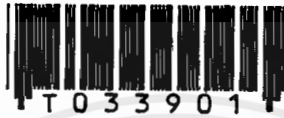


เครื่องสแกนพื้นผิว

Surface Scanning Machine



นาย วิชัย คัมภีรวุฒิ

MR. WICHAJ CUMPEERAWUT

นาย วรุทม์ ศิริประชา

MR. VARUTH SIRAPRACHA

นายรุ่งโรจน์ สิทธิวิเชียรวงศ์

MR. RUNGRODE SITIVICHIANWONG

ว.ร.  
ว.อ.  
อ.ร.

เลขหม.....
เลขทะเบียน..... 33901
วัน, เดือน, ปี..... 20 ก.ย. 2542

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมเครื่องกล

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2541

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิศวกรรมศาสตร์  
สถาบันพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
ใบรับรองปริญญาโท

หัวข้อปริญญาโท เครื่องสแกนพื้นผิว  
ชื่อนักศึกษา นาย วิชัย คัมภีร์วุฒิ รหัสประจำตัว 38014456  
นาย วรุตม์ ศิริประชา รหัสประจำตัว 38014443  
นาย รุ่งโรจน์ สิทธิวิเชียรวงศ์ รหัสประจำตัว 38014421  
หลักสูตร วิศวกรรมศาสตรบัณฑิต  
สาขาวิชา วิศวกรรมเครื่องกล  
อาจารย์ผู้ควบคุมปริญญาโท

(ดร.สรรพลิทธิ ลิ้มนรัตน์)

(อ.เอกพจน์ ตันตราภิวินัย)

(อ.อุดม จันทร์จรัสสุข)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ เครื่องสแกนพื้นผิว

ชื่อนักศึกษา นาย วิชัย คัมภีร์วุฒิ รหัสประจำตัว 38014456

นาย วรุตม์ ศิริประชา รหัสประจำตัว 38014443

นาย รุ่งโรจน์ สิทธิวิเชียรวงศ์ รหัสประจำตัว 38014421

หลักสูตร วิศวกรรมศาสตรบัณฑิต

สาขาวิชา วิศวกรรมเครื่องกล

อาจารย์ผู้ควบคุมปริญญานิพนธ์

(ดร.สรรพลิทธิ ลิ้มนรัตน์)

(อ.เอกพงษ์ ตันตราภิวัดน์)

(อ.อุดม จันทร์จรัสสุข)

### บทคัดย่อ

เนื่องจากในปัจจุบันนี้ คอมพิวเตอร์ได้เข้ามามีบทบาทสำคัญในด้านต่างๆ เช่น ด้านการออกแบบ การคำนวณ โครงสร้าง การวัด การแพทย์ และยังมีบทบาทที่สำคัญอย่างอื่นอีกมาก โดยเฉพาะในเรื่องของการออกแบบนั้น ได้มีการวิจัยและพัฒนาอย่างต่อเนื่อง เพื่อให้เกิดการใช้ประโยชน์จากคอมพิวเตอร์อย่างสูงสุด ไม่ว่าจะเป็นทางด้านการอำนวยความสะดวกในการออกแบบ การคำนวณ และการประเมินผลขณะออกแบบ ดังนั้นวัตถุประสงค์ของโครงการนี้จึงมุ่งเน้นไปถึงการนำเอาความรู้จากด้านกลไกต่างๆ และความรู้ทางด้านคอมพิวเตอร์ โดยเฉพาะในเรื่องของซอฟต์แวร์ เข้าประยุกต์ใช้ด้วยกัน โดยมีการเชื่อมต่อเข้าด้วยกันอย่างเหมาะสมและผลที่ได้จะถูกนำมาแสดงบนหน้าจอคอมพิวเตอร์ ทำให้เราสามารถทราบถึงลักษณะต่างๆของพื้นผิวชิ้นงานที่เราไม่สามารถวัดได้ด้วยเครื่องมือวัดธรรมดา

**Thesis Title**                    **Surface Scanning Machine**  
**Student**                         **MR.WICHAI CUMPEERAWUT**  
   **MR.VARUTH SIRAPRACHA**  
   **MR.RUNGRODE SITIVICHIANWONG**  
**Degree**                         **Bachelor of Engineering in Mechanical Engineering**  
**Year**                             **1998**



**ABSTRACT**

Nowadays, Computer has played an essential role in the different kind of important aspects. Such as, Designing, Measurement, Computation, Medical Science and so on. Especially, on the designing. We have been developing and improving continuously. The purpose of this enhancement is the highest efficiency we can afford from computer applications. The advantage could be more convenience in computation and evaluation while designing. Therefore, the highest aim of this study is focusing on the combination of computer software and mechanism knowledge. The result will be displayed in the computer screen. From that point, we will be able to know the pattern of the work piece surface, which can not measure by the ordinary measurement method.

## กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้ลุล่วงได้อย่างดี ด้วยการสนับสนุนและให้คำแนะนำในทุกๆด้านจากพี่เต๋า (ดร.สรรพสิทธิ์ ถิ่นนรินทร์) รวมทั้ง อ.เอกพจน์ ดันตราภิวณิช ที่ให้คำปรึกษาทางด้าน Hardware และ อ. อุคม จันทร์จรัสสุข ที่ให้คำปรึกษาทางด้าน Software

ขอขอบคุณ คุณ เบิ้ม และ คุณ หมี่ ซึ่งเป็นเพื่อนของเราอยู่ ภาควิชาวิศวกรรมโทรคมนาคม ที่เป็นกำลังสำคัญในการออกแบบวงจรต่างๆ

ขอขอบคุณ พี่มณฑรา (นาย มณฑรา เทียนเมือง) กับ พี่โก้ (นาย นิรุทธ์ ร่วมกระโทก) ที่ช่วยเหลือทางด้าน Hardware เป็นอย่างมาก

ขอขอบคุณเพื่อนๆนักศึกษาทุกคนที่ให้คำแนะนำและกำลังใจมาตลอดและที่อยากขอบคุณเป็นพิเศษคือ โชคชะตา ที่ทำให้อุปสรรคทุกอย่างผ่านไปได้อย่างดี

สุดท้ายนี้ขอขอบคุณ ภาควิชาวิศวกรรมเครื่องกล ที่ได้ให้ทุนสนับสนุนการทำวิทยานิพนธ์ในครั้งนี้

นาย วิชัย คัมภีร์วุฒิ

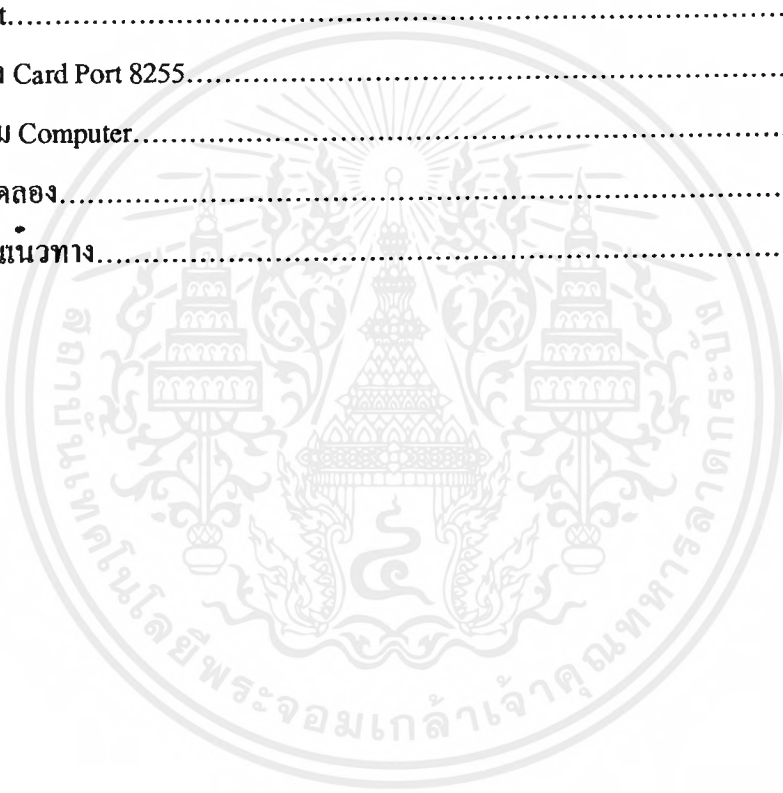
นาย วรุตม์ ศิริประชา

นาย รุ่งโรจน์ สิทธิวิเชียรวงศ์

# สารบัญ

หน้า

บทที่ 1 บทนำ.....	1
บทที่ 2 อุปกรณ์ต่างๆ.....	2
บทที่ 3 Stepping Motors.....	5
บทที่ 4 Encoder.....	13
บทที่ 5 Card Port.....	18
บทที่ 6 การติดตั้ง Card Port 8255.....	45
บทที่ 7 โปรแกรม Computer.....	49
บทที่ 8 ผลการทดลอง.....	59
บทที่ 9 สรุปและแนวทาง.....	62
ภาคผนวก	



# บทที่ 1

## บทนำ

เครื่องลอกแบบพื้นผิวหรือเครื่อง Surface Scanning เป็นเครื่องมือวัดอย่างหนึ่งซึ่งลักษณะโดยทั่วไปของพื้นผิวชิ้นงานที่จะนำมาสแกนนั้นจะมีความแตกต่างกัน โดยอาจมีความแตกต่างกันในเรื่องของความราบเรียบของพื้นผิว ซึ่งอาจจะมีทั้งขรุขระหรือมีลักษณะเป็นหลุม หรือแม้แต่ว่าลาดเอียง ซึ่งจะมีความแตกต่างกันมากหรือน้อยก็ขึ้นอยู่กับลักษณะต่างๆดังที่ได้กล่าวมาแล้วข้างต้น ดังนั้นโดยหลักการทั่วไปแล้ว ในเรื่องของความละเอียด ความแม่นยำ และความถูกต้อง จึงเป็นปัจจัยที่มีความสำคัญอย่างยิ่งกับเครื่องมือชนิดนี้ แต่เนื่องจากปัญหาทางด้านคุณภาพและขอบเขตจำกัดของวัสดุ อุปกรณ์ และเครื่องมือต่างๆ จึงทำให้โครงการนี้มีจุดบกพร่องหรือความคลาดเคลื่อนอยู่บ้างเล็กน้อย อย่างไรก็ตามประสิทธิภาพในการใช้งานโดยทั่วไปนั้นจะใช้ได้ดีเมื่อลักษณะของพื้นผิวชิ้นงานนั้นมีลักษณะต่อเนื่องและไม่มีความแตกต่างกันมากนักในเรื่องของความลาดชัน เมื่อพิจารณาถึงหลักการในการทำงานของเครื่องมือชนิดนี้ก็มีหลักการโดยสังเขปดังต่อไปนี้ ชิ้นงานที่นำมาทดสอบนั้นจะถูกสแกนโดยเครื่องมือชนิดหนึ่งที่เรียกว่าหัวโพรบ ข้อมูลที่เก็บได้บนจุดต่างๆของพื้นผิวชิ้นงานนั้นจะถูกรวบรวม จากนั้นจะถูกส่งไปยังคอมพิวเตอร์ซึ่งจะทำการประมวลผลอีกทีหนึ่ง ผลที่ได้จากการประมวลจะถูกส่งต่อไปยังโปรแกรม AutoCAD และผลสุดท้ายที่ได้จะแสดงออกมาเป็นภาพ 3 มิติบนหน้าจอคอมพิวเตอร์

## บทที่ 2

### อุปกรณ์ต่างๆ

#### 2.1 ชิ้นส่วนและอุปกรณ์ต่างๆที่สำคัญ

1. Table เคลื่อนที่ได้ 2 แกน พร้อม Lead Screw
2. ฐานรองแผ่นเหล็กแผ่นเรียบพร้อมขาตั้งปรับระดับได้
3. Lead Screw ตัวใหญ่ชนิด Ball Bearings และตัวเล็กที่ติดตั้งมาพร้อมกับ Table
4. Bearing 2 ตัวสำหรับรองรับ Lead Screw
5. เสาเหล็กพร้อมที่ยึดติดสำหรับติดตั้งหัว โพรบและเอน โคดเดอร์
6. Stepping Motors 2 ตัว
7. ชุดหัว Probe
8. ชุด Rotary Encoder
9. วงจรควบคุมการนับ
10. วงจรสำหรับขับ Stepping Motors
11. Card Port ชนิด 8255
12. สายไฟที่ใช้ในการเชื่อมต่อวงจรและ Stepping Motors
13. เครื่องคอมพิวเตอร์
14. Power Supply

#### 2.2 รายละเอียดต่างๆของชิ้นส่วนและอุปกรณ์ต่างๆโดยสังเขป

1. Table ที่ใช้นั้นมีขนาด กว้าง 32 เซนติเมตร ยาว 40 เซนติเมตร โดยประมาณ โดยการเคลื่อนที่ของ Table นั้นจะต้องมี motor ส่งกำลังไปขับ lead screw ในการหมุนในแต่ละรอบของ lead screw นั้นจะทำหน้าที่ส่งกำลังไปทำหน้าที่ในการขับ Table อีกทีหนึ่ง โดยในการเคลื่อนที่ของ Table นั้น จะสามารถเคลื่อนได้ 2 แกน โดยในการเคลื่อนที่ไปทางแกน x นั้นจะเคลื่อนที่ได้เป็นระยะทางรวมทั้งสิ้น 20 เซนติเมตรและในการเคลื่อนที่ไปทางแกน y นั้นจะเคลื่อนที่ได้เป็นระยะทางรวมทั้งสิ้น 20 เซนติเมตร
2. ฐานรองรับแผ่นเหล็กพร้อมขาตั้งปรับระดับได้นั้น ใช้เพื่อช่วยให้การเคลื่อนที่ของ Table มีความถูกต้องมากยิ่งขึ้น โดยในการปรับระดับของขาตั้งจะช่วยให้ Table สามารถเคลื่อนที่ได้เป็นระนาบที่ไม่มีการเอียงไปในทิศทางใดทิศทางหนึ่ง ซึ่งจะส่งผลทำให้การวัดพื้นผิวของชิ้นงานนั้นถูกต้องมากยิ่งขึ้น ขนาดโดยประมาณ กว้าง 50 เซนติเมตร ยาว 80 เซนติเมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. Lead Screw ทั้ง 2 ตัวที่ใช้นั้น เราจะใช้ชนิดแบบ ball bearings ซึ่งจะช่วยให้การเคลื่อนที่นั้นมีความราบเรียบ ต่อเนื่อง และแม่นยำมากขึ้น และที่สำคัญความยาวของ lead screw นั้นจะต้องมีความยาวให้เหมาะสมกับระยะทางที่ต้องการให้ table เคลื่อนที่ โดย lead screw ตัวใหญ่จะมีความยาวโดยประมาณ 30 เซนติเมตร และ lead screw ตัวเล็กจะมีความยาวโดยประมาณ 25 เซนติเมตร
4. เสาเหล็กที่ใช้สำหรับติดตั้ง Encoder และ หัว Probe นั้นจะต้องให้มีความสูงมากกว่า Table ประมาณ 20 เซนติเมตร เพื่อช่วยทำให้เราสามารถปรับระดับความสูงค่าให้เหมาะสมกับลักษณะพื้นผิวชิ้นงานต่างๆ ถ้าพื้นผิวของชิ้นงานนั้นมีลักษณะที่เป็น slope สูงชัน ตำแหน่งของหัว probe ก็ควรจะเลื่อนให้สูงตามด้วย หรือในทำนองเดียวกันถ้าชิ้นงานนั้นมีลักษณะเป็นแผ่นบางหรือไม่มีความลาดชันมากนัก ตำแหน่งของหัว probe ก็ควรจะเลื่อนลงมาให้อยู่ในระดับที่สามารถจะวัดได้
5. Bearings ที่ใช้นั้นจะใช้จำนวนทั้งสิ้น 2 ตัว ขนาดเส้นผ่าศูนย์กลางประมาณ 17 mm. และอีกตัวจะมีเส้นผ่าศูนย์กลางประมาณ 19 mm. ซึ่งเส้นผ่าศูนย์กลางที่ใช้นี้ ควรจะให้มีความใกล้เคียงกับขนาดของเพลลาของ lead screw ให้มากที่สุดเท่าที่จะทำได้ เพราะมันจะช่วยทำให้การหมุนของ lead screw นั้นเป็นไปด้วยความราบเรียบมากยิ่งขึ้น ในกรณีที่ขนาดเส้นผ่าศูนย์กลางของ bearings นั้นมีขนาดไม่พอดีกับขนาดเพลลา ผลเสียที่อาจจะเกิดขึ้นนั้นก็คือ จะทำให้เกิด load มากยิ่งขึ้นในขณะที่ lead screw หมุน ซึ่งจะมีผลทำให้ motor เสียกำลัง ใน การหมุนไปอย่างเปล่าประโยชน์
6. Stepping Motors ที่เลือกใช้ในโครงการนี้ เราจะเลือกใช้ชนิดแบบ 6 สาย 4 เฟส หรือเรียกว่า ชนิด ยูนิโพลาร์ โดยส่วนประกอบที่สำคัญของมอเตอร์ชนิดนี้สามารถสังเกตได้โดยง่าย เนื่องจากจะมีสายไฟทั้งหมด 6 เส้นด้วยกัน และในการพันขดลวดนั้นจะมีการพันทั้งหมด 2 ขด บน ขั้วแม่เหล็กของสเตเตอร์ ซึ่งในแต่ละขดจะทำให้เกิดขั้วแม่เหล็กในทิศทางตรงข้ามกัน
7. ชิ้นส่วนประกอบที่สำคัญของหัว probe ที่ใช้ในการวัดระยะนั้น ประกอบด้วยอุปกรณ์สำคัญ 2 อย่างคือ รางเฟืองตรงที่มีความยาวประมาณ 13 เซนติเมตร โดยรางเฟืองนี้จะทำหน้าที่ให้เฟืองกลมที่ติดอยู่บนแกนของ encoder นั้นขบกันและช่วยลดการสั่นไหวในขณะที่หัว probe เคลื่อนที่ขึ้นและลงในขณะทำการวัด สำหรับอุปกรณ์ที่เป็นส่วนประกอบที่สำคัญอีกอย่างหนึ่งของหัว probe ก็คือ ส่วนที่ใช้ในการลากเคลื่อนที่บนผิวชิ้นงาน ทำจากหัวลูกกลิ้งของปากกาลูกกลิ้งธรรมดา ซึ่งมีข้อดีคือ มีขนาดเล็กกระทัดรัด สามารถวัดได้อย่างละเอียด และมีความคล่องตัวสูงขณะที่ทำการวัด เนื่องจากมีความลื่นในตัวมันเอง
8. Encoder ที่ใช้นั้นจะเป็นแบบ rotary ซึ่งในการหมุนแต่ละรอบของ encoder นั้นจะส่งสัญญาณ pulse ออกมา จากนั้นสัญญาณนี้ก็จะผ่านไปยังชุดของวงจรรับในขบวนการต่อไป สำหรับใน

- โครงการนี้ ความละเอียดของ encoder ที่ใช้จะอยู่ที่ 500 pulses ต่อการหมุนหนึ่งรอบของ encoder
9. วงจรควบคุมการนับนั้นจะทำหน้าที่รับสัญญาณ pulse ที่ส่งมาจาก encoder แล้วทำการแปลงจากสัญญาณ pulse ให้เป็นสัญญาณที่ส่งไปให้คอมพิวเตอร์ประมวลผลอีกที โดยวงจรนี้จะมี ส่วนประกอบ 2 ส่วนใหญ่ๆด้วยกันดังนี้ คือ วงจรแยกความต่างเฟสและวงจรมับ
  10. วงจรสำหรับควบคุม Stepping Motors ประกอบด้วยส่วนประกอบหลักๆดังต่อไปนี้ IC# 74LS540 จำนวน 1 ตัว Opto 4N33 จำนวน 8 ตัว และ Transistor BC 547 และ TIP 3055 อย่างละ 8 ตัว
  11. Card Port ที่ใช้ในโครงงานนั้นจะเป็นชนิดที่ใช้ IC#8255 ซึ่งเป็นพอร์คขนาน และทำมาในรูปแบบของ card ประกอบด้วยส่วนต่างๆรวมอยู่ใน card แผ่นเดียว และใน card นี้จะมี 2 ส่วนใหญ่ๆด้วยกัน คือ ส่วนที่เป็นวงจร Decode และส่วนของ Port 8255 สำหรับรายละเอียดกล่าวถึงต่อไปในภายหลัง
  12. สายไฟต่างๆที่ใช้นั้น มีทั้งชนิดสายไฟที่ใช้ต่อระหว่างวงจรควบคุม stepping motors และวงจรควบคุมการนับ และสายไฟที่ใช้สำหรับต่อเข้ากับ ports ต่างๆของ computer
  13. เครื่อง Computer ที่ใช้นั้นควรจะเป็นเครื่องที่มี Spec เป็น Pentium ขึ้นไปเพื่อการประมวลที่รวดเร็ว และเพื่อการแสดงผลที่มีประสิทธิภาพ โดยมีโปรแกรมที่จำเป็นสำหรับการควบคุมการทำงานและการวัดต่างๆ ของเครื่อง คือ Delphi, Pascal, Excel, AutoCAD.
  14. Power Supply ที่ใช้นั้นควรจะเป็นชนิดที่สามารถปรับปริมาณกระแสและความต่างศักย์ได้ เพื่อการใช้งานที่เหมาะสมของขนาดของ stepping motors และขนาดหรือน้ำหนักของ Table

## บทที่ 3

# STEPPING MOTORS

### 3.1 การทำงานของสเต็ปเปอร์มอเตอร์

สเต็ปเปอร์มอเตอร์มีความแตกต่างจากมอเตอร์ทั่วไป โดยเมื่อป้อนกำลังไฟฟ้าให้กับมัน มันจะหมุนเพียงเล็กน้อยตามเส้นรอบวงและหยุด ซึ่งแตกต่างจากมอเตอร์ทั่วไปๆ ซึ่งจะหมุนทันทีและตลอดเวลา เมื่อป้อนแรงดันไฟฟ้า สเต็ปเปอร์มอเตอร์สามารถกำหนดตำแหน่งของการหมุนด้วยตัวเลขได้อย่างละเอียด โดยการใช้คอมพิวเตอร์เป็นตัวกำหนดและจัดเก็บตัวเลขเหล่านั้นไว้

สเต็ปเปอร์มอเตอร์สามารถใช้งานในระบบเปิด (open loop system) นั่นก็คือมันทำงานได้โดยไม่ต้องมีการป้อนกลับ (feedback) แต่ทุกวิธีที่ต้องการกำหนดตำแหน่งได้อย่างถูกต้องจำเป็นต้องจำเป็นต้องมีการป้อนกลับ ไปยังระบบให้รับรู้ และตัวบอกตำแหน่งว่าถูกต้องหรือเกิดการผิดพลาด

วิธีหนึ่งที่ใช้กันโดยทั่วไปกับสเต็ปเปอร์มอเตอร์ คือ การใช้สวิทช์ติดตั้งไว้ที่ตำแหน่งที่ต้องการตรวจจับ (limit switch) เมื่อสเต็ปเปอร์มอเตอร์เริ่มหมุนและหมุนจนกระทั่งถึงตำแหน่งของสวิทช์ตรวจจับสัญญาณ ก็จะถูกป้อนกลับสู่ระบบและทราบการทำงานของสเต็ปเปอร์มอเตอร์ได้ตลอดเวลาซึ่งโดยปกติในวงจรคอนโทรลเลอร์จะมีการกำหนดจุดอ้างอิง (reference point) ไว้ด้วยเพื่อให้เริ่มต้นทำงานและอ้างอิงตำแหน่งได้อย่างถูกต้อง ตัวอย่างเช่นถ้าเริ่มจ่ายกำลังไฟฟ้าให้กับฟลูออปีดิสก์ไดรฟ์ จะได้อินมันกำลังเคลื่อนที่เพื่อหาจุดอ้างอิงที่กำหนด หลังจากนั้นวงจรไดรฟ์คอนโทรลเลอร์จะเริ่มทำงานได้ โดยมันจะทราบถึงทุกๆ สเต็ปที่กำลังขับเคลื่อนหัวอ่าน เขียนไปยังแต่ละแท่งบนดิสก์ เช่นเดียวกับมอเตอร์ทั่วไปการที่จะทำให้เกิดการหมุนของโรเตอร์ (rotor) ได้ต้องมีการกระทำของสนามแม่เหล็กที่เกิดขึ้นระหว่างโรเตอร์และสเตเตอร์ (stator) ซึ่งขึ้นอยู่กับการจัดวางขั้วแม่เหล็ก (pole) การหมุนทำได้ทั้งแบบต่อเนื่องและกลับทิศทางไปมา โดยกระบวนการทางไฟสลับหรือการจัดวางแปร่งถ่าน และการจัดแยกคอมมิวเตเตอร์ และทำการสวิทซ์ซึ่งกำลังไฟฟ้าให้เกิดแรงดึงของแม่เหล็ก (magnetic attraction) ที่ขั้วแม่เหล็กสร้างและหยุดสลับกัน ผลก็คือเกิดสนามแม่เหล็กหมุนขึ้นบนสเตเตอร์โดยการจ่ายกำลังไฟฟ้าทีละคู่ของขั้วแม่เหล็กในทิศทางตรงกันข้ามไปตลอดเวลาและเมื่อต้องการหยุดหมุนทำได้โดยหยุดการเกิดขั้วแม่เหล็กที่จุดหนึ่ง โดยหยุดการสวิทซ์ซึ่งในลำดับต่อไปเสีย การหมุนกลับทิศทางก็ทำได้เช่นเดียวกับที่กล่าวมาแล้วเพียง แต่ทำการสวิทซ์ซึ่งกำลังไฟฟ้าให้เกิดสนามแม่เหล็กหมุนในทิศกลับกันหรือกลับลำดับการสวิทซ์ของมัน

โครงสร้างของขั้วแม่เหล็กบนสเตเตอร์ประกอบขึ้นจากแผ่นเหล็กวงแหวนที่มีซี่ยื่นออกมาแต่ละซี่เหล่านั้นจะมีคอยล์พันสวมอยู่บนนั้นเมื่อป้อนกระแสผ่านคอยล์ทำให้เกิดสนามแม่เหล็กไฟฟ้า (Electromagnetic) ซี่ที่ด้านตรงข้ามของแต่ละขั้วแม่เหล็กจะได้รับกระแสไฟฟ้าในขณะเดียวกัน แต่ว่าจะไหลวนในทิศทางตรงกันข้ามทำให้เกิดสนามแม่เหล็กไฟฟ้าในทิศตรงข้ามขึ้นด้วยแสดงในรูปที่ 3.1 (ก) ดังนั้นถ้าเพิ่มจำนวนของขั้วแม่เหล็กให้มากขึ้นจะเพิ่มจำนวนของสเต็ปต่อวงรอบมากขึ้นตามไปด้วย อย่างไรก็ตามผู้ใช้สามารถเพิ่มจำนวนของสเต็ปได้อีกวิธีหนึ่งโดยไม่ต้องปรับเปลี่ยน โครงสร้างภายใน โดยทำการจ่ายกำลังไฟฟ้าไปยังขั้วแม่เหล็ก 2 ขั้วที่อยู่ใกล้กันในเวลาเดียวกันซึ่งจะทำให้โรเตอร์หยุดหมุนอยู่ระหว่างกลางของ 2 ขั้วแม่เหล็กนั้นหรือเคลื่อนที่ไปครึ่งสเต็ปเท่านั้นและวิธีการนี้ยังช่วยให้เกิดแรง (torque) มากขึ้นด้วยดังแสดงในรูปที่ 3.1 (ข) ทำให้โรเตอร์เคลื่อนที่มาหยุดอยู่ระหว่างขั้วแม่เหล็กทั้งสอง



รูปที่ 3.1 (ก) แสดงสเต็ปเปอร์มอเตอร์ที่มีการต่อวงจรขดลวดภายในเพื่อกระตุ้นให้เกิดขั้วแม่เหล็กขึ้น 1 ขั้วในทิศทางตรงกันข้ามส่วนขดลวดอื่นๆจะไม่ถูกกระตุ้นเลย  
(ข) แสดงการต่อวงจรขดลวดแบบกระตุ้นให้เกิดขั้วแม่เหล็กพร้อมกัน 2 ขั้วอยู่ใกล้กัน

สเต็ปเปอร์มอเตอร์โดยทั่วไปมีจำนวนของขั้วแม่เหล็กหรือจำนวนสเต็ปต่อรอบเป็นจำนวนมากปกติอยู่ที่ประมาณ 100-400 สเต็ปต่อรอบการมีจำนวนสเต็ปมากนี้ไม่ได้เพิ่มจำนวนขั้วแม่เหล็กไฟฟ้าที่สเตเตอร์ แต่ทำได้โดยเพิ่มจำนวนซี่ขั้วแม่เหล็กที่โรเตอร์ จำนวนสเต็ปต่อรอบทั้งหมดจะได้ออกจากการคูณจำนวนขั้วแม่เหล็กบนสเตเตอร์และจำนวนซี่ที่โรเตอร์ ดังเช่นถ้ามีขั้วแม่เหล็ก 3 ขั้วบนสเตเตอร์ และ 8 ซี่ขั้วแม่เหล็กบนโรเตอร์ สเต็ปเปอร์มอเตอร์นี้จะทำงานที่ 24 สเต็ปต่อรอบหรือหมุนไปเป็นมุม 15 องศาต่อสเต็ป

การใช้วงจรดิจิทัลคอนโทรลเลอร์กำหนดการจ่ายกำลังไฟฟ้าเข้าสู่ขดลวดบนสเตเตอร์แบบซีควเอนเชียลทำให้สามารถควบคุมการเคลื่อนที่ทุกสเต็ปได้เช่นเดียวกับการควบคุมในวงจรดีซีเซอร์โว (DC servo) แต่การควบคุมด้วยดิจิทัลไม่จำเป็นต้องมีการป้อนกลับการเคลื่อนที่ทุกสเต็ปได้จากการคำนวณจำนวนรอบหรือมุมในการหมุนที่ต้องการ แล้วจึงส่งข้อมูลที่ได้ออกไปควบคุมการหมุนของมอเตอร์ มุมในการเคลื่อนที่ (ที่ติดกับเพลลา) ถูกกำหนดจากข้อมูลที่ส่งมาควบคุม

### 3.2 ชนิดของสเต็ปเปอร์มอเตอร์

สเต็ปเปอร์มอเตอร์แบ่งตามพื้นฐานได้เป็น 3 ชนิดคือ วาริเอเบิลรีลักแตนซ์ (variable reluctance VR) เพอร์มาเนนต์แมกเน็ต ( permanent magnet PM ) และแบบไฮบริดจ์ (hybrid)

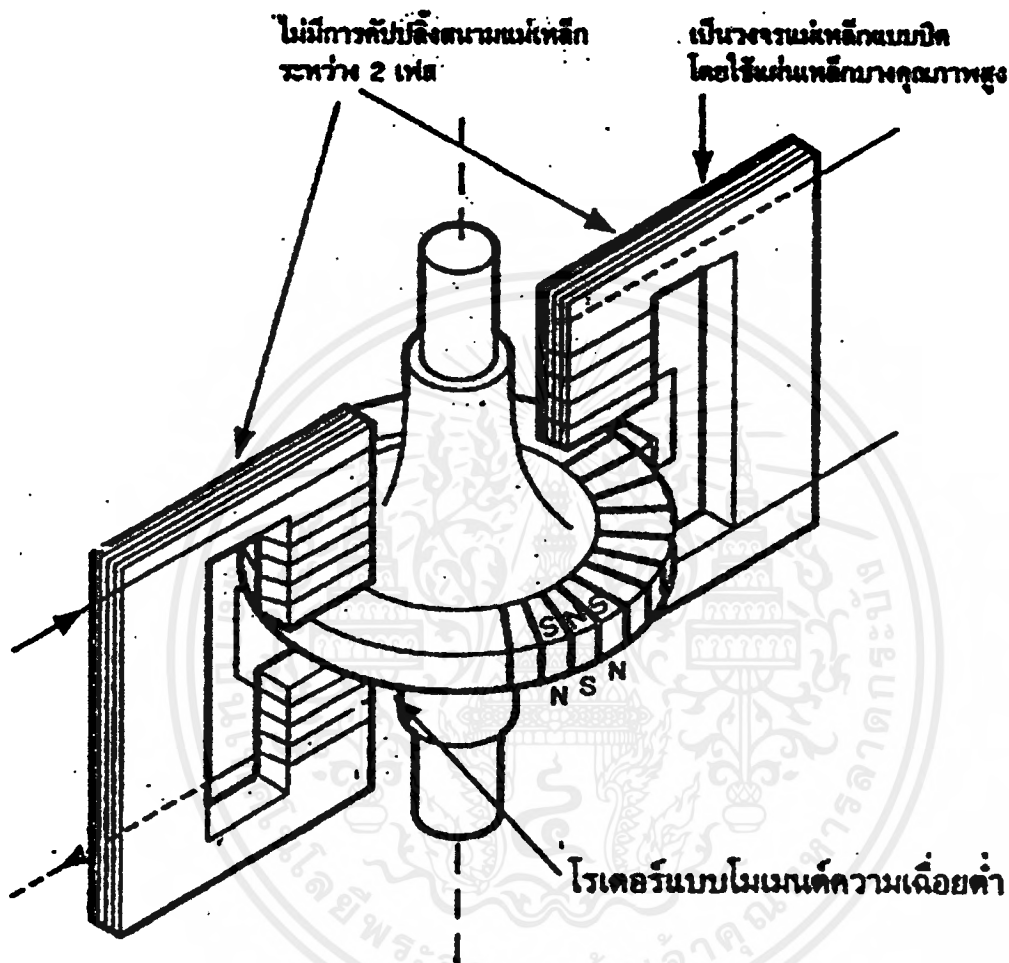
3.2.1 วาริเอเบิลรีลักแตนซ์ มีโครงสร้างของโรเตอร์แบบมัลติทูธ ( multi-tooth ) ทำจากเหล็กอ่อนเราจะทราบได้ว่าเป็นมอเตอร์ชนิดนี้โดยการทดสอบได้ง่ายมากคือใช้นิ้วหมุนเพลลาของมอเตอร์และสังเกตมอเตอร์ชนิดนี้ที่โรเตอร์จะไม่เกิดปรากฏการณ์ทางแม่เหล็ก (magnetism) มันจึงหมุนได้ตลอดโดยไม่ติดขัดแตกต่างจากชนิด PM และชนิดไฮบริดซึ่งมีสนามแม่เหล็กที่โรเตอร์เมื่อหมุนจะรู้สึกขั้วๆเหมือนเป็นฟันเฟืองสเต็ปเปอร์มอเตอร์ชนิดนี้มีจุดค้อยในเรื่องของความถูกต้องของตำแหน่งและทำงานได้ไม่คืดักเมื่อมีสเต็ปในการหมุนสูง

3.2.2 เพอร์มาเนนต์แมกเน็ต มีโครงสร้างของโรเตอร์แบบเรียบไม่มีซี่ขั้วแม่เหล็กและบนโรเตอร์จะเป็นแบบแม่เหล็กถาวร การควบคุมทำโดยป้อนกระแสกระตุ้นที่ขดลวดบนสเตเตอร์เช่นถ้าเป็นสเตเตอร์แบบ 4 เฟสจะมีขั้วแม่เหล็กอยู่ 4 ขั้วซึ่งมีขั้วลัษณะอยู่แยกจากกัน ขั้วแม่เหล็กถาวรบนโรเตอร์จะถูกดึงดูดจากขั้วแม่เหล็กบนสเตเตอร์เมื่อป้อนกระแสไฟฟ้าเข้าสู่ขดลวด และโรเตอร์จะอยู่คงที่ที่ขั้วแม่เหล็กบนสเตเตอร์นั้นถึงแม้ว่าจะไม่ป้อนกระแสไฟฟ้าอีกต่อไป ทำให้เกิดเป็นแรงยึดเหนี่ยวขึ้น สเต็ปเปอร์มอเตอร์ชนิดนี้มีข้อดีในเรื่องของความถูกต้องของตำแหน่ง และความเร็วมากขึ้นเมื่อเปรียบเทียบกับชนิดอื่น

3.2.3 ไฮบริด เป็นชนิดที่นิยมใช้งานกันมากที่สุด โดยเฉพาะนำมาใช้งานอย่างมากในอุปกรณ์ที่ใช้เครื่องคอมพิวเตอร์ ชนิดไฮบริดมีโครงสร้างภายในซึ่งได้จากการรวมเอาโครงสร้างของสเตเตอร์ชนิดวาริเอเบิลรีลักแตนซ์ และโครงสร้างของโรเตอร์จากชนิดเพอร์มาเนนต์แมกเน็ตมาประกอบเข้าด้วยกันจึงทำให้เป็นมอเตอร์ชนิดที่มีแรงยึดเหนี่ยวสูงมีแรงบิดดีและผลักได้ดีซึ่งมีความคงที่และทำงานได้ดีถึงแม้ว่าจะมีสเต็ปต่อรอบในการหมุนสูง

สเต็ปเปอร์มอเตอร์แบบใหม่อีกชนิดหนึ่งเป็นชนิดที่ปรับปรุงมาจากชนิดเพอร์มาเนนต์แมกเน็ต นั่นคือ ชนิดเรเอิร์ธเพอร์มาเนนต์แมกเน็ต (rare earth permanent magnet) ดังแสดงโครงสร้างภายในรูปที่ 2 หรือที่เรียกกันว่าชนิดดิสก์แมกเน็ตสเต็ปเปอร์มอเตอร์ ( disk magnet steppers) โครงสร้างของโรเตอร์ของมอเตอร์ชนิดนี้มีลักษณะเป็นแผ่นซึ่งยึดกับเพลลาของมอเตอร์ การทำงานของเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มอเตอร์ยังคงเป็นเช่นเดิม แต่ด้วยโครงสร้างแบบใหม่นี้ช่วยทำให้เกิด โมเมนต์ของความเฉื่อยต่ำมาก, มีอัตราเร่งสูงมอเตอร์ชนิดนี้จึงจัดเป็นอีกชนิดหนึ่งที่มันก็มีประสิทธิภาพสูงอีกหลายด้านเช่น แรงบิดสูง, กำลังทางกลที่ได้ของมอเตอร์, ความถูกต้องของตำแหน่งสูงมากและความเร็วในการเริ่มหมุนสูง อีกทั้งยังมีความสูญเสียของกำลังงานต่ำ ดังที่แสดงในรูปที่ 3.2



รูปที่ 3.2 แสดง โครงสร้างภายในพื้นฐานของชนิดเรอัวร์พอร์มาเนนต์แมกเน็ตสเต็ปเปอร์มอเตอร์

### 3.3 การจำแนกชนิดของสเต็ปเปอร์มอเตอร์ด้วยการพันคอยล์

การพันขดลวดหรือคอยล์บนสเต็ปเปอร์มอเตอร์มีอยู่ 2 วิธีคือ แบบไบโพลาร์ (bipolar) และแบบยูนิโพลาร์ (unipolar) ด้วย ขั้วเป็นแบบไบโพลาร์และที่เหลือเป็นแบบยูนิโพลาร์ซึ่งมีทั้ง

สเต็ปเปอร์มอเตอร์แบบไบโพลาร์มีการพันขดลวด 1 ขดบนแต่ละขั้วแม่เหล็กของสเตเตอร์ขั้วแม่เหล็กที่เกิดขึ้นบนสเตเตอร์ถูกกำหนดโดยทิศทางของกระแสไฟฟ้าและสามารถทำให้เกิดขั้วแม่เหล็กในทิศทางตรงข้าม ได้โดยการกลับทิศทางกระแสไหลของไหลของกระแสไฟฟ้าซึ่งการกำหนดทิศทางไหลและการกลับทิศทางของกระแสไฟฟ้าทำได้โดยการใช้วงจรสวิตชิงกลับขั้วไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับยูนิโพลาร์จะมีการพันขดลวด 2 ขด บนแต่ละขั้วแม่เหล็กของสเตเตอร์ซึ่งแต่ละขดจะทำให้เกิดขั้วแม่เหล็กในทิศทางตรงข้ามกัน การกลับขั้วแม่เหล็กเปลี่ยนไปมาทำได้โดยการสวิตชิงกระแสไฟฟ้าจากขดลวดขดหนึ่งไปยังอีกขดหนึ่งแทนเท่านั้น โดยปกติขดลวดทั้งสองจะมีการเชื่อมต่อกันหรือมีจุดร่วมเพื่อลดจำนวนของสายไฟที่ต่อจากมอเตอร์ วงจรจ่ายกำลังไฟฟ้าของมอเตอร์แบบยูนิโพลาร์ทำได้ง่ายกว่าชนิดไบโพลาร์ เพราะมันต้องการเพียงสวิตช์ธรรมดาในการเปิดและปิดกำลังไฟฟ้าให้กับขดลวดบนสเตเตอร์ในทิศทางที่ต้องการให้หมุนได้ทันที

อย่างไรก็ตามการพันขดลวดแบบยูนิโพลาร์ก็มีจุดด้อยตรงที่การพันแบบนี้จะทำให้เกิดแรงบิดน้อยกว่าแบบไบโพลาร์เพราะจะมีเพียงครึ่งหนึ่งของขดลวดที่ถูกกระตุ้นให้ทำงานเท่านั้นในระยะเวลาหนึ่ง

การพิจารณาว่าสเต็ปเปอร์มอเตอร์ตัวใดมีการพันขดลวดแบบใดสังเกตได้ง่ายโดยถ้าเป็นแบบไบโพลาร์จะมีสายไฟต่อออกจากมอเตอร์เพียง 4 สาย และถ้าเป็นแบบยูนิโพลาร์จะมี 5 สายหรือ 6 สายหรือทราบได้โดยการอ่านจากป้าย (name plate) ที่ติดอยู่กับมอเตอร์ได้

### 3.4 การกระตุ้นและควบคุมการหมุนของสเต็ปเปอร์มอเตอร์

การกระตุ้นและควบคุมการหมุนของมอเตอร์ให้เคลื่อนที่ไปแต่ละสเต็ปทำได้โดยจ่ายกำลังไฟฟ้าไปยังขดลวดแต่ละขดบนสเตเตอร์ ซึ่งต้องป้อนเป็นแบบซีควเอนเชียลในรูปแบบที่ถูกต้องแบ่งได้เป็น 3 รูปแบบ คือแบบเวฟ (wave) , แบบ 2 เฟส (two phase) และแบบครึ่งสเต็ป (half step) ทั้ง 3 แบบต่างก็มีข้อดีและข้อเสียแตกต่างกันออกไป

แบบเวฟเป็นการกระตุ้นรูปแบบที่ง่ายที่สุด โดยทำการกระตุ้นขดลวดทีละขดในเวลาหนึ่งและเรียงถัดกันไป ดังเช่นขดที่ 1,2,3,4,1 หรือ 1,4,3,2,1 ขึ้นอยู่กับทิศทางที่ต้องการให้หมุน ดังนั้น จึงมีขดลวดเพียงขดเดียวในเวลาที่ถูกกระตุ้นเท่านั้น วงจรกระตุ้นแบบเวฟจึงมีราคาถูกและง่ายขั้นตอนการทำงานต่างๆ แสดงดังในตารางที่ 1

ตารางที่ 1 แสดงขั้นตอนการกระตุ้นขดลวดแต่ละเฟสแบบเวฟ

สเต็ปที่	เฟสที่1	เฟสที่2	เฟสที่3	เฟสที่4
1	ทำงาน	-	-	-
2	-	ทำงาน	-	-
3	-	-	ทำงาน	-
4	-	-	-	ทำงาน

แบบ 2 เฟสเป็นการกระตุ้นอีกรูปแบบหนึ่งซึ่งคล้ายกับแบบเวฟแต่การกระตุ้นแบบนี้จะทำการโดยจ่ายกำลังไฟฟ้าไปที่ขดลวด 2 ขดที่อยู่ใกล้กันในเวลาเดียวกันและเรียงติดกันไปเช่นเดียวกับการเวฟคือขดลวดที่ถูกกระตุ้น 12,23,34,41,12 หรือ 14,43,32,21,14 ขึ้นอยู่กับทิศทางการหมุน การเพิ่มจำนวนของขดลวดที่ถูกกระตุ้นนี้ทำให้เพิ่มแรงบิดได้มากกว่าแบบเวฟ โรเตอร์จะเคลื่อนที่ได้ด้วยแรงดึงแบบเต็มแรง จาก 2 ขดลวดที่ถูกกระตุ้นพร้อมกันและต่อไปด้วยแรงดึงจากอีก 2 ขดลวดถัดไป สำหรับข้อเสียคือ การกระตุ้นแบบนี้ต้องใช้แหล่งจ่ายกำลังไฟฟ้ามากขึ้น ขั้นตอนการทำงานต่างๆ แสดงดังในตารางที่ 2

ตารางที่ 2 แสดงขั้นตอนการกระตุ้นขดลวดแต่ละเฟสแบบ 2 เฟส

สเต็ปที่	เฟสที่1	เฟสที่2	เฟสที่3	เฟสที่4
1	ทำงาน	ทำงาน	-	-
2	-	ทำงาน	ทำงาน	-
3	-	-	ทำงาน	ทำงาน
4	ทำงาน	-	-	ทำงาน

แบบครึ่งสเต็ปเป็นรูปแบบที่เกิดจากการผสมผสานระหว่างการกระตุ้นแบบเวฟและแบบ 2 เฟส เพื่อเพิ่มจำนวนจำนวนของสเต็ปต่อรอบอีกเท่าตัวหนึ่ง ในระบบนี้จะทำการกระตุ้นขดลวดเรียงกันไปเป็นลำดับนี้ ขดลวดที่ถูกกระตุ้น 1,12,2,23,3,34,4,41,1 หรือในการหมุนอีกทิศทางหนึ่งจะได้เป็น 1,14,4,43,3,32,2,21,1 แรงบิดที่ได้จากการกระตุ้นแบบนี้จะเพิ่มมากขึ้นอีก เพราะช่วงสเต็ปมีระยะสั้นลงและแต่ละสเต็ปเกิดแรงดึงจากขดลวด 2 ขดที่ถูกกระตุ้นพร้อมกัน ความถูกต้องของตำแหน่งมีเพิ่มมากขึ้นแต่ต้องพึงระวังไว้อีกประการหนึ่งว่าเมื่อถูกกระตุ้นให้ทำงานในรูปแบบนี้จะต้องทำการหมุนถึง 2 สเต็ปจึงจะได้เท่ากับ 1 สเต็ปเต็มเหมือนกับในการควบคุมแบบ 2 แบบแรก สำหรับแหล่งจ่ายกำลังไฟฟ้าต้องใช้เทียบเท่ากับแบบ 2 เฟสจึงจะเพียงพอขั้นตอนการทำงานต่างๆ ดังแสดงในตารางที่ 3

### ตารางที่ 3 แสดงขั้นตอนการกระตุ้นขดลวดแต่ละเฟสแบบครึ่งสเต็ป

สเต็ปที่	เฟสที่1	เฟสที่2	เฟสที่3	เฟสที่4
1	ทำงาน	-	-	-
2	ทำงาน	ทำงาน	-	-
3	-	ทำงาน	-	-
4	-	ทำงาน	ทำงาน	-
5	-	-	ทำงาน	-
6	-	-	ทำงาน	ทำงาน
7	-	-	-	ทำงาน
8	ทำงาน	-	-	ทำงาน

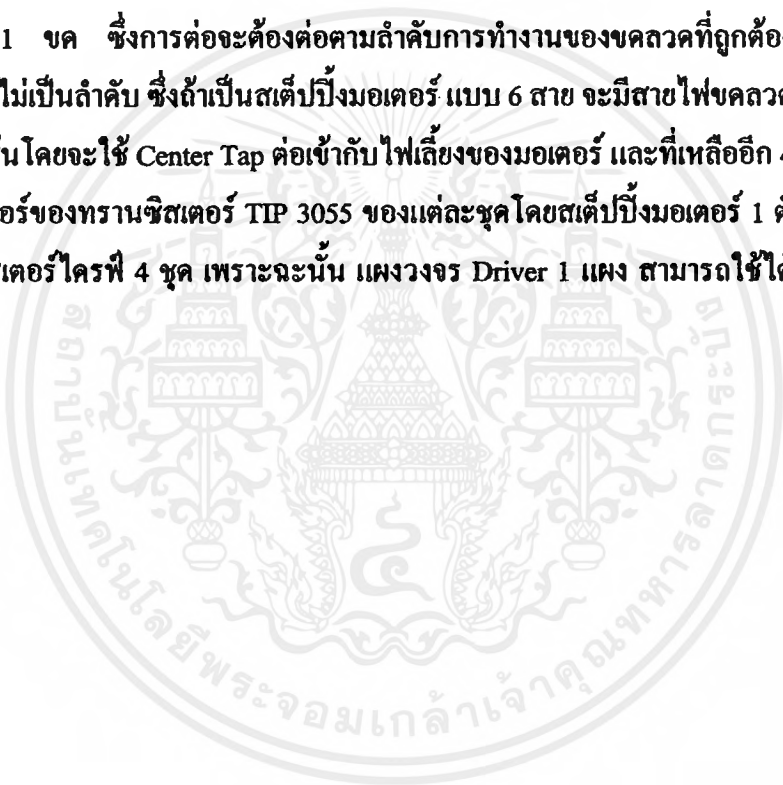
### 3.5 การทำงานของวงจรควบคุม Stepping motor

การทำงานของวงจร Driver เริ่มจากการ์ดพอร์ต 8255 ส่งผ่านตามสายสัญญาณมาที่วงจร Driver เข้ามาที่ออปโตคัปเปอร์ (Optocoupler) ซึ่งจะทำหน้าที่เป็นตัว Detect ตรวจสอบสัญญาณที่เข้ามา ซึ่งมีทั้งลอจิก 0 และลอจิก 1 ถ้ามีระดับแรงไฟเป็นลอจิกเป็น 1 จะทำให้เอาท์พุทของออปโตคัปเปอร์ซึ่งเป็นทรานซิสเตอร์ทำงานเปรียบเสมือน ON เอาท์พุทจึงเปรียบเสมือนต่อลงกราวด์และมีระดับลอจิก เป็น 0 จากนั้นสัญญาณจะถูกส่งให้ ไอซี 74LS540 ซึ่งทำหน้าที่เป็นตัว BUFFER และจะมี Inverter อยู่ในตัว ไอซี 74LS540 ด้วย ที่เอาท์พุทของ ไอซี 74LS540 จะมีตัวต้านทานต่ออนุกรมกับแอลอีดี (LED) เพื่อแสดงสัญญาณที่ออกจากเอาท์พุทของ ไอซี 74LS540 จากนั้นสัญญาณจะถูกส่งไปที่ทรานซิสเตอร์ไครฟ์ ซึ่งภาคไครฟ์จะใช้ทรานซิสเตอร์ เบอร์ BC 547 ต่อแบบคาร์ลิงตันกับทรานซิสเตอร์ เบอร์ TIP 3055 ทรานซิสเตอร์จะต่อกันแบบคาร์ลิงตันและมีไดโอดเบอร์ 1N4001 ต่อคร่อมขาคอลเลคเตอร์กับขาอีมีตเตอร์เพื่อให้ไฟลบบนขดลวดถูกบายพาส ลงกราวด์และเพื่อป้องกันแรงดันย้อนกลับในขดลวดของมอเตอร์ ซึ่งการกระตุ้นการทำงานของสเต็ปปั๊มมอเตอร์จะต้องเป็นไปตามลำดับ ( Sequence) ทั้งในทิศทางตามเข็มนาฬิกาหรือทวนเข็มนาฬิกา คือ เราจะใช้โปรแกรม Delphi เขียนโปรแกรมเพื่อติดต่อกับ 8255 Card Port ซึ่ง Card Port นี้จะต่ออยู่กับภาคขับเคลื่อน โดยใช้สายแพเป็นสายนำสัญญาณข้อมูล Bus ซึ่งต้องใช้สายแพ 34 pin

เนื่องจาก 8255 Card Port ใช้ไอซี 8255 จำนวน 3 ตัว โดยแต่ละตัวทำงานแยกจากกันเป็น 3 ชุด แต่ละชุดจะมีพอร์ตข้อมูล (Data Port) 3 พอร์ต คือ พอร์ต A พอร์ต B และ พอร์ต C

ฮาร์ดแวร์ควบคุมการทำงานของ Stepping Motor ประกอบด้วย 8255 Card Port ซึ่งเสียบอยู่กับ Slot ของ คอมพิวเตอร์ และต่อสายแพ 34 Pin 3 เส้น มายังภาคขับเคลื่อน (Driver) โดยที่ขา 1 ของออปโตคัปเปอร์ เป็น อินพุทที่ต่อมาจาก 8255 Card Port ซึ่งออปโตคัปเปอร์ 1 ตัวจะต่อกับ Data

Bus ได้ 1 เส้น ในแผงวงจรภาค Driver 1 แผง จะมีออปโตคอปเปอร์ 8 ตัว จะต่อกับ Data Bus ได้ 8 เส้น ขา 2 ของออปโตคอปเปอร์จะต่อลงกราวด์ของ 8255 Card Port ซึ่งจะต่อร่วมกับกราวด์ของคอมพิวเตอร์ ที่เอาต์พุตของออปโตคอปเปอร์ขา 5 ต่อกับไฟเลี้ยง +5 โวลต์ และต่อเข้ากับอินพุตของไอซี 74LS540 มี 8 ขา คือ A1-A8 และมีเอาต์พุต 8 ขา คือ Y1-Y8 ขา 20 เป็นไฟเลี้ยงไอซี ขา 10 เป็นขากราวด์ ขา 1 ของไอซี คือ G1 และขา 2 ของไอซี คือ G2 ขา 1 และ ขา 2 จะต่อลงกราวด์ เพื่อให้สัญญาณสามารถปรากฏที่เอาต์พุตของไอซีได้ จากเอาต์พุตของไอซี 74LS540 และต่ออนุกรมกับตัวความต้านทานไบแอสที่ขาเบสของทรานซิสเตอร์ BC 547 และทรานซิสเตอร์ TIP 3055 ที่ต่อกันแบบคาร์ลิงตัน ซึ่งมีทั้งหมด 8 ชุด แต่ละชุดต่อกับเอาต์พุตของ ไอซีแต่ละขาแยกจากกัน ที่ขาคอลเลกเตอร์ของทรานซิสเตอร์ TIP 3055 แต่ละชุดจะต่อกับขดลวดของ สเต็ปป์มอเตอร์ โดย 1 ชุดต่อกับขดลวด 1 ชุด ซึ่งการต่อจะต้องต่อตามลำดับการทำงานของขดลวดที่ถูกต้อง มิเช่นนั้นมอเตอร์จะหมุนไม่เป็นลำดับ ซึ่งถ้าเป็นสเต็ปป์มอเตอร์ แบบ 6 สาย จะมีสายไฟขดลวด 4 เส้น และ Center Tap 2 เส้น โดยจะใช้ Center Tap ต่อเข้ากับไฟเลี้ยงของมอเตอร์ และที่เหลืออีก 4 เส้น ต่อเข้ากับขาคอลเลกเตอร์ของทรานซิสเตอร์ TIP 3055 ของแต่ละชุดโดยสเต็ปป์มอเตอร์ 1 ตัว จะต้องใช้ชุดของทรานซิสเตอร์ไครี 4 ชุด เพราะฉะนั้น แผงวงจร Driver 1 แผง สามารถใช้ได้กับสเต็ปป์มอเตอร์ 2 ตัว



## บทที่ 4

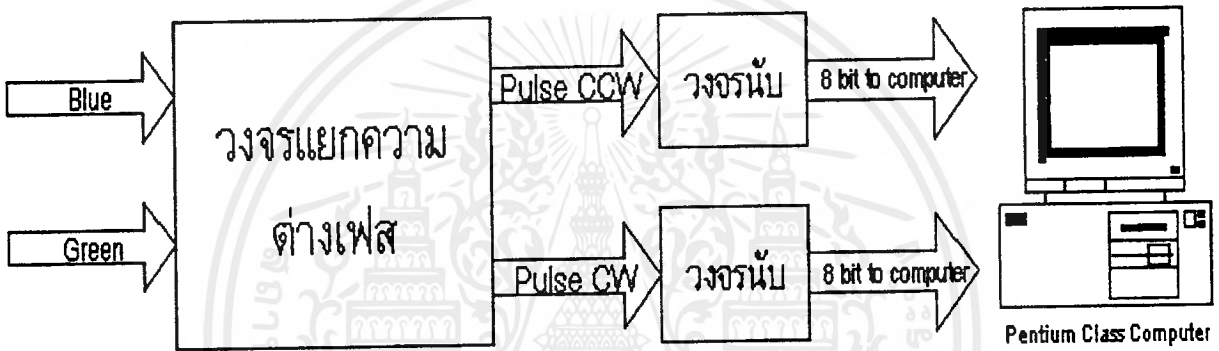
# ENCODER

### 4.1 วงจรนับพัลส์จาก encoder

วงจรรนับพัลส์ จาก encoder มีอยู่ 2 ส่วนประกอบใหญ่ๆ ด้วยกันดังนี้

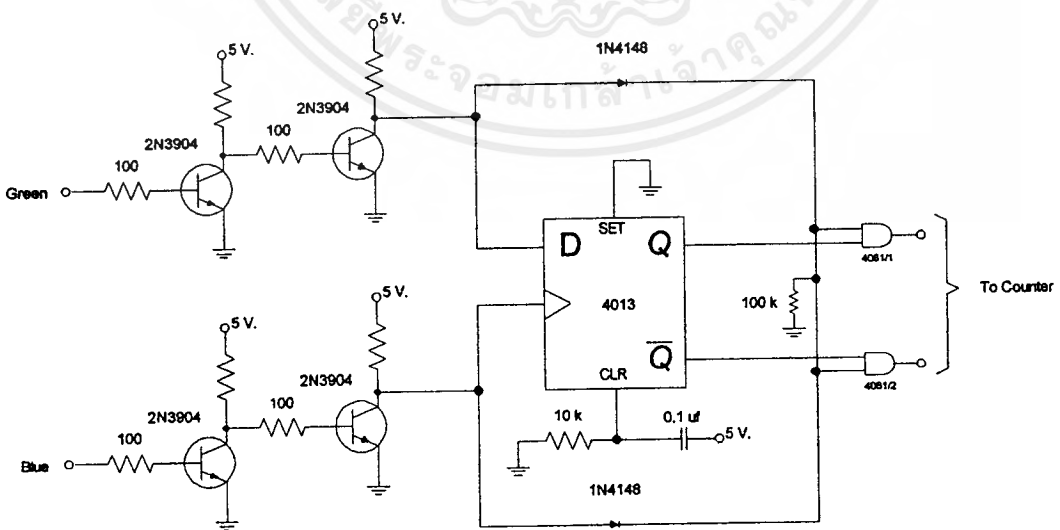
1. วงจรรนับพัลส์
2. วงจรแยกความต่างเฟส

#### 4.1.2 วงจรรนับพัลส์ ดังแสดงในรูปที่ 4.1



รูปที่ 4.1 วงจรรนับพัลส์

#### 4.1.2 วงจรแยกความต่างเฟส ดังแสดงในรูปที่ 4.2

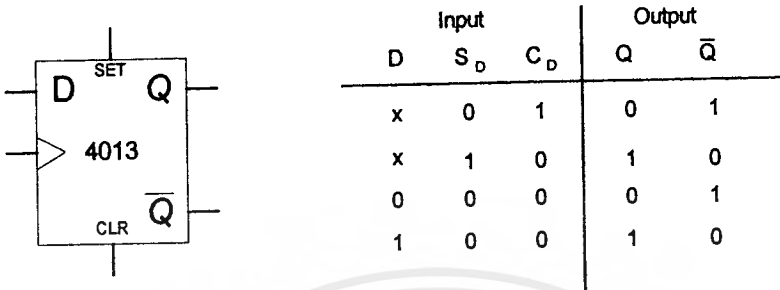


รูปที่ 4.2 วงจรแยกความต่างเฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนที่เป็น Transistors เรียงกันนั้น มีหน้าที่แต่งเหลี่ยมสัญญาณเท่านั้น ไม่มีผลในทางตรรกใดๆเลย ดังแสดงในรูปที่ 4.3

### Truth Table of D-flip flop

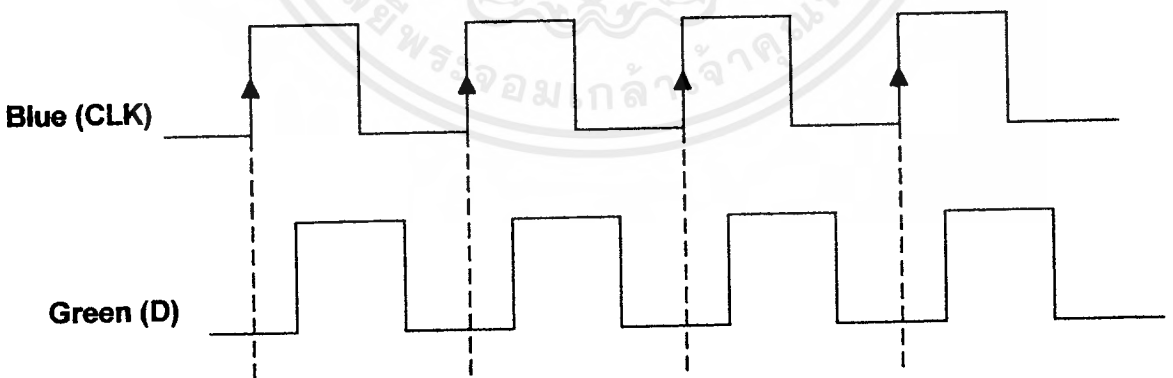


รูปที่ 4.3 แสดงค่าความจริง

ส่วนสำคัญของวงจรมีก็คือ D Flip flop โดย Flip flop ชนิดนี้ บิดข้อมูลที่เข้ามาที่อินพุต D ข้อมูลจะถูกถ่ายโอนไปยัง output ขณะที่สัญญาณนาฬิกาที่เข้ามาเป็นขอบขาขึ้น (ขอบนำ) การถ่ายโอนดังกล่าวจะเกิดขึ้นในขณะที่พัลส์กำลัง เปลี่ยนสถานะจากต่ำไปสูงเท่านั้น ส่วนเมื่อพัลส์กำลังอยู่ในสถานะคงที่ ไม่ว่าจะต่ำหรือสูง จะไม่มีการถ่ายโอนข้อมูล เกิดขึ้นอย่างเด็ดขาด

### เมื่อ Encoder หมุนทวนเข็มนาฬิกา

แสดงลักษณะสัญญาณให้เห็นในรูปที่ 4.4 ข้างล่าง



รูปที่ 4.4 สัญญาณเมื่อ Encoder หมุนทวนเข็มนาฬิกา

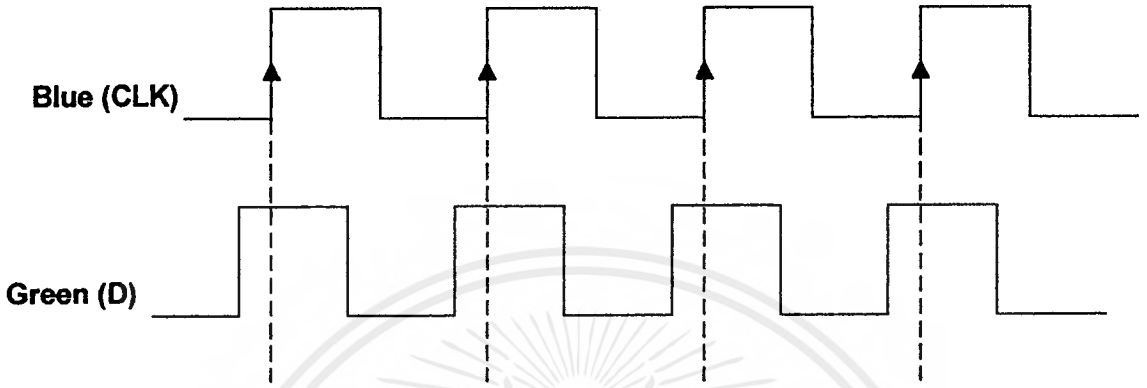
จะเห็นว่าเมื่อ ขอบสัญญาณนาฬิกาขาขึ้น ที่ขา D จะเป็น 0 เสมอ ดังนั้นที่ Q จะเป็น 0 และ  $\bar{Q}$  เป็น 1 ตลอดการหมุนสัญญาณ Q และ  $\bar{Q}$  จะถูกนำมา AND กับพัลส์ที่ผลิตโดย encoder ในกรณีนี้ ที่

สัญญาณที่ผ่าน And Gate จะเป็น 0 เสมอ ส่วน 4081/2 จะมีลักษณะสัญญาณเป็นพัลส์ที่มีลักษณะ  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความถี่เท่ากับที่ออกจาก encoder

เมื่อ Encoder หมุนตามเข็มนาฬิกา

มีลักษณะดังรูปที่ 4.5 แสดงข้างล่าง

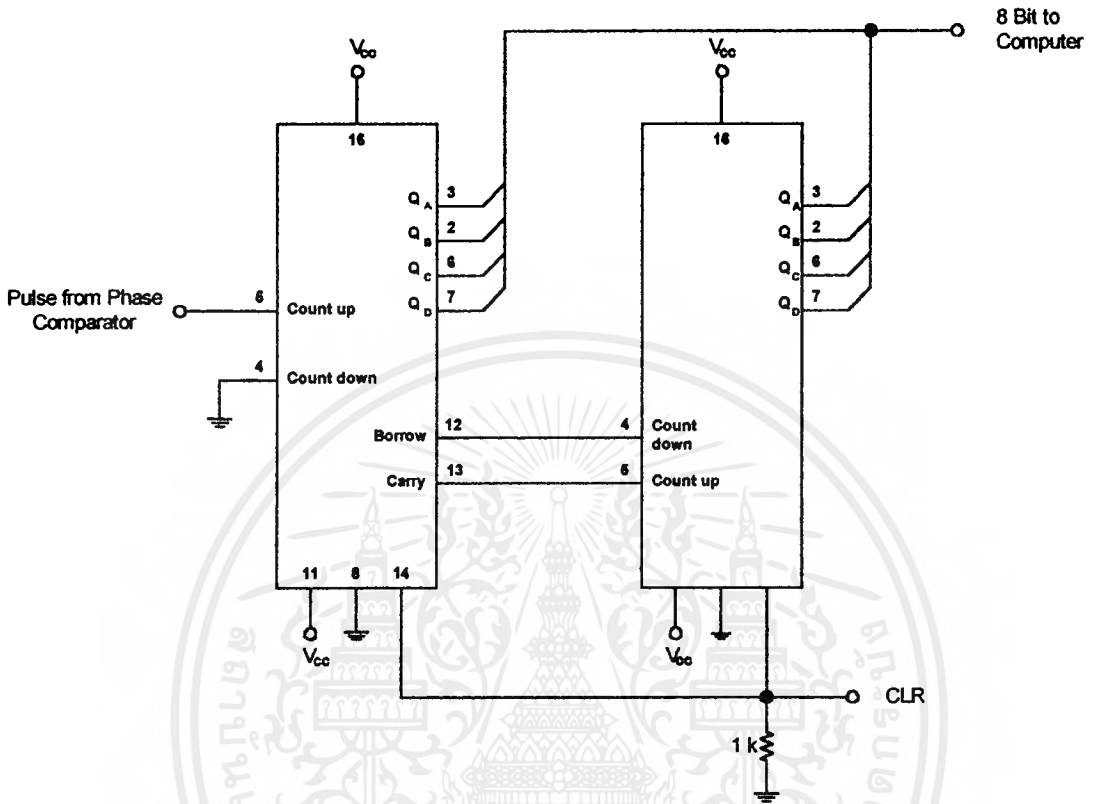


รูปที่ 4.5 ลักษณะสัญญาณเมื่อ Encoder หมุนตามเข็มนาฬิกา

เมื่อขอบสัญญาณนาฬิกาขาขึ้น ที่ขา D จะเป็น 1 เสมอ ดังนั้น Q จะเป็น 1 และ  $\bar{Q}$  จะเป็น 0 ตลอดการหมุน สัญญาณ Q และ  $\bar{Q}$  จะถูกนำมา AND กับพัลส์ที่ผลิตโดย Encoder ในกรณีนี้ 4081/1 จะมีลักษณะสัญญาณเป็นพัลส์ความถี่เท่ากับ พัลส์ที่ถูกผลิตโดย encoder ส่วน 4081/2 จะเป็น 0 ตลอดการหมุน

#### 4.1.2 วงจรนับ

เลือกใช้ IC#74LS193 เป็นตัวนับพัลส์ซึ่งเป็นวงจรรนับ 4 บิต ใน 1 วงจรรนับจะใช้ 2 ตัว ต่อกัน เป็นวงจรรนับ 8 บิต ดังที่แสดงในรูปที่ 4.6 ข้างล่างนี้



รูปที่ 4.6 วงจรรนับ

สัญญาณที่ผ่าน AND GATE ทั้ง 2 ตัว (4081/1, 4081/2) จะมีวงจรรนับของตัวเองเป็นอิสระ กรณีหมุนทวนเข็มนาฬิกา ที่ 4081/2 จะมีสัญญาณพัลส์ แต่ 4081/1 สัญญาณพัลส์ ที่นับได้จะถูกต่อกับ PA 1 (ตรงกับ address ที่ 304H)

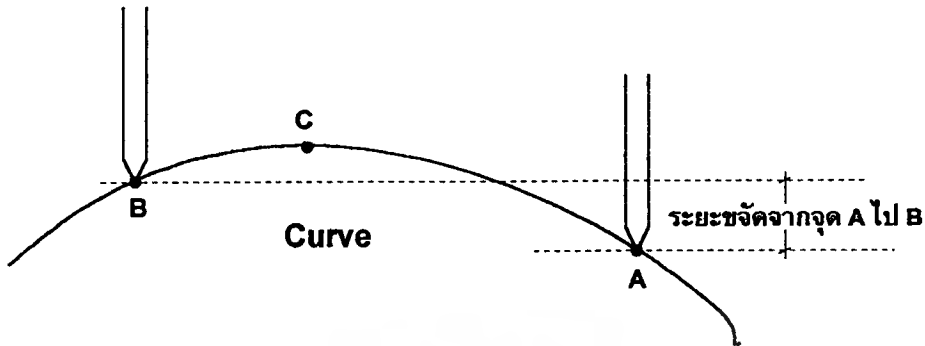
กรณีหมุนตามเข็มนาฬิกา ที่ 4081/1 จะมีสัญญาณพัลส์ แต่ 4081/2 สัญญาณพัลส์ที่นับได้ จะถูกต่อกับ PC1 (ตรงกับ address ที่ 306H)

ขา CLR ของทั้ง 2 วงจรจะถูกต่ออยู่กับขาใดขาหนึ่งของ PB1 (ตรงกับ address ที่ 305H) ดังนั้น การ clear ค่าก็ส่งข้อมูล FFH ไปที่ PB1 (305H) นี้

จากรูปข้างต้น PA1, PC1 เป็น input เข้า card 8255 PB1 เป็น output จาก card 8255

ดังนั้น Byte Control ที่ address 307H เป็น 99H (158 decimal)

เหตุที่ต้องแยกให้วงจรนับ นับแยกเป็นอิสระแก่กัน ในกรณีที่หมุนทวนเข็มนาฬิกาและตามเข็มนาฬิกา ก็เพราะว่า กรณีที่หัวโพรบ พาดผ่านจุดซึ่งเป็นจุดยอดของส่วนโค้ง ดังที่แสดงในรูปที่ 4.7 ข้างล่าง



รูปที่ 4.7 ลักษณะการเคลื่อนที่ของโพรบ

สมมติว่าจาก A ไป B หัว Probe เคลื่อนที่ ขึ้น ทำให้ Encoder หมุนทวนเข็มนาฬิกาที่ Address 304H นับได้ 120 pluse ที่ 306H นับได้ 0 pluse ต่อมาจาก ไป ที่ นับได้ นับได้ ดังนั้นระยะขจัดจาก A ไป B ในแนวตั้งคือ  $120 - 80 = 40$  pluse เมื่อได้จำนวน สุทธิที่ได้แล้ว ก็จะถูกรวมผลแปลงค่าเป็นความสูง ในหน่วย มิลลิเมตร ต่อไปในคอมพิวเตอร์ จากนั้น ข้อมูลก็จะถูก CLR ทิ้งเพื่อนับค่าต่อไป

## บทที่ 5

### CARD PORT

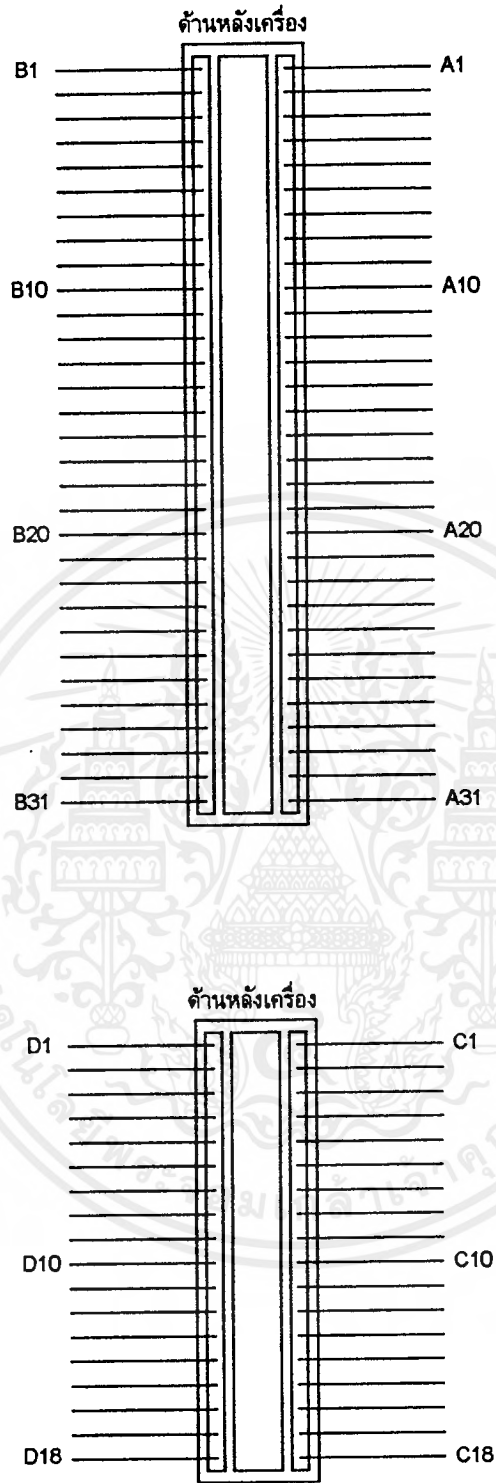
การควบคุมอุปกรณ์ไฟฟ้า โดยใช้เครื่องคอมพิวเตอร์นั้น จะต้องทำให้คอมพิวเตอร์ติดต่อกับอุปกรณ์ภายนอกให้ได้เสียก่อน แต่การที่จะทำเช่นนั้นได้ต้องผ่านอุปกรณ์ตัวหนึ่งเสียก่อน ซึ่งเรียกว่า พอร์ต (PORT) ซึ่งมีหลายลักษณะด้วยกัน แต่ในที่นี้จะใช้ IC#8255 ซึ่งเป็นพอร์ตขนานซึ่งทำมา รูปของ CARD ประกอบด้วยส่วนประกอบต่าง ๆ รวมอยู่ใน CARD แผ่นเดียว ซึ่งจะกล่าวในรายละเอียดต่อไป

#### 5.1 อุปกรณ์บน CARD

IC#8255	3	ตัว
IC#74LS139	1	ตัว
IC#74F245	1	ตัว
IC#74LS245	1	ตัว
IC#74LS688	1	ตัว
CONNECTOR 34 PIN	3	ตัว
DIP SW 8 จุด	1	ตัว
RESISTOR-PACK 10 K	11	ตัว
RESISTOR 330	1	ตัว
REISITOR 4.7 K	1	ตัว
CAPASISTOR 0.1 UF	1	ตัว
CAPASISTOR 33 UF 16 V	2	ตัว

ใน CARD นี้จะมี 2 ส่วนใหญ่ด้วยกัน ซึ่งจะเป็นส่วนของวงจร DECODE และส่วนของ PORT 8255 ซึ่งจะอธิบายในรายละเอียดต่อไป

ที่เครื่องคอมพิวเตอร์นั้นจะมีช่องเสียบเรียกว่า SLOT PC โดยการเอา CARD มาเสียบที่ SLOT นี้ เพื่อเป็นการเชื่อมต่อระหว่างเครื่องคอมพิวเตอร์กับอุปกรณ์ภายนอก ดังแสดงในรูปที่ 5.1



รูปที่ 5.1 การนับขาของสล็อตแบบ 62 ขาและ 32 ขาตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ชื่อของสัญญาณขาต่าง ๆ ของ SLOT

ขา อินพุต / เอาท์พุท	ชื่อสัญญาณ	อินพุต / เอาท์พุท
A1	-I/O CH CK	I
A2	SD7	I/O
A3	SD6	I/O
A4	SD5	I/O
A5	SD4	I/O
A6	SD3	I/O
A7	SD2	I/O
A8	SD1	I/O
A9	SD0	I
A10	I/O CH RDY	O
A11	AEN	I/O
A12	SA19	I/O
A13	SA18	I/O
A14	SA17	I/O
A15	SA16	I/O
A16	SA15	I/O
A17	SA14	I/O
A18	SA13	I/O
A19	SA12	I/O
A20	SA11	I/O
A21	SA10	I/O
A22	SA9	I/O
A23	SA8	I/O
A24	SA7	I/O
A25	SA6	I/O
A26	SA5	I/O
A27	SA4	I/O
A28	SA3	I/O
A29	SA2	I/O

A30	SA1	I/O
A31	SA0	I/O
B1	GND	กราวนด์
B2	RESET DRV	O
B3	+ 5 VDC	แหล่งจ่ายไฟเลี้ยง
B4	IR Q9	I
B5	- 5 VDC	แหล่งจ่ายไฟเลี้ยง
B6	DR Q2	I
B7	- 12 VDC	แหล่งจ่ายไฟเลี้ยง
B8	OVS	I
B9	+ 12 VDC	แหล่งจ่ายไฟเลี้ยง
B10	GND	กราวนด์
B11	- SMEMW	O
B12	- SEMER	O
B13	- IOW	I/O
B14	- IOR	I/O
B15	- DACK 3	O
B16	DR Q3	I
B17	- DACK 1	O
B18	DRQ1	I
B19	- REFRESH	I/O
B20	CLK	I
B21	IR Q7	I
B22	IR Q6	I
B23	IR Q5	I
B24	IR Q4	I
B25	IR Q3	I
B26	- DACK 2	O
B27	T/C	O
B28	BALE	O
B29	+ 5 VDC	แหล่งจ่ายไฟเลี้ยง
B30	OSC	O

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้วยการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

B31	GND	กราวนด์
C1	SBHE	I/O
C2	LA23	I/O
C3	LA22	I/O
C4	LA21	I/O
C5	LA20	I/O
C6	LA19	I/O
C7	LA18	I/O
C8	LA17	I/O
C9	- MEMR	I/O
C10	- MEMW	I/O
C11	SD08	I/O
C12	SD9	I/O
C13	SD10	I/O
C14	SD11	I/O
C15	SD12	I/O
C16	SD13	I/O
C17	SD14	I/O
C18	SD15	I/O
D1	- MEM CS16	I
D2	- I/O CS16	I
D3	IR Q10	I
D4	IR Q11	I
D5	IR Q12	I
D6	IR Q15	I
D7	IR Q14	I
D8	- DACK0	O
D9	DRQ0	I
D10	- DACK5	O
D11	DRQ5	I
D12	- DACK6	O
D13	DRQ6	I

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้拿去ใช้ประโยชน์ด้วยวิธีการ

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

D14	- DACK7	O
D15	DRQ7	I
D16	+ 5 VDC	แหล่งจ่ายไฟเลี้ยง
D17	- MASTER	I
D18	GND	กราวด์

### สัญลักษณ์ในตาราง

- I คือ INPUT  
 O คือ OUTPUT  
 I/O คือ INUPUT/OUTPUT

### สัญญาณที่ออกจาก SLOT PC ที่ใช้กับ CARD มีดังนี้

- A0 ~ ALL : เป็นแอดเดรสของระบบที่ใช้ติดต่อกับหน่วยความจำและอุปกรณ์อินพุท/เอาต์พุท  
 D0 ~ D7 : เป็นสัญญาณข้อมูลขนาด 8 บิต ที่ใช้ติดต่อกับหน่วยความจำของไมโครโปรเซสเซอร์  
 IORQ : เป็นสัญญาณอ่านอินพุท/เอาต์พุท เป็นสัญญาณที่ส่งมาจาก CPU จะแอกทีฟที่ "0"  
 IOW : เป็นสัญญาณเขียนข้อมูลลงบนอุปกรณ์อินพุท/เอาต์พุทสัญญาณนี้ควบคุมจากไมโครโปรเซสเซอร์ แอกทีฟที่ "0"  
 RES : สัญญาณนี้ใช้สำหรับรีเซ็ตระบบในขณะที่เปิดเครื่องหรือขณะที่แหล่งจ่ายไฟเลี้ยงขาด  
 AEN : อินาเบลแอดเดรส (เป็นเอาต์พุท)

เมื่อรู้สัญญาณต่าง ๆ ที่ออกมาจากเครื่องคอมพิวเตอร์แล้ว ก็มาดูรายละเอียดของวงจรบน CARD บ้าง ซึ่งมี 2 ส่วนดังที่กล่าวมาแล้วในตอนต้นแต่ทั้งสองส่วนจะทำงานร่วมกัน ส่วนของวงจร DECODE ซึ่งใน CARD นี้ประกอบด้วย IC#74LS688 74LS139 และ DIP SW. 3 PIN เพื่อที่สามารถปรับ SET DIP SW. ตั้งตำแหน่งพอร์ตพอร์ท PORT ของ CARD ได้โดยเพียงแค่ปรับที่ DIP SW. เท่านั้น สิ่งที่ต้องระวังคือ ในการปรับ DIP SW นั้นจะต้องให้ตรงกับตำแหน่ง PORT ที่เครื่องคอมพิวเตอร์ใช้อยู่แล้ว ส่วน IC#74F254 ทำหน้าที่เป็น BUFFER 2 ทางคือ อยู่ในสถานะรับข้อมูลหรือส่งข้อมูลโดยการกำหนดที่เครื่องคอมพิวเตอร์ ส่วน 74LS245 ก็เป็นบัฟเฟอร์แต่ถูกใช้ให้ข้อมูลจาก A ไป B เท่านั้น

## 5.2 การทำงานของวงจร

เริ่มต้นที่ SLOT PC โดยจากวงจรจะเห็นว่า D0~D7 จะต่อเข้ากับขา A1~A8 ของ IC#74F245 และที่ขา B1~B8 ของ IC#74F245 จะต่อเข้ากับขาที่ D0~D7 ของ PORT 8255 ทั้ง 3 ตัว โดยที่ขา DIR ของ 74F245 จะต่อเข้ากับขา B0 ของ 74F245 เหตุที่ต่อเช่นนี้เพราะขา IOR ของ SLOT PC ต่อกับขา A0 ของ 74LS245 ซึ่งจะเป็นตัวควบคุมว่าตอนนี้อยู่ในสภาวะรับหรือส่งข้อมูล ส่วนขา IOW ของ SLOT PC จะต่อกับขา A1 ของ IC 74LS245 ซึ่งขา B0, B1 ของ 74LS245 ก็จะไปต่อกับขา IOR, IOW ของพอร์ท 8255 ทุกตัวเช่นเดียวกัน เพื่อเป็นการบอกสถานะให้รู้ว่าจะรับหรือส่งข้อมูล ส่วนขา RESET A0, A1 ของ SLOT PC ก็จะต่อเข้ากับขา A2, A3, A6 ของ 74LS245 โดยขา B7 จะต่อเข้ากับขา RESET ทุกตัวของพอร์ท 8255 ส่วนขา B2, B3 จะต่อเข้ากับขา A0, A1 ของ 8255 ทุกตัว เพื่อเป็นการชี้ว่าจะใช้งานพอร์ทไหน และให้เป็นอินพุทหรือเอาต์พุท โดยเช็ทที่ A0, A1 ที่มาจาก SLOT PC จะเห็นว่าขา E ของ 74F245 จะมีไฟเลี้ยงตลอดทำให้ 74F245 ไม่ทำงานจนกว่า จะมีลอจิก "0" จากเอาต์พุทของ 74LS688 มาเท่านั้น ส่วน 74LS245 จะทำงานตลอดเพราะต่อขา E ลงกราวด์และที่ขา DIR ก็มีไฟเลี้ยงตลอดเวลา ซึ่งเป็นการเช็ทให้ข้อมูลส่งจาก A ไป B เท่านั้น ส่วนขา A2, A3 ของ SLOT PC จะเข้ากับขา A4, A5 ของ 74LS245 และออกที่ขา B4, B5 ไปเข้ากับขา A, B ของ 74LS139 ซึ่งเป็นการเช็ทให้ข้อมูลส่งจาก A ไป B เท่านั้น ส่วนขา A2, A3 ของ SLOT PC จะเข้ากับขา A4, A5 ของ 74LS245 และออกที่ขา B4, B5 ไปเข้ากับขา A, B ของ 74LS139 ซึ่งเป็นวงจร DECODE 2 LINE TO 4 LINE โดยขา A2] A3 ของ SLOT PC จะทำหน้าที่เลือกจะให้ 8255 ตัวไหนทำงานซึ่งมี 3 ตัวด้วยกันการที่จะส่งข้อมูลหรือรับข้อมูลนั้น เราต้องอ้างพอร์ทของ CARD ก่อนจากวงจรจะใช้ 74LS688 ซึ่งเป็นวงจร COMPARATOR โดยขา A4~A11 ของ SLOT จะต่อเข้ากับ P0~P7 และที่ DIP SW. ขา 1-8 จะต่อเข้ากับขา Q0~Q7 ของ 74LS688 ซึ่งถ้า ADDRESS ที่ส่งมาตรงกับที่ตั้ง DIP SW ไว้ ก็จะทำให้ขา P=Q แอคทีฟ "0" ซึ่งทำให้ขา E ของ 74F245 แอคทีฟ ทำให้รู้ว่า 8255 ตัวไหนทำงานที่ตำแหน่งที่ตำแหน่งพอร์ทเท่าไร และใช้พอร์ทไหนในการทำงาน และเป็นการรับหรือส่งข้อมูล

ตัวอย่างเช่น

เราตั้ง DIP SW ไว้ที่ตำแหน่ง 300 (00110000) เมื่อเราป้อน A0~A11 มาเป็น 300H คือ

A0~A1 เป็น 00 จะทำให้รู้ว่าเราใช้ PORT A

A2~A3 เป็น 00 จะทำให้ 74LS139 DECODE ออกที่ YO ที่ให้ 825 ตัวที่ 1 ทำงาน

A4~A11 เป็น 00001100 ก็จะตรงกับที่ตั้ง DIP SW ตั้งไว้ก็ทำให้วงจรทำงานสมบูรณ์

ในการอ้างพอร์ท (กรณี A4~A11 = 30XH)

8255ตัวที่ 1 PORT A = 300H

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับวิชาการเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PORT B = 301H

PORT C = 302H

CONTROL PORT1 = 303H

8255 ตัวที่ 2 PORT A = 304H

PORT B = 305H

PORT C = 306H

CONTROL PORT2 = 307H

8255 ตัวที่ 3 PORT A = 308H

PORT B = 309H

PORT C = 30AH

CONTROL PORT = 30BH

หมายเลข PORT สามารถเปลี่ยนแปลงได้แต่ห้ามไม่ให้ตรงกับที่ใช้อยู่ในเครื่องคอมพิวเตอร์  
อยู่แล้ว ดูได้จากตารางที่ 5.1

ตารางที่ 5.1 แสดงหมายเลขพอร์ท

หมายเลขพอร์ทรหัสฐานสิบหก	ชื่ออุปกรณ์
000-01F	DMA คอนโทรลเลอร์หมายเลข 1 8237A-5
020-03F	อินเตอร์รัพเตอร์คอนโทรลเลอร์หมายเลข 1 8259A ตัวหลัก
040-05F	ไทมเมอร์ 8254-2
060-06F	8042 คีย์บอร์ด
070-07F	นาฬิกาและ NMI และซีโมสแรม
080-09F	DMA เพจรีจิสเตอร์
0A0-0BF	อินเตอร์รัพต์คอนโทรลเลอร์หมายเลข 2 8237-5
0C0-0DF	DMA คอนโทรลเลอร์หมายเลข 2 8237A-5
0F0	เคลียร์โปรเซสเซอร์คณิตศาสตร์
0F1	รีเซตโปรเซสเซอร์คณิตศาสตร์
0F8-0FF	โปรเซสเซอร์คณิตศาสตร์
1F0-1F8	ฮาร์ดดิสก์
200-207	เกมไอโอ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## แผนผังวงจรรภายในและการจัดขาของไอซี 8255

จากรูปจะเห็น โครงสร้างภายในที่แสดงถึงกลุ่มควบคุมที่มีอยู่ 3 กลุ่มคือ

- กลุ่มควบคุมชุด A จะควบคุมพอร์ท A และพอร์ท C บน
- กลุ่มควบคุมชุด B จะควบคุมพอร์ท B และพอร์ท C ล่าง
- กลุ่มควบคุมลอจิกการเขียนและอ่าน

การทำงานของ 8255 จะใช้สัญญาณควบคุมจากตัวไมโครโปรเซสเซอร์ มาควบคุมโดยจะมีการตั้งคำสั่ง (Control Word) มาที่กลุ่มควบคุมชุด A, B แล้วกลุ่มควบคุมชุดนี้ก็จะส่งต่อไปที่พอร์ทเพื่อให้เป็นไปตามข้อกำหนดของคำสั่งนั้น ๆ เช่น ให้พอร์ท A เป็นอินพุทพอร์ท B เป็นเอาต์พุทพอร์ท เหล่านี้เป็นต้น ส่วนกรณีเมื่อมีการอ่านเขียนพอร์ทจาก CPU นั้น กลุ่มควบคุมลอจิกการเขียน และอ่านจะเป็นตัวที่ส่งสัญญาณ ไปบอกแก่กลุ่มควบคุมในแต่ละชุดอีกที ทั้งนี้แล้วแต่ว่า CPU จะมีการอ่านเขียนพอร์ทของกลุ่มควบคุมชุดใด

ขาต่าง ๆ ของไอซี 8255

D0~D7 เป็นขาของข้อมูลของ 8255 ที่ใช้ติดต่อกับตัวไมโครโปรเซสเซอร์ซึ่งข้อมูลที่จะเข้าออก พอร์ทต่าง ๆ ของ 8255 จะต้องผ่านขาข้อมูลนี้

CS เป็นขาอินพุทที่รับสัญญาณลอจิก "0" จากภายนอกเพื่อแสดงว่าต้องการเลือกใช้ไอซี เบอร์นี้ หากได้รับลอจิก "1" ก็จะทำให้ไอซีตัวนี้ไม่ทำงานคือไม่รับรู้สัญญาณใด ๆ ทั้งสิ้น

RD เป็นขาอินพุทที่รับสัญญาณจากตัวไมโครโปรเซสเซอร์ โดยหากมีลอจิกเป็น "0" จะเป็นการแสดงว่า CPU ต้องการที่จะอ่านข้อมูลจากตัว 8255

WR เป็นขาอินพุทที่รับสัญญาณจากตัวไมโครโปรเซสเซอร์ โดยหากมีลอจิกเป็น "0" ก็จะเป็น การแสดงว่า CPU ต้องการที่จะเขียนข้อมูลจากตัว 8255

A0~A1 เป็นอินพุทที่รับแอดเดรสจากตัวไมโครโปรเซสเซอร์ที่ถอดรหัสตำแหน่งของ 8255 เรียบร้อยแล้วโดยจะมีตำแหน่งใช้งาน 4 ตำแหน่งเพื่ออ่านเขียนรีจิสเตอร์ (พอร์ท) ของ 8255 ที่อยู่ด้วยกัน 4 ตัว

RESET เป็นขาอินพุทที่นับสัญญาณจากภายนอกเข้ามาทำการรีเซ็ตตัว 8255 โดยหากได้รับลอจิก "1" จะทำให้พอร์ททุกพอร์ทเป็นอินพุทพอร์ททั้งหมด เมื่อไม่ต้องการให้มีสัญญาณ ออกไปรบกวนต่อระบบภายนอกเพื่อ 8255 ได้รับสัญญาณรีเซ็ต

PA0~PA7 เป็นขาสัญญาณพอร์ท A ที่ใช้ติดต่อกับอุปกรณ์ภายนอก

PB0~PB7 เป็นขาสัญญาณพอร์ท B ที่ใช้ติดต่อกับโลกภายนอก

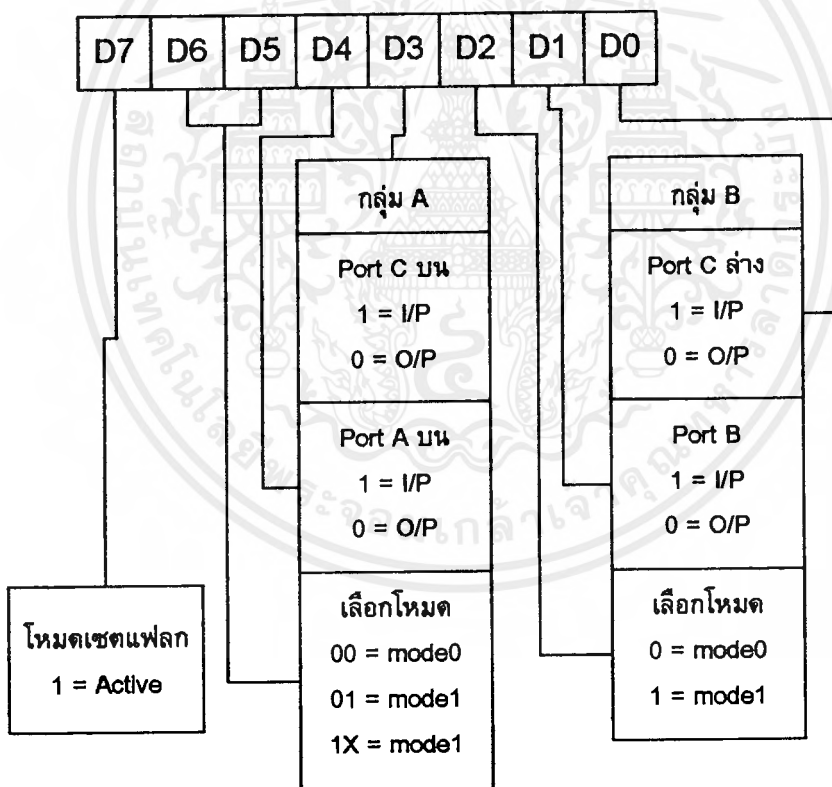
PC0~PC7 เป็นขาสัญญาณพอร์ท C ที่ใช้ติดต่อกับอุปกรณ์ภายนอกซึ่งพอร์ทนี้จะแบ่งออกเป็น 2

กลุ่มคือ PC0-PC3 และ PC4-PC7 ซึ่งสามารถโปรแกรมแยกกันได้อีกต่างหาก

### 5.3 การโปรแกรม 8255

เราได้ทราบแล้วในรูปที่ 5.2 ว่าโครงสร้างภายในของ 8255 มีกลุ่มควบคุมชุดอยู่ 3 กลุ่มซึ่งทั้ง 3 กลุ่มนี้จะทำงานร่วมกันดังที่กล่าวมาและเราสามารถจะควบคุมการทำงานของพอร์ตจาก CPU โดยตั้งงานมาที่กลุ่มควบคุมดังกล่าว แต่ตัว CPU จะมองเห็น 8255 เป็น 4 พอร์ตด้วยกันโดยแต่ละพอร์ตเสมือนเป็นรีจิสเตอร์ที่ CPU สามารถจะทำการอ่าน / เขียนได้ซึ่งหากมีการอ่านเขียนไปยังพอร์ตดังกล่าวก็จะใช้ร่วมกับสัญญาณ RD โดยสัญญาณ WR หมายถึง เอาท์พุทข้อมูลและ RD แอทิฟ หมายถึง อินพุทข้อมูล

การใช้งานเราจะต้องส่งรหัสควบคุม (Control Code) เข้าไปยังพอร์ตควบคุม (หรือเรียกอีกอย่างว่ารีจิสเตอร์ควบคุม) ซึ่งจะเป็นข้อมูลขนาด 1 ไบต์ส่งไปที่ แอดเดรส 13H (กรณีนี้เราถอดรหัสไว้ที่ 13H) โดยความหมายของแต่ละบิตที่เราไปโปรแกรม การทำงานเป็นดังรูปที่ 5.3



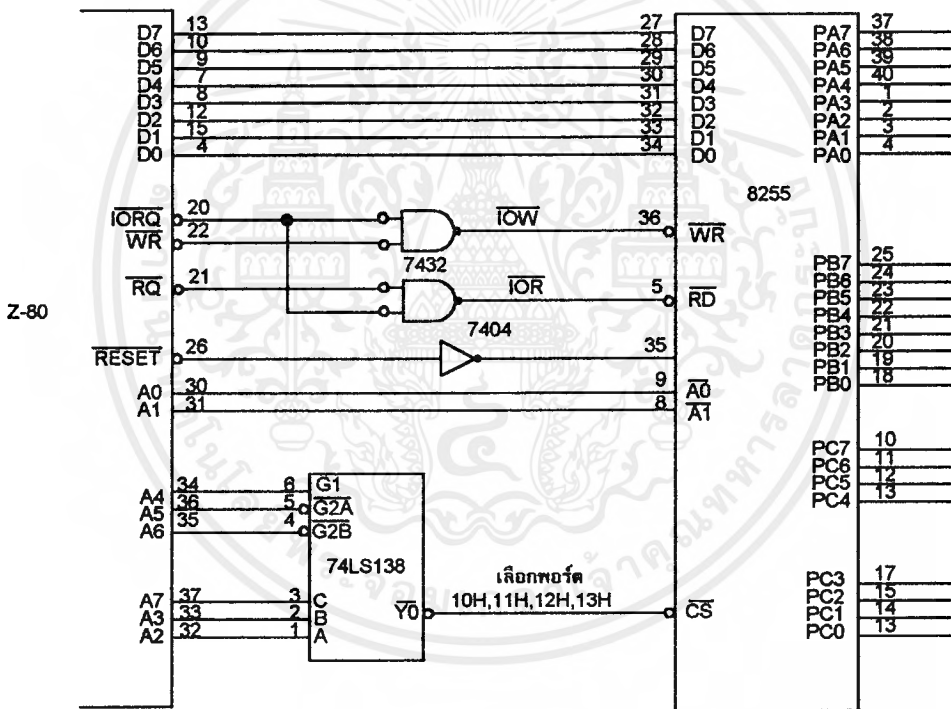
รูปที่ 5.3 ความหมายของบิตต่างๆในรหัสควบคุม

#### การทำงานในโหมด 0

จัดว่าเป็นโหมดพื้นฐานที่นิยมใช้กันมากที่สุด ทั้งสามพอร์ตเราสามารถจะใช้ให้พอร์ตใดเป็นอินพุท เอาท์พุทได้ โดยเฉพาะพอร์ต C ยังแยกให้เป็น 2 ชุด ๆ ละ 4 บิต ซึ่งในแต่ละชุดนี้ก็สามารถเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

IORQ จะหมายถึง สัญญาณ IOW หรือสัญญาณเขียนพอร์ท และถ้าให้สัญญาณ IORQ แยกที่พ  
พร้อมกับสัญญาณ RD จะหมายถึงสัญญาณ IOR หรือสัญญาณอ่านพอร์ท ซึ่งการเชื่อมต่อสาย  
สัญญาณควบคุมการเขียนและการอ่านหน่วยความจำ

เมื่อเชื่อมต่อระบบ จะต้องมีการเชื่อมสัญญาณ RESET ของ Z-80 มายังขา RESET ของ 8255  
การรีเซ็ตของ 8255 ใช้ “1” ซึ่งตรงสัญญาณควบคุมการข้ามกับ Z-80 ดังนั้นจำเป็นต้องมี อินเวอร์  
เตอร์เปลี่ยนลอจิกก่อน การที่ต้องรีเซ็ต 8255 พร้อมกับ Z-80 ก็เนื่องจากว่า ขณะที่ Z-80 รีเซ็ตเราจะ  
เริ่มจากให้พอร์ททุกพอร์ทของ 8255 เป็นอินพุทเพื่อว่าอาจจะมีข้อมูลบางส่วน ไปออกที่พอร์ทใน  
ขณะที่เรายังไม่ต้องการ ซึ่งอาจจะมีระบบอินเตอร์เฟสภายนอกมีปัญหาได้ เพราะเราไม่รู้สถานะที่  
แน่นอนของ 8255 ก่อนการโปรแกรมโหมดการทำงาน ระบบการเชื่อมต่อของ 8255 กับ Z-80 ทั้ง  
ระบบแสดงดังรูปที่ 5.5



รูปที่ 5.5 การเชื่อมต่อ 8255 กับ Z-80 ทั้งระบบ

เมื่อต่อ 8255 เข้ากับ Z-80 ได้แล้ว สิ่งที่ใช้จะต้องทำคือ การโปรแกรมให้ 8255 ทำงานตามที่  
ต้องการ จากการที่ 8255 มีพอร์ทที่ Z-80 มองเห็น 4 พอร์ท แต่ละพอร์ทจะเสมือนเป็นรีจิสเตอร์ที่  
สามารถเขียนและอ่านได้ รีจิสเตอร์แต่ละตัวนี้จึงถูกกำหนดด้วย แอดเดรสตามที่ตั้งไว้ เช่น ในกรณี  
ที่เป็นแอดเดรส 10H, 11H, 12H และ 13H รีจิสเตอร์ แต่ละตัวจะได้รับการกำหนดควบคู่กับ  
สัญญาณ RD และ WR เพื่อแสดงความหมาย ตัวอย่างเช่น พอร์ท 10H เป็นพอร์ท A ซึ่งเมื่อได้เขียนที่  
พอร์ทนี้ จะเป็นการส่งข้อมูลเอาต์พุท และถ้าอ่านพอร์ทนี้จะเป็นการอินพุทข้อมูลจากพอร์ทดังนั้น  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณของขาคควบคุมที่ประกอบกันจะแสดง ความหมายดังตารางที่ 5.2

ตารางที่ 5.2 สัญญาณควบคุมการกระทำของ 8255

RD	wr	a1	a0	ความหมาย
1	0	0	0	เขียนพอร์ท a ซึ่งเป็นข้อมูล
0	1	0	0	อ่านพอร์ท a ซึ่งเป็นข้อมูล
1	0	0	1	เขียนพอร์ท b ซึ่งเป็นข้อมูล
0	1	0	1	อ่านพอร์ท b ซึ่งเป็นข้อมูล
1	0	1	0	เขียนพอร์ท c ซึ่งเป็นข้อมูล
0	1	1	0	อ่านพอร์ท c ซึ่งเป็นข้อมูล
1	0	1	1	เขียนข้อมูลซึ่งเป็นรหัสควบคุม
0	1	1	1	อ่านเข้ามาซึ่งไม่มีความหมายใด

การใช้งาน 8255 จะต้องส่งรหัสควบคุม (control Code) เข้าไปยังพอร์ทข้อมูลควบคุมเพื่อคุมการทำงานของ 8255 โดยใช้สัญญาณการควบคุมหมายเลข 13H การควบคุมการทำงานของ 8255 มีหลายโหมด แต่ละโหมดจะแตกต่างกันออกไป การโปรแกรมให้ 8255 ทำงานจะทำได้ 3 โหมดคือ โหมด 0 โหมด 1 และ โหมด 2

### 5.3.1 โหมด 0 หรืออินพุทเอาต์พุทแบบพื้นฐาน

การกำหนดโหมดการทำงาน จะต้องส่งข้อมูลคำสั่งเข้าโปรแกรมในโหมดควบคุมของ 8255 ซึ่งในที่นี้โหมดหมายเลข 13H แต่ละบิตของข้อมูลที่ส่งไปจะมีความหมายในตัวเอง ลักษณะความหมายแต่ละบิตลักษณะควบคุมแสดงดังรูป การโปรแกรม 8255 คือ การใช้ค่ารหัสบิตต่าง ๆ เข้าไปในรหัสควบคุมแล้วส่งไปยังรีจิสเตอร์ของพอร์ทควบคุม ความหมายของบิตต่าง ๆ มีดังนี้

บิต D7 เป็นบิตที่แสดงรหัสคำสั่งควบคุม ถ้าบิตนี้เป็น "1" หมายถึงรหัสควบคุมนี้จะมีผลต่อการเปลี่ยนแปลงการเซ็ทโหมดต่าง ๆ ของ 8255

บิต D6 และ D5 เป็นการเลือกโหมดของพอร์ท A ซึ่งมี 3 โหมด คือ โหมด 0 โหมด 1 และ โหมด 2 ดังรูปที่ 5.7

ความหมายของบิตต่าง ๆ ในรหัสควบคุม

บิต D4 ถ้ามีค่าเป็น "0" หมายถึงการกำหนดพอร์ท A เป็นเอาต์พุท ถ้ามีค่าเป็น "1" จะหมายถึงการกำหนดให้พอร์ท A เป็นอินพุท

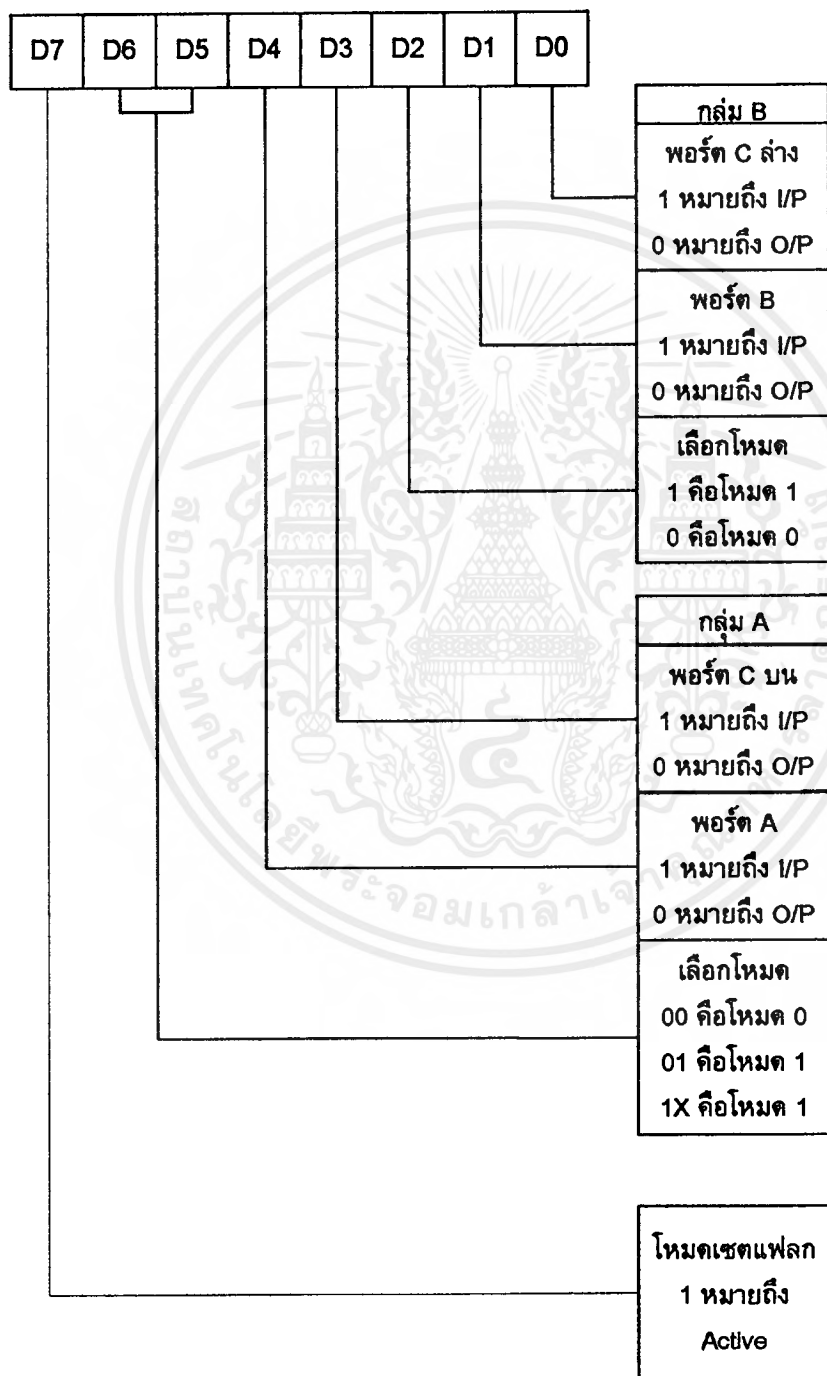
บิต D3 เป็นบิตที่บอกถึงการเซ็ทของพอร์ท C บน ถ้าเป็น "0" จะทำให้พอร์ท C บนเป็นเอาต์

พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บิต  $D_1$  เป็นการกำหนดอินพุตเอาต์พุตของพอร์ต B ถ้าเป็น “0” หมายถึง เอาต์พุตถ้าเป็น “1” หมายถึงอินพุต

บิต  $D_0$  เป็นการกำหนดอินพุตเอาต์พุตของพอร์ต C ล่างถ้าเป็น “0” หมายถึง เอาต์พุต ถ้าเป็น “1” หมายถึงอินพุต



### รูปที่ 5.7 ความหมายของบิตต่างๆในรหัสควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การโปรแกรม 8255 จะเริ่มจากเซตค่าที่ต้องการแล้วเอาต์พุตไปยังพอร์ตควบคุม ถ้าเช่น  
ต้องการโปรแกรมให้ส่งพอร์ต A, B และ C เป็นพอร์ตเอาต์พุตหมด เราจะเลือก 8255 ให้อยู่  
ในโหมด 0 โดยมีรหัสควบคุมเป็น 10000000 หรือ 80H ดังนั้นจึงเขียนคำสั่งได้เป็น

LD A,80H            หมายถึงการกำหนดรหัสควบคุม

OUT (13H),A        หมายถึงส่งไปยังพอร์ตควบคุม

หลังจากที่กระทำคำสั่ง OUT ที่ผ่านไปแล้วพอร์ต A, B และ C จะเป็นพอร์ตเอาต์พุตหมด  
ซึ่งจะส่งข้อมูลจากซีพียูไปยังพอร์ตต่างๆ ได้ เช่น ถ้าต้องการส่งข้อมูล 8AH ไปยังพอร์ต A ข้อมูล  
41H ไปยังพอร์ต B และข้อมูล 25H ไปยังพอร์ต C คำสั่งที่ใช้คือ

LD A,8AH            หมายถึงเลือกค่า 8AH

OUT [10H],A        หมายถึงส่งให้พอร์ต A

LD A,41H            หมายถึงเลือกค่า 41H

OUT [11H],A        หมายถึงส่งให้พอร์ต B

LD A,25H            หมายถึงส่งให้พอร์ต 25H

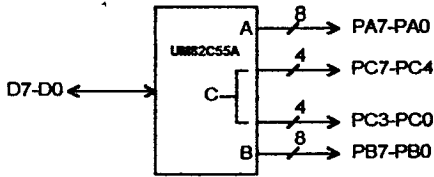
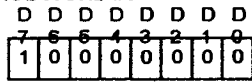
OUT [12H],A        หมายถึงส่งให้พอร์ต C

เนื่องจากมีพอร์ตสำหรับส่งข้อมูล 3 พอร์ต คือ พอร์ต A พอร์ต B และพอร์ต C ซึ่ง  
พอร์ต C จะแยกออกเป็น 2 ส่วน คือ พอร์ต C ล่าง และพอร์ต C บน เราสามารถโปรแกรม  
ให้ทั้ง 4 พอร์ตนี้เป็นอินพุตหรือเอาต์พุตก็ได้ เช่น ถ้าให้รหัสควบคุมเป็น 82H จะทำให้พอร์ต B  
เป็นอินพุตพอร์ตและพอร์ต C เป็นเอาต์พุต

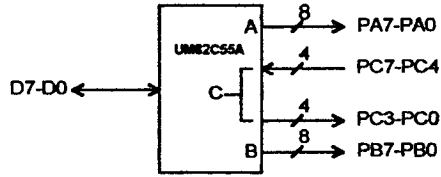
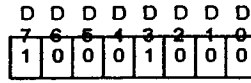
### 5.3.2 การทำงานในโหมด 0

โหมด 0 เป็นโหมดให้พอร์ตทุกพอร์ตบนตัว 8255 เป็นพอร์ตอินพุตเอาต์พุตแบบพื้นฐานรูป  
แบบความเป็นไปได้จึงมีทั้งสิ้น 16 รูปแบบตามลักษณะของพอร์ต 4 พอร์ต B พอร์ต C และ  
บน พอร์ต C ล่าง ลักษณะการควบคุมแต่ละแบบเป็นดังรูป 5.8

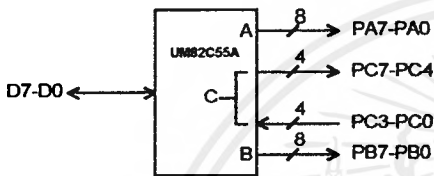
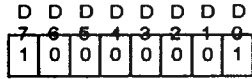
CONTROL WORD #0



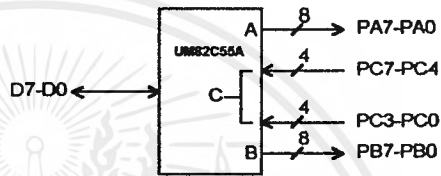
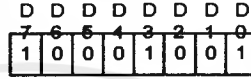
CONTROL WORD #4



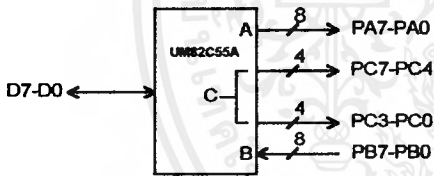
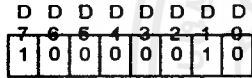
CONTROL WORD #1



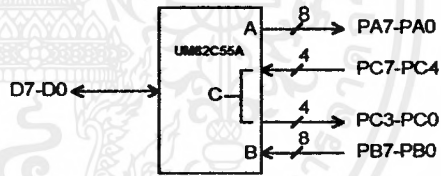
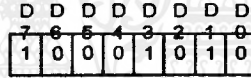
CONTROL WORD #5



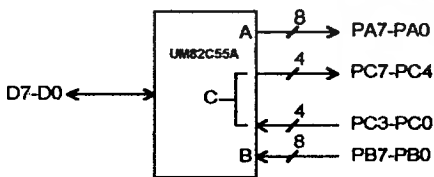
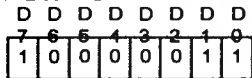
CONTROL WORD #2



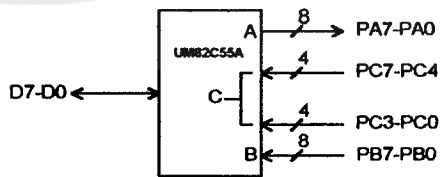
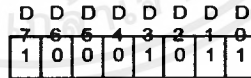
CONTROL WORD #6

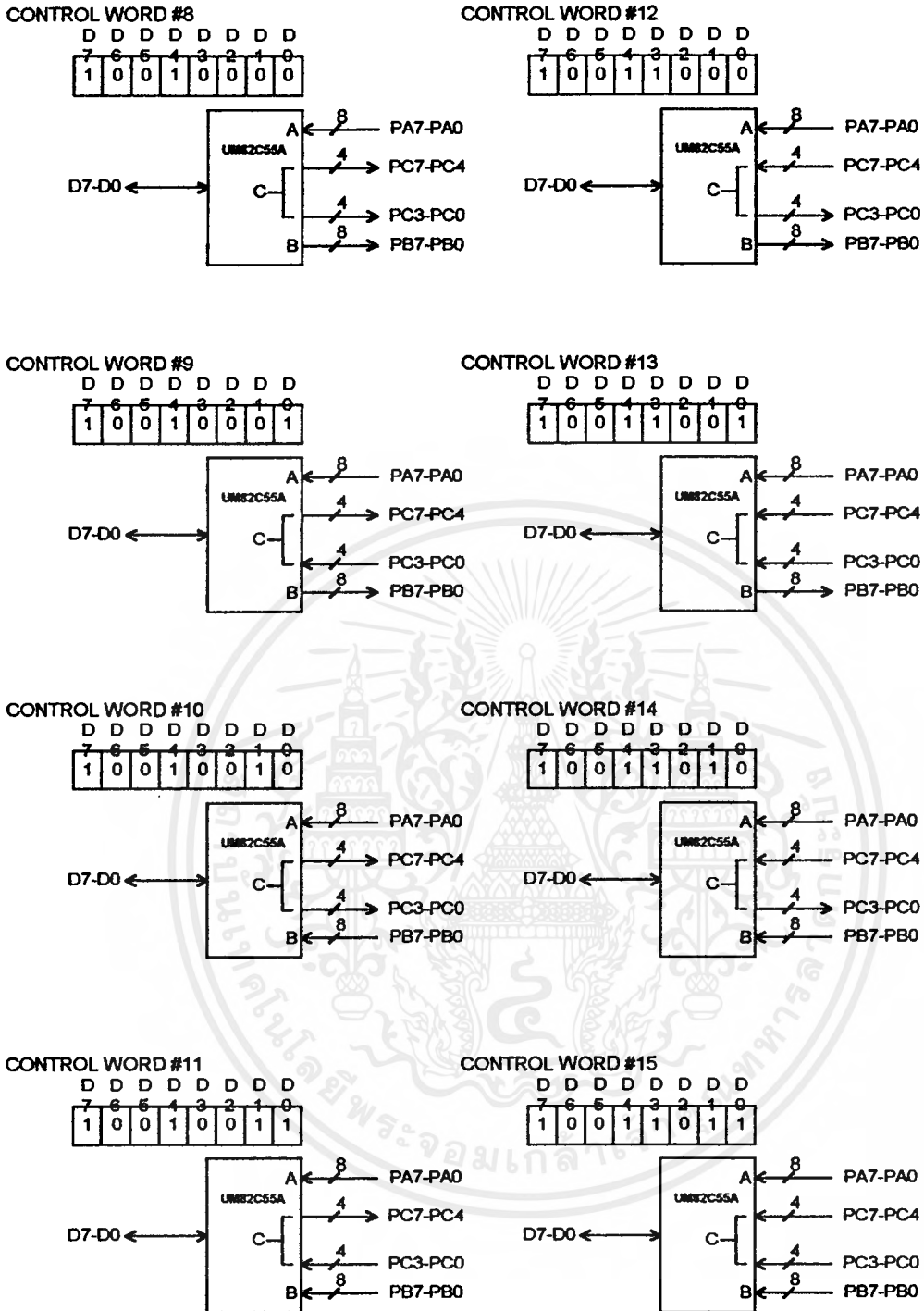


CONTROL WORD #3



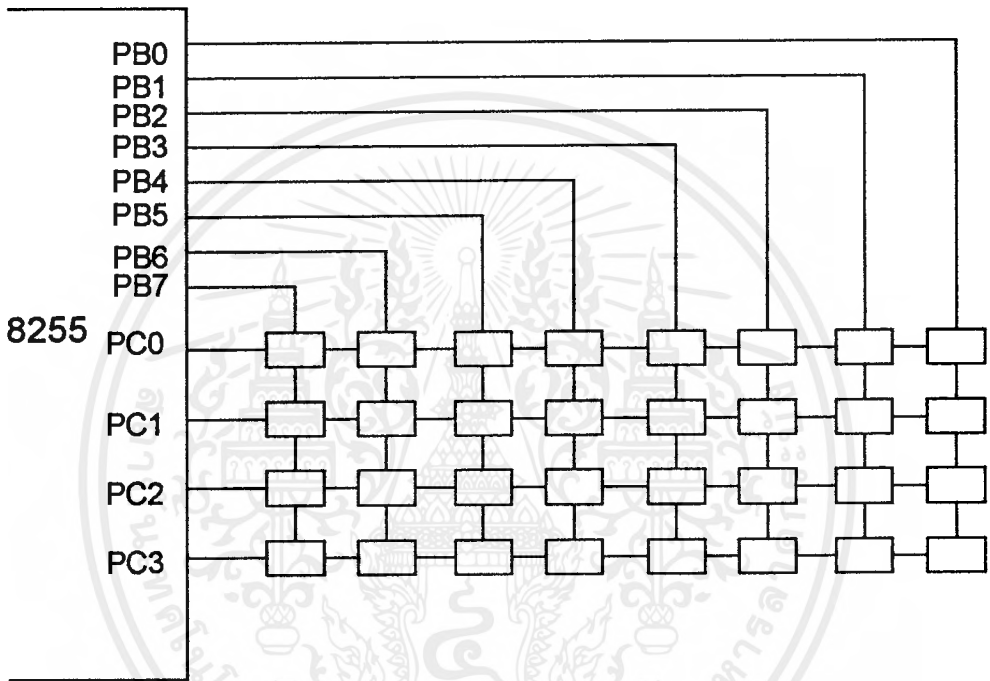
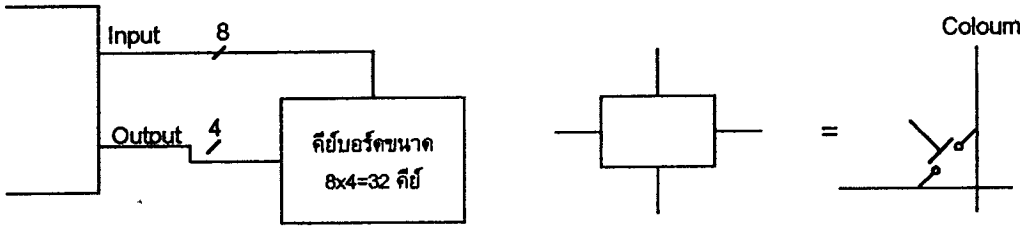
CONTROL WORD #7





รูปที่ 5.8 ลักษณะของรหัสควบคุมต่างๆในโหมด 0

ตัวอย่างการใช้งาน 8255 ในโหมด 0 นี้ เช่น เมื่อต้องการให้พอร์ต B เป็นอินพุต และพอร์ตล่างเป็นเอาต์พุต เพื่อรับรุกรการกดคีย์บอร์ดและหาความีคีย์ใดกด ซึ่งเราสามารถทำได้ จัดคีย์บอร์ด และหาความีคีย์ใดกด ซึ่งสามารถใช้การจัดคีย์บอร์ดในรูปแบบเมตริกซ์ได้ดังรูป 5.9



รูปที่ 5.9 วงจรการต่อคีย์บอร์ดขนาด 32 คีย์แบบเมตริกซ์

สาเหตุที่เรานิยมต่อคีย์จำนวนมากแบบเมตริกซ์เนื่องจากโครงสร้างทางฮาร์ดแวร์ แบบประหลาดและลดข้อยุ่งยากต่าง ๆ ได้ แต่ก็ยังต้องใช้ซอฟต์แวร์ในการควบคุมหรือตรวจสอบว่าคีย์ใดกดในรูปที่ 8 เป็นการต่อแบบเมตริกซ์ มีจำนวน 4 แถว และคอลัมน์ 8 คอลัมน์ทำให้ ได้จำนวนคีย์ทั้งสิ้น 32 คีย์

หลักการการทำงานทั่วไปเราจะทำการแสกนดู กล่าวคือกำหนดให้แต่ละแถวซึ่งพอร์ต เอต์พุต เป็น "0" หรือ "1" ในเวลาต่าง ๆ กัน เช่น แถวแรกเป็น "0" แถวอื่นเป็น "1" หมุดแล้วทำการอ่านข้อมูลที่พอร์ตพุตว่า มีบิตใดบิตหนึ่งทางคอลัมน์เป็น "0" หรือไม่ ถ้ามีก็ทราบได้ว่าแถวหรือคอลัมน์ ที่เท่าไรซึ่งเป็นคีย์ที่ได้รับการกด ลองพิจารณารูปซึ่งต่อคีย์ในลักษณะ 4 x 8 เราจะสแกนทีละแถวโดยเริ่มจากแถว 0 โดยการให้เอต์พุตที่บิต 1 เป็น "0" แล้ววนไปเรื่อยๆ โดยใช้ค่าใน

รีจิสเตอร์ C เป็นค่า สำหรับกำหนดแถว โปรแกรมที่เขียนขึ้นจะได้ดึงโปรแกรม

### 5.3.2.1 โปรแกรมส่วนตรวจสอบการปล่อยคีย์

```

RELEASE CHECK THAT KEY HAS BEEN RELEASE
RELESAE LD C,00H
CALL SCAN :เรียกโปรแกรมสแกนคีย์
IN A,11H
XOR OFF: ไม่กด = FFH
JR NZ,PRESS
INC C
LD A,40H
CP C: ครบ 4 แถว ?
RET C: ถ้าทุกคีย์ปล่อยหมด ให้ RET
JR RELEASE + 2
PRESS CALL DLY10
JR RELEASE + 5

```

### 5.3.2.2 โปรแกรมส่วนสแกนคีย์

```

SCAN KEYBOARD BY OUTPUT TOROW SCAN
SCAN LD HL, LINE
LD A,(HL)
RLC: เลื่อนไปแถวถัดไป
CP 10H: เกินแถวที่ 4 ?
JR NZ,$+4
LD A,01H: เริ่มต้นแถวที่ 1 ใหม่
LD (HL), A
OUT 10H,A
OUT 12H,A
RET
LINE DB 01H

```

ส่วนโปรแกรมหน่วงเวลา 10 มิลลิวินาที ซึ่งเราใช้โปรแกรมมวนลูปจะมีลักษณะดัง  
โปรแกรมหน่วงเวลา

```

โปรแกรมส่วนหน่วงเวลา 10 มิลลิวินาที
;DELAY LOOP 10 MILLISECOND
DLY10 PUSH ED: SAVE DE
LD DE,1247 : จำนวนลูป
DEC DE
JR NZ,$-1: ลูปให้วน
POP DE
RET

```

ผลจากโปรแกรม SCA จะบอกสถานะการแสกนว่า แสกนหมาถึงแนวใดซึ่งจะอยู่ในหน่วย  
ความจำ LINE นั่นก็คือ เราจะทราบได้ว่า แกวใดกดใดดูจาก LINE และจะตรวจสอบการกดคีย์  
ว่าเป็นคอลลัมน์ใดได้ โดยทำโปรแกรมรับคีย์เข้าทางพอร์ตอินพุตและเก็บไว้ในรีจิสเตอร์ A ตัว  
อย่างในโปรแกรมรับคีย์และตีบาวนซ์

### 5.3.2.3 โปรแกรมส่วนการรับคีย์และตีบาวนซ์

```

;CHECK THAT KEY WAS PRESSED AND DEBOUNCE IT
KEYIN CALL SCAN
IN A,11H
XOR 0FFH
JR Z,KEYIN
DEBOON LD B,A
CALL DLY10: หน่วงเวลา 10 mS
IN A, 11H: อ่านค่าคอลลัมน์
XOR 0FFH: กลับค่า
CALL Z, DLY10
JR Z,KEYIN+3
CP B: ยังกดตัวเดิมอยู่?
JR NZ,DEBOUN: ใหวนอีกครั้ง

```

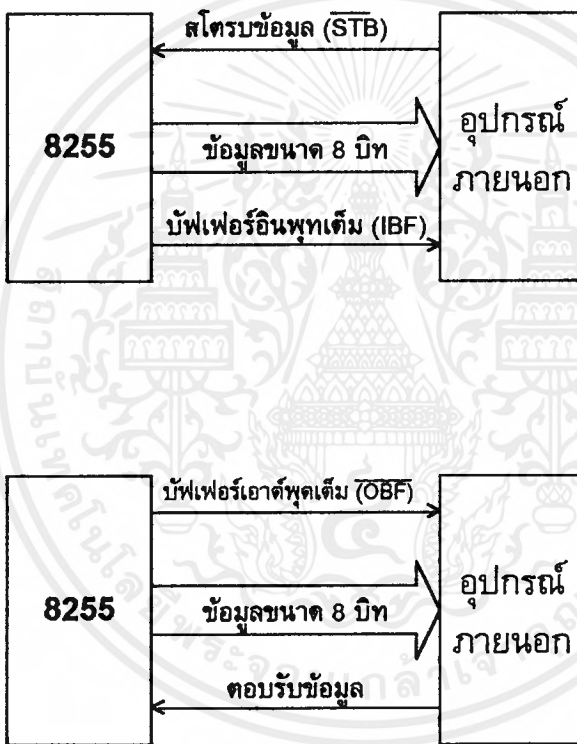
จากโปรแกรมนี้ จะได้ค่าคีย์ตามแนวคอลลัมน์อยู่ในรีจิสเตอร์ A และแกวอยู่ในตัวแปร LINE  
ซึ่งเราจะนำไปใช้ในการถอดรหัสว่าเป็นคีย์อะไรได้ สังเกตว่าเรามีการตรวจสอบเพื่อแก้การกระเด็น  
หรือการตีบาวนซ์ (debounce) คีย์ด้วย และมีการหน่วงเวลา 10 มิลลิวินาทีเพื่อป้องกันการอ่านค่า

เอกรักรณิใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คีย์ผิดพลาดได้ และสิ่งที่สำคัญคือการทำงานของซีพียูเร็วมาก ดังนั้นจึงต้องมีการตรวจสอบการปล่อยคีย์ (release) ก่อนแล้วจึงแสกนหาคอลัมน์และแถวเพื่อกำหนดตำแหน่งคีย์

### 5.3.3 การทำงานของ 8255 ในโหมด 1

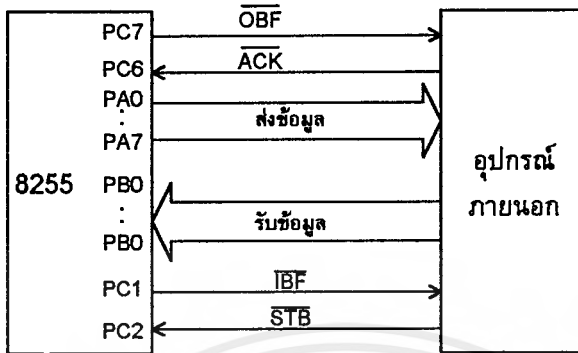
การทำงาน 8255 ในโหมด 1 เป็นโหมดที่ทำให้อินพุตเอาต์พุตมีการตรวจสอบสัญญาณ (handshaking) โดยใช้อินพุตเอาต์พุตของพอร์ต A และพอร์ต B เป็นหลัก และใช้พอร์ต C บนเป็นตัวตรวจสอบสัญญาณ (handshake) ของพอร์ต A ส่วนพอร์ต C ล่างเป็นตัวตรวจสอบสัญญาณของพอร์ต B การจัดสัญญาณต่าง ๆ เหล่านี้แสดงดังรูปแสดงโครงสร้างตัวตรวจสอบสัญญาณของพอร์ตอินพุตและพอร์ตเอาต์พุต ดังที่แสดงในรูป 5.10



รูปที่ 5.10 โครงสร้างตัวตรวจสอบสัญญาณของพอร์ตอินพุตและเอาต์พุต

แนวความคิดของการใช้พอร์ตอินพุตเอาต์พุต โดยมีตัวตรวจสอบสัญญาณก็เพื่อให้มีการชิงโครไนซ์ระหว่างอุปกรณ์ภายนอกที่ทำงานได้ช้ากับการทำงานของคอมพิวเตอร์ที่ทำงานได้เร็ว เช่น เครื่องพิมพ์ทำงานได้ช้า เมื่อคอมพิวเตอร์ส่งตัวอักษรตัวแรกมาพิมพ์ เครื่องพิมพ์รับตัวอักษรและกำลังจะพิมพ์ คอมพิวเตอร์ก็ส่งตัวอักษรครั้งที่ 2 ครั้งที่ 3 ตามมา ทำให้การประมวลผลของอุปกรณ์เครื่องพิมพ์ทำงานไม่ทัน ซึ่งอาจทำให้ข้อมูลสูญหาย ดังนั้นเครื่องพิมพ์จึงส่งสัญญาณบอกคอมพิวเตอร์ว่า “อย่าเพิ่งส่งมาเพราะยังไม่พร้อมที่จะรับ” ลักษณะของการรับส่งข้อมูลอินพุตเอาต์พุต เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ชุด แบบมีการตรวจสอบสัญญาณดังรูป จะใช้  $PA_0$ - $PA_7$  เป็นเอาต์พุต และ  $PB_0$ - $PB_7$  เป็นอินพุต โดยมีพอร์ต C เป็นตัวตรวจสอบสัญญาณ ดังแผนผังในรูปวงจรถ่ายต่อ 8255 ในโหมด 1 ดังที่แสดงในรูป 5.11



รูปที่ 5.11 วงจรถ่ายต่อ 8255 ในโหมด 1

เมื่อโปรแกรม 8255 เป็นโหมด 1 แล้ว ตัว 8255 จะให้พอร์ต C เป็นสัญญาณควบคุม โดยแต่ละบิตของพอร์ต C เป็นไปตามที่กำหนดไว้ ดังตารางด้านล่าง

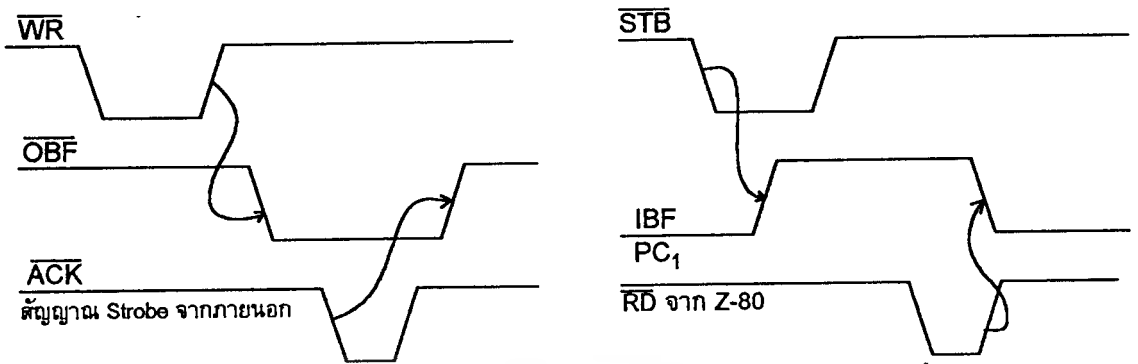
ตารางหน้าที่ของสัญญาณต่าง ๆ ของพอร์ต C ในการทำงานเป็นตัวตรวจสอบสัญญาณเมื่อ 8255 ทำงานในโหมด 1 ดังแสดงในตาราง 5.3

ตารางที่ 5.3 หน้าที่สัญญาณของพอร์ต C

ขา	กรณีอินพุต	กรณีเอาต์พุต
$PC_0$	$INTR_B$	$INTR_B$
$PC_1$	$IBF_B$	$OBF_B$
$PC_2$	$STB_B$	$ACK_B$
$PC_3$	$INTR_A$	$INTR_A$
$PC_4$	STB	I/O
$PC_5$	IBF	I/O
$PC_6$	I/O	$ACK_A$
$PC_7$	I/O	$OBF_B$

โดยปกติ 8255 จะให้สัญญาณอินเทอร์รัพต์ไปบอกซีพียูด้วย สัญญาณอินเทอร์รัพต์ของ 8255 จะเกิดขึ้นที่  $PC_0$  และ  $PC_3$  โดยที่บัพเฟอร์พร้อมแล้วและต้องการให้ซีพียูส่งอินพุตหรือเอาต์พุต มาที่บัพเฟอร์ สัญญาณอินเทอร์รัพต์ก็จะเกิดขึ้น สังเกตว่า สัญญาณอินเทอร์รัพต์เป็นสัญญาณแอกคิ  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พ “1” ซึ่งตรงกับของ 8080 แต่เมื่อใช้กับ Z-80 สัญญาณ INT ของ Z-80 จะรับด้วย “0”



โครงสร้างการตรวจสอบสัญญาณของ 8255 แสดงด้วยสัญญาณทางไฟฟ้าได้ดังรูปที่ 5.12

รูปที่ 5.12 แผนผังเวลาการรับและการส่งข้อมูลโดยใช้ตัวตรวจสอบสัญญาณ

สังเกตว่า การทำงาน 8255 จะเกี่ยวข้องกับสัญญาณ RD และ WR ซึ่งจะทำให้สัญญาณควบคุมเปลี่ยนแปลงไป การตรวจสอบสัญญาณซึ่งกันและกันนี้ เป็นวิธีการรับส่งที่มีประสิทธิภาพ เช่น ในกรณีอินพุต เมื่ออุปกรณ์ภายนอกต้องการส่งข้อมูลให้ ซีพียู ก็ส่งข้อมูลแบบขนานเข้ามา พร้อมทั้งสโตรบ [STB] 8255 ซึ่ง 8255 จะนำข้อมูลนั้นไปเก็บไว้ในรีจิสเตอร์ภายในก่อนแล้วส่งสัญญาณตอบว่า “บัฟเฟอร์ยังเต็มอยู่นะ (IBF) อย่าเพิ่มส่งมาอีก” ครั้งเมื่อซีพียูอ่านข้อมูลจากรีจิสเตอร์ไปแล้ว ส่วนของสัญญาณบัฟเฟอร์อินพุต (IBF) ก็จะบอกว่า “ว่างแล้วส่งมาได้” อุปกรณ์ภายนอกก็จะส่งข้อมูลมาให้อีก

ทำนองเดียวกัน สำหรับพอร์ตเอาต์พุต เมื่อซีพียูส่งข้อมูลออกจากพอร์ตเอาต์พุตให้กับ 8255 ตัว 8255 ก็จะรับไว้ในรีจิสเตอร์ภายใน พร้อมทั้งส่งสัญญาณออกไปบอกอุปกรณ์ภายนอกว่า “เอาต์พุตบัฟเฟอร์ของฉันมีข้อมูลนะ (OBF) มาอ่านเอาไปซิ” อุปกรณ์ภายนอกเมื่อทราบแล้วพร้อม จะอ่านก็จะส่งสัญญาณตอบรับ (ACK) พร้อมกับอ่านข้อมูลไป โดยสัญญาณ ACK จะมีความหมายว่า “ฉันอ่านข้อมูลไปแล้วนะ” ตัว 8255 ก็จะต้องกลับว่า “บัฟเฟอร์ฉันว่างแล้ว เฮอร์ก่อนนะมีข้อมูล ส่งมาให้อีก”

ในการที่จะโปรแกรมโหมด 1 นี้ เราจะใช้รหัสควบคุมเป็น 101 (I/O) 01 (I/O) 0 ในส่วน I/O หมายถึง ถ้าเป็นอินพุตก็คือ “1” ถ้าเป็นเอาต์พุตก็คือ “0” โดย I/O ตัวแรกเป็นของพอร์ต A ที่ 2 เป็นของพอร์ต B เช่น ถ้าต้องการให้พอร์ต A เป็นเอาต์พุต และพอร์ต B เป็นอินพุต เราจะใช้รหัสควบคุมเป็น 10100110 หรือ A6H

จากการพิจารณาการทำงานของซีพียูจะเห็นว่า ทำอย่างไรจึงจะเขียนหรืออ่านพอร์ตได้ ถูกต้องวิธีที่ง่ายวิธีหนึ่งคือ ซีพียูจะคอยตรวจสอบสัญญาณของ 8255 เช่น กรณีเอาต์พุต ซีพียู จะคอยอ่านพอร์ต C แล้วตรวจสอบบิต 7 (OBF) หลังจากทีส่งข้อมูลไปแล้ว ถ้าบิต 7 ยังเป็น “0”

แสดงว่ายังไม่ได้รับการสไตรบแต่ถ้าเป็น “1” แล้วแสดงว่าอุปกรณ์ภายนอกได้รับข้อมูลไปแล้ว สำหรับกรณี อินพุตก็คอยตรวจสอบสัญญาณ IBF ได้เช่นกันว่า มีข้อมูลใหม่เข้ามาหรือยัง คือ ตรวจสอบบิต  $PC_i$  ของพอร์ต C

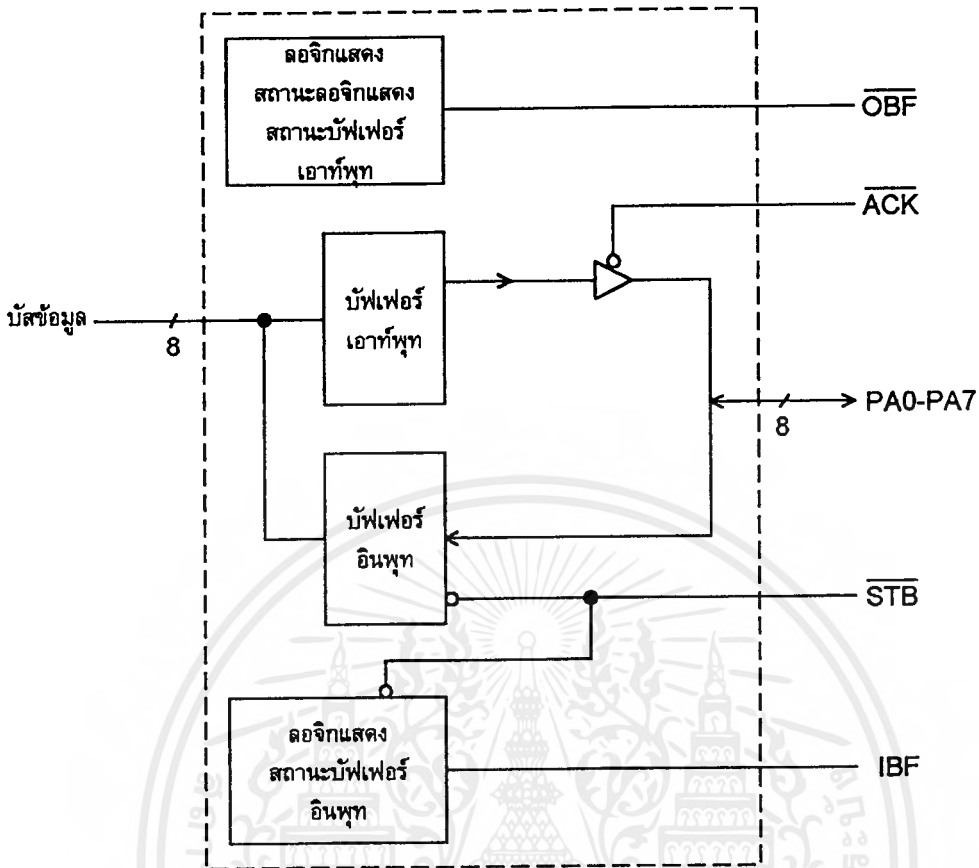
### 5.3.4 การทำงาน 8255 ในโหมด 2

8255 ยังมีโหมดการทำงานอีกโหมดหนึ่งคือโหมด 2 ซึ่งทำได้เฉพาะพอร์ต A ในโหมดนี้ 8255 จะใช้พอร์ต A ทำหน้าที่เป็นพอร์ตแบบ 2 ทิศทางคือสามารถเป็นได้ทั้งพอร์ตอินพุต และ เอาต์พุตโดยโครงสร้างของพอร์ต A ทั้งอินพุตเอาต์พุตมีการตรวจสอบสัญญาณทั้งคู่ ส่วนพอร์ต C จะทำหน้าที่ เป็นสัญญาณตรวจสอบ โดยมีสัญญาณแต่ละขาตั้งตารางที่หน้าที่ของพอร์ต C ในโหมด 2 ดังแสดงในตารางที่ 5.4

ตารางที่ 5.4 แสดงหน้าที่ของพอร์ต C ในโหมด 2

พอร์ต C	ความหมาย
$PC_0$	$INTR_B$
$PC_1$	$IBF_B$
$PC_2$	$STB_B$
$PC_3$	$INTR_A$
$PC_4$	STB
$PC_5$	IBF
$PC_6$	I/O
$PC_7$	I/O

## โครงสร้างของพอร์ต A ที่ทำงานแบบ 2 ทิศทาง แสดงได้ดังรูปที่ 5.13



รูปที่ 5.13 โครงสร้างของพอร์ต A ที่ทำงานแบบพอร์ต 2 ทิศทาง

สังเกตว่าเมื่อโปรแกรมพอร์ต A เป็นโหมด 2 แล้ว พอร์ต B จะต้องโปรแกรมเป็นโหมด 0 หรือโหมด 1 ก็ได้ซึ่งก็ทำงานแบบแยกอิสระอีก ในการใช้งานพอร์ตแบบ 2 ทิศทางนี้ใช้ได้กับงานงาน ประเภท เช่น ใช้ในการรับส่งข้อมูลของพอร์ตมาตรฐานบางประเภทเช่น IEEE 488 หรือใช้เชื่อมโยง ระหว่างคอมพิวเตอร์กับคอมพิวเตอร์ในการรับส่งข้อมูลสลับกันไปและกลับ

### บทสรุป

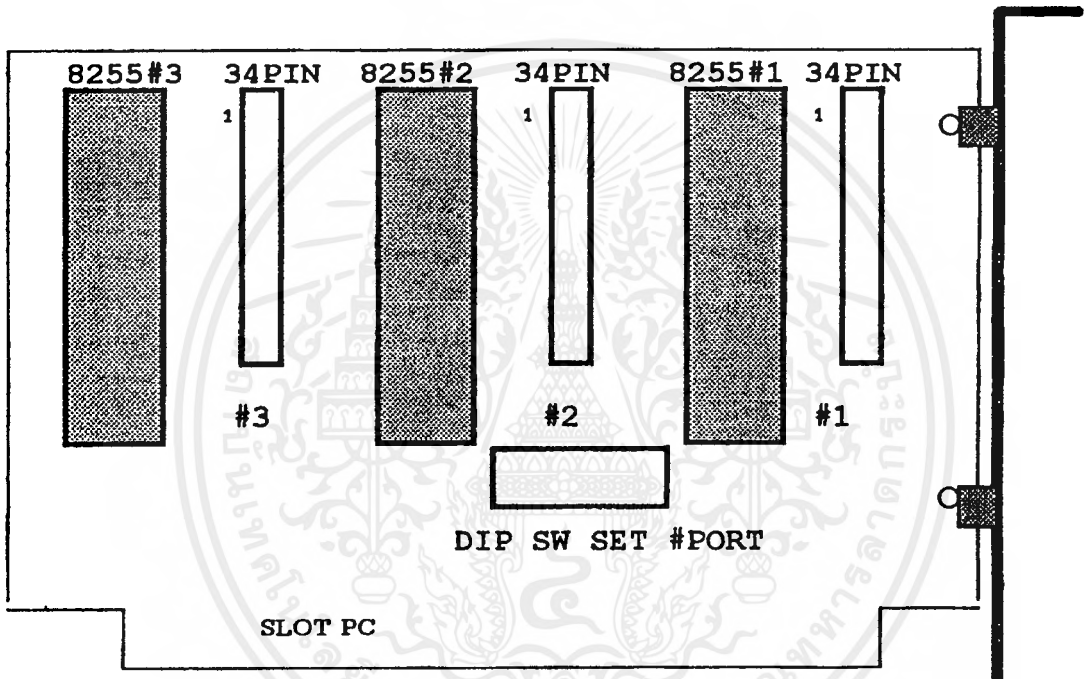
การใช้ 8255 เป็นการที่เราพบเห็นในไมโครโพรเซสเซอร์ทั่วไป เพราะใช้งานได้ง่ายและมีความคล่องตัว ผู้ออกแบบที่ต้องใช้พอร์ตขนานมักเลือกใช้ 8255 เป็นตัวอินเตอร์เฟซกับอุปกรณ์ภายนอก

## บทที่ 6

### การติดตั้ง CARD PORT 8255

#### 6.1 การเชื่อมต่อคอมพิวเตอร์กับฮาร์ดแวร์ควบคุมการทำงานของส่วนต่างๆ

นำ 8255 CARD PORT เสียบที่ Slot ของคอมพิวเตอร์และต่อสายแพ ไปยังวงจรควบคุมการทำงานของคั้งที่แสดงดังรูปที่ 6.1 ข้างล่าง



รูปที่ 6.1 แสดงการติดตั้ง CARD PORT

#### 6.2 การติดตั้ง 8255 CARD PORT

1. ปิด Switch power ของเครื่อง computer ก่อนแล้วเปิดฝาออก
  2. Set DIP SW. ตำแหน่ง PORT ของ CARD ไม่ให้ตรงกับตำแหน่ง Address ของ CARD อื่นๆ
  3. นำ Card ใส้ไปยังเครื่องคอมพิวเตอร์ ทาง Slot PC 62 PIN
  4. ต่อสายแพ 34 PIN ที่จะนำไปใช้งานให้เรียบร้อยก่อน แล้วค่อยเปิดเครื่องคอมพิวเตอร์ใช้งาน
- การกำหนด การทำงานของ 8255 CARD PORT โดยกำหนดการ DECODE PORT ณ ตำแหน่ง ADDRESS ที่ต้องการ ดังที่แสดงในตารางที่ 6.1

### ตารางที่ 6.1 ตาราง I/O ADDRESS MAP

HEX RANGE	USAGE
000-00F	DMA Chip 8237A-5
020-021	Interrupt 8259A
040-043	Timer 8253-5
060-063	PPI 8255A-5
080-083	DMA Page Registers
0AX*	NMI Mask Register
0CX	Reserved
0EX	Reserved
200-20F	Game Control
210-217	Expansion Unit
220-24F	Reserved
278-27F	Reserved
2F0-2F7	Reserved
2F8-2FF	Asynchronous Communications (Secondary)
300-31F	Prototype Card
320-32F	Fixed Disk
378-37F	Printer
380-38C**	SDLC Communications
380-389**	Binary Synchronous Communications
3A0-3A9	Binary Synchronous Communications
3B0-3BF	(Primary)
3C0-3CF	IBM Monochrome Display/ Printer
3D0-3DF	Reserved
3E0-3E7	Color/ Graphics
3F0-3F7	Reserved
3F8-3FF	Diskette
	Asynchronous Communication (Primary)

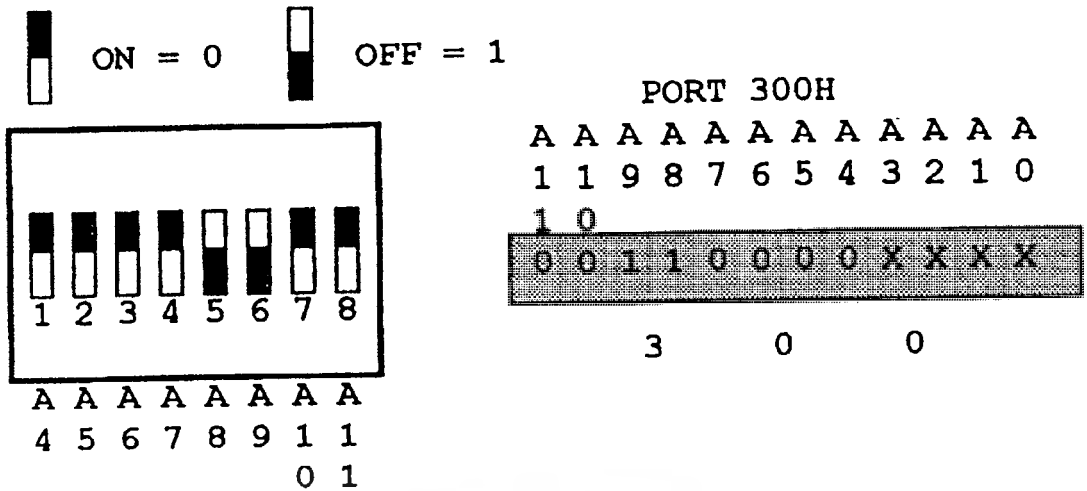
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การ DECODE PORT 8255 บน CARD จะใช้ไอซี TTL 74LS688 , TTL 74LS193 และ DIP SW. 8 PIN เป็นวงจร DECODE เพื่อให้สามารถปรับ SET DIP SW. ตั้งตำแหน่งเบอร์ PORT ของ CARD ได้ โดยในการปรับ DIP SW. นั้นจะต้องไม่ไปตรงกับตำแหน่งของ PORT ของ COMPUTER นั้นด้วย โดย 8255 CARD PORT จะให้ตำแหน่ง PORT 12 PORT ต่อ 1 CARD ดังแสดงในตารางที่ 6.2

ตารางที่ 6.2 การ Decode Port

xx0h	PORT A (1)
xx1h	PORT B (1)
xx2h	PORT C (1)
xx3h	PORT CONTROL (1)
xx4h	PORT A (2)
xx5h	PORT B (2)
xx6h	PORT C (2)
xx7h	PORT CONTROL (2)
xx8h	PORT A (3)
xx9h	PORT A (3)
xxAh	PORT A (3)
xxBh	PORT CONTROL (3)

การตั้งเบอร์ DECODE PORT ได้โดยการปรับ DIP SW. ซึ่งก็มีค่าเท่ากับ ADDRESS นั้นๆ เช่นเราตั้งตำแหน่ง 300 H จะ SET DIP SW. ดังแสดงในรูปที่ 6.2 ข้างล่างนี้



รูปที่ 6.2 แสดงการตั้ง DIP SW.

ตำแหน่ง OUT PUT ขาต่างๆของ 8255 CARD PORT ของ COMPUTER จากรูปข้างบน  
ตำแหน่ง OUTPUT ต่างๆ ของ 8255 CARD PORT เป็นดังนี้

PIN NUMBER 1-8 คือ PA0-PA7

PIN NUMBER 9-16 คือ PB0-PB7

PIN NUMBER 17-24 คือ PC0-PC7

PIN NUMBER 25 คือ Vcc

PIN NUMBER 27 คือ GND.

PIN NUMBER 32 คือ RESET

โดย CARD PORT จะมี SLOT 34 PIN 3 ชุดใน 1 CARD

# บทที่ 7

## โปรแกรมคอมพิวเตอร์

### 7.1 ส่วนประกอบของโปรแกรม

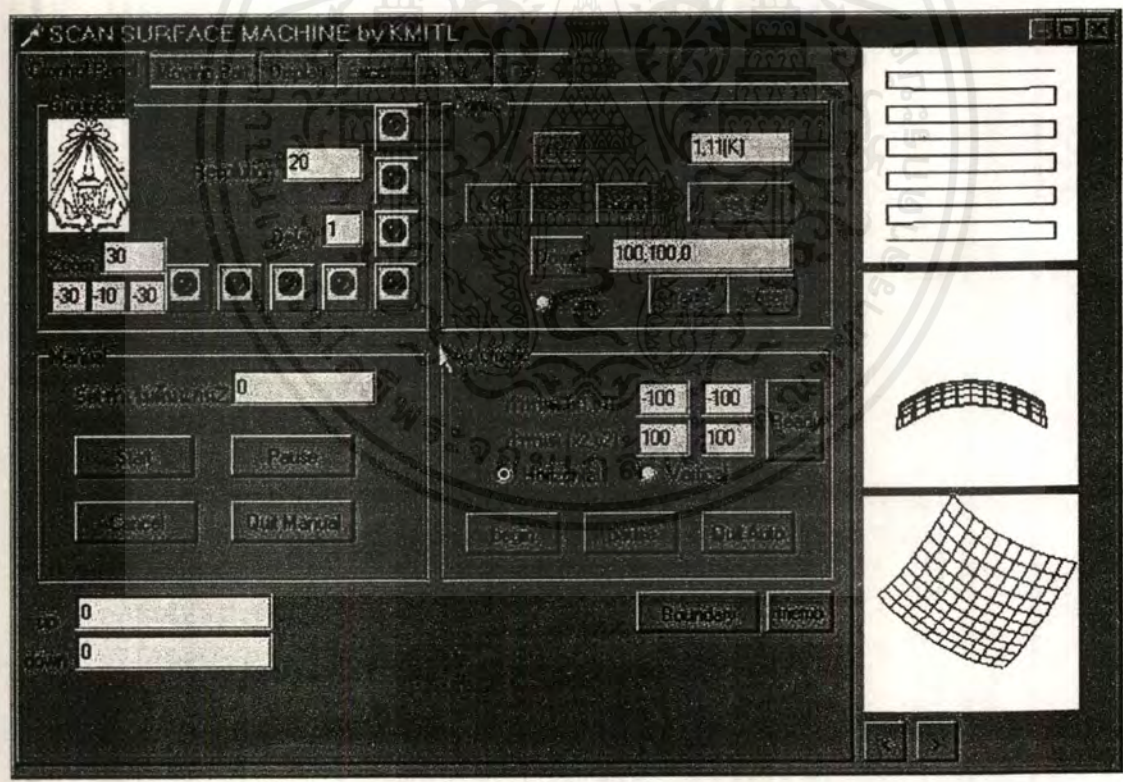
โปรแกรมที่เขียนด้วย Visual จะแบ่งส่วนสำคัญออกเป็น 2 ส่วน คือ

- 1. Form
- 2. Source code

7.1.1 Form เป็นวินโดว์อันหนึ่งมีคุณสมบัติเหมือนวินโดว์ทั่วไป คือ ประกอบด้วยคอนโทรลเมนูปุ่มมิมิเมาส์ ปุ่มเมกซ์ไมซ์ ไคเดิลบาร์ และขอบหน้าต่างที่สามารถปรับขนาดได้

Form ในโปรเจกนี้มีชื่อว่า Tcumpeerawat มีหน้าควบคุมอยู่ 6 หน้าดังนี้

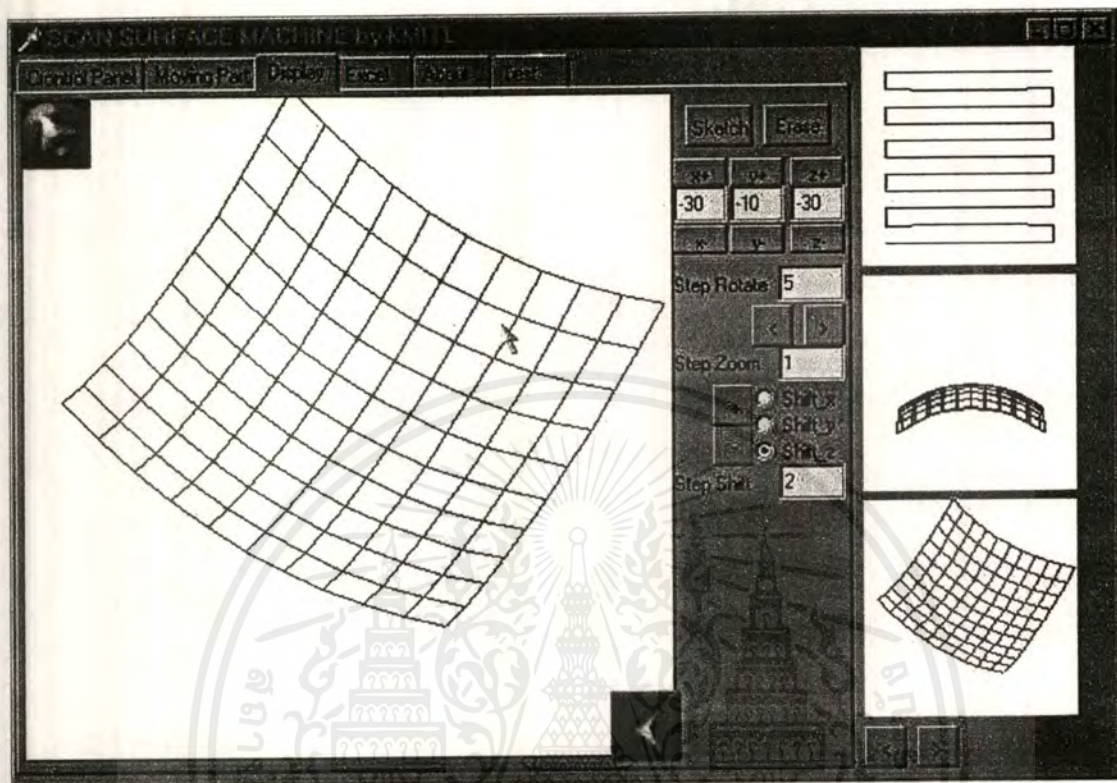
#### 7.1.1.1 Control Panel ดังรูปที่ 7.1



รูปที่ 7.1 Control Panel ของโปรแกรม

## 7.1.1.2 Moving Part

## 7.1.1.3 Display ค้างรูปที่ 7.2



รูปที่ 7.2 แสดงผลของโปรแกรม

## 7.1.1.4 Excel

## 7.1.1.5 About

## 7.1.1.6 Test

ซึ่งแต่ละหน้าจะมีปุ่มควบคุมแตกต่างกันออกเป็น เช่น Leftclick , Rightclick , Upclick , Downclick , Begin , start ซึ่งจะแสดงค้างรูปข้างต้น

7.1.2 Source code เป็นตัวโปรแกรมหลักที่ใช้ควบคุมการทำงานต่างๆโดยภาษาที่ใช้ในโปรแกรม Delphi คือภาษา Pascal โครงสร้างของ code-editor เป็นดังนี้

```
unit Unit1; Unit ชื่อของUnit
```

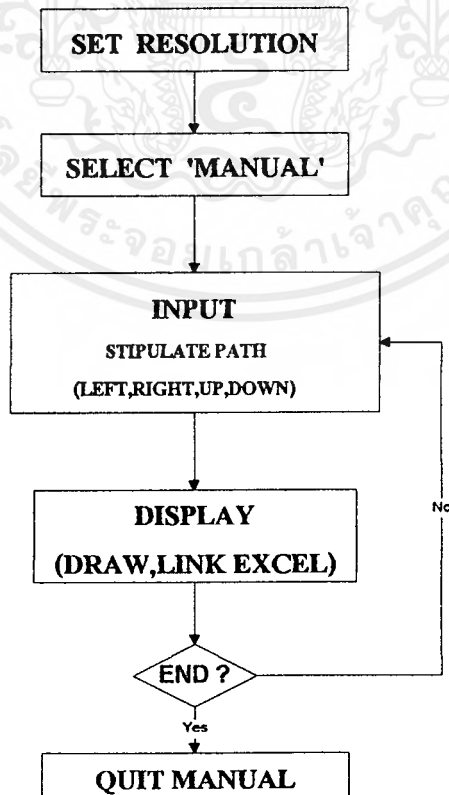
```
interface
```

```
uses
```

```
Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs;
```

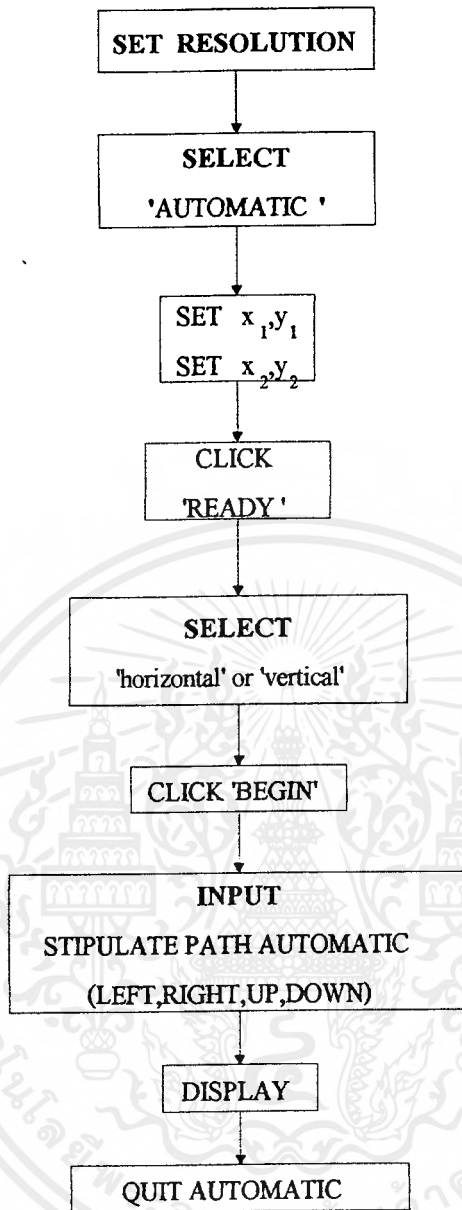
เอกสารนี้เป็นเอกสารลิขสิทธิ์ของสำนักงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

type  
 TForm1 = class(TForm) ชื่อของ form = class(tform) และสามารถประกาศชนิดของตัวแปรขึ้นมาใหม่ได้  
 private  
 { Private declarations } การประกาศ Procedure หรือ Function ที่ใช้ได้เฉพาะ unit นี้เท่านั้น  
 public  
 { Public declarations } การประกาศ Procedure หรือ Function ที่สามารถเรียกใช้ได้จาก unit อื่นได้  
 end;  
 var  
 Form1: TForm1; ชื่อของ form: TForm1, และตัวแปรอื่น  
 Implementation หลังจาก Implementation จนถึง end. เป็นตัวโปรแกรมที่ควบคุมการทำงาน  
 {SR \*.DFM} เป็นการ link กับ Form ห้ามลบเด็ดขาด  
 เป็นการควบคุมที่มี procedure และ function ควบคุมการทำงานต่างๆ  
 end.  
 Flow chart แสดงการทำงานของโปรแกรมดังรูปที่ 7.3 และ รูปที่ 7.4



รูปที่ 7.3 แสดงขั้นตอนการทำงานของโปรแกรม ในโหมด Manual

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.4 แสดงขั้นตอนการทำงานของโปรแกรม ในโหมด Automatic

## 7.2 ขั้นตอนการใช้งาน

เริ่มต้นต้อง set หาจุดอ้างอิงของเครื่อง โดยการหาจุดศูนย์กลาง ของเครื่อง ทำได้โดยใช้ Tabsheet หน้า test (หรือจะใช้มือหมุนแกนเลขก็ได้) ต่อจากนั้นทำการกำหนด resolution (ความกว้างแต่ละช่อง) ว่าจะเอาขนาดเท่าไร ยกตัวอย่างเช่น ต้องการ scan พื้นที่ขนาด 100mm. x 100mm. โดย resolution = 10mm. โปรแกรมก็จะทำการแบ่งพื้นที่ 100mm. x 100mm. ออกเป็นช่องๆ ขนาด 10mm. x 10mm. เมื่อกำหนด resolution เรียบร้อย ให้กดปุ่ม set position ที่ Tabsheet หน้า Control Panel จากนั้นการวัดจะแบ่งเป็นสองแบบ คือ Automatic กับ Manual

Automatic จะเป็นการ scan พื้นที่สี่เหลี่ยมโดยอัตโนมัติ โดยการกำหนดจุด 2 จุด ซึ่งเป็น เส้น

เอกส... ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทแยงมุมของสี่เหลี่ยมที่จะทำการวัด การใช้งานนั้นเมื่อกำหนดจุด  $X_p, Y_1$  แล้วคลิกปุ่ม Ready เครื่องจะเดินไปยังจุดดังกล่าวโดยยังไม่มีกรเก็บค่า  $z$  หลังจากนั้นก็เลือกที่จะเดินเป็นแบบแนวนอนหรือแนวตั้ง จากนั้นก็กำหนด  $X_{xx}, Y_2$  (ปลายอีกจุดหนึ่งของเส้นทแยงมุม) กด begin เพื่อเริ่มทำการ scan

Manual จะเป็นการ scan โดยกำหนดเส้นทางเดินเอง เลือกทิศทางทีละ step 1 step ก็จะได้เดินเท่ากับความละเอียดของ resolution เมื่อต้องการเริ่มวัดที่ตำแหน่งใด ให้คลิกปุ่ม start ณ ตำแหน่งที่จะทำการเริ่มวัด เมื่อต้องการจะ undo ให้กด cancel เป็นการถอยหลังทิศทางที่เราเดิน 1 ครั้ง (สามารถย้อนหลัง undo ได้ 25 ครั้ง)

ส่วนแสดงผล เนื่องจาก project นี้ถึงเห็นถึงการนำข้อมูลไปประยุกต์ใช้ จึงได้นำข้อมูลซึ่งเป็นตัวเลข (Coordinate) โดยจะแสดงใน Excel วิธีการใช้นั้น ก่อนจะรัน project.exe ต้องทำการสร้าง file C:\Mydocument\test2.excel เปิดมันขึ้นมาแล้วรัน project เมื่อคลิก set position โปรแกรมจะทำการเตรียมพื้นที่ (ลบข้อมูลเก่าสำหรับความละเอียดต่างๆ แต่ถ้าสูงจะมีจำนวนจุดที่ต้องเก็บค่ามาก ถ้าทำการ clear ค่า เครื่องอาจแสงก็ได้ ดังนั้นจะต้องลบข้อมูลเก่าออก ก่อนที่จะทำการ scan) เมื่อเริ่ม scan ค่าจะถูกส่งไปยัง Excel ซึ่งในแต่ละเซลล์ จะบอก coordinate ว่าอยู่ที่พิกัดใด เช่น (0,0,0) (20,26,8.56) เป็นต้น หากผู้ใช้ต้องการที่จะใช้ประยุกต์ร่วมกับ AutoCAD ให้ทำตามดังนี้ ( เฉพาะพื้นที่ที่เป็นสี่เหลี่ยม ) ทำการ copy ผลที่เก็บตำแหน่ง coordinate จากนั้นเปิด AutoCAD พิมพ์คำสั่ง Mesh ใน command < > และกำหนด m ( จำนวนจุดในแนวตั้ง ), n ( จำนวนจุดในแนวนอน ) ต่อไปให้ Click ขวาที่บรรทัดคำสั่ง จะมีคำสั่ง Paste ให้เลือก ทำการเลือก Paste จากนั้นจุดแต่ละจุดจะถูก transfer และก็จะถูกนำไปวาดเป็น surface ที่สวยงามต่อไป โดยหลักการนี้นำไปประยุกต์ใช้กับคำสั่งอื่นได้อีกเช่น line และสามารถปรับมุมมองได้ตามความสะดวก

### 7.3 หลักการทำงานโดยสังเขป

สามารถแบ่งออกเป็น 2 ส่วนใหญ่ คือ ส่วนการเคลื่อนที่ของ Table และ ส่วนแสดงผล การเคลื่อนที่ของ Table ยังแบ่งออกเป็น

การเคลื่อนที่ที่ Procedure ใช้ได้แก่ ort, inport, left\_ta, right\_ta, up\_ta, down\_ta, light, delay, boundary ส่วนแสดงผลของการเคลื่อนที่ เป็นการบอกตำแหน่งว่าอยู่ที่จุดใด (แต่เนื่องจากการลักษณะการควบคุมเป็นแบบ Open Loop จึงเป็นเมื่อส่งค่าออกไปแล้วอาจจะเกิดข้อผิดพลาดได้) Procedure ที่ใช้ได้แก่ display\_m display\_high (ทำการส่งค่าข้อมูลไปยัง excel ด้วย)

ส่วนแสดงผล แบ่งออกเป็น

ส่วนแสดงผลตามทางเดิน Procedure ที่ใช้ได้แก่ drawing\_expert

ส่วนแสดงผลตามข้อมูลที่เก็บได้ คือเมื่อครบจุดที่สามารถวาดเป็นตารางได้ก็จะวาดทันที Procedure ที่ใช้ได้แก่ drawing\_excellent

ส่วนช่วยการแสดงผล เช่นการหมุนรูป การZoom Procedure ที่ใช้ได้แก่ rotate\_image , zooming\_image

เมื่อรันโปรแกรมเริ่มต้นโปรแกรมจะทำการ set ค่าต่าง(procedure TCumpeerawut.FormCreate (Sender: TObject))เช่น Set port control (outport (\$303,128) outport (\$300,153) ) และ ab ซึ่งเป็น Arrayที่เก็บค่าของข้อมูลที่จะส่งไปยัง Step เช่น ถ้าส่งแบบ 2phase ab[0]:=5;ab[1]:=20;ab [2]:=80;ab[3]:=65; และเริ่มset excel มีa,b เป็นตัวแปรที่ใช้ในการระบุตำแหน่งของเซลล์ที่จะทำการ ส่งค่า i,j เป็นตัวแปรระบุตำแหน่งของ Table ที่หารด้วย resolution เมื่อทำการ set position แล้วกด left,right,up,down จะมีผลให้a,b,i,j เปลี่ยนไป ในส่วนของAutomaticก็จะมีเงื่อนไขในการเดินซึ่ง เมื่อผ่านเงื่อนไขต่างๆก็จะทำการกด left,right, down,up เมื่อทำการstart จะทำการเก็บค่าco- ordinateลงในmem\_x, mem\_y, mem\_z ซึ่งเป็นArray 2 มิติโดยใช้ a,b ในการเก็บเมื่อนำไปวาดก็จะ มีอยู่2แบบคงที่เคยได้กล่าวไว้แล้วคือวาดตามทางเดิน กับ วาดตามข้อมูลที่เก็บได้ และยังมี procedure ช่วยแสดงผลเช่นการหมุนรูปภาพ การZoom การ Shift

Procedure boundary เป็นการพัฒนาโปรแกรมให้สามารถเดินภายในขอบเขตที่กำหนด การใช้งานทำโดยการกด start (Manual)แล้วเดินเป็นBoundary เมื่อเดินครบเป็นขอบเขตปิดแล้วเครื่องจะ ทำการเดินAutomatic หลักการคือจดจำว่าขอบเขตในแต่ละแฉกมีค่าเท่าไร จากนั้นก็เดินให้ครบ เมื่อครบแล้วก็จะทำการ Ready ในปลายของแฉกถัดไปโดยเลือกปลายที่ใกล้ๆที่อยู่ปัจจุบันมากที่สุด เนื่องจาก Procedure ยังพัฒนาไม่สมบูรณ์อย่างเต็มที่อาจจะยังมีข้อผิดพลาดอยู่บ้าง

Test เป็นอีกส่วนหนึ่งในการควบคุมทดสอบมีการใช้งานนั้นก็ทำโดยการกดปุ่ม set test จากนั้นก็ทำการTest

#### 7.4 โปรแกรมนี้มี Procedure หรือ Function ที่สำคัญได้แก่

1. Procedure Tcumpeerawut.outport (portno: smallint; data: byte);

มีหน้าที่ส่งข้อมูลออกไปพอร์ตต่างๆ

วิธีใช้คือ Output ( เบอร์พอร์ตที่จะส่ง, ข้อมูลที่จะส่ง)

2. Function Tcumpeerawut.inport (portno: smallint):byte

มีหน้าที่รับข้อมูลเข้าจากพอร์ต

วิธีใช้คือ Input ( เบอร์พอร์ตที่ต้องการ)

3. Procedure tcumpeerawut.delay\_table;

มีหน้าที่ หน่วงเวลา การส่งข้อมูล เพื่อให้ step motor ขับ ได้อย่างต่อเนื่อง (ถ้าไม่หน่วงเวลาแล้ว

จะส่งข้อมูลเร็วมากจน step motor ไม่หมุน

วิธีใช้คือ delay;

4. Procedure Tcumpeerawut.left ta(turn,delay:smallint);

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับบริการเชิงงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มีหน้าที่ เดิน ไปทางซ้าย 1 step (step resolution = ค่าที่set ความละเอียดคอนเริ่มคั้น ) และรับค่าผลต่าง pulse

วิธีใช้ left\_ta( จำนวน step ของมอเตอร์ ที่จะหมุน , delay)

5. procedure Tcumpeerawut.right\_ta(turn,delay:smallint);

มีหน้าที่ เดิน ไปทางขวา 1 step (step resolution) และรับค่าผลต่าง pulse

วิธีใช้ right\_ta( จำนวน step ของมอเตอร์ ที่จะหมุน , delay)

6. procedure Tcumpeerawut.up\_ta(turn,delay:smallint);

มีหน้าที่ เดินขึ้น 1 step (step resolution) และรับค่าผลต่าง pulse

วิธีใช้ up\_ta(จำนวน step ของมอเตอร์ ที่จะหมุน , delay)

7. procedure Tcumpeerawut.down\_ta(turn,delay:smallint);

มีหน้าที่ เดินลง 1 step (step resolution) และรับค่าผลต่าง pulse

วิธีใช้ down\_ta(จำนวน step ของมอเตอร์ ที่จะหมุน , delay)

8. procedure tcumpeerawut.light;

มีหน้าที่ แสดงผลว่าข้อมูลถูกส่งถ่ายแล้วหรือยัง ถ้าส่งถ่ายแล้ว ไฟจะติด ถ้ายังไม่เกิดการส่งถ่ายไฟจะยังไม่ติด

วิธีใช้ light;

9. procedure TCumpeerawut.display\_m(direc:atad);

มีหน้าที่เป็นการแสดงผลในหน้าจอ control pannel เป็น co-ordinate เช่น (0,0) (20,0)

วิธีใช้ display\_m ('ทางเดิน') เช่น display\_m('left') จะส่งถ่ายข้อมูลไปยัง excel โดยระบุตำแหน่งเซลล์ไว้ที่หน้า control pannel

10. procedure tcumpeerawut.display\_high(high:extended);

มีหน้าที่แสดงผล โดยมีความสูงเป็น co ordinate เช่น (0,95)

วิธีใช้ Display\_high (' high')

11. procedure Tcumpeerawut.mem\_zaxis;

มีหน้าที่ เก็บค่าต่างๆทั้งหมด 3 แกนลง array

12. procedure TCumpeerawut.FormCreate(Sender: TObject);

มีหน้าที่ เมื่อเริ่มสร้างทำการ set port control กำหนด outport (\$303,128) outport (\$300,153)

และทำการ set ค่าต่างๆ

13. procedure TCumpeerawut.set\_positionClick(Sender: TObject);

มีหน้าที่ ในการ set และ clear ค่าต่างๆ

14. procedure TCumpeerawut.set\_zClick(Sender: TObject);

มีหน้าที่ ในการ set แกน z

15. procedure TCumpeerawut.start\_Click(Sender: TObject);  
มีหน้าที่ ในการเริ่ม start ของ manual
16. procedure TCumpeerawut.quit\_manualClick(Sender: TObject);  
มีหน้าที่ในการหยุดการ scan
17. procedure TCumpeerawut.cal\_ml;  
มีหน้าที่ในการจดจำทางเดินเก่าคล้ายกับ undo
18. procedure TCumpeerawut.PauseClick(Sender: TObject);  
มีหน้าที่ในการหยุดโปรแกรมชั่วคราว โดยจะใช้กับ excel
19. procedure TCumpeerawut.leftClick(Sender: TObject);  
มีหน้าที่ เมื่อคลิกปุ่ม left แล้วจะทำให้ table เคลื่อนที่ไปทางแกนซ้าย
20. procedure TCumpeerawut.rightClick(Sender: TObject);  
มีหน้าที่ เมื่อคลิกปุ่ม right แล้วจะทำให้ table เคลื่อนที่ไปทางแกนซ้าย
21. procedure TCumpeerawut.upClick(Sender: TObject);  
มีหน้าที่ เมื่อคลิกที่ปุ่มนี้แล้วจะทำให้ table เคลื่อนที่ขึ้น
22. procedure TCumpeerawut.downClick(Sender: TObject);  
มีหน้าที่ เมื่อคลิกที่ปุ่มนี้แล้วจะทำให้ table เคลื่อนที่ขึ้น
23. procedure TCumpeerawut.readyClick(Sender: TObject);  
มีหน้าที่ เมื่อกด ready เพื่อเป็นการเดิน ไปยังตำแหน่ง X1, Y1
24. procedure TCumpeerawut.begin\_Click(Sender: TObject);  
มีหน้าที่ เมื่อกดปุ่ม begin จะทำการเดิน โดยจะเดินทางแนวนอนหรือแนวตั้ง ก็กำหนดเลือก จากนั้นจะเดินครอบคลุมพื้นที่สี่เหลี่ยมที่มีเส้นทแยงมุมที่จุด X2,Y2
25. procedure tcumpeerawut.rotation(p1:\_3dpoint);  
มีหน้าที่ เป็นการกำหนดมุม 3 มุมที่จะหมุน แล้วทำการแปลงค่าให้เป็นตำแหน่งใหม่
26. procedure tcumpeerawut.trans(p1:\_3dpoint;var p2:\_3dpoint);  
มีหน้าที่ เป็นการเลื่อนทั้ง 3 แกน เหมือนการ Shift แกนเข้าหรือออก
27. procedure tcumpeerawut.purse(p1:\_3dpoint;ggg1,ggg2,ggg3,ggg4:integer;var p2:\_2dpoint);  
มีหน้าที่ เป็นการแปลงค่า 3 มิติ แสดงผลออกเป็น 2 มิติ ปรากฏหน้าจอ
28. procedure tcumpeerawut.select\_image(cm:smallint;f:\_2dpoint);  
มีหน้าที่ เป็นการเลือกว่าจะวาดรูปใด โดยการ select\_image(เลือกว่าจะวาดรูปใด)
29. procedure tcumpeerawut.drawing\_expert(dd1,dd2,dd3,dd4,rox,roy,roz,cm:smallint);  
มีหน้าที่ เป็นการวาดตามทางเดิน

วิธีใช้ dd1,dd2,dd3,dd4 เป็นการกำหนดขนาดของ image

rox, roy, roz เป็นการกำหนดมุมที่จะวาด

cm เป็นการเลือกรูปที่จะวาด

30. procedure tcumpeerawut.draw;

มีหน้าที่ เป็นการวาดภาพตามทางเดินแล้วแสดงผล 2 รูป

31. procedure tcumpeerawut.drawing\_excellent(dd1,dd2,dd3,dd4,rox,roy,roz,cm:smallint);

มีหน้าที่ เป็น Procedure ที่วาดทันทีที่ครบ จุดที่สามารถจะวาดเป็นตารางได้

32. procedure TCumpeerawut.draw\_hahaClick(Sender: TObject);

มีหน้าที่ ทำการวาดรูปหน้า display

33. procedure TCumpeerawut.traceClick(Sender: TObject);

มีหน้าที่ ทำการวาดรูป โดยวาด top view และ isometric

34. procedure TCumpeerawut.eraserClick(Sender: TObject);

มีหน้าที่ ทำการลบรูปเดิม

35. procedure tcumpeerawut.rotate\_image(ro\_sx,ro\_sy,ro\_sz:smallint;add:boolean);

มีหน้าที่ ทำการหมุนรูป

36. procedure TCumpeerawut.x\_upClick(Sender: TObject);

มีหน้าที่ หมุนตามแกน x ทิศ +

37. procedure TCumpeerawut.x\_downClick(Sender: TObject);

มีหน้าที่ หมุนตามแกน x ทิศ -

38. procedure TCumpeerawut.y\_upClick(Sender: TObject);

มีหน้าที่ หมุนตามแกน y ทิศ +

39. procedure TCumpeerawut.y\_downClick(Sender: TObject);

มีหน้าที่ หมุนตามแกน y ทิศ -

40. procedure TCumpeerawut.z\_upClick(Sender: TObject);

มีหน้าที่ หมุนตามแกน z ทิศ +

41. procedure TCumpeerawut.z\_downClick(Sender: TObject);

มีหน้าที่ หมุนตามแกน z ทิศ -

### หมายเหตุ

1. ในโปรเจกต์นี้ใช้คอมไพเลอร์เพิ่มเติมคือ Jirayu ซึ่งนำมาจากหนังสือ Borland Delphi 3 วิธีการลงหาอ่านได้ในหนังสือ Amazing

2. รายละเอียดของ Graphic ในการวาดรูป สามมิติศึกษาเพิ่มเติมได้ที่หนังสือ แนะนำ Pascal โดย Turbo Pascal

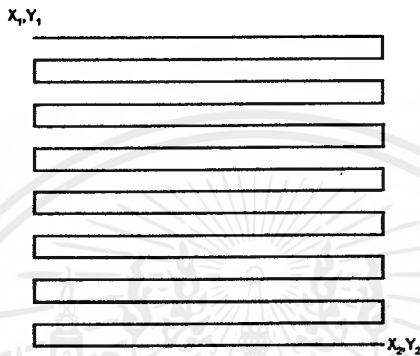
3. การหมุนแกนถ้ากำหนดมุมที่หมุนให้เป็น 0 ทั้งหมด จะได้แกนตั้งเป็นแกนของ X, แกนนอนเป็นเอกสารถึงเอกสารถึงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แกนของ Y, และแกน Z เป็นแกนที่พุ่งออกมาจากหน้าจอ

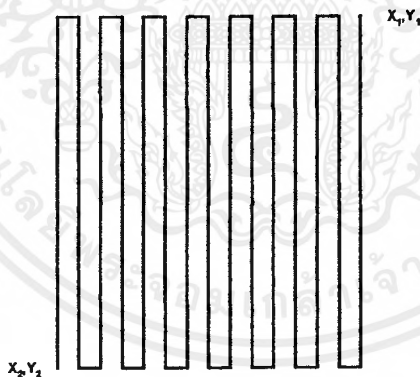
4. การควบคุมสามารถใช้ Keyboard โดยเลื่อน cursor ไปที่ส่วนแสดงผล co-ordinate จากนั้นก็สามารถทำการควบคุมด้วย Keyboard ได้

5. มี check box ชื่อ active ถ้าหากว่า check ส่วนแสดงผลจะทำการวาดทันทีที่ข้อมูลเพียงพอ (ในกรณี ที่ความละเอียดสูงอาจทำให้เครื่องช้าได้)

6. เส้นทางการเดินของ Mode Automatic แสดงดังรูปที่ 7.5 และ รูปที่ 7.6



รูปที่ 7.5 ภาพการเดิน Automatic ตามแนวแกนนอน



รูปที่ 7.6 ภาพการเดิน Automatic ตามแนวแกนตั้ง

## บทที่ 8

### ผลการทดลอง

จากที่ได้ทำการทดลอง Scan พื้นผิว ทดลองซึ่งเป็นพื้นผิวส่วนหนึ่งของแม่พิมพ์ ขึ้นรูปซ้อน ซึ่งพื้นผิวมี Slope ไม่มากจนเกินไป และมีลักษณะเรียบและต่อเนื่อง ผลการทดลองจะแสดงค่า Co ordinate ทั้ง 3 แกน x, y, z และทำการแสดงผลโดยใช้ Program AutoCAD แสดงให้เห็นดังรูปที่ 8.1



รูปที่ 8.1 รูปที่แสดงบน AutoCAD

#### Mesh

at point, X= 4.0000 Y= 0.0000 Z= 1.5000
at point, X= 8.0000 Y= 0.0000 Z= 3.1250
at point, X= 12.0000 Y= 0.0000 Z= 4.1250
at point, X= 16.0000 Y= 0.0000 Z= 4.6250
at point, X= 16.0000 Y= 0.0000 Z= 4.6250

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

at point, X= 20.0000 Y= 0.0000 Z= 4.6250
at point, X= 24.0000 Y= 0.0000 Z= 4.5000
at point, X= 28.0000 Y= 0.0000 Z= 4.0000
at point, X= 32.0000 Y= 0.0000 Z= 3.3750
at point, X= 36.0000 Y= 0.0000 Z= 2.3750
at point, X= 40.0000 Y= 0.0000 Z= 1.2500
at point, X= 44.0000 Y= 0.0000 Z= 0.0000
at point, X= 48.0000 Y= 0.0000 Z= -1.6250
at point, X= 52.0000 Y= 0.0000 Z= -3.3750
at point, X= 56.0000 Y= 0.0000 Z= -5.5000
at point, X= 4.0000 Y= -4.0000 Z= 1.8750
at point, X= 8.0000 Y= -4.0000 Z= 3.1250
at point, X= 12.0000 Y= -4.0000 Z= 4.0000
at point, X= 16.0000 Y= -4.0000 Z= 4.3750
at point, X= 20.0000 Y= -4.0000 Z= 4.5000
at point, X= 24.0000 Y= -4.0000 Z= 4.3750
at point, X= 28.0000 Y= -4.0000 Z= 3.8750
at point, X= 32.0000 Y= -4.0000 Z= 3.1250
at point, X= 36.0000 Y= -4.0000 Z= 2.1250
at point, X= 40.0000 Y= -4.0000 Z= 1.1250
at point, X= 44.0000 Y= -4.0000 Z= -0.2500
at point, X= 48.0000 Y= -4.0000 Z= -1.8750
at point, X= 52.0000 Y= -4.0000 Z= -3.6250
at point, X= 56.0000 Y= -4.0000 Z= -5.7500
at point, X= 4.0000 Y= -8.0000 Z= 1.7500
at point, X= 8.0000 Y= -8.0000 Z= 3.3750
at point, X= 12.0000 Y= -8.0000 Z= 4.1250
at point, X= 16.0000 Y= -8.0000 Z= 4.1250
at point, X= 20.0000 Y= -8.0000 Z= 3.8750

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

at point, X= 24.0000 Y= -8.0000 Z= 3.5000

at point, X= 28.0000 Y= -8.0000 Z= 3.0000

at point, X= 32.0000 Y= -8.0000 Z= 2.2500

at point, X= 36.0000 Y= -8.0000 Z= 1.2500

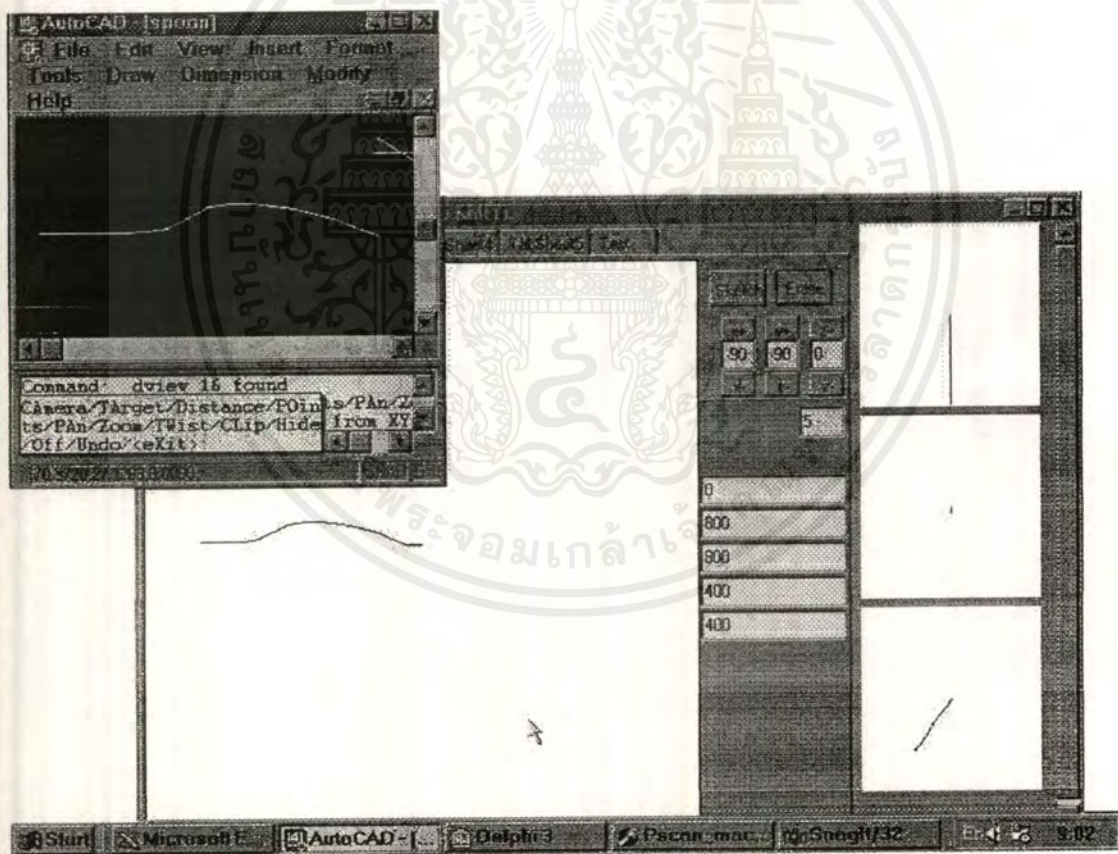
at point, X= 40.0000 Y= -8.0000 Z= 0.0000

at point, X= 44.0000 Y= -8.0000 Z= -1.3750

at point, X= 48.0000 Y= -8.0000 Z= -3.0000

at point, X= 52.0000 Y= -8.0000 Z= -4.7500

at point, X= 56.0000 Y= -8.0000 Z= -5.8750



รูปที่ 8.2 แสดงตัวอย่างผลการทดลองเพิ่มเติม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## บทที่ 9

### สรุปและแนวทาง

#### สรุป

1. การควบคุมทำได้ทั้ง manual และ automatic โดยที่ manual สามารถเลือกได้หลายทางเดิน ส่วน automatic เป็นทางเดินที่กำหนดตายตัวว่า จากซ้ายไปขวา จากขวาไปซ้ายสลับไปสลับมา เดินจนครบพื้นที่กำหนด ดังนั้นข้อดีของ manual คือสามารถเลือกทางเดินได้ ตามความต้องการของผู้ใช้

จากหลักการของ Project นี้จะอ้างอิงค่าทางแนวแกน z จากจุดเดิม ดังนั้นเส้นทางเดินจึงมีความสำคัญอย่างยิ่งขวดในการสแกน เพื่อให้ได้เส้นทางเดินที่เหมาะสมกับชิ้นงาน

-ข้อดีของระบบ Manual คือ สามารถกำหนดเส้นทางเดินเอง ได้ตามใจชอบ ทำให้สามารถเลือกเส้นทางเดินที่เหมาะสมในแต่ละชิ้นงาน แต่ทว่าทำให้สิ้นเปลืองเวลาในการวัด ในการกำหนดจุด

-ข้อดีของระบบ Automatic คือ รวดเร็วและสะดวกในการใช้ แต่ว่าอาจจะให้เกิด ข้อผิดพลาดมาก เนื่องจากเส้นทางเดินไม่เหมาะสม

2. ทางด้าน Mechanic ของหัว Probe มีข้อจำกัดในบางอย่างที่ ไม่ดี ทำให้เกิด error ในการวัด เนื่องจากการเก็บค่า z นั้น ได้จากการอ้างอิงจากจุดเดิมไปเรื่อยๆ ดังนั้นถ้าหากว่ามีการวัดผิดพลาดบ่อยๆ ในแต่ละจุดก็จะทำให้ error ทวีคูณขึ้นไปเป็นจำนวนเท่าของ error ที่เกิดขึ้น
3. การเดินของ Stepping Motors ค่อนข้างที่จะเชื่อถือได้ เช่น ถ้าสั่งให้เดินไป 10 mm. ก็จะได้เดินได้ใกล้เคียงมาก สาเหตุที่การเคลื่อนที่ไม่ได้ 10 mm. เพราะว่าเป็นการควบคุมแบบ open loop ไม่มีการป้อนกลับมาจึง จึงทำให้การเคลื่อนที่ไม่ได้ระยะทางเท่าที่กำหนด สาเหตุอีกประการก็คือ การเชื่อมต่อระหว่างเพลากับมอเตอร์ ทำให้ต้องใช้ตัวเชื่อมต่อที่เรียกว่า Coupling แต่ว่าเมื่อใช้ไปนานๆ นอตที่ใช้ยึด Coupling กับ เพลาของมอเตอร์จะคลาย เนื่องจากการสั่น ทำให้ Stepping motors หมุน free โดยที่เพลาก็ไม่ได้เคลื่อนที่ตาม แก๊วโดยขันนอตให้แน่นหรือเช็คทุกครั้งทีวัด
4. การแสดงผลทำได้ค่อนข้างสมบูรณ์ เพราะในส่วน Display จะมี การหมุนรูป (rotate ได้ทุกแกน) , การ zoom และ การ shift ไปในแต่ละแกน หรือยังสามารถส่งถ่ายไปยัง Excel ได้ แล้วทำการ copy และ paste แสดงผลโดย AutoCAD ได้อีกด้วย
5. การเคลื่อนที่ในแนวแกน z นั้นอาศัยให้น้ำหนักตกลงมาสัมผัสชิ้นงานเอง การขึ้นเนินที่ชันๆ ในบางจังหวะทำให้หัว Probe เกิดอาการติดตัวขึ้นอย่างรวดเร็ว ทำให้การวัดในจุดนั้นเกิด error ที่สูงมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## แนวทางในการพัฒนา

1. สามารถนำแนวคิด ในการขับ Table 2 แกนไปประยุกต์ใช้ในงานอื่นๆ เช่น ในเครื่องกัดขนาด เล็ก เครื่องPlotter
2. ถ้าหากการควบคุมของ Motor เป็นการควบคุมแบบป้อนกลับ จะทำให้มีความแม่นยำ ถูกต้อง มากขึ้น
3. การเคลื่อนที่ในแนวแกน z ตามที่ได้กล่าวไปแล้วในบทสรุป ถ้าหากจะใช้ Motor ควบคุมการ เคลื่อนที่เป็นแกนที่ 3 จะทำให้แก้ปัญหาในจุดนี้ได้



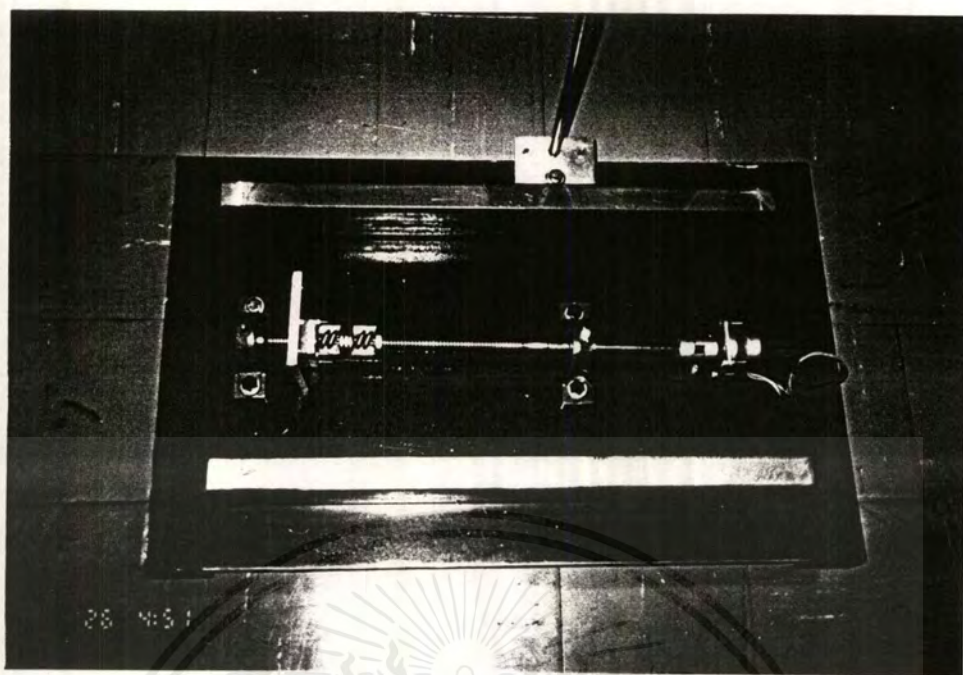
## เอกสารอ้างอิง

1. รู้ลึก รู้จริง! Style Borland Delphi ,จารุวรรณ ระวิภัคตร์
2. แนะนำภาษา Pascal โดย Tirbo Pascal ,ชัชวาล ชนดีหงส์
3. คู่มือการเขียนเครื่องมือช่วยพัฒนา โปรแกรมด้วย Borland Delphi 3 ,จิรายุ วิริยะพิบูล
4. Stepping Motor ,นิตยสาร เชมมิกอนคัลเคเตอร์ ฉบับที่167
5. คัมภีร์ โปรแกรมเมอร์ ,Peter Norton

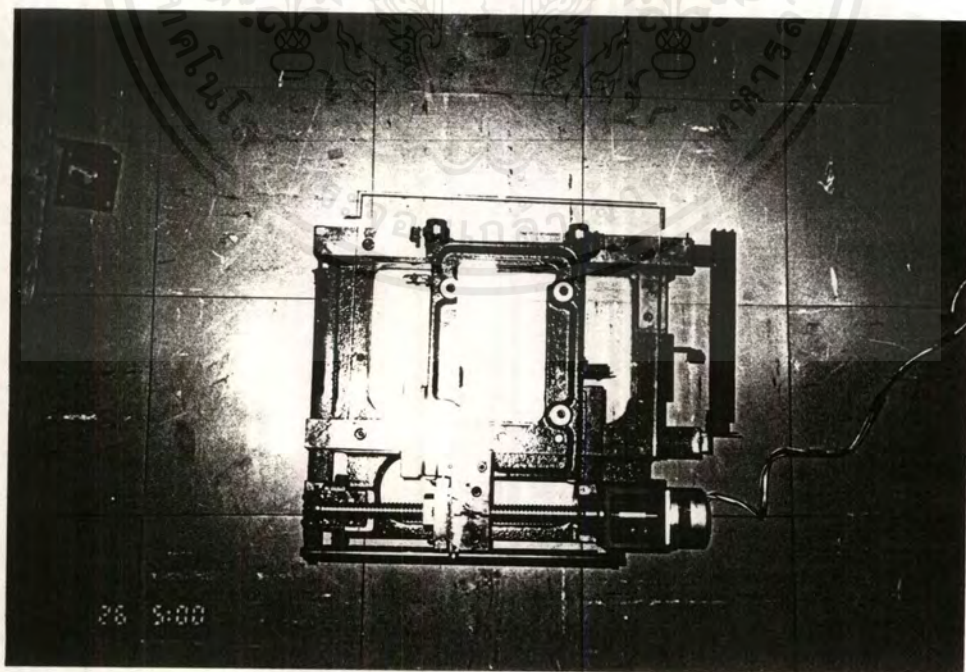




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

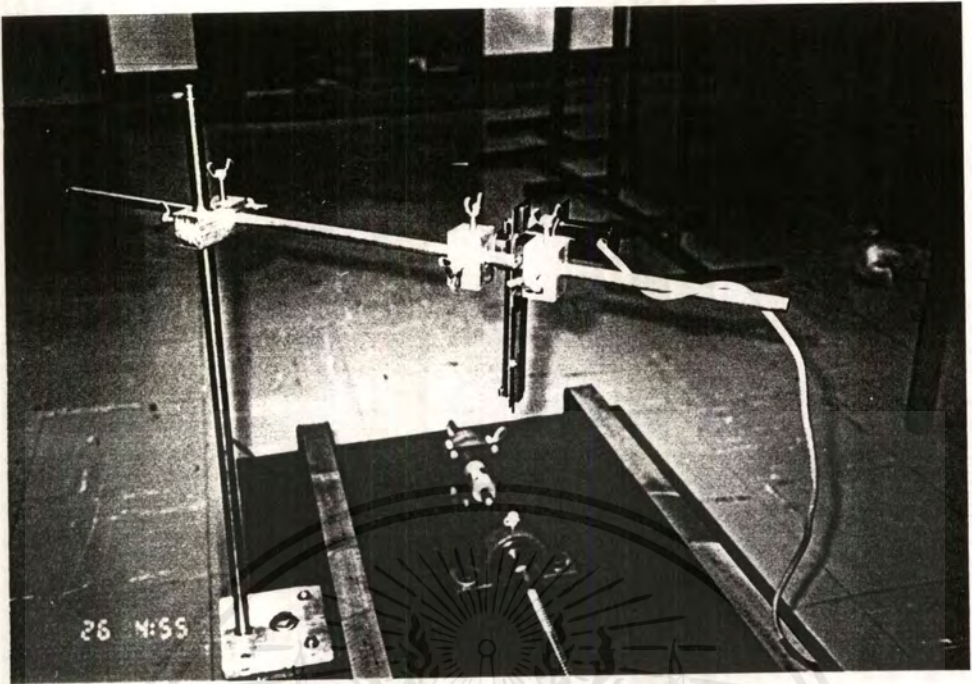


รูปที่ 1 ฐานผ่านเหล็กพร้อม Lead Screw

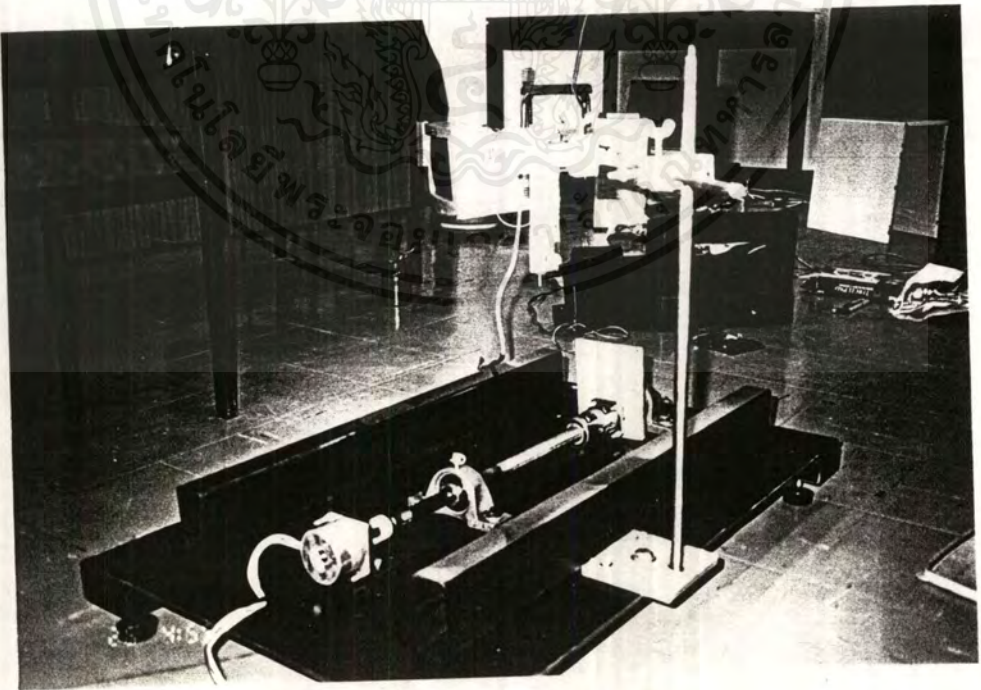


รูปที่ 2 Table ตัวบน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

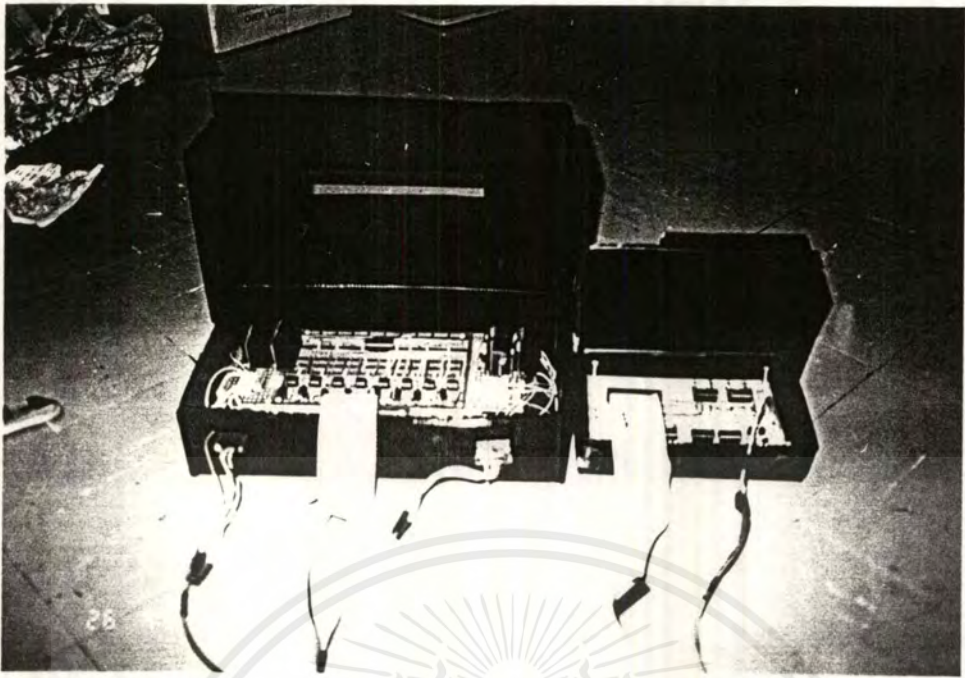


รูปที่ 3 ชุดหัว Probe

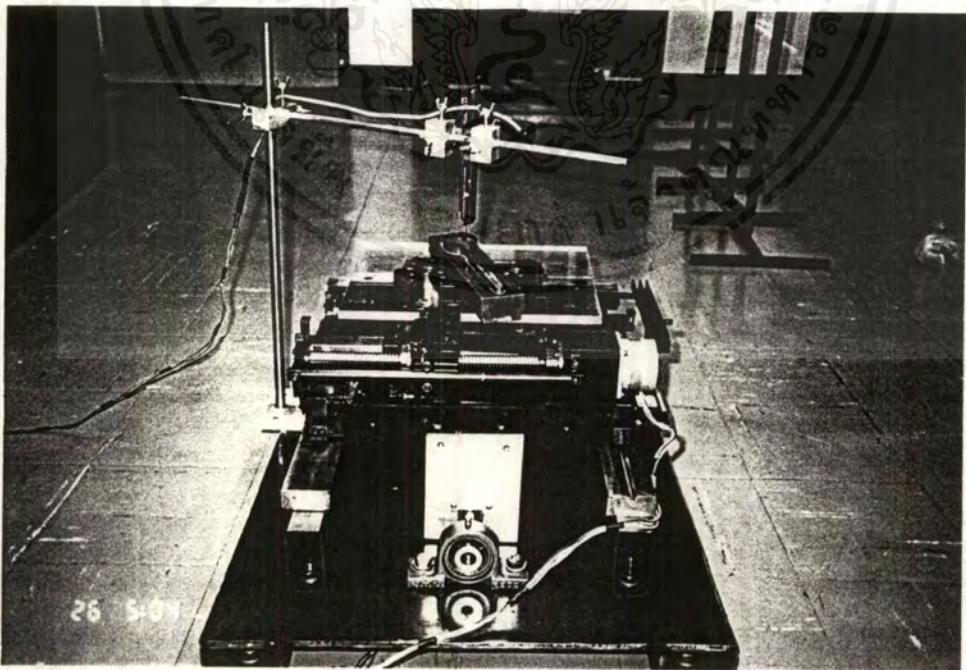


รูปที่ 4 ชุดหัว Probe พร้อมฐานแผ่นเหล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

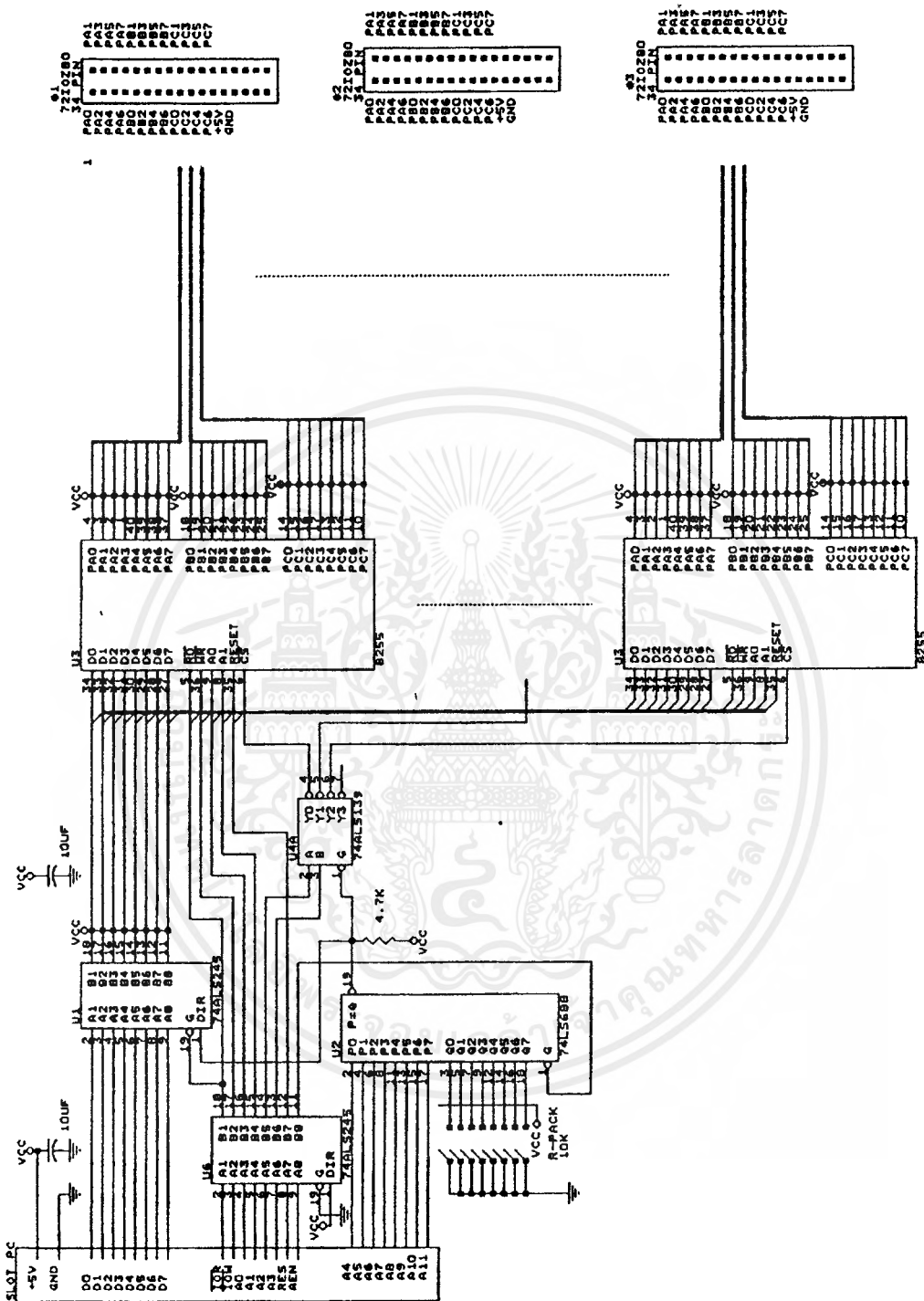


รูปที่ 5 วงจรนับและวงจร Driver ควบคุมการทำงานของ Stepping Motor



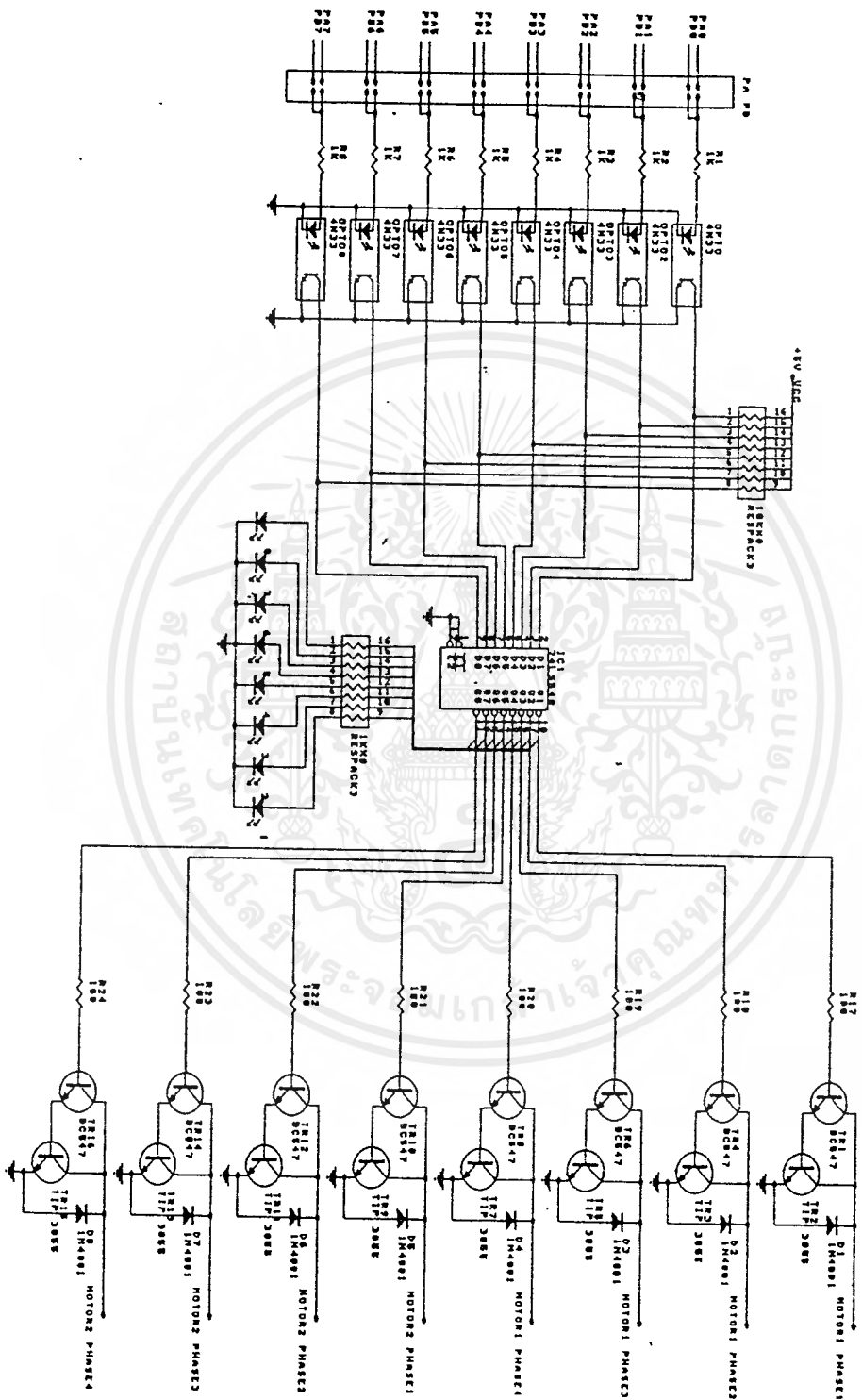
รูปที่ 6 เครื่อง Scan Surface Machine ( สมบูรณ์ )

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



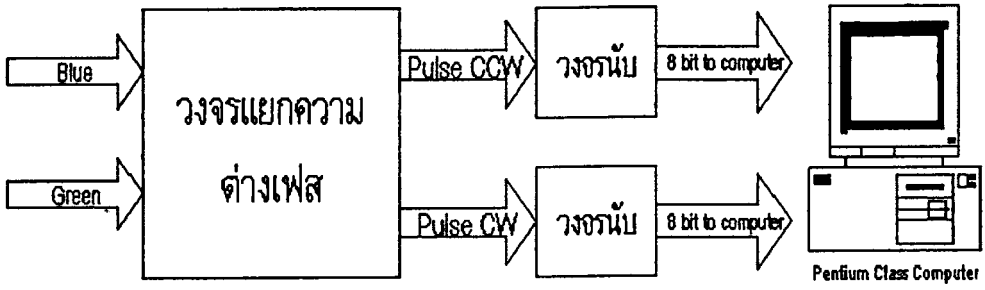
วงจร Card Port 8255 อย่างคร่าวๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

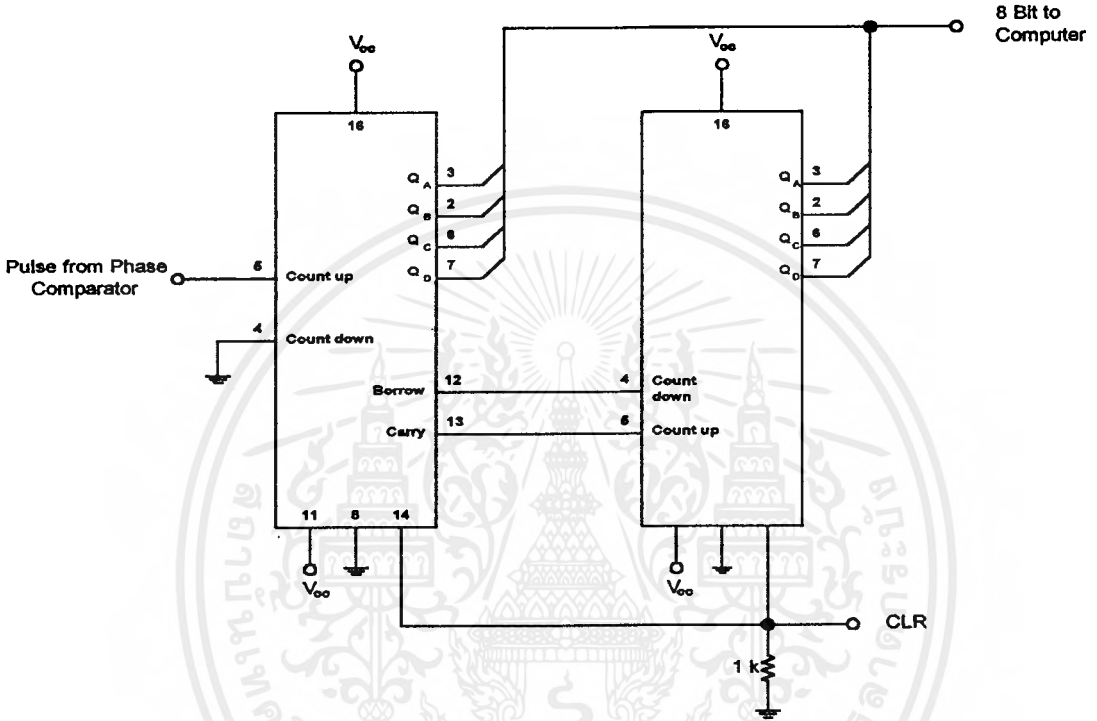


วงจร DRIVER ควบคุมการทำงานของ STEPPING MOTOR

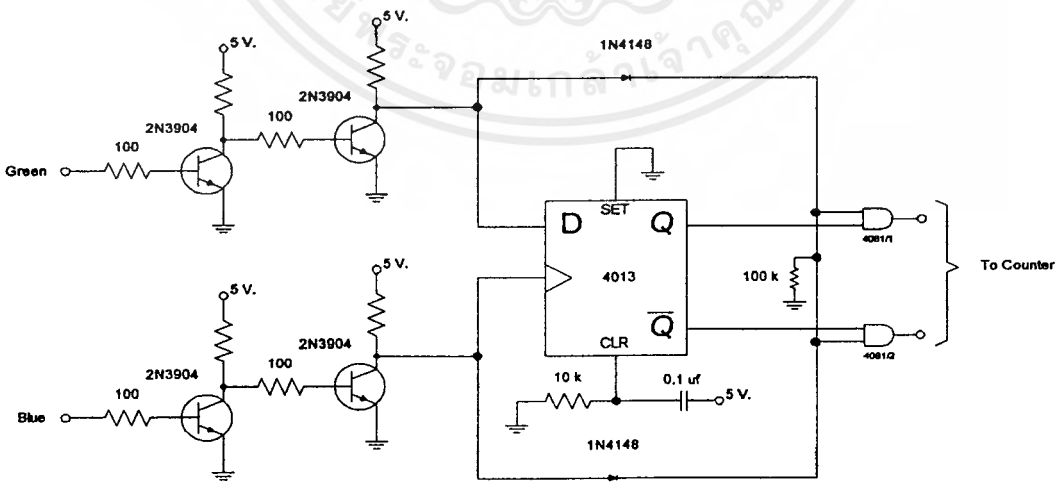
นี่เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
 ใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



แผนภูมิการต่อวงจร Encoder



วงจรมับพัลส์



วงจรแยกความต่างเฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

unit project\_work;

## interface

### uses

Windows, Messages, SysUtils, Classes, Graphics, Controls, Forms, Dialogs,  
ExtCtrls; StdCtrls, ComCtrls, DdeMan, OleCtrls, Buttons, OleCtrls, Bulb,  
ActBtn;

### type

\_2dpoint=record

x:integer;y:integer;

end;

\_3dpoint=record

x:real;y:real;z:real;

end;

atad=array[0..79] of char;

can=array[1..25] of shortint;

TCumpeerawut = class(TForm)

DdeClientConv\_pos: TDdeClientConv;

DdeClientConv\_value: TDdeClientConv;

PageControl1: TPageControl;

TabSheet1: TTabSheet;

Image2: TImage;

GroupBox1: TGroupBox;

up: TButton;

left: TButton;

right: TButton;

down: TButton;

set\_position: TButton;

Edit\_excel: TEdit;

edit\_position: TEdit;

set\_z: TButton;

GroupBox2: TGroupBox;

start\_: TButton;

Pause: TButton;

cancel: TButton;

quit\_manual: TButton;

เอ value\_z\_begin: TEdit; รวบรวมไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

GroupBox3: TGroupBox;  
ready: TButton;  
Edit\_x1: TEdit;  
Edit\_y1: TEdit;  
Edit\_x2: TEdit;  
Edit\_y2: TEdit;  
Horizontal: TRadioButton;  
Vertical: TRadioButton;  
begin\_: TButton;  
pause\_: TButton;  
quit\_automatic: TButton;  
GroupBox4: TGroupBox;  
resolution: TEdit;  
delay\_: TEdit;  
v\_x: TEdit;  
v\_y: TEdit;  
v\_z: TEdit;  
TabSheet2: TTabSheet;  
Image1: TImage;  
TabSheet3: TTabSheet;  
TabSheet4: TTabSheet;  
TabSheet5: TTabSheet;  
kmitl\_image: TImage;  
Image3: TImage;  
Erase\_: TButton;  
rota\_x: TEdit;  
rota\_y: TEdit;  
rota\_z: TEdit;  
TabSheet6: TTabSheet;  
OleContainer1: TOleContainer;  
OleContainer2: TOleContainer;  
x\_up: TSpeedButton;  
x\_down: TSpeedButton;  
y\_up: TSpeedButton;  
y\_down: TSpeedButton;



z\_up: TSpeedButton;  
z\_down: TSpeedButton;  
rota\_step: TEdit;  
draw\_haha: TSpeedButton;  
topview: TImage;  
frontview: TImage;  
isometricview: TImage;  
trace: TButton;  
eraser: TButton;  
pulse1: TEdit;  
pulse2: TEdit;  
set\_test: TButton;  
distance\_test: TEdit;  
Label1: TLabel;  
delay\_test: TEdit;  
left\_test: TBitBtn;  
right\_test: TBitBtn;  
down\_test: TBitBtn;  
Up\_test: TBitBtn;  
inport304: TEdit;  
inport306: TEdit;  
memo: TButton;  
Label2: TLabel;  
Label3: TLabel;  
Label4: TLabel;  
Label5: TLabel;  
Boundary: TBitBtn;  
Bulb1: TBulb;  
Bulb2: TBulb;  
Bulb3: TBulb;  
Bulb4: TBulb;  
Bulb5: TBulb;  
Bulb6: TBulb;  
Bulb7: TBulb;  
Bulb8: TBulb;



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Label6: TLabel;  
zoommm: TEdit;  
zoom_up: TSpeedButton;  
zoom_down: TSpeedButton;  
zoom_d: TSpeedButton;  
zoom_u: TSpeedButton;  
zoom_: TEdit;  
Label7: TLabel;  
Label8: TLabel;  
Label9: TLabel;  
ActiveBtn1: TActiveBtn;  
ActiveBtn2: TActiveBtn;  
ActiveBtn3: TActiveBtn;  
ActiveBtn4: TActiveBtn;  
ActiveBtn5: TActiveBtn;  
Label10: TLabel;  
Label11: TLabel;  
Label12: TLabel;  
Shift_x: TRadioButton;  
Shift_y: TRadioButton;  
Shift_z: TRadioButton;  
pandown: TSpeedButton;  
panup: TSpeedButton;  
panshift: TEdit;  
Label13: TLabel;  
Label14: TLabel;  
Label15: TLabel;  
active: TRadioButton;  
procedure FormCreate(Sender: TObject);  
procedure set_positionClick(Sender: TObject);  
procedure set_zClick(Sender: TObject);  
procedure start_Click(Sender: TObject);  
procedure quit_manualClick(Sender: TObject);  
procedure cancelClick(Sender: TObject);  
procedure PauseClick(Sender: TObject);
```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure leftClick(Sender: TObject);
procedure rightClick(Sender: TObject);
procedure upClick(Sender: TObject);
procedure downClick(Sender: TObject);
procedure readyClick(Sender: TObject);
procedure begin_Click(Sender: TObject);
procedure quit_automaticClick(Sender: TObject);
procedure Erase_Click(Sender: TObject);
procedure x_upClick(Sender: TObject);
procedure x_downClick(Sender: TObject);
procedure y_upClick(Sender: TObject);
procedure y_downClick(Sender: TObject);
procedure z_upClick(Sender: TObject);
procedure z_downClick(Sender: TObject);
procedure draw_hahaClick(Sender: TObject);
procedure traceClick(Sender: TObject);
procedure eraserClick(Sender: TObject);
procedure set_testClick(Sender: TObject);
procedure left_testClick(Sender: TObject);
procedure right_testClick(Sender: TObject);
procedure Up_testClick(Sender: TObject);
procedure down_testClick(Sender: TObject);
procedure memoClick(Sender: TObject);
procedure FormActivate(Sender: TObject);
procedure BoundaryClick(Sender: TObject);
procedure edit_positionKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure distance_testKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
procedure zoom_upClick(Sender: TObject);
procedure zoom_downClick(Sender: TObject);
procedure zoom_dClick(Sender: TObject);
procedure zoom_uClick(Sender: TObject);
procedure ActiveBtn1Down(Activate: TMouseOrKey);
procedure ActiveBtn2Down(Activate: TMouseOrKey);
```

```
procedure ActiveBtn3Down(Activate: TMouseOrKey);
procedure ActiveBtn4Down(Activate: TMouseOrKey);
procedure ActiveBtn5Down(Activate: TMouseOrKey);
procedure panupClick(Sender: TObject);
procedure pandownClick(Sender: TObject);
```

private

```
function inport(portno:smallint):byte;
procedure outport(portno:smallint;data:byte);
    procedure left_ta(turn:smallint;delay:smallint);
    procedure right_ta(turn:smallint;delay:smallint);
    procedure up_ta(turn:smallint;delay:smallint);
    procedure down_ta(tum:smallint;delay:smallint);
```

```
procedure light;
```

```
procedure display_m(direc:atad);
```

```
procedure display_high;
```

```
procedure cal_ml;
```

```
procedure mem_zaxis;
```

```
procedure delay_table;
```

public

```
procedure rotation(p1:_3dpoint);
```

```
procedure purse(p1:_3dpoint;ggg1,ggg2,ggg3,ggg4:integer;var p2:_2dpoint);
```

```
procedure trans(p1:_3dpoint;var p2:_3dpoint);
```

```
procedure draw;
```

```
procedure drawing_expert(dd1,dd2,dd3,dd4,rox,roy,roz,cm:smallint);
```

```
procedure select_image(cm:smallint;f:_2dpoint);
```

```
procedure rotate_image(ro_sx,ro_sy,ro_sz:smallint;add:boolean);
```

```
procedure drawing_excellent(dd1,dd2,dd3,dd4,rox,roy,roz,cm:smallint);
```

```
procedure zoomimage(zzz:smallint);
```

```
end;
```

var

```
Cumpeerawut: TCumpeerawut;
```

```
rv:smallint; delay:integer;{rv=amout revolution delay=delay}
```

```
i,j,a,b,v{v=resolution}:integer;          atemp,btemp:integer;
```

```
x1,y1,x2,y2:shortint;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

zbegin:extended; mem\_z:array[1..750,1..750]of extended;

g:byte; hight\_z:extended; mem\_x:array[1..200,1..200]of integer;

acad,hg:atad; mem\_y:array[1..200,1..200]of integer;

can1,can2,can3,can4:can;

star,direc,displacement\_z\_pluse,drawimage:string[15];

ab:array[0..3]of byte;

m,n:byte;

k,displacement\_pluse:smallint;

count,shift:integer;

port\_test:smallint;

range:smallint;

a\_boundaty:array[1..500,1..2] of smallint;

jitapa,jitapun:array[1..500,0..1] of integer; jiji:boolean;

// graphic graphic graphic graphic graphic graphic graphic

// graphic graphic graphic graphic graphic graphic graphic

x\_sin,y\_sin,z\_sin,x\_cos,y\_cos,z\_cos,x\_shift,y\_shift,z\_shift:extended;

getmaxxx,getmaxxy:integer;

zoom:integer;

a\_c\_e\_:\_3dpoint; f\_:\_2dpoint; i\_g\_h\_:\_3dpoint;j\_:\_2dpoint;

cm\_image:array [1..10]of boolean;

rohx,rohy,roh\_z:smallint;

implementation

{*\$R* \*.DFM}

procedure Tcumpeerawut.outport(portno:smallint;data:byte);

begin

asm

push dx

mov dx,portno

mov al,data

out dx,al

pop dx

end;

end;

function Tcumpeerawut.inport (portno:smallint):byte;

var

temp:byte;

เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin
asm
push dx
mov dx,portno
in al,dx
mov temp,al
pop dx
end;
inport := temp;
end;

```

```

procedure tcumpeerawut.delay_table;

```

```

var k:smallint;

```

```

begin

```

```

  for k:=1 to delay do

```

```

    begin

```

```

      for k:=1 to delay do

```

```

        begin

```

```

          for k:=1 to delay do

```

```

            begin light;

```

```

          end;

```

```

        end;

```

```

      end;

```

```

    end;

```

```

procedure Tcumpeerawut.left_ta(turn,delay:smallint);

```

```

begin

```

```

  count:=0; shift:=m;  outport($305,255);outport($305,0);

```

```

  repeat

```

```

    m:=abs(shift mod 4);

```

```

    outport($300,ab[m]); delay_table;

```

```

    shift :=shift +1; count:=count+1;

```

```

  until count >turn;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

outport($300,0);
pulse1.text:=inttostr(inport($304)); inport304.text:=inttostr(inport($304));
pulse2.text:=inttostr(inport($306)); inport306.text:=inttostr(inport($306));
if star='start' then begin
  displacement_pluse:=inport($306)-inport($304);
  hight_z:=(displacement_pluse/8);
  end;
end;

procedure Tcumpeerawut.right_ta(turn,delay:smallint);
begin
  count:=0; shift:=(((turn div 4)*4)+m+4 );outport($305,255);outport($305,0);
  repeat
  m:=abs(shift mod 4);
  outport($300,ab[m]);delay_table;
  shift :=shift -1; count:=count+1;
  until count >turn;
  outport($300,0);
  pulse1.text:=inttostr(inport($304)); inport304.text:=inttostr(inport($304));
  pulse2.text:=inttostr(inport($306)); inport306.text:=inttostr(inport($306));
  if star='start' then begin
    displacement_pluse:=inport($306)-inport($304);
    hight_z:=(displacement_pluse/8);
    end;
  end;
end;

```

```

procedure Tcumpeerawut.up_ta(turn,delay:smallint);
begin
  count:=0; shift:=n;outport($305,255);outport($305,0);
  repeat
  n:=abs(shift mod 4);
  outport($302,ab[n]);delay_table;
  shift :=shift +1; count:=count+1;
  until count >turn;
  outport($302,0);

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

pulse1.text:=inttostr(inport($304)); inport304.text:=inttostr(inport($304));
pulse2.text:=inttostr(inport($306)); inport306.text:=inttostr(inport($306));
if star='start' then begin
displacement_pluse:=inport($306)-inport($304);
high_z:=(displacement_pluse/8);//*2*pi*15{assume r= 15 mm/500)+zbegin; }
end;
end;

procedure Tcumpeerawut.down_ta(turn,delay:smallint);
begin
count:=0; shift:=(((turn div 4)*4)+n+4 );outport($305,255);outport($305,0);
repeat
n:=abs(shift mod 4);
outport($302,ab[n]);delay_table;
shift :=shift -1; count:=count+1;
until count >turn;
outport($302,0);
pulse1.text:=inttostr(inport($304)); inport304.text:=inttostr(inport($304));
pulse2.text:=inttostr(inport($306)); inport306.text:=inttostr(inport($306));
if star='start' then begin
displacement_pluse:=inport($306)-inport($304);
high_z:=(displacement_pluse /8 )//_pluse*2*pi*15{assume r= 15 mm/500)+zbegin;
end;
end;

procedure tcumpeerawut.light;
begin
if inport($300)=5 then begin bulb1.LightOn:=true; end
else begin bulb1.LightOn:=false; end;
if inport($300)=20 then begin bulb2.LightOn:=true; end
else begin bulb2.LightOn:=false; end;
if inport($300)=80 then begin bulb3.LightOn:=true; end
else begin bulb3.LightOn:=false; end;
if inport($300)=65then begin bulb4.LightOn:=true; end
else begin bulb4.LightOn:=false; end;
if inport($302)=5 then begin bulb5.LightOn:=true; end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

else begin bulb5.LightOn:=false; end;
if inport($302)=20 then begin bulb6.LightOn:=true; end
else begin bulb6.LightOn:=false; end;
if inport($302)=80 then begin bulb7.LightOn:=true; end
else begin bulb7.LightOn:=false; end;
if inport($302)=65 then begin bulb8.LightOn:=true; end
else begin bulb8.LightOn:=false; end;
end;

procedure TCumpeerawut.display_m(direc:atad);
begin
begin
if (direc='up') and(b<btemp) then begin jitapa[b,0]:=a;jitapa[b,1]:=a;end;
if (direc='down')and(b>btemp) then begin jitapa[b,0]:=a;jitapa[b,1]:=a;end;
end;
strcpy(acad,inttostr(i*v)+' '+inttostr(j*v));
edit_position.Text:=acad;
if a<=26 then begin edit_excel.Text:=inttostr(b)+' '+inttostr(a)+'('+chr(a+64)+')';end;
if a>26 then begin edit_excel.text:=inttostr(b)+' '+inttostr(a)+'('+chr(a div 26+64)+chr((a mod 26
)+64)+')'; end;
ddeclientconv_pos.PokeData('r'+inttostr(b)+'c'+inttostr(a),direc);
end;

procedure tcumpeerawut.display_hight;
begin
zbegin:=zbegin+hight_z;
strcpy(hg,acad+' '+floattostr(zbegin));
ddeclientconv_value.PokeData('r'+inttostr(b)+'c'+inttostr(a),hg);
edit_position.Text:=hg;
end;

procedure Tcumpeerawut.mem_zaxis;
begin
mem_z[a,b]:=zbegin;
mem_x[a,b]:=i*v;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
mem_y[a,b]:=j*v;
```

```
end;
```

```
procedure TCumpeerawut.FormCreate(Sender: TObject);
```

```
begin
```

```
outport($303,128);outport($307,153);
```

```
ab[0]:=5; ab[1]:=20; ab[2]:=80; ab[3]:=65;
```

```
  ddeclientconv_pos.SetLink('excel','c:\my documents\[test1.xls]sheet1');
```

```
  ddeclientconv_value.SetLink('excel','c:\my documents\[test3.xls]sheet1');
```

```
  ddeclientconv_pos.ServiceApplication:='excel';
```

```
  ddeclientconv_pos.DdeTopic:= 'c:\my documents\[test1.xls]sheet1';
```

```
  ddeclientconv_value.ServiceApplication:='excel';
```

```
  ddeclientconv_value.DdeTopic:= 'c:\my documents\[test3.xls]sheet1';
```

```
  i:=0; j:=0; display_m('center');
```

```
  a:=6; b:=6;
```

```
  can1[15]:=0; can2[15]:=0; g:=25;
```

```
  star:='end';
```

```
  rohx:=-30;
```

```
  rohy:=-10;
```

```
  rohz:=-30;
```

```
  x_shift:=0;y_shift:=0;z_shift:=0;
```

```
end;
```

```
procedure TCumpeerawut.set_positionClick(Sender: TObject);
```

```
var g,f:byte;  h,ii:shortint;
```

```
begin
```

```
v:=strtoint(resolution.text);
```

```
rv:=50*v;
```

```
delay:=strtoint(delay_.Text);
```

```
a:=(100 div v)+1; b:=a;  atemp:=a; btemp:=b;
```

```
if (v>8) then {because of computer hang}
```

```
  begin
```

```
    for f:=1 to (a+1)*2 do
```

```
      for g:=1 to (a+1)*2 do
```

```
        begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ddeclientconv_pos.PokeData('r'+inttostr(f)+'c'+inttostr(g),");
ddeclientconv_value.PokeData('r'+inttostr(f)+'c'+inttostr(g),");
end;
end;

for h:= 1 to (a+1)*2 do
begin for ii:=1 to (a+1)*2 do
begin
mem_x[h,ii]:= 9999;
mem_y[h,ii]:=9999;
mem_z[h,ii]:=0;
end;
jitapa[h,1]:=0;jitapa[h,0]:=0;
jitapun[h,1]:=0;jitapun[h,0]:=0;
end;
i:=0;
j:=0;
display_m('center');
draw; range:=(a+1)*2;
zoom:=30;
jitapa[btemp,1]:=a; jitapa[btemp,0]:=a;
end;

procedure TCumpeerawut.set_zClick(Sender: TObject);
begin
outport($305,255);outport($305,0);
zbegin :=strtoint(value_z_begin.text);
hight_z:=0;
display_hight;
end;

procedure TCumpeerawut.start_Click(Sender: TObject);
begin
star:='start';
set_zclick(sender);

```



```
mem_zaxis;
```

```
end;
```

```
procedure TCumpeerawut.quit_manualClick(Sender: TObject);
```

```
begin
```

```
    star:='end';
```

```
end;
```

```
procedure TCumpeerawut.cal_ml;
```

```
var u:byte;
```

```
begin
```

```
    can1[25]:=i; can2[25]:=j;
```

```
    can3[25]:=a; can4[25]:=b;
```

```
    begin for u:=1 to 24 do
```

```
        begin can1[u]:=can1[u+1]; can2[u]:=can2[u+1]; can3[u]:=can3[u+1]; can4[u]:=can4[u+1];end;
```

```
    end;
```

```
end;
```

```
procedure TCumpeerawut.cancelClick(Sender: TObject);
```

```
begin
```

```
    ddeclientconv_pos.PokeData('r'+inttostr(b)+'c'+inttostr(a), "");
```

```
    if star<>'end' then begin ddeclientconv_value.PokeData('r'+inttostr(b)+'c'+inttostr(a), "");end;
```

```
    g:=g-1;
```

```
    begin
```

```
        i:=can1[g];j:=can2[g]; a:=can3[g];b:=can4[g];
```

```
        if (can1[g]>can1[g-1]) then begin left_ta(rv,delay); end;
```

```
        if (can1[g]<can1[g-1]) then begin right_ta(rv,delay);end;
```

```
        if (can2[g]>can2[g-1]) then begin down_ta(rv,delay);end;
```

```
        if (can2[g]<can2[g-1]) then begin up_ta(rv,delay); end;
```

```
        display_m('cancel');
```

```
    end;
```

```
end;
```

```
procedure TCumpeerawut.PauseClick(Sender: TObject);
```

```
begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
star:='pause';
```

```
end;
```

```
procedure TCumpeerawut.leftClick(Sender: TObject);
```

```
begin
```

```
    cal_ml;g:=25;
```

```
    i:=i-1; a:=a-1; display_m('left');
```

```
    left_ta(rv,delay);          draw; memoclick(sender);
```

```
if (star='start') then begin mem_zaxis;display_high;
```

```
    if active.Checked=true then begin traceclick(sender); end;
```

```
    end;
```

```
end;
```

```
procedure TCumpeerawut.rightClick(Sender: TObject);
```

```
begin
```

```
    cal_ml;g:=25;
```

```
    i:=i+1; a:=a+1; display_m('right');
```

```
    right_ta(rv,delay);        draw; memoclick(sender);
```

```
if (star='start') then begin mem_zaxis;display_high;
```

```
    if active.Checked=true then begin traceclick(sender); end;
```

```
    end;
```

```
end;
```

```
procedure TCumpeerawut.upClick(Sender: TObject);
```

```
begin
```

```
    cal_ml;g:=25;
```

```
    j:=j+1; b:=b-1; display_m('up');
```

```
    up_ta(rv,delay);          draw; memoclick(sender);
```

```
if (star='start') then begin mem_zaxis;display_high;
```

```
    if active.Checked=true then begin traceclick(sender); end;
```

```
    end;
```

```
end;
```

```
procedure TCumpeerawut.downClick(Sender: TObject);
```

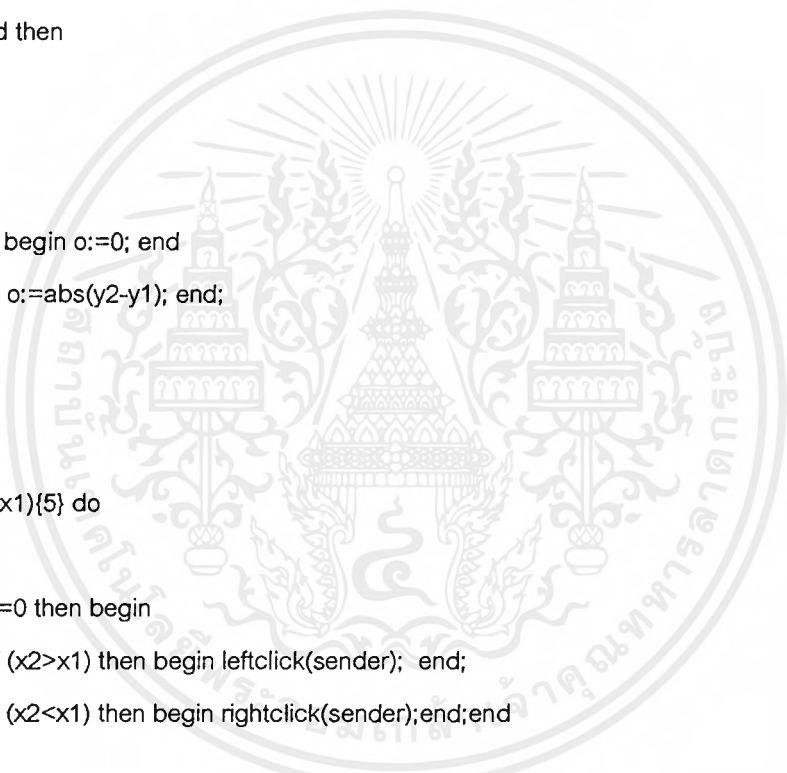
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

x1:=strtoint(edit_x1.text);  x1:=(x1) div v;
y1:=strtoint(edit_y1.text);  y1:=(y1) div v;
x2:=strtoint(edit_x2.text);  x2:=(x2) div v;
y2:=strtoint(edit_y2.text);  y2:=(y2) div v;

if (i<>x1)or(j<>y1) then begin ShowMessage('Touch Ready Now'); end
else
begin
  count :=0; draw;
  star:='start';
  mem_zaxis;
if horizontal.Checked then
begin
set_zclick(sender);
display_hight;
if abs(y2-y1)=0 then begin o:=0; end
  else begin o:=abs(y2-y1); end;
repeat
begin
count:=count+1;
for n:=1 to abs(x2-x1){5} do
begin
if (count mod 2)=0 then begin
if (x2>x1) then begin leftclick(sender); end;
if (x2<x1) then begin rightclick(sender);end;end
else begin
if (x2>x1) then begin rightclick(sender);end;
if (x2<x1) then begin leftclick(sender); end;end;
end;
if y2<>j then begin
if (y2>y1) then begin upclick(sender);end;
if (y2<y1) then begin downclick(sender);end;
end;
end;
until count=o+1 ;
end;

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

// graphic graphic graphic graphic graphic graphic graphic graphic
// graphic graphic graphic graphic graphic graphic graphic graphic
procedure tcumpeerawut.rotation(p1:_3dpoint);
begin
  x_sin:=sin(p1.x*pi/180); y_sin:=sin(p1.y*pi/180);
  z_sin:=sin(p1.z*pi/180); x_cos:=cos(p1.x*pi/180);
  y_cos:=cos(p1.y*pi/180); z_cos:=cos(p1.z*pi/180);
end;
procedure tcumpeerawut.trans(p1:_3dpoint;var p2:_3dpoint);
var a,b,c:_3dpoint;
begin
  a.x:=p1.x; a.y:=x_cos*p1.y-x_sin*p1.z; a.z:=x_sin*p1.y+x_cos*p1.z;
  b.x:=y_sin*a.z+y_cos*a.x; b.y:=a.y; b.z:=y_cos*a.z-y_sin*a.x;
  c.x:=z_cos*b.x-z_sin*b.y; c.y:=z_sin*b.x+z_cos*b.y; c.z:=b.z;
  p2.x:=c.x+x_shift; p2.y:=c.y+y_shift; p2.z:=c.z+z_shift;
end;
procedure tcumpeerawut.purse(p1:_3dpoint;ggg1,ggg2,ggg3,ggg4:integer;var p2:_2dpoint);
const d1=50;d2=50;
begin
  getmaxxx:=ggg1; getmaxxy:=ggg2;
  p2.x:=round(d2/(d1+d2-p1.z)*p1.x*(ggg1/10))+ggg3 div 2;
  p2.y:=(ggg4 div 2)-round(d2/(d1+d2-p1.z)*p1.y*(ggg2/10));
end;
procedure tcumpeerawut.select_image(cm:smallint;f:_2dpoint);
begin
  if (cm=1) and (cm_image[cm]=true) then begin image1.canvas.LineTo(f_x,f_y);end;
  if (cm=1) and (cm_image[cm]=false) then begin image1.canvas.moveTo(f_x,f_y);end;

  if (cm=2) and (cm_image[cm]=true) then begin topview.canvas.LineTo(f_x,f_y);end;
  if (cm=2) and (cm_image[cm]=false) then begin topview.canvas.moveTo(f_x,f_y);end;

  if (cm=3) and (cm_image[cm]=true) then begin image3.canvas.LineTo(f_x,f_y);end;
  if (cm=3) and (cm_image[cm]=false) then begin image3.canvas.moveTo(f_x,f_y);end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if (cm=4) and (cm_image[cm]=true) then begin isometricview.canvas.LineTo(f_x,f_y);end;
if (cm=4) and (cm_image[cm]=false) then begin isometricview.canvas.moveTo(f_x,f_y);end;
```

```
if (cm=5) and (cm_image[cm]=true) then begin frontview.canvas.LineTo(f_x,f_y);end;
if (cm=5) and (cm_image[cm]=false) then begin frontview.canvas.moveTo(f_x,f_y);end;
```

```
end;
```

```
procedure tcumpeerawut.drawing_expert(dd1,dd2,dd3,dd4,rox,roy,roz,cm:smallint);
```

```
var zz:smallint;
```

```
begin
```

```
zz:=strtoint(zoom_.text);
```

```
//x_shift:=0; y_shift:=0; z_shift:=0;
```

```
with a_ do
```

```
begin
```

```
x:= rox;
```

```
y:= roy;
```

```
z:= roz;
```

```
end;
```

```
rotation(a_);
```

```
with c_ do
```

```
begin
```

```
x:= (i*v)/zz ;
```

```
y:= (j*v)/zz ;
```

```
z:= zbegin/zz;
```

```
end;
```

```
trans(c_,e_);
```

```
purse(e_,dd1,dd2,dd3,dd4,f_);
```

```
if star<>'start' then begin cm_image[cm]:=false; select_image(cm,f_); end;
```

```
if star='start' then begin cm_image[cm]:=true;select_image(cm,f_); end;
```

```
end;
```

```
procedure tcumpeerawut.draw;
```

```
begin
```

```
drawing_expert(800,800,400,400,strtoint(v_x.Text),strtoint(v_y.Text),strtoint(v_z.Text),1);
```

```
drawing_expert(300,300,130,130,0,0,0,2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;
```

```
procedure TCumpeerawut.Erase_Click(Sender: TObject);
```

```
begin
```

```
  image3.canvas.Rectangle(0,0,400,400);
```

```
end;
```

```
procedure tcumpeerawut.drawing_excellent(dd1,dd2,dd3,dd4,rox,roy,roz,cm:smallint);
```

```
var yy,zz,nub:integer;
```

```
begin
```

```
//x_shift:=0; y_shift:=0; z_shift:=0;
```

```
  with i_ do
```

```
    begin
```

```
      x:= rox;
```

```
      y:= roy;
```

```
      z:= roz;
```

```
    end;
```

```
  rotation(i_);
```

```
  for yy:=1 to range do
```

```
    begin nub:=0;
```

```
    for zz:=1 to range do
```

```
      begin
```

```
        if (mem_y[yy,zz]=9999)and (mem_x[yy,zz]=9999) then begin nub:=0; end;
```

```
        if (mem_y[yy,zz]<>9999)and (mem_x[yy,zz]<>9999) then
```

```
          begin
```

```
            with g_ do
```

```
              begin
```

```
                x:= mem_x[yy,zz]/ zoom;
```

```
                y:= mem_y[yy,zz]/ zoom;
```

```
                z:= mem_z[yy,zz]/ zoom;
```

```
              end;
```

```
            trans(g_,h_);
```

```
            purse(h_dd1,dd2,dd3,dd4,j_);
```

```
            if nub=0 then begin cm_image[cm]:=false; select_image(cm,j_);end
```

```
            else begin cm_image[cm]:=true;select_image(cm,j_); end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

nub:=nub+1;
end;
end;
end;

for zz:=1 to range do
begin nub:=0;
for yy:=1 to range do
begin
if (mem_y[yy,zz]=9999)and (mem_x[yy,zz]=9999) then begin nub:=0; end;
if (mem_y[yy,zz]<>9999)and (mem_x[yy,zz]<>9999) then
begin
with g_ do
begin
x:= mem_x[yy,zz]/zoom;
y:= mem_y[yy,zz]/zoom;
z:= mem_z[yy,zz]/zoom;
end;
trans(g_,h_);
purse(h_dd1,dd2,dd3,dd4,j_);
if nub=0 then begin cm_image[cm]:=false; select_image(cm,j_);end
else begin cm_image[cm]:=true;select_image(cm,j_); end;
nub:=nub+1;
end;
end;
end;
end;

```

```

procedure TCumpeerawut.draw_hahaClick(Sender: TObject);

```

```

begin
drawing_excellent(800,800,400,400,-30,-10,-30,3);
end;

```

```

procedure TCumpeerawut.traceClick(Sender: TObject);

```

```

begin
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

```
drawing_excellent(260,260,130,130,-30,-10,-30,4);
```

```
drawing_excellent(260,260,130,130,90,0,0,5);
```

```
end;
```

```
procedure TCumpeerawut.eraserClick(Sender: TObject);
```

```
begin
```

```
    isometricview.canvas.Rectangle(0,0,130,130);
```

```
    topview.canvas.Rectangle(0,0,130,130);
```

```
    frontview.Canvas.Rectangle(0,0,130,130);
```

```
    image1.Canvas.rectangle(0,0,400,400);
```

```
end;
```

```
procedure tcumpeerawut.zoomimage(zzz:smallint);
```

```
begin
```

```
    zoom:=zoom+zzz;
```

```
    if zoom<=0 then begin zoom:=1; end;
```

```
end;
```

```
// end graphic
```

```
// end graphic
```

```
//          rotatable image
```

```
//          rotatable image
```

```
procedure tcumpeerawut.rotate_image(ro_sx,ro_sy,ro_sz:smallint;add:boolean);
```

```
begin
```

```
    if add=true then begin    rohx:=rohx+ro_sx;rota_x.text:=inttostr(rohx);
```

```
        rohy:=rohy+ro_sy;rota_y.text:=inttostr(rohy);
```

```
        rohz:=rohz+ro_sz;rota_z.text:=inttostr(rohz);
```

```
        drawing_excellent(800,800,400,400,rohx,rohy,rohz,3);
```

```
        end;
```

```
    if add=false then begin    rohx:=rohx-ro_sx;rota_x.text:=inttostr(rohx);
```

```
        rohy:=rohy-ro_sy;rota_y.text:=inttostr(rohy);
```

```
        rohz:=rohz-ro_sz;rota_z.text:=inttostr(rohz);
```

```
        drawing_excellent(800,800,400,400,rohx,rohy,rohz,3);
```

```
        end;
```

```
end;
```

การนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TCumpeerawut.x_upClick(Sender: TObject);
begin
erase_click(sender);
rotate_image(strtoint(rota_step.text),0,0,true);
end;
```

```
procedure TCumpeerawut.x_downClick(Sender: TObject);
begin
erase_click(sender);
rotate_image(strtoint(rota_step.text),0,0,false);
end;
```

```
procedure TCumpeerawut.y_upClick(Sender: TObject);
begin
erase_click(sender);
rotate_image(0,strtoint(rota_step.text),0,true);
end;
```

```
procedure TCumpeerawut.y_downClick(Sender: TObject);
begin
erase_click(sender);
rotate_image(0,strtoint(rota_step.text),0,false);
end;
```

```
procedure TCumpeerawut.z_upClick(Sender: TObject);
begin
erase_click(sender);
rotate_image(0,0,strtoint(rota_step.text),true);
end;
```

```
procedure TCumpeerawut.z_downClick(Sender: TObject);
begin
erase_click(sender);
```

```
rotate_image(0,0,strtoint(rota_step.text),false);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการเรียนเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;
end_rotationimage end_rotationimage end_rotationimage end_rotationimage
end_rotationimage end_rotationimage end_rotationimage end_rotationimage
```

```
test test test test test test test test test//
test test test test test test test test test//
```

```
procedure TCumpeerawut.set_testClick(Sender: TObject);
begin
```

```
inport($303,128);outport($307,153);
ab[0]:=5; ab[1]:=20; ab[2]:=80; ab[3]:=65;
:=strtoint(distance_test.text)*50;
delay:=strtoint(delay_test.text);
:=51;b:=51;
end;
```

```
procedure TCumpeerawut.left_testClick(Sender: TObject);
begin
```

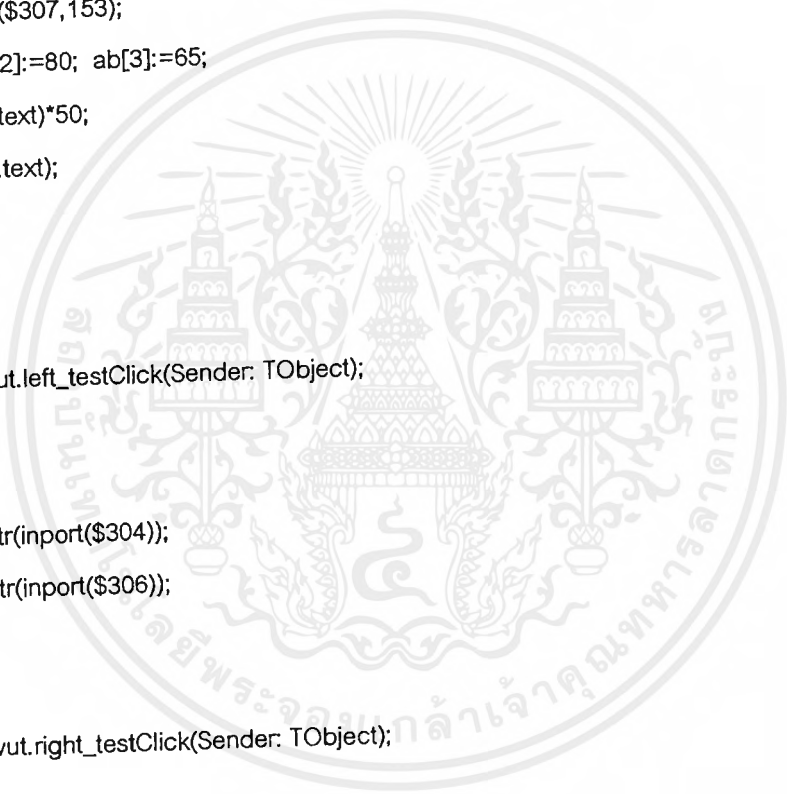
```
left_ta(rv,delay);
inport304.text:=inttostr(inport($304));
inport306.text:=inttostr(inport($306));
end;
```

```
procedure TCumpeerawut.right_testClick(Sender: TObject);
begin
```

```
right_ta(rv,delay);
inport304.text:=inttostr(inport($304));
inport306.text:=inttostr(inport($306));
end;
```

```
procedure TCumpeerawut.Up_testClick(Sender: TObject);
begin
```

```
up_ta(rv,delay);
inport304.text:=inttostr(inport($304));
inport306.text:=inttostr(inport($306));
end;
```



```

procedure TCumpeerawut.down_testClick(Sender: TObject);
begin
    down_ta(rv,delay);
    inport304.text:=inttostr(inport($304));
    inport306.text:=inttostr(inport($306));
end;

// end_test end_test end_test end_test end_test end_test end_test
// end_test end_test end_test end_test end_test end_test end_test

```

```

procedure TCumpeerawut.memoClick(Sender: TObject);
var ill:integer;
begin
    for ill:=1 to range do
        begin
            if (b=ill)and(a>jitapa[ill,1]) then begin jitapa[ill,1]:=a; end;
            if (b=ill)and(a<jitapa[ill,0]) then begin jitapa[ill,0]:=a;end;
        end;
    end;
end;

```

```

procedure TCumpeerawut.FormActivate(Sender: TObject);
begin
    ShowMessage('Touch set_position button please');
end;

```

```

procedure TCumpeerawut.BoundaryClick(Sender: TObject);
var tung,val_1,val_2:integer; less,more:byte;
begin
    less:=0;more:=1;
    for tung:=1 to range do
        begin
            jitapun[tung,0]:=jitapa[tung,0];
            jitapun[tung,1]:=jitapa[tung,1];
        end;
    end;

```

เอกสารนี้เป็นทรัพย์สินของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง  
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin
  if (jitapun[tung,0]<>0)and (jitapun[tung,1]<>0) then
  begin
    begin
      edit_x1.text:=inttostr(mem_x[jitapun[tung,less],tung]);
      edit_y1.text:=inttostr(mem_y[jitapun[tung,less],tung]);
    end;
    readyclick(sender);
    edit_x2.text:=inttostr(mem_x[jitapun[tung,more],tung]);
    edit_y2.text:=inttostr(mem_y[jitapun[tung,more],tung]);
    horizontal.Checked:=true;
    begin_click(sender);

    if (jitapun[tung+1,0]<>0)and (jitapun[tung+1,1]<>0) then
    begin
      val_1:= (strtoint(edit_x2.text))-(mem_x[jitapun[tung,0],tung]);
      val_2:= (strtoint(edit_x2.text))-(mem_x[jitapun[tung,1],tung]);
      if abs(val_2)<abs(val_1) then begin less:=1; more:=0 end
      \else begin less:=0; more:=1 end;
    end;
  end;
end;
end;
end;

procedure TCumpeerawut.edit_positionKeyDown(Sender: TObject; var Key: Word;
  Shift: TShiftState);
begin
  case key of
    vk_left: leftclick(sender);
    vk_right: rightclick(sender);
    vk_up: upclick(sender);
    vk_down: downclick(sender);
  end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

end;

```
procedure TCumpeerawut.distance_testKeyDown(Sender: TObject; var Key: Word;
```

```
Shift: TShiftState);
```

```
begin
```

```
case key of
```

```
vk_left: leftclick(sender);
```

```
vk_right: rightclick(sender);
```

```
vk_up: upclick(sender);
```

```
vk_down: downclick(sender);
```

```
end;
```

```
end;
```

```
procedure TCumpeerawut.zoom_upClick(Sender: TObject);
```

```
begin
```

```
zoomimage(strtoint(zoommm.text)*-1);
```

```
erase_click(sender);
```

```
drawing_excellent(800,800,400,400,strtoint(rota_x.text),strtoint(rota_y.text),strtoint(rota_z.text),3);
```

```
isometricview.canvas.Rectangle(0,0,130,130);
```

```
drawing_excellent(260,260,130,130,strtoint(v_x.text),strtoint(v_y.text),strtoint(v_z.text),4);
```

```
end;
```

```
procedure TCumpeerawut.zoom_downClick(Sender: TObject);
```

```
begin
```

```
zoomimage(strtoint(zoommm.text));
```

```
erase_click(sender);
```

```
drawing_excellent(800,800,400,400,strtoint(rota_x.text),strtoint(rota_y.text),strtoint(rota_z.text),3);
```

```
isometricview.canvas.Rectangle(0,0,130,130);
```

```
drawing_excellent(260,260,130,130,strtoint(v_x.text),strtoint(v_y.text),strtoint(v_z.text),4);
```

```
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TCumpeerawut.zoom_dClick(Sender: TObject);  
begin  
zoom_downClick(Sender);  
end;
```

```
procedure TCumpeerawut.zoom_uClick(Sender: TObject);  
begin  
zoom_upClick(Sender);  
end;
```

```
procedure TCumpeerawut.ActiveBtn1Down(Activate: TMouseOrKey);  
begin  
pagecontrol1.SelectNextPage(true);  
end;
```

```
procedure TCumpeerawut.ActiveBtn2Down(Activate: TMouseOrKey);  
begin  
pagecontrol1.SelectNextPage(true);  
end;
```

```
procedure TCumpeerawut.ActiveBtn3Down(Activate: TMouseOrKey);  
begin  
pagecontrol1.SelectNextPage(false);  
end;
```

```
procedure TCumpeerawut.ActiveBtn4Down(Activate: TMouseOrKey);  
begin  
pagecontrol1.SelectNextPage(true);  
end;
```

```
procedure TCumpeerawut.ActiveBtn5Down(Activate: TMouseOrKey);  
begin  
pagecontrol1.SelectNextPage(false);  
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า  
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure TCumpeerawut.panupClick(Sender: TObject);
begin
if shift_x.Checked=true then begin x_shift:=x_shift+strtoint(panshift.text);erase_click
(sender);draw_hahaclick(sender); end;
if shift_y.Checked=true then begin y_shift:=y_shift+strtoint(panshift.text);erase_click
(sender);draw_hahaclick(sender); end;
if shift_z.Checked=true then begin z_shift:=z_shift+strtoint(panshift.text);erase_click
(sender);draw_hahaclick(sender); end;
end;
```

```
procedure TCumpeerawut.pandownClick(Sender: TObject);
begin
if shift_x.Checked=true then begin x_shift:=x_shift-
strtoint(panshift.text);erase_click(sender);draw_hahaclick(sender); end;
if shift_y.Checked=true then begin y_shift:=y_shift-
strtoint(panshift.text);erase_click(sender);draw_hahaclick(sender); end;
if shift_z.Checked=true then begin z_shift:=z_shift-
strtoint(panshift.text);erase_click(sender);draw_hahaclick(sender); end;
end;

end.
```