UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS♦

JUDUL: **COIL WINDING MACHINE**

SESI PENGAJIAN: 2012/2013

Saya **SARINA BINTI SHAFIE (880531-12-5426)**
(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:

DEWAN NEGERI, ARAS 4,
B'NAN DEWAN UNDANGAN NEGERI SABAH,
PETI SURAT 11247, 88813,
KOTA KINABALU SABAH.

Tarikh: **1 Julai 2013** Tarikh: : **1 Julai 2013**

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).

COIL WINDING MACHINE

SARINA BINTI SHAFIE

Thesis submitted in partial fulfilment of the requirements
for the award of Bachelor of Mechatronic Engineering

Faculty of Manufacturing Engineering
UNIVERSITI MALAYSIA PAHANG

JUNE 2013

UNIVERSITI MALAYSIA PAHANG

FACULTY OF MANUFACTURING ENGINEERING

I certify that the thesis entitled “Coil Winding Machine” is written by Sarina binti Shafie with matric number FB09066. I have examined the final copy of this report and in my opinion, it is fully adequate in terms of language standard, and report formatting requirement for the award of the degree of Bachelor in Mechatronic Engineering. I herewith recommend that it be accepted in fulfillment of the requirements for the degree of Bachelor in Mechatronic Engineering.

Signature :

Name of Examiner : Mr. Khairul Fikri Bin Muhammad

Date : 1st July 2013

SUPERVISOR'S DECLARATION

I hereby declare that I have checked this thesis, which written by Sarina binti Shafie, and in my opinion, this thesis is adequate in terms of scopes and quality for the award of the degree of Bachelor of Mechatronic Engineering.

Signature :
Name of Supervisor : Mr. Wan Hasbullah Bin Mohd Isa
Date : 1st July 2013

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 other Degree and is not concurrently submitted for award of other degree.

Signature :

Name : Sarina binti Shafie

ID Number : FB09066

Date : 1st July 2013

ACKNOWLEDGMENT

In the name of Allah S.W.T the Most Beneficent and the Most Merciful, infinite thanks I brace upon the Almighty for giving me the strength and ability to complete my Final Year Project.

First and foremost, I would like to thanks the respective to my Final Year Project (FYP) supervisor Mr. Wan Hasbullah Bin Mohd Isa and my Final Year Project Panel Mr Khairul Fikri Bin Muhammad. I also to thank a lot to all my friend who help me in finishing my FYP, Without their outstanding support and interest, this report of mine would not have been the best it would as of now.

I would also like to express my deepest appreciation to my parents and my siblings whom always supports me financially, morality and their never-ending motivations to ensure that I complete my FYP and this report with less difficulty.

Last but not least, I am also in debt to the Faculty of Manufacturing Engineering (FKP) University Malaysia Pahang, for their guidance, starting from my starting project till the end of my successful FYP. My sincere thanks go to all staff in Lab for give me guidance in using machine and also provide me the material for my project. My sincere appreciation also reaches out to all my colleagues and friends who have provided assistance at various occasions. Also to all individuals who was involved either directly nor indirectly in the completion of this report. Indeed I could never adequately express my indebtedness to all of them. Thank you a lot.

ABSTRACT

A coil winding machine is a machine for winding coil onto a spool, bobbin and many more. This coil winding machine is one of types of winding machine that available in industries today. From multi speeded machines to medium, large and extra-large machines, these machines come in various types and categories, performing a range function. The common applications for a coil winding machine are to wind coils for transformer, inductors, motor and chokes. To complete a coil using manual coil winding machine will be inconvenience and waste of time. Therefore, fabrication of coil winding machine will be done in this project which is controlled by two stepper motor using Arduino program. This machine is inexpensive, easy to operate and build in a small-scale size. This project also can be used for training students in winding of small transformers & relay coils.

ABSTRAK

Sebuah mesin penggulungan gegelung mesin penggulungan gegelung ke kili, sepul dan banyak lagi. Ini mesin penggulungan gegelung adalah salah satu jenis mesin yang terdapat di industri pada hari ini. Dari mesin yang mempunyai pelbagai kelajuan kepada mesin sederhana, besar dan lebih besar, mesin ini datang dalam pelbagai jenis dan kategori, melakukan fungsi pelbagai. Aplikasi biasa untuk mesin penggulungan gegelung adalah untuk menggulung gegelung untuk pengubah, peraruh, motor dan "chokes". Untuk melengkapkan gegelung menggunakan manual gegelung mesin penggulungan akan menyukarkan dan membuang masa. Oleh itu, dengan mencipta gegelung mesin penggulungan akan dilakukan dalam projek ini akan dikawal oleh dua motor dan menggunakan program Arduino. Mesin ini adalah murah, mudah untuk beroperasi dan membina dalam saiz yang kecil. Projek ini juga boleh digunakan untuk melatih pelajar dalam penggulungan pengubah kecil & gegelung geganti.

TABLE OF CONTENTS

			Page
APPROVAL DOCUMENT			ii
SUPERVISOR DECLARATION			iii
STUDENT DECLARATION			iv
ACKNOWLEDGEMENTS			v
ABSTRACT			vi
ABSTRAK			vii
TABLE OF CONTENTS			ix
LIST OF FIGURES			xii
LIST OF TABLES			xiv
LIST OF SYMBOLS			xv
LIST OF EQUATIONS			xvi
CHAPTER 1	**INTRODUCTION**		1
	1.1	Background information	1
	1.2	Problem Statement	2
	1.3	Objectives of the project	2
	1.4	Scope of Project	2
	1.5	Flow chart	4
	1.6	Thesis Overview	5
CHAPTER 2	**LITERATURE REVIEW**		7
	2.1	Introduction	7
	2.2	Comparison between Round and Rectangular Bobbin	7
	2.3	Method of Windings Coil	9
	2.4	Importance of tension in Coil Winding Machine	9
	2.5	Function and advantage of stepper motor	10
	2.6	Arduino with Stepper Motor	11

CHAPTER 3	**METHODOLOGY**		12
	3.1	Introduction	12
	3.2	Project Flow Chart	13
	3.3	Block Diagram	14
	3.4	Mathematical Model	15
		3.4.1 Tension Control System	16
	3.5	Mechanical Component	17
		3.5.1 Bobbin	17
		3.5.2 Lead Screw with Long Nut	17
		3.5.3 Aluminium Rod	18
		3.5.4 Aluminium Plate	19
		3.5.5 Guide Pulley	19
		3.5.6 Bearing	20
		3.5.7 Motor Coupling	20
		3.5.8 Angle Bar	21
		3.5.9 Wire Copper Coil	21
	3.6	Electrical Component	22
		3.6.1 Stepper Motor	22
		3.6.2 Arduino UNO	23
		3.6.3 L298N	23
	3.7	Circuit Design	25
	3.8	Machine Operation Flow	26
	3.9	Programming on Arduino	27
CHAPTER 4	**RESULT AND DISCUSSIONS**		29
	4.1	Introduction	29
	4.2	Hardware Fabrication	29
	4.3	Arduino Program	31
	4.4	Winding Product	33
	4.5	Discussion	33

CHAPTER 5	**CONCLUSION AND RECOMMENDATION**	35
	5.1 Conclusion	35
	5.2 Recommendation	36
REFERENCES		37
APPENDICES		40
A	Gantt Chart for FYP1	40
B	Gantt Chart for FYP2	41
C	Stepper Motor Datasheet	42
D	L298N Datasheet	44
E	Arduino coding (FINAL)	52
F	Arduino Uno Datasheet	53

LIST OF FIGURE

Figure no.	Title	Page
1.1	Flow chart	4
2.1	Acceleration due to the changing wire path length during winding.	8
2.2	Basic winding speed tension control system	10
3.1	Flow Chart of Methodology	13
3.2	Block Diagram	14
3.3	Sketch of the wire and the bobbin	15
3.4	Body Diagram of the first wire segment to make contact with the bobbin	15
3.5	Bobbin	17
3.6	Lead screw with long nut	18
3.7	Aluminium Rod	18
3.8	Aluminium Plate	19
3.9	Wire Carriage	19
3.10	Bearing	20
3.11	Motor Coupling	20
3.12	Angle bar	21
3.13	Wire Copper Coil	21
3.14	Stepper motor	22
3.15	Arduino UNO	23
3.16	L298N	24

3.17	Circuit Design	25
3.18	Project Design	26
3.19	Arduino Program to control two stepper motor	27
4.1	Motor Driver Circuit	30
4.2	Mechanism of Movement	30
4.3	Complete Fabrication of Winding Machine	31
4.4	Coil Winding Machine Product.	33

LIST OF TABLE

Figure no.	Title	Page
4.1	Coding Explanation	32

LIST OF SYMBOLS

τ_1	Force length ratio
Ɵ	Angle in radians
T	Tension
A	Sectional area
Y	Yield strength material
V	Velocity
π	Pi

LIST OF EQUATIONS

Equation No.	Title	Page
2.1	Net force of the bobbin on the segment	15
2.2	To determine force ratio	16
2.3	To determine tension	16
2.4	To determine tension(simplified)	16
2.5	Velocity	16

CHAPTER 1

INTRODUCTION

1.1 Background information

Typically, a winding machine winds a material such as metal wire, thread, or paper, onto a core, spool, or bobbin. There are several different types of winding machines, from simple manual feed machines to complex computer-numeric-control (CNC) machines. Some of the more common uses for winding machines are coil winding, rope winding, and continuous filament winding. Many industries use these devices, including textile, electronics, and wire industries. A manual winding machine usually has a core on a spindle and the user feeds wire, rope, or other material onto the core. The user controls the spindle speed and feeds the material through user hand, guiding it to control the tension and load pattern. These simple machines may be of a bench-top size or large stand-alone winder.

A coil winding machine is a machine for winding coil onto a spool, bobbin and many more. This coil winding machine is one of types of winding machine that available in industries today. The coil winders can be classified according to their speed levels and capacity. From multi speeded machines to medium, large and extra-large machines, these machines come in various types and categories, performing a range function. The common applications for a coil winding machine are to wind coils for transformer, inductors, motor and chokes. Coil winding machine design is dictated by a coil's complexity, material tension limitations, machine versatilities, and automation / operator intervention, production volume and budgetary considerations [3]. Complete types of winding machine ideal for educational institutes, small and medium enterprise.

1.2 Problem Statement

To complete a coil using manual coil winding machine will be inconvenience and waste of time. Furthermore, to have a good quality of automatic coil winding machine requires expensive tools and not user-friendly. Therefore, fabrication of coil winding machine will be done in this project which is controlled by two stepper motor using Arduino . This machine is inexpensive, easy to operate and build in a small-scale size. This project also can be used for training students in winding of small transformers & relay coils.

1.3 Objectives of the project

Basically these projects are listing three main objectives. The objectives are a guideline in order to complete this project. This project is conducted to achieve the following objectives:

i. To design and build a coil winding machine that has a small-scale and at a lower cost.
ii. To create a program that control stepper motor movement by using Arduino.
iii. To be a learning tool for student to wind a small solenoid and transformer.

1.4 Scope of Project

Project scope is the part of project planning that involves determining and documenting a list of specific project goals, deliverables, tasks, costs and deadlines. In this project, the scope of the project is including a design and fabrication a small-scale coil winding machine at a lower cost. In this project, the coil winding machine size will be only in desktop size. All the programming and coding project will be using Arduino program to control two stepper motor motion. Some of the coils winding machine components are available at the faculty laboratory such as stepper motor, aluminium plate, and lead screw. Certain component such as round bobbin, guide pulley and motor coupling is fabricated using rapid prototyping machine.

1.5 Flow chart

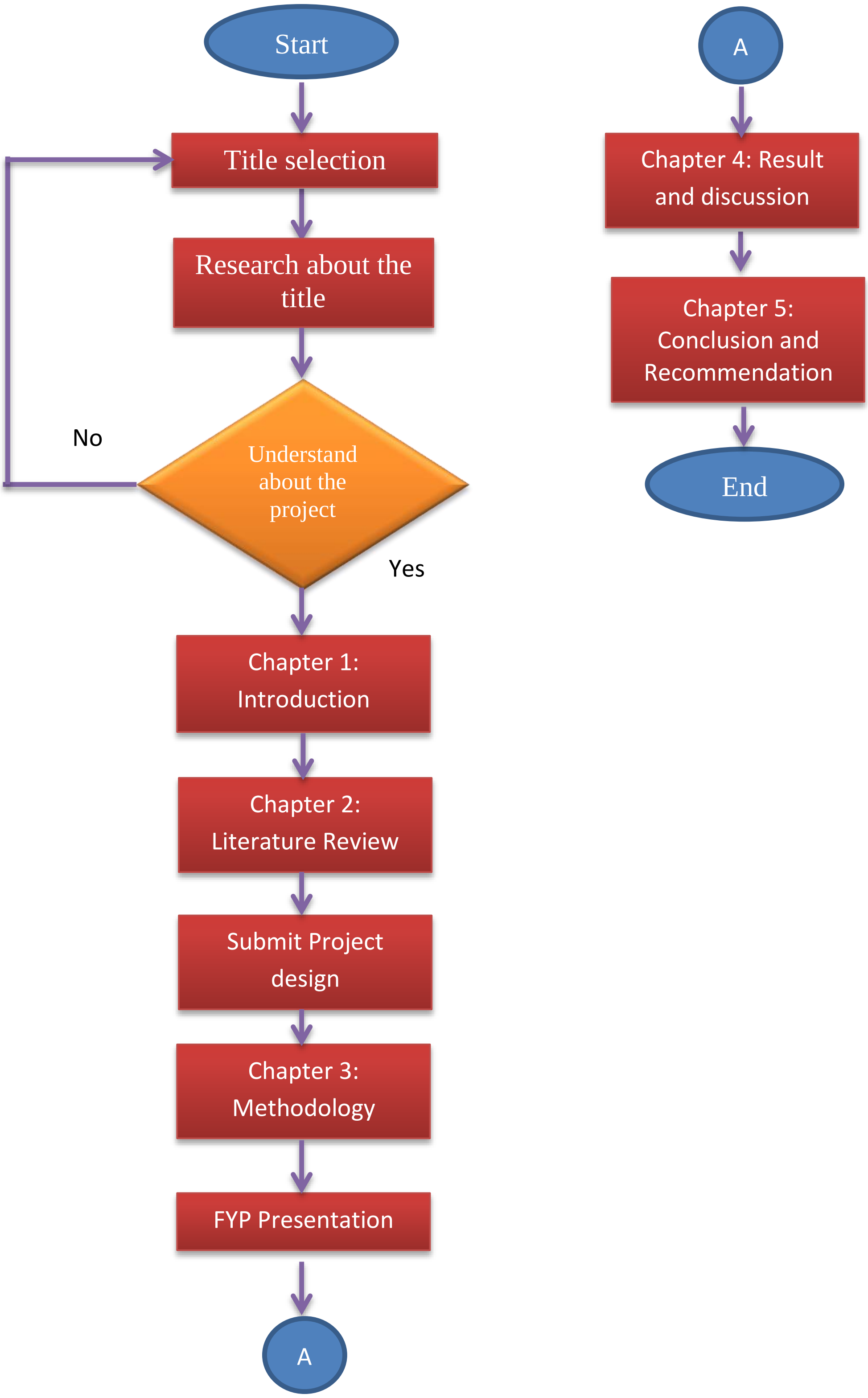


Figure 1.1: Flow chart

1.6 Thesis Overview

Coil Winding Machine final thesis is a compilation of 6 chapters that contains and elaborates specific topics such as the Introduction, Literature Review, Methodology, Architecture, Result and Analysis, Conclusion and Further Development that can be applied in this project.

i. Chapter 1: Introduction

   This Chapter 1 is an introduction which is the overview of the project, objective of the project, problem statement, scope of project and project flow chart through this semester.

ii. Chapter 2: Literature review

   This Chapter 2 content about literature review on comparison between round and rectangular bobbin, method of windings coil, also a research about importance of tension in coil winding machine, function and advantage of stepper motor, Visual basic and Parallel port connection.

iii. Chapter 3: Methodology

   This chapter will explain about the project methodology through this project. Project Flow chart, block diagram, project component, operation flow and visual basic programming.

iv. Chapter 4: Result and discussion

   Discuss all the results obtained and the limitation of the project. All discussions are concentrating on the result and performance of the coil winding machine.

v. Chapter 5: Conclusion

Discuss the conclusion about the project from the beginning until the end. Also conclude all the advantage and disadvantage through the project for further improvement.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter reviews about previous studies to fabricate coil winding machine. Before any project was started, proper and deep analysis is required to understand the basic function of the system. From that analysis, appropriate and suitable components and method can be selected to deal with the project function.

The objectives of this chapter are:

- To determine which method are appropriate to use to winds a coil.
- To identify the importance of tension control in coil winding machine.
- To ensure that stepper motor and Arduino program are suitable for this project.

2.2 Comparison between round and rectangular bobbin

There are various types of bobbin that being used in coil winding machine nowadays. The shapes of bobbin are important to winding a coil. The shape of the coil being wound has a significant impact on the quality of the tension applied by the tensioner [7]. Most of them are round bobbin and rectangular bobbin. From the previous study, many of them prefer round bobbin compare to rectangular bobbin. It because the round shapes on the round bobbin allows a uniform round when winding a coil. Compare with rectangular coil, the coil tend to breaks up because of the shape itself causing the coil windings become uneven. Besides that, rectangular coil is hard to maintain the tension control. Tension device must be absorb and maintain the control when handling a rectangular bobbin which applies 4 impacts points per revolution. In addition, rectangular coil winding speed are very limited.

As the coil turns on the winding machine shaft, the speed of the wire feeding onto the coil accelerates and decelerates as a rectangular coil is being wound. As shown in figure below, due to the continuously altering wire path length the speed become vary. Round bobbin does not have a problem with this because the wire contact point on the coil is fixed.

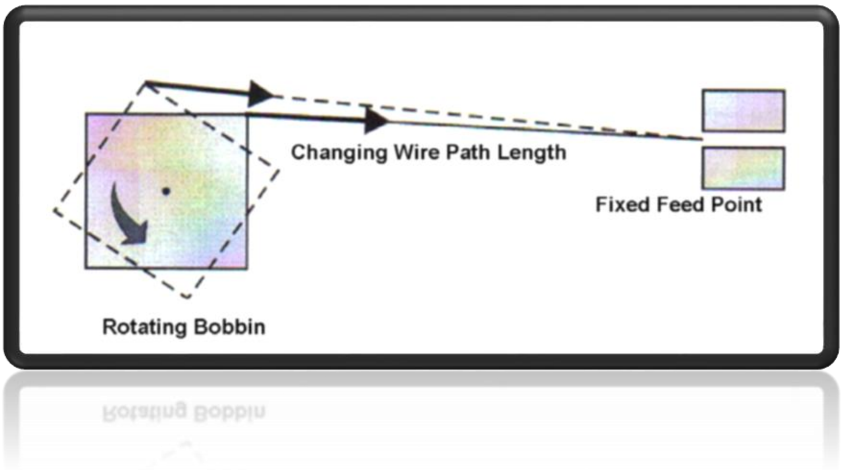


Figure 2.1: Acceleration due to the changing wire path length during winding [7].

2.3 Method of windings coil

The coordination of rotational and translational motions is necessitating when transferring wire to bobbin. There are many methods to winds a coil nowadays. Previous study shows there are two of the most common methods of winding coils are spindle winding and fly winding. In the spindle winding process, the coil is wound by rotating its core or bobbin. In the fly winding process, the bobbin is fixed and the wire is wrapped around it [2]. Spindle winding are easiest compare to fly winding process because it can control the wire position uniformly and smooth compare to fly winding. Fly winding are tend to get twisted because of inconsistent twisting and inaccurate wire displacement. The centripetal force in spindle winding machine wire tension is required to rotate the wire in circular path. It leads to a looser coil and reduces the coil tightening forces. If there is less reduction, spindle winding are desirable because of the ability to control the wire position.

2.4 Importance of tension in coil winding machine

Tension is the force required to pull the wire against the accumulation of all resistance, forces and loads imposed on the wire as it moves, including the tension device [5]. Consistent tension is important to achieving the aim of coil electrical characteristics also stable performance from coil to coil. Pretension is when the force applies to the wire path before and after tension device. Pretension is caused by the friction of the wire moving over pulleys, supply spool flange and many more. There are two types of tension state which is static and dynamic state. In dynamic state, the tension started when the wire meets the bobbin or the previous winding of the coil. Tightness and compactness of the coil are controlled by the tension.

Consistent tension is required during coil winding. Among the winding tension factors including bobbin size, bobbin shape, number of turn, winding speeds, acceleration and deceleration rates. No matter how sophisticated the motor controls has, the material easily to get wander during winding and causing irregular spaced turns when insufficient tension occur. The diameter of the wire will reduce hence reducing the strength of the wire and make the wire tends to breakage if there is too much tension. Other than that, excessive tension can cause to change the wire characteristics and damage the bobbin's surface during winding process. Major characteristic of coil is resistance. Coil that randomly wounds in the same number of turn will produce different wire lengths. . If a coil is wound with good tension it is compact and it has less wire on it then a coil wound with incorrect low tension that results in a coil with longer total length of wire [5].The longer the coil wire has, the higher the resistance of the coil.

Control the speed of the winding is also included in tension control techniques. Basic winding speed tension control is shown in figure 2.4.1 below. The wire tension through a tension pulley connected to a spring will continuously monitored by the system. The position of the pulley is output to a controller which compares this tension with the preset tension and drives the motor accordingly, speeding up for a drop and slowing down for an increase in tension [7].

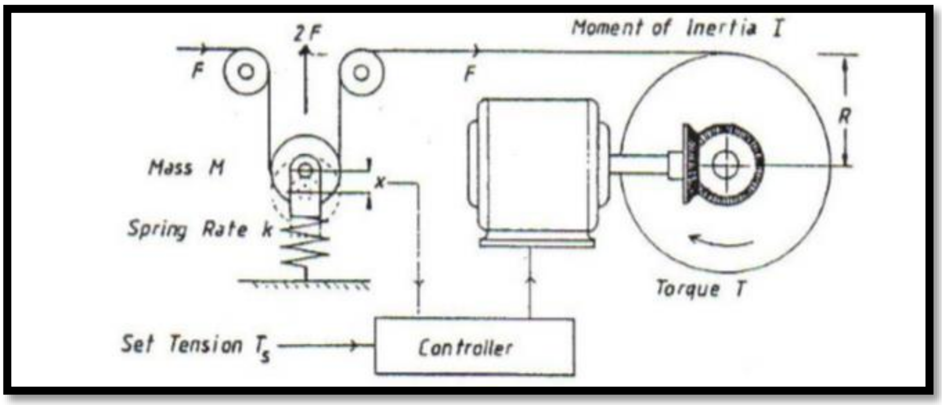


Figure 2.2: Basic winding speed tension control system [7].

2.5 Function and advantages of Stepper motor

Stepper motor is brushless, synchronous electric motor that converts digital pulses into mechanical shaft rotation. There are two types of stepper motor which is bipolar and unipolar. In this project, unipolar stepper motor will used to move the carriage that controls the coil position and the round bobbin. The unidirectional and unipolar stepper motor will become outstanding as the result of its low costs, stable performance, suitable for computer control and could be driven by unipolar stepper motor [1]. Besides that, there are additional advantages of stepper motor such as low inertia, large torque and high respond frequency.

The advantages of stepper motors over stopple AC or DC motors include no feedback requirement for position or speed control (open loop operation), noncumulative positional errors, precise electronic speed control using digital technology and compact size for driving large loads at low speeds [6].

The rotary motion of stepper motor can be converted to linear motion using a lead screw drive system as shown in figure below. The lead, or pitch, of the lead screw is the linear distance traveled for one revolution of the screw. If the lead is equal to one inch per revolution, and there are 200 full steps per revolution, then the resolution of the lead screw system is 0.005 inches per step.

2.6 Arduino with Stepper motor

Arduino is a tool for making computers that can sense and control more of the physical world than your desktop computer. It's an open-source physical computing platform based on a simple microcontroller board, and a development environment for writing software for the board. Arduino can be used to develop interactive objects, taking inputs from a variety of switches or sensors, and controlling a variety of lights, motors, and other physical outputs. Arduino projects can be stand-alone, or they can be communicate with software running on your computer (e.g. Flash, Processing, MaxMSP.) Stepper motors fall somewhere in between a regular DC motor and a servo motor. They have the advantage that they can be positioned accurately, moved forward or backwards one 'step' at a time, but they can also rotate continuously.

The Arduino programming environment is easy-to-use for beginners, yet flexible enough for advanced users to take advantage of as well. For teachers, it's conveniently based on the Processing programming environment, so students learning to program in that environment will be familiar with the look and feel of Arduino. [8]

CHAPTER 3

METHODOLOGY

3.1 Introduction

In developing a project, methodologies is one of the most important element to be consider to make sure that the development of the project is smooth and get the expected result. A good methodologist can be described the structure or the flow of the project where by it can be the guideline in managing it. It is also to avoid the project to alter course from the objectives that have been started or in order words the project follow the guideline based on the objectives.

In this chapter, the methodology is divided into three parts which is mechanical, electrical and programming coding.

3.2 Project Flow chart

Flow chart of work methodology can be simplified as shown in Figure 3.1.

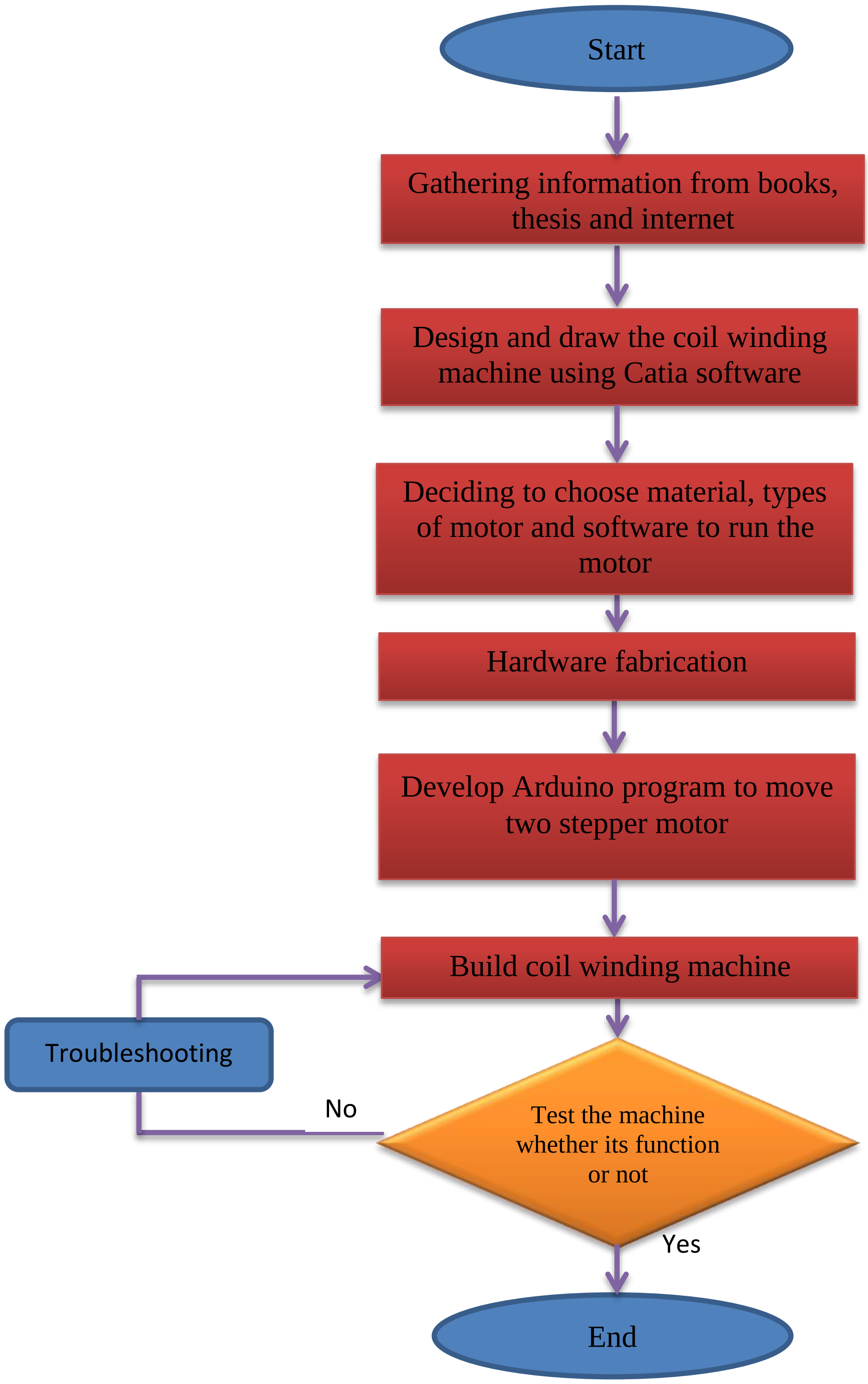


Figure 3.1: Flow chart of Methodology

3.3 Block diagram

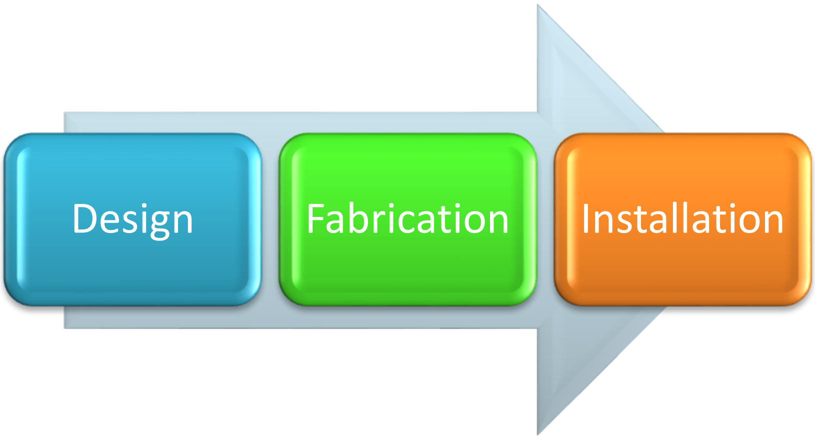


Figure 3.2: Process block diagram

As shown in a block diagram above, there are three important steps to execute this project, which are design, fabrication and installation. Design in the crucial part before fabrication. Without a proper design this project cannot proceed to another step and to fulfil this project. After researching in journals, books and internet on how to design a good quality of coil winding machine, the design have drawn using Catia software. After completing the project drawing, then do a selection for which material are suitable to fabricate.

Types of motor and types of program to control motor motion are also included in the selection. After done with the selection, the next step is fabrication. On fabrication, there are divided into two sections, which is hardware and software. Hardware also divided into two parts which is mechanical and electrical. Arduino program will be used in this project to control the motion of the motor and parallel interface as the connector between the hardware and the software. The final steps are installation. In installation process, the hardware and the software will be installed together to build a coil winding machine.

3.4 Mathematical Model

The tension is held constant as the bobbin is stopped. Figure 3.3 is a sketch of this wire and bobbin. Figure 3.4 shows a free body diagram of the first wire segment to make contact with the bobbin.

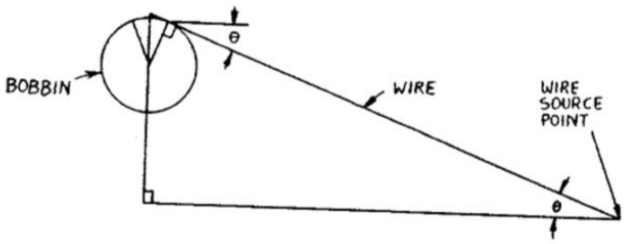


Figure 3.3: Sketch of the wire and the bobbin [2]

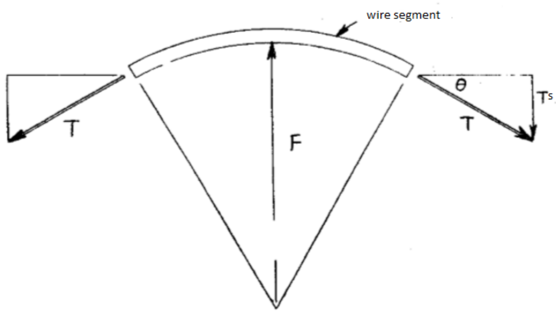


Figure3.4: Body diagram of the first wire segment to make contact with the bobbin [2]

Since the wire was wound at constant tension and there has been no slip in the wire, the tension 0 is the same on both ends of the segment. The included angle (measured in radians) between the segment edges is 20, so the angles between the tension forces and horizontal are both equal to 8 radians. The net force of the bobbin on the segment is

$$F = 2(T \times \sin \Theta) \tag{3.1}$$

The force will approach zero as the angle grows smaller, but the ratio of the force to the length of the arc (τ_1) does not. Since the arc length is equal to 20 times the radius (r):

$$\tau_1 = \text{limit as } \frac{\Theta + 0 \text{ of } (2\,\text{T} \sin\Theta)}{2\,\Theta\,\text{r}}$$
$$= \frac{T}{r} \text{ limit as } \Theta + \Theta \text{ of} (\sin\frac{\Theta}{\Theta})$$
$$= \frac{T}{r} \qquad (3.2)$$

Failure analysis of broken coils has shown that the tensions normally recommended by wire manufacturers may be too high for noncircular coils.' Instead, assume the tension should cause a stress in the wire equal to 50% of the yield strength (Y) of the material. The tension can be expressed as a function of the wires cross sectional area (A):

$$\text{T} = 0.5 \times Y \times A \qquad (3.3)$$

So,

$$\tau_1 = \frac{(0.50 \times \text{Y} \times \text{A})}{\text{r}} \qquad (3.4)$$

3.4.1 Tension Control System

The previous calculation of the coil tightening force assumed that the tension was constant while the coil was wound, so the tension forces on both ends of the segment were equal. However, when a bobbin is accelerating the tension must be different through the wire or it wouldn't accelerate. Consider a bobbin with a 20 mm diameter that is accelerated smoothly from being at rest to 10,000 $\frac{\text{r}}{\text{min}}$ in 2 seconds. At 10,000$\frac{\text{r}}{\text{min}}$, the wire velocity (v) at the source is:

$$\text{v} = 10000 * 0.02 * \pi = 628 \frac{\text{m}}{\text{min}} = 10.5 \frac{\text{m}}{\text{s}} \qquad (3.5)$$

So the acceleration would be $5.25\frac{m^2}{s}$. The accelerating force on a wire segment would be the difference in tension at its ends.

3.5 Mechanical component

3.5.1 Bobbin

As show in figure 3.5 below, bobbin in coil winding machine used to hold the coil. In this project, bobbin will be fabricated using rapid prototyping. Firstly, design the bobbin using Catia software and transfer to the rapid prototyping machine. The bobbin will rotate anti-clockwise. The Length of the bobbin is 1.5inch. The width is 1.0inch. The holes diameter is 0.1875inch. The outer diameter is 1.25inch and the inner diameter is 0.25 inch.

Figure 3.5: Bobbin [14]

3.5.2 Lead screw with long nut

In this project, lead screw functions as to convert stepper motor rotary motion to linear motion as show in figure 3.6 below. The diameter of the lead screw is 5/16 inch and the length is 13inch.

Figure 3.6: Lead screw with long nut. [12]

3.5.3 Aluminum Rod

In this project, brass tube will be used to place the large coil, wound coil and carriage. Advantage of using aluminum rod in this project is it can easily to drill and lighter than steel rod. The diameter of the aluminum rod is 0.1875 inch and the length is 13 inch. Aluminum rod is shown in figure 3.7 below.

Figure 3.7: Aluminium Rod [9]

3.5.4 Aluminium Plate

Figure 3.8: Aluminum Plate [10]

Plywood is used as a base and wall to hold the entire component such as motor, brass tube and lead screw. Plywood has been chosen in this project because of its easy to drill and easily formed.

3.5.5 Guide pulley

Coil winding Wire guide pulley (Wire roller) is commonly used component when wire passing is required. In this project, it needs three guide pulleys to ensure the wire tight before the bobbin winds the coil and the wire passing by smoothly. There are three guide pullet with three different size which is 9mm, 13mm and 19 mm of pulley diameter. Figure 3.9 shown below the guide pulley is been attached at the wire carriage.

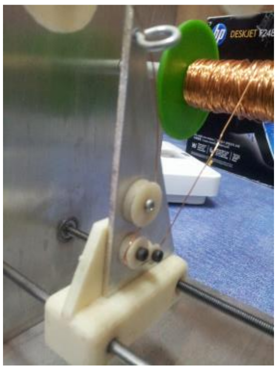


Figure 3.9: Wire Carriage

3.5.6 Bearing

A bearing is a machine element that constrains relative motion between moving parts to only the desired motion. In this project, two bearing is being used and the bearing diameter is 6mm

Figure 3.10: Bearing

3.5.7 Motor Coupling

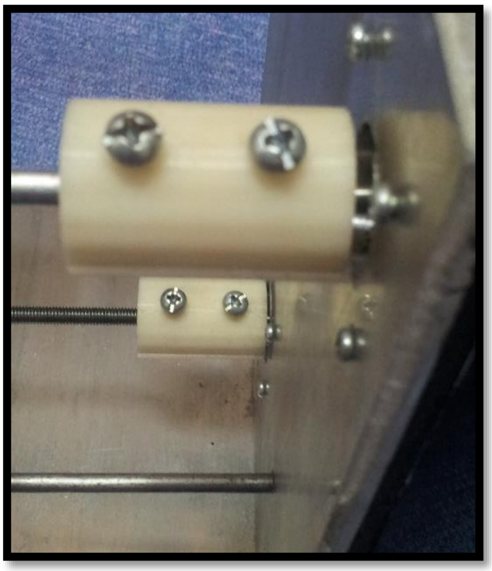


Figure 3.11: Motor Coupling

Couplings are good at making sure the motor shaft fastens to the lead screw. Couplings are simply used to enable a motor to turn a lead screw, or other type of shaft. There are two types of couplings, rigid (one piece), and flexible (an assembly of pieces). In this project, two rigid types of coupling are being used. The coupling is been done using rapid prototyping machine. Maximum shaft diameter that can be used is 6mm.

3.5.8 Angle Bar

The purpose of using angle bar in this project is to hold two wall at 90 degree angle. This project required two angle bar with 1mm of thickness and 203.2mm length.

Figure 3.12: Angle bar

3.5.9 Wire Copper coil

Copper coil is a main material for coil winding machine to produce a winding coil. In this project, wire diameter that being used is 1/4 inch.

Figure 3.13: Wire Copper coil

3.6 Electrical component

3.6.1 Stepper Motor

Figure 3.14: Stepper motor [8]

There are two stepper motor that will be used in this project with different types of motion which is;

i. **Winder motor**

Winder motor will move in rotary motion will be used in this project to move the bobbin either in anti-clockwise direction.

ii. **Carriage motor**

The carriage motor will be functions as to move the carriage in linear motion. The rotary motion of a stepper motor can be converted to linear motion using a lead screw. The linear motor is used to control the position of the coil winder.

3.6.2 Arduino UNO

Figure 3.15: Arduino UNO [13]

The Arduino Uno is a microcontroller board based on the ATmega328. (It has 14 digital input/output pins (of which 6 can be used as PWM outputs), 6 analog inputs, a 16 MHz ceramic resonator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. In this project, Arduino software will be used to control two stepper motor movement either in clockwise direction and anti-clockwise direction. Using Arduino program it can control the acceleration, speed and steps in one revolution of stepper motror.

3.6.3 L298N

The L298N is a high voltage, high current, dual full-bridge motor driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. In this project L298N is used to control stepper motor movement. This project required two L298N since one IC only can run one stepper motor. There is an advantage using this motor driver it because it can stand the heat and also the high current supply from the Arduino.

Features:

- Operating supply voltage of up to 46V
- 4.5-7VDC logic supply voltage
- Total DC current of up to 4A
- Low saturation voltage
- Over-temperature protection
- Logical '0' input up to 1.5V (high-noise immunity)

Figure 3.16: L298D [11]

3.7 Circuit design

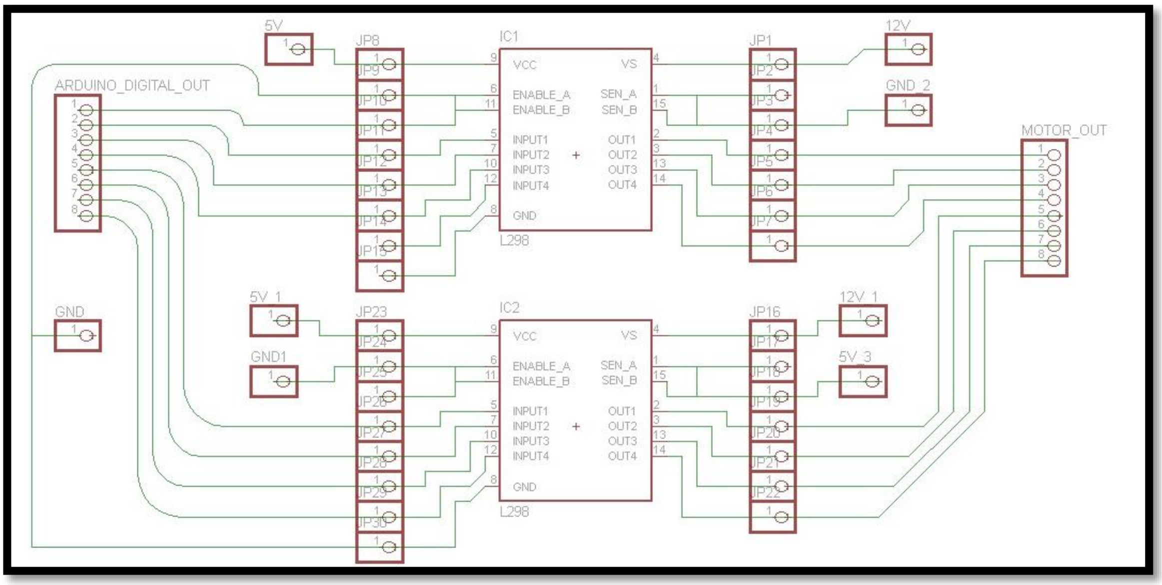


Figure 3.17: Circuit design

In this project, L298N was used to prevent over amount of voltage supply to the motherboard and also it can stand the high heat and high current supply from the Arduino. In this project required two L298N IC because one IC only can run one stepper motor since the stepper motor and the IC itself has four input and 4 output.

3.8 Machine Operation flow

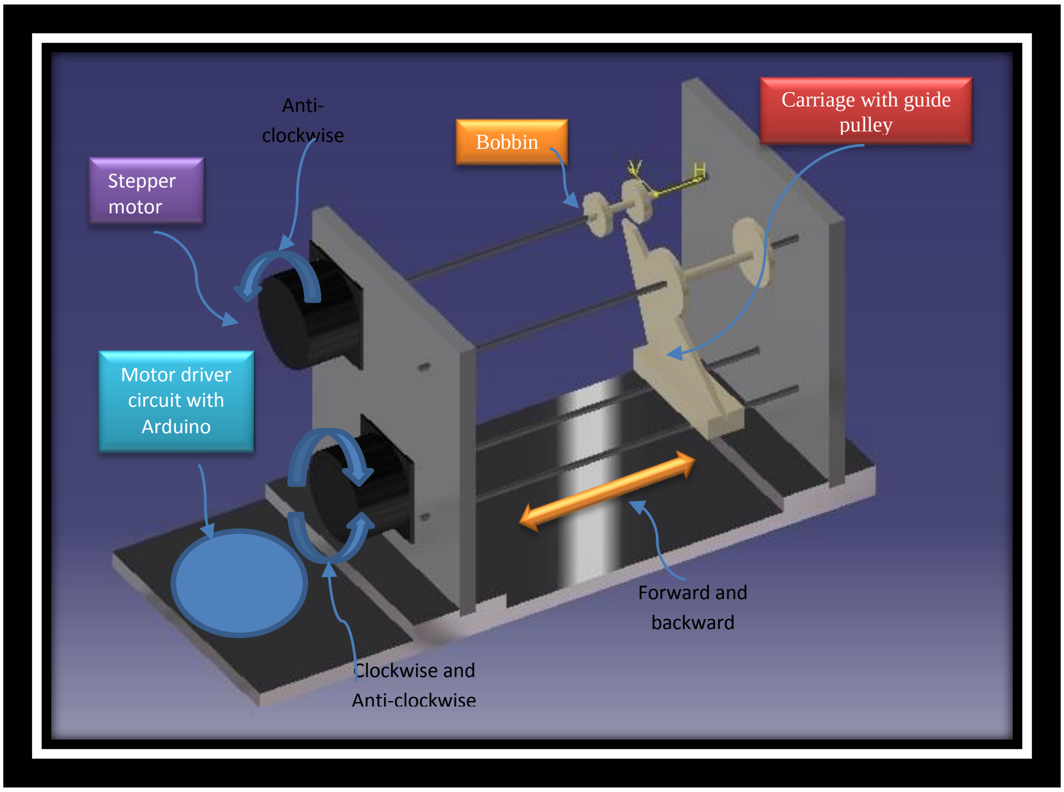


Figure 3.18: Project design

Before starting the machine, first is setting the wire from the main coil passing through the four guide pulley and tie the coil wire at the bobbin pin. After that, on switch at the electric box, the carriage motor will start to turn the shaft and at once the bobbin turn in anti-clockwise direction. After finish one round turn, while the wound motor still rotating, the carriage motor turn on and turn the shaft in clockwise direction causing the carriage to move forward and back to the intial position. Click the switch to stop the machine.

3.9 Programming on Arduino

```
#include <AccelStepper.h>

// Define some steppers and the pins the will use
AccelStepper stepper1(6,7,8,9 ); // Defaults to 4 pins on 2, 3, 4, 5
AccelStepper stepper2( 2,3,4,5);


void setup()
{
    stepper1.setAcceleration(80.0);
    stepper1.moveTo(10000);

    stepper2.setMaxSpeed(40.0);
    stepper2.setAcceleration(100.0);
    stepper2.moveTo(-3000.0);

}

void loop()
{
    // Change direction at the limits
    if (stepper1.distanceToGo() == 0)
        stepper1.moveTo(-stepper1.currentPosition());
    stepper1.run();
    stepper2.run();

}
```

Figure 3.19: Arduino Program to control two stepper motor

The following sketch uses the Serial Monitor, so once the sketch is installed and running, open the Serial Monitor and enter a number of 'steps'. Try a value of about 500, this should cause the motor to turn through about 360 degrees.

stepper2.moveTo (-500.0);

Enter -500 and it will turn back in the reverse direction. In Arduino there are library for moving stepper motor, the following command is used:

#include <AccelStepper.h>

This makes the process of using a motor very easy. After including the ‘Stepper’ library, the four control pins ‘in1’ to ‘in4’ are defined. To tell the Arduino Stepper library which pins are connected to the motor controller, the following command is used:

AccelStepper stepper1(6,7,8,9);

AccelStepper stepper2(2,3,4,5);

In this project there is two stepper motor is been used. Pin 2 to 5 is for carriage motor and pin 6 to 9 is for bobbin motor. The first parameter is the number of 'steps' that the motor will take to complete one revolution. The motor can be moved by one step at a time, for very fine positioning. Serial communications is then started, so that the Arduino is ready to receive commands from the Serial Monitor.

stepper2.setMaxSpeed(40.0);

Finally the following command sets the speed that we wish the stepper motor to move, when we subsequently tell it how many steps to rotate.

if (stepper1.distanceToGo() == 0)

stepper1.moveTo(-stepper1.currentPosition( ));

The 'loop' function is very simple. It waits for a command to come in from the Serial Monitor and converts the text of the number sent into an int using 'parseInt'. It then instructs the motor to turn that number of steps and return to the current position.

CHAPTER 4

RESULT AND DISCUSSION

4.1 Introduction

This chapter is containing the hardware fabrication of the winding machine, coding and its explanation for the Arduino and coil winding product. This chapter also containing discuss all the results obtained and the limitation of the project. All discussions are concentrating on the result and performance of the coil winding machine.

4.2 Hardware fabrication

This hardware fabrication is containing both mechanical and electrical implementation. The mechanical is the winding machine itself and the electrical part is the motor driver circuit for the stepper. Using L239D as a motor drive, this will give use to give high current to supply to the stepper motor because this integrated circuit (ic) can stand high current up to 3 ampere .

The power supply is from the Arduino itself, by plug in 12V 2A adapters, the supply will go out to Vin in Arduino. This will be used a power source to move the motor. Figure 4.0; shown the motor driver circuit for this winding machine. Digital out in Arduino was use to generate movement of the stepper motor.

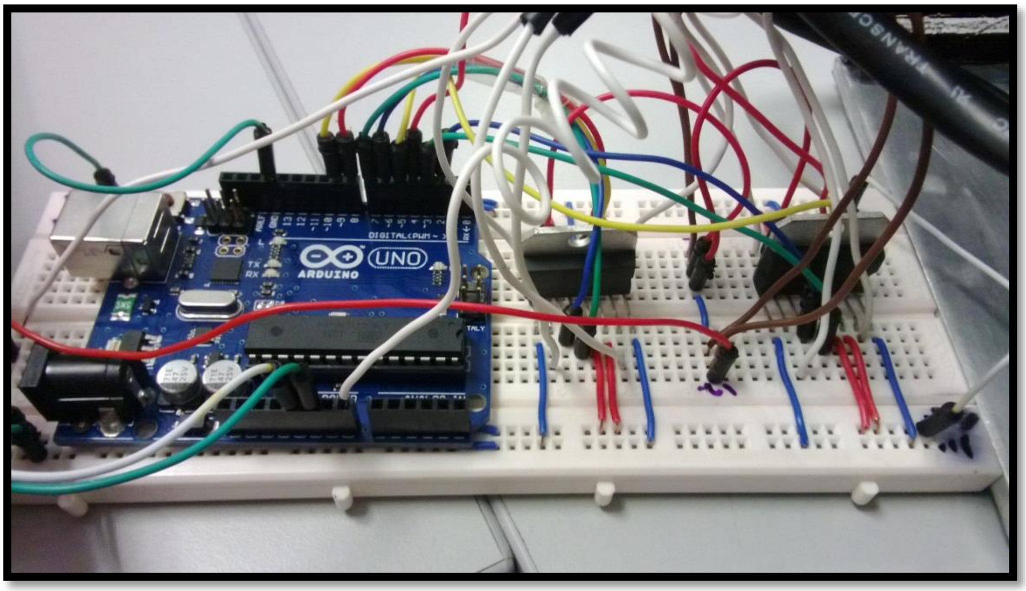


Figure 4.1: Motor driver circuit

Figure 4.2: shown mechanism of movement. The first stepper motor is coupled a lead screw to change the rotational to linear movement that is for the carriage. The stepper motor 2 is to wind the coil in the bobbin at the other side.

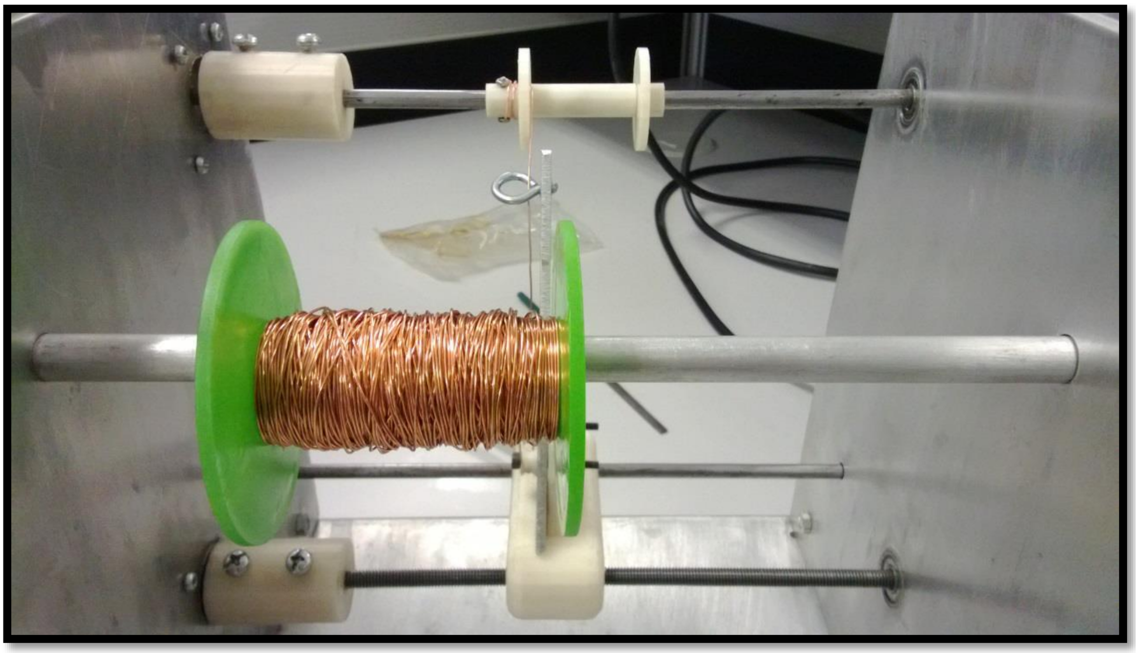


Figure 4.2: Mechanism of movement

The complete fabrication is shown below in figure 4.3 that including the winding machine and its circuit box as to protect the motor and the motor driver circuit.

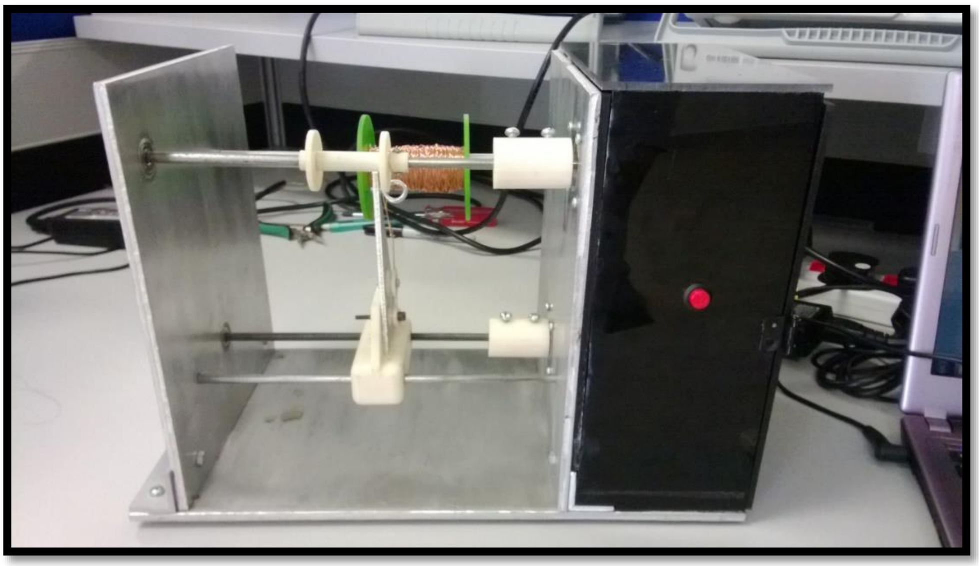


Figure 4.3: Complete fabrication of winding machine

4.3 Arduino coding

The library that was used was AccelStepper.h it use because to control the acceleration of the motor and also to use both stepper motor to move together. Table 4.1 was a brief explanation of the arduino coding that use in this winding machine project. The full coding refer to the Appendix E

Table 4.1: Coding explanation

Coding	**Explanations**
#include <AccelStepper.h>	Library used in this program
AccelStepper stepper1 (2,3,4,5) AccelStepper stepper2 (6,7,8,9)	Define the pin for the motor

stepper1.setMaxSpeed(50.0);
stepper1.setAcceleration(100.0);
stepper1.moveTo(-7950);
 	
stepper2.setMaxSpeed(70.0);
stepper2.setAcceleration(100.0);
stepper2.moveTo(-8150);
	The movement of the stepper motor
	First motor is the carriage and the second motor is for the wind motor. The speed is different because to synchronise the movement between the carriage and the winding motor.

if (stepper1.distanceToGo() == 0)
stepper1.moveTo(7600);
stepper1.run();
stepper2.run();
	This coding is to change direction at the limits. That mean when the motor wind finish winding, the motor at the carriage is going back to its original position.

4.4 Winding Product

Coil winding machine produces a product as shown below

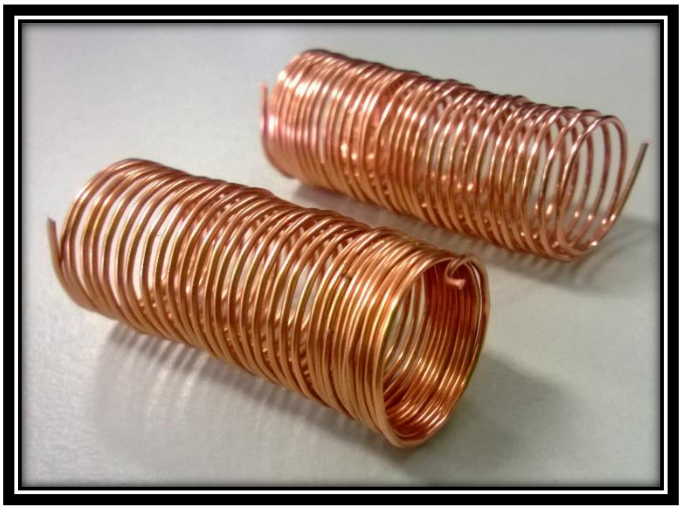


Figure 4.4: Coil Winding Machine Product

4.5 Discussion

There are a lot of problem during fabrication process, first when drilling a hole in on the wall that support the stepper motor, coil holder and carriage holder. If the hole midpoint is not parallel with the second wall it will affect the bobbin and the carriage movement. It will put a lot of friction when the motor is rotating and causing the stepper motor stop from rotating since the hole is not at the same level. To solve this problem by using Milling Machine to make sure the aluminum plate is drill correctly according to the dimension needed. This machine only can do one layer winding. The size and shape of the bobbin is limited to one shape which is round shape. To encounter this problem by design and fabricate a various size and shape of the bobbin.

In circuit preparation process, using ULN2083 is easy to connect the circuit to the Arduino Uno but there are a lot of disadvantage when using it. ULN 2083 cannot stand a high current supply from the Arduino. It only can support maximum current 1 Amps. If the current amount is more than 1 Amps the IC will damaged and cannot be used anymore. To solve this problem by using L298N. This types of motor drive can stand high current supply from Arduino up to 4 amps. It also has heat resistance that can reduce the heat when the IC is heating.

Other problem arises when creating a program in Arduino to move the carriage forward and backward direction and also the bobbin to make a complete one layer winding. Furthermore, it’s hard to make the carriage start and stop at the fixed point since there is no sensor is been used throughout this project. To encounter this problem, a lot of experiment have to be done by changing the motor step and motor speed in the program. It takes a lot of time to do the experiment just to ensure the carriage move as required.

CHAPTER 5

CONCLUSION AND RECOMMENDATIONS

5.1 Conclusion

In this project there are few objectives that need to be fulfilled. Based from the results and experiment done, the objectives of this project have been achieved. Design has been draw using Catia Software. The most important thing in building the Coil Winding Machine is the preparation on what kind of materials and machine that suitable to be use throughout the project.

In machining process which was milling operation itself, pocketing using end mill and drilling method had been used. Some part such as motor coupling and guide pulley is been draw in Catia software and produced by using Rapid Prototyping Machine. A good skills is required in handling a machine such as Milling Machine.

In circuit preparation, the most important thing is to find a suitable IC to use to create a motor drive circuit. In this project L298N has been used to control stepper motor. One L298N is compatible for one stepper motor only. So that in this project required two L298N to move two stepper motor at the same time. The advantage of using this IC compare to ULN 2803 is it can stand high current supply from the Arduino which is 2Amps and it also can stand with high heat compare to ULN 2803.

In this project to Arduino UNO program is been used to control two stepper motor at the same time. Compare to Visual Basic Program, Arduino is much easier to program and it is impossible to run two stepper motor at the same time by using Visual

Basic software. In Arduino Programming, it can control the speed, acceleration and step to be used to move the carriage motor and the bobbin motor.

5.2 Recommendations

For every project there will always have a further improvement can be done, in this project the coil winding machine only can do a coil winding for one layer only. To make a two layer and above, the bobbin wall must be steady to stand the pressure when the carriage move in forward and reverse direction. To make sure the carriage travel distance are according to the bobbin length, it must have limit switch or force sensor to make sure the carriage will stop at the end of the bobbin and reverse and stop again at the beginning of the bobbin. Size and shape of the bobbin also can be diversified. Arduino can be reprogram which it can winding a coil with different coil diameter, different shape of bobbin and size.

References

[1] Benhua, Z., Chenghua, L., Shiming, S., & Lu, G. (2011). Design on a unipolar and unidirectional stepper motor circuit. *Proceedings of 2011 International Conference on Electronic & Mechanical Engineering and Information Technology*, 1795–1797. doi:10.1109/EMEIT.2011.6023452

[2] Berg, T. A. Vanden. (1995). MANUFACTURING FLAT COIL CONFIGURATIONS FOR STEPPER MOTORS Ted A. Vanden Berg Ford Motor Company Dearborn, Michigan, 11–16. doi:10.1109/EEIC.1995.482328

[3] Delis, J. (1993). Design considerations, machinery and control options in coil winding. *Electrical Insulation Magazine, IEEE*, *9*(4), 16–22. doi:10.1109/57.223894

[4] Johnson, M. J. (1996). A PARALLEL PORT INTERFACE CIRCUIT FOR COMPUTER CONTROL APPLICATIONS, *2*, 889–892.

[5] Manning, T. A. (2003). TENSION CONTROL DURING THE COIL WINDING PROCESS Thomas A. Manning Sipro Inc., (23-25 Sept. 2003), 527–530. doi:10.1109/EICEMC.2003.1247942

[6] Montadar Abas Taher , Mohannad Jabbar Mnati, and H. F. M. (2008). POSITION CONTROL FOR FLIXEBLE JOINT MANIPULATOR USING ARTIFICIAL NEURAL Network, *01*(01), 122–138.

[7] Retrieved May 2, 2012, from:http://arduino.cc/en/Guide/Introduction

[8] [Stepper motor]. Retrieved December 3, 2012, from:
http://www.cytron.com.my/viewProduct.php?pcode=42BYGHD-234
http://www.cytron.com.my/index.php

[9] [Aluminium Rod]. Retrieved December 3, 2012, from: http://www.alumaloy.org/

[10] [Aluminium Plate]. Retrieved December 3, 2012, from:
http://www.soylualuminyum.com.tr/aluminium-plate.html/aluminium-plate-4

[11] [L298D]. Retrieved December 8, 2012, from:
http://www.engineersgarage.com/electronic-components/uln2003-datasheet

[12] [Lead screw with long nut]. Retrieved December 8, 2012, from:
http://www.lmbearings.com/index.php?p=lead-screws

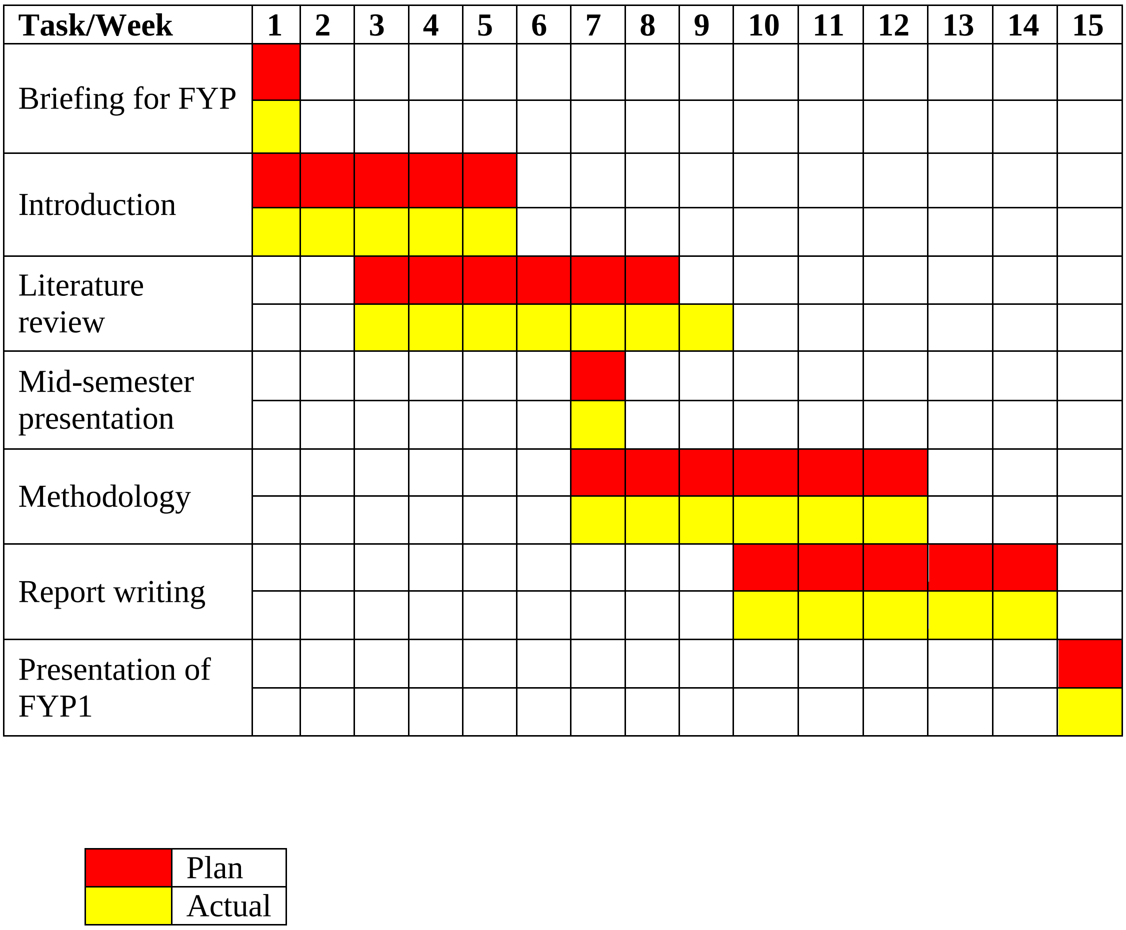
[13] [Arduino UNO]. Retrieved December 3, 2012, from:
http://tronixstuff.wordpress.com/2010/04/04/getting-started-with-arduino-chapter-zero/

[14] [Bobbin]. Retrieved December 3, 2012, from: http://www.made-in-china.com/showroom/jieru521hualiang/product-detailhAEnJWmKoYVt/China-Plastic-Spool-Bobbin-Container-PT-25-.html

Appendices

APPENDIX A : GANTT CHART FOR FYP 1

GANTT CHART



APPENDIX B: GANTT CHART FOR FYP 2

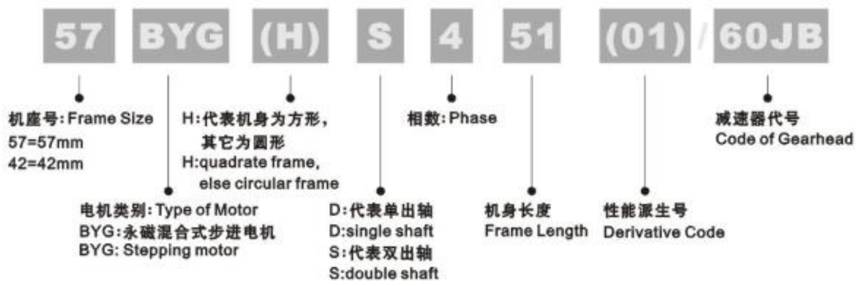
Task/Week		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
Proceed fabrication of coil winding machine	Plan	Plan	Plan	Plan	Plan	Plan	Plan	Plan	Plan	Plan	Plan	Plan	Plan						
	Actual	Actual	Actual	Actual	Actual	Actual	Actual	Actual	Actual	Actual	Actual	Actual	Actual						
Report writing (Result and Conclusion)	Plan							Plan	Plan	Plan	Plan	Plan	Plan	Plan	Plan	Plan	Plan	Plan	
	Actual							Actual	Actual	Actual	Actual	Actual	Actual	Actual	Actual	Actual	Actual		
Circuit preparation	Plan													Plan	Plan				
	Actual													Actual	Actual				
Create a program using Arduino	Plan														Plan	Plan			
	Actual														Actual	Actual			
Preparation of FYP Poster	Plan															Plan			
	Actual															Actual			
Presentation of FYP 2 (Poster)	Plan																Plan		
	Actual																Actual		
Record a video for video presentation	Plan																	Plan	
	Actual																	Actual	
Report and Video submission	Plan																		Plan
	Actual																		Actual

(red)	Plan
(yellow)	Actual

APPENDIX C : STEPPER MOTOR DATASHEET

LINIX® Since1968

电机的型号命名 MODEL NUMBER OF MOTOR



通用技术条件 GENERAL SPECIFICATION

冷态绝缘电阻：电机导电部分对机壳的冷态绝缘电阻不低于100MΩ；
…… Cold insulation resistance of theconductive part of the motor to the housing is no less than 100MΩ

绝缘介电强度：电机导电部分与机壳承受500V电压历时1min不击穿；
Insulation Dielectric strength: theconductive part and the housing of the motor withstand 500V lasting 1min.for pressureproof experiment but can not be punctured

温升＜80K …… Temperature rise：<80K

噪音＜50dB …… Noise：<50dB

工作环境要求 OPERATING CONDITIONS

海拔不超过2500米；…… Altitude within 2500m

环境温度－20℃－40℃；…… Ambient temperature:—20℃～40℃

相对湿度＜90%（温度20℃时）；…… Relative humidity≤90%(temperature 20℃)

贮存环境要求 STORAGE CONDITIONS

空气温度：－5℃－35℃；…… Ambient temperature: —5℃～35℃

相对湿度：＜70%；…… Relative humidity≤70%

通风良好；…… Good ventilation

通风良好，空气中不带腐蚀性气体；…… No corrosive gas present

严禁与强磁性物体并存；…… No strong magnetite object present

注意事项 ATTENTION

依照接线方式正确与驱动器接线，输入电压等级确保正确和正负极确保正确；…… to be sure correct input voltage grade

电机出轴与负载连接时不能重力敲打、挤压，以免造成内部变形走位，产生严重噪声或卡死现象；
……Don't beat or press when the motor connecting to the load to avoid internal deformation which will in turn cause noise and dead lock

步进电机 STEPPING MOTOR

42BYGH

LINIX® Since1968

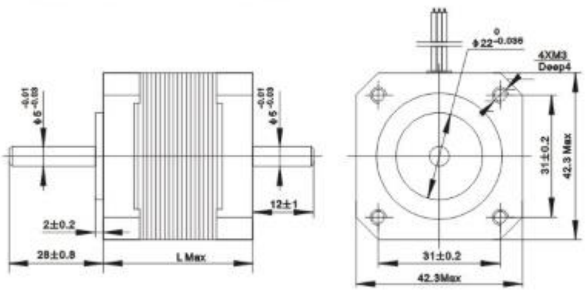
● 电机参数 SPECIFICATIONS

型号 Type	电流 Current A	相数 Phase	外形尺寸 L mm	电阻 Resistance Ω	静转矩 Hold torque Kg.cm	引出线 Lead wire
42BYGHD234-01A	0.4	2	34	30	2.0	4
42BYGHD433	0.5	4	34	10	2.0	6
42BYGHD239	0.8	2	39	15	2.5	4
42BYGHD439-02	0.8	4	39	7.5	2.5	6
42BYGHS444-02A	0.5	4	44	26	3.5	6
42BYGHS444-03A	0.5	4	44	15	3.5	6
42BYGHD248	0.8	2	48	7	3.2	4
42BYGHD448-05	1.2	4	48	3.3	4.0	6

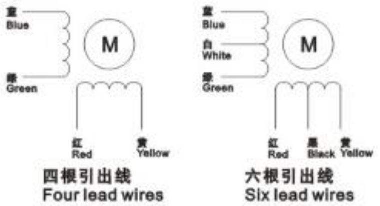
步距角 Step angle	绝缘电阻 Insulation resistance	绝缘强度 Insulation strength	环境温度 Ambient temperature	温升 Temperature rise	径向跳动 Radial jumpiness	轴向跳动 Axial jumpiness
1.8°±5%	500V DC 100MΩ Min	50Hz 1Minute 500V Min	-20℃~+40℃	80K Max.	0.02mm Max.	0.1-0.3mm

步进电机 STEPPING MOTOR

● 外形尺寸 EXTERNAL DIMENSION



● 接线图 WIRING DIAGRAM



以上仅为代表型产品数据，对于减速比和其他参数，我公司可视您的具体需要定做。上图为42BYGHS444-03A外形尺寸。
Every standard offering can be designed and tailored to meet your specific needs.

● 概述 SUMMARY

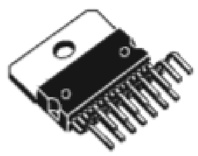
应用范围：该系列电机力矩大，噪音低，运转平稳，广泛应用于数控机床、打印机、绣花机、电脑织机、办公设备、光学设备、医疗设备等行业。
Featured with high torque, quiet, smooth running, it is popularly applied to numerically controlled machine, embroidery machine, computetrized kintting machine, optical apparatus, medical equipment.

APPENDIX D: L298N DATASHEET

L298

DUAL FULL-BRIDGE DRIVER

- OPERATING SUPPLY VOLTAGE UP TO 46 V
- TOTAL DC CURRENT UP TO 4 A
- LOW SATURATION VOLTAGE
- OVERTEMPERATURE PROTECTION
- LOGICAL "0" INPUT VOLTAGE UP TO 1.5 V (HIGH NOISE IMMUNITY)



Multiwatt15

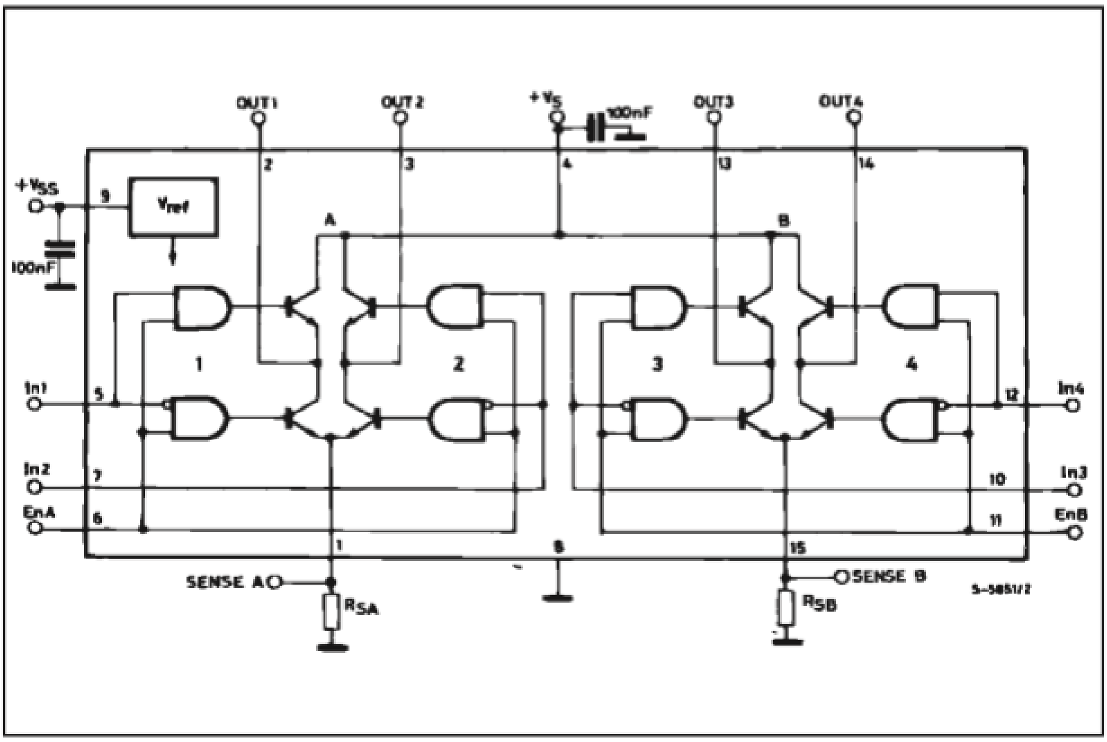
PowerSO20

ORDERING NUMBERS : L298N (Multiwatt Vert.)
L298HN (Multiwatt Horiz.)
L298P (PowerSO20)

DESCRIPTION

The L298 is an integrated monolithic circuit in a 15-lead Multiwatt and PowerSO20 packages. It is a high voltage, high current dual full-bridge driver designed to accept standard TTL logic levels and drive inductive loads such as relays, solenoids, DC and stepping motors. Two enable inputs are provided to enable or disable the device independently of the input signals. The emitters of the lower transistors of each bridge are connected together and the corresponding external terminal can be used for the connection of an external sensing resistor. An additional supply input is provided so that the logic works at a lower voltage.

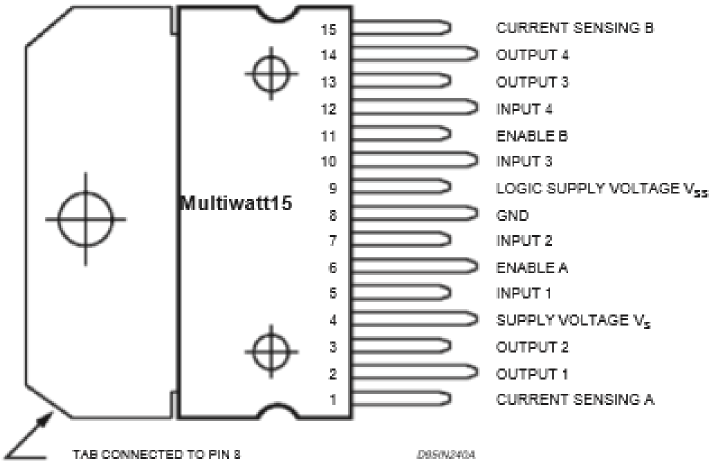
BLOCK DIAGRAM

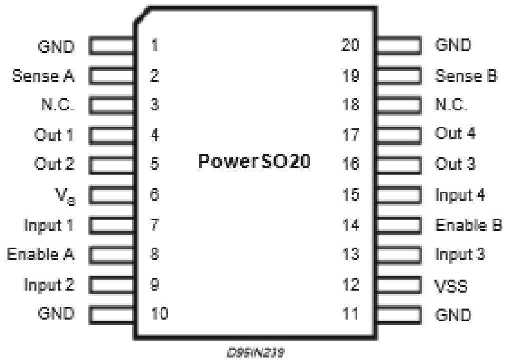


ABSOLUTE MAXIMUM RATINGS

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 = 100µs) –Repetitive (80% on –20% off; t_{on} = 10ms) –DC Operation	 3 2.5 2	 A A A
V_{sens}	Sensing Voltage	–1 to 2.3	V
P_{tot}	Total Power Dissipation (T_{case} = 75°C)	25	W
T_{op}	Junction Operating Temperature	–25 to 130	°C
T_{stg}, T_j	Storage and Junction Temperature	–40 to 150	°C

PIN CONNECTIONS (top view)





THERMAL DATA

Symbol	Parameter		PowerSO20	Multiwatt15	Unit
$R_{th\ j\text{-}case}$	Thermal Resistance Junction-case	Max.	–	3	°C/W
$R_{th\ j\text{-}amb}$	Thermal Resistance Junction-ambient	Max.	13 (*)	35	°C/W

PIN FUNCTIONS (refer to the block diagram)

MW.15	PowerSO	Name	Function
1;15	2;19	Sense A; Sense B	Between this pin and ground is connected the sense resistor to control the current of the load.
2;3	4;5	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	6	V_S	Supply Voltage for the Power Output Stages. A non-inductive 100nF capacitor must be connected between this pin and ground.
5;7	7;9	Input 1; Input 2	TTL Compatible Inputs of the Bridge A.
6;11	8;14	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	1,10,11,20	GND	Ground.
9	12	VSS	Supply Voltage for the Logic Blocks. A100nF capacitor must be connected between this pin and ground.
10; 12	13;15	Input 3; Input 4	TTL Compatible Inputs of the Bridge B.
13; 14	16;17	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.
–	3;18	N.C.	Not Connected

ELECTRICAL CHARACTERISTICS (V_S = 42V; V_{SS} = 5V, T_j = 25°C; unless otherwise specified)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
V_S	Supply Voltage (pin 4)	Operative Condition	V_{IH} +2.5		46	V
V_{SS}	Logic Supply Voltage (pin 9)		4.5	5	7	V
I_S	Quiescent Supply Current (pin 4)	V_{en} = H; I_L = 0 V_I = L V_I = H		13 50	22 70	mA mA
		V_{en} = L V_I = X			4	mA
I_{SS}	Quiescent Current from V_{SS} (pin 9)	V_{en} = H; I_L = 0 V_I = L V_I = H		24 7	36 12	mA mA
		V_{en} = L V_I = X			6	mA
V_{IL}	Input Low Voltage (pins 5, 7, 10, 12)		–0.3		1.5	V
V_{IH}	Input High Voltage (pins 5, 7, 10, 12)		2.3		VSS	V
I_{IL}	Low Voltage Input Current (pins 5, 7, 10, 12)	V_I = L			–10	μA
I_{IH}	High Voltage Input Current (pins 5, 7, 10, 12)	Vi = H ≤ V_{SS} –0.6V		30	100	μA
V_{en} = L	Enable Low Voltage (pins 6, 11)		–0.3		1.5	V
V_{en} = H	Enable High Voltage (pins 6, 11)		2.3		V_{SS}	V
I_{en} = L	Low Voltage Enable Current (pins 6, 11)	V_{en} = L			–10	μA
I_{en} = H	High Voltage Enable Current (pins 6, 11)	V_{en} = H ≤ V_{SS} –0.6V		30	100	μA
$V_{CEsat(H)}$	Source Saturation Voltage	I_L = 1A I_L = 2A	0.95	1.35 2	1.7 2.7	V V
$V_{CEsat(L)}$	Sink Saturation Voltage	I_L = 1A (5) I_L = 2A (5)	0.85	1.2 1.7	1.6 2.3	V V
V_{CEsat}	Total Drop	I_L = 1A (5) I_L = 2A (5)	1.80		3.2 4.9	V V
V_{sens}	Sensing Voltage (pins 1, 15)		–1 (1)		2	V

ELECTRICAL CHARACTERISTICS (continued)

Symbol	Parameter	Test Conditions	Min.	Typ.	Max.	Unit
T_1 (V_i)	Source Current Turn-off Delay	0.5 V_i to 0.9 I_L (2); (4)		1.5		μs
T_2 (V_i)	Source Current Fall Time	0.9 I_L to 0.1 I_L (2); (4)		0.2		μs
T_3 (V_i)	Source Current Turn-on Delay	0.5 V_i to 0.1 I_L (2); (4)		2		μs
T_4 (V_i)	Source Current Rise Time	0.1 I_L to 0.9 I_L (2); (4)		0.7		μs
T_5 (V_i)	Sink Current Turn-off Delay	0.5 V_i to 0.9 I_L (3); (4)		0.7		μs
T_6 (V_i)	Sink Current Fall Time	0.9 I_L to 0.1 I_L (3); (4)		0.25		μs
T_7 (V_i)	Sink Current Turn-on Delay	0.5 V_i to 0.9 I_L (3); (4)		1.6		μs
T_8 (V_i)	Sink Current Rise Time	0.1 I_L to 0.9 I_L (3); (4)		0.2		μs
fc (V_i)	Commutation Frequency	I_L = 2A		25	40	KHz
T_1 (V_{en})	Source Current Turn-off Delay	0.5 V_{en} to 0.9 I_L (2); (4)		3		μs
T_2 (V_{en})	Source Current Fall Time	0.9 I_L to 0.1 I_L (2); (4)		1		μs
T_3 (V_{en})	Source Current Turn-on Delay	0.5 V_{en} to 0.1 I_L (2); (4)		0.3		μs
T_4 (V_{en})	Source Current Rise Time	0.1 I_L to 0.9 I_L (2); (4)		0.4		μs
T_5 (V_{en})	Sink Current Turn-off Delay	0.5 V_{en} to 0.9 I_L (3); (4)		2.2		μs
T_6 (V_{en})	Sink Current Fall Time	0.9 I_L to 0.1 I_L (3); (4)		0.35		μs
T_7 (V_{en})	Sink Current Turn-on Delay	0.5 V_{en} to 0.9 I_L (3); (4)		0.25		μs
T_8 (V_{en})	Sink Current Rise Time	0.1 I_L to 0.9 I_L (3); (4)		0.1		μs

1) 1)Sensing voltage can be –1 V for t ≤ 50 μsec; in steady state V_{sens} min ≥ –0.5 V.
2) See fig. 2.
3) See fig. 4.
4) The load must be a pure resistor.

Figure 1 : Typical Saturation Voltage vs. Output Current.

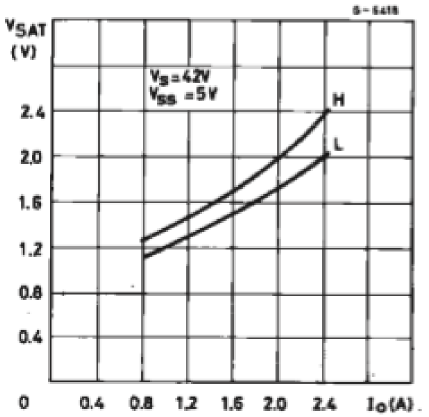
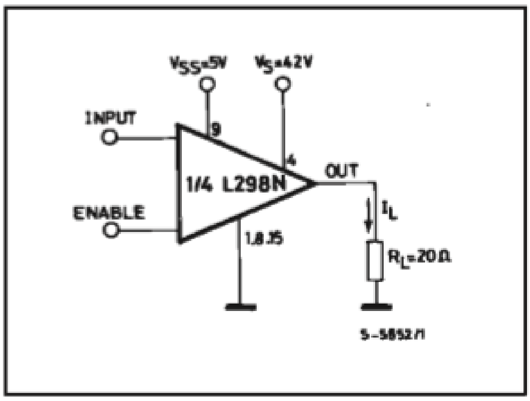


Figure 2 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
For ENABLE Switching, set IN = H

Figure 3 : Source Current Delay Times vs. Input or Enable Switching.

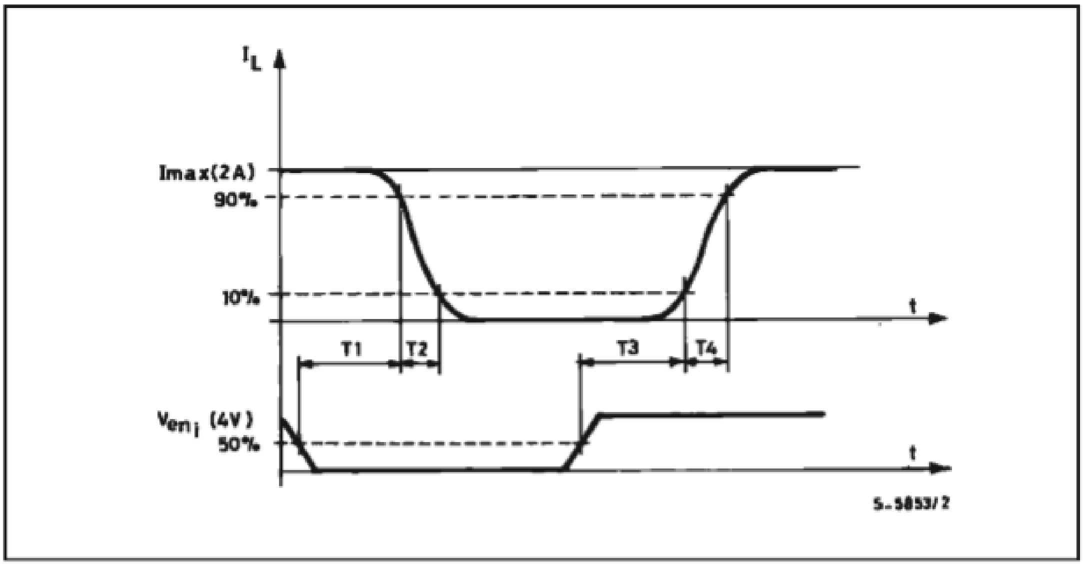
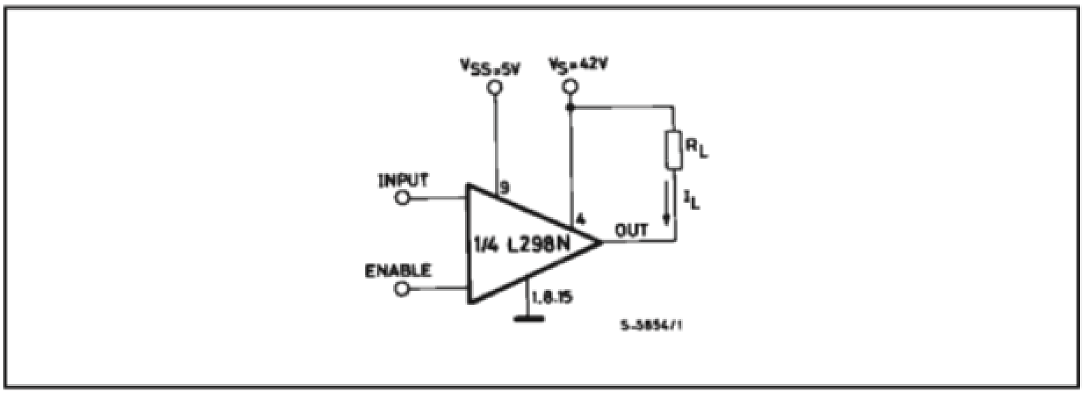


Figure 4 : Switching Times Test Circuits.



Note : For INPUT Switching, set EN = H
For ENABLE Switching, set IN = L

Figure 5 : Sink Current Delay Times vs. Input 0 V Enable Switching.

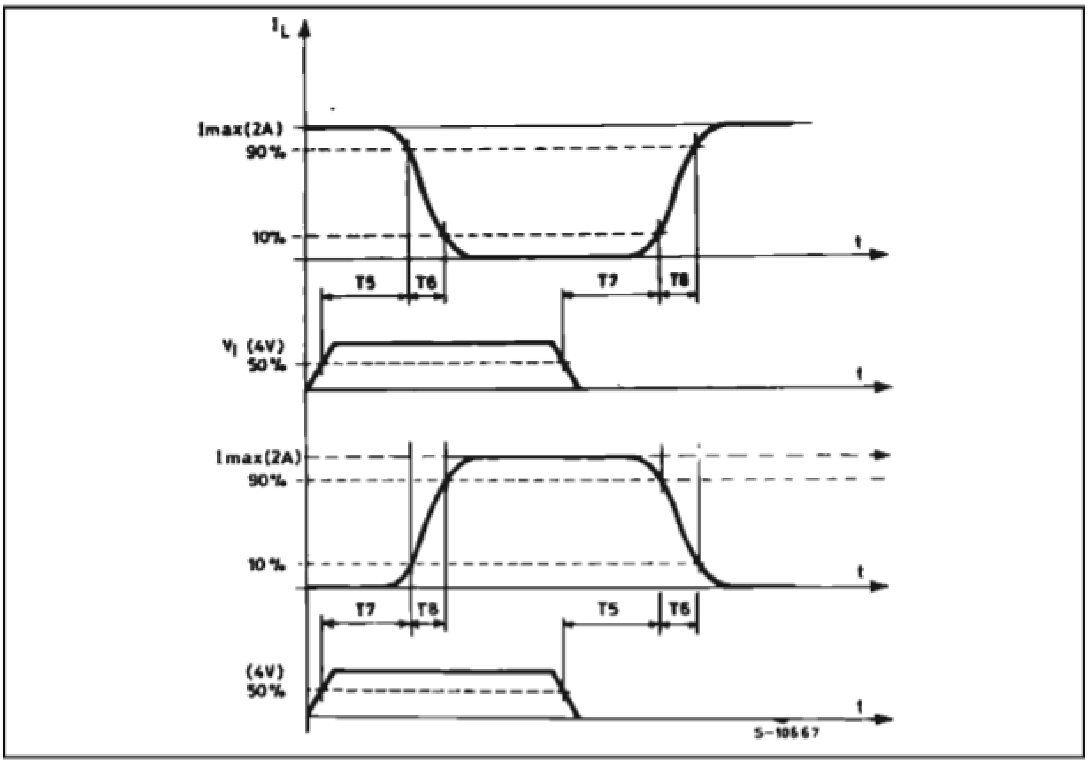
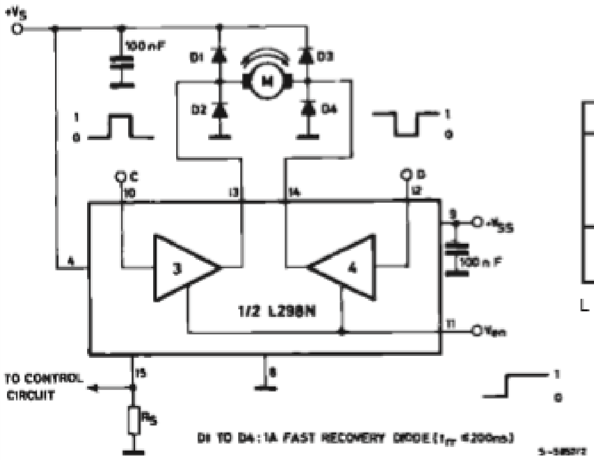


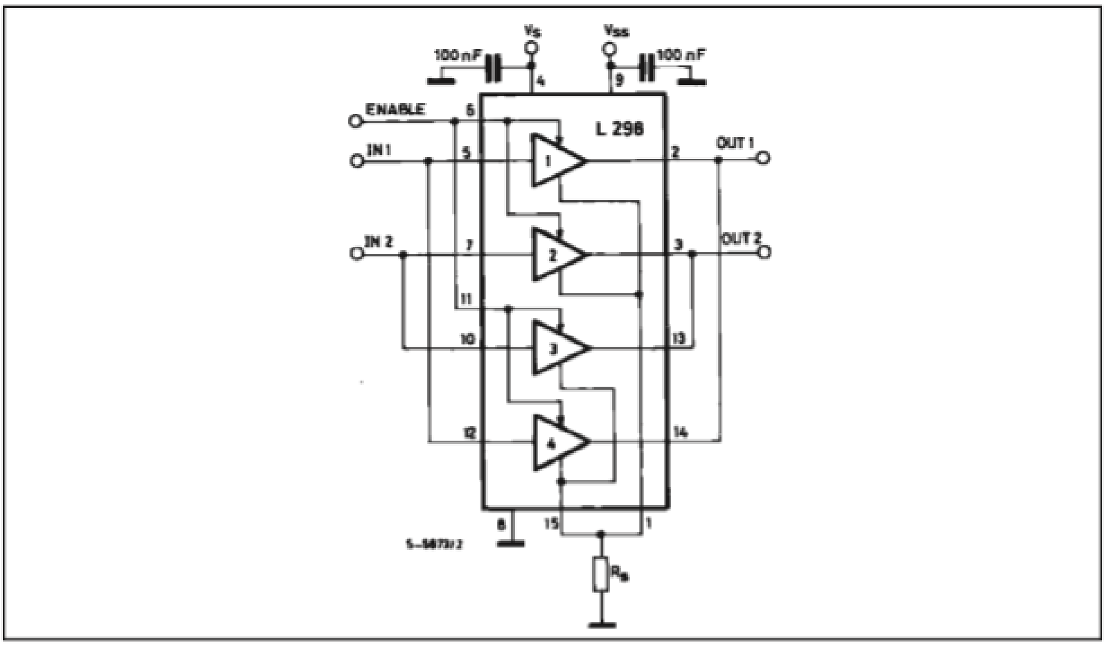
Figure 6 : Bidirectional DC Motor Control.



Inputs		Function
V_{en} = H	C = H ; D = L	Forward
	C = L ; D = H	Reverse
	C = D	Fast Motor Stop
V_{en} = L	C = X ; D = X	Free Running Motor Stop

L = Low H = High X = Don't care

Figure 7 : For higher currents, outputs can be paralleled. Take care to parallel channel 1 with channel 4 and channel 2 with channel 3.



APPLICATION INFORMATION (Refer to the block diagram)

1.1. POWER OUTPUT STAGE

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 differenzial 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.

1.2. INPUT STAGE

Each bridge is driven by means of four gates the input of which are In1 ; In2 ; EnA and In3 ; In4 ; EnB. The In inputs set the bridge state when The En input is high ; a low state of the En input inhibits the bridge. All the inputs are TTL compatible.

2. SUGGESTIONS

A non inductive capacitor, usually of 100 nF, must be foreseen between both Vs and Vss, to ground, as near as possible to GND pin. When the large capacitor of the power supply is too far from the IC, a second smaller one must be foreseen near the L298.

The sense resistor, not of a wire wound type, must be grounded near the negative pole of Vs that must be near the GND pin of the I.C.

Each input must be connected to the source of the driving signals by means of a very short path.

Turn-On and Turn-Off : Before to Turn-ON the Supply Voltage and before to Turn it OFF, the Enable input must be driven to the Low state.

3. APPLICATIONS

Fig 6 shows a bidirectional DC motor control Schematic Diagram for which only one bridge is needed. The external bridge of diodes D1 to D4 is made by four fast recovery elements ($t_{rr} \leq 200$ nsec) that must be chosen of a V_F as low as possible at the worst case of the load current.

The sense output voltage can be used to control the current amplitude by chopping the inputs, or to provide overcurrent protection by switching low the enable input.

The brake function (Fast motor stop) requires that the Absolute Maximum Rating of 2 Amps must never be overcome.

When the repetitive peak current needed from the load is higher than 2 Amps, a paralleled configuration can be chosen (See Fig.7).

An external bridge of diodes are required when inductive loads are driven and when the inputs of the IC are chopped; Shottky diodes would be preferred.

This solution can drive until 3 Amps In DC operation and until 3.5 Amps of a repetitive peak current.

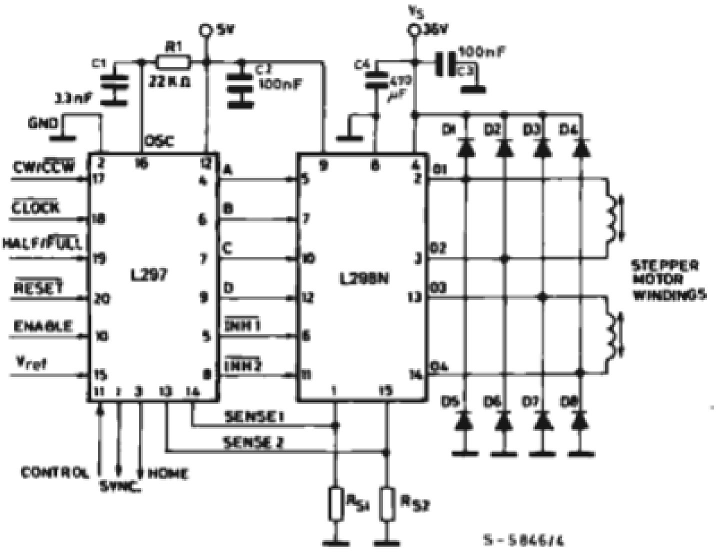
On Fig 8 it is shown the driving of a two phase bipolar stepper motor ; the needed signals to drive the inputs of the L298 are generated, in this example, from the IC L297.

Fig 9 shows an example of P.C.B. designed for the application of Fig 8.

Fig 10 shows a second two phase bipolar stepper motor control circuit where the current is controlled by the I.C. L6506.

Figure 8 : Two Phase Bipolar Stepper Motor Circuit.

This circuit drives bipolar stepper motors with winding currents up to 2 A. The diodes are fast 2 A types.



$R_{S1} = R_{S2} = 0.5\ \Omega$

D1 to D8 = 2 A Fast diodes { $V_F \leq 1.2$ V @ I = 2 A; trr $\leq$ 200 ns

APPENDIX E: ARDUINO CODING (FINAL)

_2_stepper | Arduino 1.0.1

File Edit Sketch Tools Help

_2_stepper

```
// MultiStepper.pde
// -*- mode: C++ -*-
//
// Shows how to multiple simultaneous steppers
// Runs one stepper forwards and backwards, accelerating and decelerating
// at the limits. Runs other steppers at the same time
//
// Copyright (C) 2009 Mike McCauley
// $Id: HRFMessage.h,v 1.1 2009/08/15 05:32:58 mikem Exp mikem $

#include <AccelStepper.h>

// Define some steppers and the pins the will use
AccelStepper stepper1; // Defaults to 4 pins on 2, 3, 4, 5
AccelStepper stepper2(4, 6, 7, 8, 9);


void setup()
{
    stepper1.setMaxSpeed(50.0);
    stepper1.setAcceleration(100.0);
    stepper1.moveTo(-7950);

    stepper2.setMaxSpeed(70.0);
    stepper2.setAcceleration(100.0);
    stepper2.moveTo(-8150);

}

void loop()
{
    // Change direction at the limits
    if (stepper1.distanceToGo() == 0)
        stepper1.moveTo(7600);
    stepper1.run();
    stepper2.run();

}
```

APPENDIX F: ARDUINO UNO DATASHEET

Features

- High Performance, Low Power AVR® 8-Bit Microcontroller
- Advanced RISC Architecture
  - 131 Powerful Instructions – Most Single Clock Cycle Execution
  - 32 x 8 General Purpose Working Registers
  - Fully Static Operation
  - Up to 20 MIPS Throughput at 20 MHz
  - On-chip 2-cycle Multiplier
- High Endurance Non-volatile Memory Segments
  - 4/8/16/32K Bytes of In-System Self-Programmable Flash progam memory (ATmega48PA/88PA/168PA/328P)
  - 256/512/512/1K Bytes EEPROM (ATmega48PA/88PA/168PA/328P)
  - 512/1K/1K/2K Bytes Internal SRAM (ATmega48PA/88PA/168PA/328P)
  - Write/Erase Cycles: 10,000 Flash/100,000 EEPROM
  - Data retention: 20 years at 85°C/100 years at 25°C[(1)]
  - Optional Boot Code Section with Independent Lock Bits
    In-System Programming by On-chip Boot Program
    True Read-While-Write Operation
  - Programming Lock for Software Security
- Peripheral Features
  - Two 8-bit Timer/Counters with Separate Prescaler and Compare Mode
  - One 16-bit Timer/Counter with Separate Prescaler, Compare Mode, and Capture Mode
  - Real Time Counter with Separate Oscillator
  - Six PWM Channels
  - 8-channel 10-bit ADC in TQFP and QFN/MLF package
    Temperature Measurement
  - 6-channel 10-bit ADC in PDIP Package
    Temperature Measurement
  - Programmable Serial USART
  - Master/Slave SPI Serial Interface

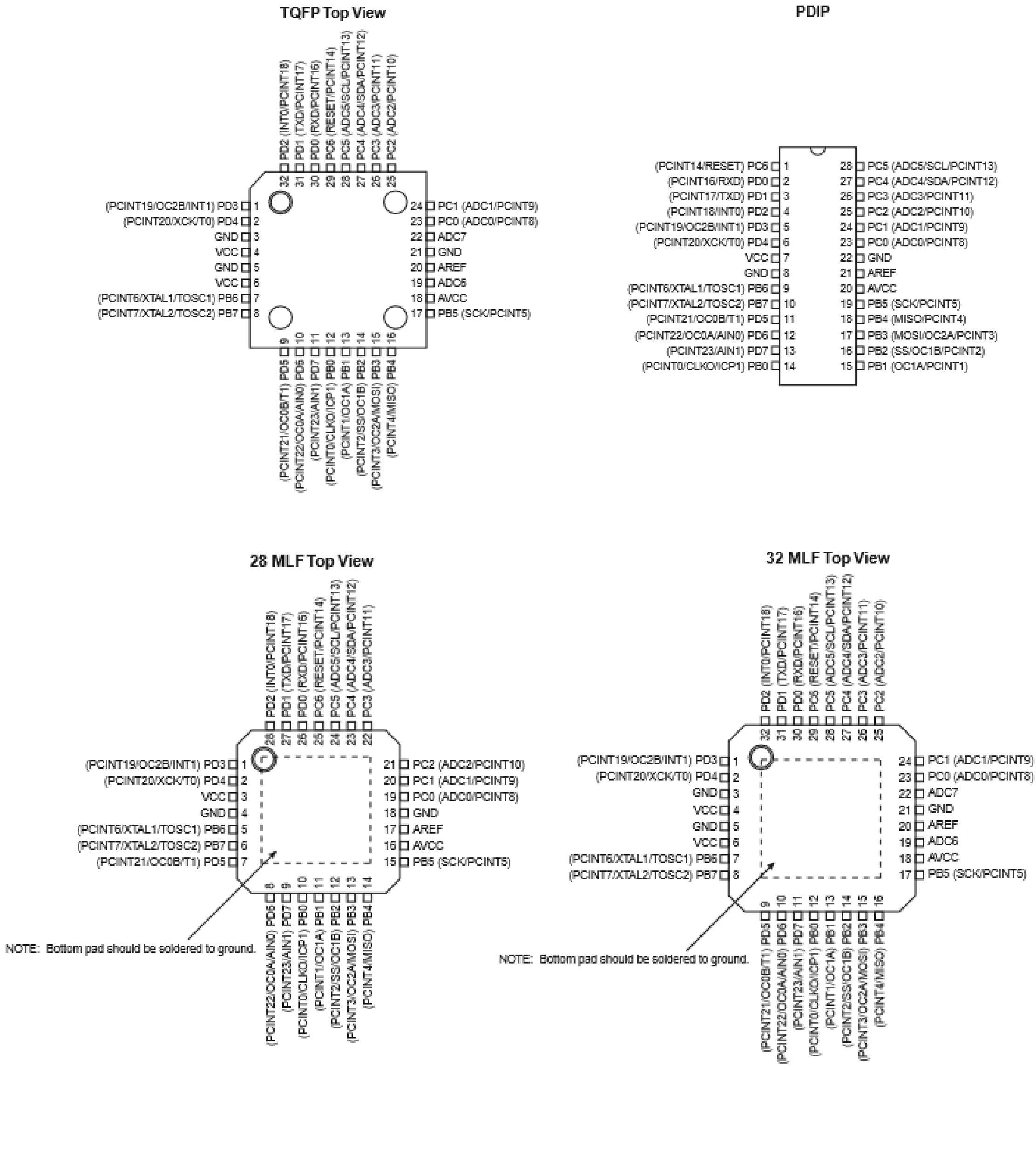
8-bit AVR® Microcontroller with 4/8/16/32K Bytes In-System Programmable Flash

ATmega48PA
ATmega88PA
ATmega168PA

  - Byte-oriented 2-wire Serial Interface (Philips I^{2}C compatible)
  - Programmable Watchdog Timer with Separate On-chip Oscillator
  - On-chip Analog Comparator
  - Interrupt and Wake-up on Pin Change
- Special Microcontroller Features
  - Power-on Reset and Programmable Brown-out Detection
  - Internal Calibrated Oscillator
  - External and Internal Interrupt Sources
  - Six Sleep Modes: Idle, ADC Noise Reduction, Power-save, Power-down, Standby, and Extended Standby
- I/O and Packages
  - 23 Programmable I/O Lines
  - 28-pin PDIP, 32-lead TQFP, 28-pad QFN/MLF and 32-pad QFN/MLF
- Operating Voltage:
  - 1.8 - 5.5V for ATmega48PA/88PA/168PA/328P
- Temperature Range:
  - -40°C to 85°C
- Speed Grade:
  - 0 - 20 MHz @ 1.8 - 5.5V
- Low Power Consumption at 1 MHz, 1.8V, 25°C for ATmega48PA/88PA/168PA/328P:
  - Active Mode: 0.2 mA
  - Power-down Mode: 0.1 µA
  - Power-save Mode: 0.75 µA (Including 32 kHz RTC)

1. Pin Configurations

Figure 1-1. Pinout ATmega48PA/88PA/168PA/328P



1.1 Pin Descriptions

1.1.1 VCC

Digital supply voltage.

1.1.2 GND

Ground.

1.1.3 Port B (PB7:0) XTAL1/XTAL2/TOSC1/TOSC2

Port B is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port B output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port B pins that are externally pulled low will source current if the pull-up resistors are activated. The Port B pins are tri-stated when a reset condition becomes active, even if the clock is not running.

Depending on the clock selection fuse settings, PB6 can be used as input to the inverting Oscillator amplifier and input to the internal clock operating circuit.

Depending on the clock selection fuse settings, PB7 can be used as output from the inverting Oscillator amplifier.

If the Internal Calibrated RC Oscillator is used as chip clock source, PB7..6 is used as TOSC2..1 input for the Asynchronous Timer/Counter2 if the AS2 bit in ASSR is set.

The various special features of Port B are elaborated in "Alternate Functions of Port B" on page 82 and "System Clock and Clock Options" on page 26.

1.1.4 Port C (PC5:0)

Port C is a 7-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The PC5..0 output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port C pins that are externally pulled low will source current if the pull-up resistors are activated. The Port C pins are tri-stated when a reset condition becomes active, even if the clock is not running.

1.1.5 PC6/$\overline{\text{RESET}}$

If the RSTDISBL Fuse is programmed, PC6 is used as an I/O pin. Note that the electrical characteristics of PC6 differ from those of the other pins of Port C.

If the RSTDISBL Fuse is unprogrammed, PC6 is used as a Reset input. A low level on this pin for longer than the minimum pulse length will generate a Reset, even if the clock is not running. The minimum pulse length is given in Table 28-3 on page 318. Shorter pulses are not guaranteed to generate a Reset.

The various special features of Port C are elaborated in "Alternate Functions of Port C" on page 85.

1.1.6 Port D (PD7:0)

Port D is an 8-bit bi-directional I/O port with internal pull-up resistors (selected for each bit). The Port D output buffers have symmetrical drive characteristics with both high sink and source capability. As inputs, Port D pins that are externally pulled low will source current if the pull-up resistors are activated. The Port D pins are tri-stated when a reset condition becomes active, even if the clock is not running.

The various special features of Port D are elaborated in "Alternate Functions of Port D" on page 88.

1.1.7 AV_{CC}

AV_{CC} is the supply voltage pin for the A/D Converter, PC3:0, and ADC7:6. It should be externally connected to V_{CC}, even if the ADC is not used. If the ADC is used, it should be connected to V_{CC} through a low-pass filter. Note that PC6..4 use digital supply voltage, V_{CC}.

1.1.8 AREF

AREF is the analog reference pin for the A/D Converter.

1.1.9 ADC7:6 (TQFP and QFN/MLF Package Only)

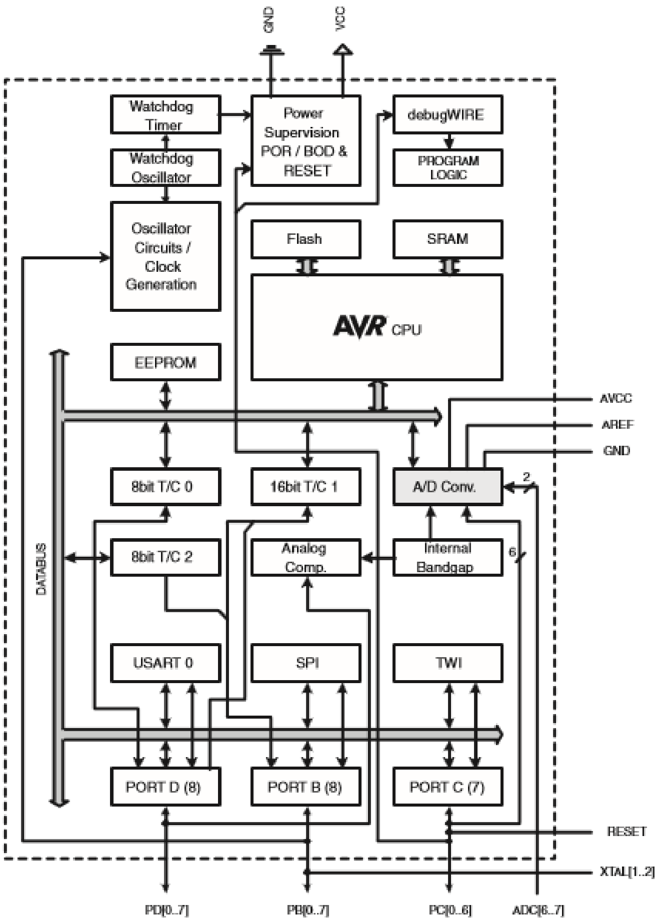
In the TQFP and QFN/MLF package, ADC7:6 serve as analog inputs to the A/D converter. These pins are powered from the analog supply and serve as 10-bit ADC channels.

2. Overview

The ATmega48PA/88PA/168PA/328P is a low-power CMOS 8-bit microcontroller based on the AVR enhanced RISC architecture. By executing powerful instructions in a single clock cycle, the ATmega48PA/88PA/168PA/328P achieves throughputs approaching 1 MIPS per MHz allowing the system designer to optimize power consumption versus processing speed.

2.1 Block Diagram

Figure 2-1. Block Diagram



The AVR core combines a rich instruction set with 32 general purpose working registers. All the 32 registers are directly connected to the Arithmetic Logic Unit (ALU), allowing two independent registers to be accessed in one single instruction executed in one clock cycle. The resulting architecture is more code efficient while achieving throughputs up to ten times faster than conventional CISC microcontrollers.

The ATmega48PA/88PA/168PA/328P provides the following features: 4K/8K bytes of In-System Programmable Flash with Read-While-Write capabilities, 256/512/512/1K bytes EEPROM, 512/1K/1K/2K bytes SRAM, 23 general purpose I/O lines, 32 general purpose working registers, three flexible Timer/Counters with compare modes, internal and external interrupts, a serial programmable USART, a byte-oriented 2-wire Serial Interface, an SPI serial port, a 6-channel 10-bit ADC (8 channels in TQFP and QFN/MLF packages), a programmable Watchdog Timer with internal Oscillator, and five software selectable power saving modes. The Idle mode stops the CPU while allowing the SRAM, Timer/Counters, USART, 2-wire Serial Interface, SPI port, and interrupt system to continue functioning. The Power-down mode saves the register contents but freezes the Oscillator, disabling all other chip functions until the next interrupt or hardware reset. In Power-save mode, the asynchronous timer continues to run, allowing the user to maintain a timer base while the rest of the device is sleeping. The ADC Noise Reduction mode stops the CPU and all I/O modules except asynchronous timer and ADC, to minimize switching noise during ADC conversions. In Standby mode, the crystal/resonator Oscillator is running while the rest of the device is sleeping. This allows very fast start-up combined with low power consumption.

The device is manufactured using Atmel's high density non-volatile memory technology. The On-chip ISP Flash allows the program memory to be reprogrammed In-System through an SPI serial interface, by a conventional non-volatile memory programmer, or by an On-chip Boot program running on the AVR core. The Boot program can use any interface to download the application program in the Application Flash memory. Software in the Boot Flash section will continue to run while the Application Flash section is updated, providing true Read-While-Write operation. By combining an 8-bit RISC CPU with In-System Self-Programmable Flash on a monolithic chip, the Atmel ATmega48PA/88PA/168PA/328P is a powerful microcontroller that provides a highly flexible and cost effective solution to many embedded control applications.

The ATmega48PA/88PA/168PA/328P AVR is supported with a full suite of program and system development tools including: C Compilers, Macro Assemblers, Program Debugger/Simulators, In-Circuit Emulators, and Evaluation kits.