จำลองสถานการณ์จริง
VIRTUAL REALITY

โดย

นายกิตติ เปรมพินิจ

นายพงศ์เชษฐ์ โพธิ์ชัย

นายเอกชัย ฝางแก้ว

อาจารย์ที่ปรึกษา

รศ.ดร.มนัส สังวรศิลป์

วัน เดือน ปี -1 ต.ค. 2541
เลขทะเบียน 038391
เลขเรียกหนังสือ T30411 ก167ป

ปริญญานิพนธ์สำหรับปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาอิเลคทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2539

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ปริญญานิพนธ์ปีการศึกษา 2539

ภาควิชาวิศวกรรมอิเลคทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง จำลองสถานการ์จริง

ผู้จัดทำ

1.นายกิตติ	เปรมพินิจ	36014029
2.นายพงศ์เชษฐ์	โพธิ์ชัย	36014274
3.นายเอกชัย	ฝางแก้ว	36014580

(รศ.ดร.มนัส สังวรศิลป์)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จำลองสถานการณ์จริง

VIRTUAL REALITY

1.นายกิตติ เปรมพินิจ 36014029

2.นายพงศ์เชษฐ์ โพธิ์ชัย 36014274

3.นายเอกชัย ฝ่างแก้ว 36014580

โครงงานได้รับการตรวจสอบแน่นอนแล้ว พร้อมที่จะทำการสอบได้

(รศ.ดร.มนัส สังวรศิลป์)

อาจารย์ที่ปรึกษา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VIRTUAL REALITY

Kitti Prempinij
Pongchet Pochai
Ekachai Fangkaew
Dr.Manas Sangworasilp Advisor
1996

ABTRACT

The purpose of this project is to illustrate interfacte between human and computer occure in the realtime.The project provides two components : hardware and software.Hardware is a design of a glove consists of sensor to detect signal from axis index and send data to a computer via interfacing device.Computer will process data from sensor order to display graphic of moving glove and able to save that data.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบาญ

เรื่อง	หน้า
บทคัดย่อ	
ABSTRACT	
บทนำ	1
บทที่ 1 ส่วนประกอบของระบบเวอร์ชวลเรียลริตี้	6
1.1 ส่วนประกอบของระบบ	6
1.1.1 ส่วนสร้างผล	6
1.1.2 เครื่องประมวลผล	7
1.1.3 โปรแกรมประยุกต์	7
1.1.4 ส่วนเก็บข้อมูล	7
1.2 เทคโนโลยีของอุปกรณ์ที่ใช้ในระบบเวอชวลเรียลลิตี้	7
1.2.1 อุปกรณ์อินพุท	7
1.2.2 อุปรณ์ไบโอลอจิก	9
1.2.3 อุปกรณ์เอาท์พุท	10
บทที่ 2 การวัดระยะด้วยอัลตราโซนิค	12
2.1 การวัดระยะด้วยอัลตราโซนิค	12
2.2 วงจรส่งคลื่นอัลตราโซนิค	16
2.3 วงจรรับคลื่นอัลตราโซนิค	17
2.4 วงจรนับฐานสองขนาด 16 บิท	18
2.5 วงจรอินเทอร์เฟซกับคอมพิวเตอร์	19
บทที่ 3 อุปกรณ์ในโครงงานเวอร์ชวลเรียลริตี้	23
3.1 ไจโรสโคป	23
3.1.1โครงสร้างของไจโรสโคป	23
3.1.2 การทำงานของอุปกรณ์อิเลคทรอนิกส์ของไจโรสโคป	24
3.2 การส่งข้อมูลเข้าไปยังคอมพิวเตอร์	25
3.3 เมาส์	26
3.3.1 ความละเอียด	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

3.4 คีย์บอร์ดและเมาส์ 31
3.4.1 การสื่อสารระหว่างคีย์บอร์ดและคอมพิวเตอร์ 32
3.4.2 การสแกนโค้ดของคีย์บอร์ด 33
3.4.3 การเขียนโปรแกรมกับคีย์บอร์ดผ่านพอร์ตคอมพิวเตอร์โดยตรง 34
3.5 การทำงานของ PS/2 เมาส์ 35
3.5.1 การเขียนโปรแกรม PS/2 เมาส์ผ่านพอร์ต 35
บทที่ 4 การเขียนโปรแกรมสร้างภาพ 3 มิติโดยใช้โอเพนจีแอล 36
4.1 ลักษณะโดยทั่วไปของโอเพนจีแอล 36
4.2 หลักการสร้างภาพสามมิติเบื้องต้น 37
4.3 ระบบแกนและพิกัดในระบบสามมิติ 38
4.4 โอเพนจีแอลกับการประยุกต์ใช้งาน 40
บทที่ 5 โครงสร้างของโครงงาน 41
5.1 โครงสร้างโดยรวม 41
5.1.1 เซ็นเซอร์ 41
5.1.2 คอมพิวเตอร์กราฟฟิค 43
5.1.3 การควบคุมแขนกล 43
5.1.4 วงจรไดร์มอเตอร์ 44
5.1.5 โครงสร้างของแขนกล 45
บทที่ 6 การทดลองและผลการทดลอง 46
6.1 ผลการทดลองในส่วนของโปรแกรม 46
6.2 ผลการทดลองในส่วนของฮาร์ดแวร์ 47
บทที่ 7 บทสรุป 48
ภาคผนวก
กิตติกรรมประกาศ
หนังสืออ้างอิง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

สารบัญภาพ

รูปที่	หน้า
รูปที่ 1.1 แสดงส่วนประกอบของระบบ	6
รูปที่ 1.2 แสดงโครงสร้างของถุงมือ	7
รูปที่ 1.3 แสดงโครงสร้างที่ใช้ระบบกลไก	8
รูปที่ 1.4 แสดงถุงมือที่ใช้ระบบอัลตราโซนิค	8
รูปที่ 1.5 แสดงระบบตรวจการเคลื่อนที่โดยใช้สนามแม่เหล็ก	9
รูปที่ 1.6 แสดงจอภาพแบบสวมศีรษะ	10
รูปที่ 2.1 แสดงการวัดระยะด้วยอัลตราโซนิค	12
รูปที่ 2.2 แสดงตำแหน่งของตัวรับอัลตราโซนิคทั้งสาม	13
รูปที่ 2.3 แสดงบล็อกไดอะแกรมของโครงงาน	15
รูปที่ 2.4 แสดงวงจรส่งคลื่นอัลตราโซนิค	16
รูปที่ 2.5 แสดงวงจรรับคลื่นอัลตราโซนิค	17
รูปที่ 2.6 แสดงวงรนับเลขฐานสองขนาด 16 บิท	18
รูปที่ 2.7 แสดงขาต่างๆใน ISA SLOT	19
รูปที่ 2.8 แสดงวงจรอินเตอร์เฟซกับเครื่องคอมพิวเตอร์ผ่านบัสของคอมพิวเตอร์	20
รูปที่ 2.9 แสดงรูปวงจรสมบูรณ์ของโครงงานที่สร้าง	22
รูปที่ 3.1 แสดงภาพของไจโรสโคปขณะอยู่นิ่งและขณะหมุน	23
รูปที่ 3.2 แสดงวงจรใช้งานพื้นฐานจากคู่มือของไจโรสโคป	24
รูปที่ 3.3 แสดงถึงความแตกต่างของโวลท์เตจเพื่อนำค่าดังกล่าวไปคำนวณหาทิศทาง	25
รูปที่ 3.4 แสดงโครงสร้างของเมาส์ที่ใช้ LED และโฟโตทรานซิสเตอร์	26
รูปที่ 3.5 แสดงค่าในบิตต่างๆของเฟรม	26
รูปที่ 3.6 แสดงการส่งข้อมูลแบบอนุกรม	27
รูปที่ 3.7 แสดงเมาส์ที่ใช้แสง	27
รูปที่ 3.8 แสดงวงจรของเมาส์	28
รูปที่ 3.9 แสดงโฟลว์ชาร์ตของโปรแกรมการอินเทอร์รับท์ของเมาส์	29
รูปที่ 3.10 แสดงรูทีนของการกดปุ่มเมาส์	30
รูปที่ 3.11 แสดงสัญญาณที่เกิดใน LED เมื่อมีการเคลื่อนที่ของเมาส์	31
รูปที่ 3.12 แสดงโครงสร้างของคีย์บอร์ดและสายสัญญาณ	32
รูปที่ 3.13 แสดงโค้ดของรหัสในคีย์บอร์ด	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 3.14 แสดงภาพของคีย์บอร์ดคอนโทรลเลอร์ 34
รูปที่ 3.15 แสดงเฟรมข้อมูลของ PS/2 เมาส์หน้า 692 35
รูปที่ 3.16 แสดงรีจิสเตอร์ควบคุม 35
รูปที่ 4.1 แสดงภาพสามมิติที่สร้างโดยโปรแกรมภาษาซีโดยอาศัยโอเพนจีแอล 36
รูปที่ 4.2 แสดงการสร้างภาพสามมิติบนจอภาพคอมพิวเตอร์ 37
รูปที่ 4.3 แสดงระบบแกนในการสร้างภาพกราฟิคบนวินโดว์ในระบบ 2 มิติ 38
รูปที่ 4.4 แสดงระบบแกนในการสร้างภาพสามมิติโดยอาศัยโอเพนจีแอล 38
รูปที่ 4.5 แสดงการทำโคออดิเนตคลิป 39
รูปที่ 4.6 แสดงการโปรเจคชั่นในแบบออโธกราฟิคและแบบเปอร์สเปคทีฟ 39
รูปที่4.7 แสดงการประยุกต์ใช้งานโอเพนจีแอลกับระบบเวอชวลเรียลลิตี 40
รูปที่ 5.1 โครงสร้างโดยรวมของโครงงาน 41
รูปที่ 5.2 แสดงการติดต่อของเซ็นเซอร์ทั้งสองจุดเชื่อมกับพอร์ตในคอมพิวเตอร์ 41
รูปที่ 5.3 แสดงถุงมือที่ติดเซนเซอร์ไว้แล้ว 42
รูปที่ 5.4ถุงมือและอุปกรณ์ที่ใช้เป็นอินพุทให้กับคอมพิวเตอร์ 42
รูปที่ 5.5 แสดงกราฟฟิคที่สร้างเพื่อจำลองการเคลื่อนที่โดยคอมพิวเตอร์ 43
รูปที่ 5.6 แสดงวงจรการ์ดอินเตอร์เฟสที่สร้างขึ้น 44
รูปที่ 5.7 บล๊อกการทำงานของแขนกลที่ควบคุมโดยคอมพิวเตอร์ 44
รูปที่ 5.8 วงจรการต่อวงจรการขับสเตปเปอร์มอเตอร์โดยใช้ทรานซิสเตอร
ขา D0-D3 ต่อกับการ์ดอินเทอร์เฟส 44
รูปที่ 5.9 วงจรขับสเตปเปอร์มอเตอร์โดยใช้มอสเฟส 45
รูปที่ 5.10 แขนกลอย่างง่ายๆที่มีจุดต่อเพียงสองจุด 45
รูปที่ 6.1 แสดงผลการทดลองที่ได้จากการเขียนโปรแกรม 46
รูปที่ 6.2 แสดงแขนกลที่สร้างขึ้น 47

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

บทนำ
ระบบคอมพิวเตอร์สำหรับการจำลองเหตุการณ์จริง

คำจำกัดความ (Definition)

คำว่า เวอร์ชวลเรียลริตี้(Virtual Reality)สามารถให้คำจำกัดความได้คือสภาพแวดล้อมในลักษณะสามมิติที่สร้างโดยคอมพิวเตอร์ในระบบเวลาจริง (Real-Time) เพื่อจำลองพฤติกรรมของผู้ใช้ตามความเป็นจริง สามารถขยายความได้คือ

เวอร์ชวลเรียลริตี้เป็นการจำลองการทำงานของมนุษย์ออกมาในรูปของคอมพิวเตอร์กราฟฟิกสามมิติ คอมพิวเตอร์จะจำลองสภาพแวดล้อมในระบบเวลาจริงที่เกิดกับอากัปกิริยาของมนุษย์

เวอร์ชวลเรียลริตี้เป็นระบบ 3มิติ ในการเชื่อมโยงกับคอมพิวเตอร์ และเวอร์ชวลเรียลริตี้เดิมทีระบบเดิมคือแมคอินทอชและวินโดว์เป็นระบบปฏิบัติการที่เป็นสองมิติและดอส,ยูนิกซ์เป็นมิติเดียว

เวอร์ชวลคือการสร้างสรรค์ทางคอมพิวเตอร์เพื่อตอบสนองต่อปฏิกิริยาของมนุษย์เช่นการจำลองการต่อสู้และในเกมส์อาเขต

เวอร์ชวลคือการจำลองสภาพแวดล้อมจริงหรืออาจจะเป็นการสร้างสถานการณ์ขึ้นมาเองในระบบคอมพิวเตอร์ปัจจุบันผู้พัฒนาพยายามทำระบบให้ง่ายต่อผู้ใช้เพื่อความสะดวกและง่ายต่อการใช้งานอ่อนไหวและซับซ้อน

การที่ระบบคอมพิวเตอร์สามารถแสดงการเคลื่อนไหวตามอากัปกิริยาของมนุษย์ทำให้มนุษย์ลดค่าใช้จ่ายลดเวลาลดการสูญเสียไปได้มาก เช่นใช้ในวงการทหาร แพทย์ และวิศวกรรม

ในปัจจุบันเวอร์ชวลเรียลริตี้ถูกแสดงด้วยเวลาจริงของผู้ใช้เวอร์ชวลเอ็นไวรอนเมนท์(Virsual Enviroment)ไม่ได้ประกอบด้วยตำแหน่งของวิว(view)ที่จัดเป็นลำดับๆ อย่างเช่นฟิล์มหรือวีดีโอมันไม่ใช่การสร้างสิ่งที่อยู่อยู่กับที่อย่างเช่น การการวาดภาพด้วยคอมพิวเตอร์แต่ผู้ใช้จะสามารถเคลื่อนมันดูได้รอบๆ เหมือนลูกโลก และการจำลองถูกสร้างให้สัมพันธ์กับการเคลื่อนไหวของผู้ใช้แสดงผลด้วยเวลาจริงหมายความว่าการแสดงแบบจำลองจะสัมพันธ์กับท่าทางของผู้ใช้โดยจะล้าช้ากว่าเพียงน้อยนิด แบบจำลองนี้สามารถเคลื่อนที่ไปข้างหน้า,หลัง,ซ้าย,ขวา,บน,ล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่1 ผู้ใช้ระบบเวอชวลเรียลลิตี้กำลังโต้ตอบกับระบบเวอชวลเอนไวรอนเมนท์

ต้นกำเนิดของเวอร์ชวลเรียลริตี้

ความเป็นมาของเวอร์ชวลเรียลริตี้เกิดมาจากมนุษย์ออกแบบที่จะสำรวจความจริงที่แตกต่างกัน ความเจริญก้าวหน้าทางคอมพิวเตอร์มีความสะดวกมากในการตอบสนองจินตนาการ

การจำลองการบิน, ทางการเคลื่อนไหวของแบบจำลองการบินและผลิตสิ่งที่แสดงภาพและเสียงในการทำให้มนุษย์คิดว่าเป็นของจริง จุดประสงค์เหล่านี้เพื่อฝึกหัดนักบินและผู้บังคับยานพาหนะต่างๆ สำหรับตัวอย่างบุคลากร 100 คน ถูกฝึกการรบท่ามกลางพายุทะเลทรายโดยการใช้แบบจำลอง

รูปที่ 2 แสดงการใช้ระบบเวอชวลเรียลลิตี้ในการสร้างแบบจำลองฝึกหัดการบิน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

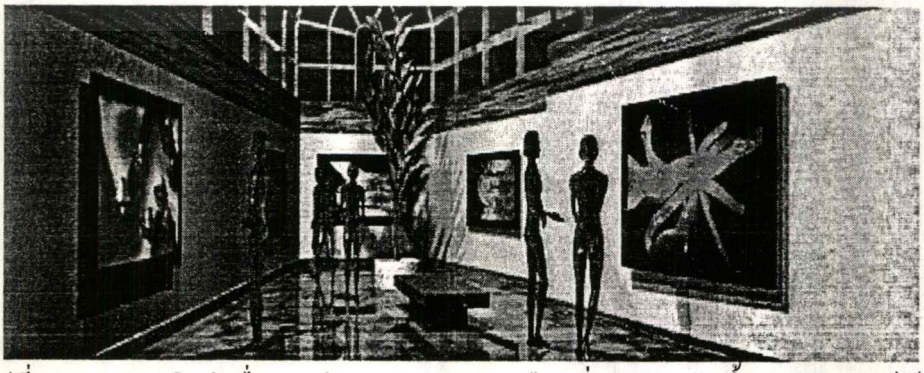
เวอร์ชวลเรียลริตี้เริ่มมีความสำคัญในปีที่ผ่านมานี้ เมื่อคนส่วนมากให้ความสำคัญกับการพัฒนาคอมพิวเตอร์ทำให้ความก้าวหน้าทางคอมพิวเตอร์เติบโตขึ้น การพัฒนากราฟฟิกใหม่ๆและอุปกรณ์อินพุท,เอาท์พุทที่เปลี่ยนไป ระบบของคอมพิวเตอร์ตอนนี้สามารถที่จะทำการรับการเคลื่อนที่แบบ 3 มิติ ของเมาส์,จอยสติกค์ ได้พร้อมกันหลายๆ อัน อุปกรณ์แสดงผลตอนนี้ได้มี เฮดเมาส์ดิสเพลย์(Head Mount Display) , 3D ออดิโอหรือ 3D เซาวน์ซึ่งแสดงทิศทางแบบ 3 มิติ; แขนหรือไหล่ซึ่งสามารถเคลื่อนไหวเมื่อผู้ใช้ได้เคลื่อนอุปกรณ์ที่ใช้บังคับ

รูปที่ 3 แสดงอุปกรณ์แสดงผลที่ใช้ในระบบเวอชวลเรียลลิตี้

ความสำเร็จของกราฟฟิกอินเทอร์เฟส(graphic interface)นี้ทำให้การแสดงผลการทดลองทางวิทยาศาสตร์สามารถแสดงได้ด้วย 3 มิติ พัฒนาการเหล่านี้ทำให้เราเพิ่มการคิดคำนวณที่สามารถแสดงผลภาพและเสียงตามเวลาจริงก็คือการสร้างโลกเสมือนจากคอมพิวเตอร์

การประยุกต์ใช้งาน

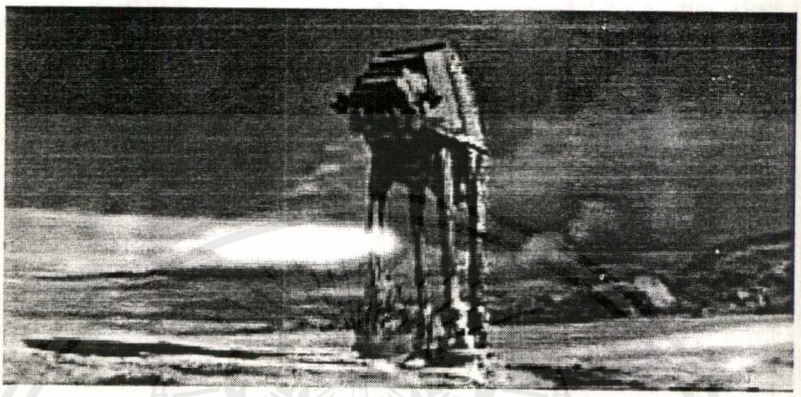
สมัยสงครามพายุทะเลทรายมีการใช้ระบบเวอร์ชวลเรียลริตี้สร้างแบบจำลองในการฝึกซ้อมขับรถถังให้กับทหารหลังจากนั้นก็ได้มีการพัฒนาไปใช้ในวงการอุตสาหกรรมหลายอย่าง เช่น**สังคมเสมือน (Virtual Communities)** เป็นการตั้งเครือข่ายขึ้นมาเพื่อการสื่อสารติดต่อสังคมผ่านระบบเสมือนจริงเช่น The Whole Earth Electronics Link(WELL)เป็นบริการบนเครือข่าย อินเตอร์เน็ต



รูปที่ 4 แสดงการติดต่อสื่อสารผ่านระบบสังคมเสมือนที่กำลังจะเกิดขึ้นผ่านอินเตอร์เน็ท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

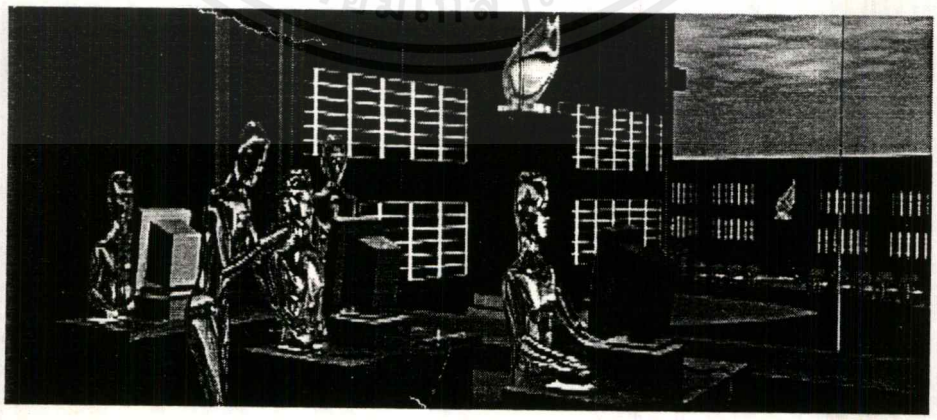
ความบันเทิง (Entertainment) ใช้ในการสร้างภาพยนต์ สร้างจินตนาการของมนุษย์ ถ่ายทอดความเป็นมาของประวัติศาสตร์สมัยโบราณ เช่นภาพจำลองของไดโนเสาร์ การจำลองเหตุการณ์การเกิดทอร์นาโดในอนาคตจะมีการแสดงคอนเสิร์ตผ่านเครือข่ายอินเตอร์เน็ต



รูปที่ 5 แสดงการใช้ระบบวีอาร์ช่วยในการทำภาพอนิเมชั่นในภาพยนตร์

การศึกษา (Education) มีการผลิตโปรแกรมที่ทำให้คนดูสามารถดูโครงสร้างของโปรตีน รวมทั้งมีการสร้างเครือข่ายเพื่อให้ผู้สอนและผู้เรียนมีการติดต่อสื่อสารถึงกันและยังมีการใช้จำลองในการเสริมความรู้เด็ก

ใช้ตามสำนักงาน (Office) ในสำนักงานปัจจุบันได้ออกแบบให้อยู่ในรูปของการใช้กระดาษน้อยที่สุดจะประกอบไปด้วยระบบหลายสื่อ(Multimedia)นอกจากนั้น แล้วยังมีการใช้เพื่อควบคุมระบบอัตโนมัติในโรงงานยานอวกาศ



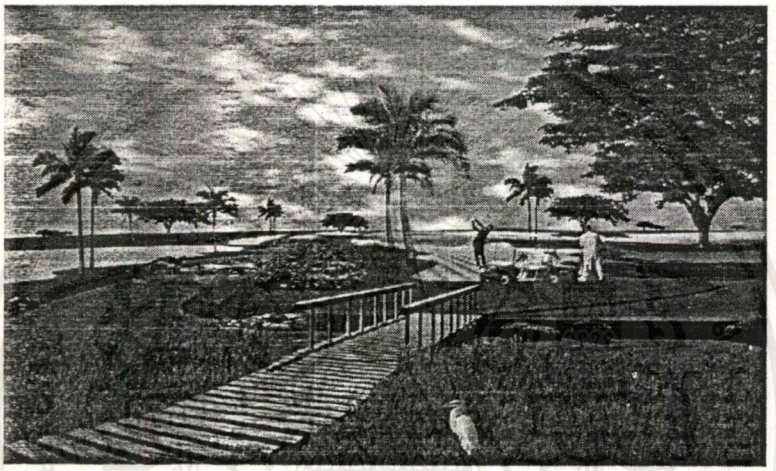
รูปที่ 6 ระบบสำนักงานในอนาคตที่ใช้ระบบวีอาร์ทำให้สามารถทำงานเสมือนอยู่ใกล้กันได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การประชุมทางไกลและการออกแบบ (Teleconference and Design)

เมื่อระบบเสมือนจริงมีการแพร่หลายจึงถูกใช้ในการใช้ในการประชุมทางคอมพิวเตอร์และการออกแบบมีการใช้เวอร์ชวลเรียลริตี้กันอย่างแพร่หลาย

วงการการศึกษาใช้ในการประกอบการเรียนการสอนทางไกลและวงการอุตสาหกรรมสามารถเพิ่มประสิทธิภาพมากขึ้นคือการนำระบบทำเป็นระบบวีดีโอคอนเฟอร์เรนท์(Video conferrence) ซึ่งจะทำให้มีการโต้ตอบกันได้แลกเปลี่ยนความคิดระหว่างผู้เรียนและผู้สอน

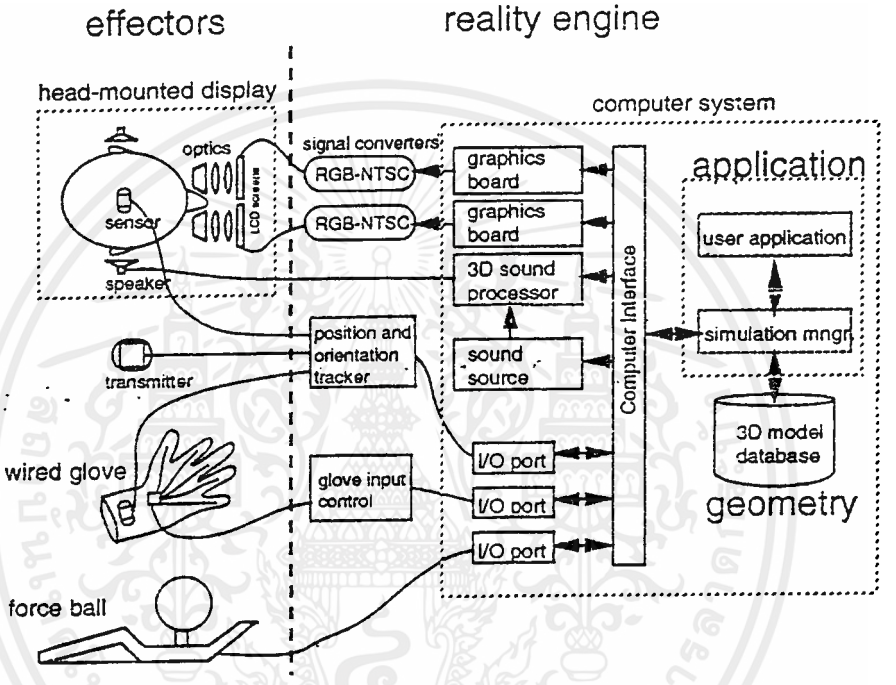


รูปที่ 5 แสดงระบบประชุมทางไกลที่ใช้ระบบวีอาร์ ทำให้เสมือนกับว่าได้พบกันจริงๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
ส่วนประกอบของระบบเวอร์ชวลเรียลริตี้

1.1 ส่วนประกอบของระบบเวอร์ชวลเรียลริตี้ สามารถแบ่งออกได้เป็น 4 ส่วน ดังรูป



รูปที่ 1.1 แสดงส่วนประกอบของระบบ

1.1.1 ส่วนสร้างผล (Effectors)

เป็นส่วนที่ทำการรับอินพุทจากการเคลื่อนไหวป้อนเข้าไปยังคอมพิวเตอร์และรับเอาท์พุทจากการประมวลผลโดยคอมพิวเตอร์ออกมาแสดงผล ตัวอย่างของอุปกรณ์นี้ได้แก่ ถุงมือ (wired glove) ซึ่งจะเปลี่ยนการเคลื่อนที่ของมือเป็นข้อมูลป้อนเข้าไปยังระบบคอมพิวเตอร์ แว่นตาและจอภาพแบบสวมศีรษะ (Goggle and Head Mounted Display) เป็นส่วนแสดงภาพที่เกิดจากการประมวลผลให้แก่ผู้ใช้ นอกจากอุปกรณ์อินพุทและเอาท์พุทพื้นฐานดังกล่าวเทคโนโลยีได้ทำให้เกิดอุปกรณ์ที่ใช้อีกหลายชนิดดังจะได้กล่าวต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.1.2 **เครื่องประมวผล (Reality Engine)**

คือระบบคอมพิวเตอร์ที่ใช้ในการประมวลผลข้อมูลและจำลองสถานการณ์เสมือนขึ้นนั่นเอง เป็นระบบที่มีความสามารถในการประมวลผลอย่างรวดเร็ว ทั้งทางด้านคณิตศาสตร์และระบบกราฟฟิกเพื่อให้สามารถโต้ตอบกับผู้ใช้ได้อย่างรวดเร็วถือว่าเป็นหัวใจสำคัญของระบบเวอร์ชวลเรียลริตี้

1.1.3 **โปรแกรมประยุกต์ (Application Software)**

เป็นส่วนที่ทำหน้าที่จัดการกับข้อมูลที่รับเข้ามา แล้วคอยกำหนดความสัมพันธ์ระหว่างส่วนต่างๆของสถานการณ์ที่สร้างขึ้นกับความเป็นจริง

1.1.4 **ส่วนเก็บข้อมูล (Geometry)**

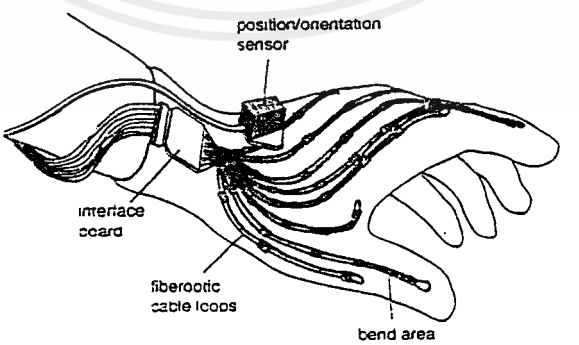
คือระบบการจัดเก็บข้อมูลที่ใช้อธิบาย ลักษณะของวัตถุ (รูปร่าง,ตำแหน่ง,สี และอื่นๆ) และฐานข้อมูลของโลกเสมือน (Virtual Space) ที่สร้างขึ้นโดยโปรมแกรมประยุกต์จะนำข้อมูลต่างๆเหล่านี้มาหาความสัมพันธ์และสร้างสถานการณ์จำลองขึ้น เช่นการสร้างภาพเคลื่อนไหวจำลองของมือที่สัมพันธ์กับการเคลื่อนไหวจริง การ

1.2 **เทคโนโลยีของอุปกรณ์ที่ใช้ในระบบเวอร์ชวลเรียลริตี้**

อุปกรณ์ในระบบเวอร์ชวลเรียลริตี้ใช้เทคโนโลยีในการสร้างหลายอย่างเนื่องจากต้องการอุปกรณ์ที่มีประสิทธิภาพสูงในการใช้งานที่ต่างกัน ตัวอย่างของอุปกรณ์ดังกล่าวเช่น

1.2.1.**อุปกรณ์อินพุท (Input Device)** ทำหน้าที่ในการเปลี่ยนการเคลื่อนไหวของร่างกายของผู้ใช้เป็นข้อมูลป้อนเข้าไปยังระบบเวอร์ชวลเรียลริตี้ตัวอย่างเช่น

- ถุงมือ (Glove) ส่วนประกอบของถุงมือประกอบด้วยอุปกรณ์ตรวจจับการเคลื่อนที่หรือบอกตำแหน่งของมือ และอุปกรณ์ตรวจการขยับของนิ้วดังรูปที่ 1

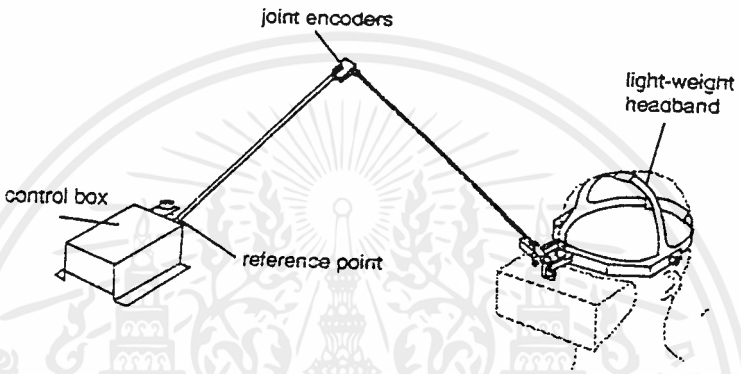


รูปที่ 1.2 แสดงโครงสร้างของถุงมือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

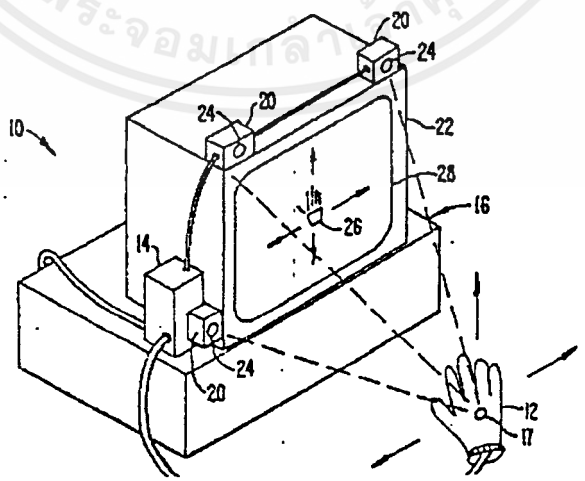
อุปกรณ์บอกตำแหน่งของมืออาจสร้างได้จากเทคโนโลยีหลายอย่าง เช่น

- ระบบกลไก (Mechanical) จะใช้แมคคานิคเชื่อมต่อระหว่างจุดอ้างอิงกับถุงมือโดยทั่วไป เป็นโครงน้ำหนักเบาที่มีจุดหมุนซึ่งเปลี่ยนการเคลื่อนไหวให้เป็นข้อมูลป้อนเข้าไปยังเครื่องคอมพิวเตอร์ได้ดังรูปที่ 2 ข้อดีของวิธีนี้คือความเร็วและความแม่นยำสูง แต่การเคลื่อนไหวจะถูกจำกัดด้วยโครงสร้างที่ใช้



รูปที่ 1.3 แสดงโครงสร้างที่ใช้ระบบกลไก

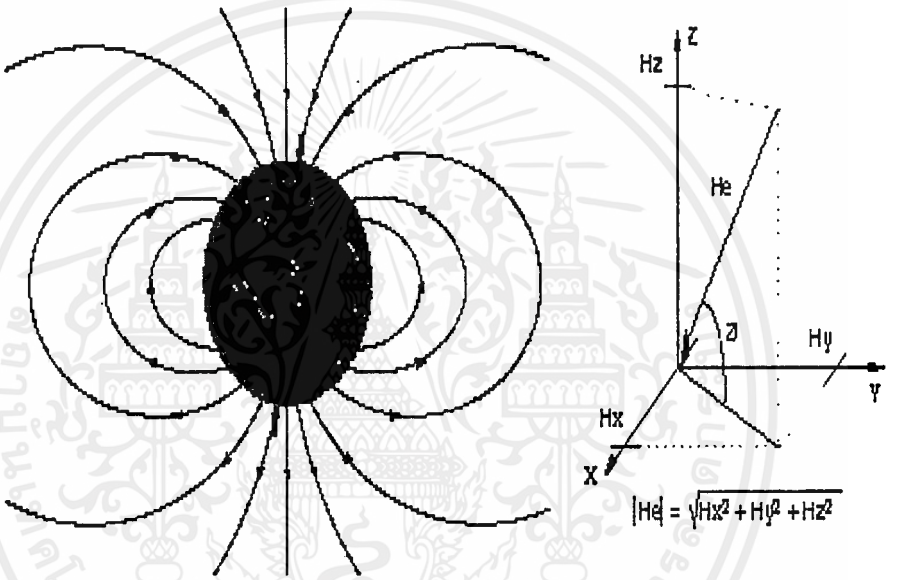
- ระบบอัลตราโซนิค (Ultrasonic) ใช้ตัวรับอัลตราโซนิคจำนวนสามตัวรับสัญญาณจากตัวส่งที่ติดไว้กับถุงมือเพื่อใช้วัดระยะของมือกับตัวรับในสามมิติดังรูปที่1.4 แล้วส่งข้อมูลเข้าไปยังคอมพิวเตอร์ ข้อดีคือความเร็วในการวัดสูง แต่ยังมีข้อเสียคือตัวรับและตัวส่งต้องอยู่ในทิศเดียวกันเพื่อให้รับสัญญาณกันได้ทำให้การเคลื่อนไหวไม่อิสระ



รูปที่ 1.4 แสดงถุงมือที่ใช้ระบบอัลตราโซนิค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

-ระบบสนามแม่เหล็ก (Magnetic) ใช้ขดลวดสร้างสนามแม่เหล็กส่งออกมาและใช้ขดลวดรับสนามแม่เหล็กติดไว้กับถุงมือเมื่อวัดความเข้มสนามแม่เหล็กก็จะสามารถทราบตำแหน่งของมือในสามมิติได้ดังรูป ระบบนี้ให้ข้อดีคือความเร็วในการวัดสูงและสามารถเคลื่อนไหวได้อย่างอิสระ ข้อเสียคือความยุ่งยากในการสร้างจึงทำให้ต้องลงทุนสูง



รูปที่ 1.5 แสดงระบบตรวจการเคลื่อนที่โดยใช้สนามแม่เหล็ก

-ระบบตรวจจับการเคลื่อนที่โดยใช้ไจโรสโคป ใช้เซ็นเซอร์ที่เรียกว่าไจโรสโคปซึ่งสามารถตรวจจับการหมุนของมือในทิศทางต่างๆแล้วส่งข้อมูลเข้าไปยังคอมพิวเตอร์เพื่อจำลองลักษณะการเคลื่อนที่ของมือบนจอภาพคอมพิวเตอร์ ดังจะกล่าวต่อไปในรายละเอียดของโครงงาน

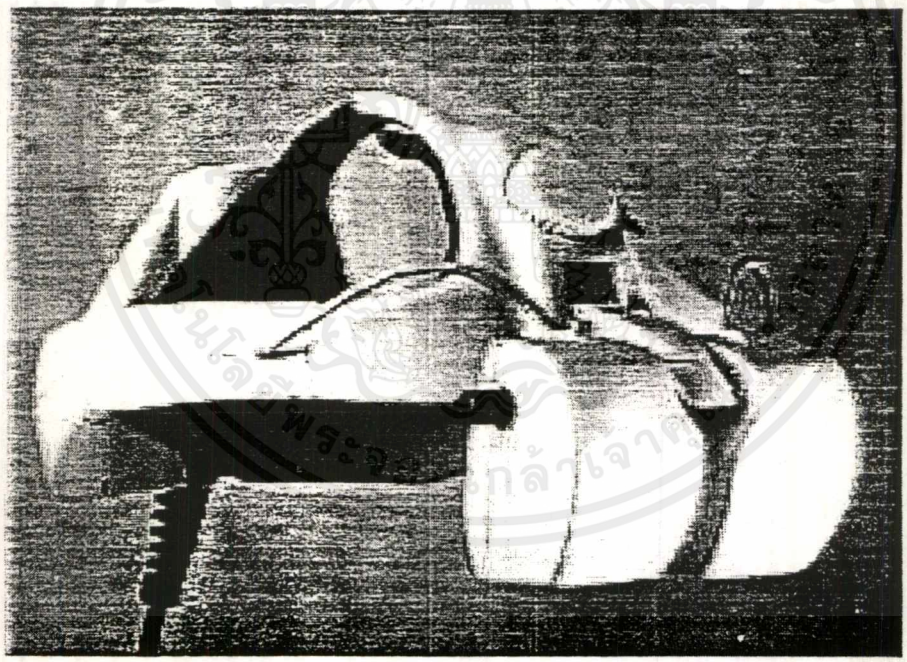
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.2.3 **อุปกรณ์เอาท์พุท (Output Device)** เป็นอุปกรณ์แสดงผลข้อมูลที่ผ่านการประมวลผลโดยคอมพิวเตอร์เพื่อโต้ตอบกับผู้ใช้ ตัวอย่างเช่น

- จอแสดงผล (Display) ส่วนใหญ่ใช้จอที่มีความละเอียดสูงและสามารถแสดงสีได้มากเพื่อให้มีความเหมือนจริงแต่มีข้อเสียคือจอมีขนาดใหญ่ไม่สะดวกใช้งาน

- แว่นตา (Goggle) เป็นจอแสดงผลขนาดเล็กสร้างจากเทคโนโลยี LCD (Liquid Cristal Display) ซึ่งมีความบางน้ำหนักเบาสามารถติดตั้งไว้กับแว่นตาสวมให้กับผู้ใช้ทำให้การมองเห็นมีความเหมือนจริงมาก ข้อเสียคือมีราคาแพง

- จอภาพแบบสวมศีรษะ (Head Mounted Display , HMD) มีลักษณะคล้ายหมวกกันน็อกใช้สวมศีรษะดังรูป โดยด้านหน้าของหมวกจะมีจอแสดงผลแบบ LCD ตั้งอยู่ และภายในหมวกยังมีอุปกรณ์ตรวจการเคลื่อนที่ของศีรษะแล้วป้อนข้อมูลเข้าไปยังคอมพิวเตอร์เพื่อสร้างภาพเคลื่อนไหวที่มีความสัมพันธ์กับการเคลื่อนไหวของศีรษะ



รูปที่ 1.6 แสดงจอภาพแบบสวมศีรษะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- เสียง ใช้อุปกรณ์สังเคราะห์เสียง (Digital Sound Processing) สร้างเสียงในสามมิติขึ้นมา แล้วส่งผ่านไปยังผู้ใช้โดยผ่านหูฟังในระบบสเตอริโอ ทำให้ ผู้ใช้รู้สึกถึงตำแหน่งของของเสียงที่สัมพันธ์กับภาพที่เห็น

นอกจากตัวอย่างที่ได้กล่าวมายังมีอุปกรณ์อีกหลายอย่างที่สร้างขึ้นมาเพื่อใช้กับระบบเวอร์ชวลเรียลริตี้ ซึ่งสามารถหารายละเอียดได้จากหนังสืออ้างอิงท้ายเล่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

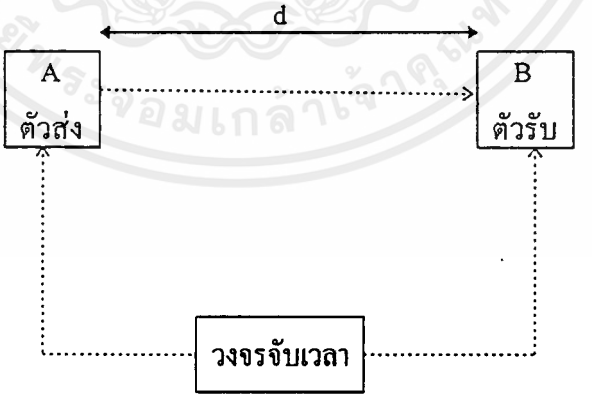
บทที่ 2
การวัดระยะด้วยอัลตราโซนิค

2.1 การวัดระยะด้วยอัลตราโซนิค (Ultrasonic Positioning System)

คลื่นอัลตราโซนิคเป็นคลื่นเสียงที่มีความถี่สูงกว่า 15 KHZ สามารถสร้างขึ้นโดยอาศัยตัวส่งอัลตราโซนิค (Ultrasonic Transmitter) และสามารถตรวจจับได้โดยอาศัยตัวรับคลื่นอัลตราโซนิค (Ultrasonic Reciever) การวัดระยะทางโดยอาศัยคลื่นอัลตราโซนิคมีหลักการดังรูปที่ 1 โดยอาศัยตัวส่งอัลตราโซนิค ซึ่งวางอยู่ที่จุด A เป็นตัวส่งคลื่นอัลตราโซนิคไปยังตัวรับอัลตราโซนิคที่วางอยู่ที่จุด B ซึ่งห่างออกไปเป็นระยะ d ถ้าเราทำการจับเวลาที่คลื่นอัลตราโซนิคใช้ในการเคลื่อนที่จากจุด A ไปยังจุด B จึงทำให้สามารถคำนวณหาระยะ d ได้จาก

$$d = vt$$

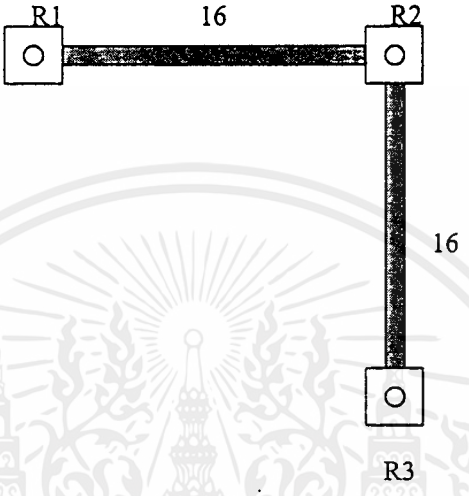
โดย d คือระยะห่างระหว่างตัวส่งกับตัวรับมีหน่วยเป็นเมตร
v ความเร็วคลื่นเสียงในอากาศมีค่าประมาณ 346 m/s
t ช่วงเวลาที่เสียงเคลื่อนที่จากจุด A ไปยังจุด B มีหน่วยเป็นวินาที



รูปที่ 2.1 แสดงการวัดระยะด้วยอัลตราโซนิค

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากหลักการวัดระยะที่กล่าวมาเป็นการวัดระยะในมิติเดียวเราสามารถนำมาใช้ในการบอกตำแหน่งของวัตถุในสามมิติ (x,y,z) ได้โดยอาศัยหลักการดังรูปที่ 2



รูปที่ 2.2 แสดงตำแหน่งของตัวรับอัลตราโซนิคทั้งสาม

จากรูปที่2 ตัวส่งอัลตราโซนิคจะติดตั้งไว้กับวัตถุและตัวรับอัลตราโซนิคจะจัดวางเป็นรูปสามเหลี่ยม เราจะให้ชื่อตัวรับแต่ละตัวเป็น R1,R2 และ R3 โดย R1 เป็นตัวรับที่อยู่มุมบนซ้าย, R2 เป็นตัวรับที่อยู่มุมบนขวา และ R3 เป็นตัวรับที่อยู่มุมล่างขวา ถ้าเราสมมุติให้ตัวรับ R1 อยู่ที่ตำแหน่ง 0,0,0 เราสามารถบอกตำแหน่งของตัวรับตัวอื่นๆ ได้ถ้าให้ระยะห่างระหว่างตัวรับ R1 กับตัวรับ R2 เป็น 16 หน่วย และระยะห่างระหว่างตัวรับ R2กับ R3 เป็น 16 หน่วย ดังนั้นตำแหน่งของตัวรับ R2 คือ 16,0,0 และตำแหน่งของตัวรับ R3 คือ 16,16,0 จะเห็นว่าตำแหน่งในแกน z ของตัวรับทั้งสามตัวเป็น 0 เนื่องจากตัวรับทั้งสามวางตัวอยู่ในระนาบ x-y

จากนั้นสมมุติว่าตัวรับทั้งสามเป็นจุดกึ่งกลางของรูปทรงกลมสามรูปดังรูปที่3 และเราต้องการทราบตำแหน่ง(x,y,z) ของตัวส่งซึ่งสมมุติให้อยู่ที่ผิวทรงกลมทั้งสาม จากหลักการวัดระยะที่กล่าวมาทำให้เราสามารถหาระยะห่างระหว่างตัวส่งกับตัวรับทั้งสามได้ ให้แทนด้วย R1d,R2d และ R3d ตามลำดับ ดังนั้นระยะ R1d จึงเป็นรัศมีของทรงกลมซึ่งมีจุดกึ่งกลางอยู่ที่ตัวรับR1 และตัวส่งอยู่ที่ตำแหน่ง x,y,z ใดๆที่ผิวของทรงกลม และเช่นเดียวกับตัวรับที่เหลืออีก 2 ตัว

เราสามารถหาค่าตำแหน่ง x,y,z ของตัวส่งได้โดย พิจารณาสมการของรูปทรงกลม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$\text{รัศมี}^2 = (x-x_{center})^2+(y-y_{center})^2+(z-z_{center})^2$$

ถ้าเราแทนโคออรดิเนตของตัวรับทั้งสามตัวลงไป เราจะได้สมการทรงกลมสามสมการคือ

$$R1d^2 = (x-x1_{center})^2+(y-y1_{center})^2+(z-z1_{center})^2$$

$$R2d^2 = (x-x2_{center})^2+(y-y2_{center})^2+(z-z2_{center})^2$$

$$R3d^2 = (x-x3_{center})^2+(y-y3_{center})^2+(z-z3_{center})^2$$

จากนั้นแทนค่าโคออรดิเนตของ R1,R2,R3 ลงไปจะได้

$$R1d^2 = x^2+y^2+z^2$$

$$R2d^2 = (x-16)^2+y^2+z^2$$

$$R3d^2 = (x-16)^2+(y-16)^2+z^2$$

แก้สมการหาค่า x,y,z จะได้

$$x = \frac{(R1d2-R2d2+256)}{32}$$

$$y = \frac{(R2d2-R3d2+256)}{32}$$

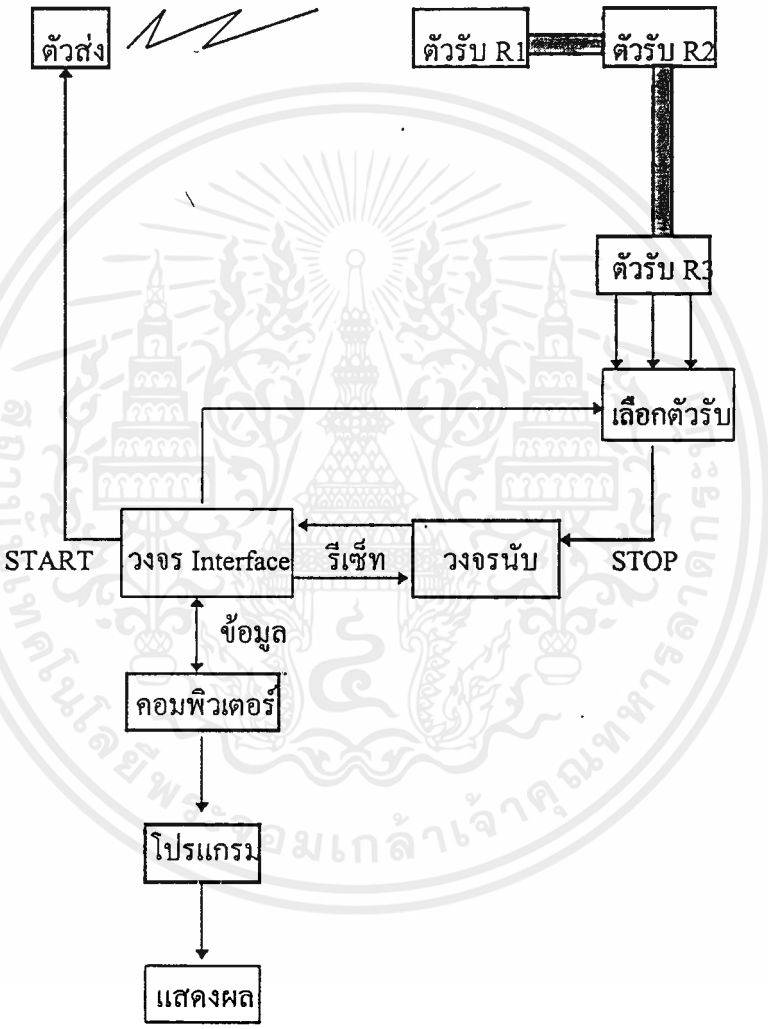
$$z = SQRT(R1d2+x2+y2)$$

จากสมการทั้งสามถ้าเราแทนค่าระยะ R1d,R2d,R3d ซึ่งเป็นระยะห่างของตัวส่งกับตัวรับทั้งสามตามลำดับก็จะทำให้ทราบตำแหน่งในสามมิติ (x,y,z) ของตัวส่งได้

จากหลักการดังกล่าวจึงสามารถนำมาประยุกต์ใช้ในโครงงานโดยติดตัวส่งอัลตราโซนิคไว้กับมือของผู้ใช้ดังนั้นจึงสามารถบอกตำแหน่งของมือของผู้ใช้แล้วป้อนเข้าไปยังเครื่องคอมพิวเตอร์เพื่อนำข้อมูลที่ได้ไปใช้ในประมวลผล เช่น การสร้างกราฟฟิกแสดงการเคลื่อนที่ของมือ ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในบทนี้จะเป็นการนำเอาหลักการบอกตำแหน่งด้วยอัลตราโซนิคมาใช้ในการบอกตำแหน่งของมือที่เคลื่อนที่แล้วป้อนเข้าไปยังคอมพิวเตอร์ บล็อกไดอะแกรมของโครงงานแสดงดังรูปที่ 1 ตัวส่งจะถูกติดไว้ที่มือของผู้ใช้ ส่วนตัวรับจำนวนสามตัวจะถูกติดตั้งไว้เป็นรูปสามเหลี่ยมดังที่กล่าวมาแล้ว การวัดระยะห่างระหว่างตัวส่งและตัวรับแต่ละตัวจะทำทีละคู่เป็นจำนวน 3 ครั้งโดยมีการทำงานดังนี้



รูปที่ 2.3 :แสดงบล็อกไดอะแกรมของโครงงาน

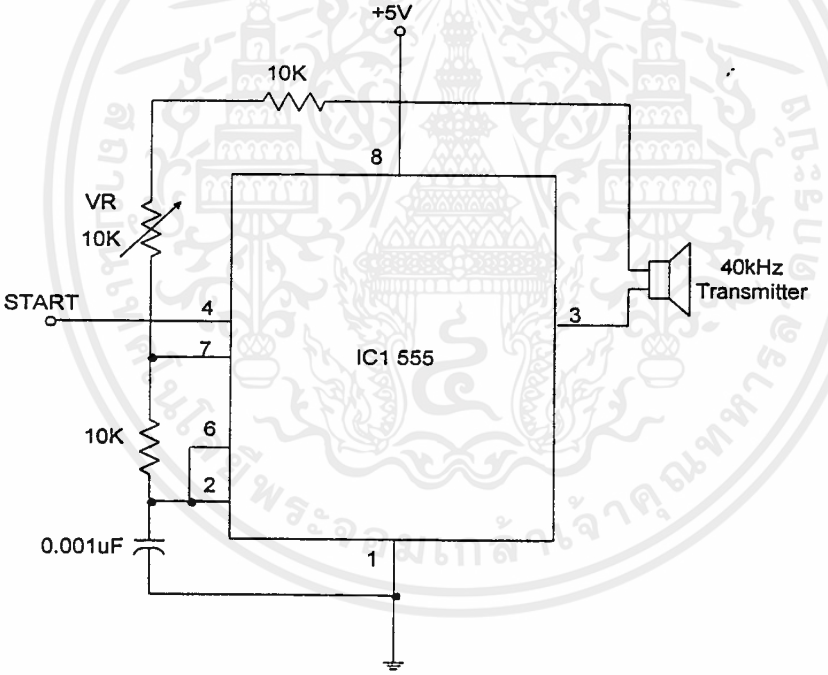
อันดับแรกคอมพิวเตอร์จะเลือกให้ตัวรับ R1 ทำงานจากนั้นคอมพิวเตอร์ส่งสัญญาณสตาร์ต(START) มาจะทำให้ตัวส่งส่งคลื่นอัลตราโซนิคที่มีความถี่ 40 เฮิร์ซที่ได้จากวงจรออสซิลเล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เตอร์ออกมาและวงจรนับเลขฐานสองขนาด 16 บิทก็จะเริ่มทำการนับ เมื่อตัวรับ R1 ได้รับสัญญาณจากตัวส่งก็จะส่งสัญญาณสต็อบ(STOP)ออกมาซึ่งจะทำให้วงจรนับเลขฐานสองหยุดนับ และคอมพิวเตอร์ก็จะทำการอ่านค่าเวลาที่ได้จากวงจรนับผ่านวงจรอินเทอร์เฟสเข้าไปแล้วทำการคำนวณระยะห่างระหว่างตัวส่งและตัวรับ R1 ออกมา จากนั้นก็จะเริ่มเลือกให้ตัวรับ R2 และ R3 ทำงานต่อไปตามลำดับ เมื่อได้ระยะห่างครบทั้งสามค่าก็จะนำมาคำนวณหาตำแหน่งของมือ(x,y,z) เมื่อเทียบกับตัวรับโดยใช้สมการในบทที่ผ่านมา แล้วนำค่าตำแหน่งที่ได้ไปสร้างกราฟฟิกแสดงการเคลื่อนที่ของมือบนหน้าจอคอมพิวเตอร์ แล้วจึงเริ่มกระบวนการวัดระยะใหม่ทั้งหมดอีกครั้ง

2.2 วงจรส่งคลื่นอัลตราโซนิค (Ultrasonic Transmitter)

ในส่วนของวงจรส่งคลื่นอัลตราโซนิคแสดงดังรูปที่ 2.4



รูปที่ 2.4 แสดงวงจรส่งคลื่นอัลตราโซนิค

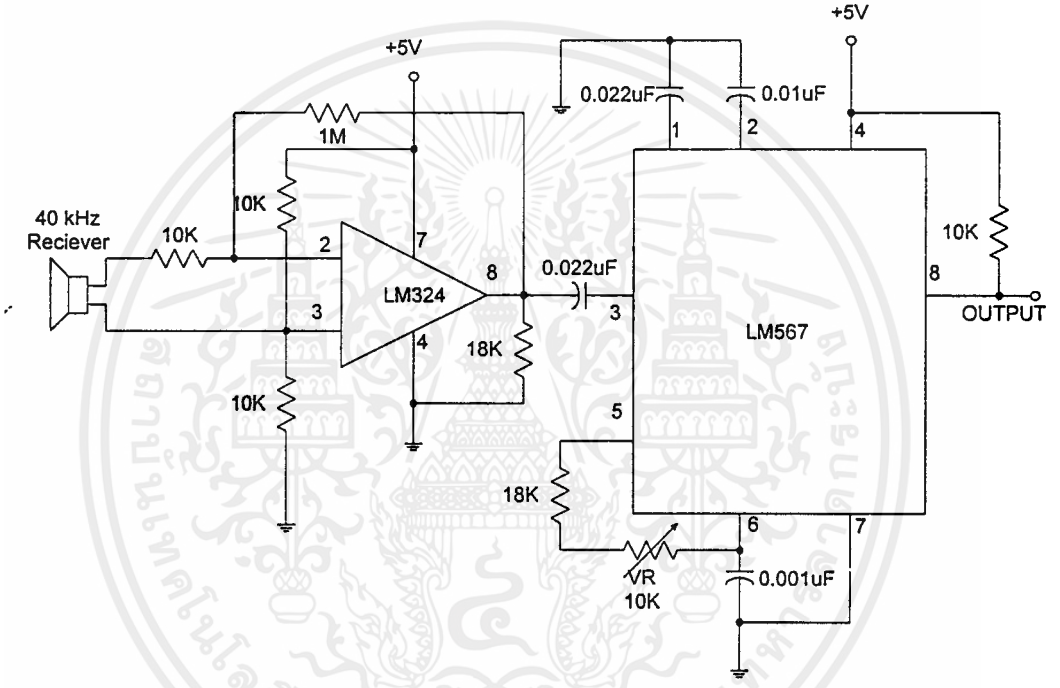
การทำงานของวงจรเริ่มจากไอซี 555 ต่อเป็นวงจรอะสเตเบิ้ลมัลติไวเบรเตอร์สร้างพัลซ์รูปสี่เหลี่ยมที่มีความถี่ 40 เฮิร์ซออกมาที่ขา3 ผ่านไปยังตัวส่งอัลตราโซนิคซึ่งจะแปลงพลังงานไฟฟ้าให้เป็นพลังงานเสียงส่งออกไปยังตัวรับ การทำงานของไอซี 555 สามารถควบคุมได้ที่ขา 4 ซึ่งเป็นขารี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เซ็ท ถ้าสถานะที่ขา 4 มีลอจิกเป็น "HIGH" ไอซี 555 ก็จะทำงานสร้างพัลซ์ออกมา แต่ถ้าที่ขา 4 มีลอจิกเป็น"LOW" เอาท์พุทของไอซี 555 ก็จะเป็น"LOW" ดังนั้นจึงสามารถควบคุมการทำงานของวงจรส่งคลื่นโดยผ่านคอมพิวเตอร์ได้

2.3 วงจรรับคลื่นอัลตราโซนิค (Ultrasonic Reciever)

ส่วนของวงจรรับคลื่นอัลตราโซนิคแสดงดังรูปที่ 2.5



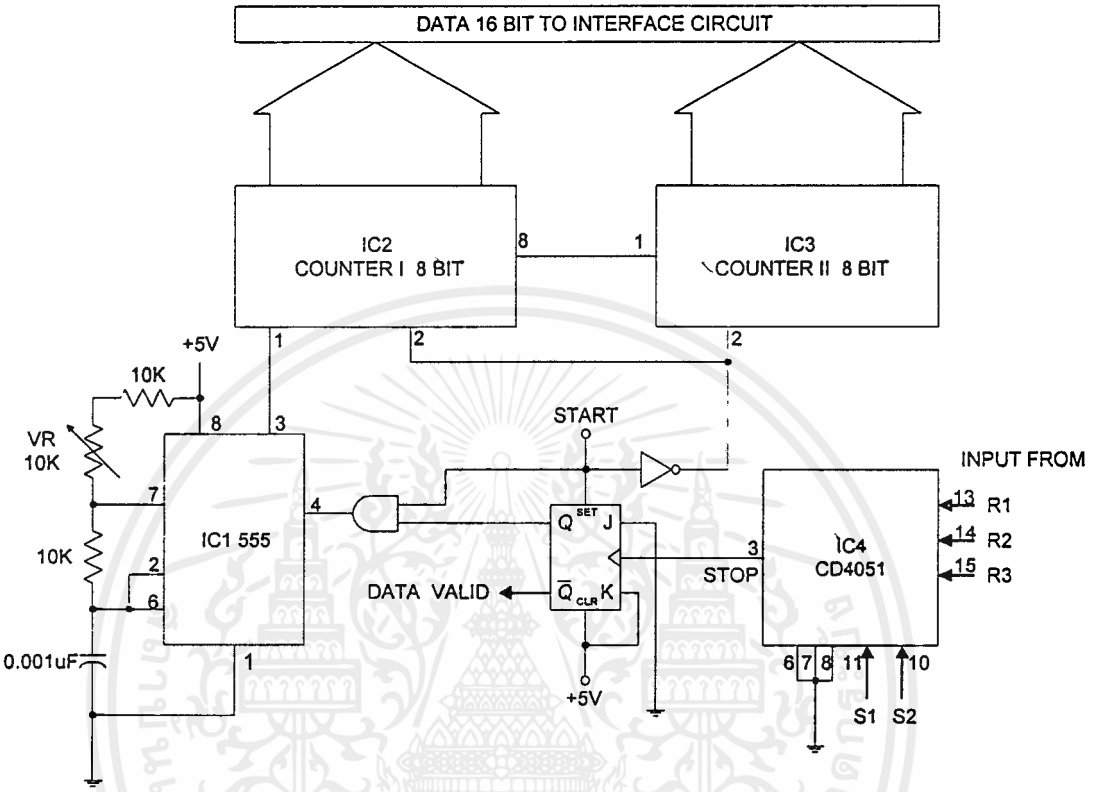
รูปที่ 2.5 แสดงวงจรรับคลื่นอัลตราโซนิค

การทำงานของวงจรเริ่มจากตัวรับอัลตราโซนิคได้รับสัญญาณ 40 kHz จากตัวส่งเข้ามาจากนั้นจะถูกขยายโดย IC1 ได้เอาท์พุทออกมายังขา 8 ผ่านไปยัง IC2 ซึ่งเป็นไอซีเฟสล็อกลูปโดยจะให้เอาท์พุทที่ขา 8 เป็น "LOW" เมื่ออินพุทที่เข้ามามีความถี่ตรงกับค่าความถี่ที่กำหนด โดยในวงจรนี้คือ 40 เฮิร์ซดังนั้นเมื่อได้รับคลื่นอัลตราโซนิคความถี่ 40 เฮิร์ซเข้ามาก็จะทำให้วงจรรับให้เอาท์พุทเป็น "LOW" ถ้ายังไม่ได้รับคลื่นก็จะให้เอาทุ์ทเป็น "HIGH" จากนั้นเอาท์พุทจาก IC2 จะผ่านไปยัง IC3 ซึ่งเป็น IC Flipflop ซึ่งจะแลทช์ค่าเอาท์พุทจาก IC2 ไว้เพื่อใช้เป็นสัญญาณสต๊อบ สำหรับวงจรนับฐานสองต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 วงจรนับฐานสองขนาด 16 บิท (16 Bit Binary Counter)

ส่วนของวงจรนับแสดงดังรูปที่ 2.6



รูปที่ 2.6 แสดงวงรนับเลขฐานสองขนาด 16 บิท

การทำงานของวงจรจะเริ่มนับเมื่อมีสัญญาณ "HIGH" ที่ขา START จะทำให้สัญญาณนาฬิกาจากวงจรออสซิลเลเตอร์ ที่สร้างจาก IC1 ความถี่ 34 เฮิร์ซผ่านไปยัง IC2 และ IC3 ซึ่งต่อเป็นวงจรนับเลขฐานสองขนาด 8 บิทจำนวน 2 ชุด ทำให้สามารถนับได้สูงสุดเป็น 16 บิท ดังนั้น IC2 และ IC3 จะนับสัญญาณนาฬิกา ที่เข้ามาเรื่อยๆ จนกระทั่งเมื่อขา Q ของ J-K ฟลิบฟลอบมีสถานะเป็น "LOW" สัญญาณนาฬิกาจาก IC1 จะไม่สามารถผ่านไปยัง IC2 ,IC3 ได้ดังนั้นวงจรนับจะหยุดนับ และขา DATA VALID จะมีสถานะเป็น "HIGH" ซึ่งจะทำให้คอมพิวเตอร์ทราบว่าสิ้นสุดการนับแล้วและทำการอ่านข้อมูลที่ได้เข้าไป ขา Clock ของ J-K FlipFlop จะถูกต่อกับ IC4 ซึ่งทำหน้าที่เลือกสัญญาณเพียงหนึ่งเส้นจากเอาท์พุทของวงจรรับอัลตราโซนิคสามวงจร โดยการเลือกของ IC4 จะถูกควบคุมโดยคอมพิวเตอร์ผ่านขา S1 และ S2 ดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

S1 S2

0 0 วงจรรับ R1 จะถูกเลือก

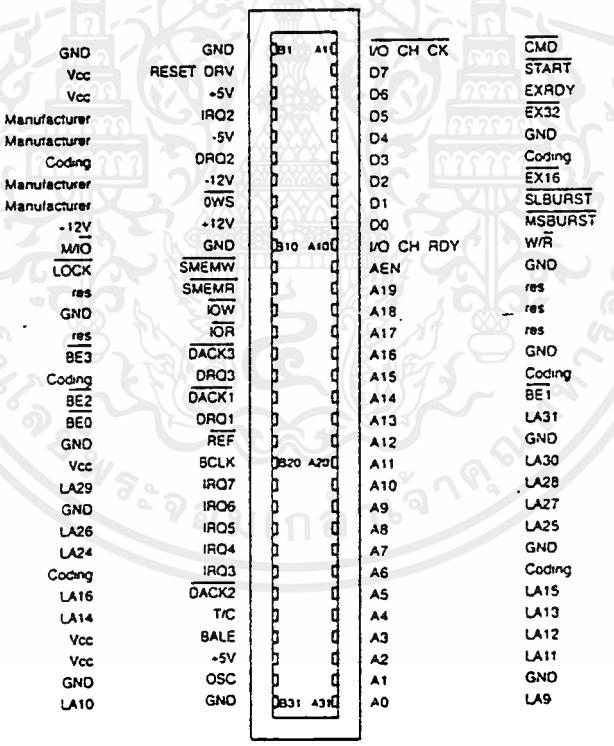
0 1 วงจรรับ R2 จะถูกเลือก

1 0 วงจรรับ R3 จะถูกเลือก

ส่วนขา 2 เป็นขารีเซ็ทเมื่อมีสถานะเป็น"LOW"วงจรนับจะถูกรีเซ็ทให้เป็นศูนย์

2.5 วงจรอินเตอร์เฟซ กับคอมพิวเตอร์ (Computer Interface)

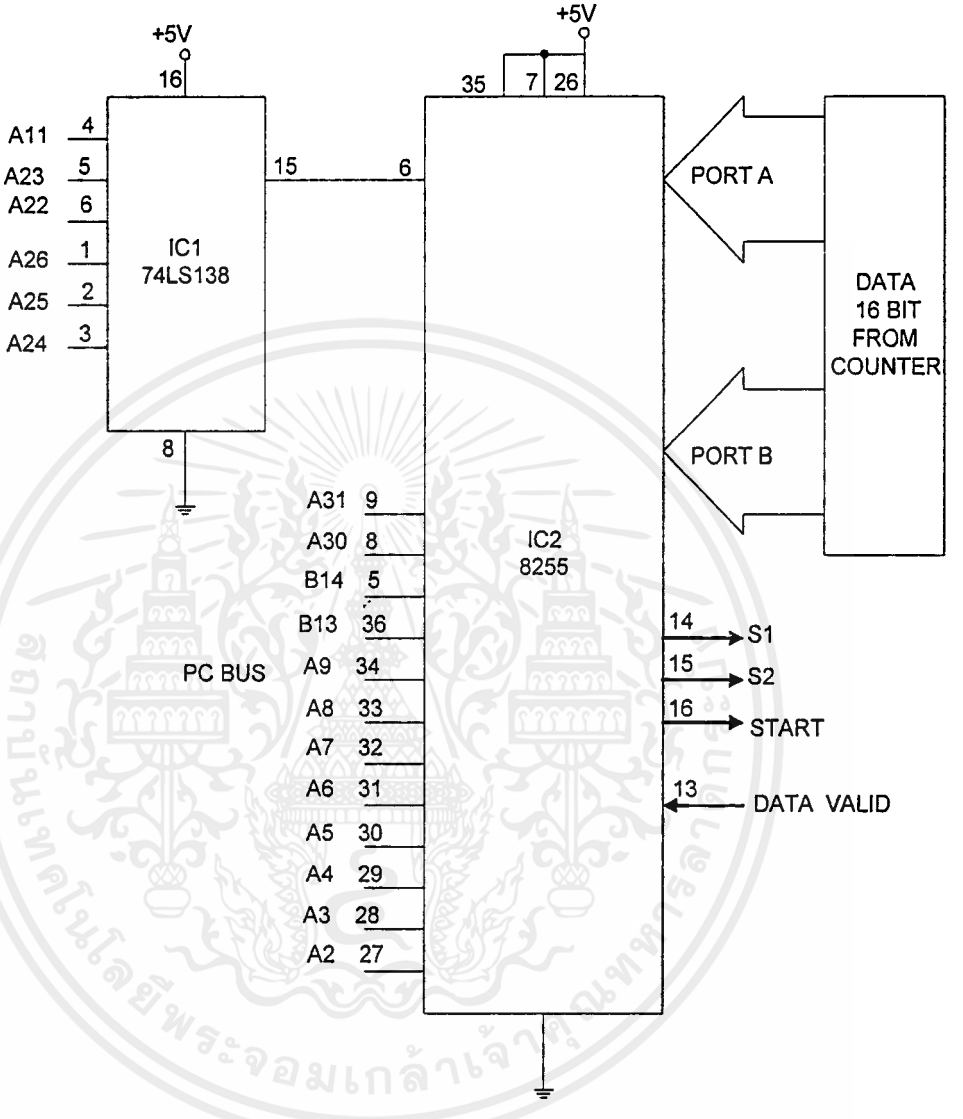
ในการส่งข้อมูลเข้าไปยังคอมพิวเตอร์จะใช้การอินเตอร์เฟซโดยผ่านบัสของคอมพิวเตอร์ โดยการสร้างการ์ดอินเทอร์เฟสเสียบลงในช่อง ISA ซึ่งรายละเอียดของช่อง ISA แสดงดังรูปที่ 2.7



รูปที่ 2.7 แสดงขาต่างๆใน ISA SLOT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
038391

ส่วนของวงจรอินเตอร์เฟซที่สร้างแสดงดังรูปที่ 2.8



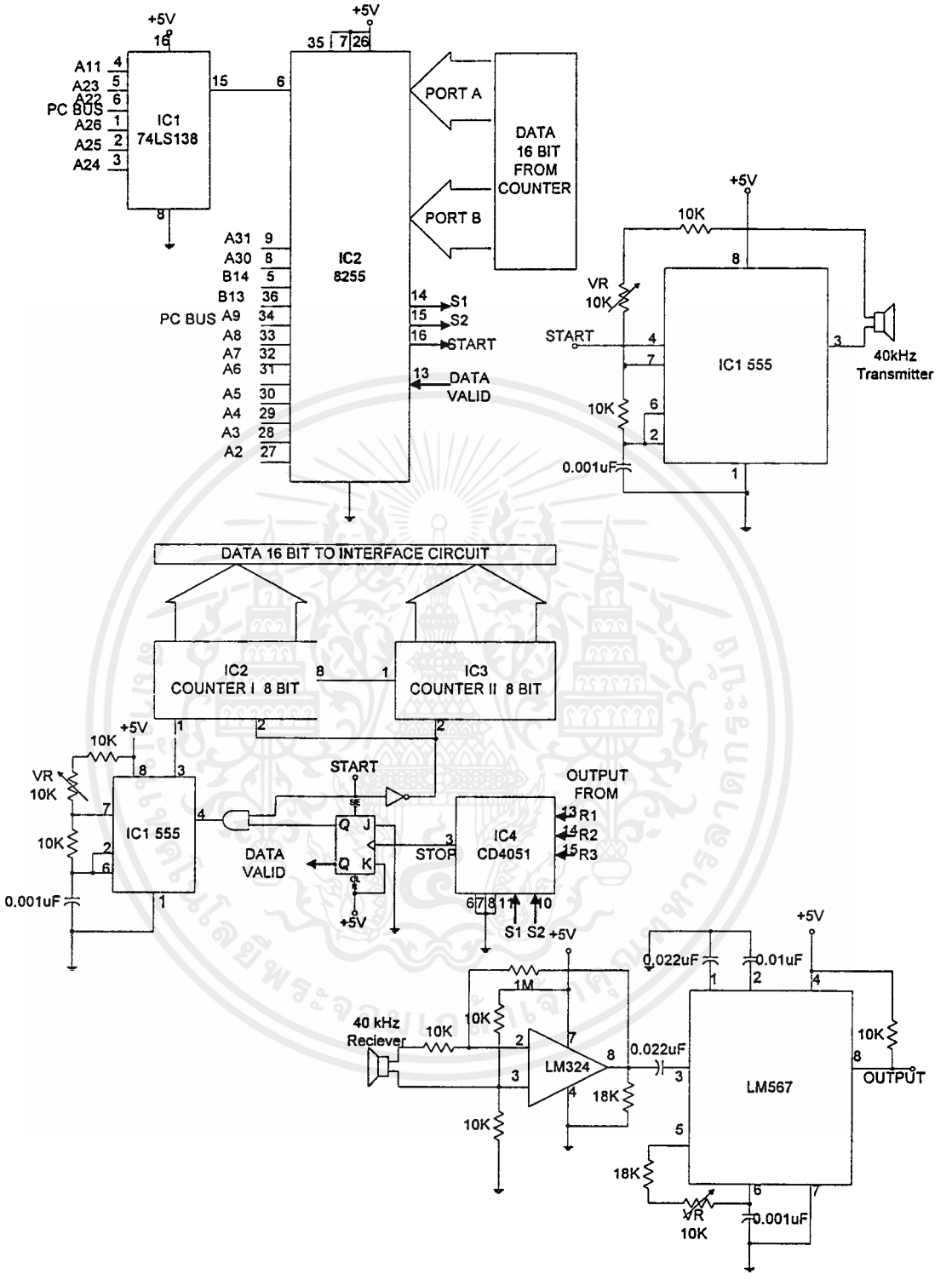
รูปที่ 2.8 แสดงวงจรอินเตอร์เฟซกับเครื่องคอมพิวเตอร์ผ่านบัสของคอมพิวเตอร์

จากรูปเป็นการอินเตอร์เฟซโดยอาศัยไอซี8255 ซึ่งทำให้ได้พอร์ต ขนาด 8 บิทจำนวน 3 พอร์ตโดยเราจะชั้ 2 พอร์ตแรกรวมกันเป็น 16 บิท เพื่อรับข้อมูลจากวงจรนับ ส่วน พอร์ตที่เหลือเราจะใช้ 4 บิทแรก เป็นเอาท์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในการเลือกตัวรับซึ่งใช้ 2 บิทแรก บิทที่ 3 จะใช้เป็นสัญญาณสตาร์ทให้กับวงจรนับ ส่วนอีก 4 บิทหลังของพอร์ทนี้เราจะใช้เป็นอินพุทโดยใช้ 1 บิทในการรับสัญญาณ DATA VALID เพื่อบอกให้คอมพิวเตอร์ทำการอ่านข้อมูลจากวงจรนับเข้าไป ส่วน IC1 เป็นตัวทำหน้าที่ถอดรหัสแอดเดรสของคอมพิวเตอร์เพื่อกำหนดแอดเดรสของวงจรอินเตอร์เฟซในการติดต่อระหว่างโปรแกรมกับวงจรซึ่งกำหนดไว้ที่แอดเดรส 200H จะได้วงจรสมบูรณ์ดังรูปที่2.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 แสดงรูปวงจรสมบูรณ์ของโครงงานที่สร้างในรูปแสดงวงจรรับอัลตราโซนิคเอาไว้เพียงชุดเดียวคือชุด R1 แต่ต้องใช้ทั้งหมดจำนวน 3 ชุดซึ่งมีวงจรเหมือนกันตามที่กล่าวมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
อุปกรณ์ในโครงงานเวอร์ชวลเรียลริตี้

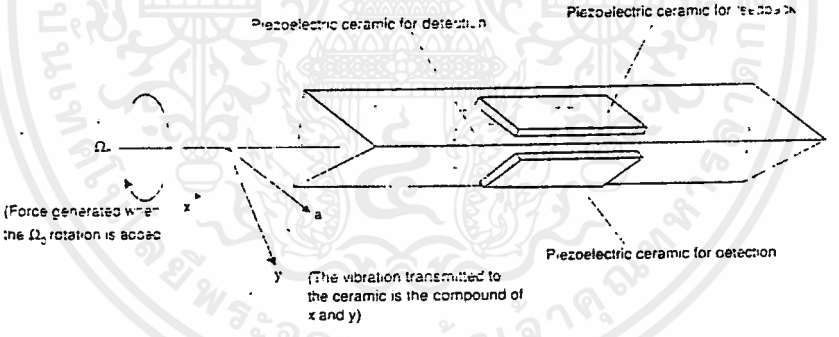
3.1 ไจโรสโคป(Gyroscopes)

ไจโรสโคปเป็นอุปกรณ์เซ็นเซอร์ ซึ่งทำหน้าที่บอกทิศทางในการหมุนของวัตถุ เอาท์พุทที่ได้จากไจโรสโคปอาจเป็นค่าของแรงดันที่แปรผันตามความเร็วเชิงมุมที่วัตถุหมุนไป หรือเป็นพัลซ์ที่มีความถี่เปลี่ยนแปลงตามความเร็วเชิงมุมในการหมุนของวัตถุ

จากคุณสมบัติเบื้องต้นของไจโรสโคป ทำให้สามารถนำมาใช้ในการตรวจจับการเคลื่อนไหวของร่างกายเพื่อใช้เป็นสัญญาณป้อนเข้าไปให้คอมพิวเตอร์ทำการประมวลผลตอบสนองออกมา ดังนั้นในโครงงานนี้ จึงใช้ไจโรสโคปเป็นอุปกรณ์อินพุทของระบบเวอชวลเอนยิน(Virtual Engine)

3.1.1 โครงสร้างของไจโรสโคป

ไจโรสโคปที่ใช้เป็นเซ็นเซอร์ของบริษัทมุราตะ(Murata Corporation) เบอร์ ENC-05E ซึ่งจากข้อมูลของบริษัท มีโครงสร้างพื้นฐานดังรูปที่ 3.1



รูปที่ 3.1 แสดงภาพของไจโรสโคปขณะอยู่นิ่งและขณะหมุน

โครงสร้างประกอบด้วยแผ่นเพียซโซเซรามิคจำนวน 3 ชิ้นวางอยู่บนด้านทั้งสามของปริซึมรูปสามเหลี่ยมคือ ชิ้น A,B และ C ตามลำดับ

ในขณะภาวะปกติแผ่น C จะถูกทำให้สั่นด้วยการป้อนความถี่จากวงจรออสซิลเลเตอร์ซึ่งจะส่งผลให้แผ่น A และ B สั่นด้วยความถี่ค่าเดียวกันได้เป็นสัญญาณที่มีความถี่ค่าเดียวกันกับวงจรออสซิลเลเตอร์ จากนั้นสัญญาณจาก A และ B จะผ่านไปยังวงจรดิฟเฟอร์เรนเชียล ได้เป็นสัญญาณเอาท์พุทของไจโรสโคปซึ่งจะเป็นผลต่างระหว่างสัญญาณเอาท์พุทจาก A และ B

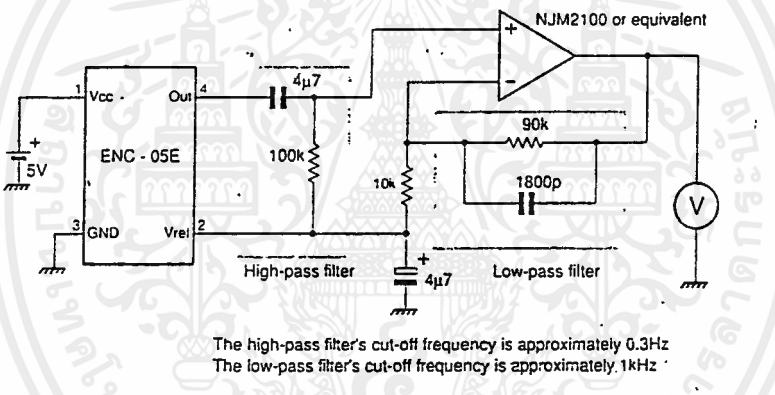
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$output(t) = a(t)-b(t)$$

ในขณะอยู่นิ่งสัญญาณจากเอาท์พุทของ A และ B มีขนาดเท่ากัน ดังนั้นเอาท์พุทที่ได้จากไจโรสโคปจึงมีค่าเป็นศูนย์

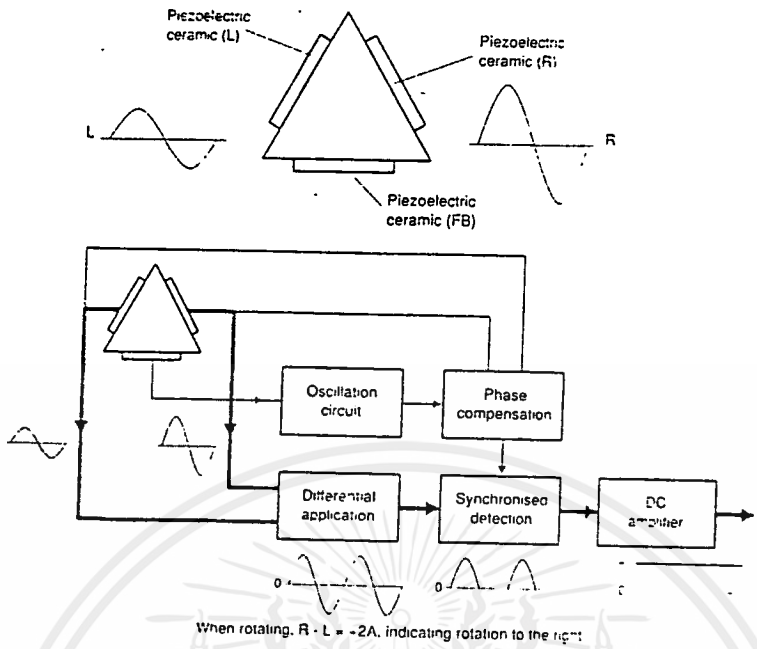
ในขณะที่มีการหมุนตามแนวยาวของปริซึม จะทำให้เกิดแรงกระทำบนเพียซโซเซรามิค ซึ่งทำให้สัญญาณเอาท์พุทของ A และ B มีขนาดที่แตกต่างกันตามรูป ดังนั้นเมื่อผ่านวงจรดิฟเฟอร์เรนเชียลจึงทำให้ได้สัญญาณเอาท์พุทที่มีขนาดแปรผันตามความเร็วเชิงมุมในการหมุนของไจโรสโคป

3.1.2 การทำงานอุปกรณ์อิเลคทรอนิกส์ของไจโรสโคป



รูปที่ 3.2 แสดงวงจรใช้งานพื้นฐานจากคู่มือของไจโรสโคป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แสดงถึงความแตกต่างของโวลท์เตจเพื่อนำค่าดังกล่าวไปคำนวณหาทิศทาง

3.2 การส่งข้อมูลเข้าไปยังคอมพิวเตอร์

ในบทที่แล้วเป็นการใช้ไจโรสโคปในการตรวจวัดการหมุนของวัตถุ ซึ่งได้เอาท์พุทเป็นสัญญาณที่มีขนาดเปลี่ยนแปลงตามการหมุนของวัตถุ ในบทนี้จะเป็นการส่งข้อมูลดังกล่าวไปยังคอมพิวเตอร์เพื่อใช้ในการประมวลผลต่อไป ในการส่งข้อมูลเข้าไปยังคอมพิวเตอร์มีด้วยการหลายวิธี เช่น ทางพอร์ทขนาน(Pararel port), ทางพอร์ทอนุกรม(Serial port) แต่เพื่อความง่ายในการเขียนโปรแกรมในการเชื่อมต่อ (Interface) จึงเลือกใช้การส่งข้อมูลผ่านทางพอร์ทอนุกรมจากเอาท์พุทที่ได้จากไจโรสโคปเป็นสัญญาณที่มีขนาดแอมปลิจูดเปลี่ยนแปลงตามความเร็วเชิงมุมในการหมุน ดังนั้นจึงสามารถบอกขนาดของมุมที่เปลี่ยนแปลงไปได้ เราจึงได้การเปลี่ยนแปลงของมุมในการหมุนในสองแกนคือรอบแกน X และรอบแกน Y ดังนั้นของมูลที่จะส่งเข้าไปยังคอมพิวเตอร์จึงเป็นค่าของการเปลี่ยนแปลงของมุมในรูปของเลขฐานสอง ซึ่งมีสองชุดคือรอบแกน X และ รอบแกน Y เพื่อความง่ายในการเขียนโปรแกรมจึงใช้รูปแบบการส่งในรูปของเมาส์(Mouse) ซึ่งเป็นอุปกรณ์อินพุทที่ทำหน้าที่เปลี่ยนการเคลื่อนที่ของลูกบอล(Roller)ให้เป็นข้อมูลการเคลื่อนที่ของเคอร์เซอร์ในแกน X และแกน Y บนจอภาพคอมพิวเตอร์

ดังนั้นจึงทำให้สามารถเปลี่ยนการหมุนของไจโรสโคปเป็นค่าตำแหน่งการเคลื่อนที่ของเคอร์เซอร์บนจอภาพคอมพิวเตอร์ได้ เขียนเป็นบล็อกไดอะแกรมได้ดังรูปที่3 เนื่องจากการส่งข้อมูลทำในรูปแบบเดียวกับการส่งข้อมูลของซีเรียลพอร์ทเมาส์ ดังนั้นเราจึงสามารถเขียน

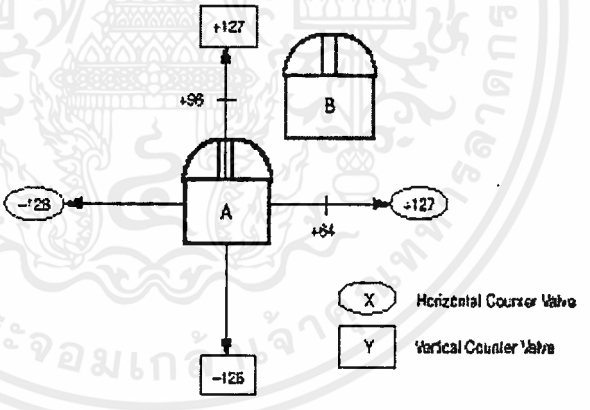
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมเพื่อตอบสนองต่อการเปลี่ยนแปลงของร่างกายได้เช่นเดียวกับการเขียนโปรแกรมเพื่อตอบสนองต่อการเคลื่อนที่ของเมาส์

3.3 เมาส์

หลักการส่งข้อมูลผ่านซีเรียลพอร์ทของเมาส์อาศัยการส่งข้อมูลการเปลี่ยนแปลงตำแหน่งของเมาส์ในรูปทูคอมพลีเมนท์(two's complement) ของเลขฐานสองขนาด 8 บิท ในการตรวจจับการเคลื่อนที่จะมีวงจรนับทั้งในแนวแกน X (แกนนอน) และ Y (แนวตั้ง) เนื่องจากเมาส์จะส่งเฉพาะข้อมูลการเปลี่ยนแปลงของตำแหน่งจากตำแหน่งสุดท้ายดังนั้นข้อมูลที่จะส่งเข้าไปจึงสามารถมีค่าอยู่ในช่วง -128 ถึง 127 หน่วย เครื่องหมายใช้ในการบอกว่าเป็นการเคลื่อนที่ไปทางซ้ายหรือขวาตามแกน X และบอกว่าเป็นการเคลื่อนที่ขึ้นหรือลงตามแกน Y

ดังรูปที่ 4 จะเห็นว่าที่ตำแหน่ง A มีค่าแกน X,Y=(0,0) และที่ตำแหน่ง B มีค่าแกน X,Y=(64,96) ข้อมูลที่ส่งให้กับคอมพิวเตอร์จากเมาส์โดยไมโครคอนโทรลเลอร์ที่อยู่บนเมาส์จะจัดการส่งเฟรมของข้อมูลโดยส่งในมาตรฐานของ RS-232 รูปแบบมาตรฐานของการส่งในหนึ่งเฟรมข้อมูลประกอบไปด้วย 3 ไบต์โดยที่บิตสูงสุดของแต่ละไบต์จะไม่ได้ใช้จึงทำให้มีเฉพาะ 21 บิตเท่านั้นที่เป็นข้อมูลที่จะถูกส่ง



รูปที่ 3.4 แสดงโครงสร้างของเมาส์ที่ใช้ LED และโฟโตทรานซิสเตอร์

Table 1. Three Byte Information Packet

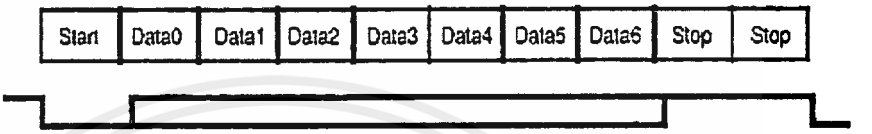
Bit No.	7	6	5	4	3	2	1	0
1st Byte	1	1	L	R	V7	V6	H7	H6
2nd Byte	1	0	H5	H4	H3	H2	H1	H0
3rd Byte	1	0	V5	V4	V3	V2	V1	V0

Note:
L = Left button data bit
R = Right button data bit
H0-H7 = Horizontal data byte
V0-V7 = Vertical data byte

รูปที่ 3.5 แสดงค่าในบิตต่างๆของเฟรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

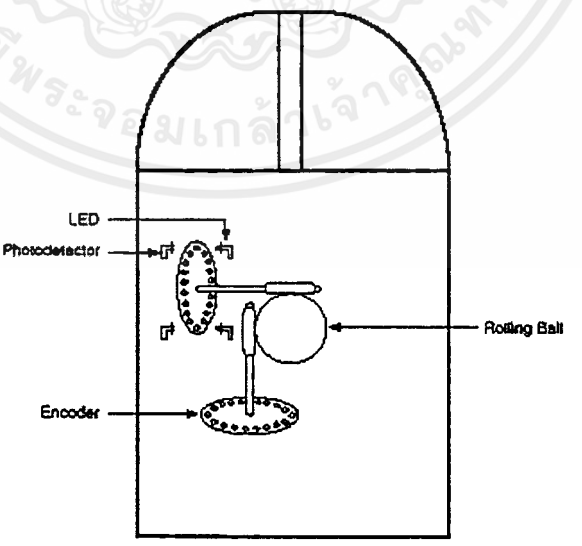
รูปแบบในสามไบต์แสดงดังรูปไบต์แรกจะบรรจุไปด้วยสถานะของสองปุ่มกด และสองบิตสูงสุดของแต่ละค่าของทิศทางการเคลื่อนไหวไบต์ที่สองประกอบไปด้วย 6 บิต ล่างของค่าการเคลื่อนไหวในแนวนอนและไบต์ที่สามประกอบไปด้วยค่าตำแหน่งของแนวตั้งฉากเมื่อส่งข้อมูลมาแล้วโดยบิตเริ่มต้นและสามารถใช้บิตที่ 7 ของอีกสองบิตเป็นสต๊อบบิต(Stop bit)ได้อีกด้วย



รูปที่ 3.6 แสดงการส่งข้อมูลแบบอนุกรม

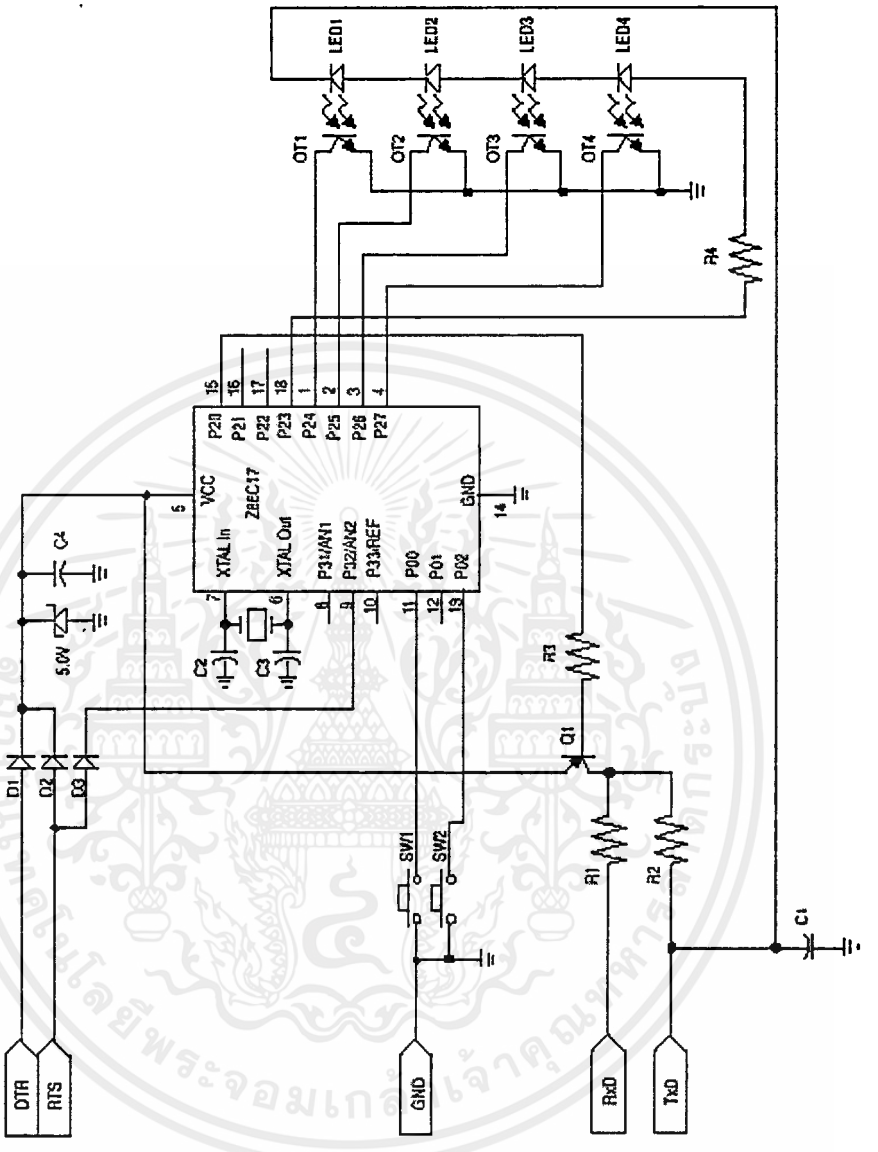
3.3.1 ความละเอียด(Resolution)

ความละเอียดต่อนิ้วที่เมาส์สามารถบอกถึงความแตกต่างได้เรียกเป็นจุดต่อนิ้วทั้งนี้ขึ้นอยู่กับโปรแกรมที่ใช้ด้วยโดยปกติความละเอียดอาจจะเป็น 100,200 และ 400 จุดต่อนิ้วเมาส์ที่ใช้โดยทั่วไปจะมีอยู่สามชนิดในการแปลงการเคลื่อนไหวให้เป็นตัวเลขเมาส์ที่ใช้แสงอาศัยการเคลื่อนไหวของลูกกลิ้ง,จะมีตัวเอ็นโค้ดเดอร์(Encoder)LEDs และโฟโตทรานซิสเตอร์จะแปลงการเคลื่อนไหวของลูกกลิ้งให้เป็นสัญญาณทางไฟฟ้า



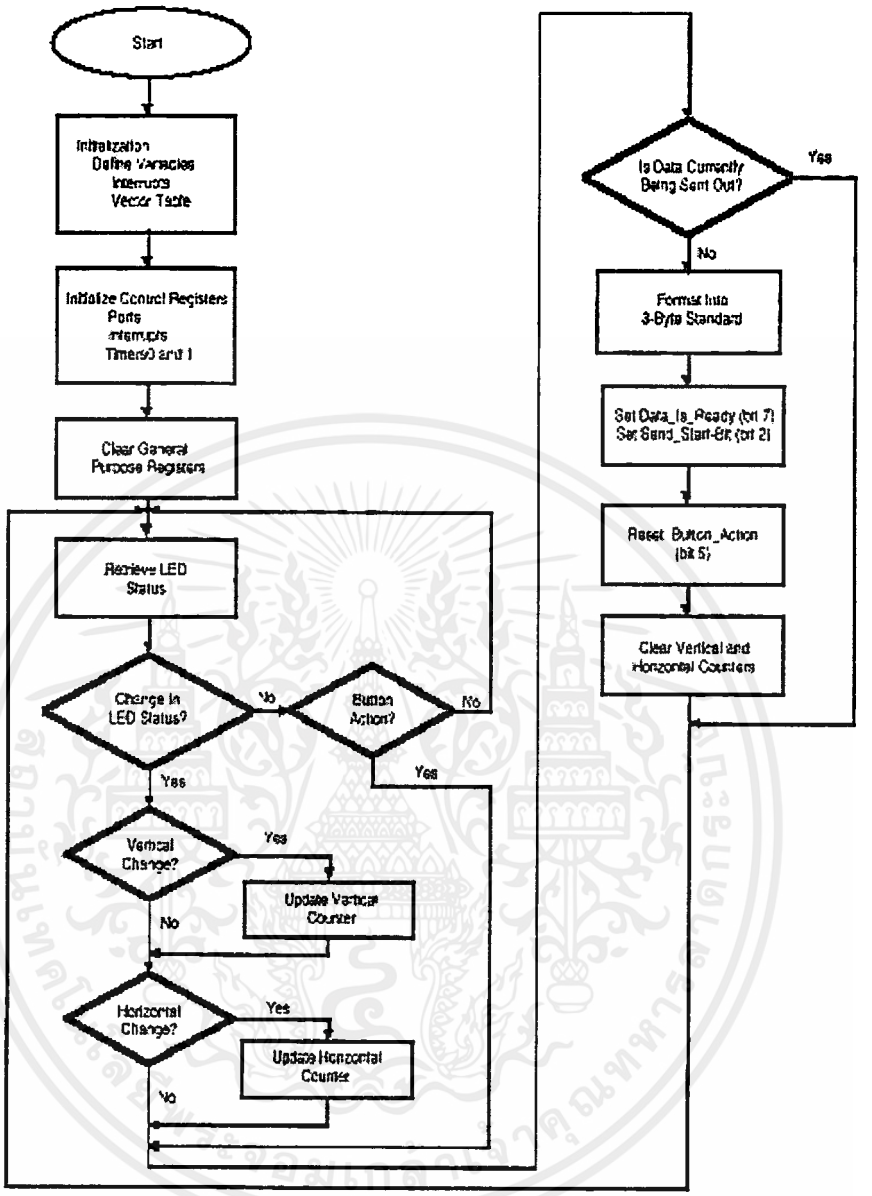
รูปที่ 3.7 แสดงเมาส์ที่ใช้แสง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 แสดงวงจรของเมาส์

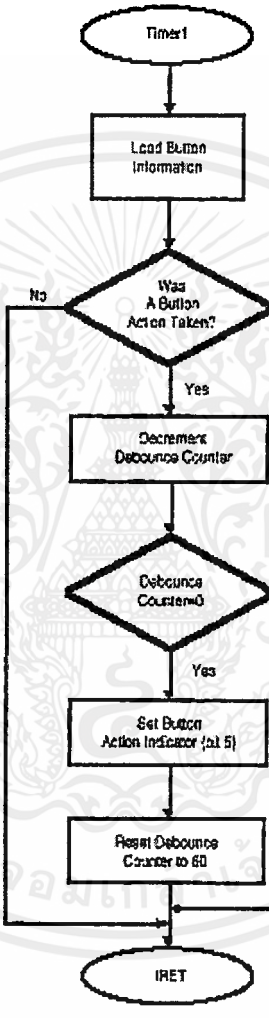
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 แสดงโฟลว์ชาร์ตของโปรแกรมการอินเทอร์รับท์ของเมาส์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

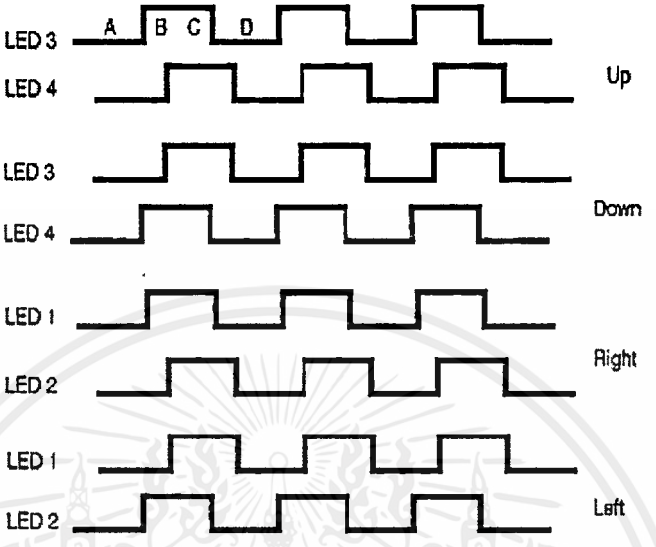
แต่ละปุ่มในเมาส์จะต่อกับสายอินเทอร์รับท์ของไมโครคอนโทรลเลอร์ ถ้ามีปุ่มมีการกดอินเทอร์รับท์รูทีนจะเริ่มทำงานโดยส่งข้อมูลของปุ่มที่กดไปยังพอร์ทอนุกรม



รูปที่ 3.10 แสดงรูทีนของการกดปุ่มเมาส์

โปรแกรมจะอาศัยสัญญาณจากโฟโตทรานซิสเตอร์ทุกครั้งที่มีการเปลี่ยนแปลงโดยไมโครคอนโทรลเลอร์จะเปรียบเทียบกับค่าที่มีอยู่ก่อนจากนั้นจะคำนวนค่าที่แตกต่างออกไปหากโปรแกรมตรวจพบว่ามีการเปลี่ยนแปลงคอนโทรลเลอร์จะตัดสินใจว่าเมาส์ได้เกิดการเคลื่อนที่การเปลี่ยนแปลงตำแหน่งนี้จะถูกส่งไปให้คอมพิวเตอร์ผ่านเฟรมของข้อมูลใน 3 ไบต์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.11 แสดงสัญญาณที่เกิดใน LED เมื่อมีการเคลื่อนที่ของเมาส์

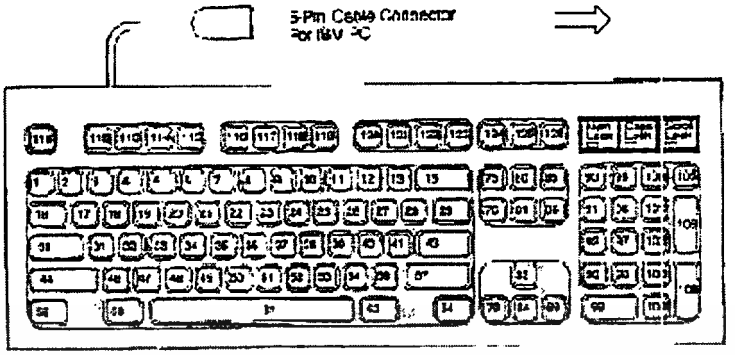
การเปลี่ยนทิศทางจะอาศัยหลักการทางเมคานิคอลของเมาส์ซึ่งประกอบไปด้วย LED/โฟโตดีเทคเตอร์ ช่องว่างในเอ็นโค้ดเดอร์จะถูกตั้งค่าให้สร้างรูปคลื่นดังรูปขึ้นมาเมื่อมีการเคลื่อนที่ของเมาส์โฟโตดีเทคเตอร์ถูกเลือกโดยโปรแกรมหลัก หากตรวจการเปลี่ยนแปลงของโฟโตทรานซิสเตอร์ แล้วพบว่าไม่มีการเปลียนแปลงของสัญญาณก็จะตัดสินใจได้ว่าเมาส์ไม่ได้มีการเคลื่อนที่

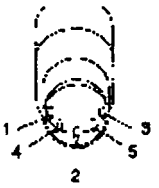
จากรูปแสดงคลื่นของสัญญาณสมมุติพบว่า LEDs 3 และ LEDs 4 (0,0) ที่ตำแหน่ง A และตอนนี้พบว่า LEDs 3 ยังคงเป็น 0 สำหรับ LEDs 4 (1,0) ตำแหน่ง B

3.4 คีย์บอร์ดและเมาส์

คอมพิวเตอร์ PC/XT/AT โดยปกติจะมีคีย์อยู่ 101 หรือ 102 คีย์ต้องการแหล่งจ่ายไฟ 5 V สายกราวด์ และสายข้อมูลและสายสัญญาณนาฬิกาโดยโหมดการสื่อสารจะเป็นแบบอนุกรมดังรูป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้





DIN Pins	Signal Name	Signal Type
1	+KBD CLK	I/O
2	+KBD Data	I/O
3		
4	GND	Power
5	+5.0 V_{dc}	Power
SHIELD	Frame GND	

รูปที่ 3.12 แสดงโครงสร้างของคีย์บอร์ดและสายสัญญาณ

3.4.1 การสื่อสารระหว่างคีย์บอร์ดและคอมพิวเตอร์

การติดต่อสื่อสารระหว่างคอมพิวเตอร์และคีย์บอร์ดจะเกิดขึ้นทุกๆ 250 ไมโครวินาที คีย์บอร์ดและอุปกรณ์อื่นๆที่อินเตอร์เฟสคล้ายกับคีย์บอร์ดโดยเฉพาะอย่างยิ่งคอมพิวเตอร์ AT จะได้ MF II Keyboard(multifunction keyboard)ซึ่งจะเป็นอุปกรณ์ที่ใช้ได้หลายอย่างในปัจจุบันทุกๆ คีย์บอร์ดจะมีชิปสำหรับคีย์บอร์ดโดยเฉพาะ ในคีย์บอร์ดจะมีชิบ 8048 ซึ่งเป็นชิบที่สร้างขึ้นมาเพื่อใช้ในการสแกนคีย์เพื่อลดเวลาการทำงานของโปรเซสเซอร์ทำโดยตัวชิบจะสแกนคีย์บอร์ดจะส่งค่าที่สแกนได้ให้กับโปรเซสเซอร์โดยผ่านสายส่งในสายส่งมีทั้งหมด 4 เส้นมีกราวด์,ไฟเลี้ยง,สายข้อมูลและสัญญาณนาฬิกาซึ่งการสื่อสารดังกล่าวจะเป็นแบบซิงโครนัส(การส่งข้อมูลไปพร้อมๆกับสัญญาณที่ให้จังหวะ)ชิบในคีย์บอร์ดจะตรวจค่าในสแกนเมตริก(Scan matrix)ว่ามีการเปิดหรือปิดวงจรอยู่เป็นการตรวจสถานะของสแกนเมตริกจุดประสงค์เพื่อที่จะตรวจจับค่าในแนวแกน X และในแนวแกน Y คือการตำแหน่งในแกนโดออดิเนตโดยในชิปจะทำหน้าที่เป็นตัวบัฟเฟอร์เก็บค่าไว้ก่อนชั่วขณะหนึ่งหลังจากนั้นก็จะส่งให้กับคอมพิวเตอร์โดยผ่านสายส่งสำหรับคีย์บอร์ดสัญญารนาฬิกาจากคีย์บอร์ดจะว่งให้กับแผงวงจรรวมของคอมพิวเตอร์ดังนั้นการสื่อสารจึงเป็นแบบซิงโครนัส

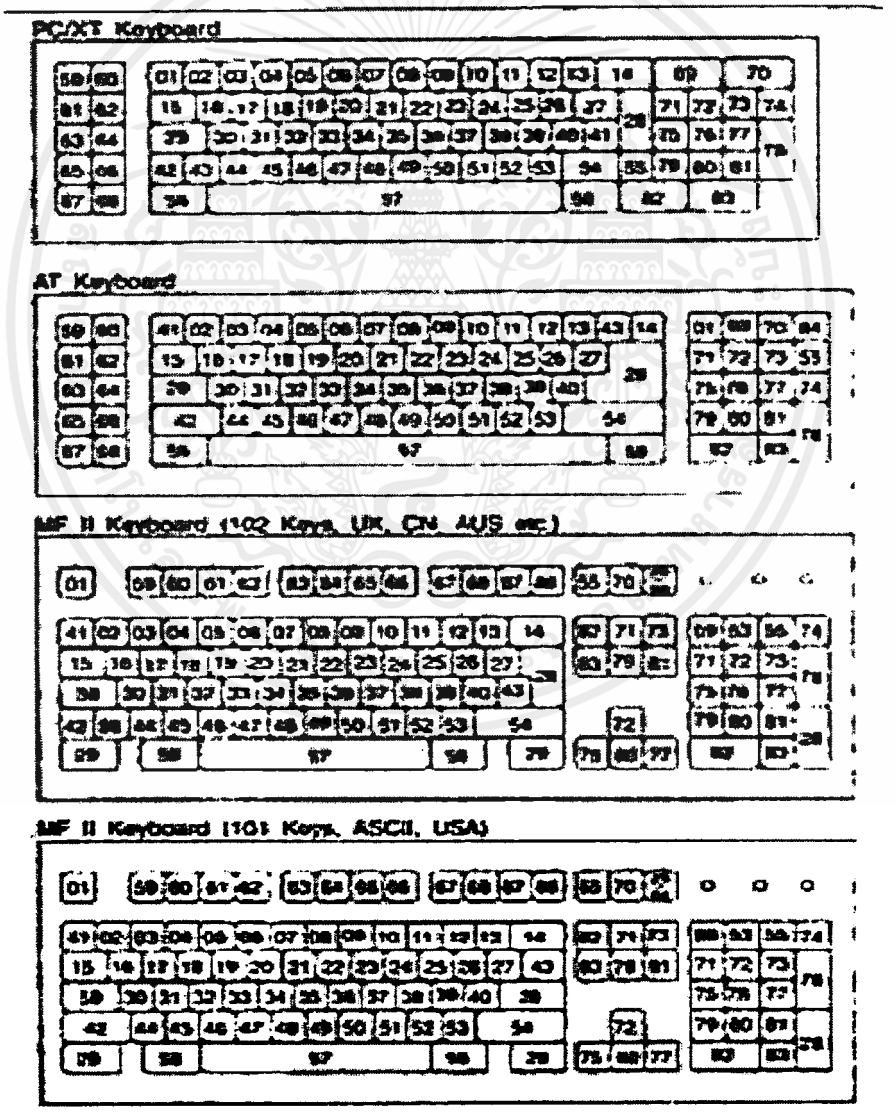
ในคอมพิวเตอร์ AT คีย์บอร์ดอินเทอร์เฟสถูกติดตั้งเรียบร้อยโดยมากใช้ชิป 8042 ในรุนใหม่จะใช้ชิป 8741 หรือ 8742 ดังนั้นคีย์ในรุ่นใหม่ๆนั้นจะมีความสามารถที่ดีกว่ารุ่นก่อนๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชิปคอนโทรลเลอร์ในปัจจุบันทำให้เราสามารถโปรแกรมได้เช่นสามารถที่จะสั่งให้คีย์บอร์ดไม่ต้องทำงานก็ได้หรือจะให้คีย์ส่งค่าต่างๆได้ตามที่ต้องการยิ่งไปกว่านั้นการสื่อสารสองทางยังเป็นประโยชน์สำหรับการพัฒนาโปรแกรมที่ต้องการอุปกรณ์อินพุทที่เป็นกรณีพิเศษเช่นการพัฒนาระบบเสมือนจริงหรือการนำระบบคอมพิวเตอร์มาใช้ในทางอุตสาหกรรมซึ่งต้องการอุปกรณ์อินพุทที่ไม่เหมือนกับคอมพิวเตอร์ทั่วๆไป

3.4.2 การสแกนโค้ดของคีย์บอร์ด

ความแตกต่างของรุ่นในคอมพิวเตอร์และความสามารถที่ดีกว่าเดิมทำให้ระบบคีย์บอร์ดต้องพัฒนาตามพื้นฐานของการสแกนคีย์จะเหมือนกันจะแตกต่างกันเฉพาะปริมาณคีย์ที่ถูกป้อนเข้าไปในรูปจะเห็นความแตกต่างของคีย์บอร์ดระบบต่างๆ

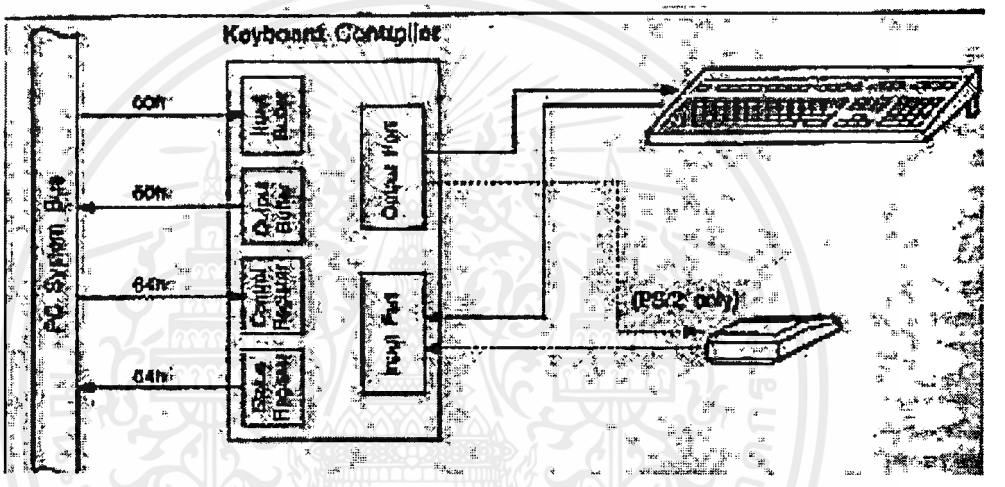


รูปที่ 3.13 แสดงโค้ดของรหัสในคีย์บอร์ด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.3 การเขียนโปรแกรมกับคีย์บอร์ดผ่านพอร์ตคอมพิวเตอร์โดยตรง

เราสามารถโปรแกรมคีย์บอร์ดและ MF II คีย์บอร์ดคล้ายๆกับอุปกรณ์อื่นๆ ได้ใน คอมพิวเตอร์ PC/XT คีย์บอร์ดซึ่งการส่งข้อมูลจะส่งไปทางเดียวเท่านั้นในทิศทางจากคีย์บอร์ดไปที่คอมพิวเตอร์เท่านั้นจะเป็นไปไม่ได้ที่จะโปรแกรมการทำงานดังกล่าวแต่ในคอมพิวเตอร์ PC/AT จะมีความยืดหยุ่นกว่าดังนั้นหากต้องการติดต่อกับคีย์บอร์ดชนิด AT หรือ PS/2 เมาส์ต้องเขียนโปรแกรมคอนควบคุมจากคีย์บอร์ดคอนโทรลเลอร์ดังแสดงในรูป 3.14



รูปที่ 3.14 แสดงภาพของคีย์บอร์ดคอนโทรลเลอร์

-รีจิสเตอร์และพอร์ต

การโปรแกรมจะใช้ตำแหน่งแอดเดรสสองตำแหน่งที่จัดสรรให้คือ 60h และ 64h การใช้งานพอร์ตดังกล่าวต้องติดต่อกับอินพุทบัฟเฟอร์,เอาท์พุทบัฟเฟอร์และรีจิสเตอร์ควบคุมจากรูปแสดงโครงสร้างการทำงานของพอร์ทซึ่งสามารถกำหนดการเขียนโปรแกรมเพื่อควบคุมพอร์ทโดยเซ็ตค่าที่รีจิสเตอร์ควบคุมการทำงานอาจจะใช้ภาษาแอสแซมบลีหรือทางคอมพิวเตอร์อื่นๆ

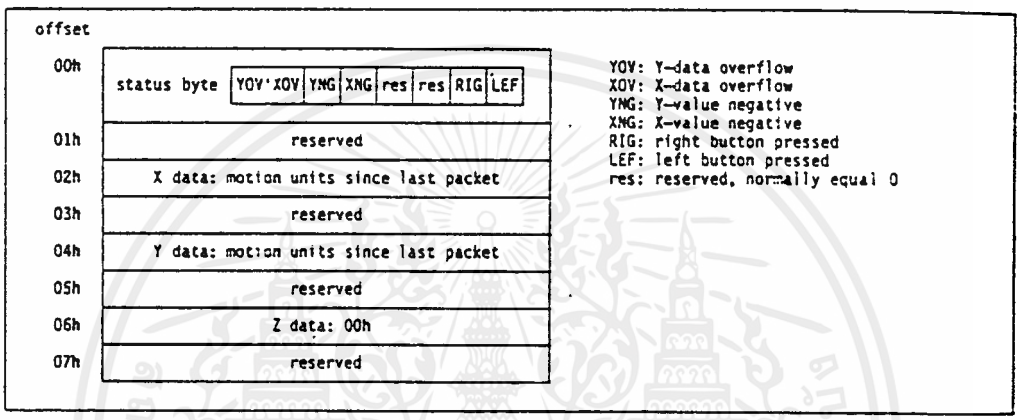
พอร์ท	รีจิสเตอร์	อ่าน(Read)เขียน(Write)
60h	Output buffer	R
60h	Input buffer	W
64h	Control register	W
64h	Status register	R

แสดงรีจิสเตอร์ควบคุมซึ่งอยู่ที่ตำแหน่งแอดเดรส64h

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.5 การทำงานของ PS/2 เมาส์

อุปกรณ์อินพุทของคอมพิวเตอร์ที่ใช้ในระบบกราฟฟิก IBM ได้เตรียมพอร์ทสำหรับการติดต่อกับพอร์ทลักษณะดังกล่าวโดยการติดต่อจะอยู่ในรูปของ PS/2 เมาส์ก็จะเป็นอุปกรณ์ชี้ในโปรแกรมที่เป็นกราฟฟิกชนิดหนึ่งที่ติดต่อโดยผ่านพอร์ทในรูปแบบของ PS/2 อุปกรณ์ที่ติดต่อไม่จำเป็นต้องเป็นเมาส์เท่านั้นปากกาแสงหรือเซ็นเซอร์ต่างๆสามารถใช้เป็นอุปกรณ์อินพุทได้ทั้งนั้นโดยผ่านพอร์ทที่ตำแหน่ง 60h และ 64h รูปแบบการส่งข้อมูลแบบ PS/2 เป็นดังรูปที่ 3.15

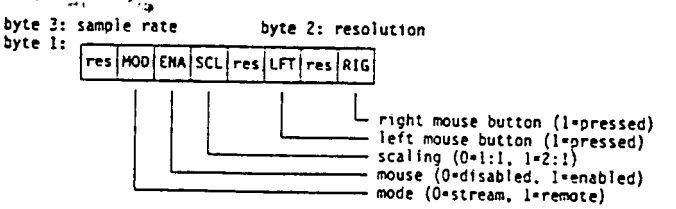


รูปที่ 3.15 แสดงเฟรมข้อมูลของ PS/2 เมาส์หน้า 692

3.5.1 การเขียนโปรแกรม PS/2 เมาส์ผ่านพอร์ต

การส่งข้อมูลหรือคำสั่งให้กับเมาส์(อุปกรณ์อินพุทและเอาท์พุทอื่นๆ) ลำดับแรกต้องส่งค่าให้กับคีย์บอร์ดคอนโทรลเลอร์โดยใช้คำสั่งไรท์ที่ตำแหน่งแอดเดรส 64h ก่อนหลังจากนั้นให้เขียนข้อมูลลงที่แอดเดรส 60h อุปกรณ์ที่อินพุทเอาท์พุทดังกล่าวจะถูกแยกแยะตั้งแต่การเขียนคำสั่งลงที่รีจิสเตอร์ควบคุมแล้ว

Code	Command	Description
e6h	reset scaling	resets X-Y-scaling factor to 1:1
e7h	set scaling	sets X-Y-scaling to 2:1
e8h	set resolution	sets mouse resolution corresponding to following data byte (00h=1 count/mm, 01h=2 counts/mm, 02h=4 counts/mm, 03h=8 counts/mm)
e9h	determine status	supplies a 3-byte status:



รูปที่ 3.16 แสดงรีจิสเตอร์ควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

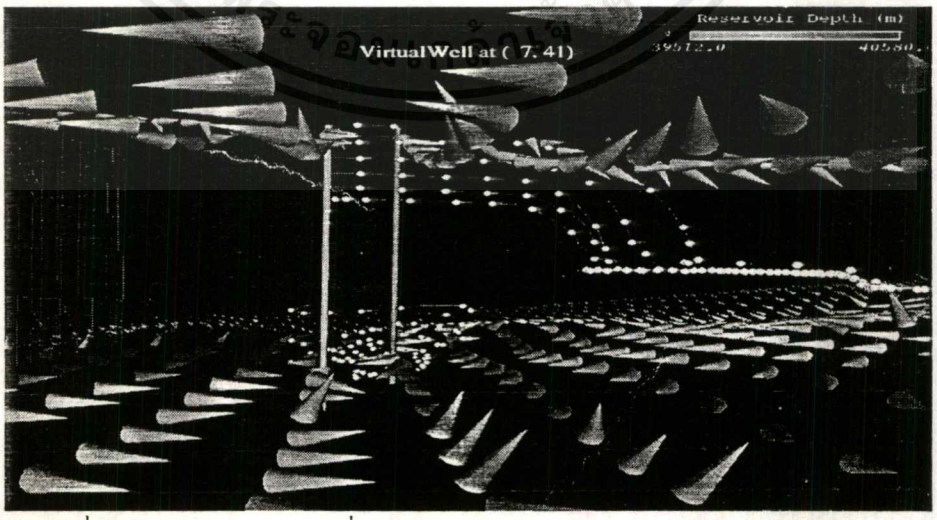
บทที่ 4

การเขียนโปรแกรมสร้างภาพ 3 มิติโดยใช้ โอเพนจีแอล(OPENGL)

โอเพนจีแอลย่อมาจากโอเพนกราฟิคไลบรารี(Open Graphics Library) เป็นไลบรารีที่ช่วยในการเขียนโปรแกรมสร้างภาพ 3 มิติ ที่พัฒนาโดยบริษัทซิลิคอนกราฟิค(Silicon Graphics Inc.) ซึ่งเป็นบริษัทที่พัฒนาระบบสร้างภาพสามมิติระดับสูงบนเครื่องเวอคสเตชั่น(Workstation) โดยใช้ในงานสร้างภาพยนตร์และโฆษณา ต่อมาทางบริษัทไมโครซอฟท์ได้ซื้อลิขสิทธิ์เพื่อนำ โอเพนจีแอลมาใช้บนเครื่องพีซีโดยรันภายใต้ระบบปฏิบัติการวินโดว์(Windows) โดยให้มากับโปรแกรมระบบปฏิบัติการวินโดว์เอนทีเวอร์ชั่น 4.0 ในรูปของไดนามิคลิงค์ไลบราลี (DLL) ซึ่งผู้ใช้งานสามารถนำไปใช้ในการเขียนโปรแกรมสร้างภาพสามมิติโดยใช้โปรแกรมภาษาซีที่ทำงานบนระบบปฏิบัติการวินโดว์เช่น ไมโครซอฟท์วิชวลซีพลัสพลัส(Microsoft Visual C++) บอร์แลนด์ซีพลัสพลัส (Borland C++)ทำให้ได้ภาพสามมิติที่มีคุณภาพสูงและสามารถทำงานได้อย่างรวดเร็ว

4.1 ลักษณะโดยทั่วไปของโอเพนจีแอล

เป็นไลบรารีที่มีลักษณะเป็นไดนามิคลิงค์ไลบรารี สำหรับลิงค์กับโปรแกรมที่เขียนด้วยภาษาซีบนระบบปฏิบัติการวินโดว์ ประกอบด้วยคำสั่งจัดการเกี่ยวกับการสร้างภาพสามมิติ เช่น โพลีกอน(Polygon) , ทรงกลม (Sphere) , การให้แสง(Lighting), เงา(Shadow),การทำอนิเมชั่น(Animation),การสร้างเอฟเฟคต่างๆอีกมาก ซึ่งสามารถเรียกใช้ได้เช่นเดียวกับคำสั่งในไลบรารีทั่วไปของภาษาซี ซึ่งช่วยให้ผู้ที่มีความรู้พื้นฐานการเขียนโปรแกรมภาษาซีบนระบบปฏิบัติการวินโดว์สามารถ เขียน โปรแกรมสร้างภาพสามมิติได้สะดวกและให้ผลลัพธ์เป็นที่น่าพอใจมาก (ควรจะมีพื้นฐานการเขียนโปรแกรมภาษาซีบนระบบปฏิบัติการวินโดว์ดีพอสมควร)ดังรูปที่ 4.1



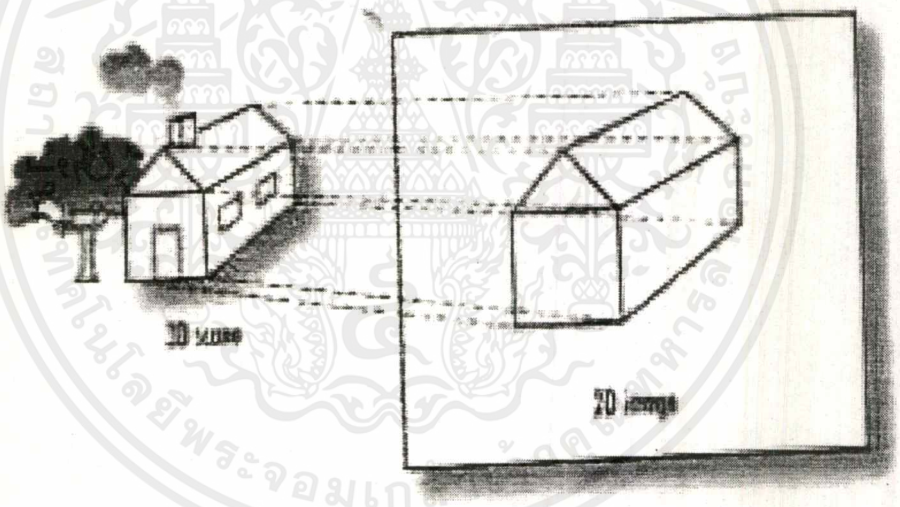
รูปที่ 4.1 แสดงภาพสามมิติที่สร้างโดยโปรแกรมภาษาซีโดยอาศัยโอเพนจีแอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปัจจุบันนี้โอเพนจีแอลออกเวอร์ชั่นล่าสุดคือ 1.1 ซึ่งสามารถดาวน์โหลด(Download) ผ่านอินเตอร์เน็ทมาใช้งานฟรีได้ที่ http://www.sgi.com/Technology/OpenGL/

4.2 หลักการสร้างภาพสามมิติเบื้องต้น

การสร้างภาพสามมิติบนจอคอมพิวเตอร์ในความเป็นจริงแล้วเป็นเพียงภาพสองมิติเพราะจอภาพของคอมพิวเตอร์เป็นจอที่มีลักษณะแบนให้ภาพได้เพียงสองมิติ ในการที่จะทำให้มองเห็นภาพสามมิติจริงๆภาพที่ตาทั้งสองข้างมองเห็นจะต้องมีมุมมองที่แตกต่างกันส่งผลให้สมองทำการแปรความหมายออกมาเป็นภาพที่มีความลึกเป็นสามมิติ ดังนั้นในการสร้างภาพสามมิติบนจอภาพคอมพิวเตอร์จึงต้องทำการทำให้ภาพที่เกิดบนจอมีความตื้นลึก ตัวอย่างเช่นวัตถุที่อยู่ไกลจะมีขนาดเล็กกว่าวัตถุที่อยู่ใกล้กว่าหรือการที่สามารถเปลี่ยนมุมมองที่ใช้ในการมองวัตถุได้(Perspective) การให้แสงฉายลงบนผิวด้านหนึ่งของวัตถุแล้วทำให้เกิดเงาขึ้นที่ด้านหนึ่งของวัตถุ เหล่านี้เรียกว่าการสร้างภาพสามมิติบนจอคอมพิวเตอร์ซึ่งมีลักษณะเป็น 2 มิติ ดังรูปที่ 4.2

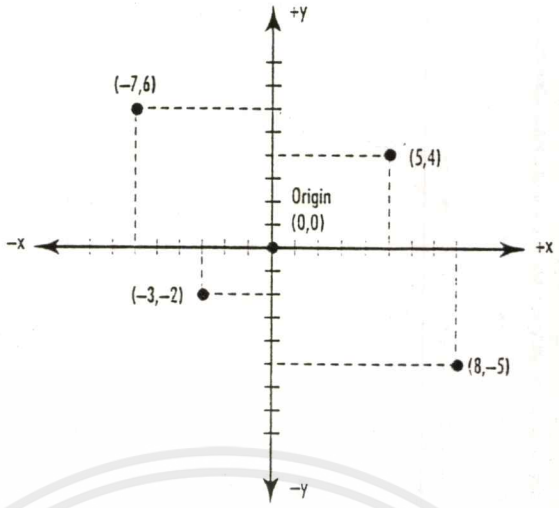


รูปที่ 4.2 แสดงการสร้างภาพสามมิติบนจอภาพคอมพิวเตอร์

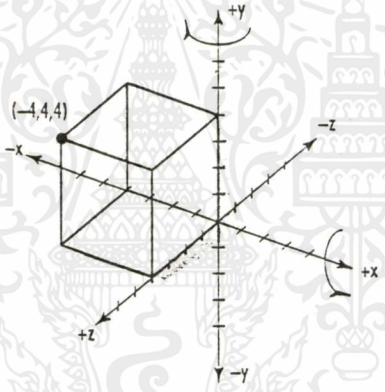
4.3 ระบบแกนและพิกัดในระบบสามมิติ

โดยปกติการเขียนโปรแกรมสร้างภาพกราฟิคบนระบบปฏิบัติการวินโดว์ในสองมิติจะมีเพียงสองแกนคือแกน Xและแกน Y โดยมีพิกัดดังรูปที่ 4.3 แต่ในการสร้างภาพสามมิติโดยอาศัยโอเพนจีแอลจะประกอบด้วยแกนสามแกนคือ X,Yและ Z โดยมีพิกัดแตกต่างจากปกติดังรูปที่4.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.3 แสดงระบบแกนในการสร้างภาพกราฟิคบนวินโดว์ในระบบ 2 มิติ

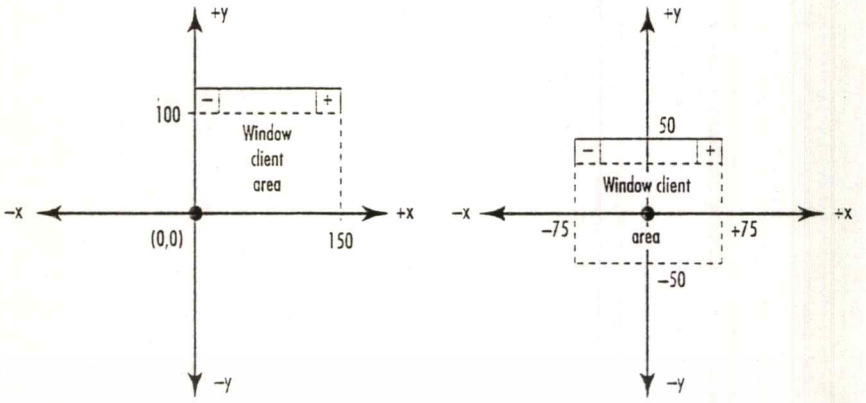


รูปที่ 4.4 แสดงระบบแกนในการสร้างภาพสามมิติโดยอาศัยโอเพนจีแอล

โคออดิเนตคลิป(Coordinate Clipping)

เป็นการตัดภาพที่เกินจากขอบเขตที่กำหนดในพิกัดที่ต้องการออกไปซึ่งทำให้มองเห็นเฉพาะภาพที่อยู่ในขอบเขตที่มองเห็นจึงช่วยให้การสร้างภาพมีความเร็วขึ้นเพราะสร้างภาพเฉพาะบริเวณที่มองเห็นเท่านั้นดังรูปที่ 4.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



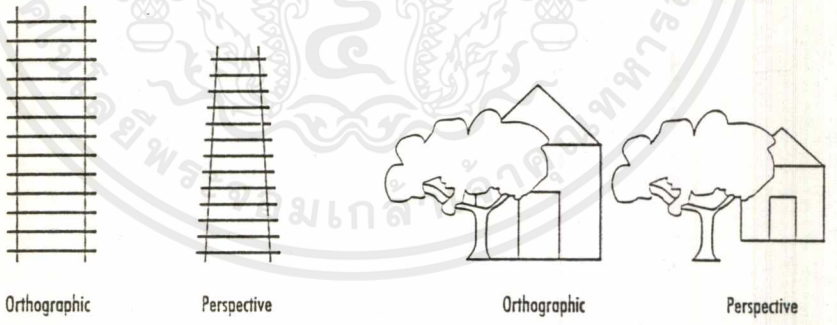
รูปที่ 4.5 แสดงการทำโคออดิเนตคลิป

โปรเจคชั่น(Projection)

เป็นการนำค่าโคออดิเนตที่สร้างได้ในระบบสามมิติมาเปลี่ยนเป็นพิกัดในสองมิติสร้างเป็นภาพที่สามารถปรากฏบนจอภาพในสองมิติการทำโปรเจคชั่นมีสองรูปแบบคือ

- ออร์โธกราฟิค(Orthographic Projection)

เป็นการสร้างภาพที่มีสัดส่วนความกว้างคงที่ไม่ว่าจะเป็นด้านบน,ล่าง,ซ้ายหรือขวารวมทั้งด้านลึก ซึ่งใช้ในงานด้านแคด(CAD)และการออกแบบวัตถุทั่วไปดูรูปที่ 4.6



รูปที่ 4.6 แสดงการโปรเจคชั่นในแบบออโธกราฟิคและแบบเปอร์สเปคทีฟ

- เปอร์สเปคทีฟ(Perspective Projection)

เป็นการสร้างภาพที่มีสัดส่วนความกว้างของภาพเปลี่ยนแปลงไปตามมุมมองที่เปลี่ยนแปลงไป ตัวอย่างเช่นการที่เรามองเห็นภาพที่อยู่ใกล้มีขนาดใหญ่กว่าภาพที่อยู่ในระยะไกล หรือ การมองเห็นภาพในมุมมองกว้างมีความลึกน้อยกว่าการมองภาพในมุมมองกว้างดังรูปที่ 4.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 โอเพนจีแอลกับการประยุกต์ใช้งาน

จากประโยชน์ของโอเพนจีแอลจะเห็นได้ว่าสามารถประยุกต์ใช้งานได้อย่างกว้างขวางและในปัจจุบันมีการนำโอเพนจีแอลมาสร้างเป็นภาษาคอมพิวเตอร์ขึ้นใหม่ เรียกว่า VRML(VIRTUAL REALITY MODELING LANGUAGE) ซึ่งนำมาใช้ในการสร้างโลกเสมือน(Virtual Word) เพื่อจำลองสถานการณ์ที่ต้องการในรูปแบบของสามมิติ ซึ่งVRMLช่วยให้การสร้างภาพสามมิติง่ายมากขึ้นโดยที่เราไม่จำเป็นที่ต้องมีความรู้การเขียนโปรแกรมภาษาซี เพียงแต่เรารู้คำสั่งต่างๆในVRML เราก็สามารถสร้างวัตถุขึ้นในสามมิติรวมทั้งโลกเสมือนเพื่อจำลองสถานการณ์ให้ผู้ใช้งานเสมือนเข้าไปมีส่วนร่วมในเหตุการณ์นั้นจริงๆดังรูปที่ 4.7

Virtual models of Airbus Industrie aircraft help designers at Europe's largest aircraft manufacturer share their visions.

รูปที่4.7 แสดงการประยุกต์ใช้งานโอเพนจีแอลกับระบบเวอชวลเรียลลิตีกับการออกแบบที่นั่งบนเครื่องบินโดยสาร โดยสร้างภาพจำลองของสภาพภายในหลังการออกแบบ ก่อนนำไปสร้างของจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 โครงงาน

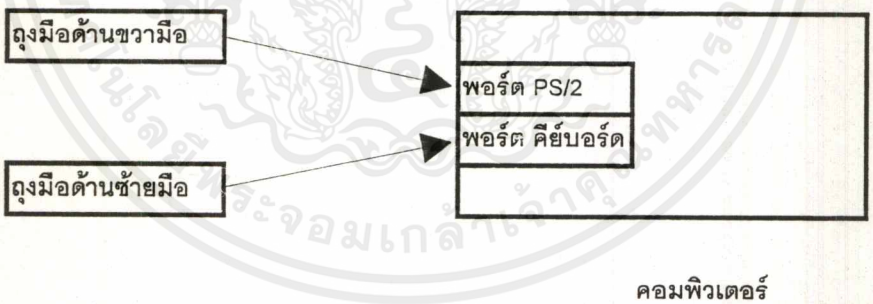
5.1 โครงสร้างโดยรวม

รูปที่ 5.1 โครงสร้างโดยรวมของโครงงาน

เซ็นเซอร์จะป้อนอินพุทให้กับคอมพิวเตอร์เพื่อจำลองการทำงานของแขนมนุษย์แล้วแสดงภาพออกมาเป็นกราฟฟิคเหมือนกับวัตถุนั้นๆ ในที่นี้วัตถุคือแขนมนุษย์พร้อมกันนั้นคอมพิวเตอร์จะส่งค่าเอาท์พุทให้กับแขนกลให้เคลื่อนไหวตามแขนดังกล่าว

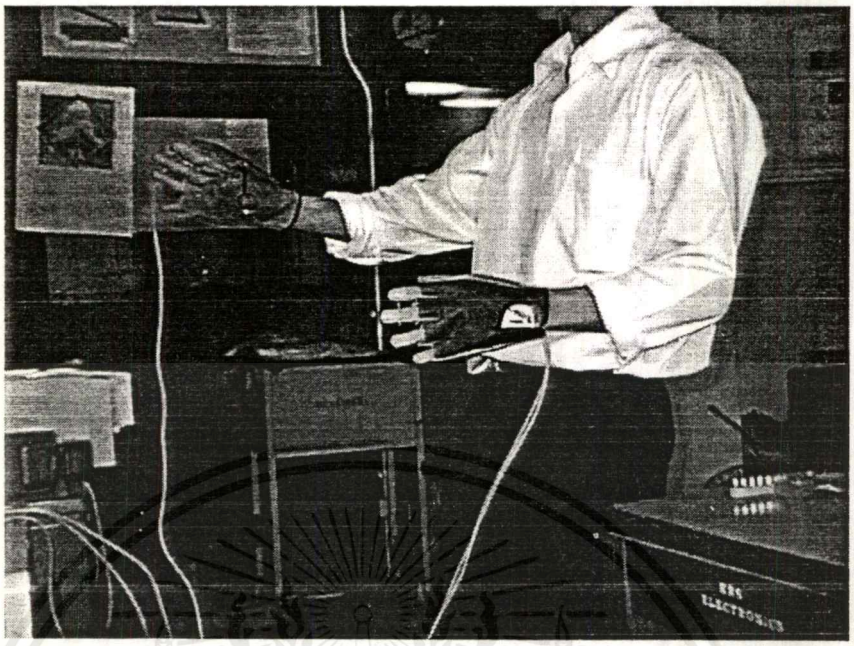
5.1.1 เซ็นเซอร์

เซ็นเซอร์ในที่นี้คือไจโรสโคปเซ็นเซอร์ที่ติดอยู่ในเมาส์และคีย์บอร์ดในที่นี้จะใช้ชิพเบอร์ 8049 ซึ่งภายในชิพจะมีโปรแกรมในการสแกนคีย์อยู่แล้วในโครงงานนี้ได้ใช้คีย์บอร์ดเพียง 4 คีย์เท่านั้น โดยเซนเซอร์ทั้งสองตัวจะติดอยู่ที่ถุงมือดังรูป



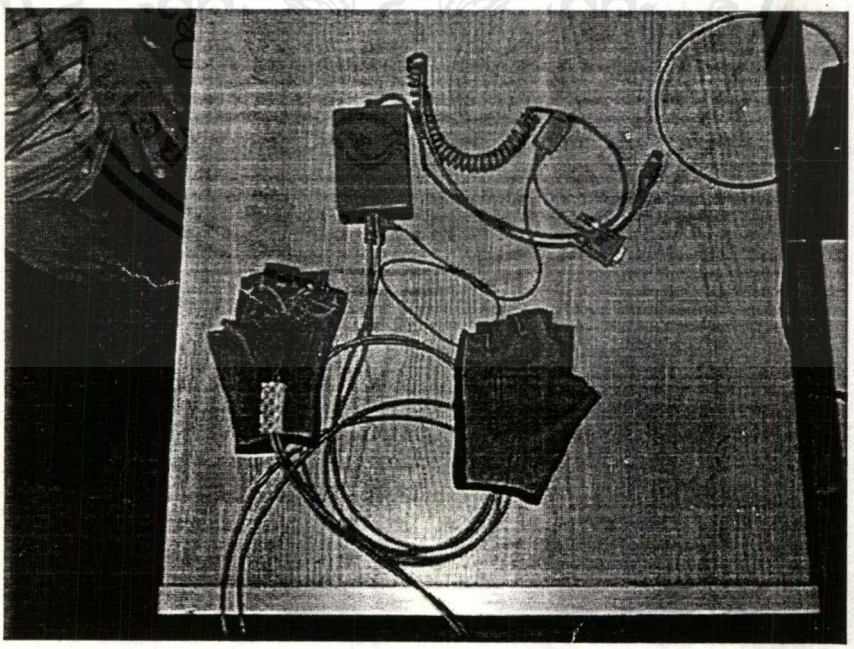
รูปที่ 5.2 แสดงการติดต่อของเซ็นเซอร์ทั้งสองจุดเชื่อมกับพอร์ตในคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.3 แสดงถุงมือที่ติดเซนเซอร์ไว้แล้ว

ด้านขวามือจะติดตั้งเมาส์ที่มีไจโรเซนเซอร์เพื่อรองรับการเคลื่อนไหวของแขน ส่วนซ้ายมือจะติดคีย์บอร์ดทั้งหมด 4 คีย์เพื่อใช้ในงานควบคุมอุปกรณ์อิเลคทรอนิกส์อื่นๆ

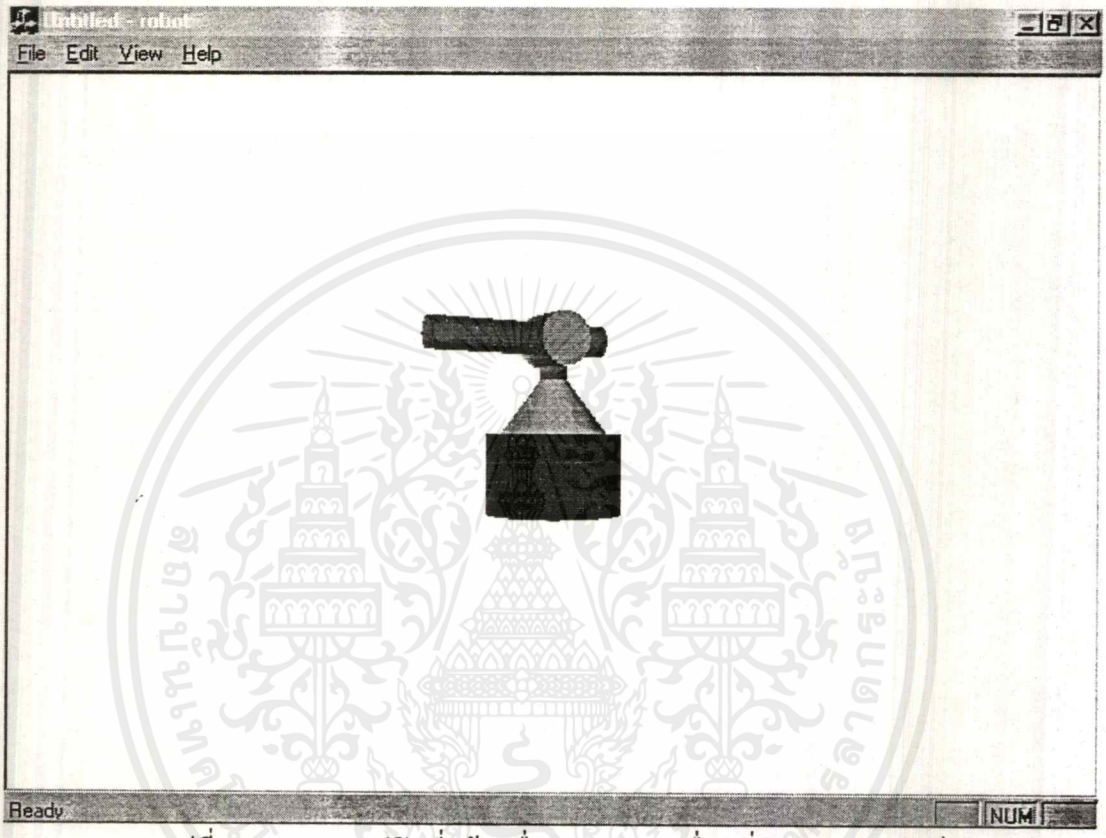


รูปที่ 5.4ถุงมือและอุปกรณ์ที่ใช้เป็นอินพุทให้กับคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.1.2 คอมพิวเตอร์กราฟฟิค

โปรแกรมคอมพิวเตอร์จะจำลองการทำงานของแขนมนุษย์โดยการรับอินพุทจากเซ็นเซอร์ ดังรูป



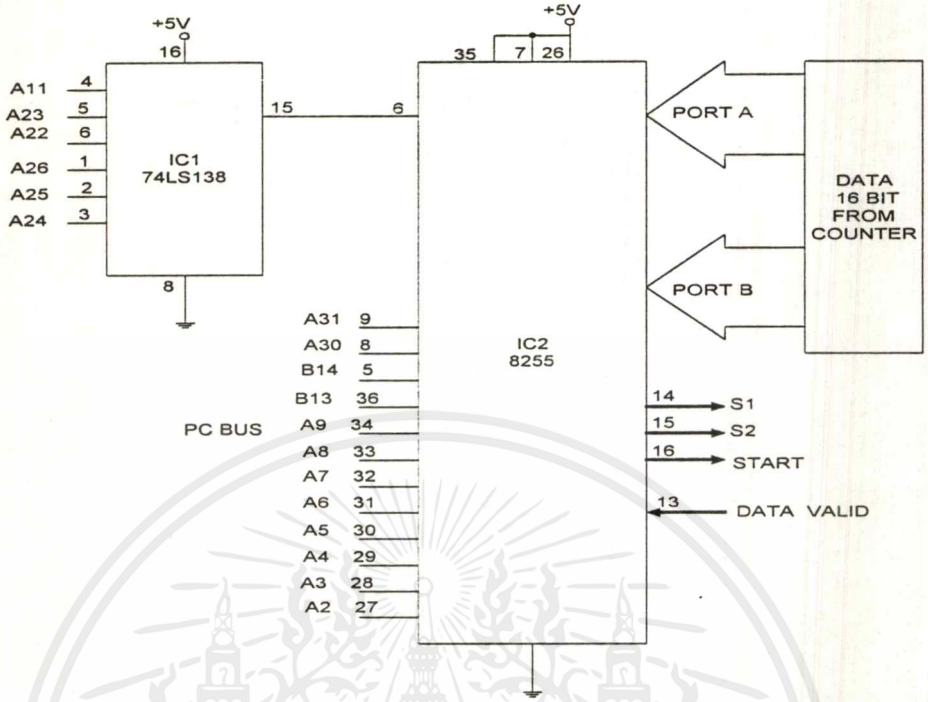
รูปที่ 5.5 แสดงกราฟฟิคที่สร้างเพื่อจำลองการเคลื่อนที่โดยคอมพิวเตอร์

5.1.3 การควบคุมแขนกล

ในการควบคุมการเคลื่อนที่ของแขนกลจะส่งข้อมูลโดยผ่านการ์ดอินเตอร์เฟสที่สร้างขึ้นโดยใช้ ไอซี 8255 และ 74LS138 ในการถอดรหัสแอดเดรสของเครื่องคอมพิวเตอร์ที่ แอดเดรส 280H ถึง 283H เพื่อใช้ในการส่งข้อมูลควบคุมสเต็ปปิ้งมอเตอร์ 2 ตัวดังวงจรแสดงดังรูปที่ 5.6

เนื่องจากสเต็ปเปอร์มอเตอร์มีเพียงสองตัวใช้ในแขนที่มีเพียงสองข้อต่อทำให้แขนสามารถเคลื่อนในแนวแกน X คือวนรอบฐานได้ 360 องศา และในแนวแกน Y กำหนดให้เคลื่อนที่ได้ 180 องศาดังรูปที่ 5.7

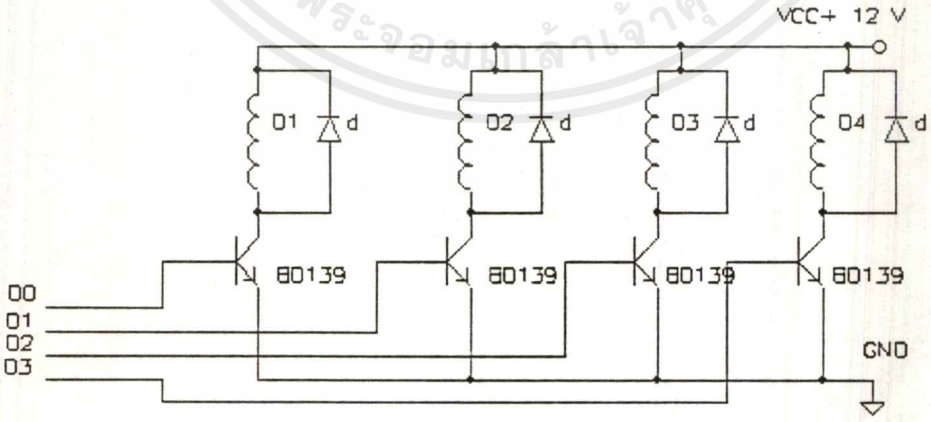
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.6 แสดงวงจรการ์ดอินเตอร์เฟสที่สร้างขึ้น

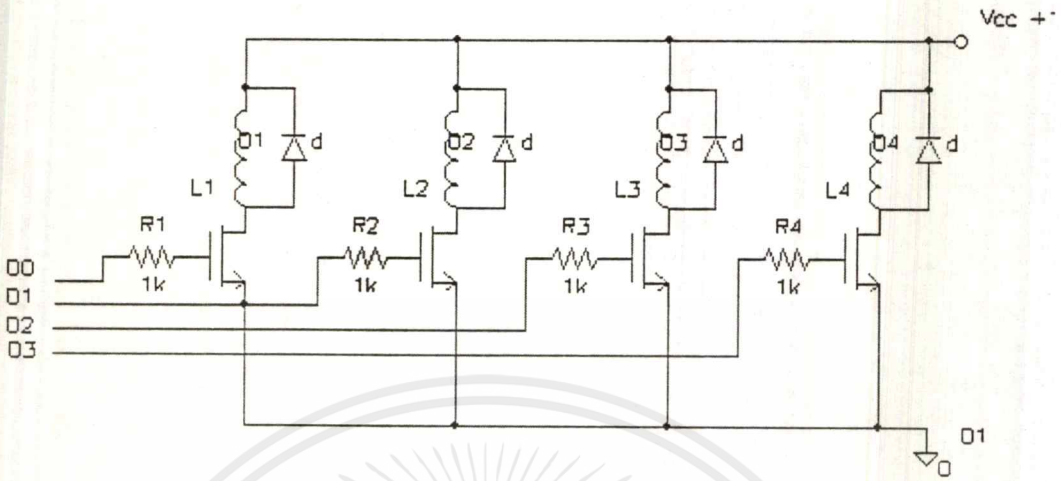
รูปที่ 5.7 บล๊อกการทำงานของแขนกลที่ควบคุมโดยคอมพิวเตอร์

5.1.4 วงจรไดรว์มอร์เตอร์



รูปที่ 5.8 วงจรการต่อวงจรการขับสเตปเปอร์มอเตอร์โดยใช้ทรานซิสเตอร์
ขา D0-D3 ต่อกับการ์ดอินเทอร์เฟส

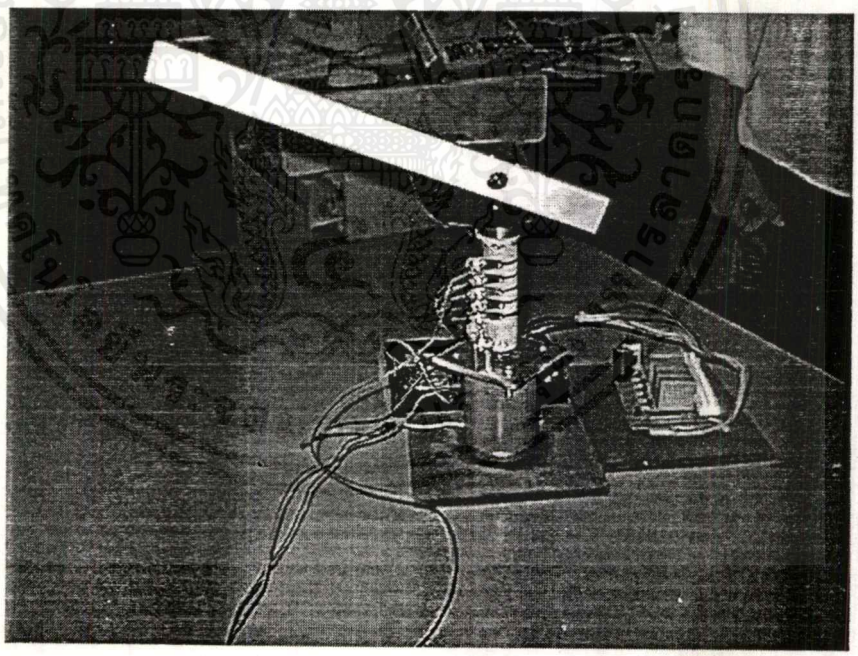
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 5.9 วงจรขับสเตปเปอร์มอเตอร์โดยใช้มอสเฟส

5.1.5โครงสร้างของแขนกล

รูปแสดงแขนกลที่มีจุดการเคลื่อนไหวได้สองจุด



รูปที่ 5.10 **แขนกลอย่างง่ายๆที่มีจุดต่อเพียงสองจุด**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

การทดลองและผลการทดลอง

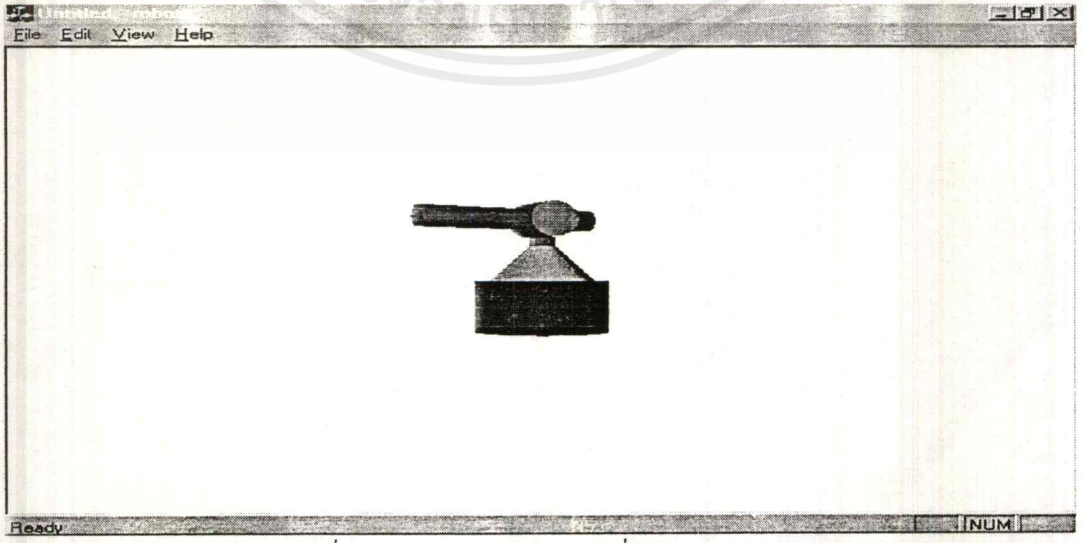
จากการทดลองสร้างอุปกรณ์อินพุทโดยใช้ระบบวัดระยะในด้วยอัลตราโซนิคในบทที่ 2 พบว่ามีข้อเสียดังนี้

1. การผิดพลาดของตำแหน่งมีค่ามากเนื่องจากความเร็วของเสียงในอากาศมีการเปลี่ยนแปลงตามอุณภูมิทำให้ตำแหน่งที่อ่านได้มีค่าไม่คงที่
2. ความละเอียดของตำแหน่งมีค่าน้อย เช่น การเคลื่อนที่ของมือเพียงเล็กน้อยบางครั้งไม่สามารถตรวจจับได้

ในการใช้งานเป็นอุปกรณ์อินพุทสำหรับระบบเวอชวลเรียลลิตี้ต้องการความแม่นยำสูงดังนั้นในโครงงานจึงทดลองใช้ไจโรสโคปเซ็นเซอร์ในการสร้างโครงงานแทนพบว่าสามารถใช้งานได้ดีและมีความแม่นยำสูงกว่าการใช้อัลตราโซนิค อีกทั้งยังสามารถเชื่อมต่อกับคอมพิวเตอร์ได้สะดวกโดยผ่านรูปแบบการส่งของเมาส์และคีย์บอร์ดทำให้สามารถเขียนโปรแกรมอินเตอร์เฟสโดยรับข้อมูลมาสร้างกราฟฟิคจำลองการเคลื่อนที่ของแขนและยังสามารถนำขอมูลที่ได้ไปควบคุมการเคลื่อนที่ของแขนกลได้

6.1 ผลการทดลองในส่วนของโปรแกรม

โปรแกรมอินเตอร์เฟสและสร้างภาพสามมิติจำลองการเคลื่อนที่ของแขน เขียนโดยใช้ไมโครซอฟท์วิชวลซีพลัสพลัส เวอร์ชั่น 4.2 เขียนรับข้อมูลผ่านเมาส์และคีย์บอร์ดเพื่อตอบสนองต่อการเคลื่อนที่ของมือ โดยโปรแกรมจะสร้างรูปจำลองของแขนกลขึ้นบนจอภาพซึ่งสามารถเคลื่อนไหวตามลักษณะการเคลื่อนที่ของแขนดังรูปที่ 6.1



รูปที่ 6.1 แสดงผลการทดลองที่ได้จากการเขียนโปรแกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

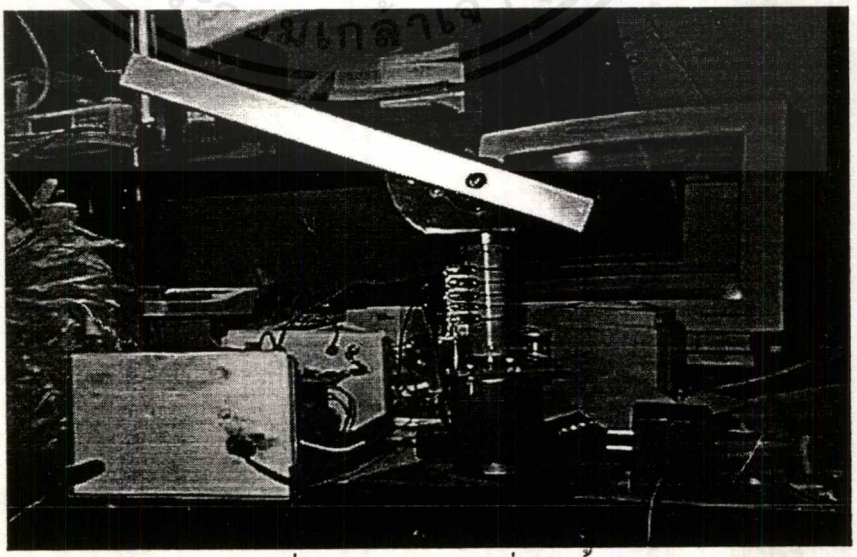
6.2 ผลการทดลองในส่วนของฮาร์ดแวร์

การเขียนโปรแกรมอินเตอร์เฟสผ่านวินโดว์โดยใช้การ์ดอินเตอร์เฟสต้องมีการเขียนโปรแกรมไดรเวอร์ซึ่งต้องใช้เวลาในการศึกษาค่อนข้างนานและค่อนข้างยุ่งยาก ดังนั้นโปรแกรมนี้จึงใช้การติดต่อกับพอร์ตโดยตรงโดยใช้คำสั่ง OUTP และ INP ในภาษาซี ซึ่งมีอยู่ในไลบรารีของไมโครซอฟท์วิชวลซี 4.2 ซึ่งยังไม่เป็นการถูกต้องนักในการเขียนโปรแกรมบนวินโดว์เพื่อติดต่อกับฮาร์ดแวร์โดยตรง

จากการทดลองโดยสร้างการ์ดอินพุทและเอาท์พุทโดยใช้ IC 8255 และ 74LS138 อย่างละ 1 ตัวในการถอดรหัสพอร์ท ทำให้สร้างพอร์ทได้ 3 พอร์ทโดยตั้งค่าพอร์ทเริ่มต้นที่แอดเดรส 280H,281H,282H โดยมีแอดเดรสของพอร์ทคอนโทรลอยู่ที่ 283H พบว่าสามารถส่งข้อมูลผ่านพอร์ต 280H ขนาด 8 บิท เพื่อใช้ในการสั่งงานอุปกรณ์ไฟฟ้าต่างได้ ดังนั้นจึงสามารถนำข้อมูลนี้มาควบคุมการหมุนของสเต็ปปิ้งมอเตอร์ได้โดยการส่งข้อมูลควบคุมทีละ 4 บิทต่อมอเตอร์ 1 ตัว ดังนั้นเพื่อให้มอเตอร์ทั้งสองตัวทำงานพร้อมกันครบรอบจึงต้องทำการส่งข้อมูลแยกกันคนละพอร์ตดังนั้นจึงใช้พอร์ต 1 พอร์ตต่อมอเตอร์ 1 ตัว

- พอร์ท 280H ควบคุมการทำงานของสเต็ปปิ้งมอเตอร์ตัวที่ 1
- พอร์ท 281H ควบคุมการทำงานของสเต็ปปิ้งมอเตอร์ตัวที่ 2
- พอร์ท 282H ว่างใช้ในการควบคุมอุปกรณ์อื่นๆได้ เช่น โคมไฟ

จากนั้นทำการส่งข้อมูลเพื่อสั่งให้มอเตอร์หมุนพบว่าต้องมีการหน่วงเวลาในระหว่างการส่งข้อมูลในแต่ละสเต็ปของการเคลื่อนที่เพื่อให้มอเตอร์มีการเคลื่อนที่อย่างราบเรียบ(Smooth) พบว่าค่าการหน่วงเวลาที่เหมาะสมคือประมาณ 100 ms ซึ่งในช่วงที่หน่วงเวลานี้จะใช้ในการสร้างภาพกราฟฟิคแขนกลที่สัมพันธ์กับการเคลื่อนที่ของแขนของผู้ใช้



รูปที่ 6.2 แสดงแขนกลที่สร้างขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7
บทสรุป

เวอชวลเรียลลิตี้เป็นสาขาที่ถูกนำไปใช้งานอย่างกว้างขวางในหลายๆด้านซึ่งในอนาคตระบบเวอชวลเรียลลิตี้จะเข้ามามีบทบาทอย่างมากกับชีวิตประจำวัน ในโครงงานนี้จึงเป็นการกระตุ้นให้เกิดการศึกษาเกี่ยวกับระบบเวอชวลเรียลลิตี้เพื่อใช้งานในอีกด้านหนึ่ง โดยแสดงให้เห็นถึงการประยุกต์ใช้งานระบบเวอชวลเรียลลิตี้กับการควบคุมการเคลื่อนที่ของแขนกลโดยใช้ไจโรสโคปเซ็นเซอร์ตรวจจับการเคลื่อนที่ของมือเป็นข้อมูลป้อนเข้าไปยังคอมพิวเตอร์จากนั้นคอมพิวเตอร์จะนำข้อมูลที่ได้ไปสร้างภาพจำลองการเคลื่อนที่ของแขนบนหน้าจอคอมพิวเตอร์นอกจากนั้นยังนำการเคลื่อนที่ของมือที่ตรวจจับได้ไปใช้ในการควบคุมสเต็ปปิ้งมอเตอร์ที่ควบคุมการทำงานแขนกลหรืออุปกรณ์อื่นๆอีกได้โดยผ่านการ์ดอินเตอร์เฟสที่สร้างขึ้น

ปัญหาและอุปสรรค

1. เนื่องจากเป็นสาขาที่ยังใหม่ดังนั้นข้อมูลทางด้านนี้จึงหาได้ยาก
2. อุปกรณ์ที่ใช้ในการทำโครงงานมีน้อยอีกทั้งราคาแพง บางอย่างไม่สามารถหาซื้อได้ในประเทศ เช่น เซ็นเซอร์ไจโรสโคปต้องสั่งซื้อไปที่ประเทศอเมริกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

ในภาคผนวกนี้เป็นซอร์ซโค้ดในส่วนของโปรแกรมหลักที่เขียนขึ้นบนวิชวลซีพลัสพลัสในส่วนของโปรแกรมสร้างภาพกราฟฟิคสามมิติและส่วนควบคุมแขนกล ถ้าผู้อ่านสนใจซอร์ซโค้ดโปรแกรมสมบูรณ์หรือมีปัญหาในการเขียนโปรแกรมสามารถติดต่อกับผู้เขียนได้ที่

EmailAddress : pongchet@hotmail.com

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รายละเอียดของโปรแกรม

```
// robotView.cpp : implementation of the CRobotView class
//
#include "stdafx.h"
#include "robot.h"
#include "robotDoc.h"
#include "robotView.h"

#ifdef _DEBUG
#define new DEBUG_NEW
#undef THIS_FILE
static char THIS_FILE[] = __FILE__;
#endif

/////////////////////////////////////////////////////////////////////////////
// CRobotView

IMPLEMENT_DYNCREATE(CRobotView, CView)

BEGIN_MESSAGE_MAP(CRobotView, CView)
        //{{AFX_MSG_MAP(CRobotView)
        ON_WM_CREATE()
        ON_WM_SIZE()
  ON_WM_QUERYNEWPALETTE()
  ON_WM_PALETTECHANGED()
        ON_WM_LBUTTONDOWN()
        ON_WM_LBUTTONUP()
        ON_WM_MOUSEMOVE()
        ON_WM_KEYDOWN()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        //}}AFX_MSG_MAP
END_MESSAGE_MAP()
////////////////////////////////////////////////////////////////////////////
// CRobotView construction/destruction

CRobotView::CRobotView()
{
        // TODO: add construction code here
   m_nUpAngle = 360.0f;
   m_nBaseAngle = 0.0f;
   m_hPalette = NULL;
        m_bCaptured=FALSE;
        xRot = 0;
        yRot = 0;
}
CRobotView::~CRobotView()
{
 HGLRC hrc = wglGetCurrentContext();
   if(hrc)
   {
// Get the DC associated with the rendering context
     HDC hdc =  wglGetCurrentDC();
     wglMakeCurrent(NULL, NULL);

// Delete rendering context and release DC
     wglDeleteContext(hrc);
     ::ReleaseDC(GetSafeHwnd(), hdc);
   }
   if(m_hPalette != NULL)
       DeleteObject(m_hPalette);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//    ::KillTimer(GetSafeHwnd(), m_nTimer);
}
BOOL CRobotView::PreCreateWindow(CREATESTRUCT& cs)
{
        // TODO: Modify the Window class or styles here by modifying
        //  the CREATESTRUCT cs
  cs.style |= WS_CLIPSIBLINGS | WS_CLIPCHILDREN;

   return CView::PreCreateWindow(cs);
}
/////////////////////////////////////////////////////////////////////////////
// CRobotView drawing

void CRobotView::OnDraw(CDC* pDC)
{
        CRobotDoc* pDoc = GetDocument();
        ASSERT_VALID(pDoc);

        // TODO: add draw code for native data here
  DrawScene();
}

/////////////////////////////////////////////////////////////////////////////
// CRobotView diagnostics

#ifdef _DEBUG
void CRobotView::AssertValid() const
{
        CView::AssertValid();
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void CRobotView::Dump(CDumpContext& dc) const
{
	CView::Dump(dc);
}
CRobotDoc* CRobotView::GetDocument() // non-debug version is inline
{
	ASSERT(m_pDocument->IsKindOf(RUNTIME_CLASS(CRobotDoc)));
	return (CRobotDoc*)m_pDocument;
}
#endif //_DEBUG

/////////////////////////////////////////////////////////////////////////////
// CRobotView message handlers

int CRobotView::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
	if (CView::OnCreate(lpCreateStruct) == -1)
		return -1;

	// TODO: Add your specialized creation code here
	HDC hdc = ::GetDC(GetSafeHwnd());
  SetDCPixelFormat(hdc);

  HGLRC hrc = wglCreateContext(hdc);
  wglMakeCurrent(hdc, hrc);
  InitRC();

//   m_nTimer = ::SetTimer(GetSafeHwnd(), 1, 50, NULL);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        return 0;
}

void CRobotView::OnSize(UINT nType. int cx, int cy)
{
        CView::OnSize(nType, cx, cy);
        // TODO: Add your message handler code here
        m_glnWidth = (GLsizei) cx;
  m_glnHeight = (GLsizei) cy;
  m_gldAspect = (GLdouble) m_glnWidth / (GLdouble) m_glnHeight;
  glMatrixMode(GL_PROJECTION);
  glLoadIdentity();
  gluPerspective(45.0, m_gldAspect, 1.0, 100.0);
  glViewport(0, 0, m_glnWidth, m_glnHeight);
}
BOOL CRobotView::OnQueryNewPalette()
{
  HDC hdc = wglGetCurrentDC();
  int n = 0;
  if(m_hPalette != NULL && hdc)
  {
    if((n = RealizePalette (hdc)))
      InvalidateRect(NULL, FALSE);
  }
  return n;
}
//---------------------------------------------------------------
void CRobotView::OnPaletteChanged(CWnd* pFocusWnd)
{
  if((m_hPalette != NULL) && (pFocusWnd != this))
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    {
      HDC hdc = wglGetCurrentDC();
      if(hdc && RealizePalette (hdc))
        UpdateColors (hdc);
    }
}

// SetDCPixelFormat sets the pixel format for a device context in
// preparation for creating a rendering context.

void CRobotView::SetDCPixelFormat(HDC hdc)
{
  HANDLE     hHeap;
  int        nColors, i;
  LPLOGPALETTE lpPalette;
  BYTE       byRedMask, byGreenMask, byBlueMask;

  static PIXELFORMATDESCRIPTOR pfd =
  {
    sizeof(PIXELFORMATDESCRIPTOR),  // Size of this structure
    1,                              // Version number
    PFD_DRAW_TO_WINDOW |            // Flags
    PFD_SUPPORT_OPENGL |
    PFD_DOUBLEBUFFER,
    PFD_TYPE_RGBA,                  // Use RGBA pixel values
    24,                             // Try to use 24-bit color
    0, 0, 0, 0, 0, 0,               // Don't care about these
    0, 0,                           // No alpha buffer
    32, 0, 0, 0, 0,                 // 32-bit accumulation buffer
    32,                             // 32-bit depth buffer
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    0,                  // No stencil buffer
    0,                  // No auxiliary buffers
    PFD_MAIN_PLANE,         // Layer type
    0,                  // Reserved (must be 0)
    0, 0, 0              // No layer masks
};
int nPixelFormat;
nPixelFormat = ChoosePixelFormat(hdc, &pfd);
SetPixelFormat(hdc, nPixelFormat, &pfd);
DescribePixelFormat(hdc, nPixelFormat,
            sizeof(PIXELFORMATDESCRIPTOR),
            &pfd);

if(pfd.dwFlags & PFD_NEED_PALETTE)
{
    nColors = 1 << pfd.cColorBits;
    hHeap = GetProcessHeap();

    lpPalette = (LPLOGPALETTE) HeapAlloc(hHeap, 0,
            sizeof(LOGPALETTE) + (nColors *
                sizeof(PALETTEENTRY)));
    lpPalette->palVersion = 0x300;
    lpPalette->palNumEntries = nColors;
    byRedMask = (1 << pfd.cRedBits) - 1;
    byGreenMask = (1 << pfd.cGreenBits) - 1;
    byBlueMask = (1 << pfd.cBlueBits) - 1;
    for(i = 0; i < nColors; i++)
    {
        lpPalette->palPalEntry[i].peRed =
            (((i >> pfd.cRedShift) & byRedMask) * 255) /
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                        byRedMask;
            lpPalette->palPalEntry[i].peGreen =
              (((i >> pfd.cGreenShift) & byGreenMask) * 255) /
                                        byGreenMask;
            lpPalette->palPalEntry[i].peBlue =
              (((i >> pfd.cBlueShift) & byBlueMask) * 255) /
                                        byBlueMask;
            lpPalette->palPalEntry[i].peFlags = 0;
          }

  // Create the palette and free the allocated memory
          m_hPalette = CreatePalette(lpPalette);
          HeapFree(hHeap, 0, lpPalette);

  // Realize the color palette
          if(m_hPalette != NULL)
          {
            SelectPalette(hdc, m_hPalette, FALSE);
            RealizePalette(hdc);
          }
      }
  }
  //  InitRC initializes the current rendering context.
  void CRobotView::InitRC()
  {
      GLfloat glfLightAmbient[] = { 0.1f, 0.1f, 0.1f, 1.0f };
      GLfloat glfLightDiffuse[] = { 0.7f, 0.7f, 0.7f, 1.0f };
      GLfloat glfLightSpecular[] = { 0.0f, 0.0f, 0.0f, 1.0f };

  // Initialize state variables.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    glFrontFace(GL_CCW);
    glCullFace(GL_BACK);
    glEnable(GL_CULL_FACE);
    glDepthFunc(GL_LEQUAL);
    glEnable(GL_DEPTH_TEST);
// Set background color to white
    glClearColor(1.0f, 1.0f, 1.0f, 0.0f);
// Add lights to the scene.
    glLightfv(GL_LIGHT0, GL_AMBIENT, glfLightAmbient);
    glLightfv(GL_LIGHT0, GL_DIFFUSE, glfLightDiffuse);
    glLightfv(GL_LIGHT0, GL_SPECULAR, glfLightSpecular);
    glEnable(GL_LIGHTING);
    glEnable(GL_LIGHT0);
}
//-------------------------------------------------------------
void CRobotView::DrawCone(GLdouble gldBaseRadius,
                          GLdouble gldTopRadius,
                          GLdouble gldHeight, UINT nSlices)
{
    GLUquadricObj *glquad;
    glquad = gluNewQuadric();
    gluCylinder(glquad, gldBaseRadius, gldTopRadius,
                gldHeight, nSlices, 1);
// Draw the end caps for the cone
    gluQuadricOrientation(glquad, GLU_INSIDE);
    gluDisk(glquad, 0.0, gldBaseRadius, nSlices, 1);
    glTranslatef(0.0f, 0.0f, (GLfloat) gldHeight);
    gluQuadricOrientation(glquad, GLU_OUTSIDE);
    gluDisk(glquad, 0.0, gldTopRadius, nSlices, 1);
    gluDeleteQuadric(glquad);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
void CRobotView::DrawSphere(GLdouble gldRadius,
                    GLint glnSlices, GLint glnStacks)
{
   GLUquadricObj *glquad;
   glquad = gluNewQuadric();
   gluSphere(glquad, gldRadius, glnSlices, glnStacks);
   gluDeleteQuadric(glquad);
}
void CRobotView::DrawScene()
{
// Define a few colors. In OpenGL, RGB values are specified as
// floating-point numbers between 0.0 and 1.0.

   GLfloat glfBlue[] = { 0.0f, 0.0f, 1.0f, 1.0f };
   GLfloat glfYellow[] = { 1.0f, 1.0f, 0.0f, 1.0f };
   GLfloat glfGreen[] = { 0.0f, 1.0f, 0.0f, 1.0f };
   GLfloat glfRed[] = { 1.0f, 0.0f, 0.0f, 1.0f };
   GLfloat glfCyan[] = { 0.0f, 1.0f, 1.0f, 1.0f };

// Clear the color and depth buffers.
   glClear(GL_COLOR_BUFFER_BIT | GL_DEPTH_BUFFER_BIT);
   glMatrixMode(GL_MODELVIEW);
   glLoadIdentity();
   glTranslatef(0.0f, 0.0f, -64.0f);
        glRotatef(xRot,1.0f,0.0f,0.0f);
        glRotatef(yRot,0.0f,1.0f,0.0f);

// A Big Cylinder
        glPushMatrix();
```

- เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
        glEnable(GL_COLOR_MATERIAL);
        glBlendFunc(GL_SRC_ALPHA,GL_ONE_MINUS_SRC_ALPHA);
        glEnable(GL_BLEND);
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, glfRed);
    glTranslatef(-18.0f,0.0f,-5.0f);
        DrawCone(5.0,5.0,15.0, 20);
        //glDisable(GL_BLEND);
        glDisable(GL_COLOR_MATERIAL);
    glPopMatrix();

  // Draw a cylindrical base.
    glPushMatrix();
    glRotatef(90.0f, 1.0f, 0.0f, 0.0f);
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, glfRed);
    DrawCone(5.0,5.0,6.0, 20);
    glPopMatrix();

   glRotatef(m_nBaseAngle, 0.0f, 1.0f, 0.0f);
  // Next a cylindrical cone on top of the base.
    glPushMatrix();
    glRotatef(-90.0f, 1.0f, 0.0f, 0.0f);
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, glfYellow);
    DrawCone(4.0, 1.0, 4.0, 20);
    glPopMatrix();
  // A Joint
    glPushMatrix();
        glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, glfCyan);
    glTranslatef(0.0f,4.0f,0.0f);
        DrawSphere(2.0,20, 8);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        glPopMatrix();
// A cylindrical arm
    glPushMatrix();
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, glfBlue);
    glTranslatef(0.0f,5.0f, 0.0f);
    glRotatef(m_nUpAngle, 1.0f, 0.0f, 0.0f);
        DrawCone(1.0, 1.0, 10.0,20);
// A Joint
        glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, glfCyan);
        DrawSphere(2.0,20, 8);
// Next cylindrical segment
    glRotatef(m_nUpAngle /6.0f, 1.0f, 0.0f, 0.0f);
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, glfBlue);
    DrawCone(1.0, 1.0, 8.0,20);
// A sphere at the end of the second cylindrical segment
    glMaterialfv(GL_FRONT, GL_AMBIENT_AND_DIFFUSE, glfCyan);
    DrawSphere(1.5,20, 8);
        glMaterialfv(GL_FRONT,GL_AMBIENT_AND_DIFFUSE,glfBlue);
        glTranslatef(-1.0f,0.6f,1.0f);
        DrawCone(0.4,0.2,2.0,5);
        glTranslatef(0.6f,0.0f,-2.0f);
        DrawCone(0.4,0.2,2.2,5);
        glTranslatef(0.6f,0.0f,-2.0f);
        DrawCone(0.4,0.2,1.8,5);
        glTranslatef(0.6f,0.0f,-2.0f);
        DrawCone(0.4,0.2,1.5,5);
        glPopMatrix();

// Render the scene in the window's DC
    SwapBuffers(wglGetCurrentDC());
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}
void CRobotView::OnLButtonDown(UINT nFlags, CPoint point)
{
        // TODO: Add your message handler code here and/or call default
        CRect clientRect;

        //      TRACE("entering COneview::OnLButtonDown-
point=%d,%d\n",point.x,point.y);
        GetClientRect(clientRect);
        if (clientRect.PtInRect(point)){
                SetCapture();
                m_bCaptured = TRUE;
                m_mousePos = point;
                ::SetCursor(::LoadCursor(NULL,IDC_CROSS));
        }
        CView::OnLButtonDown(nFlags, point);
}
void CRobotView::OnLButtonUp(UINT nFlags, CPoint point)
{
        // TODO: Add your message handler code here and/or call default
        if (m_bCaptured){
                ReleaseCapture();
                m_bCaptured = FALSE;
        }
        CView::OnLButtonUp(nFlags, point);
}
void CRobotView::OnMouseMove(UINT nFlags, CPoint point)
{
        // TODO: Add your message handler code here and/or call default
                CRect clientRect;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        if (m_bCaptured){
                GetClientRect(clientRect);
                if (clientRect.PtInRect(point)){
                        m_nBaseAngle += (point.x-m_mousePos.x)/3;
                        m_nUpAngle  += (point.y-m_mousePos.y)/5;
                        if(m_nUpAngle <= 179.0f)
                                m_nUpAngle = 180.0f;
                        if(m_nUpAngle >= 361.0f)
                                m_nUpAngle = 360.0f;
                        m_nBaseAngle = ((int)m_nBaseAngle) % 360;
                        Invalidate(FALSE);
                        m_mousePos =point;
                }
        }
        CView::OnMouseMove(nFlags, point);
}
void CRobotView::OnKeyDown(UINT nChar, UINT nRepCnt, UINT nFlags)
{
        // TODO: Add your message handler code here and/or call default
        if(nChar == VK_UP)
                xRot += 4;
        if(nChar == VK_DOWN)
                xRot -= 4;
        if(nChar == VK_LEFT)
                yRot += 4;
        if(nChar == VK_RIGHT)
                yRot -= 4;
        Invalidate(FALSE);
        CView::OnKeyDown(nChar, nRepCnt, nFlags);
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ชิ้นนี้จะไม่สามารถสำเร็จลุล่วงไปได้ถ้าไม่ได้รับการช่วยเหลือจากบุคคลต่อไปนี้

- คำปรึกษาและคำแนะนำที่มีค่าจากรศ.ดร.มนัส สังวรศิลป์
- เทคนิคและวิธีการสร้างแขนกลจาก เป้า ภาคเครื่องกล
- บริษัทมุราตะที่ให้ข้อมูลของไจโรสโคปเซ็นเซอร์
- อาจารย์ภาควิชาอิเลคทรอนิคส์ทุกท่านที่ให้คำแนะนำและติชม เพื่อเป็นแนวทางในการแก้ไขและปรับปรุงโครงงาน
- สมาชิกในกลุ่มและเพื่อนๆทุกคน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หนังสืออ้างอิง

Carl Eugene Loeffler and Tim Anderson "The Virtual Reality Construction Kit " McGRAW-HILL,1993

William Gibson"Virtual Reality Case Book",Prentice-Hall,1992

Loren Heiny,"Windows Graphics Programming with Borland C++"John Wiley & Sons Inc,1994

http://edge.stud.u-szeged.hu/sanner/

Richard S.Wright JR.Michael Sweet,"OpenGL Superbible",Waite Group Press,1996

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้