Drawbot: A Microcontroller Based Drawing Robot

by

Daniel B. Chua
Ryan Henry C. Latigay
Toni Rose C. Panganiban
Virnali G. Sy

A Design Documentation Submitted to the
School of EE-ECE-CoE
in Partial Fulfilment of the Requirements for the Program

Bachelor of Science in Computer Engineering

Mapua Institute of Technology
March 2008

APPROVAL SHEET

This is to certify that this design study entitled **“Drawbot: A Microcontroller Based Drawing Robot”** prepared by **Daniel B. Chua, Ryan Henry C. Latigay, Toni Rose C. Panganiban, and Virnali G. Sy** in partial fulfilment of the requirements for the degree **Bachelor of Science in Computer Engineering** have been supervised the preparation of and read the design documentation and hereby recommended for final examination by the Oral Examination Committee.

____________________	Engr. Marc Joseph B. Cabrera
Reader	Adviser

As members of the Oral Examination Committee, we hereby **APPROVED** this design study which was presented before a Panel of Examiners of the School of EE-ECE-COE on March 17, 2008.

Engr. Analyn N. Yumang	Engr. Ernesto M. Vergara Jr.
Panel Member 1	Panel Member 2

Engr. Donnalee Silanga
Panel Member 3

Accepted in partial fulfilment of the requirements for the degree **Bachelor of Science in Computer Engineering.**

Dr. Felicito Caluyo
Dean, School of EE-ECE-CoE

ACKNOWLEDGMENT

Our design would not be a reality without the help of those who took part in finishing our project. With this, we would like to thank all of them.

First of all, we would like to thank our Lord Almighty who gave us the wisdom to come up with the ideas and concepts for this project. We also thank HIM for giving us the strength and time to finish our project, our Drawbot.

We also would like to thank our parents for giving us all the love and care, for providing us our needs and for their ever willing financial support.

Our gratitude likewise goes to our friends who willingly and patiently helped us no matter what; to the other computer design groups who helped us on finding the components we were seeking; and to our Mapua professors for sharing their profound knowledge and wisdom to our young minds and for equipping us with the proper skills we need in our chosen field.

Our group would also like to extend our gratitude to Engr. Marc Joseph Cabrera our adviser. The time we spent with him to help us improve the prototype was all worth it.

And most especially we want to thank our professor, Engr. Noel Linsangan, for giving us the opportunity to show and enhance our talents and knowledge.

Daniel B. Chua
Ryan Henry C. Latigay
Toni Rose C. Panganiban
Virnali G. Sy

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

A robot is said to be an automatic device that performs functions normally attributed to humans or machines. The study was made in order to gain critical skills in computer engineering and to explore the use of microcontrollers through a robot which could be directed to perform a specific task. The design named Drawbot has the capability of drawing basic shapes with a definite size. These basic shapes are circle, square, triangle and rectangle with sizes of small, medium, or large. The main purpose of this study was to design a robot that used stepper motors and PIC microcontroller to draw. The PIC microcontroller used was the PIC16F877. This microcontroller was chosen because of its numerous bidirectional I/O ports. It will receive data from the user through the push buttons and control the stepper motors to perform specific actions based on the data. The study was conducted by using descriptive research to determine the extent to which the method would yield data supporting similar conclusions about the design being developed. The design draws the shapes in a single sweep which a person cannot normally do and which may lead to inaccuracy of the drawing. Using the design could make drawing faster thus save a great amount of time for the user since it does the work without stopping until it reaches the end of its course.

Keywords: Microcontroller, PIC, PIC16F877, bidirectional I/O port, Stepper motor

Chapter 1

INTRODUCTION AND REVIEW OF RELATED LITERATURE AND STUDIES

Research Setting

The development of robots is motivated by man's inner wish to build human-like creatures. Robots should resemble us both physically and intellectually, displaying human-like behaviors. A robot is said to be an automatic device that performs functions normally attributed to humans or machines. It could be "a mechanism which moves and reacts to its environment" by Ben Nagler in his year 2000 published book, Your First Robot. As it is defined, it already pertains to a mobile robot. It comes as no surprises since all humans, animals and insects are the typical role models for this robot genre.

A non-neglecting part of our time is devoted to entertain ourselves. Thus, while we may provide robots with the capacity to perform useful tasks, we may also want those to entertain us. Across the centuries, humans have shown an eager interest in drawing. There are many ways to draw. The basics of them all are the shapes. Shapes are the basic technique for image processing that have slowly crawled their way into our everyday making of a picture. In this paper, the researchers utilized entertainment as an interesting application for mobile robots. As an improvement, the researchers have thought of a way to utilize the use of a mobile robot and add another specific task – the process of drawing. This project was named DRAWBOT which is a combination of two words, draw and robot. The design is a drawing mobile robot, which has the capability of drawing shapes in a definite size. This robot has the ability to create basic shapes and these are the circle, square, triangle and rectangle.

Review of Related Literature and Related Studies

ROBOTS

A robot is a machine designed to execute tasks repeatedly that is a lot advantageous than human capability in terms of speed and precision. Robots are capable of performing human tasks often in environments that are too hazardous for humans or in situations that are too repetitious or tedious for humans. Just as long as there are tasks to perform, different types of robots can exist.

The word "Robot" comes from the 1921 play "R.U.R." (Rossum's Universal Robots) by the Czech writer Karel Capek (pronounced "chop'ek"). "Robot" comes from the Czech word "robota", meaning "forced labor." This came from Moravec, Hans in his book, Robot, published in 1999.

Robots may be controlled directly by a human, such as remotely-controlled bomb-disposal robots, robotic arms, or may act according to their own decision-making ability provided by artificial intelligence. However, the majority of robots fall in between these extremes, being controlled by pre-programmed computers. Such robots may include feedback loops such that they can interact with their environment. This will involve much more complex design. Robots can be classified into categories depending upon their function and the purpose for which they are designed for. In this research, robots are classified into two categories, Industrial robots and Service robots.

INDUSTRIAL ROBOTS

According to the Robotic Industries Association, an Industrial robots is an automatically controlled, reprogrammable, multipurpose manipulator programmable in three or more axes which may be either fixed in place or mobile for use in industrial automation application.

These robots are growing in complexity and their use in industry is becoming more widespread. The main use of robots has so far been in the automation of mass production industries, where the same, definable, tasks must be performed repeatedly in exactly the same fashion. Car production is the primary example of the employment of large and complex robots for producing products. Robots are used in that process for the painting, welding and assembly of the cars. Robots are good for such tasks because the tasks can be accurately defined and must be performed in same way every time.

SERVICE ROBOTS

According to the International Federation of Robotics, another professional organization, a service robot is a robot which operates semi or full automatically to perform services useful to the well-being of humans and equipment, excluding manufacturing operations.

Personal robots are service robots that educate, assist or entertain at home. These include domestic robots that perform daily chores and robots that can serve as companions or pets for entertainment. Now available are domestic robots that perform simple tasks such as vacuum cleaning and grass cutting.

Mobile robots fall on this category. They are robots capable of movements on their own. One good example is an autonomous vehicle. The field of mobile robot navigation is active and vibrant, with more great systems and ideas being developed continuously. Mobile Robots generally fall into two classes, linked manipulators and mobile robots. Mobile robots have the capability to move around in their environment and are not fixed to one physical location. In contrast, industrial manipulators usually consist of a jointed arm and gripper assembly that is attached to a fixed surface. Mobile robots are the focus of a great deal of current research and almost every major university has one or more labs that focus on mobile robot research. Mobile robots are also found in industry, military and security environments. This was mentioned in Joe Campbell's book, Assembly Robots 101, which was published in 2004.

Basically a mobile robot is provided with a sensor-based and map-based navigation system for navigating in its environment. The navigation is based on the classification of pre-existing instructions. This system can then compensate for an accurate robot system's motion control, sensor information, and landmark classification.

Other domestic robots are aimed at providing companionship or play partners to people. Examples are Sony's Aibo, a commercially successful robot pet dog, Paro, a robot baby seal intended to soothe nursing home patients. Other humanoid robots are in development with the aim of being able to provide robotic functions in a form that may be more aesthetically pleasing to customers, thereby increasing the likelihood of them being accepted in society.

PERSONAL ROBOTS AND CYBORGS

The new information technology focuses on the development of robot technologies. The information society is characterized by an increasing speed in accessing information. The development of robotics during the last decades confirms this observation. Personal Robots and Cyborgs employ the idea of allowing information to flow without a human body as interface. By looking into the point of view of those involved in the construction of technical subsystems into the human body, the designers were able to understand how they considered a human being through a robot as well as the boundaries it undertook from it.

Personal robots are analogous to personal computers appointed to one or more persons. They are defined as robots which share physical and emotional spaces with the user. Their working field lies in everyday areas: business, household and recreation. Personal robots are characterized by interactivity, autonomy, intelligence and a close link to man. They denote the relationship "robot – individual". The distance between man and machine is close to but greater than zero. In the book, Robots do it better, R. Siegwart, K.O. Arras, H. Sachs, C. Scheidegger,M. Schnegg mentioned the evolution of personal robots and cyborgs. This was published in 2000.

Cyborgs (from cybernetic organism) are systems whose functions are based on an irreversible union of an organic-human and technical subsystems. Typical properties are biocompatibility, distributed intelligence and partial autonomy. Examples include pacemakers, mechatronic limbs, hearing and optical aids directly attached to the nervous system or wheelchairs for the seriously disabled. The distance between man and machine disappears. In the book, Robots do it better, R. Siegwart, K.O. Arras, H. Sachs, C. Scheidegger,M. Schnegg mentioned the evolution of personal robots and cyborgs. This was published in 2000.

GRASS DRAWING ROBOT

This amazing invention gives the designers the idea of a Drawbot. Similarly drawing lines but of different outcome and approach. Here the Translator II: Grower as the name of the robot is a rover robot which navigates hugging a room's walls and responding to the carbon dioxide levels in the air by drawing varying heights of "grass" on the walls in green ink.

Grower senses the CO2 level in the air via a digital sensor. The more people in the exhibit space breathing in oxygen and exhaling CO2, the higher the grass line. This simulated grass is just like the grass found in nature: it needs CO2 to grow. This was a research done by Sabrina Raaf in her Grass Drawing Robot study done in 2004.

Conceptual Framework

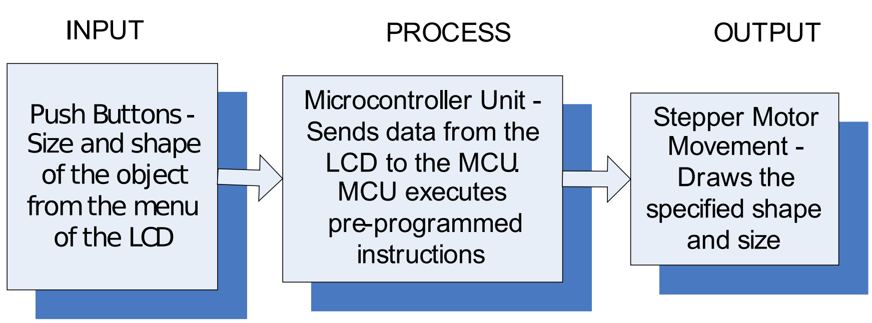


Figure 1 – Conceptual Framework

When the Drawbot is turned on, the LCD will display the options of the menu that are available. Refer to Figure 1 – Conceptual Framework. The system should receive its input from the push buttons.. The Drawbot has three push buttons that can control the menu on the LCD namely; left[<], right[>], enter[E]. The options on the menu of the LCD will enable the user to input the desired shape and size. The menu has twelve options to choose from which correspond to the format [size] [shape]. The definite sizes are small, medium and large. The small sizes are the ones presented first followed by the medium and large sizes. There are only four shapes to choose from namely circle, square, triangle and rectangle. The shapes together with the sizes are the inputs of the system that will come from the LCD.

The Drawbot will process the data and automatically execute the instruction given by the microcontroller unit. The output of the microcontroller will run the stepper motor drivers. This in turn will execute the movement of the stepper motors. The turning of the stepper motors will draw the shape desired by the user.

Statement of the Problem

The study teaches critical skills to computer engineering students. The project was time consuming but invaluable to the designers for mastering the skills needed for designing embedded microcontroller prototypes. As the title suggests, the designers focused on robots to apply this designing method. In this study, exploring the use of microcontroller to create the robot will be analyzed. The kind of robot that can be created with its use should be directed to do specific actions. These will be its navigation and it will be intended to implement shape drawing. Its programming capabilities should help navigate the pre-determined path.

Objective of the Design

The main objective of the group is to be able to design a robot that uses stepper motors and PIC microcontroller to draw the following objects: circle, square, triangle, and rectangle. Specifically, the group wishes to attain the following:

1. to program a microcontroller that can receive data and control the stepper motors to perform specific actions based on the data.
2. to design the PIC in a way that it controls the robot to move in a programmed path similar to drawing the shapes mentioned.

Significance of the Design

For us students, the design is very important because through this study, the designers are able to apply and practice their technical skills. The things learned on this research were mostly the components and how they could be used to develop the design. This study showed the PIC16F877 capability as a microcontroller. This study was an improvement of the mobile robot. The study suggests that the PIC16F877 can be programmed to move on a specified path by controlling the stepper motors. The paths are the shapes that were stated. Doing this study could serve as future reference for other related studies. The PIC features and stepper motor capability shown through the design suggests the flexibility of the microcontroller programming.

For the community and common people, technology comes in by having a robot or a machine that can be used as a teaching material. Drawing basic shapes would help a child familiarize himself with the use of technology. The task would be simpler because the parent would only need to press a button and the robot would be the one to draw the shape. In the industry, it could be used for automatic drawing like painting jobs. It would no longer need a person to manually draw. He would only need to push the buttons on the robot.

Scope and Delimitations

The designer's group prototype named Drawbot has the capability of drawing shapes in a definite size. Like any other designs, it has limitations when it comes to its results and processing's. Listed are the capacity and extent of its functionality.

1. The user can manually turn the Drawbot on or off using the switch button.
2. The three buttons represent the control of the LCD for user input.
3. The pen hole holds the writing instrument and a screw is used to lock it in place. The pen hole can basically hold a regular sized pencil.
4. The options for the shapes and sizes are displayed on an LCD.
5. The four (4) shapes, namely circle, square, rectangle and an equilateral triangle that have three (3) different sizes which are small, medium and large are the options that can be selected.

6. Stepper motors are used to move the Drawbot when drawing any of the three shapes; triangle, square and rectangle, depending on its size and a gear motor is used to move the pen when drawing the circle.

The Drawbot also has its limitations. Due to the designer's beginning expertise on the subject, it was not unavoidable for the Drawbot to have its setback. The following limitations are listed.

1. It needs a 12 volt electric cell to supply the voltage needed. It consumes the battery in order for it to work which can last for a maximum of four hours.
2. The Drawbot can only run on a smooth surface (uneven flooring and irregular surfaces are to be avoided).
3. Collision during the drawing process will produce a distorted output
4. The drawing of shapes will be limited within the exact measurements programmed in the microcontroller. (Refer to Table 1)

Shape	Size	Measurement in inches	Measurement in cm
Circle (radius)	Small	4 in	10.1 cm
	Medium	4.5 in	11.4 cm
	Large	5 in	12.7 cm
Equilateral Triangle (one side)	Small	6.5 in	16.5 cm
	Medium	9.5 in	24.1 cm
	Large	16 in	40.6 cm
Square (one side)	Small	6.5 in	16.5 cm
	Medium	9.5 in	24.1 cm
	Large	16 in	40.6 cm
Rectangle (length x width)	Small	9.5 x 6.5 in	24.1 x 16.6 cm
	Medium	12.5 x 9 in	31.8 x 22.9 cm
	Large	15.5 x 12.5 in	39.4 x 31.8 cm

Table 1 –Drawbot Shape Measurments

5. There is a small discrepancy that would be left as a gap for the rotation of the pen.
6. The Pen should be placed on the Drawbot at a stationary point.

Definition of Terms

1. Current – Electric current is the flow of electric charge. (Electronics and Radio Today – Basic Electronic Concepts)
2. Direct Current – Direct current (DC or "continuous current") is the constant flow of electric charge. (Electronics and Radio Today – Basic Electronic Concepts)
3. Electricity – Electricity is a general term for a variety of phenomena resulting from the presence and flow of electric charge. (Electronics and Radio Today – Basic Electronic Concepts)
4. High Impedance – It means high resistance, that is, there is no current flow. (Electronics and Radio Today – Basic Electronic Concepts)
5. I/O port – It refers to the input and output pin in a microcontroller. (*Lessons In Electric Circuits* copyright (C) 2000-2008 Tony R. Kuphaldt)
6. Microcontroller – A microcontroller is a computer-on-a-chip. It is a type of microprocessor emphasizing self-sufficiency and cost-effectiveness, in contrast to a general-purpose microprocessor. (*Lessons In Electric Circuits* copyright (C) 2000-2008 Tony R. Kuphaldt)
7. Power – Electric power is defined as the rate at which electrical energy is transferred by an electric circuit. (Electronics and Radio Today – Basic Electronic Concepts)

8. RAM – It is the data memory used by a program during its execution. (*Lessons In Electric Circuits* copyright (C) 2000-2008 Tony R. Kuphaldt)
9. Tristate Circuitry – It is a special kind of digital circuitry which allows control of I/O pins. (*Lessons In Electric Circuits* copyright (C) 2000-2008 Tony R. Kuphaldt)
10. Volt – The International System unit of electric potential and electromotive force, equal to the difference of electric potential between two points on a conducting wire carrying a constant current of one ampere when the power dissipated between the points is one watt. (Columbia Encyclopedia)
11. Voltage – Voltage which sometimes also called electric potential or electrical tension is the difference of electrical potential between two points of an electrical or electronic circuit, expressed in volts. (Electronics and Radio Today – Basic Electronic Concepts)

Chapter 2

METHODS AND PROCEDURES

Research Design

This study utilized the method of descriptive research in order to determine the extent to which the method would yield data supporting similar conclusions about the design being developed. For the descriptive analyses, certain topics about the components were gathered to determine what could be used to further improve the design. The capability and specifications of the components that were gathered through readings were carefully matched for the system to work.

The design was also subjected to an assessment and testing where it was exposed to functional analyses that manipulated variables in an ideal environment. In this case, the device underwent certain trials where all its functions, the shape drawings, were tested under a fully charged battery. The results were analyzed to identify relevant events for the development of the device. Using the outcomes of the testing analysis as the standard for comparison, the results indicated that the descriptive analysis was useful in identifying the components that affected the enhancement of the Drawbot.

The data gathering procedure is shown in Figure 2. The flowchart presents the problem analysis. Certain information was collected through readings and related topics were likewise gathered for the development of a solution. This procedure helped the researchers in choosing the components and important supplies that made up the system. A draft was constructed out of the information that was brought together. This draft served as the blue print for the design. The next step was the determination of the supplies that would be used. The resources were first completed before the construction of the design. When the design was fully assembled, testing and analysis of the desired output took place.

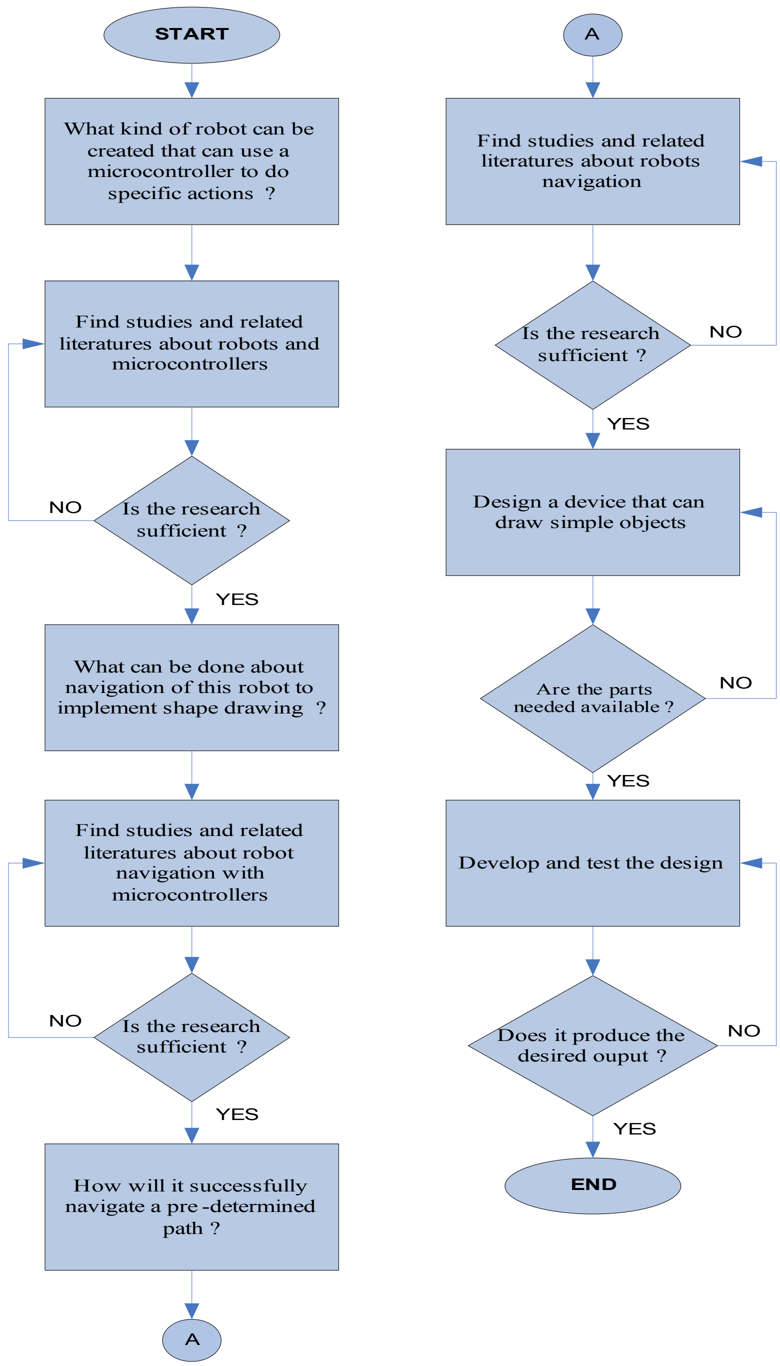


Figure 2 – Data Gathering Procedure Flowchart

Figure 3 indicates the flow of the system which starts with the push buttons, the one responsible for accepting inputs from the user to control the menu on the LCD. The buttons represents three types of input which are left, right and enter key. The menu displays the size and shape to be drawn. The input is then passed on the microcontroller which will process the data using the specific instruction programmed in the unit. The stepper motors or the drivers are then moved depending on the instructions given by the microcontroller. Finally the motor creates the desired output.

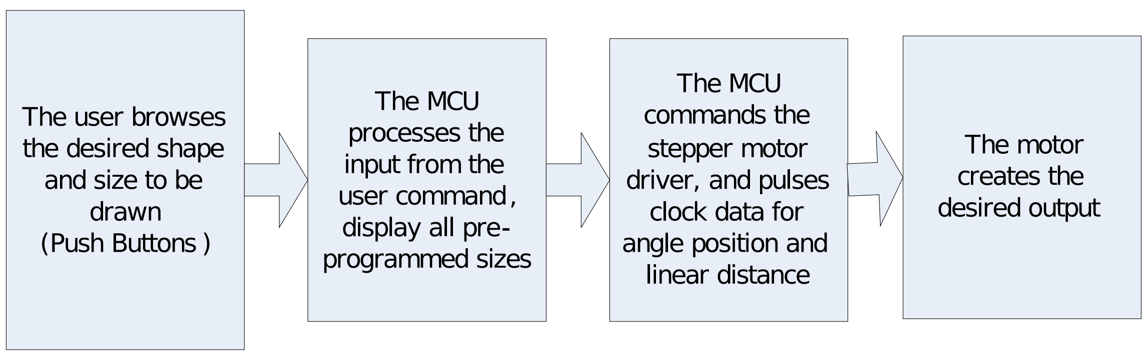


Figure 3 – System Flow Diagram

System Flowchart

Figure 4 presents the system flowchart that shows the flow of data and operations for the whole design. When Drawbot is turned on, a welcome message appears on the LCD. It will then wait for an enter-key press for it to display the menu. After the menus is displayed, the user can press the right or left key corresponding to the [size][shape] option the user wants the Drawbot to draw. The microcontroller then processes the option to perform the necessary instructions. The output will be passed to the stepper motor drivers which will operate the stepper motors. The movement will cause the robot to draw the shape.

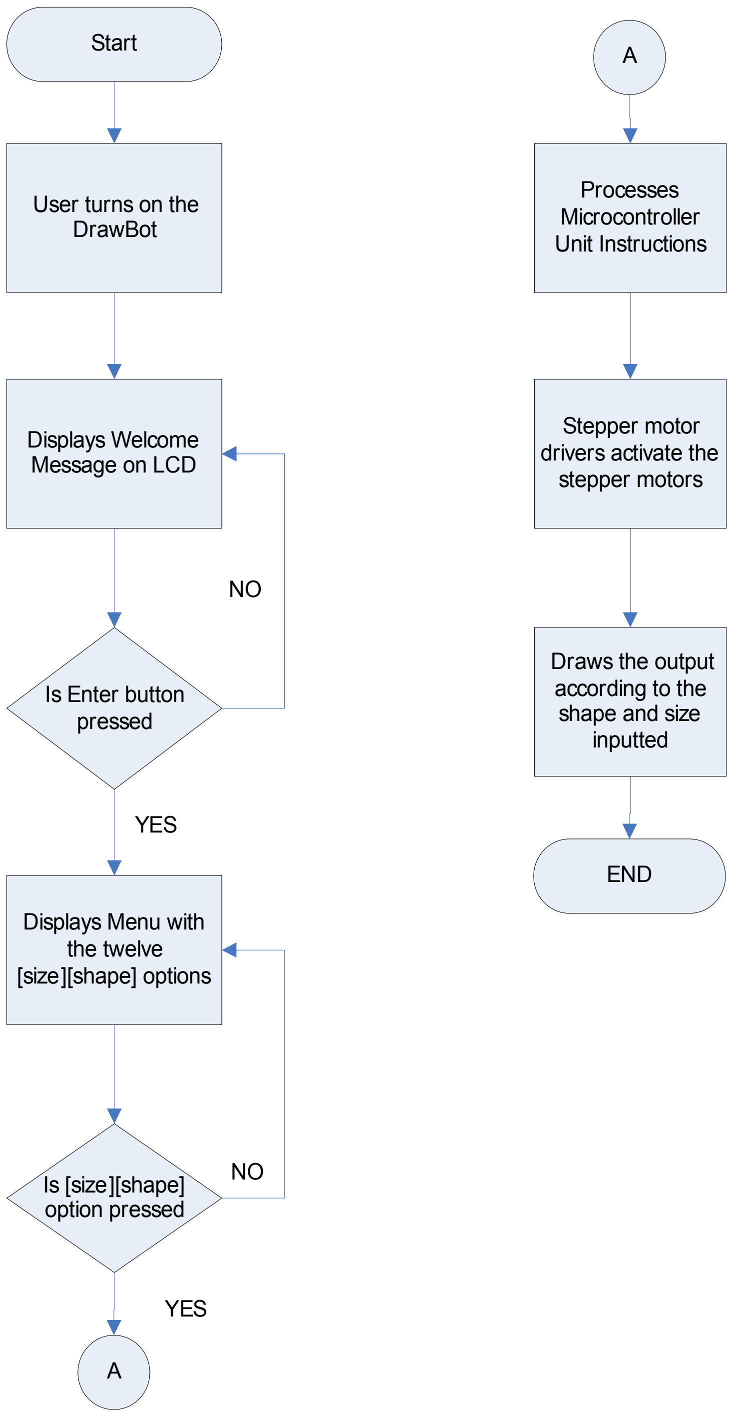


Figure 4 – System Flowchart

Hardware Design

The hardware design is shown as a system block diagram as presented in Figure 5. The Drawbot system would comprise of components that would be further explained. The LCD display with three push buttons as the key switches. The key switches are labeled left[<], right[>], and enter[E]. The microcontroller unit consists of the PIC16F877 for the main program. Another microcontroller, PIC16F84A serves as a stepper motor driver. The relay driver will be responsible with the DC Gear Motor. And lastly, the stepper motors and the gear motor itself. The LCD display can be manipulated by the push buttons. Then, the microcontroller unit controls the stepper motors. The other parts, the stepper motors and wheel, are controlled by the microcontroller unit. The two stepper motors are connected to the two wheels. These are the locomotive parts that produce the motion ability of the robot.

The key switches will serve as input devices. This will browse the size and shape. The inputted values will then be processed in the Drawbot's microcontroller. Accumulated results will drive the motors inside the Drawbot. Lastly, the stepper motors are the ones responsible for moving the wheels of the Drawbot.

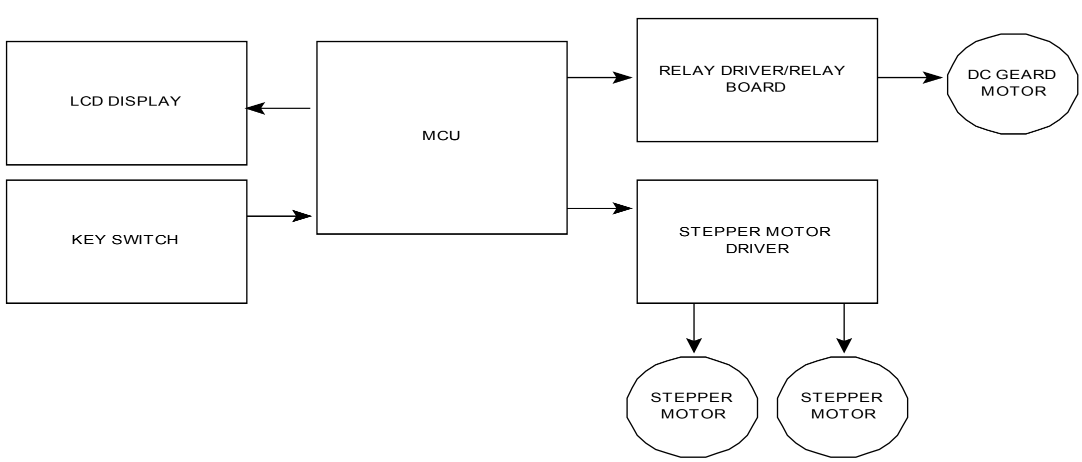
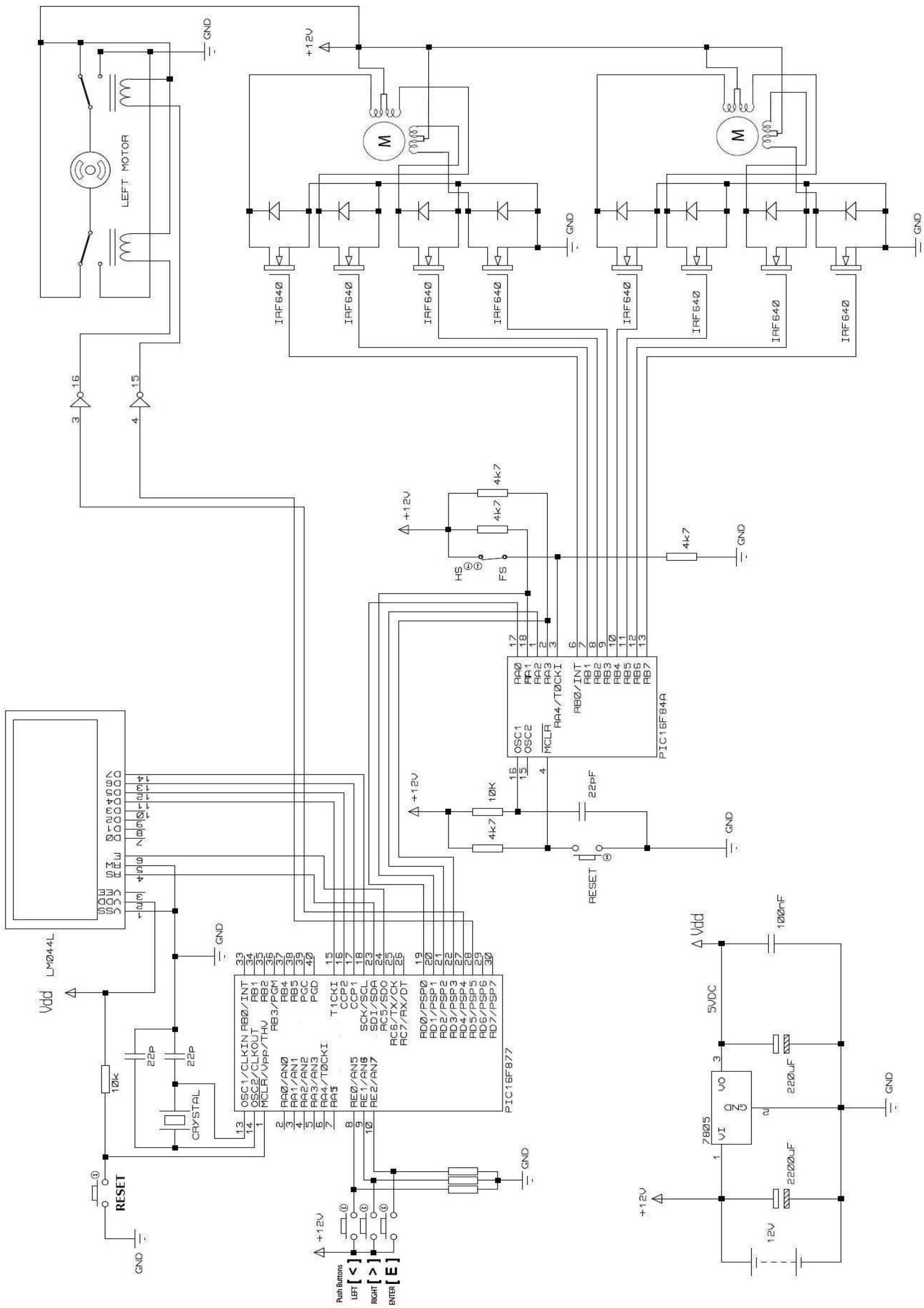


Figure 5 – System Block Diagram

Circuit Design



The heart of the Drawbot is the PIC16F877 microcontroller. It controls all the functions including the drawing measurements and LCD display functions. The full schematic diagram of the Drawbot is shown in Figure 6.

Figure 6 – Full Schematic Diagram of Drawbot

Upon power ON, all ports of the PIC16F877 are initialized. Then, the LCD display shows the main menu and put it in standby mode. Pressing the Enter Key will show all pre-programmed functions of the circle, square, triangle and rectangle. Selecting the desired shape and size, the firmware will execute the appropriate shape routine. The PIC16F877 will pulse a signal going to another microcontroller, the PIC16F84A, and turn on the transistor switch IRF640 to rotate the stepper motor. The number of pulses is equivalent to the length of travel or rotation of the stepper motor. The directions of stepper motor are also controlled by the PIC16F877 by putting its pin to high or low output. The microcontrollers are controlled by a firmware program. The LM7805 regulates the 12volts DC input to +5volts required by the microcontrollers.

Another motor is the DC gear motor. It is meant for sizing the shape circle. It moves on the left if the circle size is small, on the middle if the size is medium and on the right if the circle drawing is large. The motor rotations on the left and on the right are controlled by two relays configured as H-bridged configuration. It means that supplying the motor with the correct polarity will move the motor on a clockwise direction while on the other side the motor rotates counter clockwise.

All inputs are tied at ground which means that the microcontroller inputs are triggered only in high inputs and all levels are monitored by the microcontroller program. Both microcontrollers (PIC16F877 and PIC16F84A) are clocked at 4MHz. The DC gear motor and stepper motor are directly supplied on its 12volts DC lead acid maintenance free battery.

List of Materials

RESISTORS	
QTY	**VALUE**
4	**4k7**
5	**10k**
CAPACITORS	
QTY	**VALUE**
2	**22p**
1	**22pF**
1	**220uF**
1	**2200uF**
1	**100nF**
INTEGRATED CIRCUITS	
QTY	**DESCRIPTION**
1	**PIC16F877**
1	**ULN2803_1**
1	**7805**
1	**PIC16F84A**
TRANSISTORS	
QTY	**DESCRIPTION**
8	**IRF640**
DIODES	
QTY	**DESCRIPTION**
8	**1N4001**

Table 2 – List of Materials

Hardware Components

Component Name: Push Buttons

The Push buttons serve as an input device of the key-in circuit. The device consists of 3 switches namely “<”, “>” and “E”.

Component Name: Microcontroller

Microcontroller is responsible for all the process of the system.

PIC16F877 –This MCU processes the inputs obtained from the push switches and it is configured to receive the input signals at PORTE and to produce output signals at PORTC.

Component Name: Stepper Motor Driver

Stepper motor driver is responsible for controlling the stepper motors.

PIC16F84A – serves as the Stepper motor driver for the stepper motors. these are the stepper motor drivers and they are being used as buffers for the output signals to drive the stepper motors since the microcontroller can not produce sufficient current to drive these motors. The output current ranges from 1.3A up to 3A. PIC16F84 is a stepper motor driver. It is a motor control integrated circuit that particularly handles a DC motor. In this design, it pertains to the stepper motor.

Component Name: Stepper Motor and Gear Motor

Stepper motors are responsible for the movements of the wheels. These stepper motors are being controlled by the MCU (Microcontroller). They serve as the muscle part of the Drawbot and provide mobility by turning the wheels. The gear motor is responsible for moving the pen when drawing the circle shape on its desired size. It is also controlled by the microcontroller.

Component Name: Lead Acid Battery

The Lead Acid Battery is used to provide power voltage in the Drawbot system. It supplies 12 V, 1.3A/hr and can last up to 4hrs depending on its usage.

Software Design

The main program is created using PICBASIC. The PICBASIC compiler made it easy for the designer team to write programs for the PIC16F77 Microcontroller. The system programmability of this chip makes the flexibility of the design possible after assembling and testing have been completed. This capability can be used for easy PICBASIC programming. It can be used to improve programs on finished design prototypes. The program of the Drawbot is explained below. It covers two parts. The initializations needed for the microcontroller to functions are declared. The next part is about the simple flow of operations within the program on how it controls the stepper motor and how the researchers programmed the menus on LCD display.

Initialization Processs

During the initialization process, the ports to be used on the microcontroller were declared. The I/O mode of port A, B, C, D, and E were setup. In the development of the Drawbot, the ports E, C, and D were the ones used. Port E was utilized for the push buttons. This served as the only input on the system. Port C and D were for the outputs. Port C was connected with the LCD display and Port D was connected to the stepper motor drivers for the controlling of the stepper motors and gear motor. The following declarations are shown.

```
Device = 16F877
Declare XTAL 4
```

This defined the device to be used as the microcontroller PIC16F877. The crystal oscillator has a capacity of 4Mhz.

```
LCD_DTPIN = PORTC.0
LCD_RSPIN = PORTC.4
LCD_ENPIN = PORTC.5
LCD_INTERFACE = 4
LCD_LINES = 4
LCD_TYPE = 0

TRISC=%00000000
TRISB=%00000000
TRISD=%00000000
TRISE=%111
```

These were the port modes that were declared for the microcontroller. The LCD was defined for port C. Each pin configuration was declared. The next part pertained to the tristate mode of the circuitry. This tristate circuitry controls whether the given port would be an output or an input.

On the work area initialization, the only literal used aside from the declaration of the ports was the literal i declared as byte. It was used for the looping of the pulses that would move the stepper motor. This is shown below.

```
Dim i As Byte
```

Stepper motor Routine and Procedures

The controlling of the stepper motors was the next part of the program. The sample procedure below illustrates how the flow of operation on the main program works. It also explains how the researchers were able to find a way to control the stepper motors.

```
label:

    High PORTD.0
    For i = 0 To 300
    PORTD.1=i
    DelayMS 1
    Next i
    Low PORTD.0
    Low PORTD.1
    Return
```

In controlling the stepper motors, the particular pin on port D should be enabled. The literal i would then be used to loop the pulse of the opened (or high enabled) pin. The delay would serve as the incremental of the literal i. The next step would be to close the high enabled pins to stop the motor. Each pulse corresponds to the movement of the stepper motor. This would convert the movement to the length of any shape. The movement of each stepper motor would be dependent on the shape that would be used as well as the pins to be enabled to.

```
Gosub label            causes the program to call a subroutine
                       base on the index value. A subsequent
                       RETURN will continue the program
                       immediately after the On Gosub command.
```

The gosub command was then used to cause the program to call a subroutine. The subsequent RETURN command would continue the program immediately after the On Gosub command.

<u>LCD screen layout Routine and Procedure</u>

After the LCD configurations were defined at the initialization, the simple command for displaying the output would be the Print At command together with the line number and the character position of the text. The sample displays are shown below.

```
disp2:
    Cls
    Print At 1, 1,  "    Chua, Daniel    "
    Print At 2, 1,  " Latigay, Ryan Henry"
    Print At 3, 1,  " Panganiban, Toni R."
    Print At 4, 1,  "     Sy, Virnali    "
```

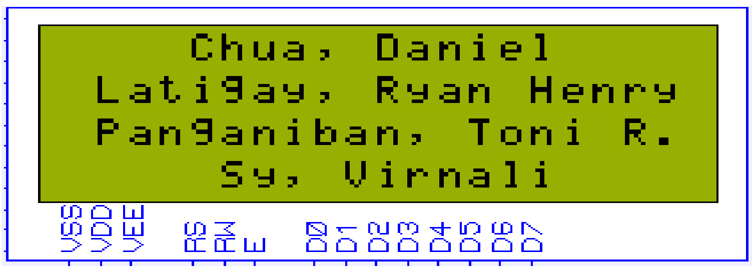


Figure 7 – LCD welcome message of the Drawbot

```
main:
        Print At 1, 1, "      DRAW BOT      "
        Print At 2, 1, " [<] Small CIRCLE   "
        Print At 3, 1, " [>] Small SQUARE   "
        Print At 4, 1, " [E] NEXT           "
```

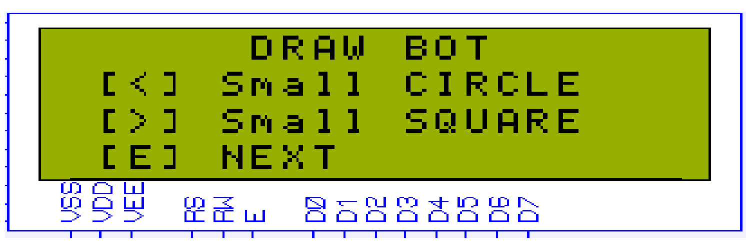


Figure 8 – LCD Menu layout of the Drawbot

Chapter 3

PRESENTATION AND INTERPRETATION OF DATA

Accuracy of the Design

The testing of the design required the checking of the shapes on their sizes. This would test the accuracy of the design with its desired figure. In testing, while the Drawbot was switched to ON, the three buttons representing the browsing of the shapes will be the ones pressed first. The Drawbot then automatically performed its process based on the inputs.

There were problems encountered during the assemblage and testing. The overall function of the Drawbot was not in order. This was due to the uncharged battery. It is recquired that the lead acid battery be checked first for its capacity for the Drawbot to work on full operation. The sizes were the last that occurred. The designers made it sure that the small shapes were the first ones to be successfully drawn before programming the IC for the larger sizes.

On the trials, the user first pressed an option for the desired size and shape. The user checked the exact measurement of the sizes that were drawn. In obtaining the accuracy of the design, the researchers were faced with several questions on how to prove it. The first question that was asked was "what data should the researchers need to get for the testing?”. Knowing what data to look for, the researchers now prepared the environment to which testing would be done. In this set-up it was assumed that all components were in proper working condition including the supply of the battery. There would only be five trials with each size and shape. The next question concerned with how the designers would prove the accuracy of the design. The step by step procedure and the formulas that were used are presented to further explain the process of obtaining the accuracy of the design and the reason for using such procedure and formulas.

The trials done on the Drawbot helped the researchers anticipate any errors that might occur. The checking of the battery on how long it would last was considered during the simulation. It could last the simulation for 4 hours on an assumption that the lead acid battery was fully charged.

Assumptions before the trials:

1. The pen is at a stationary position.
2. The pen to be used is a fine-tip point ball pen to avoid error in the measurement of the thickness of the drawing.
3. All measurements are in centimeters.
4. The circle has no error that is, perfect circle.
5. The length of the rectangle is 3 inches bigger than the width of the rectangle. Converted to centimeters, it is 7.6 cm.

Step by step Procedure

1. Measure the actual shape drawing. The obtained measurement will be the data needed to prove the efficiency of the design to draw the desired shape.

   The designers came up with the solution for testing the Drawbot by calculating the approximation of errors in measurement. The plan was that if a quantity x (e.g. side of a square) is obtained by measurement and a quantity y (e.g. area of the square) is calculated as a function of x, as $y = f(x)$, then any error involved in the measurement of x produces an error in the calculated value of y as well. The error in x can be considered as a change in x, and thus is denoted by Δx . On the succeeding trials, it meant that by calculating y denoted as $dy = f'(x)$ dx, the area of the shape, what is actually obtained was the allowable percentage error of the shape since the error in the measurement of x has been associated with the calculated value of y.

2. Get the absolute error of the corresponding dimension by $dx = actual\ measurement - ideal\ measurement$

   On the trials, the x corresponds to the measured dimension of the drawn shape. The Δx will then be from its comparison between the ideal measurement of the dimension and the actual measurement from the trial. The ideal measurement would come from the programmed size of the microcontroller. The actual measurement would be assessed on the five trials. The average of the five trials will be the error in x.

3. Get the allowable percentage error of the calculated area.

   The y represents the calculated derivative area of the shape. By using the error in x the allowable percentage error of the calculated area can be determined as $dy = f'(x)\ dx$. This would be the allowable percentage error of the calculated area because it was related to its corresponding error that was caused by the actual measurement. By getting its derivation, the allowable would be based on the absolute error made on the measurement.

For each shape, the area formulas are in Table 3.

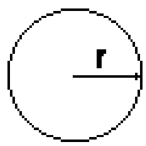
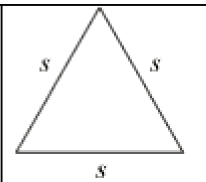
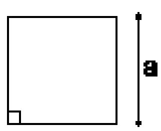
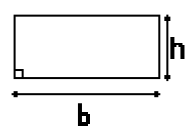
 Circle Area = πr^2 r = radius	 Equilateral Triangle Area = $\frac{\sqrt{3}}{4}s^2$ s = length of a side
 Square Area = a^2 a = length of side	 Rectangle Area = $b \times h$ b = breadth h = height

Table 3 – Area Formula of the Shapes

For each shape, the solution and formula for obtaining the Allowable Percentage Error of the Calculated Area are presented as Equation 1 to Equation 4.

$$\begin{aligned}
&\text{Circle}\\
&\text{Area} = \pi r^2\\
&\text{Allowable Percentage Error of Calculated Area :}\\
&\frac{\Delta A}{A} = \frac{dA}{A} = \frac{2\pi r\,dr}{\pi r^2} = \frac{2dr}{r}\\
&\frac{dA}{A}\left(\frac{100}{100}\right) = \left(\frac{2dr}{r}\cdot 100\right)\%
\end{aligned}$$

Equation 1 – Allowable Percentage Error for Circle

$$\begin{aligned}
&\text{Square}\\
&\text{Area} = s^2\\
&\text{Allowable Percentage Error of Calculated Area :}\\
&\frac{\Delta A}{A} = \frac{dA}{A} = \frac{2s\,ds}{s^2} = \frac{2ds}{s}\\
&\frac{dA}{A}\left(\frac{100}{100}\right) = \left(\frac{2ds}{s}\cdot 100\right)\%
\end{aligned}$$

Equation 2 – Allowable Percentage Error for Square

$$\begin{aligned}
&\text{Triangle (Equal Sides)}\\
&\text{Area} = \frac{\sqrt{3}}{4}s^2\\
&\text{Allowable Percentage Error of Calculated Area :}\\
&\frac{\Delta A}{A} = \frac{dA}{A} = \frac{\frac{\sqrt{3}}{4}s^2\,ds}{\frac{\sqrt{3}}{4}s^2} = \frac{\frac{\sqrt{3}}{2}s\,ds}{\frac{\sqrt{3}}{4}s^2} = \frac{2ds}{s}\\
&\frac{dA}{A}\left(\frac{100}{100}\right) = \left(\frac{2ds}{s}\cdot 100\right)\%
\end{aligned}$$

Equation 3 – Allowable Percentage Error for Equilateral Triangle

$$\begin{aligned}
&\text{Rectangle}\\
&\text{Area} = l \times w\\
&w = x\\
&l = x + 7.6\\
&A = x(x + 7.6)\\
&\text{Allowable Percentage Error of Calculated Area :}\\
&\frac{\Delta A}{A} = \frac{dA}{A} = \frac{x^2 + 7.6x\;dx}{x^2 + 7.6x} = \frac{(2x + 7.6)\;dx}{x^2 + 7.6x}\\
&\frac{dA}{A}\left(\frac{100}{100}\right) = \left(\frac{(2x + 7.6)\;dx}{x^2 + 7.6x}\cdot 100\right)\%
\end{aligned}$$

Equation 4 – Allowable Percentage Error for Rectangle

The results of the trials are tabulated below:

Small Circle			
Trial	**Ideal Radius**	**Measured Radius**	**Absolute Error**
1	**10.1**	**10.1**	**0.00**
2	**10.1**	**10.1**	**0.00**
3	**10.1**	**10.1**	**0.00**
4	**10.1**	**10.1**	**0.00**
5	**10.1**	**10.1**	**0.00**
		Average Absolute Error	**0.00**
		Allowable Percentage Error of Calculated Area	**0.00**

Table 4A –Tabulated Results of Allowable Percentage for Small Circle

Refer to Table 4A shows the results of the trials for the small circle. The allowable percentage error of the calculated area would be the maximum allowable percentage error. In the use of this formula (see Equation 1), the absolute error produced by the Drawbot would be related to its calculated area. The percentage error of the calculated showed the approximation of errors in measuring the dimensions since the absolute error was taken into consideration.

To interpret the data, the percentage error represents the percentage it would take for the area to be derailed from the ideal measurement. In the case of drawing circles, it had no errors. The Drawbot perfectly draw the circle. The radius could not be diverted to the ideal measurement because the drawing would not look like a circle if it would be so. Having no errors, the allowable percentage error yield 0%.

Medium Circle			
Trial	**Ideal Radius**	**Measured Radius**	**Absolute Error**
1	**11.4**	**11.4**	**0.00**
2	**11.4**	**11.4**	**0.00**
3	**11.4**	**11.4**	**0.00**

4	**11.4**	**11.4**	**0.00**
5	**11.4**	**11.4**	**0.00**
		Average Absolute Error	**0.00**
		Allowable Percentage Error of Calculated Area	**0.00**

Table 4B –Tabulated Results of Allowable Percentage for Medium Circle

Table 4B shows the results of the trials for the medium circle. The allowable percentage error of the calculated area would be the maximum allowable percentage error. In the use of this formula (see Equation 1), the absolute error produced by the Drawbot would be related to its calculated area. The percentage error of the calculated showed the approximation of errors in measuring the dimensions since the absolute error was taken into consideration.

To interpret the data, the percentage error represents the percentage it would take for the area to be derailed from the ideal measurement. In the case of drawing circles, it had no errors. The Drawbot perfectly draw the circle. The radius could not be diverted to the ideal measurement because the drawing will not look like a circle if it would be so. Having no errors, the allowable percentage error yield 0%.

Large Circle			
Trial	**Ideal Radius**	**Measured Radius**	**Absolute Error**
1	**12.7**	**12.7**	**0.00**
2	**12.7**	**12.7**	**0.00**
3	**12.7**	**12.7**	**0.00**
4	**12.7**	**12.7**	**0.00**
5	**12.7**	**12.7**	**0.00**
		Average Absolute Error	**0.00**
		Allowable Percentage Error of Calculated Area	**0.00**

Table 4C –Tabulated Results of Allowable Percentage for Large Circle

Table 4C shows the results of the trials for the large circle. The allowable percentage error of the calculated area would be the maximum allowable percentage error. In the use of this formula (see Equation 1), the absolute error produced by the Drawbot would be related to its calculated area. The percentage error of the calculated showed the approximation of errors in measuring the dimensions since the absolute error was taken into consideration.

To interpret the data, the percentage error represents the percentage it would take for the area to be derailed from the ideal measurement. In the case of drawing circles, it had no errors. The Drawbot perfectly draw the circle. The radius could not be diverted to the ideal measurement because the drawing will not look like a circle if it would be so. Having no errors, the allowable percentage error yield 0%.

Small Triangle			
Trial	**Ideal Side**	**Measured Side**	**Absolute Error**
1	**16.5**	**17.2**	**0.70**
2	**16.5**	**17.0**	**0.50**
3	**16.5**	**16.9**	**0.40**
4	**16.5**	**16.9**	**0.40**
5	**16.5**	**16.9**	**0.40**
Average Absolute Error			**0.48**
Allowable Percentage Error of Calculated Area			**5.82%**

Table 5A –Tabulated Results of Allowable Percentage for Small Triangle

Table 5A shows the results of the trials for the small triangle. The allowable percentage error of the calculated area would be the maximum allowable percentage error. In the use of this formula (see Equation 2), the absolute error produced by the

Drawbot would be related to its calculated area. The percentage error of the calculated showed the approximation of errors in measuring the dimensions since the absolute error was taken into consideration.

To interpret the data, the percentage error represents the percentage it would take for the area to be derailed from the ideal measurement. With the small triangle, the inaccuracy in the placement of the pen and how it affected the Drawbot's rotation on it made the percentage errors. These were already mentioned on the scope and delimitation section of this study. The gap made by the pen when the Drawbot rotated would be considered the absolute error of the measurement. As seen in the tabulated results, it was only the triangle, rectangle and square that was with errors. These proved the effect of the pen. The allowable percentage error of calculated area yielded 5.82%.

Medium Triangle			
Trial	**Ideal Side**	**Measured Side**	**Absolute Error**
1	**24.1**	**24.6**	**0.50**
2	**24.1**	**24.6**	**0.50**
3	**24.1**	**24.5**	**0.40**
4	**24.1**	**24.7**	**0.60**
5	**24.1**	**24.7**	**0.60**
		Average Absolute Error	**0.52**
		Allowable Percentage Error of Calculated Area	**4.32%**

Table 5B –Tabulated Results of Allowable Percentage for Medium Triangle

Table 5B shows the results of the trials for the medium triangle. The allowable percentage error of the calculated area would be the <u>maximum</u> allowable percentage

error. In the use of this formula (see Equation 2), the absolute error produced by the Drawbot would be related to its calculated area. The percentage error of the calculated showed the approximation of errors in measuring the dimensions since the absolute error was taken into consideration.

To interpret the data, the percentage error represents the percentage it would take for the area to be derailed from the ideal measurement. With the small triangle, the inaccuracy in the placement of the pen and how it affected the Drawbot's rotation on it made the percentage errors. These were already mentioned on the scope and delimitation section of this study. The gap made by the pen when the Drawbot rotated would be considered the absolute error of the measurement. As seen in the tabulated results, it was only the triangle, rectangle and square that was with errors. These proved the effect of the pen. The allowable percentage error of calculated area yielded 4.32%.

Large Triangle			
Trial	**Ideal Side**	**Measured Side**	**Absolute Error**
1	**40.6**	**41.3**	**0.70**
2	**40.6**	**41.3**	**0.70**
3	**40.6**	**41.2**	**0.60**
4	**40.6**	**41.2**	**0.60**
5	**40.6**	**41.2**	**0.60**
Average Absolute Error			**0.64**
Allowable Percentage Error of Calculated Area			**3.15%**

Table 5C –Tabulated Results of Allowable Percentage for Large Triangle

Table 5C shows for the results of the trials for the large triangle. The allowable percentage error of the calculated area would be the maximum allowable percentage

error. In the use of this formula (see Equation 2), the absolute error produced by the Drawbot would be related to its calculated area. The percentage error of the calculated showed the approximation of errors in measuring the dimensions since the absolute error was taken into consideration.

To interpret the data, the percentage error represents the percentage it would take for the area to be derailed from the ideal measurement. With the small triangle, the inaccuracy in the placement of the pen and how it affected the Drawbot's rotation on it made the percentage errors. These were already mentioned on the scope and delimitation section of this study. The gap made by the pen when the Drawbot rotated would be considered the absolute error of the measurement. As seen in the tabulated results, it was only the triangle, rectangle and square that was with errors. These proved the effect of the pen. The allowable percentage error of calculated area yielded 3.15%.

Small Square			
Trial	**Ideal Side**	**Measured Side**	**Absolute Error**
1	**16.5**	**16.9**	**0.40**
2	**16.5**	**17.1**	**0.60**
3	**16.5**	**17.1**	**0.60**
4	**16.5**	**16.9**	**0.40**
5	**16.5**	**16.9**	**0.40**
		Average Absolute Error	**0.48**
		Allowable Percentage Error of Calculated Area	**5.82%**

Table 6A –Tabulated Results of Allowable Percentage for Small Square

Table 6A shows the results of the trials for the small square. The allowable percentage error of the calculated area would be the <u>maximum</u> allowable percentage

error. In the use of this formula (see Equation 3), the absolute error produced by the Drawbot would be related to its calculated area. The percentage error of the calculated showed the approximation of errors in measuring the dimensions since the absolute error was taken into consideration.

To interpret the data, the percentage error represents the percentage it would take for the area to be derailed from the ideal measurement. With the small triangle, the inaccuracy in the placement of the pen and how it affected the Drawbot's rotation on it made the percentage errors. These were already mentioned on the scope and delimitation section of this study. The gap made by the pen when the Drawbot rotated would be considered the absolute error of the measurement. As seen in the tabulated results, it was only the triangle, rectangle and square that was with errors. These proved the effect of the pen. The allowable percentage error of calculated area yielded 5.82%

Medium Square			
Trial	**Ideal Side**	**Measured Side**	**Absolute Error**
1	**24.1**	**24.7**	**0.60**
2	**24.1**	**24.7**	**0.60**
3	**24.1**	**24.8**	**0.70**
4	**24.1**	**24.6**	**0.50**
5	**24.1**	**24.6**	**0.50**
		Average Absolute Error	**0.58**
		Allowable Percentage Error of Calculated Area	**4.81%**

Table 6B –Tabulated Results of Allowable Percentage for Medium Square

Table 6B shows the results of the trials for the medium square. The allowable percentage error of the calculated area would be the <u>maximum</u> allowable percentage

error. In the use of this formula (see Equation 3), the absolute error produced by the Drawbot would be related to its calculated area. The percentage error of the calculated showed the approximation of errors in measuring the dimensions since the absolute error was taken into consideration.

To interpret the data, the percentage error represents the percentage it would take for the area to be derailed from the ideal measurement. With the small triangle, the inaccuracy in the placement of the pen and how it affected the Drawbot's rotation on it made the percentage errors. These were already mentioned on the scope and delimitation section of this study. The gap made by the pen when the Drawbot rotated would be considered the absolute error of the measurement. As seen in the tabulated results, it was only the triangle, rectangle and square that was with errors. These proved the effect of the pen. The allowable percentage error of calculated area yielded 4.81%

Large Square			
Trial	**Ideal Side**	**Measured Side**	**Absolute Error**
1	**40.6**	**41.3**	**0.70**
2	**40.6**	**41.3**	**0.70**
3	**40.6**	**41.2**	**0.60**
4	**40.6**	**41.2**	**0.60**
5	**40.6**	**41.3**	**0.70**
		Average Absolute Error	**0.66**
		Allowable Percentage Error of Calculated Area	**3.25%**

Table 6C –Tabulated Results of Allowable Percentage for Large Square

Table 6C shows the results of the trials for the large square. The allowable percentage error of the calculated area would be the <u>maximum</u> allowable percentage

error. In the use of this formula (see Equation 3), the absolute error produced by the Drawbot would be related to its calculated area. The percentage error of the calculated showed the approximation of errors in measuring the dimensions since the absolute error was taken into consideration.

To interpret the data, the percentage error represents the percentage it would take for the area to be derailed from the ideal measurement. With the small triangle, the inaccuracy in the placement of the pen and how it affected the Drawbot's rotation on it made the percentage errors. These were already mentioned on the scope and delimitation section of this study. The gap made by the pen when the Drawbot rotated would be considered the absolute error of the measurement. As seen in the tabulated results, it was only the triangle, rectangle and square that was with errors. These proved the effect of the pen. The allowable percentage error of calculated area yielded 3.25%

Small Rectangle			
Trial	**Ideal Width**	**Measured Width**	**Absolute Error**
1	**16.6**	**17.1**	**0.50**
2	**16.6**	**17.1**	**0.50**
3	**16.6**	**17.2**	**0.60**
4	**16.6**	**17.1**	**0.50**
5	**16.6**	**17.2**	**0.60**
		Average Absolute Error	**0.54**
		Allowable Percentage Error of Calculated Area	**5.48%**

Table 7A –Tabulated Results of Allowable Percentage for Small Rectangle

Table 7A shows the results of the trials for the small rectangle. The allowable percentage error of the calculated area would be the <u>maximum</u> allowable percentage error. In the use of this formula (see Equation 4), the absolute error produced by the

Drawbot would be related to its calculated area. The percentage error of the calculated showed the approximation of errors in measuring the dimensions since the absolute error was taken into consideration.

To interpret the data, the percentage error represents the percentage it would take for the area to be derailed from the ideal measurement. With the small triangle, the inaccuracy in the placement of the pen and how it affected the Drawbot's rotation on it made the percentage errors. These were already mentioned on the scope and delimitation section of this study. The gap made by the pen when the Drawbot rotated would be considered the absolute error of the measurement. As seen in the tabulated results, it was only the triangle, rectangle and square that was with errors. These proved the effect of the pen. The allowable percentage error of calculated area yielded 5.48%

Medium Rectangle			
Trial	**Ideal Width**	**Measured Width**	**Absolute Error**
1	**22.9**	**23.6**	**0.70**
2	**22.9**	**23.6**	**0.70**
3	**22.9**	**23.5**	**0.60**
4	**22.9**	**23.6**	**0.70**
5	**22.9**	**23.7**	**0.80**
		Average Absolute Error	**0.70**
		Allowable Percentage Error of Calculated Area	**5.35%**

Table 7B –Tabulated Results of Allowable Percentage for Medium Rectangle

Table 7B shows the results of the trials for the medium rectangle. The allowable percentage error of the calculated area would be the maximum allowable percentage error. In the use of this formula (see Equation 4), the absolute error produced by the Drawbot would be related to its calculated area. The percentage error of the calculated showed the approximation of errors in measuring the dimensions since the absolute error was taken into consideration.

To interpret the data, the percentage error represents the percentage it would take for the area to be derailed from the ideal measurement. With the small triangle, the inaccuracy in the placement of the pen and how it affected the Drawbot's rotation on it made the percentage errors. These were already mentioned on the scope and delimitation section of this study. The gap made by the pen when the Drawbot rotated would be considered the absolute error of the measurement. As seen in the tabulated results, it was only the triangle, rectangle and square that was with errors. These proved the effect of the pen. The allowable percentage error of calculated area yielded 5.35%

Large Rectangle			
Trial	**Ideal Width**	**Measured Width**	**Absolute Error**
1	**31.8**	**32.5**	**0.70**
2	**31.8**	**32.5**	**0.70**
3	**31.8**	**32.6**	**0.80**
4	**31.8**	**32.5**	**0.70**
5	**31.8**	**32.6**	**0.80**
		Average Absolute Error	**0.74**
		Allowable Percentage Error of Calculated Area	**4.21%**

Table 7C –Tabulated Results of Allowable Percentage for Large Rectangle

Table 7C shows the results of the trials for the large rectangle. The allowable percentage error of the calculated area would be the maximum allowable percentage error. In the use of this formula (see Equation 4), the absolute error produced by the Drawbot would be related to its calculated area. The percentage error of the calculated showed the approximation of errors in measuring the dimensions since the absolute error was taken into consideration.

To interpret the data, the percentage error represents the percentage it would take for the area to be derailed from the ideal measurement. With the small triangle, the inaccuracy in the placement of the pen and how it affected the Drawbot's rotation on it made the percentage errors. These were already mentioned on the scope and delimitation section of this study. The gap made by the pen when the Drawbot rotated would be considered the absolute error of the measurement. As seen in the tabulated results, it was only the triangle, rectangle and square that was with errors. These proved the effect of the pen. The allowable percentage error of calculated area yielded 4.21%.

SUMMARY RESULTS OF THE ALLOWABLE PERCENTAGE ERRORS			
Shape	**Size**	**Absolute Error, *dx***	**Allowable Percentage Error of Calculated Area**
Circle (radius)	**Small**	**0.00**	**0.00%**
	Medium	**0.00**	**0.00%**
	Large	**0.00**	**0.00%**
Equilateral Triangle (one side)	**Small**	**0.48**	**5.82%**
	Medium	**0.52**	**4.32%**
	Large	**0.64**	**3.15%**
Square (one side)	**Small**	**0.48**	**5.82%**
	Medium	**0.58**	**4.81%**
	Large	**0.66**	**3.25%**
Rectangle (length x width)	**Small**	**0.54**	**5.48%**
	Medium	**0.70**	**5.35%**
	Large	**0.74**	**4.21%**

Table 8 –Summary Results of Allowable Percentage Errors of Calculated Area

Table 8 presents the summary results of the allowable percentage error of the calculated area. As shown in the previous tables, this would be the <u>maximum</u> allowable percentage error. In the use of this formula, the absolute error produced by the Drawbot would be related to its calculated area. The percentage error of the calculated showed the approximation of errors in measuring the dimensions since the absolute error was taken into consideration.

To interpret the data, the percentage error represents the percentage it would take for the area to be derailed from the ideal measurement. The trials showed that the circle had no errors. The allowable percentage error was 0%. With the other shapes, the inaccuracy in the placement of the pen and how it affected the Drawbot's rotation on the shapes made the percentage errors. These were already mentioned on the study's scope and delimitation. The gap made by the pen when the Drawbot rotated was considered the absolute error of the measurement. As seen in the tabulated result, only the triangle, rectangle and square had the errors. These proved the effect of the pen. The range of the allowable percentage error reached almost up to 6% with 5.82% as the maximum. It was observed however that as the shape gets bigger in size, absolute error increased and the allowable percentage error decreased. This was due to the fact that the bigger the size to be drawn, the higher the occurrence of errors in measurement to occur. It was then to be concluded that the Drawbot's accuracy to draw the shape was within the allowable range. With its estimated 6% allowable percentage error, the shape drawing was almost 90% correct.

Chapter 4

CONCLUSION AND RECOMMENDATION

Conclusion

The researchers were able to meet the required objective to develop the prototype. The program of the microcontroller can receive data and control the stepper motors to perform specific actions. The PIC controls the robot in a way that it moves in a programmed path similar to drawing the shapes. The system is able to perform its major function which is to draw shapes and the size selected for each shape. Input from the user is obtained and transmitted to the microcontroller of the Drawbot. The Drawbot then processes the input and draws the output depending on the size and shape entered on the push buttons. With the use of the design, users can be able to draw the shapes accurately without effort and mistakes. The design can help people in drawing because they no longer need to bring measuring tools to determine the exact dimensions of the shape. The design also draws the shapes in a single sweep which a person cannot normally do and which may lead to inaccuracy of the drawing. Using the design could make drawing faster thus save a great amount of time on the part of the user since it does the work without stopping until it reaches the end of its course.

Recommendation

The design can be further improve by implementing the following:

1. Addition of more shapes that the Drawbot can draw to increase its function.
2. The design works properly only on flat surfaces so if it is used in rough or uneven surfaces, problems in the drawing will occur. It is therefore recommended that the pen holder be improved so that it can draw even on rough surfaces. The prototype should be able to add that specific feature through modifying the screw that holds the pen in place.
3. Improvements on its motion are also recommended so that it can work on different types of surfaces.

4. The Design Modification can be made so that it can be used for large scale projects such as road works and floor painting. Such design may greatly decrease the time and effort needed in doing these types of projects.
5. A better power supply is also advisable so that it can operate for a longer period of time.

BIBLIOGRAPHY

Floyd, Thomas L. (2007), *Electronic Devices (Conventional Current Version),* 8th Edition. Prentice Hall, Colorado.

Sison, Luis (2000). *Learning Microcontrollers thru Experiments.* Alexan Foundation, Metro Manila.

Casey Reas, Ben Fry (2007) *Processing: A Programming Handbook for Visual Designers and Artists*

Cook, David (2002) *Robot Building for Beginners*

Irwin, J. David (1997) *The Industrial Electronics Handbook*

Gottlieb, Irving M. (1994) *Electric Motors and Control Techniques*

APPENDICES

APPENDIX A

Main Program Source Code

```
Device = 16F877
Declare XTAL 4

LCD_DTPIN = PORTC.0
LCD_RSPIN = PORTC.4
LCD_ENPIN = PORTC.5
LCD_INTERFACE = 4
LCD_LINES = 4
LCD_TYPE = 0
ALL_DIGITAL = TRUE


Dim i As Byte

 TRISA = %00000
 TRISC = %00000000
 TRISB = %00000000
 TRISD = %00000000
 TRISE = %111
```

```
 DelayMS 500

 PORTA=0
 PORTB=0
 PORTC=0
 PORTD=0
 PORTE=0
 i = 1


display:
    Cls
    Print At 1, 1,  "     M A P U A      "
    Print At 2, 1,  "School of EE-ECE-CoE"
    Print At 3, 1,  " (c) copyright 2008 "
    Print At 4, 1,  "   The Wizard's     "
    DelayMS 2000

disp2:
    Cls
    Print At 1, 1,  "    Chua, Daniel    "
    Print At 2, 1,  " Latigay, Ryan Henry"
    Print At 3, 1,  " Panganiban, Toni R."
    Print At 4, 1,  "     Sy, Virnali    "

disp3:
        If PORTE.2 = 1 Then
        DelayMS 250
        GoTo main
        EndIf

    GoTo disp3

main:
        Print At 1, 1, "      DRAW BOT      "
        Print At 2, 1, " [<] Small CIRCLE   "
        Print At 3, 1, " [>] Small SQUARE   "
        Print At 4, 1, " [E] NEXT           "
select1:
        If PORTE.0 = 1 Then
        DelayMS 250
        GoTo cir_sml
        EndIf

        If PORTE.1 = 1 Then
        DelayMS 250
        GoTo sqr_sml
        EndIf

        If PORTE.2 = 1 Then
        DelayMS 250
        GoTo next_s
        EndIf

        GoTo select1

next_s:
        Print At 1, 1, "      DRAW BOT      "
```

```
        Print At 2, 1, " [<] Small TRIANGLE "
        Print At 3, 1, " [>] Small RECTANGLE"
        Print At 4, 1, " [E] NEXT           "

next_s1:
        If PORTE.0 = 1 Then
        DelayMS 250
        GoTo tri_sml
        EndIf

        If PORTE.1 = 1 Then
        DelayMS 250
        GoTo rec_sml
        EndIf

        If PORTE.2 = 1 Then
        DelayMS 250
        GoTo menu_med
        EndIf

        GoTo next_s1



menu_med:
        Print At 1, 1, "      DRAW BOT      "
        Print At 2, 1, " [<] medium CIRCLE  "
        Print At 3, 1, " [>] medium SQUARE  "
        Print At 4, 1, " [E] NEXT           "
select2:
        If PORTE.0 = 1 Then
        DelayMS 250
        GoTo cir_med
        EndIf

        If PORTE.1 = 1 Then
        DelayMS 250
        GoTo sqr_med
        EndIf

        If PORTE.2 = 1 Then
        DelayMS 250
        GoTo next_menu
        EndIf

        GoTo select2

next_menu:
        Print At 1, 1, "      DRAW BOT      "
        Print At 2, 1, " [<]medium TRIANGLE "
        Print At 3, 1, " [>]medium RECTANGLE"
        Print At 4, 1, " [E] NEXT           "

next_s2:
        If PORTE.0 = 1 Then
        DelayMS 250
        GoTo tri_med
        EndIf
```

```
        If PORTE.1 = 1 Then
        DelayMS 250
        GoTo rec_med
        EndIf

        If PORTE.2 = 1 Then
        DelayMS 250
        GoTo menu_lrg
        EndIf

        GoTo next_s2

menu_lrg:
        Print At 1, 1, "      DRAW BOT      "
        Print At 2, 1, " [<] large CIRCLE   "
        Print At 3, 1, " [>] large SQUARE   "
        Print At 4, 1, " [E] NEXT           "
select3:
        If PORTE.0 = 1 Then
        DelayMS 250
        GoTo cir_med
        EndIf

        If PORTE.1 = 1 Then
        DelayMS 250
        GoTo sqr_med
        EndIf

        If PORTE.2 = 1 Then
        DelayMS 250
        GoTo next_menus
        EndIf

        GoTo select3

next_menus:
        Print At 1, 1, "      DRAW BOT      "
        Print At 2, 1, " [<] large TRIANGLE "
        Print At 3, 1, " [>] large RECTANGLE"
        Print At 4, 1, " [E] NEXT           "

next_s3:
        If PORTE.0 = 1 Then
        DelayMS 250
        GoTo tri_lrg
        EndIf

        If PORTE.1 = 1 Then
        DelayMS 250
        GoTo rec_lrg
        EndIf

        If PORTE.2 = 1 Then
        DelayMS 250
        GoTo display    'main
        EndIf
```

```
        GoTo next_s3

cir_sml:

    High PORTD.5
    Low  PORTD.6

    Print At 1, 1, "      DRAW BOT      "
    Print At 2, 1, "    Please Wait....."
    Print At 3, 1, " Drawing the small  "
    Print At 4, 1, "    circle now      "

    GoSub cir_loop
    GoSub cir_loop
    GoSub cir_loop
    GoSub cir_loop
    GoSub cir_loop
    GoSub cir_loop
    GoSub cir_loop

    GoSub cir_loop
    GoSub cir_loop
    GoSub cir_loop
    GoSub cir_loop
    GoSub cir_loop
    GoSub cir_loop
    GoSub cir_loop

    GoSub cir_loop
    GoSub cir_loop
    Cls
    GoTo main

cir_loop:
    High PORTD.4
    High PORTD.0
    For i = 0 To 300
    PORTD.1=i
    DelayMS 1
    Next i
    Low PORTD.0
    Low PORTD.1
    Low PORTD.4
    Return


cir_med:

    Low PORTD.5
    High PORTD.6

    If PORTD.7 = 1 Then
    Low PORTD.5
    Low PORTD.6
    GoTo next1
    EndIf

    GoTo cir_med
```

```
next1:
    Print At 1, 1, "      DRAW BOT       "
    Print At 2, 1, "    Please Wait....."
    Print At 3, 1, " Drawing the medium "
    Print At 4, 1, "    circle now       "

    GoSub cir_loop_med
    GoSub cir_loop_med
    GoSub cir_loop_med
    GoSub cir_loop_med
    GoSub cir_loop_med
    GoSub cir_loop_med
    GoSub cir_loop_med

    GoSub cir_loop_med
    GoSub cir_loop_med
    GoSub cir_loop_med
    GoSub cir_loop_med
    GoSub cir_loop_med
    GoSub cir_loop_med
    GoSub cir_loop_med

    GoSub cir_loop_med
    GoSub cir_loop_med

    High  PORTD.5
    Low PORTD.6
    DelayMS 1000
    Cls
    GoTo main

cir_loop_med:
    High PORTD.4
    High PORTD.0
    For i = 0 To 300
    PORTD.1=i
    DelayMS 1
    Next i
    Low PORTD.0
    Low PORTD.1
    Low PORTD.4
    Return

cir_lrg:

    Low PORTD.5
    High PORTD.6

    Print At 1, 1, "      DRAW BOT       "
    Print At 2, 1, "    Please Wait....."
    Print At 3, 1, " Drawing the large   "
    Print At 4, 1, "    circle now       "

    GoSub cir_loop_lrg
    GoSub cir_loop_lrg
    GoSub cir_loop_lrg
    GoSub cir_loop_lrg
```

```
    GoSub cir_loop_lrg
    GoSub cir_loop_lrg
    GoSub cir_loop_lrg

    GoSub cir_loop_lrg
    GoSub cir_loop_lrg
    GoSub cir_loop_lrg
    GoSub cir_loop_lrg
    GoSub cir_loop_lrg
    GoSub cir_loop_lrg
    GoSub cir_loop_lrg

    GoSub cir_loop_lrg
    GoSub cir_loop_lrg

    High  PORTD.5
    Low PORTD.6
    DelayMS 1000
    Cls
    GoTo main

cir_loop_lrg:
    High PORTD.4
    High PORTD.0
    For i = 0 To 300
    PORTD.1=i
    DelayMS 1
    Next i
    Low PORTD.0
    Low PORTD.1
    Low PORTD.4
    Return



sqr_sml:
    Print At 1, 1, "      DRAW BOT      "
    Print At 2, 1, "    Please Wait....."
    Print At 3, 1, " Drawing the small  "
    Print At 4, 1, "    square now      "


    GoSub sqr_line
    GoSub sqr_line
    DelayMS 1000

    GoSub sqr_90loop
    GoSub sqr_90loop
    DelayMS 1000
'-------------------
    GoSub sqr_line
    GoSub sqr_line
    DelayMS 1000

    GoSub sqr_90loop
    GoSub sqr_90loop
    DelayMS 1000
'-------------------
```

```
    GoSub sqr_line2
    GoSub sqr_line2
    DelayMS 1000

    GoSub sqr_90loop
    GoSub sqr_90loop
    DelayMS 1000
'--------------------
    GoSub sqr_line
    GoSub sqr_line
    DelayMS 1000
    GoTo main

sqr_line:
    High PORTD.4
    High PORTD.0
    High PORTD.2
    For i = 0 To 240
    PORTD.1=i
    PORTD.3=i
    DelayMS 5
    Next i
    Low PORTD.0
    Low PORTD.1
    Low PORTD.2
    Low PORTD.3
    Low PORTD.4
    Return

sqr_line2:
    High PORTD.4
    High PORTD.0
    High PORTD.2
    For i = 0 To 230
    PORTD.1=i
    PORTD.3=i
    DelayMS 5
    Next i
    Low PORTD.0
    Low PORTD.1
    Low PORTD.2
    Low PORTD.3
    Low PORTD.4
    Return


sqr_90loop:
    High PORTD.4
    High PORTD.0
    Low PORTD.2
    For i = 0 To 240
    PORTD.1=i
    PORTD.3=i
    DelayMS 5
    Next i
    Low PORTD.0
    Low PORTD.1
    Low PORTD.2
```

```
    Low PORTD.3
    Low PORTD.4
    Return


sqr_med:
    Print At 1, 1, "      DRAW BOT      "
    Print At 2, 1, "    Please Wait....."
    Print At 3, 1, " Drawing the medium "
    Print At 4, 1, "    square now      "


    GoSub med_sqr_line
    GoSub med_sqr_line
    GoSub med_sqr_line
    DelayMS 1000

    GoSub med_sqr_90loop
    GoSub med_sqr_90loop
    DelayMS 1000
'--------------------
    GoSub med_sqr_line
    GoSub med_sqr_line
    GoSub med_sqr_line
    DelayMS 1000

    GoSub med_sqr_90loop
    GoSub med_sqr_90loop
    DelayMS 1000
'--------------------
    GoSub med_sqr_line
    GoSub med_sqr_line
    GoSub med_sqr_line

    GoSub med_sqr_90loop
    GoSub med_sqr_90loop
    DelayMS 1000
'--------------------
    GoSub med_sqr_line
    GoSub med_sqr_line
    GoSub med_sqr_line
    DelayMS 1000
    GoTo main

med_sqr_line:
    High PORTD.4
    High PORTD.0
    High PORTD.2
    For i = 0 To 240
    PORTD.1=i
    PORTD.3=i
    DelayMS 5
    Next i
    Low PORTD.0
    Low PORTD.1
    Low PORTD.2
    Low PORTD.3
    Low PORTD.4
```

```
    Return

med_sqr_90loop:
    High PORTD.4
    High PORTD.0
    Low PORTD.2
    For i = 0 To 240
    PORTD.1=i
    PORTD.3=i
    DelayMS 5
    Next i
    Low PORTD.0
    Low PORTD.1
    Low PORTD.2
    Low PORTD.3
    Low PORTD.4
    Return

lrg_sqr:
    Print At 1, 1, "      DRAW BOT      "
    Print At 2, 1, "    Please Wait....."
    Print At 3, 1, " Drawing the large  "
    Print At 4, 1, "    square now      "


    GoSub lrg_sqr_line
    GoSub lrg_sqr_line
    GoSub lrg_sqr_line
    GoSub lrg_sqr_line
    DelayMS 1000

    GoSub lrg_sqr_90loop
    GoSub lrg_sqr_90loop
    DelayMS 1000
'--------------------
    GoSub lrg_sqr_line
    GoSub lrg_sqr_line
    GoSub lrg_sqr_line
    GoSub lrg_sqr_line
    DelayMS 1000

    GoSub lrg_sqr_90loop
    GoSub lrg_sqr_90loop
    DelayMS 1000
'--------------------
    GoSub lrg_sqr_line
    GoSub lrg_sqr_line
    GoSub lrg_sqr_line
    GoSub lrg_sqr_line

    GoSub lrg_sqr_90loop
    GoSub lrg_sqr_90loop
    DelayMS 1000
'--------------------
    GoSub lrg_sqr_line
    GoSub lrg_sqr_line
    GoSub lrg_sqr_line
```

```
    GoSub lrg_sqr_line
    DelayMS 1000
    GoTo main

lrg_sqr_line:
    High PORTD.4
    High PORTD.0
    High PORTD.2
    For i = 0 To 240
    PORTD.1=i
    PORTD.3=i
    DelayMS 5
    Next i
    Low PORTD.0
    Low PORTD.1
    Low PORTD.2
    Low PORTD.3
    Low PORTD.4
    Return


lrg_sqr_90loop:
    High PORTD.4
    High PORTD.0
    Low PORTD.2
    For i = 0 To 240
    PORTD.1=i
    PORTD.3=i
    DelayMS 5
    Next i
    Low PORTD.0
    Low PORTD.1
    Low PORTD.2
    Low PORTD.3
    Low PORTD.4
    Return



tri_sml:
    Print At 1, 1, "      DRAW BOT       "
    Print At 2, 1, "    Please Wait....."
    Print At 3, 1, " Drawing the small   "
    Print At 4, 1, "   triangle  now     "


    GoSub tri_line
    GoSub tri_line
    DelayMS 1000

    GoSub tri_deg_loop_sml
    GoSub tri_deg_loop_sml
    GoSub tri_deg_loop_sml
    GoSub tri_deg_loop_sml

    GoSub tri_line
    GoSub tri_line
    DelayMS 1000
```

```
    GoSub tri_deg_loop_sml
    GoSub tri_deg_loop_sml
    GoSub tri_deg_loop_sml
    GoSub tri_deg_loop_sml

    GoSub tri_line
    GoSub tri_line
    DelayMS 1000

    GoTo main


tri_line:
    High PORTD.4
    High PORTD.0
    High PORTD.2
    For i = 0 To 240
    PORTD.1=i
    PORTD.3=i
    DelayMS 5
    Next i
    Low PORTD.0
    Low PORTD.1
    Low PORTD.2
    Low PORTD.3
    Low PORTD.4
    Return

tri_deg_loop_sml:
    High PORTD.4
    High PORTD.0
    Low PORTD.2
    For i = 0 To 163
    PORTD.1=i
    PORTD.3=i
    DelayMS 5
    Next i
    Low PORTD.0
    Low PORTD.1
    Low PORTD.2
    Low PORTD.3
    Low PORTD.4

    Return


tri_med:
    Print At 1, 1, "      DRAW BOT       "
    Print At 2, 1, "    Please Wait....."
    Print At 3, 1, " Drawing the medium "
    Print At 4, 1, "   triangle  now     "


    GoSub tri_line_med
    GoSub tri_line_med
    GoSub tri_line_med
    DelayMS 1000
```

```
    GoSub tri_deg_loop_med
    GoSub tri_deg_loop_med
    GoSub tri_deg_loop_med
    GoSub tri_deg_loop_med

    GoSub tri_line_med
    GoSub tri_line_med
    GoSub tri_line_med
    DelayMS 1000

    GoSub tri_deg_loop_med
    GoSub tri_deg_loop_med
    GoSub tri_deg_loop_med
    GoSub tri_deg_loop_med

    GoSub tri_line_med
    GoSub tri_line_med
    GoSub tri_line_med
    DelayMS 1000

    GoTo main


tri_line_med:
    High PORTD.4
    High PORTD.0
    High PORTD.2
    For i = 0 To 240
    PORTD.1=i
    PORTD.3=i
    DelayMS 5
    Next i
    Low PORTD.0
    Low PORTD.1
    Low PORTD.2
    Low PORTD.3
    Low PORTD.4
    Return

tri_deg_loop_med:
    High PORTD.4
    High PORTD.0
    Low PORTD.2
    For i = 0 To 163
    PORTD.1=i
    PORTD.3=i
    DelayMS 5
    Next i
    Low PORTD.0
    Low PORTD.1
    Low PORTD.2
    Low PORTD.3
    Low PORTD.4

    Return
```

```
tri_lrg:
    Print At 1, 1, "      DRAW BOT      "
    Print At 2, 1, "    Please Wait....."
    Print At 3, 1, " Drawing the large  "
    Print At 4, 1, "   triangle  now    "


    GoSub tri_line_lrg
    GoSub tri_line_lrg
    GoSub tri_line_lrg
    GoSub tri_line_lrg
    DelayMS 1000

    GoSub tri_deg_loop_lrg
    GoSub tri_deg_loop_lrg
    GoSub tri_deg_loop_lrg
    GoSub tri_deg_loop_lrg


    GoSub tri_line_lrg
    GoSub tri_line_lrg
    GoSub tri_line_lrg
    GoSub tri_line_lrg
    DelayMS 1000

    GoSub tri_deg_loop_med
    GoSub tri_deg_loop_med
    GoSub tri_deg_loop_med
    GoSub tri_deg_loop_med

    GoSub tri_line_lrg
    GoSub tri_line_lrg
    GoSub tri_line_lrg
    GoSub tri_line_lrg
    DelayMS 1000

    GoTo main


tri_line_lrg:
    High PORTD.4
    High PORTD.0
    High PORTD.2
    For i = 0 To 240
    PORTD.1=i
    PORTD.3=i
    DelayMS 5
    Next i
    Low PORTD.0
    Low PORTD.1
    Low PORTD.2
    Low PORTD.3
    Low PORTD.4
    Return

tri_deg_loop_lrg:
    High PORTD.4
    High PORTD.0
```

```
    Low PORTD.2
    For i = 0 To 163
    PORTD.1=i
    PORTD.3=i
    DelayMS 5
    Next i
    Low PORTD.0
    Low PORTD.1
    Low PORTD.2
    Low PORTD.3
    Low PORTD.4

    Return



rec_sml:
    Print At 1, 1, "       DRAW BOT       "
    Print At 2, 1, "     Please Wait....."
    Print At 3, 1, " Drawing the small   "
    Print At 4, 1, "    rectangle now    "


    GoSub rec_line
    GoSub rec_line
    DelayMS 1000

    GoSub rec_90loop
    GoSub rec_90loop
    DelayMS 1000

    GoSub rec_line
    GoSub rec_line
    GoSub rec_line
    DelayMS 1000

    GoSub rec_90loop
    GoSub rec_90loop
    DelayMS 1000

    GoSub rec_line
    GoSub rec_line
    DelayMS 1000

    GoSub rec_90loop
    GoSub rec_90loop
    DelayMS 1000

    GoSub rec_line
    GoSub rec_line
    GoSub rec_line
    DelayMS 1000

    GoTo main

rec_line:
    High PORTD.4
    High PORTD.0
```

```
    High PORTD.2
    For i = 0 To 240
    PORTD.1=i
    PORTD.3=i
    DelayMS 5
    Next i
    Low PORTD.0
    Low PORTD.1
    Low PORTD.2
    Low PORTD.3
    Low PORTD.4
    Return


rec_90loop:
    High PORTD.4
    High PORTD.0
    Low PORTD.2
    For i = 0 To 240
    PORTD.1=i
    PORTD.3=i
    DelayMS 5
    Next i
    Low PORTD.0
    Low PORTD.1
    Low PORTD.2
    Low PORTD.3
    Low PORTD.4
    Return

rec_med:
    Print At 1, 1, "      DRAW BOT      "
    Print At 2, 1, "    Please Wait....."
    Print At 3, 1, " Drawing the medium "
    Print At 4, 1, "   rectangle now    "


    GoSub rec_line_med
    GoSub rec_line_med
    GoSub rec_line_med
    DelayMS 1000

    GoSub rec_90loop_med
    GoSub rec_90loop_med
    DelayMS 1000

    GoSub rec_line_med
    GoSub rec_line_med
    GoSub rec_line_med
    GoSub rec_line_med
    DelayMS 1000

    GoSub rec_90loop_med
    GoSub rec_90loop_med
    DelayMS 1000

    GoSub rec_line_med
    GoSub rec_line_med
```

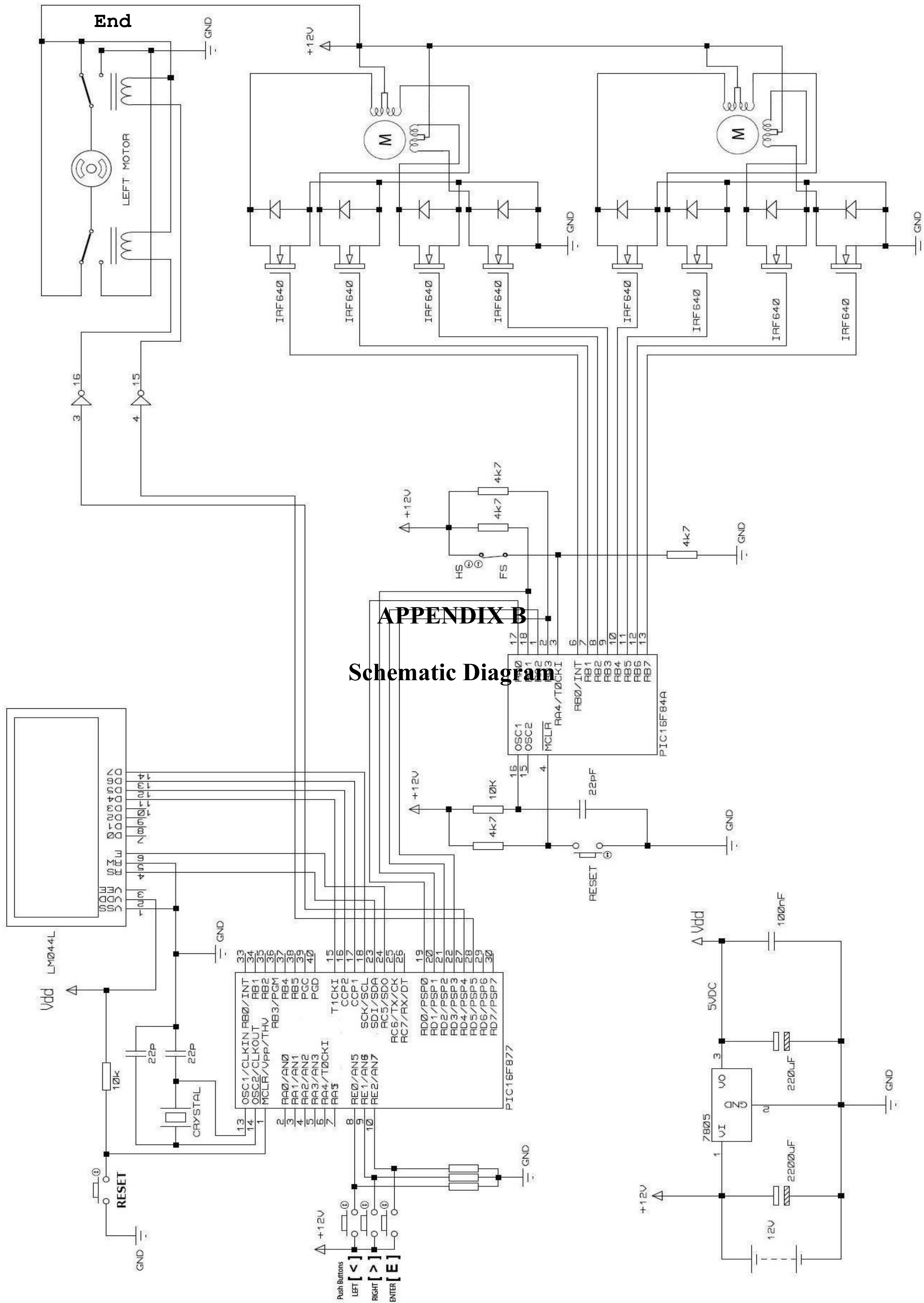
```
    GoSub rec_line_med
    DelayMS 1000

    GoSub rec_90loop_med
    GoSub rec_90loop_med
    DelayMS 1000

    GoSub rec_line_med
    GoSub rec_line_med
    GoSub rec_line_med
    GoSub rec_line_med
    DelayMS 1000

    GoTo main

rec_line_med:
    High PORTD.4
    High PORTD.0
    High PORTD.2
    For i = 0 To 240
    PORTD.1=i
    PORTD.3=i
    DelayMS 5
    Next i
    Low PORTD.0
    Low PORTD.1
    Low PORTD.2
    Low PORTD.3
    Low PORTD.4
    Return


rec_90loop_med:
    High PORTD.4
    High PORTD.0
    Low PORTD.2
    For i = 0 To 240
    PORTD.1=i
    PORTD.3=i
    DelayMS 5
    Next i
    Low PORTD.0
    Low PORTD.1
    Low PORTD.2
    Low PORTD.3
    Low PORTD.4
    Return
rec_lrg:
    Print At 1, 1, "      DRAW BOT      "
    Print At 2, 1, "    Please Wait....."
    Print At 3, 1, " Drawing the large  "
    Print At 4, 1, "   rectangle now    "


    GoSub rec_line_lrg
    GoSub rec_line_lrg
    GoSub rec_line_lrg
    GoSub rec_line_lrg
```

```
    DelayMS 1000

    GoSub rec_90loop_lrg
    GoSub rec_90loop_lrg
    DelayMS 1000

    GoSub rec_line_lrg
    GoSub rec_line_lrg
    GoSub rec_line_lrg
    DelayMS 1000

    GoSub rec_90loop_lrg
    GoSub rec_90loop_lrg
    DelayMS 1000

    GoSub rec_line_lrg
    GoSub rec_line_lrg
    GoSub rec_line_lrg
    GoSub rec_line_lrg
    DelayMS 1000

    GoSub rec_90loop_lrg
    GoSub rec_90loop_lrg
    DelayMS 1000

    GoSub rec_line_lrg
    GoSub rec_line_lrg
    GoSub rec_line_lrg
    DelayMS 1000

    GoTo main

rec_line_lrg:
    High PORTD.4
    High PORTD.0
    High PORTD.2
    For i = 0 To 240
    PORTD.1=i
    PORTD.3=i
    DelayMS 5
    Next i
    Low PORTD.0
    Low PORTD.1
    Low PORTD.2
    Low PORTD.3
    Low PORTD.4
    Return


rec_90loop_lrg:
    High PORTD.4
    High PORTD.0
    Low PORTD.2
    For i = 0 To 240
    PORTD.1=i
    PORTD.3=i
    DelayMS 5
    Next i
```

```
Low PORTD.0
Low PORTD.1
Low PORTD.2
Low PORTD.3
Low PORTD.4
Return

End
```

APPENDIX B

Schematic Diagram



APPENDIX C

PIC16F877 Microcontroller Data Sheet

MICROCHIP

PIC16F87X Data Sheet

28/40-Pin 8-Bit CMOS FLASH

Microcontrollers

PIC16F87X

28/40-Pin 8-Bit CMOS FLASH Microcontrollers

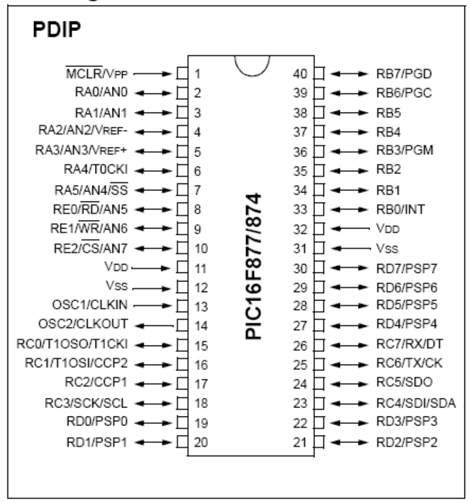
Devices Included in this Data Sheet:

- PIC16F873
- PIC16F874
- PIC16F876
- PIC16F877

Microcontroller Core Features:

- High performance RISC CPU
- Only 35 single word instructions to learn
- All single cycle instructions except for program branches which are two cycle
- Operating speed: DC - 20 MHz clock input
  DC - 200 ns instruction cycle
- Up to 8K x 14 words of FLASH Program Memory,
  Up to 368 x 8 bytes of Data Memory (RAM)
  Up to 256 x 8 bytes of EEPROM Data Memory
- Pinout compatible to the PIC16C73B/74B/76/77
- Interrupt capability (up to 14 sources)
- Eight level deep hardware stack
- Direct, indirect and relative addressing modes
- Power-on Reset (POR)
- Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator for reliable operation
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options
- Low power, high speed CMOS FLASH/EEPROM technology
- Fully static design
- In-Circuit Serial Programming™ (ICSP) via two pins
- Single 5V In-Circuit Serial Programming capability
- In-Circuit Debugging via two pins
- Processor read/write access to program memory
- Wide operating voltage range: 2.0V to 5.5V
- High Sink/Source Current: 25 mA
- Commercial, Industrial and Extended temperature ranges
- Low-power consumption:
  - < 0.6 mA typical @ 3V, 4 MHz
  - 20 μA typical @ 3V, 32 kHz
  - < 1 μA typical standby current

Pin Diagram



Peripheral Features:

- Timer0: 8-bit timer/counter with 8-bit prescaler
- Timer1: 16-bit timer/counter with prescaler, can be incremented during SLEEP via external crystal/clock
- Timer2: 8-bit timer/counter with 8-bit period register, prescaler and postscaler
- Two Capture, Compare, PWM modules
  - Capture is 16-bit, max. resolution is 12.5 ns
  - Compare is 16-bit, max. resolution is 200 ns
  - PWM max. resolution is 10-bit
- 10-bit multi-channel Analog-to-Digital converter
- Synchronous Serial Port (SSP) with SPI™ (Master mode) and I^{2}C™ (Master/Slave)
- Universal Synchronous Asynchronous Receiver Transmitter (USART/SCI) with 9-bit address detection
- Parallel Slave Port (PSP) 8-bits wide, with external RD, WR and CS controls (40/44-pin only)
- Brown-out detection circuitry for Brown-out Reset (BOR)

Key Features PICmicro™ Mid-Range Reference Manual (DS33023)	PIC16F873	PIC16F874	PIC16F876	PIC16F877
Operating Frequency	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz	DC - 20 MHz
RESETS (and Delays)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)	POR, BOR (PWRT, OST)
FLASH Program Memory (14-bit words)	4K	4K	8K	8K
Data Memory (bytes)	192	192	368	368
EEPROM Data Memory	128	128	256	256
Interrupts	13	14	13	14
I/O Ports	Ports A,B,C	Ports A,B,C,D,E	Ports A,B,C	Ports A,B,C,D,E
Timers	3	3	3	3
Capture/Compare/PWM Modules	2	2	2	2
Serial Communications	MSSP, USART	MSSP, USART	MSSP, USART	MSSP, USART
Parallel Communications	—	PSP	—	PSP
10-bit Analog-to-Digital Module	5 input channels	8 input channels	5 input channels	8 input channels
Instruction Set	35 instructions	35 instructions	35 instructions	35 instructions

APPENDIX D

PIC16F84A Microcontroller Data Sheet

MICROCHIP

PIC16F84A Data Sheet

18-pin Enhanced FLASH/EEPROM

8-bit Microcontroller

MICROCHIP

PIC16F84A

18-pin *Enhanced* FLASH/EEPROM 8-Bit Microcontroller

High Performance RISC CPU Features:

- Only 35 single word instructions to learn
- All instructions single-cycle except for program branches which are two-cycle
- Operating speed: DC - 20 MHz clock input
  DC - 200 ns instruction cycle
- 1024 words of program memory
- 68 bytes of Data RAM
- 64 bytes of Data EEPROM
- 14-bit wide instruction words
- 8-bit wide data bytes
- 15 Special Function Hardware registers
- Eight-level deep hardware stack
- Direct, indirect and relative addressing modes
- Four interrupt sources:
  - External RB0/INT pin
  - TMR0 timer overflow
  - PORTB<7:4> interrupt-on-change
  - Data EEPROM write complete

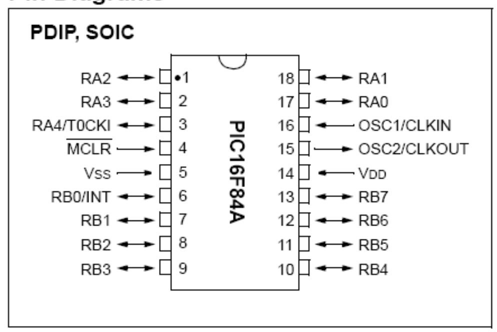
Peripheral Features:

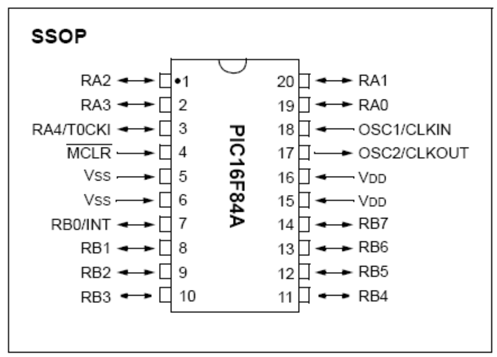
- 13 I/O pins with individual direction control
- High current sink/source for direct LED drive
  - 25 mA sink max. per pin
  - 25 mA source max. per pin
- TMR0: 8-bit timer/counter with 8-bit programmable prescaler

Special Microcontroller Features:

- 10,000 erase/write cycles *Enhanced* FLASH Program memory typical
- 10,000,000 typical erase/write cycles EEPROM Data memory typical
- EEPROM Data Retention > 40 years
- In-Circuit Serial Programming™ (ICSP™) - via two pins
- Power-on Reset (POR), Power-up Timer (PWRT), Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own On-Chip RC Oscillator for reliable operation
- Code protection
- Power saving SLEEP mode
- Selectable oscillator options

Pin Diagrams





CMOS Enhanced FLASH/EEPROM Technology:

- Low power, high speed technology
- Fully static design
- Wide operating voltage range:
  - Commercial: 2.0V to 5.5V
  - Industrial: 2.0V to 5.5V
- Low power consumption:
  - < 2 mA typical @ 5V, 4 MHz
  - 15 µA typical @ 2V, 32 kHz
  - < 0.5 µA typical standby current @ 2V

PIC16F84A

TABLE 1-1: PIC16F84A PINOUT DESCRIPTION

Pin Name	PDIP No.	SOIC No.	SSOP No.	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	16	16	18	I	ST/CMOS(3)	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	15	15	19	O	—	Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, OSC2 pin outputs CLKOUT, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
$\overline{\text{MCLR}}$	4	4	4	I/P	ST	Master Clear (Reset) input/programming voltage input. This pin is an active low RESET to the device.
						PORTA is a bi-directional I/O port.
RA0	17	17	19	I/O	TTL	
RA1	18	18	20	I/O	TTL	
RA2	1	1	1	I/O	TTL	
RA3	2	2	2	I/O	TTL	
RA4/T0CKI	3	3	3	I/O	ST	Can also be selected to be the clock input to the TMR0 timer/counter. Output is open drain type.
						PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-up on all inputs.
RB0/INT	6	6	7	I/O	TTL/ST(1)	RB0/INT can also be selected as an external interrupt pin.
RB1	7	7	8	I/O	TTL	
RB2	8	8	9	I/O	TTL	
RB3	9	9	10	I/O	TTL	
RB4	10	10	11	I/O	TTL	Interrupt-on-change pin.
RB5	11	11	12	I/O	TTL	Interrupt-on-change pin.
RB6	12	12	13	I/O	TTL/ST(2)	Interrupt-on-change pin. Serial programming clock.
RB7	13	13	14	I/O	TTL/ST(2)	Interrupt-on-change pin. Serial programming data.
V_{SS}	5	5	5,6	P	—	Ground reference for logic and I/O pins.
V_{DD}	14	14	15,16	P	—	Positive supply for logic and I/O pins.

Legend: I= input O = Output I/O = Input/Output P = Power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as the external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

APPENDIX E

LM7805 Voltage Regulator Data Sheet

MC78XX/LM78XX/MC78XXA

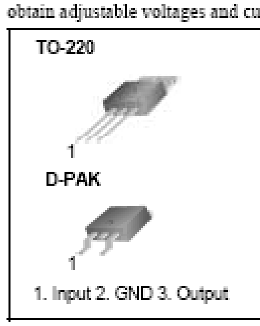
3-Terminal 1A Positive Voltage Regulator

Features

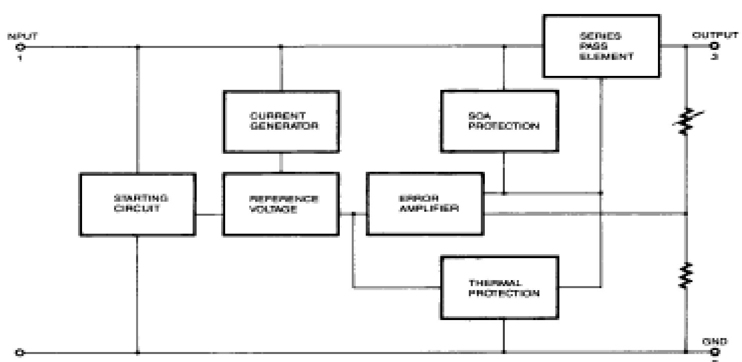
- Output Current up to 1A
- Output Voltages of 5, 6, 8, 9, 10, 12, 15, 18, 24V
- Thermal Overload Protection
- Short Circuit Protection
- Output Transistor Safe Operating Area Protection

Description

The MC78XX/LM78XX/MC78XXA series of three terminal positive regulators are available in the TO-220/D-PAK package and with several fixed output voltages, making them useful in a wide range of applications. Each type employs internal current limiting, thermal shut down and safe operating area protection, making it essentially indestructible. If adequate heat sinking is provided, they can deliver over 1A output current. Although designed primarily as fixed voltage regulators, these devices can be used with external components to obtain adjustable voltages and currents.



Internal Block Digram



Absolute Maximum Ratings

Parameter	Symbol	Value	Unit
Input Voltage (for V_O = 5V to 18V) (for V_O = 24V)	V_I V_I	35 40	V V
Thermal Resistance Junction-Cases (TO-220)	$R_{\theta JC}$	5	$^oC/W$
Thermal Resistance Junction-Air (TO-220)	$R_{\theta JA}$	65	$^oC/W$
Operating Temperature Range	T_{OPR}	0 ~ +125	^oC
Storage Temperature Range	T_{STG}	-65 ~ +150	^oC

Electrical Characteristics (MC7805/LM7805)

(Refer to test circuit ,0°C < T_J < 125°C, I_O = 500mA, V_I = 10V, C_I= 0.33µF, C_O= 0.1µF, unless otherwise specified)

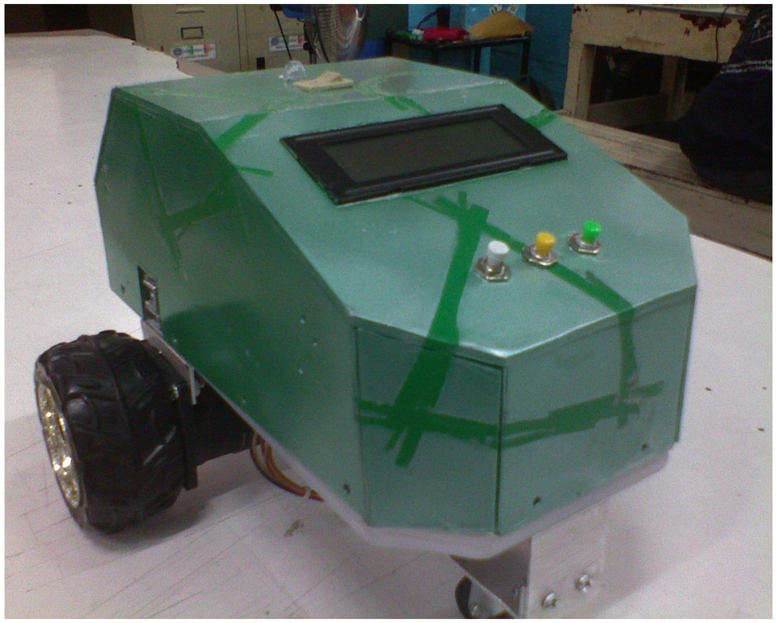
Parameter	Symbol	Conditions		MC7805/LM7805			Unit
				Min.	Typ.	Max.	
Output Voltage	V_O	T_J =+25 ^oC		4.8	5.0	5.2	V
		5.0mA ≤ Io ≤ 1.0A, P_O ≤ 15W V_I = 7V to 20V		4.75	5.0	5.25	
Line Regulation (Note1)	Regline	T_J=+25 ^oC	V_O = 7V to 25V	-	4.0	100	mV
			V_I = 8V to 12V	-	1.6	50	
Load Regulation (Note1)	Regload	T_J=+25 ^oC	I_O = 5.0mA to1.5A	-	9	100	mV
			I_O =250mA to 750mA	-	4	50	
Quiescent Current	I_Q	T_J =+25 ^oC		-	5.0	8.0	mA
Quiescent Current Change	ΔI_Q	I_O = 5mA to 1.0A		-	0.03	0.5	mA
		V_I= 7V to 25V		-	0.3	1.3	
Output Voltage Drift	$\Delta V_O/\Delta T$	I_O= 5mA		-	-0.8	-	mV/ ^oC
Output Noise Voltage	V_N	f = 10Hz to 100KHz, T_A=+25 ^oC		-	42	-	µV/Vo
Ripple Rejection	RR	f = 120Hz V_O = 8V to 18V		62	73	-	dB
Dropout Voltage	V_{Drop}	I_O = 1A, T_J =+25 ^oC		-	2	-	V
Output Resistance	r_O	f = 1KHz		-	15	-	mΩ
Short Circuit Current	I_{SC}	V_I = 35V, T_A =+25 ^oC		-	230	-	mA
Peak Current	I_{PK}	T_J =+25 ^oC		-	2.2	-	A

Note:

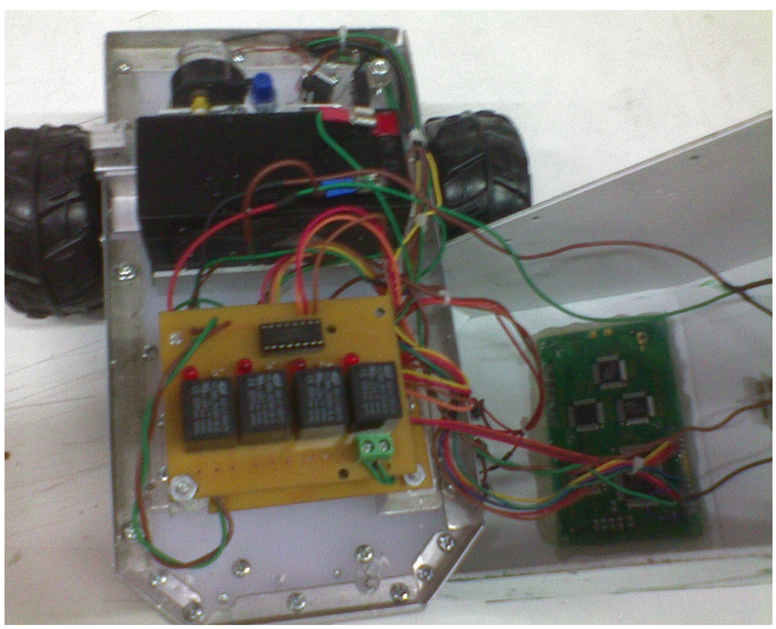
1. Load and line regulation are specified at constant junction temperature. Changes in V_0 due to heating effects must be taken into account separately. Pulse testing with low duty is used.

APPENDIX F

Picture of Prototype



Actual Design Front View



Actual Design Top View

Actual Design Back View

APPENDIX G

List of Miscellaneous Materials

MISCELLANEOUS MATERIALS		
QTY	PART-REFS	DESCRIPTION
1	BAT1	12V
1	LCD1	LM044L
1	X1	CRYSTAL
1	Sheets	Sentra board
100	6x32	Machine Bolt
100	6x32	Nut
2	Motor	Stepper
1	Motor	DC Geared Motor
1	1mtr Length	Angle bar alumminun
2	Wheel	Rubber wheel
4	connector	M/F Wire connector
1	IC socket	40pin IC Socket
2	IC socket	18pin IC Socket
2	Header	40pin Header male
2	LED	10mm LED
1	SWITCH	SPST Switch
2	Connector	Battery terminal connector
1	meter	Heat shrinkable tube 2mm
1	Adhesive	Mightybond
1	Pen	Pentel pen
3	acetate foil	acetate foil
2	pack	Ferric chloride
1	pack	DP-50 Developer
1	sheet	12x12 Presensitized PCB

APPENDIX H

Drawbot: A Microcontroller-Based Drawing Robot

User Manual

Mapua Institute of Technology - COE461D Computer Design

About Drawbot

The design named Drawbot has the capability of drawing basic shapes with a definite size. These are only basic shapes that consist of circle, square, triangle and rectangle with sizes of small, medium, or large. The main purpose of this study is to design a robot that uses stepper motors and PIC microcontroller to draw.

Characteristics

1 Uses Microcontrollers PIC16F877 for the firmware

2 It uses an LCD for the menu.

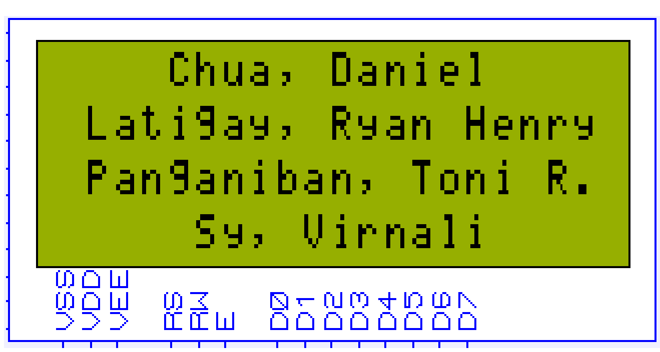
3 The three buttons represents the control of the LCD for user input.

4 The pen holder pin downs the pen in place

5 The LCD displays the options for drawing the four (4) shapes, namely circle, square, rectangle and a equilateral triangle that have three (3) different sizes, which are, small, medium and large.

User Guide

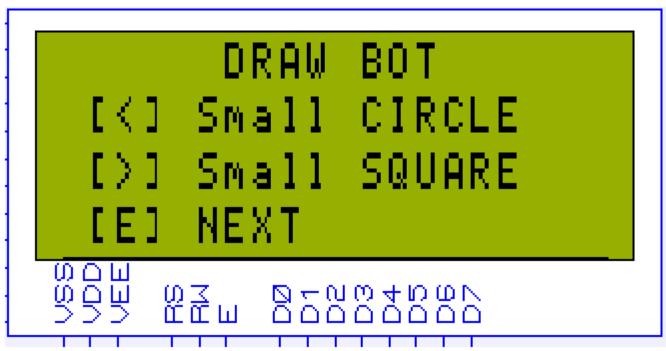
1 When the Drawbot is turned on, the LCD would display the options of the menu that are available.



2 The Drawbot has three push buttons that can control the menu on the LCD namely; left[<], right[>], enter[E].

3 The options on the menu of the LCD will enable the user to input the desired shape and size.

4 The menu has twelve options to choose from. It corresponds to the format [size] [shape].



The LCD Menu layout of the Drawbot

5 Once the option is pressed, the Drawbot would draw the shape and the LCD would display a waiting message ("Drawbot Please wait… Drawing small circle now…")

6 After the drawing, the Drawbot would return to its menu awaiting the new input from the user.

APPENDIX I

Approximations of Errors In Measurement

Calculus 1 Problems & Solutions – Chapter 5 – Section 5.3.2

5.3.2 Approximations Of Errors In Measurement

Return To Contents
Go To Problems & Solutions

Review

1. Approximations

If a quantity x (eg, side of a square) is obtained by measurement and a quantity y (eg, area of the square) is calculated as a function of x, say $y = f(x)$, then any error involved in the measurement of x produces an error in the calculated value of y as well.

Suppose the distance x_m between 2 points is measured to be 1,000 m with an error of 1 m. This means that the measured distance x_m is 1,000 m: x_m = 1,000, and the actual distance x_a is somewhere between 1,000 m − 1 m = 999 m and 1,000 m + 1 m = 1,001 m; it's somewhere in the interval [999, 1,001]. We say that the distance is measured to be 1,000 m ± 1 m. The error of 1 m is how much x_a can differ from x_m: $|x_a - x_m| = 1$. Now x_m and x_a are 2 particular values of a variable say x. More precisely x_m is a value of x and x_a is a value which x can change from x_m to. So the error of 1 m can be considered as a change in x.

The error in x can be considered as a change in x, and thus is denoted by Δx. The change Δx in x is the differential dx of x. Consequently the error in x is $\Delta x = dx$. See Fig. 1.1. The corresponding error in y is $\Delta y = f(x + dx) - f(x)$, and the corresponding differential of f at x is $dy = f'(x)\,dx$. Intutively we can approximate Δy by dy. Now let's show that we can do such an approximation. By the 1st part of the previous section, We have:

$$f(x + dx) \approx f(x) + f'(x)\,dx.$$

Hence:

$$\Delta y = f(x + dx) - f(x) \approx (f(x) + f'(x)\,dx) - f(x) = f'(x)\,dx = dy.$$

Indeed we can approximate Δy by dy. Intuitively approximating the height of the graph of f by that of the tangent amounts to approximating the length Δy by the length dy, as clearly seen in Fig. 1.1. In summary:

$\Delta x = dx$,
$\Delta y \approx dy = f'(x)\, dx$.

That is, the error in x is dx and the corresponding *approximate* error in y is $dy = f'(x)\, dx$.

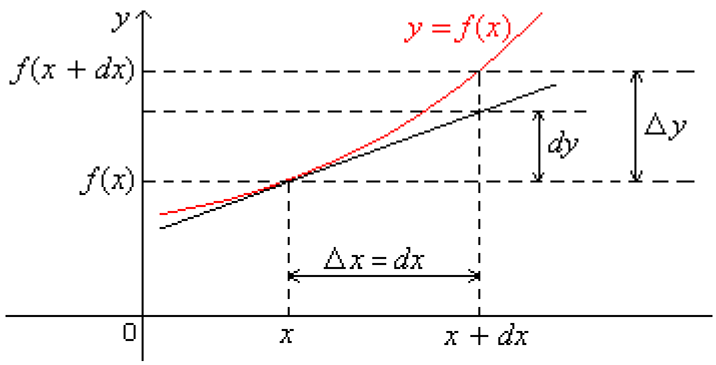


Fig. 1.1

$\Delta x = dx$,
$\Delta y \approx dy = f'(x)\, dx$.

As seen above we say that the distance between 2 points is measured to be 1,000 m ± 1 m when its measurement is 1,000 m with an error of 1 m. The measured distance is x = 1,000, the error is dx = 1, and the actual distance x_a is somewhere in the interval [1,000 − 1, 1,000 + 1] = [999, 1,001]. If we instead interpret differently by treating that x_a is such that x = 1,000 is somewhere in the interval [x_a − 1, x_a + 1] or $x_a - 1 \le 1{,}000 \le x_a + 1$, then from $x_a - 1 \le 1{,}000$ we get $x_a \le 1{,}001$ and from $1{,}000 \le x_a + 1$ we get $999 \le x_a$. It follows that $999 \le x_a \le 1{,}001$, ie x_a is somewhere in the interval [999, 1,001], the same situation as in the 1st interpretation. Also see Fig. 1.2.

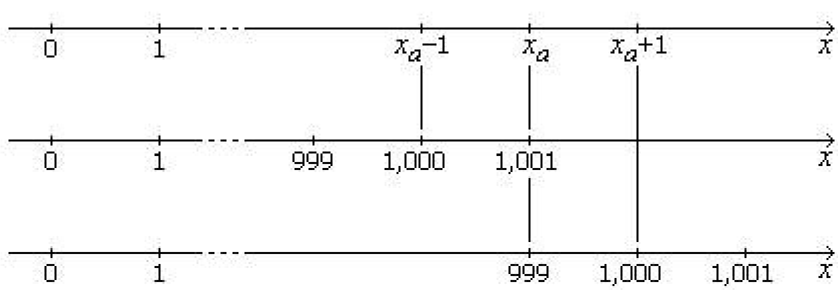


Fig. 1.2

– 1st and 2nd axes: if 1,000 = x_a – 1 then x_a = 1,001,
– 1st and 3rd axes: if 1,000 = x_a + 1 then x_a = 999,
therefore x_a is somewhere in [999, 1,001].

We say that a quantity is measured to be (or its measurement is) $x \pm dx$ where $dx \ge 0$ when its measurement is x and the error in the measurement is dx. The measured value of the quantity is x, the error is dx, and the actual value of the quantity is somewhere in the interval [$x - dx$, $x + dx$].

Example 1.1

The side of a square is measured to be 1,000 m ± 1 m. Find the approximate error of the calculated area of the square.

Solution

Let s be the side and A the area of the square. Then $A = s^2$. The error of the side is ds = 1 m. The approximate error of the calculated area is:

$dA = 2s\ ds = 2(1{,}000)(1) = 2{,}000\ \text{m}^2$.

EOS

Note that we calculate dA from the equation $A = s^2$, since the values of s and ds are given. To find the differential of A we must have an equation relating A to s. So even if the measured value of the side is given we still define the variable s that takes on as a value the measured value.

In general, when the measured value say V of a quantity and the error say E in the measurement are given, we define a variable say x for the quantity, so that $x = V$ and $dx = E$, which will be used later on in the solution. When using the quantity, first use the variable x, not the value V, then use the value V when a value is to be obtained.

2. Types Of Errors

A measurement of distance d_1 yields d_1 = 100 m with an error of 1 m. A measurement of distance d_2 yields d_2 = 1,000 m with an error of 1 m. Both measurements have the same absolute error of 1 m. However, intuitively we feel that measurement of d_2 has a smaller error because it's 10 times larger and yet has the same absolute error. Clearly the effect of 1 m out of 1,000 m is smaller than that of 1 m out of 100 m. This leads us to consider an error relative to the size of the quantity being expressed. This relative error is accomplished by representing the absolute error as a fraction of the quantity being expressed. For example, the relative error for d_1 is 1 m / 100 m = 1/100 = 0.01 and that for d_2 is 1 m / 1,000 m = 1/1,000 = 0.001. As desired the relative error for d_2 is smaller than that for d_1.

The percentage error is the absolute error as a percentage of the quantity being expressed. For example, the percentage error for d_1 is (1 m / 100 m)(100/100) = (1/100)(100)% = (0.01)(100)% = 1% and that for d_2 is

(1 m / 1,000 m)(100/100) = (1/1,000)(100)% = (0.001)(100)% = 0.1%. We see that the percentage error is the relative error expressed as a percentage. If the relative error is r then the percentage error is p% = r . (100/100) = (r . 100)%.
So conversely if the percentage error is p% then the relative error is r = p/100.

In general:

dx = **absolute error** of x,

$\frac{dx}{x}$ = **relative error** of x,

$\frac{dx}{x}\left(\frac{100}{100}\right) = \left(\frac{dx}{x}\cdot 100\right)\%$ = **percentage error** of x,

dy = **approximate absolute error** of y,

$\frac{dy}{y}$ = **approximate relative error** of y,

$\frac{dy}{y}\left(\frac{100}{100}\right) = \left(\frac{dy}{y}\cdot 100\right)\%$ = **approximate percentage error** of y.

Example 2.1

The side of a square is measured to be 200 m ± 60 cm. Find the percentage error of the side and the approximate percentage error of the calculated area of the square.

Solution

Let s be the side and A the area of the square. Then $A = s^2$. The absolute error of the side is ds = 60 cm = 0.6 m. The relative error of the side is:

$$\frac{ds}{s} = \frac{0.6}{200} = 0.003.$$

So the percentage error of the side is (0.003)(100/100) = 0.3%.

The approximate relative error of the calculated area is:

$$\frac{dA}{A} = \frac{2s\,ds}{s^2} = \frac{2\,ds}{s} = \frac{2(0.6)}{200} = 0.006.$$

Thus the approximate percentage error of the calculated area is (0.006)(100/100) = 0.6%.

EOS

Problems & Solutions

1. The side of a square is measured to be 200 m ± 60 cm. Find the approximate error of the calculated area of the square.

Solution

Let s be the side and A the area of the square. Then $A = s^2$. The error of the side is ds = 60 cm = 0.6 m. The approximate error of the calculated area is:

$dA = 2s\,ds = 2(200)(0.6) = 240$ m^2.

2. The radius of a sphere is measured to be 5 m ± 10 cm. Find the approximate percentage error of the calculated volume of the sphere. (The volume V of a sphere of radius r is $V = (4/3)\pi r^3$.)

Solution

Let r and V be the radius and volume of the sphere respectively. The absolute error of the radius is dr = 10 cm = 0.1 m. As $V = (4/3)\pi r^3$ the approximate relative error of the calculated volume of the sphere is:

$$\frac{dV}{V} = \frac{\frac{4}{3}(3)\pi r^2\,dr}{\frac{4}{3}\pi r^3} = \frac{3\,dr}{r} = \frac{3(0.1)}{5} = 0.06.$$

So the approximate percentage error of the calculated volume of the sphere is (0.06)(100/100) = 6%.

3. The edge of a cube is measured to within 2% tolerance. Approximately what percentage error can result in the calculation of the volume of the cube?

Solution

Let a be the edge and V the volume of the cube. Then $V = a^3$. The percentage error of the edge is 2% and so its relative error is da/a = 2/100 = 0.02. The approximate relative error that can result in the calculation of the volume is:

$$\frac{dV}{V} = \frac{3a^2\,da}{a^3} = 3\frac{da}{a} = 3(0.02) = 0.06.$$

Thus the approximate percentage error that can result in the calculation of the volume is (0.06)(100/100) = 6%.

4. It is desired that the computed area of a circle is with at most 2% error by measuring its radius. Approximate the
maximum allowable percentage error that may be made in measuring the radius.

Solution

Let r and A be the radius and area of the circle respectively. Then $A = \pi r^2$. The maximum relative error of the computed area is $\Delta A/A = 2/100 = 0.02$. Now:

$$\frac{\Delta A}{A} \approx \frac{dA}{A} = \frac{2\pi r\,dr}{\pi r^2} = \frac{2\,dr}{r}.$$

So the maximum allowable relative error that may be made in measuring the radius is:

$$\frac{dr}{r} \approx \frac{1}{2}\left(\frac{\Delta A}{A}\right) = \frac{1}{2}(0.02) = 0.01.$$

Thus the approximate maximum allowable percentage error that may be made in measuring the radius is
(0.01)(100/100) = 1%.

Note

Here the exact error of the function (area) is given and the approximate error of the variable (radius) is to be found. The
symbol:

$$\frac{dr}{r}$$

represents the relative error, not an approximate relative error, of the radius. It's the value 0.01 that's an approximate
value of this relative error.

Source: CALCULUS 1 PROBLEMS & SOLUTIONS An Introductory–Calculus Tutorial By Pheng Kim Ving, BA&Sc, MSc
http://www.geocities.com/phengkimving/index.htm