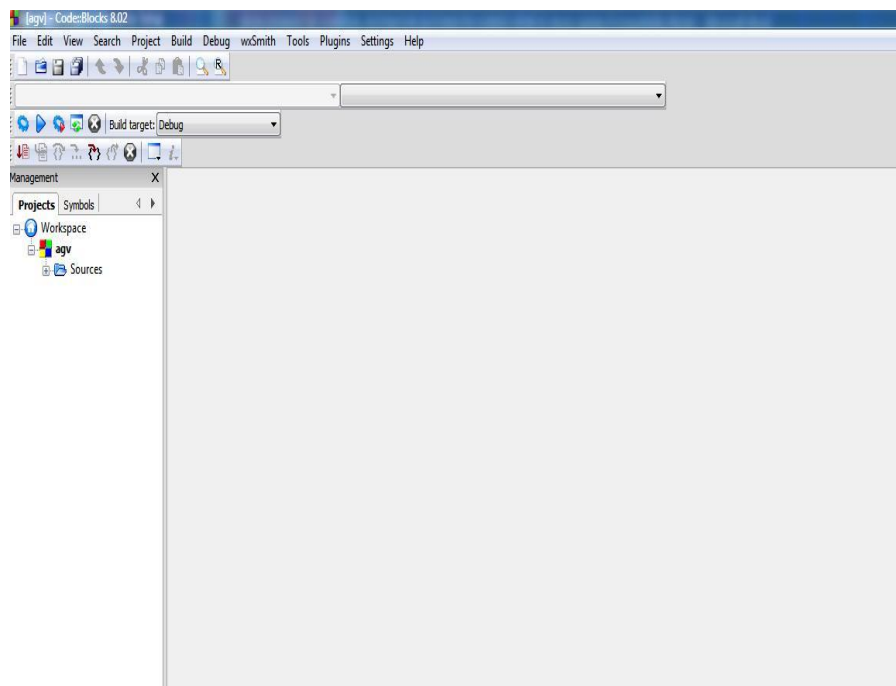
6. Write file name with extension '.cpp'. Then click next



7. Click Finish



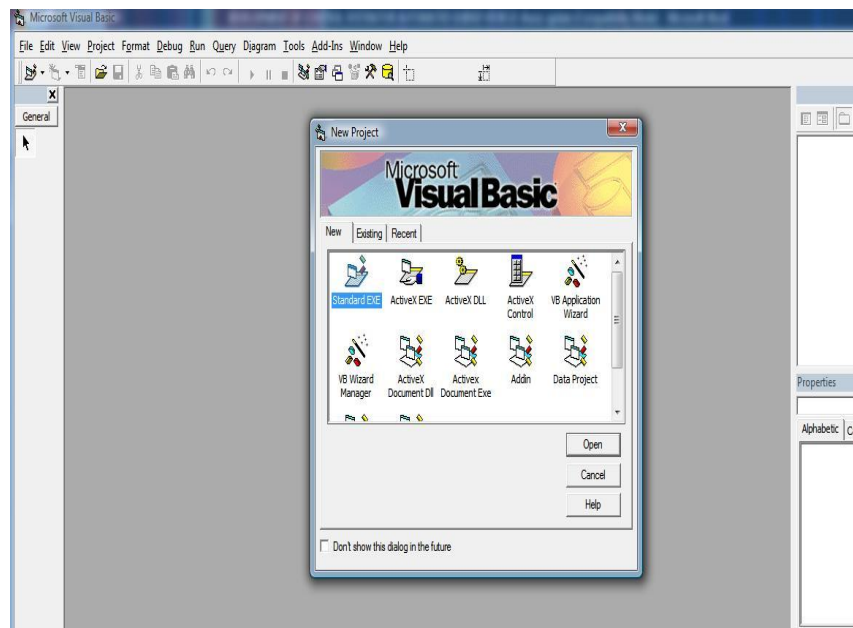
8. Code Blocks Work Project will appear.



APPENDIX C
VISUAL BASIC 6 (VB 6) SETUP

Appendix C

1. Open Visual Basic 6. Then chose 'Standard EXE'



2. Standard EXE work project will appear.

