DESIGN AND CONSTRUCTION OF PEN MOUSE

By

FAIROLIZWAN BIN ABDULLAH

FINAL PROJECT REPORT

Submitted to the Electrical & Electronic Engineering Programme
in Partial Fulfillment of the Requirements
for the Degree
Bachelor of Engineering (Hons)
(Electrical & Electronic Engineering)

Universiti Teknologi Petronas
Bandar Seri Iskandar
31750 Tronoh
Perak Darul Ridzuan

CERTIFICATION OF APPROVAL

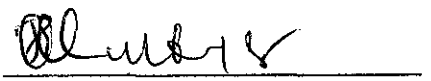
DESIGN AND CONSTRUCTION OF PEN MOUSE

by

Fairolizwan Bin Abdullah

A project dissertation submitted to the
Electrical & Electronic Engineering Programme
Universiti Teknologi PETRONAS
in partial fulfilment of the requirement for the
Bachelor of Engineering (Hons)
(Electrical & Electronic Engineering)

Approved:



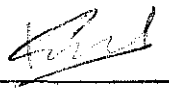
Dr. Mumtaj Begam Kasim Rawthar
Project Supervisor

UNIVERSITI TEKNOLOGI PETRONAS
TRONOH, PERAK

June 2007

CERTIFICATION OF ORIGINALITY

This is to certify that I am responsible for the work submitted in this project, that the original work is my own except as specified in the references and acknowledgements, and that the original work contained herein have not been undertaken or done by unspecified sources or persons.



Fairolizwan Bin Abdullah

ABSTRACT

This project is about design and construction of a suitable mechanical assembly to convert the conventional mouse to a pen shaped mouse. The purpose of the design is to improve the capability of presentation using computer using the Microsoft Office PowerPoint Software. We are accustomed to using pen for writing or drawing in our daily life. Therefore a pen shape mouse design is better than the conventional mouse for power point presentation. In addition, the conventional mouse needs more space for pointer movement compared to a pen shape mouse. The project work consists of literature research, pen shape mouse hardware design prototype, circuit fabrication and performance evaluation. The report also discusses the function of the pointer in the PowerPoint slide presentation and the correct way of using it. The Pen shape mouse device is able to move the cursor on the screen and perform the normal mouse function. However the design is still lacking in the motion sensitivity. Nonetheless the Pen mouse design proved to be very useful in enhancing PowerPoint presentation that allows natural writing just like using a real pen.

ACKNOWLEDGEMENTS

I would like to take opportunity to express my gratitude to Dr. Saeed Ul- Hassan Naqvi for his continuing guidance, valuable discussions, and kind encouragement throughout this study. He has contributed many of his idea on the design mechanism.

Also to my replacement supervisor, Dr. Mumtaj Begam Kasim Rawthar not only served as my supervisor but also, for her many suggestions and constant support during this research. They both have patiently guided me through the project work, never accepting less than my best efforts.

I am thankful to the entire electrical lab technician especially to Kak Hawa and Mr. Isnani who helps me a lot in the project papers and project work. I could not have done much better without their guidance. I thank them all.

June 2007

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

DPI -- DPI stands for Dots Per Inch. To understand DPI in mouse: Vertical lines multiplied by the Horizontal lines divided by the screen size and finally divided by the DPI of the mouse.

IC -- Integrated circuit is a miniaturized electronic circuit that has been manufactured in the surface of a thin substrate of semiconductor material.

IDE -- integrated development environment also known as integrated design environment and integrated debugging environment, is a type of computer software that assists computer programmers in developing software.

OHP -- overhead projector is a display system that is used to display images to an audience.

PCB -- Printed circuit Board is used to electrically connect electronic components using conductive pathways, or traces, etched from copper sheets.

PIC -- Programmable integrated circuit

PS2 -- PS/2 connector is used for connecting a keyboard and a mouse to a PC compatible computer system.

LED -- Light Emitting Diode is a semiconductor device that emits incoherent narrow-spectrum light when electrically biased in the forward direction of the p-n junction.

CHAPTER 1
INTRODUCTION

1.1 Background of Study

The project is to design and construct a suitable mechanical assembly to convert the conventional mouse to a pen shaped mouse. This can be achieved by transferring the signals from the designed mechanical motion tracker of the pen shaped mouse to the computer. This device would help quite a number of people to present their lectures and presentations more comfortably. The name Presentation pen mouse is really suited to it purpose for enhancing power point presentation in writing and drawing. It is a pen-shaped mouse for PC that allows natural writing just like using a real pen. Presentation pen mouse is lightweight, durable, and accurate. User can perform natural handwriting to do more creative work or special task that was previously impossible with conventional mouse. Especially, because it requires very small surface to write on, it is very ideal for notebook users and perfect for places with limited workspace is allowed.

1.2 Problem Statement

Pointing devices are supposed to assist user to use the computer according to their working condition. For example, in a presentation the user may need to describe the topic by drawing or writing for interactive communication with the audience. Usually the presenters use an overhead projector (OHP) for this purpose which causes an eye strain to the presenter due to high intensity of the lamp. The use of white board is not preferable for a large audience. It also involves time consuming cleaning of board for getting more writing space. With a pen shape mouse the problems stated above can be eliminated.

1.3 Objective and Scope of Study

The project consists of three main objectives which are described below:

- To construct a functional pen shape mouse by modifying the conventional mouse.
- To enhance presentation in power point software using the mouse pen shape design which gives the natural feeling of writing or drawing.
- The performance evaluation of the design will also be done to rate its effectiveness and quality.

The Scope of study consists of two main items which will be covered throughout the project as stated below:

- To improve the usage of conventional mouse by producing a working design mouse in the shape of a pen which also perform like a pen input to the computer
- The project will be focusing on computer system design which involves programming, modeling, circuit design, simulation and product design.

CHAPTER 2
LITERATURE REVIEW

The pen shape mouse is a new design that will help people in doing better Power Point presentation. The design function the same as the normal mouse in term of its communication with computer through PS2 port. The literature review will briefly describe the information concerning the mouse design.

2.1 Introduction of Pointing Device

Pointing devices are one of the main devices for most PCs used to move an onscreen pointer (usually an arrow or pointing hand) and to select objects located on the screen. There are several ways of moving the onscreen pointer ball type takers. Some devices move on a surface, rotating a ball by finger on top of the device etc. Usually the design in all these devices is same for selecting the object and clicking a button on the device for final selection. Other common type of pointing devices are the optical mouse, light pen, touch pen, touch screen, joystick, pad game.

2.2 A Comparison of Different Mouse Types

The Pen shape mouse design mechanism is from the opto-mechanical mouse due to the result of comparison of different mouse types. The other mouse such as conventional mouse, touch screen, trackball and touch pad does not give the natural feeling of writing. These devices also have a lot of disadvantage such as muscle tiredness due to repeated motion of certain muscles. However with the pen shape mouse the size is smaller and the motion does not strain on certain muscles instead all the motion distributed through the all fingers muscle. The comparisons on the devices advantage and disadvantage describe earlier are summarize in Table 1.

Table 1: Advantage and disadvantage of different mouse types

Mouse type	Advantage	Disadvantage
Conventional Mouse	• Moves cursor around the screen faster than using keystrokes.	• Requires moving hand from keyboard to mouse and back. • Repeated motion can lead to carpal tunnel syndrome
Trackball	• Does not need as much desk space as a mouse. • Is not as tiring since less motion is needed.	• Requires fine control of the ball with just one finger or thumb. • Repeated motions of the same muscles are tiring and can cause carpal tunnel syndrome.
Glide pad / Touchpad	• Does not need as much desk space as a mouse. • Can readily be built into the keyboard. • Has finer resolution. That is, to achieve the same cursor movement onscreen takes less movement of the finger on the glide pad than it does mouse movement. • Can use either buttons or taps of the pad for clicking.	• The hand tires faster than with a mouse since there is no support. • Some people don't find the motion as natural as a mouse. • Harder for precisions movement example for drawing
Touchscreen	• It's natural to do - reach out and touch something.	• It's tiring if many choices must be made. • It takes a lot of screen space for each choice since fingers are bigger than cursors.
Pen Input	• Can use handwriting instead of typing • Can use gestures instead of typing commands • Small size	• Must train device to recognize handwriting. • Must learn gestures or train device to recognize the ones you create • Can lose the pen which is not usually attached to the device

2.3 Mouse Operation System

Sensors are the movement detectors (typically opto-mechanical) which sense the mouse movement and switches make the final selection using left or right click buttons. Mouse controller reads the state of those sensors and takes account of existing mouse position. When the position changes, the mouse controller sends a packet of data to the computer data interface controller. The mouse driver in the computer receives that data packet and decodes the information from it and takes action based on the information. Typical mouse driver has the information of the existing mouse state (position and button states) to inform the application or operating system when needed. The mouse driver calls mouse cursor moving routines when mouse is moved and sends messages to the software when buttons are pressed. The overall mouse operation systems are simplified in figure 1.

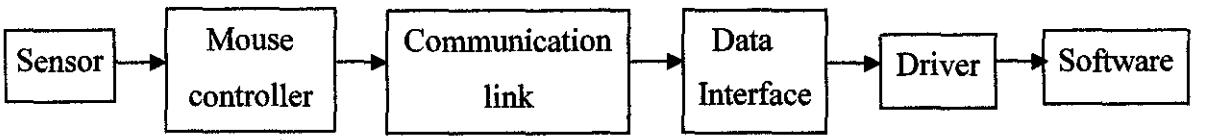


Figure 1: Typical mouse controlling system stages

2.4 How Ball Type Mouse Works

The most typical construction of ball type mouse is movement detectors which are opto-mechanical detectors. As the mouse is dragged along the surface, the ball rotates due to friction. Its motion is resolved into two axes by the rollers located at right angels to each other. Each roller is attached to a shaft that rotates an optical encoder. When the encoder rotates, it interrupts a beam of light between an LED and a photo detector. Each pulse is fed to the computer and by calculating the number of pulses received the length of the mouse motion can be determined. Circuit in the mouse translates the ball's movement into a signal that can be read by the computer [1].

2.5 How Optical Type Mouse Works

The optical mouse uses different method of movement sensing. It uses a tiny camera to track the difference of the thousand snapshots per second with the assistance of red Light Emitting Diode (LED) which affects the contrast of the surface. The changes in the snapshots provide the necessary information on the direction and speed of the movement.

2.6 Mouse Encoder operation

The ball type mice are very well known devices. They use a ball which rotates two encoders. A set of two rectangular waveforms are generated by each encoder. The pulses are counted and the cursor position is a function of number of pulses counted. To be able to make distinction between the back and forth motion, two set of pulses are used. These pulses are 90 ^{o} out of phase (quadrature pulses) as shown in figure 2.

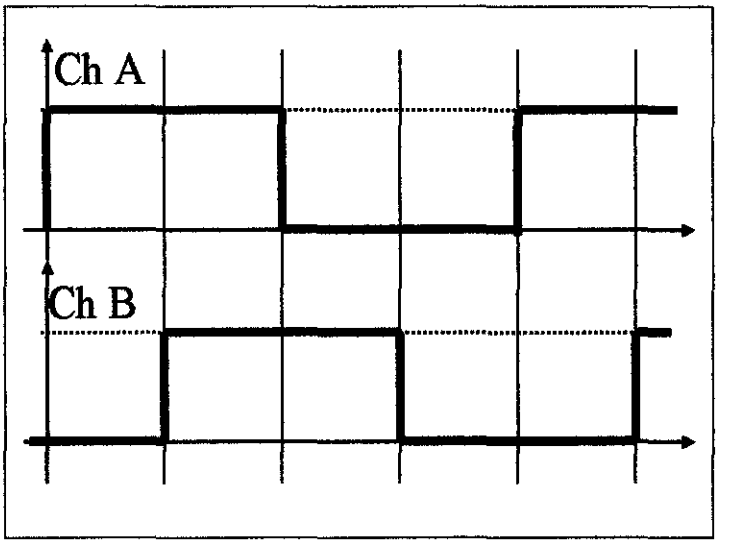


Figure 2: The channel waves of the mouse encoder

When the mouse is moving in one direction, the channel B waveform is 90 ° out of phase with channel A waveform (see figure2). In case of movement in opposite direction, the channel A waveform is 90 ° out of phase with channel B waveform. Based on these two waveforms, the mouse circuit is able to detect the direction of motion. Another circuit that can detect the direction of motion from these two waveforms is based on D type flip-flop. In this circuit one channel is used as clock and the other channel is used as input for the flip-flop [2].

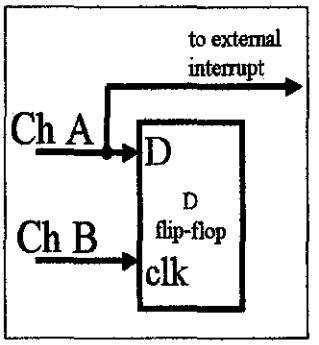


Figure 3: Flip flop for mouse encoder

The figure 3 shows a D flip-flop where channel B is used as clock signal. Basically, the flip-flop will copy the input to D on the output when the clock pulses arrive. This action takes place on the rising edge of the clock pulse. When the channel B wave is out of phase with respect to channel A (indicating that when channel B has rising edge, the A is already on 1), the output of D flip-flop will be 1. When channel A wave is out of phase with channel B (this means that when channel B has rising edge, the A is already on 0), the output of D flip-flop will be 0 [2].

2.7 Mouse Performance Factors

The factors that determine the quality of an optical mouse are resolution, refresh rate, image sensor, lens purity, light color, surface condition and processor. In the case of ball type mouse, the factors that determine the quality are ball traction and the count per inches characteristic. For any mouse design the common performance factors are its ergonomic design to be comfortable to user, resolution of the pointer movement, feasibility and cost effectiveness of the device.

2.8 PowerPoint Annotation Pen function

The ideal presentation in PowerPoint slides, the main agenda is to impress audience and providing them a better understanding. If the audience is interested in the presentation, the presenter will have to answer questions that he/she may not have slides to show. By using the Pen mouse function one can spontaneously emphasize key points, draw diagrams, incorporate input from the audience, take notes, and do lot more.

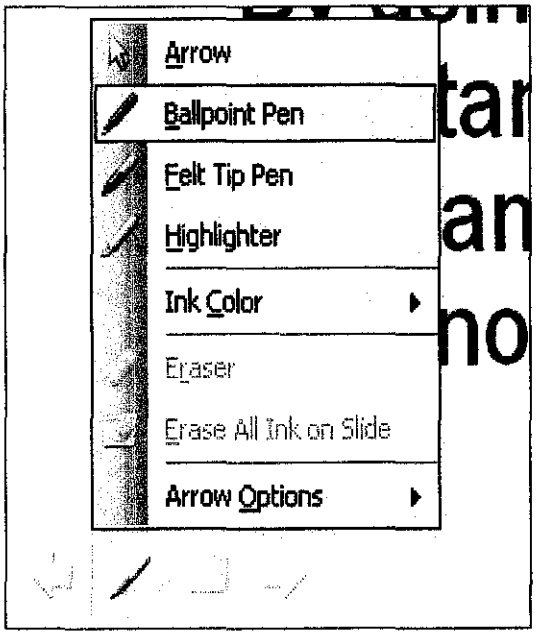


Figure 4: Pen function menu of PowerPoint Slide

The pen icon at the bottom left of the slide show screen to switch from the arrow to a ballpoint pen, a felt tip pen, or a highlighter. The menu provide ink color, save or discard ink marks when exit the slide. Keyboard shortcuts can, however, dramatically reduce preparation time and make slide shows much shorter. The slide show shortcuts can be viewed by pressing the help button [F1] during a slide show.

Among the shortcuts [b] or [w] keys are useful which turn the screen black or white. This function allows the use of PowerPoint screen as a blackboard or whiteboard for drawing diagrams and writing. However, pointer must be switched to pen mode before pressing [b] or [w], otherwise any writing will disappear when you exit the black or white screen.

Shown below are the proper ways of using the pointer function during the slide presentation:

1. Key points can be emphasized by highlighting, circling or underlining words during presentation as shown in figure 5. However for a dark background, the highlighter may be difficult to see.

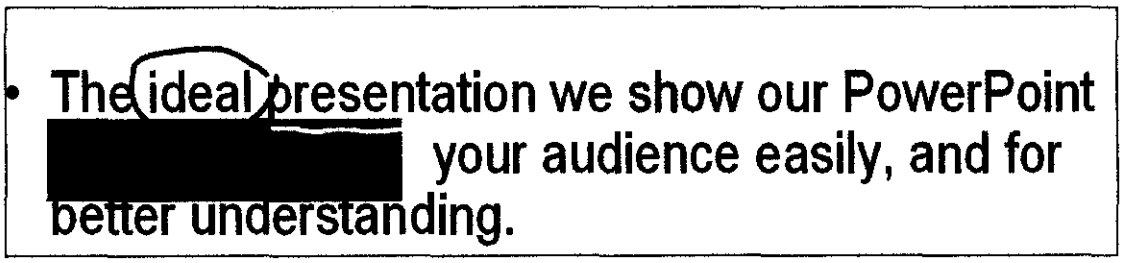


Figure 5: Example of good slide presentation using PowerPoint

2. For very strong emphasis, the word or number can be written on the screen. This leaves a very strong impact on the audience since handwriting looks different and the word stays on the screen during explanation.
3. Pausing for too long writing will leaves a distracting, uncomfortable silence in the room, and it destroys the rhythm of the presentation. The screen will get dirty and marks will lose impact if we use too many ink marks as shown in figure 6.

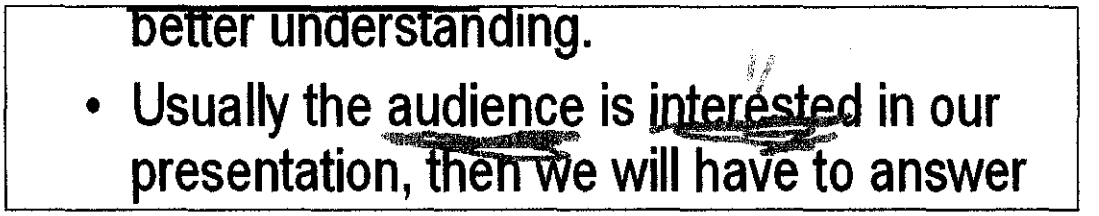


Figure 6: Example of bad slide presentation using PowerPoint

4. The blank screens [B] for black or [W] for white or prepared blank slides can be used as discussion space for notes, diagrams or equation examples.

2.9 Handwriting Recognition Function

Presenter handwriting can be the cause for a bad presentation. This can be solved by handwriting recognition software that enables the computer to receive handwritten input. Handwriting recognition principally consists of optical character recognition. However, a complete handwriting recognition system also handles formatting, performs correct segmentation into characters and finds the most plausible words. One of software equipped with this function is JustWrite Office Suite. JustWrite Office is an expansion for Microsoft Office programs and appears in the Office menu bar following installation. JustWrite Office comprises five components, each available according to the specific program. These components are JustWrite, JustSign, JustTag, JustShow, and Screen MarkUp. During preparation, JustWrite allows us to customize the width of the pen and pen color from the toolbar and we can create user pens for favorite colors. During the JustShow slideshow, in addition to the pen function, there are functions for laser pointers, drawing rectangles and circles, and unlike PowerPoint Ink, we can save the notes we take on blank slides. We can also print directly from the slide show.

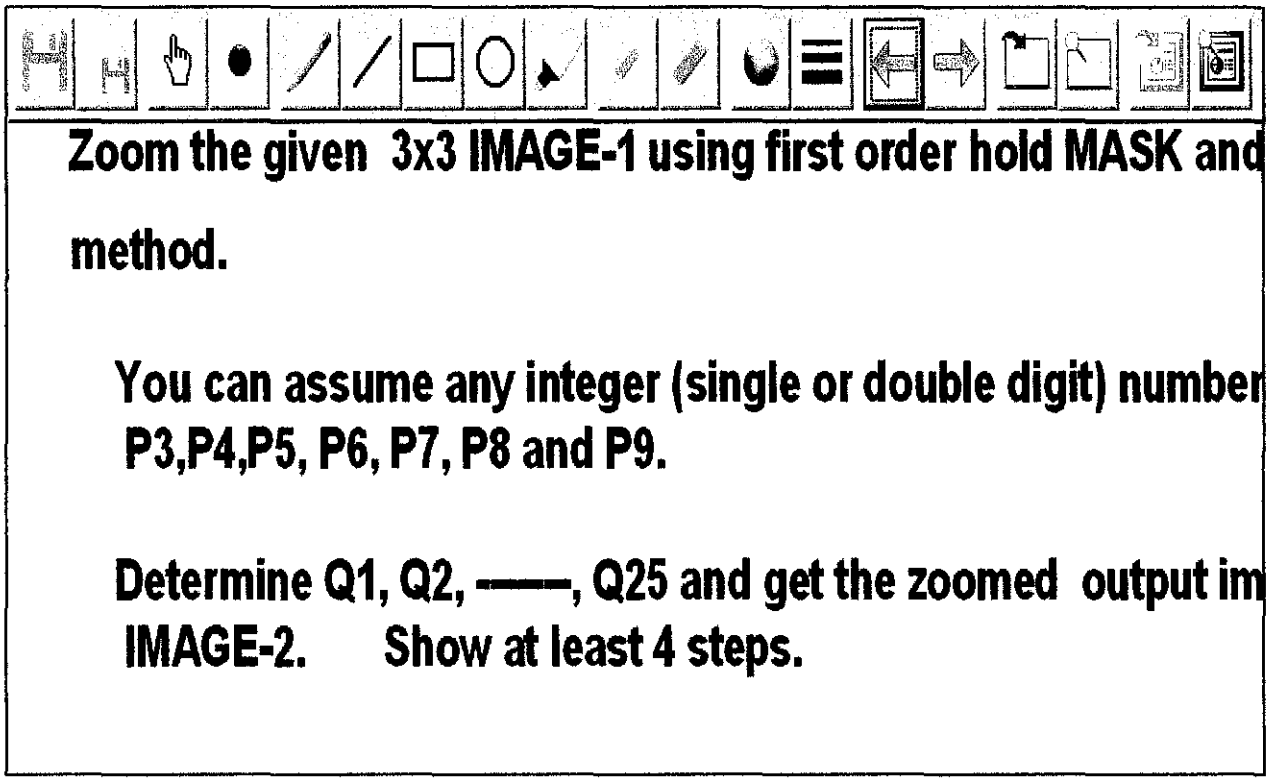


Figure 7: JustShow office software toolbar on Power Point slideshow

CHAPTER 3
METHODOLOGY/PROJECT WORK

In order to make this project work successfully few methods are applied. First I have divided the project work into 3 major phases which are phase I, Phase II and Phase III. The phases are divided according to its function and time of completion. By dividing the project work I manage to concentrate on each task and complete the task on time. The overall project work flow is shown in figure 8.

3.1 Project Work Phase I

Phase I of the project has two major components. The first consists of literature review and a survey of how different type of mouse function and how it communicates with the computer. I need to understand how the mouse circuit translates the motion of the ball to electrical signal so that I can further explore in microcontroller circuit. Moreover I need to do research on the software that can enhance the design mouse to work better with Power Point presentation. The second part of Phase I deal with the development of a detailed work plan for Phases II and III in Gantt chart together with its expected time of completion.

3.2 Project Work Phase II

In Phase II which is the main part of the project to produce a workable mouse prototype. The components involved in this phase are:

1. Draft the mouse design by selecting the movement detection method, material, and design drawing. This part will determine the sensitivity of the mouse design.
2. The construction of the mouse circuitry. This task consists of transferring the mouse circuit to the PCB board with fixed width and height to fit into the pen shape mouse design.
3. Construction of the design body and mechanical component.

4. Troubleshooting and testing of the design performance.
5. Presentation of the mouse design concept and functional.

3.3 Project Work Phase III

Phase III involves the project design improvement and evaluation of its performance. This is conducted by comparing it performance with the conventional mouse to prove whether the pen shape mouse design is better to be use to enhance power point presentation. Finally all activities of the study will be documented in a Final Report. The detailed project planning Gantt chart is attached in the appendices for references.

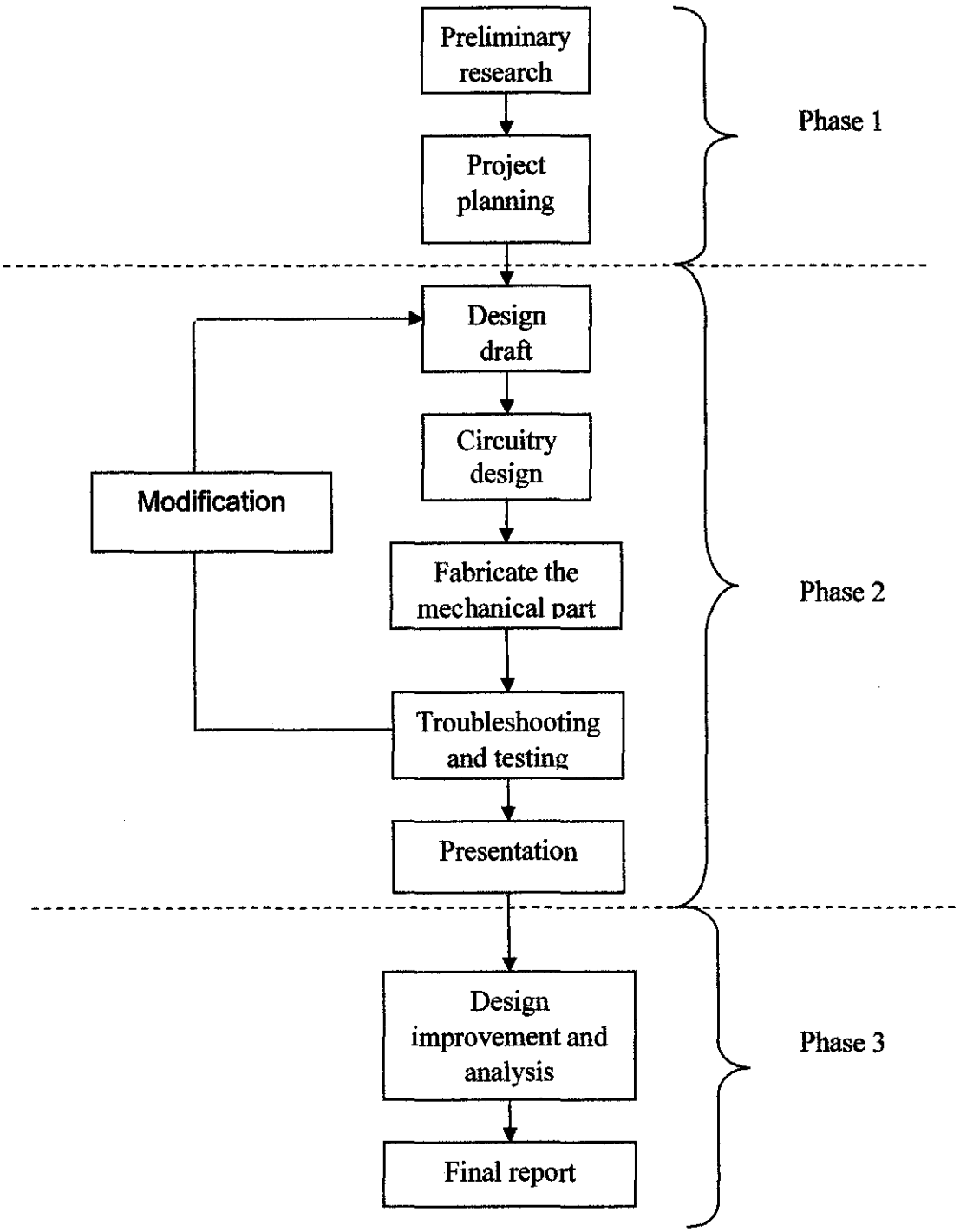


Figure 8: The full project work flow of the designing the pen shape mouse

3.4 Tool and Equipment

The project requires hardware and software of the mouse design for it to complete. The hardware equipment is needed to construct the mechanical part for detecting the direction motion and circuit part to encode the motion signal into electrical signal for the computer to understand. The software part is needed to draw the mouse design and to simulate the mouse assembly language controller coding.

3.4.1 Hardware equipment

For testing purpose thus I will need to use the electrical lab and the microprocessor lab. As for the hardware part the mechanical part needs to be fabricated using the workshops machines. Most of the activities are about connection-based process and some soldering. Below are the listed equipment used:

- Conventional Mouse Circuit
- Oscilloscope
- Multimeter
- Soldering station
- Power Supply
- Electronic components
- Small hand tools such as wire stripper and cutter
- Breadboard
- Personal Computer
- Microcontroller programmer circuit

3.4.2 Software equipment

The software that is used for this project is to simulate the desired output of the circuit. The simulation can predict the incoming result of the circuit whether work or not. The software that I used is:

- MPlab IDE
- PSPICE 9.2
- Eagle software

CHAPTER 4
RESULT AND DISCUSSION

4.1 Design Description

The pen mouse is still in the construction of the mechanical part. I have come out with a of design ideas of how to do the mechanical part. I have placed the buttons and the circuit of the design in the holder. The design consists of two cylinders hold together by another cylinder in between. The first cylinder is used to place the mechanical and two buttons. The other cylinder is used to place the circuit of the design. At the tip of the first cylinder is a cap that used to hold the ball.

The design is extremely simple and natural to use. At the front of the mouse are two buttons, the right- and left-click buttons, are easily accessible with your index finger. The bottom button is the left-click button, and the top is the right-click button. The thumb and middle finger are closed together to grip and move the mouse.

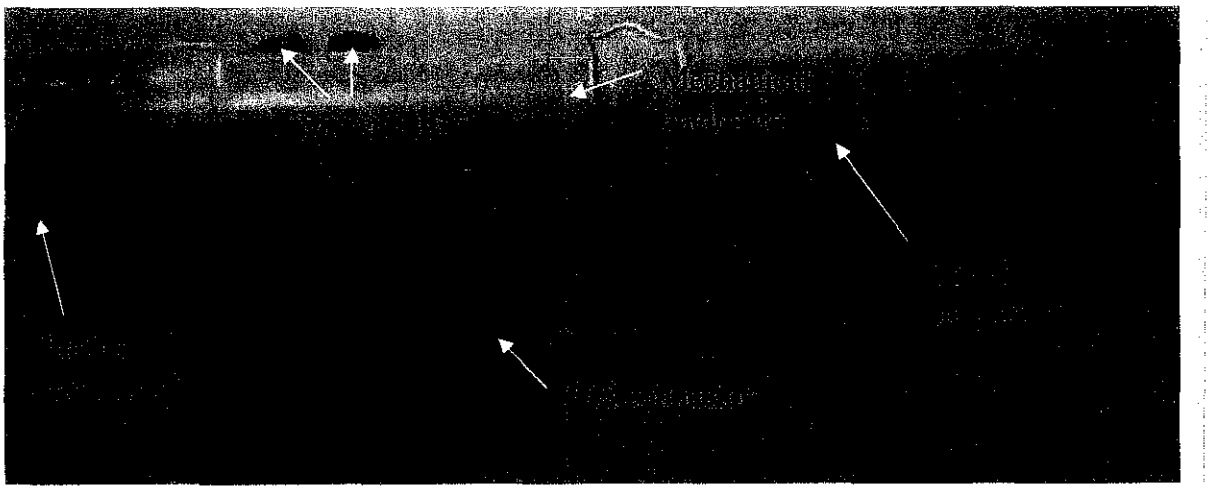


Figure 9: The pen shape mouse design final hardware

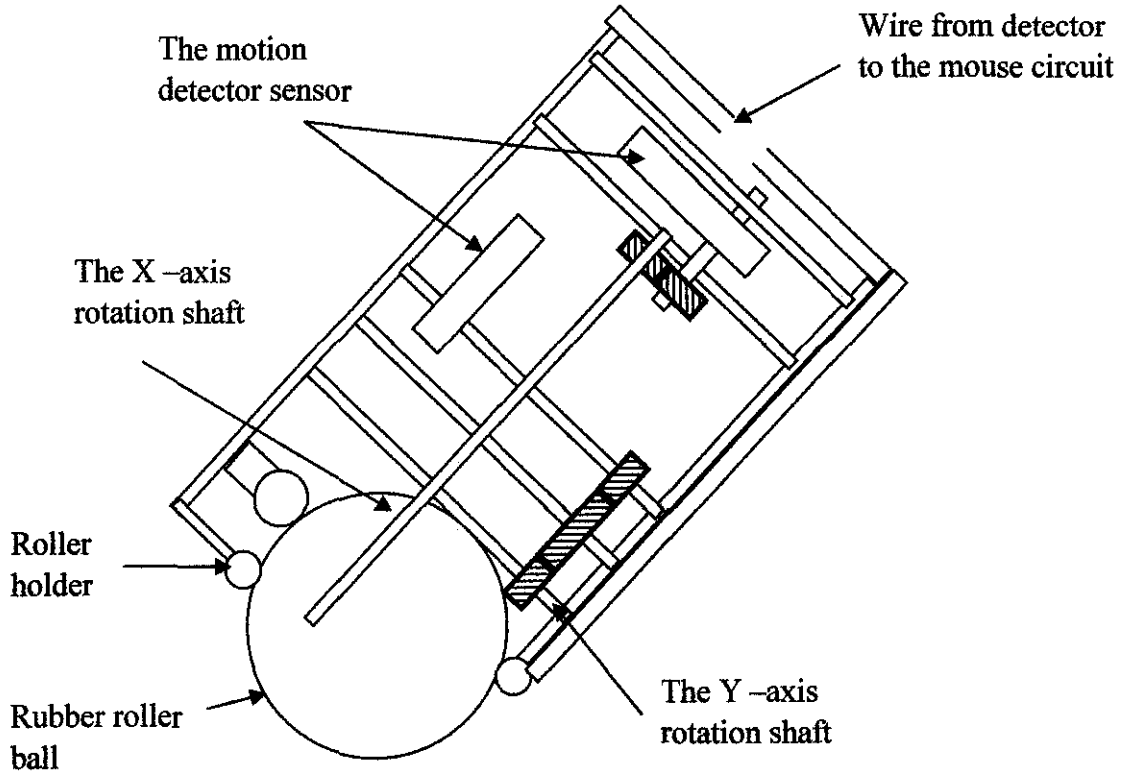


Figure 10: Pen Shape Mouse Design final drawing selection

The design mechanism has two shaft touching the rubber roller ball and position as in figure 10. The shafts will move the gearing systems which are connected to the motion detector sensor. The direction of up and down will have different signal 90 degree lag of each other. The same apply to the direction to right and left. This information is send to the circuit and sent to the computer in serial. The computer will then convert the binary instruction into motion to move the cursor on the screen.

4.2 Design Evaluation

4.2.1 Advantages

1. The Presentation Pen Mouse brings the feeling of using a pen or pencil, while still being as fully functional as a standard two-button mouse.
2. Since the mouse can be hold as though holding a pen (or paintbrush), the Presentation Pen Mouse works superbly in graphics applications.
3. The Presentation Pen Mouse doesn't add any clutter to your space and can be easily bring around for the traveler.

4.2.2 Disadvantages

1. The DPI resolution. It's too low. At 400 DPI, the mouse does very little to catch up with other optical mice that are twice as powerful.
2. There is no scroll wheel button in the design. But what this translates to is that it will take more effort to scroll down a page than with a regular mouse.

4.3 Design Specification

1. The design operates with a 400 DPI opto-mechanical and is PS2 compatible.
2. No device driver is required for it to be detected by the computer.
3. It is compact and weighs 100 grams, making it a nice companion for the person on the go.

4.4 PCB Board Fabrication

I have design the PCB circuit board for it to fits in the design casing. The design casing is cylinder in shape with inner diameter of 2 cm. Therefore the design PCB circuit board width is set to 1.9 cm. I have discussed with the PCB Lab technician about the fabrication process. I found out that the process will take at least one week. There are two processes in fabricating the circuit, which are the chemical process and the drilling process. Both of the processes require the circuit drill and Gerber files from the simulation software. I choose the drilling process for the fabrication of the board since my circuit is small and the process is much faster. I have just finish designing PCB board and board fabrication. Figure 9 shows the schematic diagram of the PCB circuit board that I design using Eagle PCBoards software.

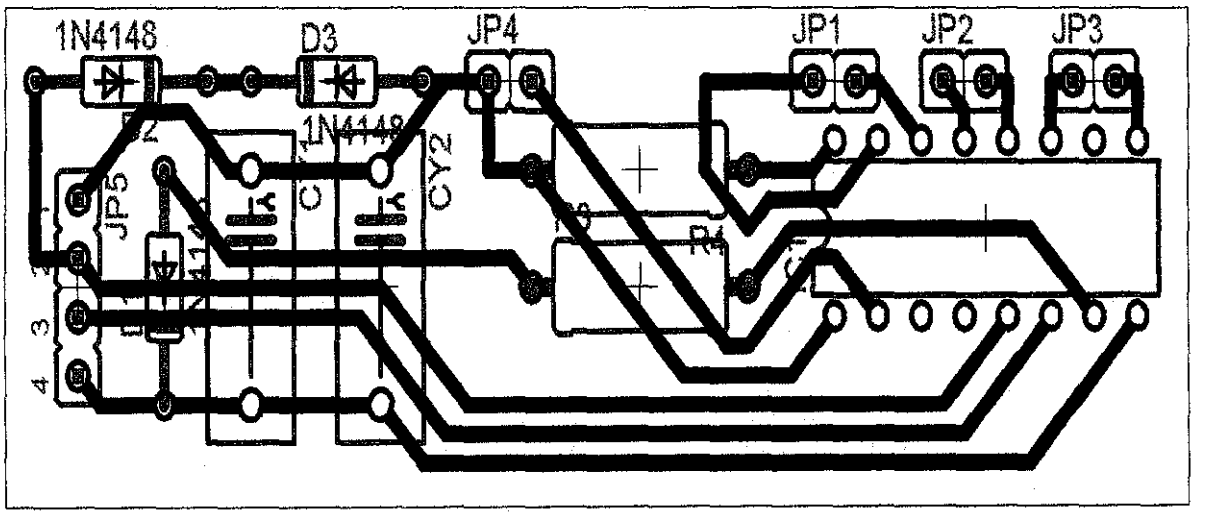


Figure 11: Schematic diagram of PCB circuit board

4.5 Mouse PIC Assembly Code

I have also work on the design assembly coding for the 16F84a microcontroller for my circuit. The reason I am pursuing this area is because I want to learn more on how the circuit interact with computer bit by bit and also to test my ability in programming. For this reason I have also built the microcontroller programmer circuit. Currently I am using the mouse design controller IC EM84510 as a backup alternative. There are several errors in the coding that I need to fix. If I am able to complete the coding in time I will design a new PCB board for the new controller.

CHAPTER 5
CONCLUSION AND RECOMMENDATION

5.1 Conclusion

I have done literature search that covers information on mouse including how the devices work, comparison of other types of mouse and how to use the pen option in Power Point as presentation tool. This information can help the reader to understand the project and its related problems in a better way. From the results that obtained, the project meets the objectives requirement. The Pen shape mouse device is able to move the cursor on the screen and perform the normal mouse function. However the design is still lacking in the motion sensitivity. Nonetheless the Pen mouse design proved to be very useful in enhancing PowerPoint presentation that allows natural writing just like using a real pen, lightweight and durable compared to other mouse type.

5.2 Recommendations

The design is very useful for power point presentation for precise writing and drawing purpose. However the design can be further improved to enhance it performance such as increase its DPI rating and wireless connection capability. The mouse design has only 400 dpi which limit the mouse capability detect writing motion perfectly. The connectivity of the mouse design to the computer can be improved by having wireless connection for better durability and more freedom when writing.

CHAPTER 6
REFERENCES

1. John Hyde, 1999 "USB Design by Example, A Practical Guide to Building I/O Devices"
2. Charles S. Parker, Deborah Morley, 2002 edition "Understanding Computers Today and Tomorrow"
3. Jenny Preece, 1994 "Human Computer Interaction" Pearson Education Limited
4. Chip (Computer and communication magazine), July 2005 edition, Pg 76 " Optical Mouse Technology" by Gui Jue Fok (SS)
5. Fitts' Group "Generalized Fitts' Law Model Builder" December 17, 1996, http://ei.cs.vt.edu/~cs5724/g1/tap.html
6. Wikipedia Free Encyclopedia, "Pointing device" April 24, 2000 http://en.wikipedia.org/wiki/Pointing_device
7. Marshall Brain, Carmen Carmack. Howstuffworks "How Computer Mice Work" http://www.howstuffworks.com/mouse.htm

APPENDICES

APPENDIX A

PEN SHAPE MOUSE DESIGNS

APPENDIX - I

PEN MOUSE DESIGN 1:

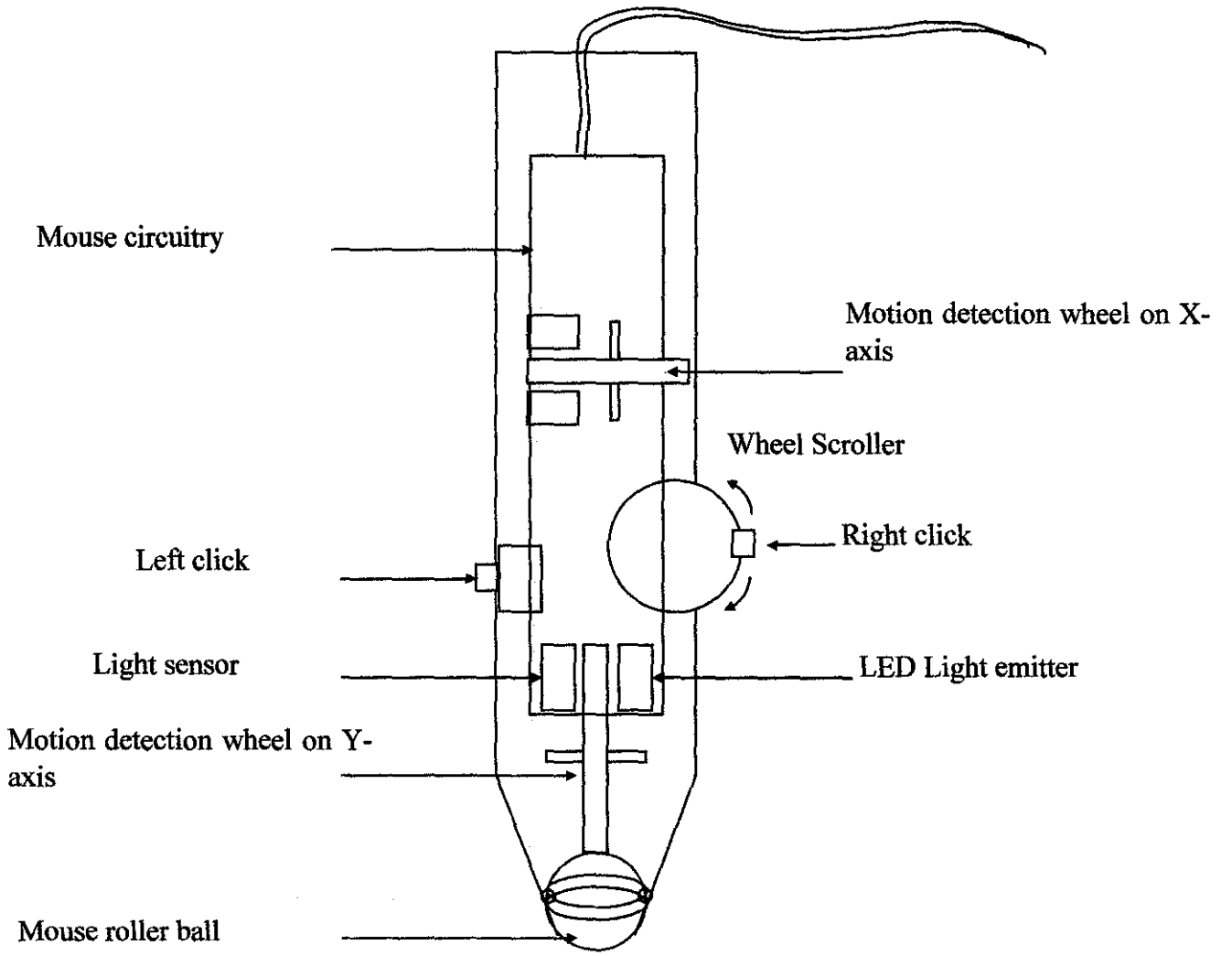


Figure 12: Pen mouse design 1 has unpractical shaft position that allows the ball to move in only one axis. The design is not selected to construct the pen shape mouse.

APPENDIX - II

PEN MOUSE DESIGN 2:

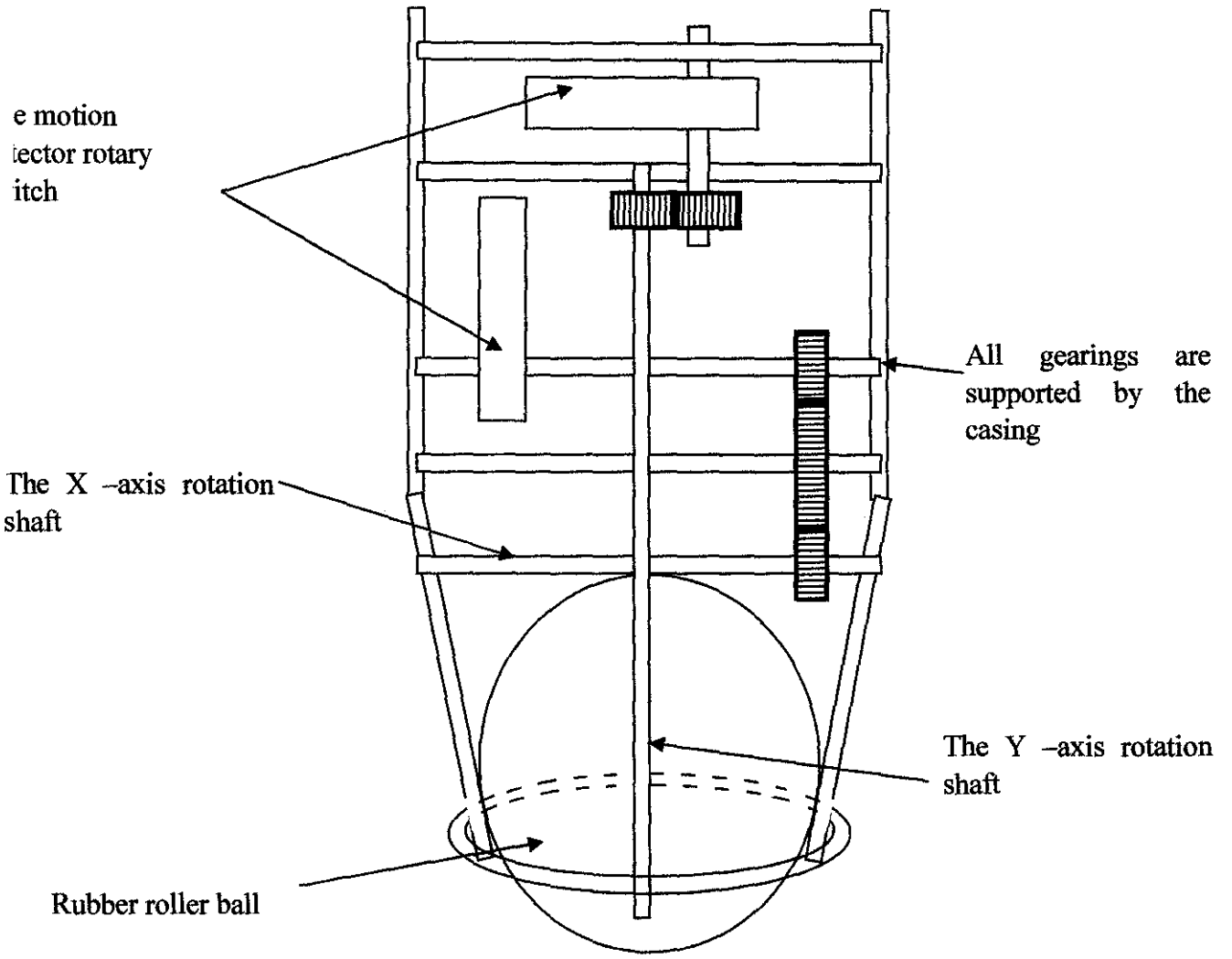


Figure 13: Pen Mouse design with mechanical gearing to move the encoder shaft for the photo sensitive to detect the waveform required for the computer to translate it in motion in x and y directions.

APPENDIX - III

PEN MOUSE DESIGN 3:

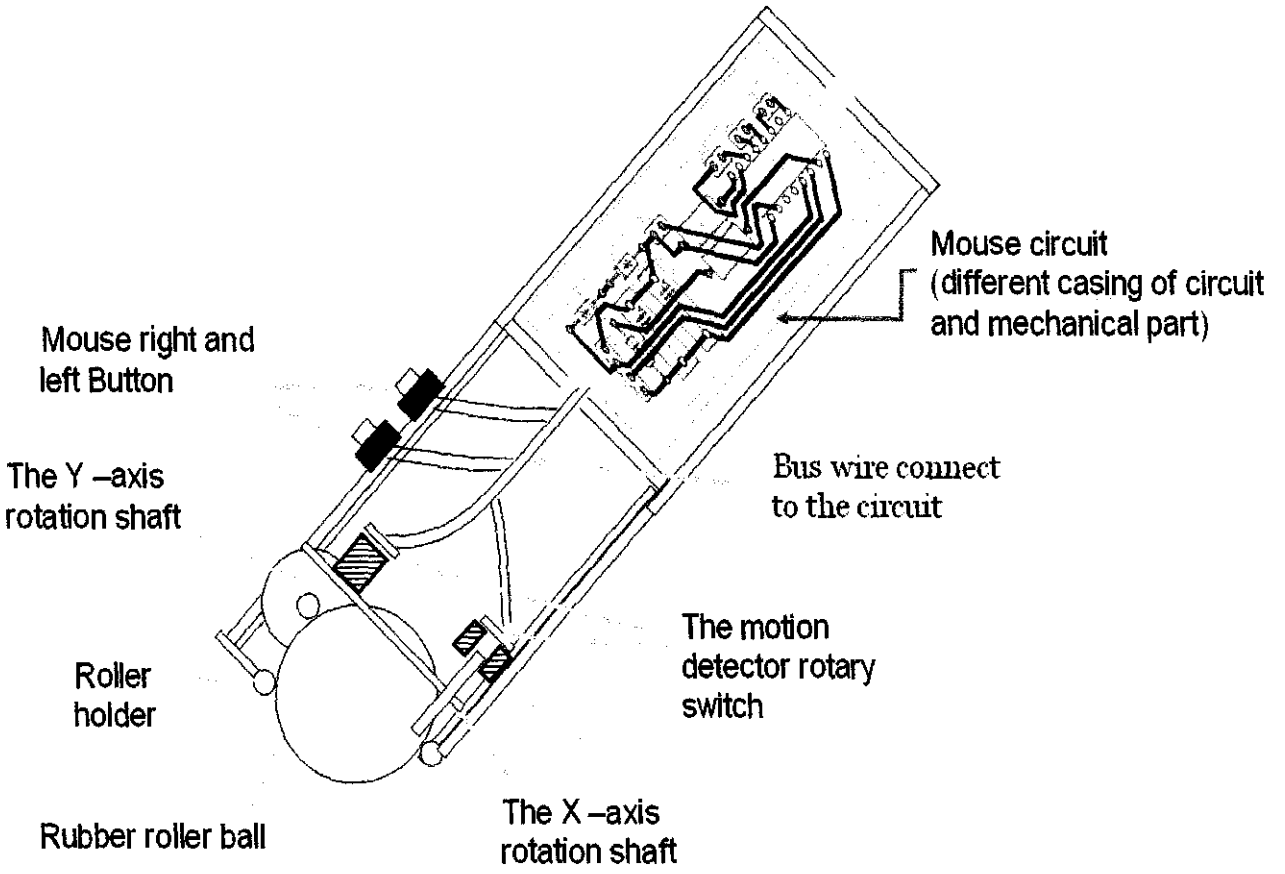


Figure 14: Improvement to Pen Mouse design 3 with roller holder. The project mouse model currently refers to this design. Due to its practical position of the shaft.

APPENDIX B

SUMMARY OF POWERPOINT SLIDE SHOW SHORTCUT

Table 2: Shortcut during the PowerPoint Slide show Presentation

Shortcut Key	Function
N, ENTER, PAGE DOWN, RIGHT ARROW, DOWN ARROW, or the SPACEBAR (or click the mouse)	Perform the next animation or advance to the next slide
P, PAGE UP, LEFT ARROW, UP ARROW, or BACKSPACE	Perform the previous animation or return to the previous slide
number+ENTER	Go to slide *number*
A or =	Show or hide arrow pointer
B or PERIOD	Display a black screen, or return to the slide show from a black screen
W or COMMA	Display a white screen, or return to the slide show from a white screen
S or PLUS SIGN	Stop or restart an automatic slide show
ESC, CTRL+BREAK, or HYPHEN	End a slide show
E	Erase on-screen annotations
H	Go to the next hidden slide
T	Set new timings while rehearsing

O	Use original timings while rehearsing
M	Use mouse-click to advance while rehearsing
1+ENTER (or press both mouse buttons for 2 seconds)	Return to the first slide
CTRL+P	Redisplay hidden pointer and/or change the pointer to a pen
CTRL+A	Redisplay hidden pointer and/or change the pointer to an arrow
CTRL+H	Hide the pointer and navigation button immediately
CTRL+M	Show or hide ink markup
CTRL+U	Hide the pointer and navigation button in 15 seconds
CTRL+T	View task bar
CTRL+S	**All Slides** dialog box
SHIFT+F10 (or right-click)	Display the shortcut menu
TAB	Go to the first or next hyperlink on a slide
SHIFT+TAB	Go to the last or previous hyperlink on a slide
ENTER while a hyperlink is selected	Perform the "mouse click" behavior of the selected hyperlink
SHIFT+ENTER while a hyperlink is selected	Perform the "mouse over" behavior of the selected hyperlink

APPENDIX C

PEN SHAPE MOUSE SCHEMATIC DIAGRAM

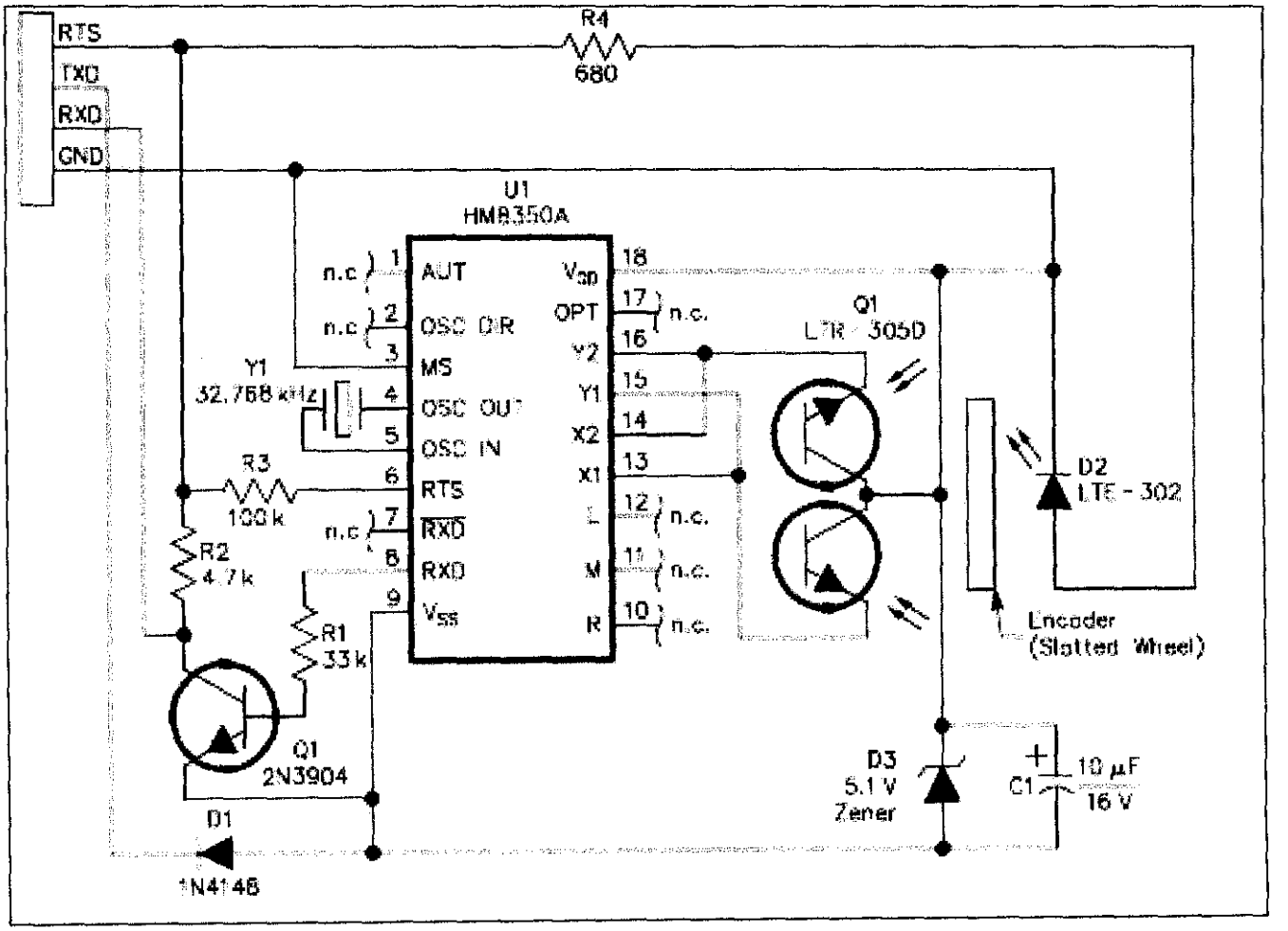


Figure 15: Schematic diagram of a conventional ball type mouse. The circuit consist of power input, input for X and Y axis motion signal, 3 button input (Left, Right and middle click), and three led source for generating the motion binary signal.

APPENDIX D

MICROCONTROLLER PROGRAMMER CIRCUIT SCHEMATIC DIAGRAM

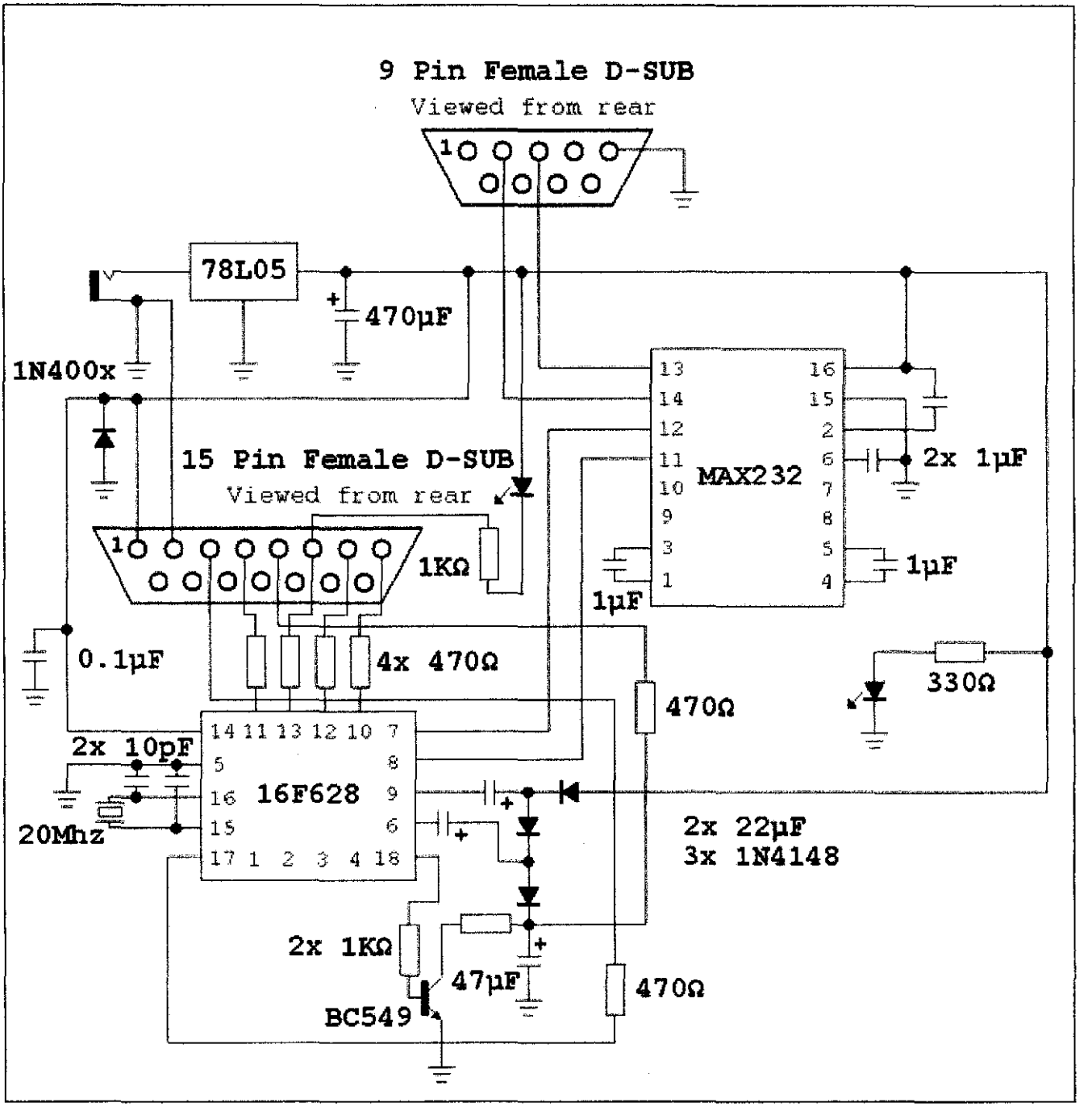


Figure 16: Microcontroller Programmer circuit schematic diagram. The circuit is use for programming the microprocessor controller for the pen shape mouse design circuit.

APPENDIX E

FLOW CHART OF MAIN MICROCONTROLLER PROGRAM

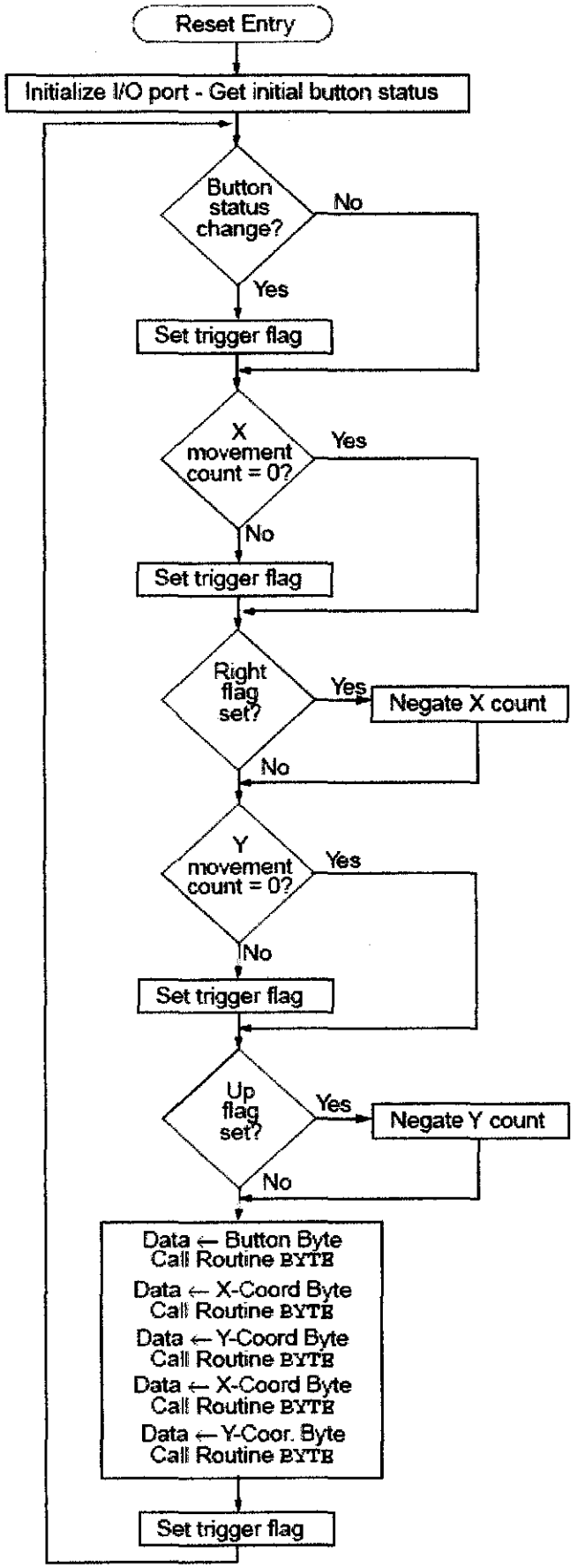


Figure 17: Flow Chart of Main Microcontroller program

APPENDIX F

MICROCONTROLLER ASSEMBLY CODE

```
            PROCESSOR 16f84a
            #include "p16f84a.inc"

          __CONFIG _CP_OFF & _WDT_OFF & _XT_OSC
;-----------------------------------------------------------------
; DEFINES:
;-----------------------------------------------------------------
          #DEFINE DATA PORTA, 2 ;May be assigned to any I/O pin
          #DEFINE CLOCK PORTA, 3 ;May be assigned to any I/O pin
          #DEFINE PS2_Yp PORTB, 0 ;May be assigned to any I/O pin
          #DEFINE PS2_Yn PORTB, 1 ;May be assigned to any I/O pin
          #DEFINE PS2_Xp PORTB, 2 ;May be assigned to any I/O pin
          #DEFINE PS2_Xn PORTB, 3 ;May be assigned to any I/O pin
          #DEFINE PS2_Bl PORTB, 4 ;May be assigned to any I/O pin
          #DEFINE PS2_Br PORTB, 5 ;May be assigned to any I/O pin
          #DEFINE PERIOD 20 ;Time between reading of inputs. Min=(osc
frequency)/204800
          #DEFINE DISTANCE 2 ;Amount by which X/Y counters are
incremented/decremented
;-----------------------------------------------------------------
; RAM ALLOCATION:
;-----------------------------------------------------------------
          cblock 0x0C
            TEMP0, TEMP1
            RECEIVE, PARITY, COUNTER ;Used in I/O routines
            REPORT_RATE, RESOLUTION ;Used for responses to status
requests
            FLAGS, XY_FLAGS
            dBUTTONS ;"Delta Button States"
            X_COUNTER
            Y_COUNTER
          endc
;-----------------------------------------------------------------
;FLAGS:
; bit 7 -- Always 0
; bit 6 -- Stream(0)/Remote(1) mode
; bit 5 -- Disable(0)/Enable(1) reporting
; bit 4 -- 1:1(0)/2:1(1) Scaling
; bit 3 -- Always 0
; bit 2 -- Always 0
; bit 1 -- Always 0
; bit 0 -- Always 0
MODE      equ 6
```

```
ENABLE    equ 5
SCALE     equ 4
;-------------------------------------------------------------------
; XY_FLAGS:
; bit 7 -- Y Counter overflow
; bit 6 -- X Counter overflow
; bit 5 -- Y counter sign bit
; bit 4 -- X counter sign bit
; bit 3 -- Always 1
; bit 2 -- Always 0 (middle button)
; bit 1 -- Previous right button state
; bit 0 -- Previous left button state
yOVF      equ 7
xOVF      equ 6
ySIGN     equ 5
xSIGN     equ 4
;dBUTTONS
; bit 7 -- Always 0
; bit 6 -- Always 0
; bit 5 -- Always 0
; bit 4 -- Always 0
; bit 3 -- Always 0
; bit 2 -- Always 0
; bit 1 -- Change in right buton state
; bit 0 -- Change in left button state
;-------------------------------------------------------------------
          cblock ;Contains to-be-sent packet and last packet sent
            LENGTH
            SEND1
            SEND2
            SEND3
          endc
;-------------------------------------------------------------------
; MACROS:
;-------------------------------------------------------------------
;Delay "Cycles" instruction cycles
Delay macro Time
            if (Time==1)
            nop
            exitm
            endif
```

```
  if (Time==2)
  goto $ + 1
  exitm
  endif
  if (Time==3)
  nop
  goto $ + 1
exitm
endif
if (Time==4)
  goto $ + 1
  goto $ + 1
exitm
endif
if (Time==5)
  goto $ + 1
  goto $ + 1
  nop
exitm
endif
if (Time==6)
  goto $ + 1
  goto $ + 1
  goto $ + 1
exitm
endif
if (Time==7)
  goto $ + 1
  goto $ + 1
  goto $ + 1
  nop
exitm
endif
if (Time%4==0)
movlw (Time-4)/4
call Delay_us
exitm
endif
if (Time%4==1)
  movlw (Time-5)/4
  call Delay_us
```

```
            goto $ + 1
            nop
            exitm
          endif
            endm
;-----------------------------------------------------------------------
; ORG 0x000:
;-----------------------------------------------------------------------
          org 0x000
          goto Start
;-----------------------------------------------------------------------
; HANDLE COMMAND:
;-----------------------------------------------------------------------
          if (high Table1End != 0)
          ERROR "Command handler table must be in low memory page"
          endif
Command
            movlw 0x04 ;Test for a resolution value
            subwf RECEIVE, w
            bnc SetResolution
            movlw 0xC8 ;Test for report rate value
            subwf RECEIVE, w
            bnc SetReportRate
            movlw 0xE6 ;0xE6 is lowest code
            subwf RECEIVE, w
            bnc MainLoop
HandlerTable
            addwf PCL, f ;Add offset
            goto Mouse_E6 ;0xE6 - Set Scaling 1:1
            goto Mouse_E7 ;0xE7 - Set Scaling 2:1
            goto MainLoop ;0xE8 - Set Resolution
            goto Mouse_E9 ;0xE9 - Status Request
            goto Mouse_EA ;0xEA - Set Stream Mode
            goto Report ;0xEB - Read Data
            goto MainLoop ;0xEC - Reset Wrap Mode
            goto MainLoop ;0xED -
            goto WrapMode ;0xEE - Set Wrap Mode
            goto MainLoop ;0xEF
            goto Mouse_F0 ;0xF0 - Set Remote Mode
            goto MainLoop ;0xF1
            goto Mouse_F2 ;0xF2 - Read Device Type
```

```
            goto MainLoop ;0xF3 - Set Report Rate
            goto Mouse_F4 ;0xF4 - Enable
            goto Mouse_F5 ;0xF5 - Disable
            goto Mouse_F6 ;0xF6 - Set Default
            goto MainLoop ;0xF7
            goto MainLoop ;0xF8
            goto MainLoop ;0xF9
            goto MainLoop ;0xFA
            goto MainLoop ;0xFB
            goto MainLoop ;0xFC
            goto MainLoop ;0xFD
            goto PacketOut ;0xFE - Resend
Table1End
            goto Reset ;0xFF - Reset
;-----------------------------------------------------------------
; START:
;-----------------------------------------------------------------
Start
            clrf PORTA
            clrf PORTB
            bsf STATUS, RP0 ;(TRISA=TRISB=0xFF by default)
            movlw 0x57 ;Timer mode, assign max. prescaler, enable
pullups
            movwf OPTION_REG
            bcf STATUS, RP0
            movlw 0x08 ;Bit 3 always = 1, clear previous button
states
            movwf XY_FLAGS
; goto Reset
;-----------------------------------------------------------------
; Reset Mode:
;-----------------------------------------------------------------
Reset
            movlw 0xAA
            movwf SEND1 ;Load BAT completion code
            call LoadDefaults
            clrf SEND2 ;Load Device ID (0x00)
            movlw 0x02
            movwf LENGTH
            call BATdelay
            goto PacketOut ;Output 2-byte "completion-code, device
ID" packet
```

```
;-------------------------------------------------------------------
; Stream/Remote Mode:
;-------------------------------------------------------------------
MainLoop
            clrf X_COUNTER ;Clear movement counters
            clrf Y_COUNTER
MainLoop1
            btfss DATA ;Check for host request-to-send
            goto PacketIn
            movlw PERIOD ;Report period
            subwf TMR0, w
            btfss STATUS, C ;TMR0=report period?
            goto MainLoop1 ; No--loop
            clrf TMR0 ; Yes--reset TMR0, then read inputs...
            call ReadInputs
            btfsc FLAGS, MODE ;Stream(0)/Remote(1) mode
            goto MainLoop1
            btfss FLAGS, ENABLE ;Disable(0)/Enable(1) reporting
            goto MainLoop1
            movf X_COUNTER, w ;Test for X-movement
            iorwf Y_COUNTER, w ;Test for Y-movement
            iorwf dBUTTONS, w ;Test for change in button states
            bz MainLoop1
            goto Report
;-------------------------------------------------------------------
; REPORT:
;-------------------------------------------------------------------
Report
            movf dBUTTONS, w
            xorwf XY_FLAGS, f ;Find current button state
            movf XY_FLAGS, w
            movwf SEND1
            movf X_COUNTER, w
            movwf SEND2
            movf Y_COUNTER, w
            movwf SEND3
            movlw 0x03 ;Movement data report length
            movwf LENGTH
            goto PacketOut
```

```
;-----------------------------------------------------------------
; OUTPUT PACKET
;-----------------------------------------------------------------
PacketOut
            movlw SEND1 ;First byte of packet
            movwf FSR
            movf LENGTH, w ;Length of packet
            movwf TEMP1
PacketOutLoop
            movf INDF, w ;Get data byte
            call ByteOut ; Output that byte
            xorlw 0xFF ;Test for RTS error
            bz PacketIn
            xorlw 0xFE ^ 0xFF ;Test for inhibit error
            bz PacketOut
            incf FSR, f ;Point to next byte
            decfsz TEMP1, f
            goto PacketOutLoop
            goto MainLoop
;-----------------------------------------------------------------
; READ PACKET
;-----------------------------------------------------------------
PacketIn
            call ByteIn
            xorlw 0xFF ;Test for parity/framing error
            bz Mouse_ERR
            xorlw 0xFE ^ 0xFF ;Test for inhibit error
            bz MainLoop1
            movlw 0xFE ;Test for "Resend" command
            xorwf RECEIVE, w
            bz PacketOut
Acknowledge
            movlw 0xFA ;Acknowledge
            call ByteOut
            goto Command
;-----------------------------------------------------------------
; READ INPUTS:
;-----------------------------------------------------------------
ReadInputs
            movlw DISTANCE
            btfss PS2_Xp ;Read inputs
            addwf X_COUNTER, f
```

```
            btfss PS2_Yp
            addwf Y_COUNTER, f
            btfss PS2_Xn
            subwf X_COUNTER, f
            btfss PS2_Yn
            subwf Y_COUNTER, f
            bcf XY_FLAGS, xSIGN
            btfsc X_COUNTER, 7
            bsf XY_FLAGS, xSIGN
            bcf XY_FLAGS, ySIGN
            btfsc Y_COUNTER, 7
            bsf XY_FLAGS, ySIGN
            movf XY_FLAGS, w ;Get previous button states
            andlw b'00000111'
            btfss PS2_Bl ;Find changes in button states
            xorlw b'00000001'
            btfss PS2_Br
            xorlw b'00000010'
            movwf dBUTTONS ;Save *change* in button state
            retlw 0x00
;-----------------------------------------------------------------------
; WRAP MODE:
;-----------------------------------------------------------------------
WrapMode
            btfsc DATA ;Wait for RTS
            goto WrapMode
            call ByteIn ;Read one byte from host
            xorlw 0xFE ;Test for aborted transmission
            bz WrapMode
            movf RECEIVE, w
            xorlw 0xFF ;Test for "Reset" command
            bz Acknowledge
            xorlw 0xFF^0xEC ;Test for "Reset Wrap Mode" command
            bz Acknowledge
            xorlw 0xEC
            call ByteOut ;Else, echo
            goto WrapMode
;-----------------------------------------------------------------------
; LOAD DEFAULT VALUES:
;-----------------------------------------------------------------------
LoadDefaults
```

```
            movlw 100 ;Default report rate
            movwf REPORT_RATE
            movlw 0x02 ;Default resolution
            movwf RESOLUTION
            clrf FLAGS ;Stream mode, 1:1 scaling, disabled
            retlw 0x00
;-----------------------------------------------------------------
; EMULATE BAT:
;-----------------------------------------------------------------
BATdelay
            clrf TEMP0 ;Used for a 400 ms delay at power-on
            clrf TEMP1
DelayLoop
            Delay 6
            decfsz TEMP0, f
            goto DelayLoop
            decfsz TEMP1, f
            goto DelayLoop
            retlw 0x00
;-----------------------------------------------------------------
; HANDLE COMMANDS:
;-----------------------------------------------------------------
SetResolution
            movf RECEIVE, w
            movwf RESOLUTION
            goto MainLoop
SetReportRate
            movf RECEIVE, w
            movwf REPORT_RATE
            goto MainLoop
;0xE6 - Set Scaling 1:1
Mouse_E6
            bcf FLAGS, SCALE
            goto MainLoop
;0xE7 - Set Scaling 2:1
Mouse_E7
            bsf FLAGS, SCALE
            goto MainLoop
;0xE9 - Status Request
Mouse_E9
            movf FLAGS, w
```

```
            btfss PS2_Bl
            iorlw 0x04
            btfss PS2_Br
            iorlw 0x01
            movwf SEND1
            movf RESOLUTION, w
            movwf SEND2
            movf REPORT_RATE, w
            movwf SEND3
            movlw 0x03
            movwf LENGTH
            goto PacketOut
;0xEA - Set Stream Mode
Mouse_EA
            bcf FLAGS, MODE
            goto MainLoop
;0xF0 - Set Remote Mode
Mouse_F0
            bsf FLAGS, MODE
            goto MainLoop
;0xF2 - Get Device ID
Mouse_F2
            clrf SEND1
            movlw 0x01
            movwf LENGTH
            goto PacketOut
;0xF4 - Enable Reporting
Mouse_F4
            bsf FLAGS, ENABLE
            goto MainLoop
;0xF5 - Disable Reporting
Mouse_F5
            bcf FLAGS, ENABLE
            goto MainLoop
;0xF6 - Set Default
Mouse_F6
            call LoadDefaults
            goto MainLoop
;Invalid command
Mouse_ERR
            movlw 0xFE
```

```
            call ByteOut
            goto MainLoop
;-----------------------------------------------------------------
; OUTPUT ONE BYTE:
;-----------------------------------------------------------------
ByteOut
            movwf TEMP0
InhibitLoop
            btfss CLOCK ;Test for inhibit
            goto InhibitLoop
            Delay 100 ;(50 microsec = 58 clock cycles, min)
            btfss CLOCK
            goto InhibitLoop
            btfss DATA ;Check for request-to-send
            retlw 0xFF
            clrf PARITY
            movlw 0x08
            movwf COUNTER
            movlw 0x00
            call BitOut ;Start bit (0)
            btfss CLOCK ;Test for inhibit
            goto ByteOutEnd
            Delay 4
ByteOutLoop
            movf TEMP0, w
            xorwf PARITY, f
            call BitOut ;Data bits
            btfss CLOCK ;Test for inhibit
            goto ByteOutEnd
            rrf TEMP0, f
            decfsz COUNTER, f
            goto ByteOutLoop
            Delay 2
            comf PARITY, w
            call BitOut ;Parity bit
            btfss CLOCK ;Test for inhibit
            goto ByteOutEnd
            Delay 5
            movlw 0xFF
            call BitOut ;Stop bit (1)
            Delay 48
```

```
            retlw 0x00
ByteOutEnd
            bsf STATUS, RP0
            bsf DATA
            bsf CLOCK
            bcf STATUS, RP0
            retlw 0xFE
BitOut
            bsf STATUS, RP0
            andlw 0x01
            btfss STATUS, Z
            bsf DATA
            btfsc STATUS, Z
            bcf DATA
            Delay 21
            bcf CLOCK
            Delay 45
            bsf CLOCK
            bcf STATUS, RP0
            Delay 5
            return
;-------------------------------------------------------------
; READ ONE BYTE: (Takes about 1ms)
;-------------------------------------------------------------
ByteIn
            btfss CLOCK ;Test for Request-to-send
            retlw 0xFE
            btfsc DATA
            retlw 0xFE
            movlw 0x08
            movwf COUNTER
            clrf PARITY
            Delay 28
ByteInLoop
            call BitIn ;Data bits
            btfss CLOCK ;Test for inhibit
            retlw 0xFE
            bcf STATUS, C
            rrf RECEIVE, f
            iorwf RECEIVE, f
            xorwf PARITY,f
```

```
            decfsz COUNTER, f
            goto ByteInLoop
            Delay 1
            call BitIn ;Parity bit
            btfss CLOCK ;Test for inhibit
            retlw 0xFE
            xorwf PARITY, f
            Delay 5
ByteInLoop1
            Delay 1
            call BitIn ;Stop bit
            btfss CLOCK ;Test for inhibit
            retlw 0xFE
            xorlw 0x00
            btfsc STATUS, Z
            clrf PARITY
            btfsc STATUS, Z ;Stop bit = 1?
            goto ByteInLoop1 ; No--keep clocking.
            bsf STATUS, RP0 ;Acknowledge
            bcf DATA
            Delay 11
            bcf CLOCK
            Delay 45
            bsf CLOCK
            Delay 7
            bsf DATA
            bcf STATUS, RP0
            btfss PARITY, 7 ;Parity correct?
            retlw 0xFF ; No--return error
            Delay 45
            retlw 0x00
BitIn
            Delay 8
            bsf STATUS, RP0
            bcf CLOCK
            Delay 45
            bsf CLOCK
            bcf STATUS, RP0
            Delay 21
            btfsc DATA
            retlw 0x80
```

```
            retlw 0x00
;-------------------------------------------------------------------------
; DELAY:
;-------------------------------------------------------------------------
;Delays 4w+4 cycles (including call,return, and movlw)
;(0=256)
            Delay_us
            addlw 1                 ;Precise delays used in I/O
            btfss STATUS, Z
            goto Delay_us
            return

            end
```

APPENDIX G
PROJECT GANTT CHART FYP I

Table 3: Project Gantt Chart for Semester 1 of Pen Shape Mouse Design

No.	Detail/week	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16
1	**Selection of Project Topic**																
2	**Preliminary Research Work**				18/8												
3	**Project Work**																
3.1	-Design draft/application																
3.2	-Circuit design																
3.3	-Mechanical assembly																
4	**Progress Report**								22/9								
5	**Project Work**																
5.1	-Mechanical assembly																
5.2	-Troubleshooting																
6	**Interim Report Final Draft**											16/10					
7	**Oral Presentation**															6/11	8/11
8	**Submission of Interim Report**													31/10			

APPENDIX H

PROJECT GANTT CHART FYP II

Table 4: Project Gantt Chart for Semester 2 of Pen Shape Mouse Design

No.	Detail/week	W1	W2	W3	W4	W5	W6	W7	W8	W9	W10	W11	W12	W13	W14	W15	W16	W17	W18
1	**Project Work**																		
1.1	-Design construction																		
1.2	- PIC coding test																		
2	**Progress Report 1**																		
3	**Project Work**																		
3.1	-Mechanical assembly																		
3.2	-Troubleshooting																		
4	**Progress Report 2**																		
5	**Design evaluation**																		
6	**Draft Report**																		
7	**Final Report (Soft Cover)**																		
8	**Technical Report**																		
9	**Oral Presentations**																		
10	**Final Report (Hard Cover)**																		22/6