HAIDIL HAZHA BIN HUSAINI BACHELOR OF ELCTRICAL ENGINEERING (ELECTRONICS) 2007-11-27 UMP

DEVELOPMENT OF AXIS MOVEMENT CONTROLLER FOR 2-AXIS CONFIGURATION MACHINE CONTROL

HAIDIL HAZHA BIN HUSAINI

UNIVERSITI MALAYSIA PAHANG

UNIVERSITI MALAYSIA PAHANG

BORANG PENGESAHAN STATUS TESIS ♦

JUDUL: **DESIGN X-AXIS ALGORITHM ON PRINTER MACHINE**

SESI PENGAJIAN: 2007/2008

Saya **HAIDIL HAZHA BIN HUSAINI (850818-14-6067)**
(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:

G-07-349 TAMAN SEGAR
CHERAS
56100 KUALA LUMPUR

ADDIE IRAWAN BIN HASHIM
(Nama Penyelia)

Tarikh: **29 NOVEMBER 2007**

Tarikh: **29 NOVEMBER 2007**

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).

"I hereby acknowledge that the scope and quality of this thesis is qualified for the award of the Bachelor`s Degree of Electrical Engineering (Electronics)"

Signature : ____________________________

Name : ADDIE IRAWAN BIN HASHIM

Date : 29 NOVEMBER 2007

DEVELOPMENT OF AXIS MOVEMENT CONTROLLER FOR 2-AXIS CONFIGURATION MACHINE CONTROL

HAIDIL HAZHA BIN HUSAINI

A thesis is submit as partial fulfillment of the requirements for the award of the Bachelor Degree of Electrical Engineering (Electronics)

Faculty of Electrical & Electronics Engineering
Universiti Malaysia Pahang

NOVEMBER 2007

“All the trademark and copyright use herein are property of their respective owner. References of information from other sources are quoted accordingly; otherwise the information presented in this report is solely work of the author”

Signature : ____________________

Author : HAIDIL HAZHA BIN HUSAINI

Date : 29 NOVEMBER 2007

‘To my beloved father, mother, brother and sisters.’

ACKNOWLEDGMENTS

Thank you Allah,

I wish to extend my sincere appreciation to Mr Addie Irawan Bin Hashim, my supervisor, for his continuous supervision, guidance, advice and constructive ideas throughout the course of this study and making this project successful. Without his support, this would have been a very difficult task.

Special thanks go to lecturers, students, and colleagues who also helped in various ways with my bachelor program, it would be impossible to list them all. Lastly, thank you to my parents and family for your encouragement and your non-technical assistance with my bachelor degree.

ABSTRACT

This project is done to design an axis movement controller for 2-axis configuration machine control. The focus is to control the stepper motor (unipolar) inside the printer machine. The controller is design by driving the stepper motor to control the movement of conveyor on printer system. The aim of the project is to control the movement of conveyor for every 1cm per step, forward and reverse. The axis control algorithm is program in the PIC16F877 microcontroller using software PIC proBasic. This project also will present the expected performance of the 2-axis motor, construction of hardware and software development to gather the result. Finally, the result will be evaluated by comparing the system analysis with theory calculation. From the performance result, shows that stepper motor will move around 144 degree for 1 cm. In conclusion a stepper motor inside the printer system has some error in running around 0.02cm after each expected 1cm per step.

Abstrak

Projek ini dibuat untuk mereka pengawalan gerakan paksi untuk 2-paksi konfigurasi kawalan motor. Fokus projek ini adalah mengawal stepper motor(unipolar) dalam mesin printer. Projek ini menyediakan pemacu stepper motor untuk menggerakkan conveyor kecil pada sistem printer. Projek ini menjurus kepada pergerakan conveyor 1 cm setiap detik, ke hadapan dan ke belakang. Algoritma paksi kawalan diprogramkan di dalam microcontroller PIC16F877 menggunakan perisian PIC proBasic. Projek ini juga akan menerangkan keupayaan jangkaan kawalan stepper motor, pembinaan perkakasan dan pembangunan perisian untuk mendapatkan data keupayaan system tersebut. Akhir sekali, keupayaan sistem ini akan diuji berdasarkan perbandingan diantara pengiraan teori dan data keupayaan sistem yang telah dianalisis. Daripada data keupayaan analisis, ia menunjukkan bahawa stepper motor bergerak 144 darjah untuk pergerakkan 1 cm. Kesimpulannya, stepper motor dalam sistem printer ada sedikit masalah lebih kurang 0.02cm pada pergerakan 1cm setiap detik.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATION

UMP	- Universiti Malaysia Pahang
DC	- Direct Current
AC	- Alternating Current
V	- Voltage
R	- Resistor
L	- Inductor
C	- Capacitor
Vin	- Input Voltage
Vout	- Output Voltage

LIST OF APPENDICES

CHAPTER 1

INTRODUCTION

1.1 Overview

This chapter will discuss on axis movement controller in printer system and its implementation to industry. The problem statement which contributes to the creation and development of this project, overview, objective and scope of this project is also presented in this chapter.

This project is focusing on development of axis movement controller for 2-axis configuration machine control. The movement of the conveyor in the system design is driven by a (unipolar) stepper motor. The project has prepare a driver for stepper motor to control the movement of conveyor on printer system. This project expected to move a conveyor of printer system for 1 cm per step, forward and reverse. The stepper motor is control using PIC16F877.

The stepper motor used in this project has 48 pulse and each pulse movement has 7.5 degree. The driver of a stepper motor that use in this system is ULN2003. This driver consist of 16 pinned with high input voltage and current. The advantage of using this driver is because the driver consist seven open collector darlington pairs with common emitter.

1.2 Project Objective

The objective of this project is the development of axis movement controller for 2-axis configuration machine control. The project has prepared a driver for stepper motor to control the movement of conveyor on printer system. The motor will rotate one step per digital signal change.

1.3 Project Scope

The scope of the project are to design the stepper motor controller unit for printer system axis movement control. The type of stepper motor that used in the printer system is unipolar stepper motor. Besides that, the focus of the project wil be on drive the movement of conveyor for 1 cm per step. In addition, the movement of conveyor is forward and reverse with difference length in scale of 1cm.

1.4 Thesis Organization

Chapter 1 discussed briefly about the project in the overview topic. The objectve and scope of the project is also discussed in this chapetr .

Chapter 2 discussed on literature review that related to this project, stepper motor charateristic, microcontoller PIC16F877, stepper motor driver ULN2003, axis concept on industrial machine and printer machine.

Chapter 3 discussed about the methodologies of firmware design to drive stepper motor using PIC16F877 microcontroller. This chapter also explain about electronic hardware design and mechanical design for printer system.

Chapter 4 discussed about results obtained from the printer movement performance. The discussion is focused on actual performance of conveyor movement results compare to the expected setting in firmware codes..

Chapter 5 will conclude all the discussions on the previous chapter. The recommendation for the future progress also described in this chapter.

CHAPTER 2

LITERATURE REVIEW

2.1 Stepper Motor Control

Stepping motors come in two varieties, permanent magnet and variable reluctance (there are also hybrid motors, which are indistinguishable from permanent magnet motors. Permanent magnet motors tend to "cog" as you twist the rotor with your fingers, while variable reluctance motors almost spin freely. Variable reluctance motors usually have three (sometimes four) windings, with a common return, while permanent magnet motors usually have two independent windings, with or without center taps. Center-tapped windings are used in unipolar permanent magnet motors.[3]

Stepping motors come in a wide range of angular resolution. The coarsest motors typically turn 90 degrees per step, while high resolution permanent magnet motors are commonly able to handle 1.8 or even 0.72 degrees per step. With an appropriate controller, most permanent magnet and hybrid motors can be run in half-steps, and some controllers can handle smaller fractional steps or microsteps.[3]

For both permanent magnet and variable reluctance stepping motors, if just one winding of the motor is energised, the rotor (under no load) will snap to a fixed angle and then hold that angle until the torque exceeds the holding torque of the motor, at which point, the rotor will turn, trying to hold at each successive equilibrium point.[3]

2.1.1 Type of Stepper motor

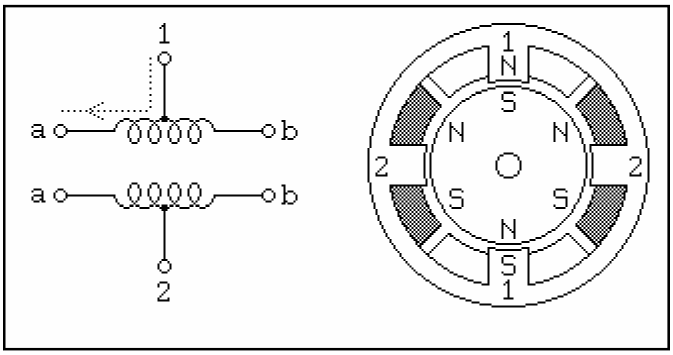


Figure 2.1: Unipolar Stepper Motor

Unipolar stepping motors, both permanent magnet and hybrid stepping motors with 5 or 6 wires are usually wired as shown in the schematic in Figure 2.1, with a center tap on each of two windings. In use, the center taps of the windings are typically wired to the positive supply, and the two ends of each winding are alternately grounded to reverse the direction of the field provided by that winding.[4]

The motor cross section shown in Figure 2.1 is of a 30 degree per step permanent magnet or hybrid motor. The difference between these two motor types is not relevant at this level of abstraction. Motor winding number 1 is distributed between the top and bottom stator pole, while motor winding number 2 is distributed between the left and right motor poles. The rotor is a permanent magnet with 6 poles, 3 south and 3 north, arranged around its circumference.[4]

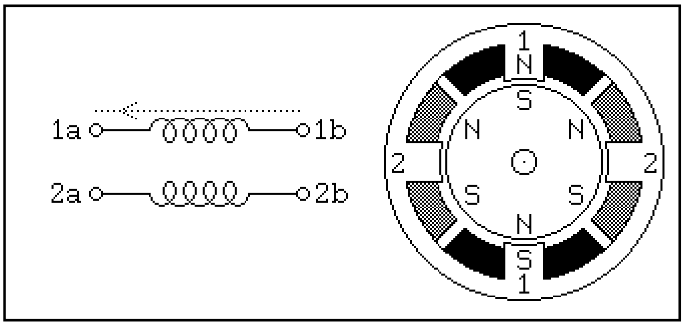


Figure 2.2: Bipolar stepper motor

Bipolar permanent magnet and hybrid motors are constructed with exactly the same mechanism as is used on unipolar motors, but the two windings are wired more simply, with no center taps. Thus, the motor itself is simpler but the drive circuit needed to reverse the polarity of each pair of motor poles is more complex. The schematic in Figure 2.2 shows how such a motor is wired. The motor cross section shown here is exactly the same as the cross section shown in Figure 2.1.[4]

To distinguish a bipolar permanent magnet motor from other 4 wire motors, measure the resistances between the different terminals. It is worth noting that some permanent magnet stepping motors have 4 independent windings, organized as two sets of two. Within each set, if the two windings are wired in series, the result can be used as a high voltage bipolar motor. If they are wired in parallel, the result can be used as a low voltage bipolar motor. If they are wired in series with a center tap, the result can be used as a low voltage unipolar motor.[4]

The ULN2003 are high voltage and high current darlington array each containing seven open collector darlington pairs with common emitter. Each channel rated at 500mA and can withstand peak current of 600mA. Suppression diodes are included for inductive load driving and the inputs are pinned opposite the output to simplify board layout.

These versatile devices are useful for drivinga wide range of loads including selonoids relay DC motors, LED displays filament lamps, thermal printheads and high power buffers. The ULN2003 are supplied in 16 pin plastic DIP packages with a cooper lead frame to reduce thermal resistance.

2.1.2 Stepper Motor Control Concept

The ULN2003 are high voltage and high current darlington array each containing seven open collector darlington pairs with common emitter. Each channel rated at 500mA and can withstand peak current of 600mA. Suppression diodes are included for inductive load driving and the inputs are pinned opposite the output to simplify board layout.

These versatile devices are useful for drivinga wide range of loads including selonoids relay DC motors, LED displays filament lamps, thermal printheads and high power buffers. The ULN2003 are supplied in 16 pin plastic DIP packages with a cooper lead frame to reduce thermal resistance.

The scope of the project is on controlling unipolar stepper motor. Therefore, focusing about this topic on how to control this motor and how to make it move step by step is apply.

Things are more complicated for unipolar permanent magnet stepping motors because these have center taps on their windings. Therefore, to reverse the direction of the field produced by a motor winding, we need to reverse the current through the winding.

A stepper motor operates as a result of electromagnetism. Magnets are assembled in a stepper motor in pairs that will create two poles of magnetic attraction. A stepper motor is a digitally controlled motor that has non-accumulative rotational error. This means that the number of rotations that the motor turns can be accurately controlled and measured. The motor will rotate one step per digital signal change.

Current will be supplied from a 5V signal that is generated at one pole of the magnet. For example in Figure 2.3 (Step 1), a 5V signal is being applied to both A1 and B1. In this situation, the opposite pole will be connected to ground and the current will flow through the wire and create the opposite magnetic poles. The magnet that is represented in the middle will be the magnet that is connected to the shaft of the motor. If 5V is next applied to A2 and B1, the magnet connected to the shaft will rotate 90° (Step 2). This sequence can be repeated for the next step, the next one after that, and so on. In this example there are four poles and the motor will rotate 25 percent, or 90 degrees, for every step. The sequence for switching the poles is displayed in Table 1 below.

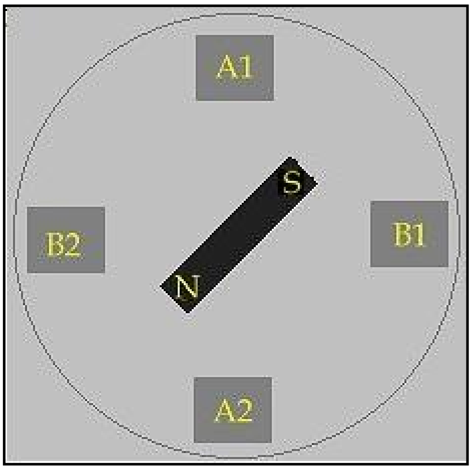


Figure 2.3(a): motor 0 degree

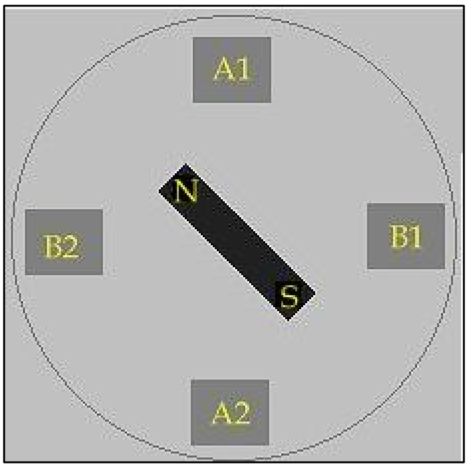


Figure 2.3(b): motor 90 degree

Table 2.1: sequence for motor rotate 90 degree

A1	A2	B1	B2
1	0	1	0
0	1	1	0
0	1	0	1
1	0	0	1

This is the illustration of a motor operating in a full step mode. Greater resolution can be gained if the motor is rotated half a step at a time. If this mode is selected less torque is generated by the motor. There is less torque because for every half step only one magnet pair is attracting the magnet connected to the shaft. Half steps are created by turning off one pair of poles in between transitions from one step to another. The sequence for switching the poles in a half step mode is displayed in Table 2. Figure 2.5 is an illustration of a stepper motor operating in half step mode.

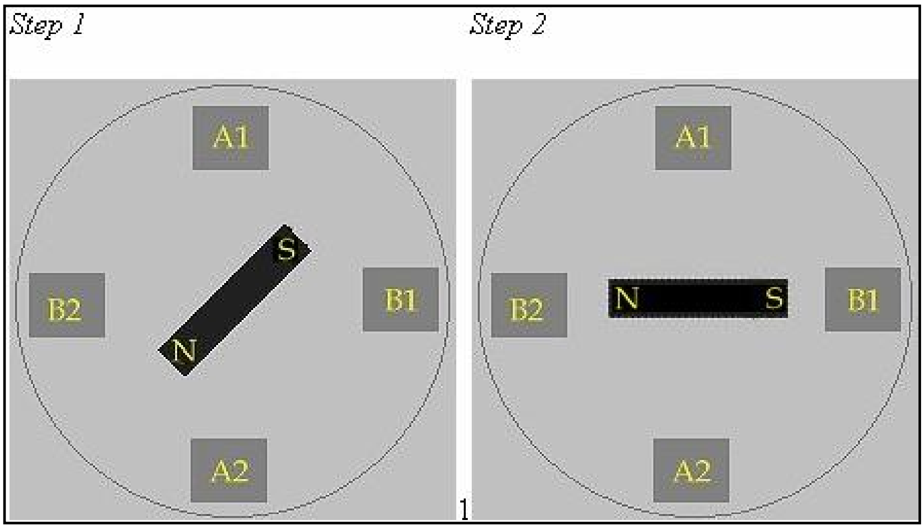


Figure 2.4: motor with half stepping 45 degree

Table 2.2: sequence for motor rotate 45 degree

A1	A2	B1	B2
1	0	1	0
0	0	1	0
0	1	1	0
0	1	0	0
0	1	0	1
0	0	0	1
1	0	0	1
1	0	0	0

2.2 Axis Concept On Printer Conveyor

There are several axis that can be implement to the machine that is x-axis, y-axis and z-axis. The x-axis means the axis in a plane Cartesian coorsinate system to which abscissas are measured. The x-axis eventually has only one direct movement and the its coordinate has only one value. Figure 3.3 shows the plane and x-axis movement.[7]

Figure 3.4 shows the plane and xy-axis movement. The x-axis and y-axis also known axis in two-dimensional coordinate system, the horizontal axis and vertical axis in a two-dimensional coordinate system..

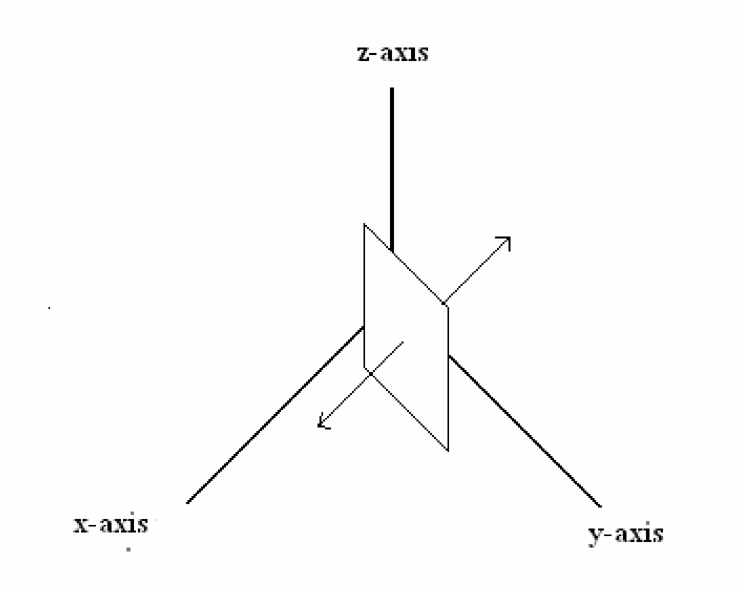


Figure 2.5: X-axis Movement

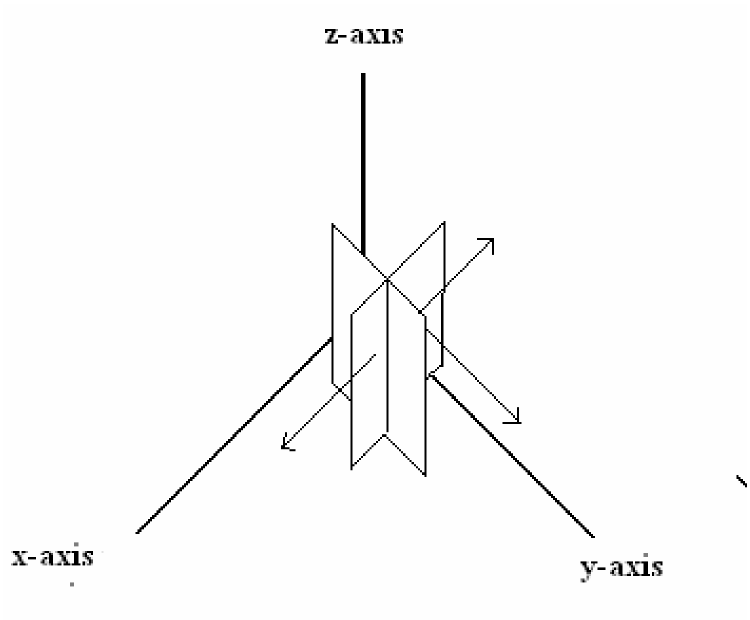


Figure 2.6: X-axis and Y-axis movement

The machine that can move within three combination axis (x-axis, y-axis and z-axis) will provide better product or output because it is not only move horizontal and vertical, but also can change its height to different high. It also needs bigger workspace to operate. Combining the three axis with a suitable tools also can make the machine to do 360degree workspace. For example, the construction of ball using machine that have all three axis. With precise calculation and determination, the spherical ball can be created.

2.3 Printer Machine Configuration

Printer machine is a device that prints text or illustrations on paper. There are many different types of printers that ussually used nowadays. In terms of the technology utilized, printers fall into the following categories:

(i) **Daisy-wheel** - Similar to a ball-head typewriter, this type of printer has a plastic or metal wheel on which the shape of each character stands out in relief. A hammer presses the wheel against a ribbon, which in turn makes an ink stain in the shape of the character on the paper.Daisy-wheel printers produce letter-quality print but cannot print graphics.

(ii) **Dot-matrix** - Creates characters by striking pins against an ink ribbon. Each pin makes a dot, and combinations of dots form characters and illustrations.

(iii) **Ink-jet** - Sprays ink at a sheet of paper. Ink-jet printers produce high-quality text and graphics.

(iv) **Laser** - Uses the same technology as copy machines. Laser printers produce very high quality text and graphics.

(v) **LCD & LED** - Similar to a laser printer, but uses liquid crystals or light-emitting diodes rather than a laser to produce an image on the drum.

(vi) **Line printer** - Contains a chain of characters or pins that print an entire line at one time. Line printers are very fast, but produce low-quality print.

(vii) **Thermal printer** - An inexpensive printer that works by pushing heated pins against heat-sensitive paper. Thermal printers are widely used in calculators and fax machines.

2.3.1 Classification of Printer Machine

Printers are also classified by the following characteristics according to manufacturing product:

(i) **Quality of type** - The output produced by printers is said to be either letter quality (as good as a typewriter), near letter quality, or draft quality. Only daisy-wheel, ink-jet, and laser printers produce letter-quality type. Some dot-matrix printers claim letter-quality print, but if you look closely, you can see the difference.

(ii) **Speed** - Measured in characters per second (cps) or pages per minute (ppm), the speed of printers varies widely. Daisy-wheel printers tend to be the slowest, printing about 30 cps. Line printers are fastest (up to 3,000 lines per minute). Dot-matrix printers can print up to 500 cps, and laser printers range from about 4 to 20 text pages per minute.

(iii) **Impact or non-impact** - Impact printers include all printers that work by striking an ink ribbon. Daisy-wheel, dot-matrix, and line printers are impact printers. Non-impact printers include laser printers and ink-jet printers. The important difference between impact and non-impact printers is that impact printers are much noisier.

(iv) **Graphics** - Some printers (daisy-wheel and line printers) can print only text. Other printers can print both text and graphics.

(v) **Fonts** - Some printers, notably dot-matrix printers, are limited to one or a few fonts. In contrast, laser and ink-jet printers are capable of printing an almost unlimited variety of fonts. Daisy-wheel printers can also print different fonts, but you need to change the daisy wheel, making it difficult to mix fonts in the same document

2.4 PIC16F877 microcontroller

PICs are easily programmable cheap microcontrollers. PIC is the name for the Microchip microcontroller (MCU) family, consisting of a microprocessor, I/O ports, timer(s) and other internal, integrated hardware. The main advantages are low external part count, a wide range of chip sizes (from 8-pin up), great availability of compilers and source code and easy programming. Flash-type devices are reprogrammable in-circuit, while OTP versions are very cheap to use at the final stage. A wide range of simple programmer hardware and software is downloadable from the net. PIC16F84 seems to be the "standard" for small gadgets.

The PIC16F877 Microcontroller includes 8kb of internal flash Program Memory, together with a large RAM area and an internal EEPROM. An 8-channel 10-bit A/D converter is also included within the microcontroller, making it ideal for real-time systems and monitoring applications. All port connectors are brought out to standard headers for easy connect and disconnect. In-Circuit program download is also provided, enabling the board to be easily updated with new code and modified as required, without the need to remove the microcontroller.

The new PIC16F877 Controller is the ideal solution for use as a standard controller in many applications. The small compact size combined with easy program updates and modifications make it ideal for use in machinery and control systems, such as alarms, card readers, real-time monitoring applications and much more. This board is ideal as the brains of your robot or at the center of your home-monitoring system. Save time and money, by simply building your ancillary boards and monitoring circuits around this inexpensive and easy to use controller.

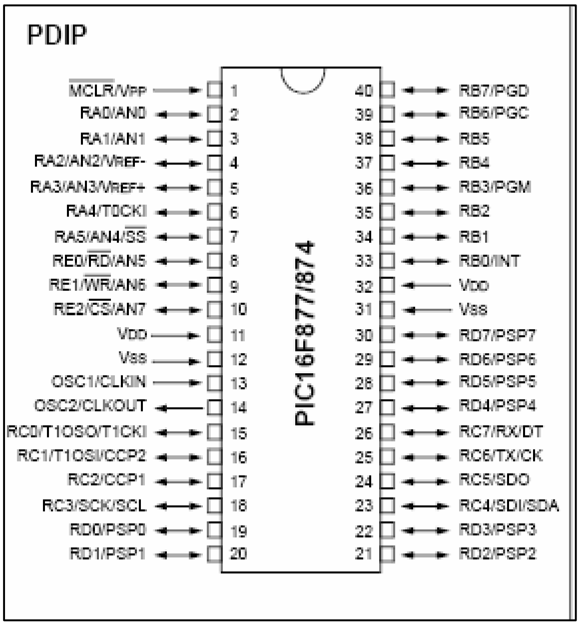


Figure 2.7: PIC16F877 Microcontroller pin configuration

The features that have in PIC16F877 microcontroller is show as below:

(i) Includes Powerful Microchip PIC16F877 Microcontroller with 8kb Internal Flash Program Memory
(ii) Operating Speed at 10MHz
(iii) Direct In-Circuit Programming for Easy Program Updates
(iv) Up to 28 I/O points with easy to connect standard headers
(v) RS232 Connection with MAX232
(vi) Internal EEPROM
(vii) 8 Channel 10-bit A/D Converter
(viii) One 16-bit Timer with Two 8-bit Timers
(ix) Power and Programming LED
(x) Reset Button - Ideal as an Interchangeable Controller for Real-Time Systems

CHAPTER 3

HARDWARE DESIGN

3.1 Electronic Hardware Design

In the hardware part,the circuit is design to run the stepper motor for about 1cm per step. The circuit includes PIC16F877 microcontroller, ULN2003 stepper motor driver and other basic components. The crystal oscillator that used as clock for microcontroller is valued 8MHz. The microcontroller is program first before run and connect it into the circuit. This is because the circuit will be operated in offline condition(open loop system) in order to drive the motor. The complete circuit for the system is shown in APPENDIX A.

The microcontroller that used to control the stepper motor is PIC16F877 and it has 40 pins in it. The input voltage, VDD for the microcontroller is on pin 11 and pin 32 and its value after passing the regulator is 5 Volt. The output voltage is on RB0, RB1, RB2 and RB3. The crystal oscillator is connected to pin 13 and pin 14 of the microcontroller. The function of the regulator of the system design is to regulate the volage from supply 9 Volt to 5 Volt using IC LM7805C.

The driver of the stepper motor that is in system design is ULN2003 and is shown in figure 3.3.3. The ULN2003 are high voltage, high current darlington arrays each containing seven open collector darlington pairs with common emitters. Each channel rated at 500mA and can withstand peak currents of 600mA. Suppression diodes are included for inductive loaded driving and the inputs are pinned opposite the outputs to simplify the board layout.

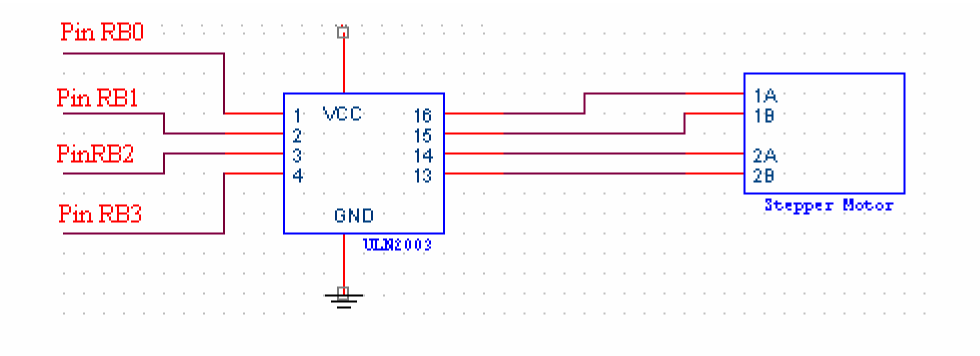


Figure 3.1: Stepper Motor Driver Part

3.2 Firmware Design

The software that used to program the microcontroller is PIC PROBasic. The software is created in order to interface with the microcontroller and driver motor. All coding is depending on output and configuration needed to achieve the project objective. The first command that used to program the PIC is shown as below. The command show that the movement of the motor will move forward and reverse continuously.

Forward running code set

Step 1 = %00001010

Step 2 = %00000110

Step 3 = %00000101

Step 4 = %00001001

Reverse running code set

Step 1 = %00001001

Step 2 = %00000110

Step 3 = %00000101

Step 4 = %00000110

In the design, the stepper motor will running starting from 0cm, 14cm forward, 10cm reverse, 6cm forward, 10 cm reverse and then the stepper motor will running continuously. This design can change by changing the software for forward and reverse according to their on choice but every of the changing will move in 1cm perstep. Below show the flow chart of the software design to help understand how the stepper motor is programmed.

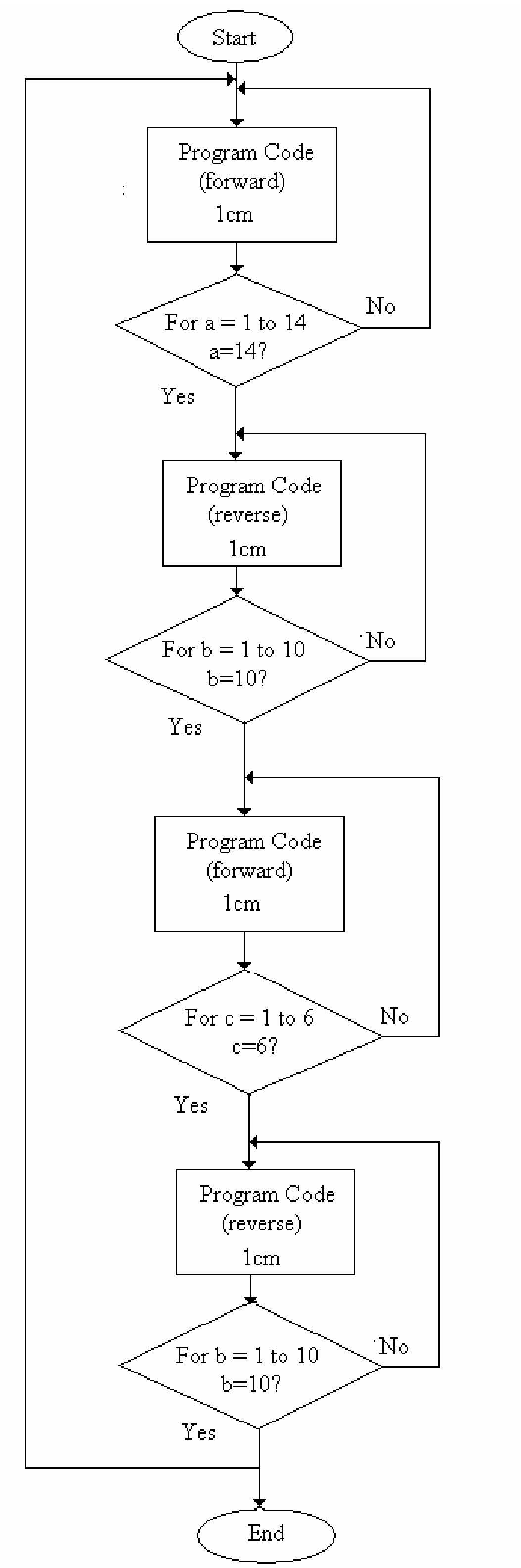


Figure 3.2: Flow Chart of firmware design

In making the firmware program, we must understand the pulse of the stepper motor. In this chapter will discuss about how to program the PIC microcontroller. Firstly, the driver of the stepper motor is connected to port B of microcontroller. The configuration that connected from microcontroller that use to run the stepper motor is RB0, RB1, RB2 and RB3. so the command to set to the stepper motor is TRISB = %11110000

There are several stepping modes that you can use to drive the stepper motor. The mode that can use to drive stepper motor is single stepping, high torque stepping and half stepping mode.

Single Stepping is the simplest mode turns one coil ON at a time. 48 pulses are needed to complete one revolution. Each pulse moves rotor by 7.5 degrees. The following sequence has to be repeated 12 times for motor to complete one revolution.

Table 3.1: Single Stepping Mode

Pulse	Coil a1	Coil b1	Coil a2	Coil b2
1	ON			
2		ON		
3			ON	
4				ON

High Torque Stepping is high power / precision mode turns ON two coils on at a time. 48 pulses are needed to complete one revolution. Each pulse moves rotor by 7.5 degrees. The following sequence has to be repeated 12 times for motor to complete one revolution.

Table 3.2: High Torque Stepping Mode

Pulse	Coil a1	Coil b1	Coil a2	Coil b2
1	ON	ON		
2		ON	ON	
3			ON	ON
4	ON			ON

Half Stepping is doubled and motor needs 96 pulses to complete one revolution. Each pulse moves rotor by approximately 3.75 degrees. Notice the mix of single stepping mode (lighter green) and high torque mode (darker green).

Table 3.3: Half Stepping Mode

Pulse	Coil a1	Coil b1	Coil a2	Coil b2
1	ON			
2	ON	ON		
3		ON		
4		ON	ON	
5			ON	
6			ON	ON
7				ON
8	ON			ON

3.3 Mechanical Design

In the mechanical design, the unusable printer machine is redesigned to make the movement of x-axis control algorithm. The unsuable component from the printer machine is removed. The unipolar stepper motor is inserted into the printer and the electronic hardware design is put behind the printer. There is a few number shown in front of the printer machine to determine and identified the value of stepper motor movement. The number is shown in centrimetres from 0cm to 14cm.

The hard part in this mechanical design is to determine weither the stepper motor is fit in inserted into the printer. The motor then combine with a tiny rope and work as an indicator along the rope. The indicator and the number in front of the printer machine is set so that the value of the indicator is shown 0cm (starting point).

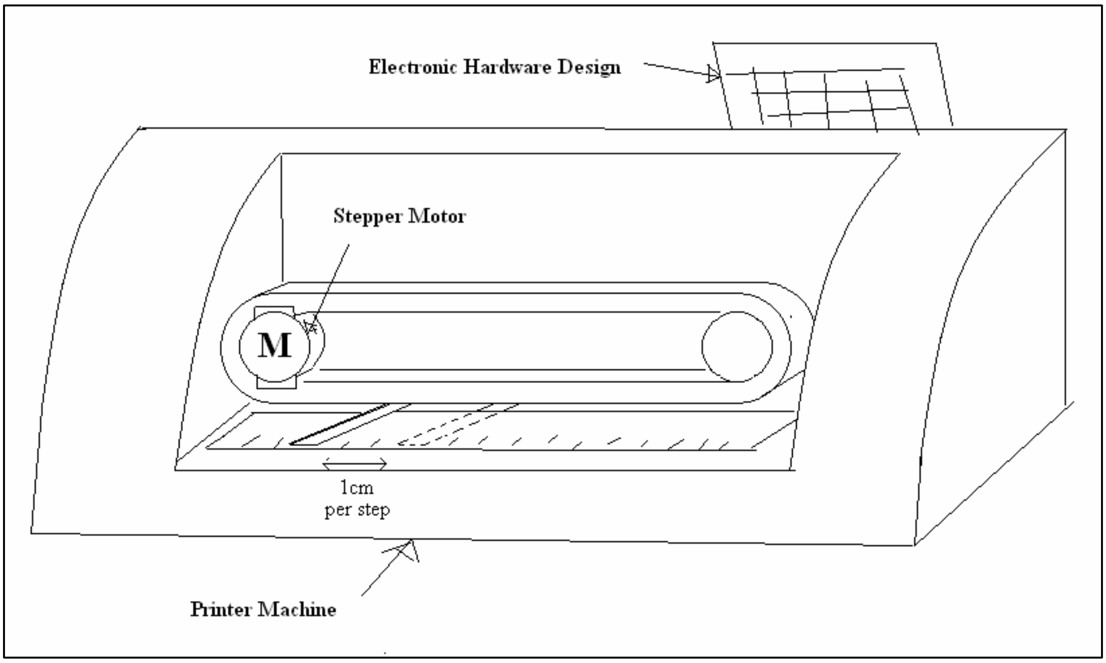


Figure 3.3: Printer Machine Diagram

3.3.1 Type in the printer machine

Stepper motors have numerous wire that is 4,5,6, or 8 wire. When you turn the shaft you will usually feel a "notched" movement. Motors with 4 wires require a Bipolar controller, such as the Dual H-Bridge. Others may be used with either a Unipolar or Bipolar controller. You can use an Ohm-meter to find the center tap--the resistance between the center and a leg is 1/2 that from leg to leg. Measuring from one coil to the other will show an open circuit.

The type of stepper motor that use in printer machine is unipolar stepper motor 5 pin. The 4 wire in the stepper is controlled from the microcontroller and the next wire is from supply 12.5Vol.

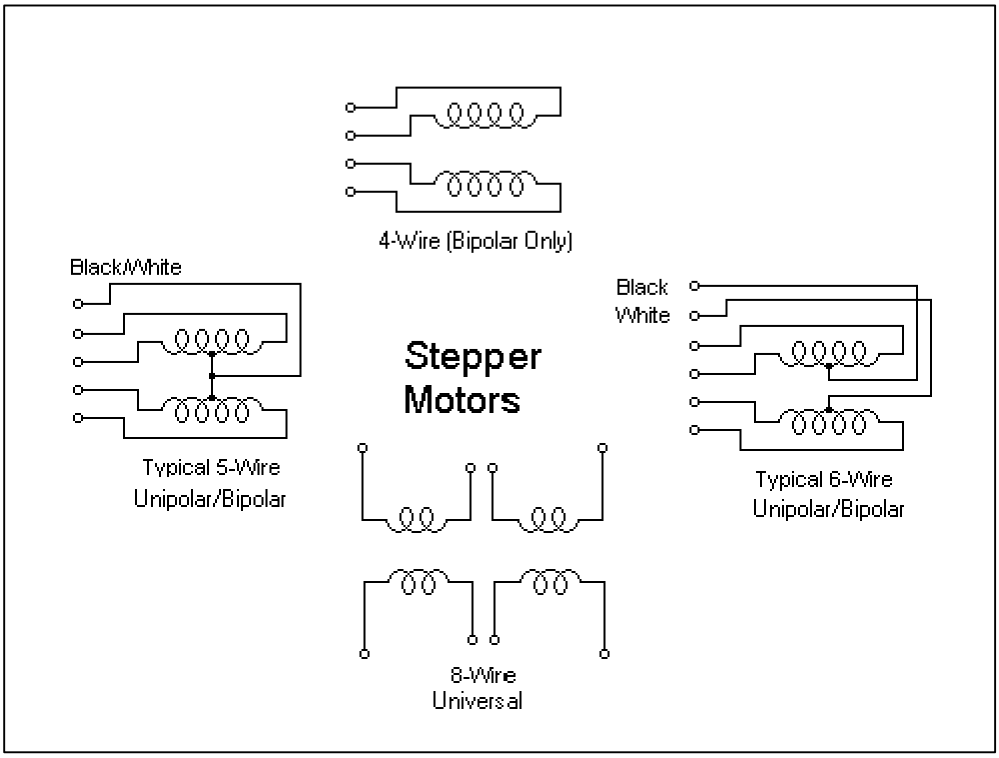


Figure 3.4: Identifying Stepper Motor

One of the benefits of the Bipolar drive circuit is that it allows us to use both "Bipolar" and "Unipolar" motors. Bipolar motors are simple Unipolar without the center taps, which simplifies construction of the motor. So we can drive 4,6, and 8 wire steppers. Furthermore, Unipolar motors may be used in 2 different configurations. One, simply ignore the center taps. Two, use half the coil only by using the center tap and one of the terminals. This will produce less hold torque but allows higher top speeds because of the lower inductance.

3.3.2 Heat Considerations

Over-heating can be an early indicator of a problem or need for additional heat sinking. This is true of both the controller and motors. Components can be warm to the touch, but not so hot that you can't leave your finger on them for a few seconds.

Motors are designed to be mounted in such a way that heat is drawn away from the motors. This is usually accomplished with a metal mounting bracket. Motors that are not yet mounted may require some type of temporary heat sinking. Motors heat more running at the LOW speeds or in Hold Mode.

If a component or motor is running too hot, try using the Wave Drive stepping mode only, if it still runs too hot, try heat sinking or a fan. When this things apply to the circuit, it will help to reduce the heat happen at the stepper motor. If it still runs too hot, something is wrong.

In summation, make calculations carefully, and always apply caution when making any kind of modification to a circuit. Check the circuit often for over-heating before funning the stepper motor. The better way to reduce this problem is always take precaution and troubleshoot the circuit after wraping the wire into the design.

3.3.3 Selecting a Current Limiting Resistor

It is important that neither the motor nor controller exceed their rated currents. The value of a current-limiting resistor in series with the motor can be determined from the following equation.(1)

$$Vsupply = Vdrop + (I \times Rmotor) + (I \times Rlimit) \quad \text{..........................equ(1)}$$

Vdrop = voltage drop of Transistors used (2 volts for Unipolar, 4 volts for Bipolar Drivers)

I = current in circuit

Rmotor = resistance of single motor coil

Rlimit = current limiting power resistor

The objective is to find Rlimit that satisfies the desired current rating I[10]. For example, Using a power supply of 12 volts, what current limiting resistor should be chosen to deliver 1 amp of current to a motor with a coil resistance of 5 ohms. Assume 2 volt drop due to the transistor:.

$$12v = 2v + (1\ amp \times 5\ ohm) + (1\ amp \times Rlimit)$$

$$12v = 2v + (5v) + (1\ amp \times Rlimit)$$

$$5v = 1\ amp \times Rlimit$$

$$Rlimit = 5v / 1\ amp = 5\ ohm$$

This calculation is for determining the current in a single coil. Most motors and stepper circuits are rated in terms of this coil rating - amps per phase. If the hi-torque, or half-step sequence is to be used, then more than 1 coil will be activated, thus more current will be required of the power supply. Ideally, two matched current limiting resistors should be used, one in series with each of the 2 motor coils. (The Unipolar driver can get away with sharing a single power resistor, with the center taps tied together.) The figure below show the configuration for the Bipolar Controller.

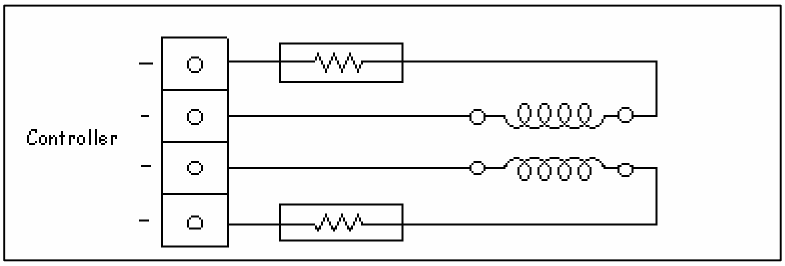


Figure 3.5: Current Limiting with Bipolar Controller

3.4 Stepper Motor Calculation

To make the motor run 1cm per step, we must know the pulse of the motor and its degre per pulse. Calculation below shows how to make the motor run about 1cm per step.

First we must know the sequence of the stepper motor in binary 8 bit using basic stepping motor command.

The command to run the stepper motor forwards
Step 1 = %00001010
Step 2 = %00000110
Step 3 = %00000101
Step 4 = %00001001

The command to run the stepper motor reverses
Step 1 = %00001010
Step 2 = %00001001
Step 3 = %00000101
Step 4 = %00000110

Next we will get that the pulse of the stepper motor is 48 pulse per rotation. Then, using the half stepping motor mode, we will get the pulse twice the single stepping, that is 96 pulse per rotation.

Degree for 1 pulse half stepping motor from equation(2)

$$\text{number of degree per rotate} = \frac{360 \text{ degree}}{\text{Pulse per rotate}} \quad \text{.............equ(2)}$$

$$= \frac{360 \text{ degree}}{96 \text{ pulse per rotate}}$$

$$= 3.75 \text{ degree per rotate}$$

Diameter of the gear of stepper motor d = 0.8 cm

Circle = 2 x pi x (d / 2)

= 2 x pi x (0.8 / 2)

= 2.5132cm

≈ 2.5cm

For 1 cm per step:

$$\frac{1 \text{ cm} \times 360 \text{ degree}}{2.5 \text{ cm}} = 144 \text{ degree}$$

To make the stepper motor move about 144 degree:

$$\text{Steps} = \frac{144 \text{ degree}}{3.75 \text{ degree}}$$

$$= 38.4 \text{ steps}$$

$$\approx 38 \text{ steps}$$

The calculation show that to move the stepper motor 1cm using have stepping mode, we will get that the steps is 38 pulse per cm.

In the system design, the motor will run about 1cm per step by 40 pulse. the stepper will run various of length by using 1cm per step. The minimum length that the motor can run is 1cm. In the design, the stepper motor will run 14cm forward, 10cm reverse, 6cm forward , 10cm reverse and repeat continuously.

The rotation for 14cm forward:

$$\text{Rotation} = \frac{144 \text{ degree x } 14 \text{ step}}{360 \text{ degree}}$$

$$= 5.6 \text{ rotation}$$

The rotation for 10cm reverse:

$$\text{Rotation} = \frac{144 \text{ degree x } 10 \text{ step}}{360 \text{ degree}}$$

$$= 4 \text{ rotation}$$

The rotation for 6cm forward:

$$\text{Rotation} = \frac{144 \text{ degree x } 6 \text{ step}}{360 \text{ degree}}$$

$$= 2.4 \text{ rotation}$$

The rotation for 10cm reverse:

$$\text{Rotation} = \frac{144 \text{ degree x } 10 \text{ step}}{360 \text{ degree}}$$

$$= 4 \text{ rotation}$$

To calculate the stepper motor speed in rotate per minute(RPM), the time delay for each step is determined. Below show the calculation to get the rpm of the stepper motor.

144 degree = 1.4 second times

= 0.023 minutes

$$1 \text{ degree} = \frac{1.4}{144}$$

$= 9.722 \times 10^{-3}$

360 degree $= 9.722 \times 10^{-3} \times 360$

= 3.5 second times

= 0.058 minutes

So, the rpm of the stepper motor is 0.058rpm.

CHAPTER 4

SYSTEM PERFORMANCE

4.1 Movement Analysis Procedure

The aim ofthis project is to control the axis movement controller for printer conveyor 1 cm per steps, forward and reverse. However, the result has some errors on certain steps in forward and reverse mode. The result is not 100% same as expected result and fulfill the objective. The stepper motor run forward and reverse but not approximately 1cm and the error of it is about 1cm ± 0.02cm.

The data that needed to be collected and measured is the output voltage from PIC in Port B (RB0, RB1, RB2 and RB3) to stepper motor driver. The multimeter is used as measurement devices. The input voltage to the regulator is about 9 volt and the output of the regulator will produce 5 volt. The regulator is used to regulate the voltage from 9 volt to 5 volt. This design is to make the microcontroller running because the maximum value microcontroller can withstand according to the datasheet is about 5 volt.

From the regulator the voltage will go into pin 1(master clear), pin 11(VDD) and pin 32(VDD). The output voltage from microcontroller to stepper motor driver is connectted in pin 33(RB0), pin 34(RB1), pin 35(RB2) and pin 36(RB3). The output from microcontroller depends on the programming to the microcontroller.

The input voltage is divide into two parts. The first part is to supply the microcontroller with the value 9 volts. The second input voltage is supply to the stepper motor in 12 volts. This difference of the input value of this parts happen because the microcontroller maximum input value can reach only 5 volt. After the voltage 9 volts go through regulator, it then is regulated to 5 volt and then going into microcontroller and the microcontroller is running completely. The stepper motor must have the input value 12 volts to make the stepper motor running continuosly without interference from the environment.

4.2 Conveyor Stepping Analysis

After the stepper motor is running, the lengthof each step is determined by using a ruler. The stepper motor value from the result is compare with the calculation and theory that discussed in the Chapter 3. The distance of each step is determined.

The movement of the stepper motor is continuous at constant speed. The speed of the motor is depends on the delay that has been programming in the microcontroller. The delay for every 1 cm stepper motor movement is 1.4 second. The 0.4 second delay is produced by using the value of hall stepping steppermotor, 40 steps times 10 milisecond. In programming, times to run the stepper motor 1cm is set so that each steps has delay 10 milisecond.

The table below is shown the output value of the stepper motor corresponding to the input value given from microcontroller, its rotational and error in each step.

Table 4.1: Steps and rotational of stepper motor

Steps	Actual Length (cm)	Length Error (cm)	Rotation of the stepper motor (per rotate)	Rotation of the stepper motor (degree)
0	0	0	0	0
1	1	1.02	0.4	144
2	2	2.04	0.8	288
3	3	3.06	1.2	432
4	4	4.08	1.6	576
5	5	5.1	2	720
6	6	6.12	2.4	864
7	7	7.14	2.8	1008
8	8	8.16	3.2	1152
9	9	9.18	3.6	1296
10	10	10.2	4	1440
11	11	11.22	4.4	1584
12	12	12.24	4.8	1728
13	13	13.26	5.2	1872
14	14	14.28	5.6	2016

4.3 Discussion

From the table steps and rotational, the stepper motor can be program into a difference configuration and degree. The stepper motor move 144 degree per step and it also move 0.2 rotation per step.

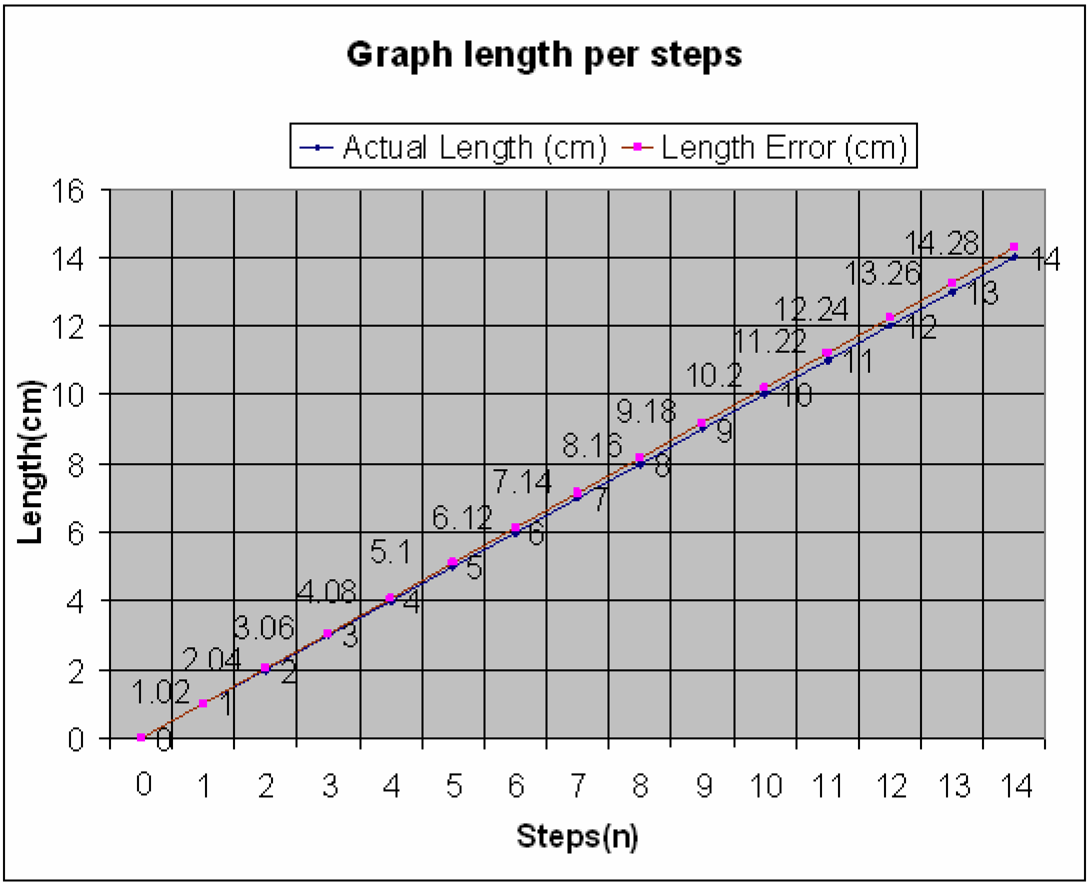


Figure 4.1: Graph length per Steps

The graph shows that the value of length error is increase every step around 0.02cm. This happen because the steps of the motor has big degree per step. The error value for 14cm movement is 0.28cm. the error value for 4cm is four times error per step. At steps 10, the value of error length is 10.2cm. There is no different value at steps 0. However, the error length at step 7 is half the error at 14cm length.

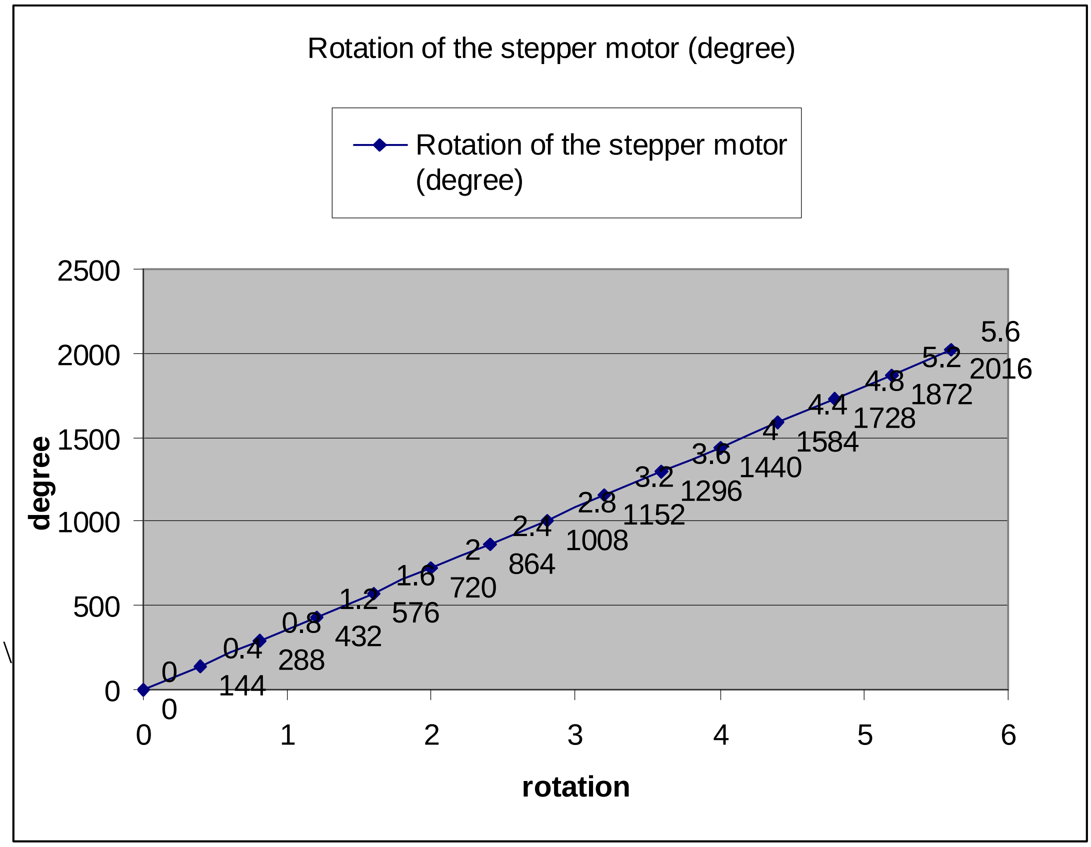


Figure 4.2: Graph degree per rotation

Figure 4.2 shows that the value of degree and rotation per step. the value of each rotation and degree is linear. There is no different in degree and rotationwhen the step is 0. when the step of stepper motor is 14, the value of its rotation is 5.6 and the degree is 2016°. When the stepper motor move 2 rotation, it actually move 864°. For every 1 step, the stepper motor will move 144 degree and 0.4 rotation.

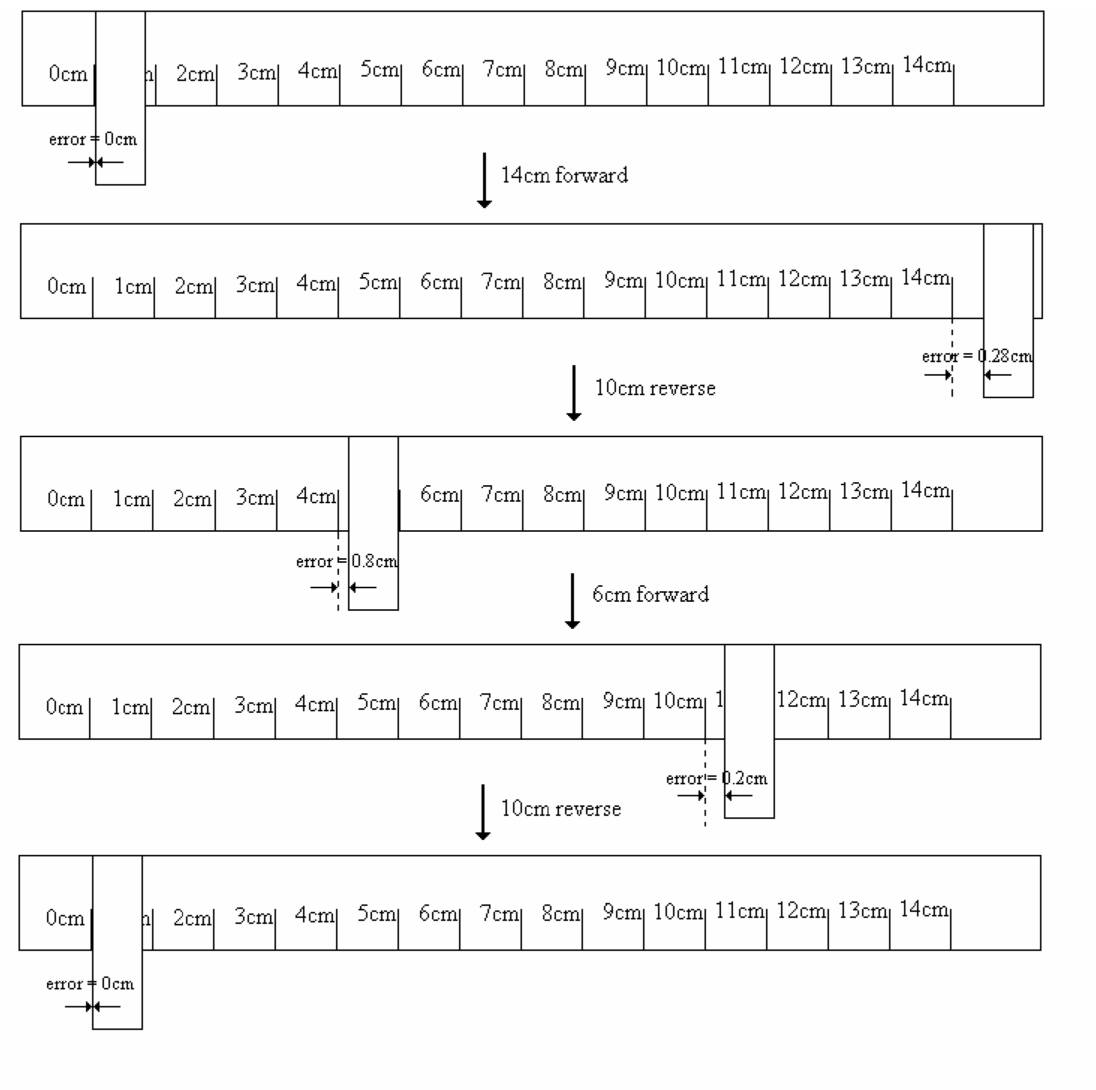


Figure 4.3: Printer conveyor movement actual results

The speed in rotate per minutes(RPM) for the stepper motor is define by calculating the time delay of the stepper motor for one rotation. The stepper motor is move at constant speed in forward and reverse direction for every step. The speed play a role part in the system besides program it to move 1cm. This is because, in industrial the productivity can be determined according to the work done by the stepper motor and moreover the time of the operation can be measured.[10]

4.4 Problem Encounter and Terms

There are several errors that encountered during the development of this project. The result of the stepper motor movement is the final result in this project. However the stepper motor in this system does not move approximately 1cm. The problem occurs when a time to determine the steps or pulse for 1cm. The stepper motor that used in this system has 7.5 degree per step and it has 48 steps per rotation. To get 1cm more accurately, the stepper motor must have more steps per rotation and the degree per step must much smaller.

Besides that, there has some connection less during the experiment especially in voltage input and grounding. This important of this connection because is to make sure that the circuit has a source to ON and OFF

Other than that, there is some mistake while define the value of the component such as resistor and capacitor. This is important to calculate the value of this component because the output of the inverter circuit is depending on this component. Finally, it cause by component which is already damage before doing the project.

CHAPTER 5

CONCLUSION AND RECOMMENDATION

5.1 Conclusion

The result shows the movement of conveyor is not approximately 1 cm per step because there is error around 0.02 cm for each step of the stepper motor movement. By using PIC 16F877 as a microcontroller the stepper motor driver can be program. The IC ULN2003 is used as the driver for unipolar stepper motor five pins. This system can be successful when the development of computer-based system is implementing and looks possible for future development. The operation of this project is moving forward and reverses by 1 cm per step in different length configuration. The movement of the conveyor is 14 cm forward, 10cm reverse, 6 cm forward, 10 cm reverse and its running continuously.

5.2 Recommendation

For the future development and enhancement, there are much more can do with this system like control the movement of the stepper motor by using pc based controller from computer. When input is given to the computer, the stepper motor will run according to the instruction given.

Besides that, the system also can be developing by implement the sensor to circuit in the system. The sensor will help to detect object and help the stepper motor respond to the sensor according to the situation. For example, a conveyor movement with a sensor. When sensor detect the object, the stepper motor will stop and let the robot along the line conveyor do the work before the conveyor move it again.

The speed of the motor can be senses without disturbing the system performances. It is because, the sensor circuit will not affect the Electromagnetic Interference (EMI) characteristics of the system and it also can detect over current faults that occur by short circuit. [11]

5.2.1 Costing and Commercialization

The total cost for this project is only RM 100.00. This project is designed to bevery affordable and still functions properly. This project has very high potential for various productions because conveyor is very important to running or moving material usually in industrial. If the project is to be commercialize, the project surely will have a good market because it can be used for so many things like making small conveyor, small winding machines and many more.

REFERENCES

[1] 29 May 2007, Speech Microcontoller, citing internet source url
http://www.engg.uaeu.ac.ae/units/tra/gra/presentation

[2] 29 May 2007, Programming PIC16F877 microcontroller, citing source url
http://www.imagesco.com/catalog/pic/pic.html

[3] 29 May 2007, PIC introduction, PIC microchip, citing source url
http://www.microchip.com/

[4] 11 June 2007, Stepping Motor Types, citing internet source url
http://www.cs.uiowa.edu/

[5] 11 June 2007, PIC15F877 controller, citing internet source url
http://www.rentron.com/picbasic1/htm

[6] 2 August 2007, Stepper Motor Systems, citing internet source url
http://www.automationdirect.com/motioncontrol/motorsystem

[7] 2 August 2007, Stepper Motor Theory, citing internet source url
http://www.etidweb.tamu.edu/projects/labview/stepper/stepper.htm

[8] 2 August 2007, stepper motor drive ULN2003, citing internet source url
http://www.datasheetcatalog.com

[9] 23 October 2007, stepper motors and control,
http://www.stepperworld.com/

[10] 21 October 2007, stepper motor –free wikipedia, citing internet source url
http://en.wikipedia.org/wiki/Stepper_motor

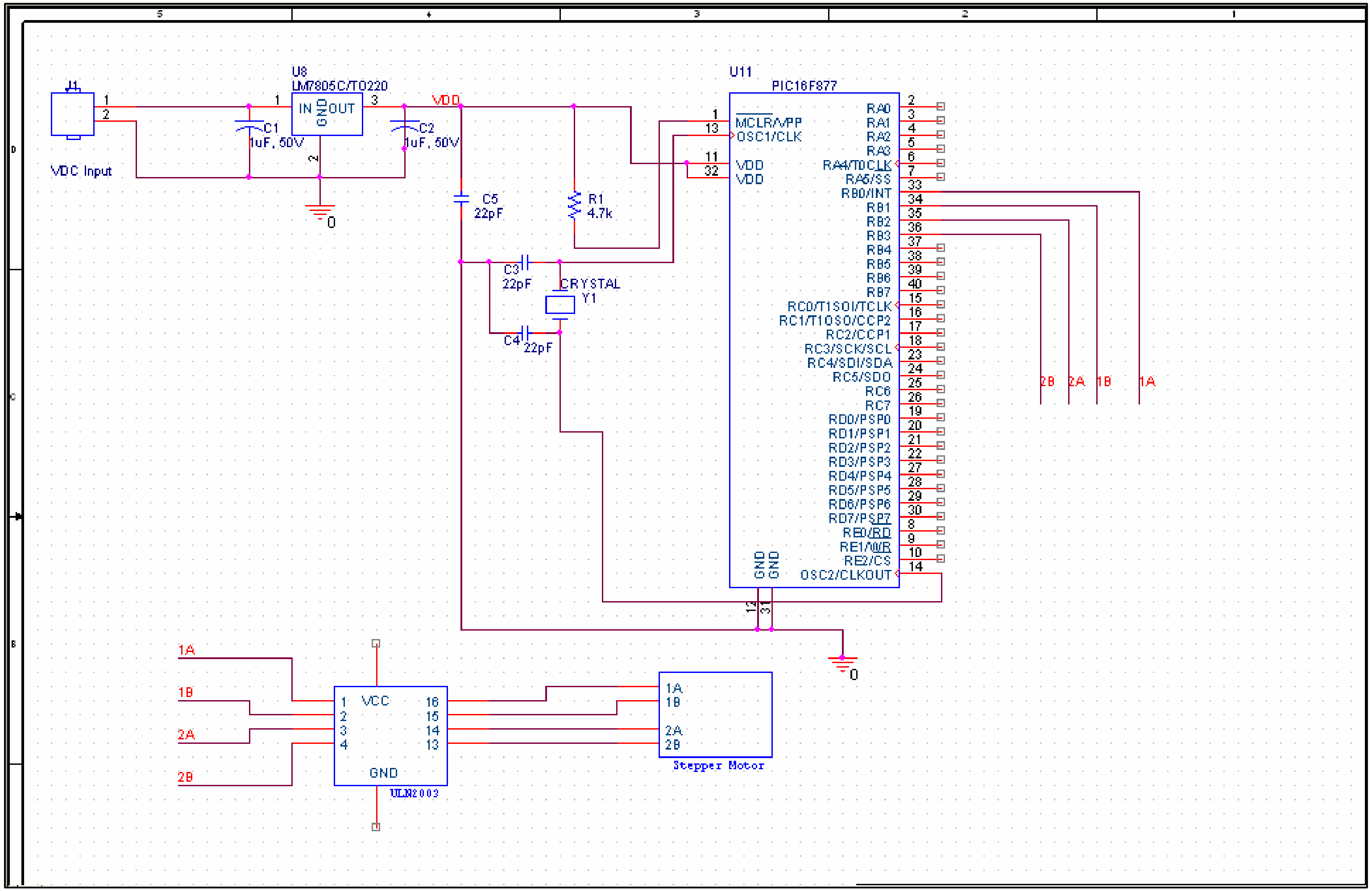
[11] 23 October 2007, controlling stepper motor with a parallel port,
Citing internet source url
http://www.electronics-diy.com/index.php

[12] PIC16F87X Data Sheet 28/40-Pin 8-Bit CMOS FLASH Microcontrollers, 1987, Datasheet: PIC16F877 technical data, Microchip Technology Company

[13] ULN2003A-ULN2004A-ULN2001A-ULN2002A, 1987, Datasheet: ULN2003 technical data, Microchip Technology Company

APPENDIX A

CONTROLLER UNIT SCHEMATIC DIAGRAM


J1
VDC Input
U8
LM7805C/TO220
IN
GND
OUT
C1
1uF, 50V
C2
1uF, 50V
VDD
C5
22pF
R1
4.7k
C3
22pF
C4
22pF
CRYSTAL
Y1
U11
PIC16F877
MCLR/VPP
OSC1/CLK
VDD
VDD
GND
GND
RA0
RA1
RA2
RA3
RA4/T0CLK
RA5/SS
RB0/INT
RB1
RB2
RB3
RB4
RB5
RB6
RB7
RC0/T1SOI/TCLK
RC1/T1OSO/CCP2
RC2/CCP1
RC3/SCK/SCL
RC4/SDI/SDA
RC5/SDO
RC6
RC7
RD0/PSP0
RD1/PSP1
RD2/PSP2
RD3/PSP3
RD4/PSP4
RD5/PSP5
RD6/PSP6
RD7/PSP7
RE0/RD
RE1/WR
RE2/CS
OSC2/CLKOUT
2B
2A
1B
1A
VCC
GND
ULN2003
Stepper Motor

APPENDIX B

FIRMWARE CODE

```
define osc 8

a var byte
b var byte
c var byte
d var byte
trisb = %00000000
start:
for a = 1 to 14
portb = %00000101
pause 10
portb = %00001001
pause 10
portb = %00001010
pause 10
portb = %00000110
pause 10
portb = %00000101
pause 10
portb = %00001001
pause 10
portb = %00001010
pause 10
portb = %00000110
pause 10
portb = %00000101
pause 10
portb = %00001001
pause 10
portb = %00001010
pause 10
portb = %00000110
pause 10
portb = %00000101
pause 10
portb = %00001001
pause 10
portb = %00001010
pause 10
portb = %00000110
pause 10
portb = %00000101
pause 10
portb = %00001001
pause 10
portb = %00001010
pause 10
portb = %00000110
pause 10
portb = %00000101
pause 10
portb = %00001001
pause 10
portb = %00001010
pause 10
portb = %00000110
pause 10
portb = %00000101
pause 10
portb = %00001001
pause 10
portb = %00001010
pause 10
portb = %00000110
pause 10
portb = %00000101
pause 10
portb = %00001001
pause 10
portb = %00001010
pause 10
portb = %00000110
pause 10
portb = %00000101
pause 10
portb = %00001001
pause 10
portb = %00001010
pause 10
portb = %00000110
pause 1000
next a
```

```
for b = 1 to 10
portb = %00000101
pause 10
portb = %00000110
pause 10
portb = %00001010
pause 10
portb = %00001001
pause 10
portb = %00000101
pause 10
portb = %00000110
pause 10
portb = %00001010
pause 10
portb = %00001001
pause 10
portb = %00000101
pause 10
portb = %00000110
pause 10
portb = %00001010
pause 10
portb = %00001001
pause 10
portb = %00000101
pause 10
portb = %00000110
pause 10
portb = %00001010
pause 10
portb = %00001001
pause 10
portb = %00000101
pause 10
portb = %00000110
pause 10
portb = %00001010
pause 10
portb = %00001001
pause 10
portb = %00000101
pause 10
portb = %00000110
pause 10
portb = %00001010
pause 10
portb = %00001001
pause 10
portb = %00000101
pause 10
portb = %00000110
pause 10
portb = %00001010
pause 10
portb = %00001001
pause 10
portb = %00000101
pause 10
portb = %00000110
pause 10
portb = %00001010
pause 10
portb = %00001001
pause 10
portb = %00000101
pause 10
portb = %00000110
pause 10
portb = %00001010
pause 10
portb = %00001001
pause 10
portb = %00000101
pause 10
portb = %00000110
pause 10
portb = %00001010
pause 10
portb = %00001001
pause 1000
next b
```

```
for c = 1 to 6
portb = %00000101
pause 10
portb = %00001001
pause 10
portb = %00001010
pause 10
portb = %00000110
pause 10
portb = %00000101
pause 10
portb = %00001001
pause 10
portb = %00001010
pause 10
portb = %00000110
pause 10
portb = %00000101
pause 10
portb = %00001001
pause 10
portb = %00001010
pause 10
portb = %00000110
pause 10
portb = %00000101
pause 10
portb = %00001001
pause 10
portb = %00001010
pause 10
portb = %00000110
pause 10
portb = %00000101
pause 10
portb = %00001001
pause 10
portb = %00001010
pause 10
portb = %00000110
pause 10
portb = %00000101
pause 10
portb = %00001001
pause 10
portb = %00001010
pause 10
portb = %00000110
pause 10
portb = %00000101
pause 10
portb = %00001001
pause 10
portb = %00001010
pause 10
portb = %00000110
pause 10
portb = %00000101
pause 10
portb = %00001001
pause 10
portb = %00001010
pause 10
portb = %00000110
pause 10
portb = %00000101
pause 10
portb = %00001001
pause 10
portb = %00001010
pause 10
portb = %00000110
pause 10
portb = %00000101
pause 10
portb = %00001001
pause 10
portb = %00001010
pause 10
portb = %00000110
pause 1000
next c
```

```
for d = 1 to 10
portb = %00000101
pause 10
portb = %00001001
pause 10
portb = %00001010
pause 10
portb = %00000110
pause 10
portb = %00000101
pause 10
portb = %00001001
pause 10
portb = %00001010
pause 10
portb = %00000110
pause 10
portb = %00000101
pause 10
portb = %00001001
pause 10
portb = %00001010
pause 10
portb = %00000110
pause 10
portb = %00000101
pause 10
portb = %00001001
pause 10
portb = %00001010
pause 10
portb = %00000110
pause 10
portb = %00000101
pause 10
portb = %00001001
pause 10
portb = %00001010
pause 10
portb = %00000110
pause 10
portb = %00000101
pause 10
portb = %00001001
pause 10
portb = %00001010
pause 10
portb = %00000110
pause 10
portb = %00000101
pause 10
portb = %00001001
pause 10
portb = %00001010
pause 10
portb = %00000110
pause 10
portb = %00000101
pause 10
portb = %00001001
pause 10
portb = %00001010
pause 10
portb = %00000110
pause 10
portb = %00000101
pause 10
portb = %00001001
pause 10
portb = %00001010
pause 10
portb = %00000110
pause 10
portb = %00000101
pause 10
portb = %00001001
pause 10
portb = %00001010
pause 10
portb = %00000110
pause 1000
next d

goto start
en
```

APPENDIX C

LIST OF COMPONENT

No.	Component	Specification	Cost per unit	Quantity	Cost
1	ULN2003		RM1.20	1	RM1.20
2	PIC16F877		RM25.00	1	RM25.00
3	Crystal	4Mhz	RM1.20	4	RM4.80
4	Independent Board		RM5.00	1	RM5.00
5	Capacitor	30pF	RM0.07	2	RM0.14
6	Capacitor	10uF	RM0.07	1	RM0.07
7	Capacitor	22pF	RM0.07	2	RM0.14
8	Resistor	4.7k ohm	RM0.06	4	RM0.24
9	IC base	16pins	RM0.20	2	RM0.40
10	IC base	20pins	RM0.20	1	RM0.20
11	IC base	40pins	RM0.20	1	RM0.60
12	Max232		RM4.00	1	RM4.00
13	74LS205		RM1.05	1	RM1.05
15	Wire wrap		RM15.00	1	RM15.00
				Total Cost	RM57.87