



ปีการศึกษา 2531

ภาคการศึกษาที่ 1

MACHINE INTELLIGENCE

โดย

จรูญศักดิ์ วังวิวัฒน์

อาจารย์ที่ปรึกษา

รศ.ดร.โยธิน เปรมปรางค์รัตน์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ภาควิชาระบบควบคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ

023151

-8.คค.2532

ปีการศึกษา 2531

ภาควิชาระบบควบคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหาร ลาดกระบัง

เรื่อง MACHINE INTELLIGENCE

ผู้จัดทำ นายจรูญศักดิ์ วังวิวัฒน์ 281045

.....อาจารย์ที่ปรึกษา
(รศ.ดร.โยธิน เปรมปราชญ์)

MACHINE INTELLIGENCE

จรูญศักดิ์ วังวิวัฒน์

รศ.ดร.โยธิน เปรมปราชญ์ วิชาที่ปรึกษา

ภาคการศึกษาที่ 1 ปีการศึกษา 2531

บทคัดย่อ

มีวิธีการอยู่หลายวิธีที่จะสั่งให้เครื่องจักรอัตโนมัติทำงาน โครงงานนี้ได้ศึกษาถึงวิธีการต่าง ๆ โดยมุ่งเน้นไปที่การสอนให้เครื่องจักรทำงานด้วยวิธีการป้อนข้อมูลให้เครื่องจักรในลักษณะต่าง ๆ ได้แก่ การป้อนคำสั่งเป็นโปรแกรมคำสั่ง การป้อนคำสั่งในลักษณะรูปภาพ และการจับสอน

ในการสอนจำเป็นต้องมีการรับรู้การสอน การประมวลผล และการเก็บข้อมูลคำสั่งที่สอนในหน่วยความจำ ในโครงงานนี้ได้ศึกษาและทดลองเทคนิคการรับข้อมูล เทคนิคการประมวลผลข้อมูล และเทคนิคการเก็บข้อมูลลงหน่วยความจำ รวมทั้งได้ศึกษาความเป็นไปได้ และแนวทางที่จะนำเอาการสอน ไปประยุกต์ใช้ในอุตสาหกรรม

บทคัดย่อ

มีวิธีการอยู่หลายวิธีที่จะสั่งให้เครื่องจักรอัตโนมัติทำงาน โครงงานนี้ได้ศึกษาถึงวิธีการต่าง ๆ โดยมุ่งเน้นไปที่การสอนให้เครื่องจักรทำงานด้วยวิธีการป้อนข้อมูลให้เครื่องจักรในลักษณะต่าง ๆ ได้แก่ การป้อนคำสั่งเป็นโปรแกรมคำสั่ง การป้อนคำสั่งในลักษณะรูปภาพ และการจับสอน

ในการสอนจำเป็นต้องมีการรับรู้การสอน การประมวลผล และการเก็บข้อมูลคำสั่งที่สอนในหน่วยความจำ ในโครงงานนี้ได้ศึกษาและทดลองเทคนิคการรับข้อมูล เทคนิคการประมวลผลข้อมูล และเทคนิคการเก็บข้อมูลลงหน่วยความจำ รวมทั้งได้ศึกษาความเป็นไปได้ และแนวทางที่จะนำเอาการสอน ไปประยุกต์ใช้ในอุตสาหกรรม

MACHINE INTELLIGENCE

JAROONSAKDI WANGVIWATTANA

ASSC.PROF.DR.YOTHIN PREMPRANEERACH ADVISOR

PART 1 , 1988

ABSTRACT

THERE ARE MANY METHODS TO TEACH MACHINE TO WORK. IN THE TEACHING MODE, THE MOST IMPORTANT THING IS THE SENSOR PART. THE THREE METHODS OF SENSING ARE CONSIDERED IN THIS PROJECT. THE FIRST METHOD, THE LINEAR VARIABLE RESISTANCE IS USED TO BE A POSITION SENSOR. IN THIS CASE, IT MUST BE INCLUDED AN "ANALOG TO DIGITAL" CONVERTER TO INTERFACE TO COMPUTER. THE SECOND METHOD IS TO USE AN ENCODER TO SENSE THE POSITION AND DISPLACEMENT. AND THE THIRD METHOD, THE POSITION IS TRACKED BY DETECTING AND PROCESSING THE VOLTAGE SIGNAL GENERATED FROM THE COILS OF EACH PHASE INSIDE THE STEPPING MOTOR WHEN THE MOTOR IS ROTATED. THIS VOLTAGE SIGNAL IS VERY SMALL AND MIXED UP WITH VERY MUCH NOISE, IN THIS CASE, THE DETECTOR MAY GENERATE AN ERROR DISPLACEMENT OR AN INCORRECT POSITION.

THE "REAL-TIME" POSITIONS ARE LOADED TO THE MEMORY AND READY TO BE MOVED BACK LATER. THE PROBLEM IS THE METHOD OF SAVING THE REAL-TIME POSITION DATA INTO THE LIMITED MEMORY. THIS CAN BE DEVELOPED BY USING THE DATA STRUCTURE PROCESSING BY THE FAST AND INTELLIGENT SOFTWARE IN ASSEMBLY AS WELL AS PASCAL PROGRAMMING LANGUAGE ON THE IBM-PC/XT MICRO COMPUTER.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการสื่อสารเท่านั้น ไม่อนุญาตให้แก้ไขหรือดัดแปลงเอกสารโดยไม่ได้รับอนุญาตจากเจ้าของเอกสาร และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSTRACT

THERE ARE MANY METHODS TO TEACH MACHINE TO WORK. IN THE TEACHING MODE, THE MOST IMPORTANT THING IS THE SENSOR PART. THE THREE METHODS OF SENSING ARE CONSIDERED IN THIS PROJECT. THE FIRST METHOD, THE LINEAR VARIABLE RESISTANCE IS USED TO BE A POSITION SENSOR. IN THIS CASE, IT MUST BE INCLUDED AN "ANALOG TO DIGITAL" CONVERTER TO INTERFACE TO COMPUTER. THE SECOND METHOD IS TO USE AN ENCODER TO SENSE THE POSITION AND DISPLACEMENT. AND THE THIRD METHOD, THE POSITION IS TRACKED BY DETECTING AND PROCESSING THE VOLTAGE SIGNAL GENERATED FROM THE COILS OF EACH PHASE INSIDE THE STEPPING MOTOR WHEN THE MOTOR IS ROTATED. THIS VOLTAGE SIGNAL IS VERY SMALL AND MIXED UP WITH VERY MUCH NOISE, IN THIS CASE, THE DETECTOR MAY GENERATE AN ERROR DISPLACEMENT OR AN INCORRECT POSITION.

THE "REAL-TIME" POSITIONS ARE LOADED TO THE MEMORY AND READY TO BE MOVED BACK LATER. THE PROBLEM IS THE METHOD OF SAVING THE REAL-TIME POSITION DATA INTO THE LIMITED MEMORY. THIS CAN BE DEVELOPED BY USING THE DATA STRUCTURE PROCESSING BY THE FAST AND INTELLIGENT SOFTWARE IN ASSEMBLY AS WELL AS PASCAL PROGRAMMING LANGUAGE ON THE IBM-PC/XT MICRO COMPUTER.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

1. บทนำ	1
2. วิธีการควบคุมและการสอน	2
3. วิธีการรับรู้ข้อมูลแบบต่าง ๆ	5
3.1 การรับรู้ข้อมูลแบบแมนวล	5
3.2 การรับรู้ข้อมูลแบบโปรแกรมสั่งงาน	12
3.3 การรับรู้ข้อมูลแบบภาพ	19
3.3.1 วิธีการรับข้อมูลภาพจากหน่วยความจำ	20
3.3.2 วิธีการรับข้อมูลภาพแบบการกวาด	21
3.3.3 วิธีการรับข้อมูลภาพแบบการเคลื่อนที่ติดตามกรอบภาพ	28
3.3.4 วิธีการรับข้อมูลภาพแบบใช้กล้องถ่ายภาพ	44
3.4 การรับรู้ข้อมูลแบบจับสอน	45
3.4.1 การใช้ความต้านทานปรับค่าได้เป็นอุปกรณ์รับรู้	46
3.4.2 การใช้เอนโคเดอร์เป็นอุปกรณ์รับรู้	47
3.4.3 การใช้การตรวจจับสัญญาณในการรับรู้	49
4. วิธีการประมวลผลและเก็บข้อมูล	74
4.1 วิธีการรับข้อมูล	74
4.1.1 การรับข้อมูลแบบใช้เวลาเป็นโดเมน	75
4.1.2 การรับข้อมูลแบบใช้ตำแหน่งเป็นโดเมน	75
4.1.3 การรับข้อมูลแบบใช้การขจัดเป็นโดเมน	76
4.2 วิธีการประมวลผลข้อมูล	77
4.2.1 การขยายหรือย่อข้อมูลด้วยอัตราส่วนลงตัว	78
4.2.2 การขยายหรือย่อข้อมูลด้วยอัตราส่วนจำนวนจริง	78

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 รูปแบบการเก็บข้อมูลในหน่วยความจำ	89
4.3.1 การเก็บแบบโดเมนเวลา	89
4.3.2 การเก็บแบบโดเมนตำแหน่ง	90
4.3.3 การเก็บแบบใช้เทคนิคพิเศษ	97
4.4 การเรียกใช้ข้อมูลภายหลัง	104
4.5 โปรแกรมสมบูรณ์ในการทดลอง	105
5. การประยุกต์ใช้งาน	116
5.1 เครื่องเจาะแผ่นวงจรอัตโนมัติ	116
5.2 เครื่องตัดแผ่นเหล็กอัตโนมัติ	117
5.3 แขนกลในอุตสาหกรรม	118
6. บทสรุป	120
7. หนังสืออ้างอิง	122
8. กิตติกรรมประกาศ	126
9. ภาคผนวก	127

1 . บทนำ

ในปริณิพนธ์ฉบับนี้ จะกล่าวถึงรูปแบบและวิธีการต่าง ๆ เพื่อที่จะนำไปสู่การสอนให้เครื่องจักรสามารถทำงานได้เอง ในบทที่ 2 จะกล่าวถึงลักษณะการควบคุมเครื่องจักรอัตโนมัติด้วยวิธีการต่าง ๆ และเสนอรูปแบบการสอนให้เครื่องจักรอัตโนมัติทำงาน ในบทที่ 3 จะกล่าวถึง การรับรู้การสอนของเครื่องจักร โดยจะแยกอธิบายตามวิธีการรับรู้และชนิดของอุปกรณ์ที่ใช้ในการรับรู้ รวมถึงผลการทดลองและโปรแกรมที่ใช้กับคอมพิวเตอร์ที่ใช้ในการทดลอง

ในบทที่ 4 จะกล่าวถึงวิธีการประมวลผลข้อมูลที่ได้จากการรับรู้ และรวมถึงวิธีการและรูปแบบการเก็บข้อมูลไว้ใช้ในคราวต่อ ๆ ไป ในบทนี้จะกล่าวถึงเทคนิคการประมวลผลที่จำเป็น และมีทางเป็นไปได้ในการประยุกต์ใช้งานจริงในอุตสาหกรรม รวมทั้งวิธีการเก็บข้อมูลแบบต่าง ๆ และจะเสนอผลการทดลองที่ได้ทดลองจริงในรูปแบบและวิธีการต่าง ๆ

และในบทที่ 5 จะกล่าวถึงวิธีการและแนวทางในการนำเอาการสอนนี้ ไปประยุกต์ใช้งานกับอุตสาหกรรมต่อไป

2. วิธีการควบคุมและการสอน

ในระบบการทำงานของเครื่องจักรอัตโนมัติโดยทั่วไป การทำงานของเครื่องจักร จะใช้การควบคุมจากหน่วยควบคุมกลาง ซึ่งอาจเป็นเครื่องควบคุมโดยเฉพาะหรืออาจเป็นเครื่องไมโครคอมพิวเตอร์ ในการสั่งงานให้เครื่องจักรทำงานนั้น หน่วยควบคุมจะส่งสัญญาณซึ่งเครื่องจักรเข้าใจได้ไปให้เครื่องจักร เครื่องจักรก็รับคำสั่งไปปฏิบัติ ยกตัวอย่างง่าย ๆ คือ ให้สแตปมอเตอร์เป็นเครื่องจักร และหน่วยควบคุมก็เป็นเครื่องไมโครคอมพิวเตอร์ เครื่องไมโครคอมพิวเตอร์ก็จะส่งสัญญาณที่สแตปมอเตอร์เข้าใจได้ นั่นก็คือ สัญญาณหนึ่งหรือศูนย์ของแต่ละเฟสของสแตปมอเตอร์

ข้อมูลที่หน่วยควบคุมจะส่งไปให้เครื่องจักรทำงานนั้น ก็จะได้มาจากการประมวลจากหน่วยประมวลผล ซึ่งจะประมวลผลจากคำสั่งของผู้ใช้อีกทีหนึ่ง ในการสั่งงานของผู้ใช้ผ่านทางหน่วยควบคุมนี้ มีวิธีการอยู่หลายวิธีคือสามารถสั่งในรูปของแมนวล ตัวอย่างก็คือ รถตักดิน ซึ่งจะมีคนคอยควบคุมนั่งอยู่บนรถ คนควบคุมจะโยกคันโยกสั่งงานให้หน่วยควบคุมส่งสัญญาณให้ส่วนเคลื่อนที่ที่ควบคุมอยู่ทำงาน ในที่นี้สัญญาณที่สั่งงานจากหน่วยควบคุมก็คือสัญญาณไฮดรอลิค การสั่งงานของคนควบคุมผ่านหน่วยควบคุมลักษณะนี้ เป็นการควบคุมแบบแมนวล

ในการทำงานในอุตสาหกรรมบางอย่าง การควบคุมแบบแมนวลดังกล่าวข้างต้นนั้น ไม่ค่อยจะเหมาะสม เช่นการทำงานที่ต้องทำอยู่อย่างนั้นทุก ๆ ครั้ง ตัวอย่างคือ ในอุตสาหกรรมการผลิตแผ่นวงจร การเจาะรูที่แผ่นวงจรจำนวนมาก ๆ และเหมือน ๆ กันให้แม่นยำนั้น ใช้คนคอยควบคุมเครื่องจักรแบบแมนวลไม่เหมาะสมเลย เพราะงานที่ได้จะไม่มีความแม่นยำและดีพอ ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการที่เหมาะสมและใช้กันอยู่ในปัจจุบันก็คือ ใช้หลักการของโปรแกรม ซึ่งหน่วยควบคุมจะทำหน้าที่อ่านโปรแกรมซึ่งผู้ใช้ส่งมาเป็นลำดับ และทำงานตามคำสั่งนั้น ในกรณีเครื่องเจาะแผ่นวงจรนี้ โปรแกรมคำสั่งจะอยู่ในรูปคำสั่งเคลื่อนหัวเจาะให้เคลื่อนไปตามแกนตั้งหรือแกนนอนจนอยู่ ณ ตำแหน่งที่ต้องการ จากนั้นก็สั่งให้ทำการเจาะ และสั่งให้เคลื่อนที่ไปที่จุดที่จะเจาะต่อไปจนครบทุกจุด จากนั้นก็ทำการเจาะเหมือนเดิมกับแผ่นต่อ ๆ อย่างถูกต้องและแม่นยำเหมือน ๆ กันทุก ๆ แผ่น

วิธีการเขียนโปรแกรมสั่งงานดังกล่าวข้างต้นนั้น ทำได้ดีเฉพาะงานที่มีข้อมูลที่คงที่ เช่นการเจาะแผ่นวงจรเหมือน ๆ กันจำนวนมาก ๆ เพราะการเขียนโปรแกรมสั่งงานแต่ละครั้งนั้น ค่อนข้างจะยุ่งยากพอสมควร และภาษาของคำสั่งในโปรแกรมสั่งงานนั้น มีผู้เข้าใจน้อย ขอยกตัวอย่างเครื่องตัดแผ่นเหล็กเป็นรูปร่างต่าง ๆ ตามแบบ สมมติเรามีแบบที่จะตัดแผ่นเหล็กให้ได้ตามแบบนั้น ๆ ถ้าเป็นการเขียนโปรแกรมสั่งงานให้เคลื่อนที่หัวตัดแผ่นเหล็กไปตามกรอบแนวของแบบนี้ ๆ จะยุ่งยากมาก โดยเฉพาะเมื่อแบบที่ได้มามีลักษณะโค้งหรือไม่มีรูปแบบที่เป็นเรขาคณิต การเขียนโปรแกรมให้หัวตัดเคลื่อนที่ไปตัดแผ่นเหล็กให้ได้ตามแบบอย่างแม่นยำและถูกต้อง คงเป็นไปได้ยาก

ปัญหาดังกล่าวข้างต้นนั้น แก้ไขได้โดยเปลี่ยนวิธีการสั่งงานหน่วยควบคุมใหม่จากการที่จะเขียนโปรแกรมสั่งให้เคลื่อนที่ทีละขั้นตอน ก็ใช้วิธีให้มีการป้อนข้อมูลแบบใช้ข้อมูลภาพ นั่นคือใช้ภาพเป็นข้อมูลในการสั่งงาน วิธีการนี้มีอยู่หลายรูปแบบ เช่น การใช้หัวอ่านภาพทำการกวาดภาพแบบทั้งหมดแล้วส่งภาพข้อมูลที่ได้ให้หน่วยควบคุมทำการประมวลผลเปลี่ยนภาพเป็นข้อมูลที่ได้มาเป็นข้อมูลที่เครื่องจักรเข้าใจ และสั่งให้เครื่องจักรทำงานตามลำดับได้ หรืออีกวิธีคือการใช้หัวอ่าน อ่านภาพของแบบ โดยอ่านติดตามกรอบของแบบไปเรื่อย ๆ และสั่งให้เครื่องจักรเคลื่อนที่ไปในลักษณะเดียวกับที่อ่านติดตามกรอบของแบบนี้ ๆ หรือวิธีที่กำลังวิจัยอยู่ในต่างประเทศก็คือ วิธีใช้กล้องถ่ายภาพ หรือ กล้องวิดีโอทำการอ่านภาพแบบออกมา และประมวลผลเป็นสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ในงานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ข้อมูลให้เครื่องจักรเคลื่อนที่ไปตามแบบได้เช่นกัน

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

พิจารณาอุตสาหกรรมการพนันสิ ซึ่งต้องพนันสุพรรณหรือวัสดุต่าง ๆ เหมือน ๆ กันจำนวนมาก ๆ การใช้การเขียนโปรแกรมสั่งงานให้แขนกลที่พนันสิ เคลื่อนที่พนันสิไปตามโปรแกรมสั่งนั้นอาจจะเป็นไปได้ แต่เมื่อคำนึงถึงในทาง ปฏิบัติจริง ๆ ผู้ที่จะสามารถเขียนโปรแกรมสั่งงานหน่วยควบคุมได้นั้น มักจะพนัน สิไม่เป็น ส่วนผู้ที่พนันสิได้อย่างชำนาญนั้นก็ไม่มีความรู้พอที่จะมาเขียนโปรแกรม สั่งงานแขนกลได้ เมื่อเกิดปัญหาเช่นนี้ การนำเอาแขนกลมาประยุกต์ใช้งาน กับอุตสาหกรรมประเภทที่ต้องใช้ฝีมือและความชำนาญ เช่นการพนันสินี้ ย็น้อยอยู่ และการใช้งานแขนกลที่ทำงานตามโปรแกรมสั่งงานที่เขียนขึ้นนั้น ทำให้ได้งาน ที่ไม่ดีพอ อาจจะไม่วายเท่าผู้ชำนาญทำกับมือเอง ปัญหาดังกล่าวแก้ได้โดย แทนที่จะให้คนเขียนโปรแกรมสั่งงานตามลำดับ ก็ใช้วิธีให้ผู้ชำนาญงานทำการ จับแขนกลสอนให้แขนกลทำงาน และด้วยความชำนาญของพนักงานกับเครื่อง ประมวลผลที่มีประสิทธิภาพ ก็จะสามารถทำให้แขนกลสามารถทำงานได้ดีเท่า กับผู้ชำนาญงานเลยก็เดียวและยังสามารถผลิตชิ้นงานที่ประณีตได้จำนวนมาก ๆ โดยมีความปราณีตเท่า ๆ กันทุก ๆ ชิ้นได้ด้วย

วิธีการต่าง ๆ ดังกล่าวข้างต้นทั้งหมด จะแยกอธิบายและเสนอ ผลการทดลองในการทดลองจริง ไว้ในตอนต่อ ๆ ไป

3. วิธีการรับรู้ข้อมูลแบบต่าง ๆ

ในการสั่งงานเครื่องจักรอัตโนมัติ วิธีการที่หน่วยควบคุมจะรับรู้คำสั่งจากผู้ใช้และสามารถนำไปปฏิบัตินั้น มีอยู่หลายวิธีดังได้อธิบายไว้ในบทที่ 2 ซึ่งจะอธิบายอย่างละเอียดในบทนี้ และสามารถแบ่งการรับรู้ข้อมูลของหน่วยควบคุมออกเป็น 4 แบบคือ การรับรู้ข้อมูลแบบแมนวล การรับรู้ข้อมูลแบบโปรแกรมสั่งงาน การรับรู้ข้อมูลแบบภาพ และการรับรู้ข้อมูลแบบจับสั่น

3.1 การรับรู้ข้อมูลแบบแมนวล

เป็นวิธีการรับรู้ของหน่วยควบคุมซึ่งง่ายที่สุด และใช้งานกันมากที่สุด ความยุ่งยากของอุปกรณ์หรือวิธีการแทบจะไม่มีเลย หลักการของการรับรู้วิธีนี้ก็คือ ส่วนที่ติดต่อกับผู้ใช้ จะทำหน้าที่รับข้อมูลที่ผู้ใช้ส่งเข้ามา และจัดการแปลงข้อมูลเป็นข้อมูลที่เครื่องจักรเข้าใจได้ และส่งสัญญาณไปควบคุมเครื่องจักรให้ทำงานตามจุดประสงค์ของผู้ใช้ ตัวอย่าง เช่นการควบคุมรถตักดิน การขับรถยนต์ การใช้รถเครนในโรงงานอุตสาหกรรม และอื่น ๆ อีกมากมาย และข้อเสียของวิธีการนี้ก็มากเช่นกัน คือ เมื่อต้องทำงานในลักษณะเหมือนกันตลอดทุก ๆ ครั้ง คนควบคุมก็จะเกิดความล้า หรือ เบื่อ งานที่ออกมาก็จะไม่ดีเท่าที่ควร งานที่เหมาะสมที่จะใช้วิธีการนี้ก็คือ การที่ไม่ต้องการความแม่นยำในทุก ๆ ครั้ง และงานที่ไม่มีรูปแบบที่เหมือนกันเลยในแต่ละครั้ง เช่น การขุดดิน เป็นต้น

จะขอยกตัวอย่าง โปรแกรมการควบคุมเครื่องเอกซเรย์พลอตเตอร์ให้เคลื่อนที่หัวอ่านความเข้มของภาพไปตามพื้นที่ต่าง ๆ และแสดงผลค่าความเข้มของภาพที่อ่านได้บนจอภาพ โดยการใช้งานนั้น ผู้ใช้จะสั่งโดยการกดคีย์บอร์ดของเครื่องไมโครคอมพิวเตอร์โดยตรง เครื่องจะเคลื่อนที่ไปในทิศทางตามคำสั่งในทันที และอ่านค่าความเข้มภาพที่อ่านค่าได้ในตำแหน่งนั้นแสดงผลบนจอภาพด้วย

```
program project;
{$i graph.p}
const
  drive_byte : array[0..8] of byte =
    ( 0,32,160,128,144,16,80,64,96 );
  ( ( 00000000b 0 ,
    10000000b 128 ,
    11000000b 192 ,
    01000000b 64 ,
    01100000b 96 ,
    00100000b 32 ,
    00110000b 48 ,
    00010000b 16 ,
    10010000b 144 );
  motor_a_port : integer = 768 ;
  motor_b_port : integer = 769 ;
  opto_in_port : integer = 770 ;
  control_port : integer = 771 ;
  control_word : byte = 145 ;
```

```
var
```

```
  i,j,k,x,y : integer ;
```

```
  m,n : byte ; a : char ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin
  graphmode ;
  port[control_port] := control_word ;
  port[motor_a_port] := 1 ;
  port[motor_b_port] := 0 ;
  for x := 1 to 8 do begin
    port[motor_a_port] := drive_byte[x] ;
    port[motor_b_port] := drive_byte[x] ;
  end;
  x := 1 ;
  y := 1 ;
  i := 50 ;
  j := 50 ;
  repeat
    port[motor_a_port] := drive_byte[x] ;
    port[motor_b_port] := drive_byte[y] ;
    k := port[opto_in_port] ;
    m := trunc((k and { 10000000b } 128)/128) ;
    n := trunc((k and { 01000000b } 64)/64) ;
    gotoxy(1,1) ;
    writeln(x,' ',i,' ') ;
    writeln(y,' ',j,' ') ;
    write(m,' ',n) ;
    circle(i,j,3,3) ;
    fillshape(i,j,n,3) ;
    read(kbd,a) ;
    fillshape(i,j,0,3) ;
    circle(i,j,3,0) ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

case a of
'2' :
  begin
    j := j+1 ;
    y := y+1 ;
    if y > 8 then y := 1 ;
  end;
'8' :
  begin
    j := j-1 ;
    y := y-1 ;
    if y < 1 then y := 8 ;
  end;
'6' :
  begin
    i := i+1 ;
    x := x+1 ;
    if x > 8 then x := 1 ;
  end;
'4' :
  begin
    i := i-1 ;
    x := x-1 ;
    if x < 1 then x := 8 ;
  end;
'3' :
  begin

```



```

x := x+1 ;
y := y+1 ;
if x > 8 then x := 1 ;
if y > 8 then y := 1 ;
end;
'9' :
begin
  i := i+1 ;
  j := j-1 ;
  x := x+1 ;
  y := y-1 ;
  if x > 8 then x := 1 ;
  if y < 1 then y := 8 ;
end;
'7' :
begin
  i := i-1 ;
  j := j-1 ;
  x := x-1 ;
  y := y-1 ;
  if x < 1 then x := 8 ;
  if y < 1 then y := 8 ;
end;
'1' :
begin
  i := i-1 ;
  j := j+1 ;
  x := x-1 ;
  y := y+1 ;

```

```

    if x < 1 then x := 8 ;
    if y > 8 then y := 1 ;
end;
end;
until a = chr(27) ;
port[motor_a_port] := 1 ;
port[motor_b_port] := 0 ;
end.

```

อธิบายตัวโปรแกรมอย่างสังเขปตามลำดับดังนี้ บรรทัด const และ var เป็นการกำหนดค่าคงที่และพื้นที่ของตัวแปรต่าง ๆ ที่ใช้ในโปรแกรม ตัวโปรแกรมหลักเริ่มที่นี้

```

graphmode ;
port[control_port] := control_word ;
port[motor_a_port] := 1 ;
port[motor_b_port] := 0 ;

```

เป็นการกำหนดค่าเริ่มต้น

```

for x := 1 to 8 do begin
    port[motor_a_port] := drive_byte[x] ;
    port[motor_b_port] := drive_byte[x] ;
end;

```

เป็นการเคลื่อนที่สเตปมอเตอร์ให้อยู่ในตำแหน่งเริ่มต้นที่ถูกต้อง สอดคล้องกับตำแหน่งในโปรแกรม

```

x := 1 ; y := 1 ; i := 50 ; j := 50 ;

```

กำหนดตำแหน่งอ้างอิงเริ่มต้นในพื้นที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

repeat

port[motor_a_port] := drive_byte[x] ;

port[motor_b_port] := drive_byte[y] ;

เคลื่อนที่หัวอ่านภาพไปในทิศทางที่คำนวณได้

k := port[opto_in_port] ;

m := trunc((k and { 10000000b } 128)/128) ;

n := trunc((k and { 01000000b } 64)/64) ;

gotoxy(1,1) ; writeln(x,' ',i,' ') ;

writeln(y,' ',j,' ') ; write(m,' ',n) ;

circle(i,j,3,3) ; fillshape(i,j,n,3) ;

อ่านภาพและแสดงผลความเข้มของภาพที่ได้

read(kbd,a) ;

fillshape(i,j,0,3) ; circle(i,j,3,0) ;

รับคำสั่งจากทางคีย์บอร์ด และลบการแสดงผลของเก่าออกพร้อมที่แสดงผลข้อมูลใหม่บนจอภาพ

case a of

เป็นแปลคำสั่งและคำนวณการเคลื่อนที่ต่อไปของหัวอ่านภาพ โดยเชื่อว่าคีย์ที่รับเข้ามาเป็นตัวอะไร หมายถึงอะไร และต้องจัดการอย่างไรกับการกดคีย์นั้น ๆ ของผู้ใช้ ข้อมูลการเคลื่อนที่ ที่คำนวณได้ จะถูกนำไปสั่งเครื่องให้เคลื่อนที่อีกครั้งในการวนรอบของโปรแกรมครั้งต่อไป

until a = chr(27) ;

port[motor_a_port] := 1 ;

port[motor_b_port] := 0 ;

สุดท้าย วนกลับไปอ่านภาพและอ่านคีย์บอร์ดต่อ ๆ ไปเรื่อย ๆ จนมีการกดคีย์ยกเลิกการทำงาน โปรแกรมก็จะทำการเซตค่าสุดท้ายให้กับอุปกรณ์

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 การรับรู้ข้อมูลแบบโปรแกรมสั่งงาน

วิธีการนี้ หน่วยควบคุมจะต้องมีตัวประมวลผล ซึ่งอาจเป็นระบบไมโครโปรเซสเซอร์ หรือเป็นระบบไมโครคอมพิวเตอร์ ซึ่งจะทำหน้าที่รับโปรแกรมการสั่งงานจากผู้ใช้ และเก็บโปรแกรมสั่งงานดังกล่าวไว้ในหน่วยความจำของเครื่อง จากนั้น ก็สั่งให้เครื่องจักรทำงานตามโปรแกรมสั่งงานตามลำดับอย่างต่อเนื่อง และทำงานตามโปรแกรมสั่งงานเหมือน ๆ กัน ทุกชิ้นอย่างถูกต้องและแม่นยำ

ตัวอย่างการเขียนโปรแกรมสั่งงานในลักษณะนี้ก็คือ การเขียน
โปรแกรมให้เครื่องเจาะแผ่นวงจรทำงานตามสั่งดังนี้

ลำดับที่ 1 เคลื่อนหัวเจาะจากตำแหน่งอ้างอิงไปทางแกนตั้ง 10 ตำแหน่งในขณะเดียวกันเคลื่อนไปทางแกนนอน 5 ตำแหน่ง

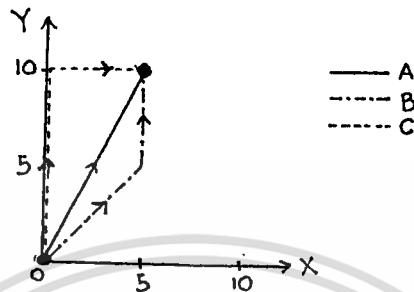
ลำดับที่ 2 เจาะลง

ลำดับที่ 3 ยกหัวเจาะขึ้น

ลำดับที่ 4 เคลื่อนหัวเจาะ . . .

หน่วยควบคุมก็จะทำการอ่านโปรแกรมจากหน่วยความจำ แล้วทำการประมวลผล เช่นในลำดับที่ 1 หน่วยควบคุมก็จะสั่งให้สเตปมอเตอร์ที่ขับเคลื่อนส่วนของแกนตั้ง เคลื่อนที่ไป 10 สเตป และสั่งให้ส่วนแกนนอนเคลื่อนที่ไป 5 สเตป ซึ่งในโปรแกรมอาจจะมีเทคนิคแตกต่างกัน เช่น อาจจะทำให้เคลื่อนไปทางแกนตั้งก่อนจนครบ 10 สเตปแล้วค่อยเคลื่อนที่ไปตามแกนนอน หรืออาจจะทำให้เคลื่อนที่ไปตามแกนตั้งและแกนนอนไปพร้อม ๆ กัน โดยอัตราส่วนการเคลื่อนที่เป็น 2 ต่อ 1 หรืออาจจะทำให้เคลื่อนที่ไปทั้งทางแกนตั้งและทางแกนนอนพร้อม ๆ กัน 5 สเตป แล้วค่อยเคลื่อนที่ไปทางแกนตั้งที่เหลืออีก 5 สเตป ดังรูปที่ 3.1

เอกสารนี้เป็นเอกสารพื้นฐานไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.1

การเคลื่อนที่ของหัวเจาะแผ่นวงจรตามโปรแกรมสั่งงาน

เหล่านี้ เป็นรายละเอียดปลีกย่อยของเทคนิคการเขียนโปรแกรมต่อไปจะยกตัวอย่าง โปรแกรมที่สั่งให้เครื่องเอกซเรย์พลอตเตอร์ซึ่งใช้สแตมป์มอเตอร์ 4 เฟส 2 ตัว ทำการวาดรูปเกียร์

```
program project8;
const
  drive_byte : array[0..8] of byte =
    ( 0,32,160,128,144,16,80,64,96 );
  pp : array[1..9] of integer =
    ( 262,294,330,349,392,440,494,523,548 );
  motor_a_port : integer = 768 ;
  motor_b_port : integer = 769 ;
  opto_in_port : integer = 770 ;
  control_port : integer = 771 ;
  control_word : byte = 145 ;
var
  directionx : array[0..6400] of 0..2;
```

```
  directiony : array[0..6400] of 0..2;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
i,j,k,x,y,a,b,sn,sn1 : integer ;
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

r,s,m,u,v : real ;
p,q : byte ;
c : char ;
song : string[100] ;
procedure music ;
begin
if sn1=15 then begin
  sn1:=1;
  c:=copy(song,sn,1);
  if c='$' then begin
    nosound;
    halt;
    sn:=1;
    c:=copy(song,sn,1);
  end;
  case c of
    'c' : sound(pp[1]);
    'd' : sound(pp[2]);
    'e' : sound(pp[3]);
    'f' : sound(pp[4]);
    'g' : sound(pp[5]);
    'a' : sound(pp[6]);
    'b' : sound(pp[7]);
    'C' : sound(pp[8]);
    'D' : sound(pp[9]);
  end;
  sn:=sn+1;

```

```

end

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

sn1:=sn1+1;
end;
procedure set_motor;
begin
    port[motor_a_port] := 1 ;
    port[motor_b_port] := 0 ;
end;
procedure incr(var x:integer);
begin
    x:=x+1;
    if x>8 then x:=1;
end;
procedure decr(var x:integer);
begin
    x:=x-1;
    if x<1 then x:=8;
end;
procedure pt(x,y:integer);
begin
    plot(159+x,99+y,15);
end;
procedure drctn;
begin
    directionx[j]:=(x-a)+1;
    directiony[j]:=(y-b)+1;
    j:=j+1;
end;

```

```

procedure jjjj;
begin
  pt(x,y);
  drctn;
  a:=x; b:=y;
end;
begin
  nosound;
  sn:=1; sn1:=1;
  song:='ggggcdefggggccccaagfgabCCCCcccc
        fffffgfedeeefedcdddcdecddddddddd$';
  read(kbd,c);
  port[control_port] := control_word ;
  set_motor;
  graphmode;
  pt(0,0);
  a:=0;b:=0;j:=0;
  r:=2*pi/624; s:=2*pi/432;
  for i:=0 to 7 do begin
    for k:=0 to 38 do begin
      m:=(78*i+k)*r;
      x:=trunc(100*cos(m)); y:=trunc(100*sin(m));
      jjjj;
    end;
    u:=cos(m); v:=sin(m);
    for k:=100 downto 70 do begin
      x:=trunc(k*u); y:=trunc(k*v);
      jjjj;
    end;
  end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

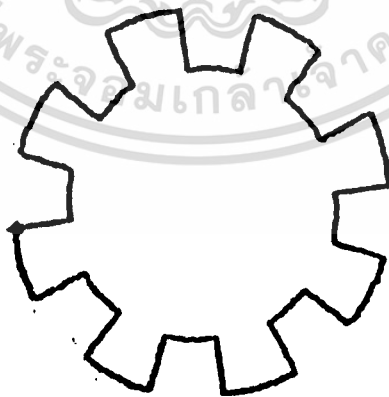
```

u:=pi/8;
for k:=0 to 26 do begin
    m:=(54*i+k)*s+u;
    x:=trunc(70*cos(m)); y:=trunc(70*sin(m));
    jjjj;
end;
u:=cos(m); v:=sin(m);
for k:=70 to 100 do begin
    x:=trunc(k*u); y:=trunc(k*v);
    jjjj;
end;
end;
read(kbd,c);
repeat
for x := 8 downto 1 do begin
    port[motor_a_port] := drive_byte[x] ;
    port[motor_b_port] := drive_byte[x] ;
end;
y:=x;
for i:=0 to j do begin
    if keypressed then read(kbd,c); delay(1);
    case directionx[i] of
        2:incr(x);
        0:decr(x);
    end;
    case directiony[i] of
        2:incr(y);
        0:decr(y);
    end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
port[motor_a_port] := drive_byte[x] ;  
port[motor_b_port] := drive_byte[y] ;  
delay(7);  
if (directionx[i]<>1) or (directiony[i]<>1) then music;  
end;  
nosound;  
set_motor;  
read(kbd,c);  
until c='#' ;  
end.
```



รูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ผลของการวาดภาพเกียร์ของโปรแกรม
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ลักษณะการทำงานของโปรแกรมโดยสังเขปมีดังนี้ เริ่มจากโปรแกรมทำการคำนวณลักษณะของรูปเกียร์ตามวิชาเรขาคณิต โดยได้ข้อมูลของเส้นผ่านศูนย์กลางทั้งภายในและภายนอก จำนวนซี่เกียร์ เมื่อคำนวณภาพได้ก็จะแสดงภาพนั้นออกทางจอภาพ จากนั้นจะคำนวณครั้งที่ 2 โดยคำนวณทิศทางที่จะบังคับให้ตัวเอกซ์วายพลอตเตอร์เคลื่อนหัวปากกาไป เป็นลักษณะข้อมูลแบบ " ทิศทาง " และจะจำข้อมูลที่ได้ในหน่วยความจำ จากนั้น ก็จะวาดรูปเกียร์ออกมา ตามจำนวนครั้งตามต้องการได้ โปรแกรมนี้มีส่วนของดนตรีประกอบด้วย ซึ่งไม่ใช่ส่วนสำคัญและไม่เกี่ยวข้องกับการวาดเกียร์

ลักษณะสำคัญของโปรแกรมหกข้อก็คือ การนำข้อมูลที่เก็บในหน่วยความจำซึ่งเป็นโปรแกรมที่ผู้ใช้สั่งงานตามลำดับ นำมาทำงานได้หลายครั้ง เหมือน ๆ กันได้ ซึ่งเป็นลักษณะการรับรู้ข้อมูลของหน่วยควบคุมโดยรับรู้โปรแกรมสั่งงานตามขั้นตอน

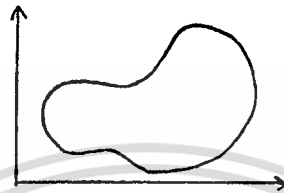
3.3 การรับรู้ข้อมูลแบบภาพ

ดังที่กล่าวไว้ในบทที่ 2 การรับรู้ข้อมูลแบบภาพเป็นการรับรู้ข้อมูลที่มีความยุ่งยากมาก แต่มีประโยชน์ เป็นการทำให้เครื่องจักรสามารถทำงานได้โดยไม่ต้องเขียนโปรแกรมสั่งงานเป็นขั้น ๆ ซึ่งทำได้ยากในทางปฏิบัติโดยเฉพาะการสั่งงานให้เกิดการเคลื่อนที่ที่ไปเป็นไปตามรูปแบบทางเรขาคณิต เช่น การเคลื่อนที่แบบโค้งและไร้รูปแบบดังรูปที่ 3.3 ถ้าใช้วิธีการเขียนโปรแกรมสั่งงาน ผู้เขียนจะเกิดความยุ่งยากมากดังรูปที่ 3.4

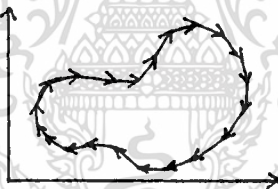
วิธีการที่จะแก้ปัญหายุ่งยากในการเขียนโปรแกรมสั่งงาน ก็คือการใช้วิธีให้เครื่องอ่านภาพออกมาจากแบบและประมวลผลภาพเอง ทำให้ลดความยุ่งยากในการเขียนโปรแกรมสั่งงาน วิธีการอ่านภาพดังกล่าว มีวิธีการรับรู้ภาพและเทคนิคโปรแกรมแตกต่างกันไป ดังจะได้อธิบายต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3
รูปที่ไม่เป็นรูปแบบทางเรขาคณิต



รูปที่ 3.4
การเขียนโปรแกรมสั่งงานให้เครื่องจักรเคลื่อนที่ไปตามรูปแบบในรูป 3.3

3.3.1 วิธีการรับข้อมูลภาพจากหน่วยความจำ

เป็นวิธีที่ง่ายที่สุดในการรับข้อมูลภาพ เพราะภาพจากหน่วยความจำไม่ต้องผ่านกระบวนการทางอาร์คแวร์เลย ภาพที่ได้จากหน่วยความจำสามารถนำออกมาใช้งานได้เลย ตัวอย่างการรับข้อมูลภาพดังกล่าว ได้แสดงไว้ในหัวข้อที่ 3.2 เป็นโปรแกรมวาดรูปเกียร์ ซึ่งรูปเกียร์ที่คำนวณได้จะถูกนำไปเก็บไว้ในหน่วยความจำ ซึ่งอาจเป็นแผ่นดิสก์ หรือ หน่วยความจำแม่เหล็ก และสามารถอ่านข้อมูลภาพเกียร์จากหน่วยความจำ สั่งให้เครื่องพลอตเตอร์วาดรูปออกมาได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการนี้ จะมีผลก็ต่อเมื่อโปรแกรมสามารถเข้าใจภาพที่เก็บในหน่วยความจำด้วย ถ้าโปรแกรมไม่สามารถเข้าใจภาพได้ก็ไม่มีประโยชน์ หรืออีกนัยหนึ่ง การเก็บภาพลงในหน่วยความจำ จะต้องมียุทธวิธีที่เป็นมาตรฐาน หรือที่ทำให้โปรแกรมสามารถเข้าใจภาพได้ง่ายด้วย

3.3.2 วิธีการรับข้อมูลภาพแบบการกวาด

ลักษณะของวิธีการนี้ก็คือ นำเอาภาพต้นแบบที่ต้องการจะให้มีการรับรู้ มาทำการกวาดเก็บข้อมูลทั้งหมดของภาพลงในหน่วยความจำ และประมวลผลที่ได้ให้อยู่ในรูปแบบที่เครื่องจักรสามารถเข้าใจได้ เช่น การหาเส้นขอบ การหาทิศทางของเส้น เป็นต้น กระบวนการดังกล่าวจะต้องใช้การประมวลผลซึ่งค่อนข้างซับซ้อน ต้องใช้คอมพิวเตอร์ในการจัดการประมวลผลภาพ

ตัวอย่างโปรแกรมการรับข้อมูลภาพแบบกวาดแสดงไว้ดังนี้

```
program project4 ;
const
  drive_byte : array[0..8] of byte =
    ( 0,32,160,128,144,16,80,64,96 );
  p : array[1..9] of integer =
    ( 262,294,330,349,392,440,548,523,465 );
  motor_a_port : integer = 768 ;
  motor_b_port : integer = 769 ;
  opto_in_port : integer = 770 ;
  control_port : integer = 771 ;
  control_word : byte = 145 ;
```

```

var
  i,j,k,x,y,sn : integer ;
  m,n,s : byte ;
  a,c : char ;
  song : string[100] ;
{$i graph.p}
procedure music ;
begin
  ct:=copy(song,sn,1);
  if c='*' then begin
    sn:=1;
    c:=copy(song,sn,1);
  end;
  case c of
    'c' : sound(p[1]);
    'd' : sound(p[2]);
    'e' : sound(p[3]);
    'f' : sound(p[4]);
    'g' : sound(p[5]);
    'a' : sound(p[6]);
    'D' : sound(p[7]);
    'C' : sound(p[8]);
    'x' : sound(p[9]);
  end;
  sn:=sn+1;
end;
begin
  nosound;

```

```

song := 'ccdffdeedcccfhgadcggggg
        CCeggdeefcccgggxdeffff*';

sn := 1;
hires;
for i:=220 to 419 do begin
    plot(i,50,1);
    plot(i,149,1);
end;
for i:=50 to 149 do begin
    plot(220,i,1);
    plot(419,i,1);
end;
for i:=440 to 639 do begin
    plot(i,50,1);
    plot(i,149,1);
end;
for i:=50 to 149 do begin
    plot(440,i,1);
    plot(639,i,1);
end;

port[control_port] := control_word ;
port[motor_a_port] := 0 ;
port[motor_b_port] := 0 ;
for x := 1 to 8 do begin
    port[motor_a_port] := drive_byte[x] ;
    port[motor_b_port] := drive_byte[x] ;
end;

x := 1 ; y := 1 ;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 for j:=26 to 74 do
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin
  port[motor_b_port] := drive_byte[y] ;
  delay(5);
  y:=y -1; if y<1 then y:=8;
  y:=y -1; if y<1 then y:=8;
  music ;
  for i:=225 to 409 do begin
    n:=0; s:=0;
    port[motor_a_port] := drive_byte[x] ;
    x:=x+1; if x>8 then x:=1;
    if keypressed then read(kbd,a);
    delay(5);
    m := 192 and port[opto_in_port] ;
    if m>n then begin
      n := m ; s := 1 ;
    end ;
    if m<n then begin
      n := m ; s := 0 ;
    end ;
    plot(i,j*2-1,s);
  end;
  port[motor_b_port] := drive_byte[y] ;
  delay(5);
  y:=y -1; if y<1 then y:=8;
  y:=y -1; if y<1 then y:=8;
  music ;
  for i:=415 downto 231 do begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 port[motor_a_port] := drive_byte[x] ;
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

x:=x-1; if x<1 then x:=8;
if keypressed then read(kbd,a);
delay(5);
m := 192 and port[opto_in_port] ;
if m>n then begin
  n := m ; s := 1 ;
end ;
if m<n then begin
  n := m ; s := 0 ;
end ;
plot(i,j*2,s);
end;
end;
port[motor_a_port] := 1 ;
port[motor_b_port] := 0 ;
nosound;
read(kbd,c);
for x:=220 to 419 do begin
  i:=0;
  for y:=50 to 149 do begin
    j:=getdotcolor(x,y);
    if j<>i then begin
      i:=j;
      plot(x+220,y,1);
    end;
  end;
end;
for y:=50 to 149 do begin
  i:=0;

```

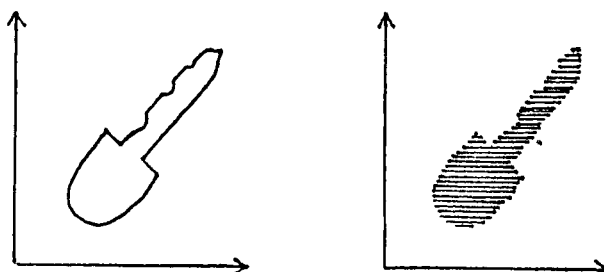
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

for x:=440 to 639 do begin
  j:=getdotcolor(x,y);
  if j<>i then begin
    i:=j;
    plot(x+220,y,1);
  end;
end;
end;
end.

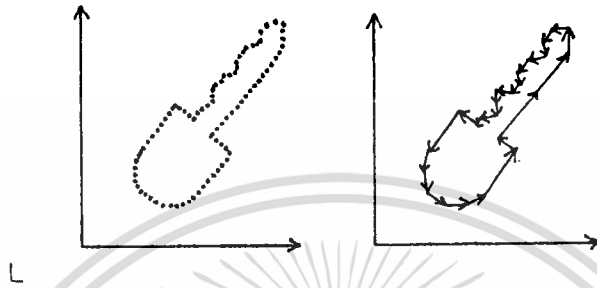
```

อธิบายตัวโปรแกรมลงไปได้ว่า เริ่มจากการเซตค่าเริ่มต้นของตัวแปรและอุปกรณ์ภายนอก จากนั้น รอรับคีย์เริ่มต้น เมื่อผู้ใช้เลื่อนภาพต้นแบบให้อยู่ในตำแหน่งที่ต้องการ โปรแกรมจะสั่งให้เครื่องเอกซเรย์เคลื่อนที่หัวอ่านภาพในลักษณะการกวาดภาพทีละแถว กวาดไปและกวาดกลับ ข้อมูลที่ได้จะเก็บลงในหน่วยความจำและแสดงออกทางจอภาพขณะกำลังกวาด เมื่อภาพได้ครบตามรูปต้นแบบ โปรแกรมจะทำการหาขอบของภาพ และต่อไปก็จะหาทิศทางของกรอบของภาพ ข้อมูลสุดท้ายคือทิศทางของการเคลื่อนที่นั้น จะเก็บไว้ในหน่วยความจำ และสามารถนำออกมาสั่งงานให้เครื่องพลอตเตอร์ วาดรูปกลับมาเหมือนภาพเดิมได้ในทันที รูปแสดงการทำงานของโปรแกรมแสดงไว้ในรูปที่ 3.5 และรูปที่ 3.6



รูปที่ 3.5

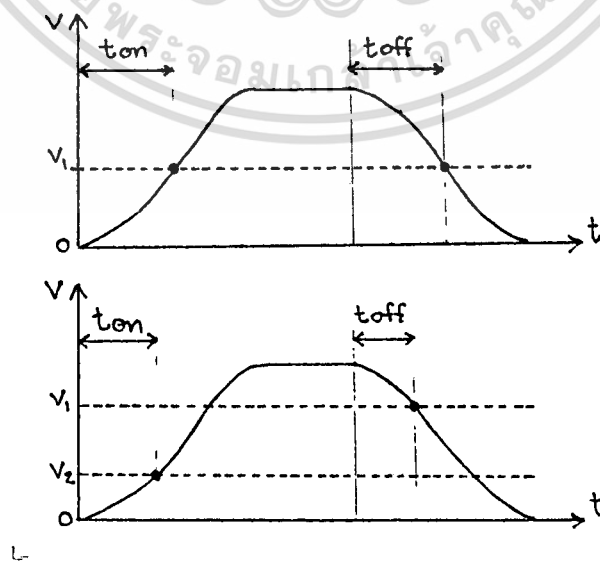
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า การกวาดภาพและข้อมูลที่ได้
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6

การหาเส้นขอบและหาทิศทางของเส้นขอบ

วิธีการหาเส้นขอบนั้น ในโปรแกรมตัวอย่างดังกล่าว ได้ใช้วิธีพิจารณาความแตกต่างของความเข้มของวัตถุ คือ สีขาวและดำ เมื่อพบส่วนที่มีสีขาวและสีดำติดกัน จุดนั้นจะเป็นขอบของภาพ ส่วนวิธีการหาทิศทาง การเคลื่อนที่ของเส้นขอบ เพื่อจะนำข้อมูลไปใช้ตอนวาดรูปกลับออกมานั้น จะอธิบายไว้ในหัวข้อที่ 3.3.3



รูปที่ 3.7

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
วิธีการลดการหน่วงสัญญาณของอุปกรณ์อ่านภาพ
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดลองจริง ได้พบปัญหาในการใช้วิธีการนี้ก็คือความเร็วในการกวาดภาพยังจำกัดอยู่ที่ตัวขับเคลื่อน ในการทดลองใช้สเตปมอเตอร์จึงมีความเร็วที่จำกัดอยู่ อีกข้อหนึ่งก็คือความไวของหัวอ่านภาพ ปกติหัวอ่านภาพจะมีการหน่วงสัญญาณมาก การแก้ไขซึ่งได้ทดลองเทคนิคการลดการหน่วงของสัญญาณโดยใช้การเปรียบเทียบสัญญาณแบบสามระดับ แทนการเปรียบเทียบแบบเดิมคือแบบสองระดับ แสดงไว้ดังรูปที่ 3.7

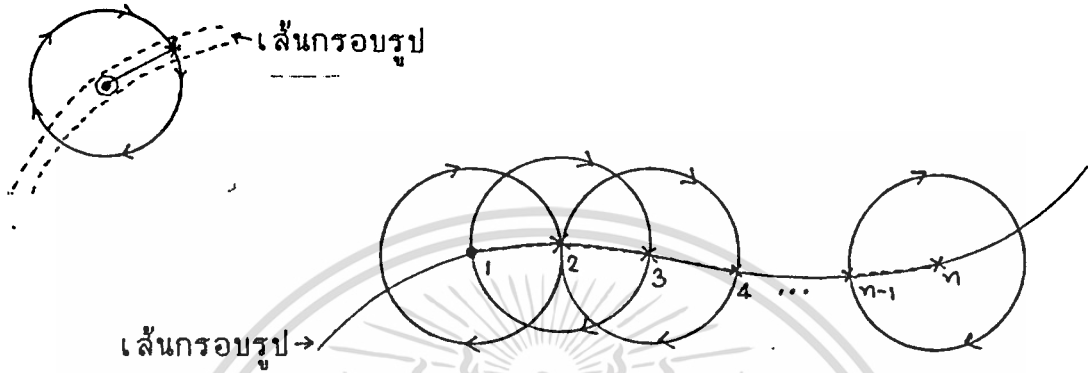
อธิบายขณะที่หัวอ่านกำลังอ่านจุดขาวอยู่ก่อน และเลื่อนไปอ่านจุดดำ หัวอ่านจะค่อย ๆ รับรู้ที่กำลังเปลี่ยนแปลงจากสีขาวเป็นสีดำ ช่วงที่กำลังรับรู้สีขาวกำลังเปลี่ยนเป็นสีดำ แต่ยังไม่ดำก็ให้ถือว่าตรงจุดเปลี่ยนแปลงนั้นเป็นจุดสีดำเลย โดยไม่ต้องรอให้เป็นสีดำจริง ๆ ดังนั้น การรับรู้การเปลี่ยนแปลงของหัวอ่านก็จะไวขึ้น แต่ความแม่นยำย่อมน้อยลงไป

3.3.3 วิธีการรับข้อมูลภาพแบบการเคลื่อนที่ติดตามกรอบภาพ

หลักการก็คือ การให้ข้อมูลแก่เครื่องโดยให้เครื่องหากรอบภาพเอง โดยเริ่มจากเครื่องจะหาจุดเริ่มอ้างอิงของกรอบภาพต้นแบบ แล้วทำการติดตามกรอบภาพต้นแบบ หรือติดตามลายเส้นกรอบภาพไปจนได้ภาพต้นแบบครบ เป็นวิธีการรับข้อมูลภาพอีกวิธีหนึ่ง โดยข้อมูลเข้ามา อยู่ในรูปแบบของทิศทางการเคลื่อนที่ได้เลยโดยไม่ต้องประมวลผลภาพ รูปแบบในการติดตามกรอบของภาพมีอยู่หลายวิธีการด้วยกัน ซึ่งแต่ละวิธีการมีข้อดีข้อเสียแตกต่างกันไป ดังจะแยกอธิบายไว้ดังนี้

3.3.3.1. ติดตามลายเส้นโดยรู้ทิศทางที่จะไปอย่างแน่นอนล่วงหน้า อาศัยอุปกรณ์ทางแมคคานิกส์ที่ค่อนข้างจะซับซ้อน โดยการพยายามหาทิศทางที่จะไปด้วยการใช้อุปกรณ์ตรวจสอบ หมุนรอบ ๆ จุดที่อ้างอิง ซึ่งจะพบทางออกจากวงกลมที่ทดสอบอยู่สองทางด้วยกัน คือ ทางแรก เป็นทางที่เข้ามาในครั้งแรก ส่วนอีกทางก็เป็นทางที่จะไปต่อไป ดังรูปที่ 3.8 ทำอย่างนี้ต่อไปอย่างต่อเนื่อง ก็จะสามารถหาทิศทางของลายเส้นได้

ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งยังมีเหตุตบแต่งเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8

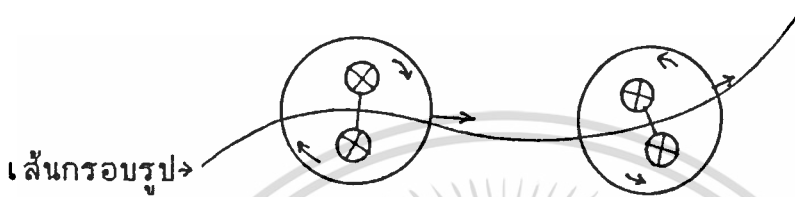
การหาทิศทางกรอภาพโดยหาทิศทางล่วงหน้า

วิธีนี้เป็นวิธีที่ได้ผลค่อนข้างจะแน่นอน เพราะมีการตรวจสอบถึงสองครั้ง คือ ครั้งแรกทดสอบทางที่จะไปเป็นการล่วงหน้า และเมื่อพอรู้ว่าจะไปในทิศทางใดแล้ว ก็จะเคลื่อนที่ไป ณ จุดนั้น และทำการทดสอบอีกครั้งหนึ่ง ถ้าเป็นลายเส้นจริง ๆ แล้วก็จะหาทิศทางที่จะไปต่อไป เป็นการแน่นอนมาก

แต่วิธีการนี้ค่อนข้างจะลงทุนมาก และมีความซับซ้อนในเรื่องหัวอ่านลายเส้น เพราะต้องออกแบบให้มีลักษณะที่ต้องหมุนรอบได้ และกวาดหาทิศทางที่จะไปได้อย่างรวดเร็วและแน่นอน ความซับซ้อนในเรื่องหัวอ่านนี้เองเป็นตัวทำให้ความเชื่อถือได้ของระบบลดลงไป เพราะมีความน่าจะเป็นในการผิดพลาดมากนั่นเอง

3.3.3.2. ติดตามลายเส้น โดยพยายามให้ลายเส้นอยู่ระหว่างตัวตรวจจับลายเส้นสองตัว วิธีการนี้มีหลักการง่าย ๆ ว่า พยายามให้ตัวตรวจจับสองตัวที่วางคู่กันนั้น ติดตามลายเส้นไปเรื่อย ๆ เมื่อตัวตรวจจับตัวใดตัวหนึ่งพบลายเส้น ก็จะหมายความว่าลายเส้นกำลังจะหลุดจากการติดตาม ตัวตรวจจับก็จะปรับตัวเอง ให้หมุนทิศทางไปในทางที่สามารถติดตามลายเส้น

ได้ต่อไป ดังรูปที่ 3.9
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

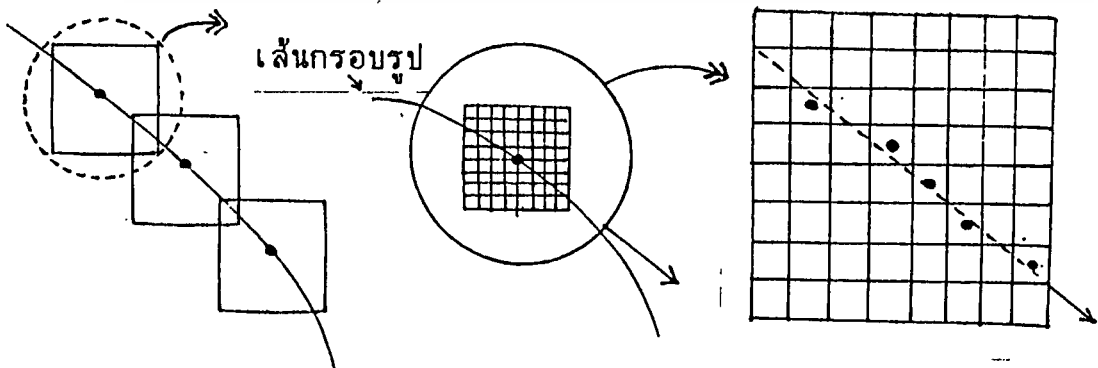


รูปที่ 3.9

การพยายามให้กรอบภาพอยู่ระหว่างตัวตรวจจับภาพ

วิธีการนี้ ก็เช่นเดียวกับวิธีการแรกคือ มีความยุ่งยากทางอุปกรณ์ ตัวตรวจจับ ยังต้องมีการหมุนหัวอ่านด้วย ยิ่งทำให้ความน่าจะเป็นที่จะหลุดจากการติดตามย่อมมีมากขึ้น การทดสอบก็กระทำเพียงครั้งเดียวเท่านั้น จึงมีความเชื่อถือได้ค่อนข้างต่ำ

3.3.3.3. ติดตามลายเส้นแบบทดสอบพื้นที่บางส่วนที่สนใจ

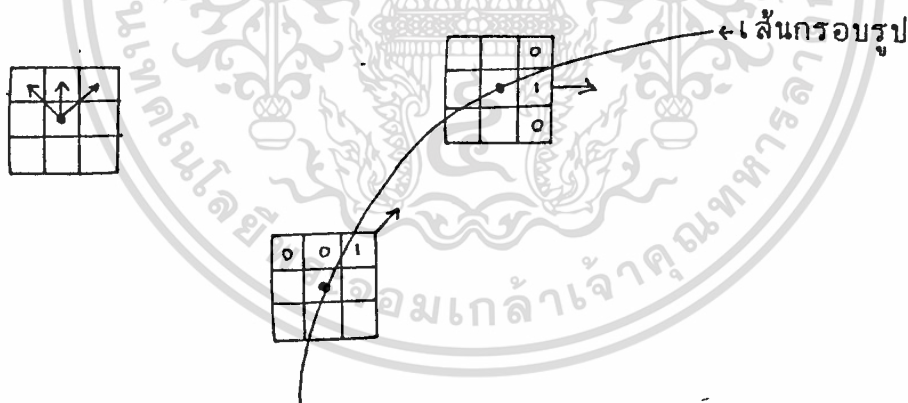


รูปที่ 3.10

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
พิจารณาพื้นที่ที่น่าสนใจ
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานจะเริ่มจากหาทิศทางที่จะไปล่วงหน้าให้ได้ก่อน โดยการตรวจสอบพื้นที่บริเวณนั้นพร้อม ๆ กัน ด้วยหัวอ่านที่เป็นแบบเมตริกซ์ ซึ่งมีจำหน่ายตั้งแต่ 4×4 จุดไปจนถึง 1024×1024 จุด ซึ่งจะตรวจสอบทีเดียวพร้อมกันหมด หาทิศทางที่จะเคลื่อนที่ต่อไปและตรวจสอบจุดที่ตั้งอยู่ขณะนั้นว่ามีลายเส้นหรือไม่ วิธีการนี้ให้ความเชื่อถือได้สูงมาก และรวดเร็วในการตัดสินใจหาทางไปล่วงหน้า รวมทั้งไม่มีความยุ่งยากมากนักในตัวอุปกรณ์ตรวจจับ วิธีการนี้ แสดงไว้ในรูปที่ 3.10

3.3.3.4 ติดตามลายเส้นแบบใช้หัวอ่านหัวเดียว ทำการหาทิศทางของลายเส้นที่จะติดตามต่อไป ด้วยการเคลื่อนหัวอ่านไปล่วงหน้าและตรวจสอบหาลายเส้น เมื่อเจอก็จะสรุปว่าทิศทางต่อไปต้องติดตามไปอย่างไร แต่ถ้ายังไม่เจอก็จะต้องพยายามเคลื่อนหัวอ่านให้หาทิศทางของลายเส้นให้เจอ ดังรูปที่ 3.11



รูปที่ 3.11

การหาทิศทางกรอบภาพโดยใช้หัวอ่านหัวเดียว

จากการทดลองใช้วิธีการดังกล่าว พบว่า การเคลื่อนที่หัวอ่านไปอ่านนั้น จะมีโอกาสความน่าจะเป็นที่จะมีทิศทางของการติดตามลายเส้นในทิศทางเดิมมากที่สุด ส่วนโอกาสความน่าจะเป็นที่จะมีทิศทางของการติดตามลายเส้นเบี่ยงเบนไปในทิศทางตรงกันข้ามกับทิศทางที่เบี่ยงเบนไปในครั้งก่อน จะมี

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการที่มีหัวอ่านเพียงหัวเดียวเท่านั้น การตรวจสอบก็จะทำได้เพียงทิศทิศทาง หากไม่เจอบิตทิศทางของลายเส้น ก็จะต้องหาไปเรื่อย ๆ ทำให้สิ้นเปลืองเวลาการหาทิศทางมาก แต่เมื่อเราใช้หลักสถิติดังกล่าวข้างต้น นำมาใช้ประกอบการตัดสินใจว่า จะตรวจสอบลายเส้นในทิศทางใดก่อนจึงจะเหมาะสม และสามารถจะเจอบิตทิศทางของลายเส้นกรอบภาพได้รวดเร็วกว่ากัน

ในโปรแกรมการหาลายเส้นด้วยวิธีนี้ จะทำการหาทางเดินของลายเส้นของรูปต้นแบบที่มีลักษณะเป็นชิ้นมีกรอบรอบ โดยจะต้องอ่านแบบแปลนการตัดเหล็กแผ่น ซึ่งเขียนเป็นรูปต้นแบบในกระดาษ เครื่องจะต้องทำการอ่านแบบแปลนนั้นเก็บไว้ในหน่วยความจำก่อน จากนั้น เมื่อต้องการตัดเหล็กแผ่นให้ได้รูปตามแบบแปลนก็ทำได้โดยสั่งให้โปรแกรมวาดภาพเดิมในหน่วยความจำออกมาซึ่งสามารถ ที่จะย่อขนาดหรือ ขยายขนาด หรือ หมุนภาพได้ตามต้องการสามารถตัดเหล็กได้ขนาดและรูปร่างตามต้องการได้ โปรแกรมได้แสดงไว้ดังนี้

```
program project7;
{-----}
{ Line tracker }
{-----}

const
  drive_byte : array[0..8] of byte =
  (
    0,32,160,128,144,16,80,64,96
  );
  motor_a_port : integer = 768;
  motor_b_port : integer = 769;
  opto_in_port : integer = 770;
  control_port : integer = 771;
  control_word : byte = 145;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

ypos:=0;
direction[0]:=3;
last_turn:=true;
dnum:=1;
end;

```

```

{-----}

```

```

procedure move;

```

```

begin

```

```

  i:=x;

```

```

  j:=y;

```

```

  case dnum of

```

```

    1:

```

```

      begin

```

```

        decr(i);

```

```

        incr(j);

```

```

      end;

```

```

    2:

```

```

      incr(j);

```

```

    3:

```

```

      begin

```

```

        incr(i);

```

```

        incr(j);

```

```

      end;

```

```

    4:

```

```

      incr(i);

```

```

    5:

```

```

      begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    incr(i);
    decr(j);
end;
6:
    decr(j);
7:
    begin
        decr(i);
        decr(j);
    end;
8:
    decr(i);
end;
port[motor_a_port]:=drive_byte[i];
port[motor_b_port]:=drive_byte[j];
end;

procedure find_direction_0;
begin
    x:=i;
    y:=j;
    if (drt1=3) or (drt1=4) or (drt1=5) then xpos:=xpos+1;
    if (drt1=1) or (drt1=7) or (drt1=8) then xpos:=xpos-1;
    if (drt1=1) or (drt1=2) or (drt1=3) then ypos:=ypos+1;
    if (drt1=5) or (drt1=6) or (drt1=7) then ypos:=ypos-1;
end;

procedure find_direction_1;
begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

move;
delay(50);
in_opto(p);
end;

```

```

procedure find_direction_2;
begin
  find_direction_0;
  direction[dnum]:=drt1;
  dnum:=dnum+1;
end;

```

```

procedure find_direction_3;
begin
  dnum:=dnum-1;
  if drt1<5 then drt1:=drt1+8;
  drt1:=direction[dnum]-4;
  move;
  delay(20);
  find_direction_0;
end;

```

```

procedure find_direction;
begin
  drt1:=direction[dnum-1];
  decr(drt1);
  find_direction_1;

  r:=p;

```

```

  incr(drt1);

```

```

find_direction_1;

q:=p;

incr(drt1);

find_direction_1;

if (q=192)and(p<>0)and(r<>0) then
begin
    decr(drt1);
    find_direction_2;
end
elseif (r=0)and(p<>0)and(q<>0) then
    find_direction_2
elseif (p=0)and(q<>0)and(r<>0) then
begin
    decr(drt1);
    make_sure_xy;
    decr(drt1);
    find_direction_2
end
else
begin
    decr(drt1);
    p:=q;
    if p<>0 then drt2:=drt1;
    case last_turn of
        true :incr(drt1);
        false:decr(drt1);
    end;
    find_direction_1;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin
  find_direction_2;
  last_turn:=not last_turn;
end
else
begin
  if (p<>0) and (drt2=0) then drt2:=drt1;
  case last_turn of
    false:incr(drt1);
    true :decr(drt1);
  end;
  make_sure_xy;
  case last_turn of
    false:incr(drt1);
    true :decr(drt1);
  end;
  find_direction_1;
  if p=192 then
    find_direction_2
  else
    begin
      if (p<>0) and (drt2=0) then drt2:=drt1;
      case last_turn of
        false:incr(drt1);
        true :decr(drt1);
      end;
      find_direction_1;
      if p=192 then
        find_direction_2

```

```

else
begin
make_sure_xy;
drt1:=drt1+4;
if drt1>8 then drt1:=drt1-8;
find_direction_1;
if p=192 then
begin
find_direction_2;
last_turn:=not last_turn;
end
else
begin
if drt2<>0 then
begin
drt1:=drt2;
find_direction_1;
find_direction_2;
end
else
begin
make_sure_xy;
find_direction_3;
if dnum>2 then find_direction_3;
end;
end;
end;
end;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 end;
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;
```

```
end;
```

```
{-----}
```

```
procedure draw_back;
```

```
begin
```

```
  gotoxy(1,5);
```

```
  write('Press any key to draw back.');
```

```
  read(kbd,c);
```

```
  clrscr;
```

```
  lwr_lft_pos;
```

```
  for k:=0 to dnum do begin
```

```
    drt1:=direction[k];
```

```
    move;
```

```
    delay(20);
```

```
    x:=i;
```

```
    y:=j;
```

```
  end;
```

```
end;
```

```
{---- main() ----}
```

```
begin
```

```
  setting;
```

```
  find_start_point;
```

```
  repeat
```

```
    if keypressed then read(kbd,c);
```

```
    find_direction;
```

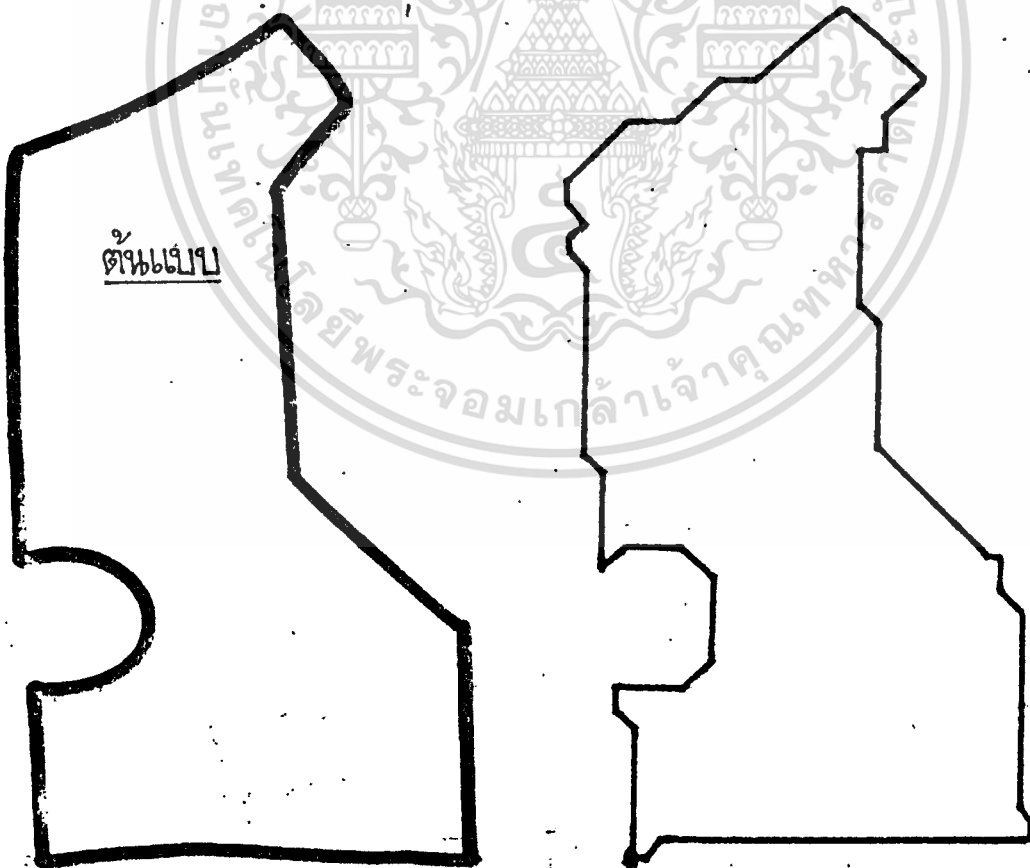
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

until
  (dnum=dnum_max) or
  ((abs(xpos)<safe_dis)and(abs(ypos)<safe_dis)and(dnum>safe_dis+3));
set_motor;
draw_back;
set_motor;
end.

```

จากการทดลองใช้งานโปรแกรมนี้ สามารถติดตามหรือจดจำรูปแบบของแบบแปลนที่ให้ไว้ก่อนได้ผลการทดลอง ดังรูปที่ 3.12



รูปที่ 3.12

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ผลการทดลองโปรแกรม
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากผลการทดลอง ได้รูปซึ่งมีลักษณะที่ไม่เรียบ มีลักษณะหักเป็นมุมตลอดการติดตาม ที่เป็นเช่นนี้เพราะลักษณะการติดตามลายเส้นของโปรแกรมนี้ทำได้โดยการตรวจสอบจุดลายเส้นล่วงหน้าแค่ในสเตปต่อไปเพียง 1 สเตปเท่านั้น ทำให้เมื่อกำลังอ่านลายเส้นอยู่ เมื่อหลุดออกจากลายเส้นโดยตรวจสอบว่าตกขอบของลายเส้นไปแล้ว ก็จะพยายามกลับเข้ามาในลายเส้นอีก จนไปตกขอบของลายเส้นอีกด้านหนึ่งจึงจะเปลี่ยนทิศทางการติดตามอีกครั้งหนึ่ง

วิธีการแก้ปัญหาดังกล่าว ก็คือทำลายเส้นของรูปต้นแบบให้มีขนาดความกว้างลดลงไปอีก เพื่อที่จะได้มีการเปลี่ยนทิศทางการติดตามบ่อย ๆ ซึ่งจะทำให้ภาพการติดตามมีลักษณะที่เรียบขึ้นและใกล้เคียงภาพเดิมมากยิ่งขึ้น

3.3.4 วิธีการรับข้อมูลภาพแบบใช้กล้องถ่ายภาพ

การใช้กล้องถ่ายภาพหรือกล้องวิดีโอ รับข้อมูลภาพ เป็นการประยุกต์ใช้งานที่ค่อนข้างจะซับซ้อน มีการวิจัยกันอย่างกว้างขวางในต่างประเทศ เพื่อคิดค้นวิธีที่จะทำให้สามารถเข้าใจภาพที่ได้มา

ในการรับข้อมูลแบบภาพโดยใช้กล้องถ่ายภาพ มีลักษณะหลักการคล้าย ๆ กับแบบการกวาดภาพ ข้อมูลที่ได้จากกล้องถ่ายภาพจะถูกนำมาเก็บไว้ในหน่วยความจำ จากนั้นก็ทำการหาขอบภาพ เพื่อประมวลผลภาพต่อไป วิธีการประมวลผลภาพต่อไปนั้น ก็เหมือนกับวิธีการกวาดภาพ ข้อดีของการใช้กล้องถ่ายภาพนี้คือ สามารถรับรู้ข้อมูลได้อย่างรวดเร็วมาก

3.4 การรับรู้ข้อมูลแบบจับสอน

ในการใช้งานเครื่องจักรอัตโนมัติ ปัญหาของการทำงานโดยใช้องการเขียนโปรแกรมสั่งงานหรือใช้การป้อนข้อมูลด้วยภาพต้นแบบ ก็คือ การที่ไม่สามารถทำให้ได้ผลงานที่ปราณีตเหมือนกับผู้ชำนาญงานทำกับมือ เช่นการเขียนโปรแกรมให้แขนกลทำการพับเสื้อตัวจำนวนมาก ๆ นั้น ชิ้นงานที่ได้จะสวยหรือไม่สวยขึ้นกับลักษณะการโปรแกรมของผู้เขียนโปรแกรมสั่งงาน ซึ่งผู้เขียนโปรแกรมส่วนใหญ่ไม่มีความชำนาญในการพับเสื้อ และผู้ที่มีความชำนาญในการพับเสื้อก็ไม่มีความรู้ความสามารถพอที่จะเขียนโปรแกรมสั่งงาน

การจับแขนกลเพื่อสอนให้แขนกลทำงาน จึงเป็นวิธีที่เหมาะสมที่จะใช้ในตัวอย่างข้างต้น เพราะไม่ต้องมีการเขียนโปรแกรมให้ยุ่งยาก ผู้ชำนาญงานก็สามารถสอนให้แขนกลทำงานได้เหมือนที่เขาทำ ผลงานที่ได้ก็เหมือนกับที่เขาทำ และเมื่อมีการสอน ก็สามารถเลือกการทำงานครั้งที่ดีที่สุด เพื่อเก็บข้อมูลลักษณะการทำงานนั้น ๆ ไว้เป็นมาตรฐานสำหรับการทำงานต่อ ๆ ไป ทำให้ผลงานที่ได้ต่อไปมีความสวยงามทุก ๆ ชิ้นเท่า ๆ กัน

หลักการในการสอนให้เครื่องจักรได้รับรู้นั้น ส่วนที่สำคัญจะแบ่งได้เป็นสามส่วน คือส่วนรับรู้การสอน ส่วนประมวลผล และส่วนที่จะนำข้อมูลที่ได้จากการสอนกลับมาใช้อีก ทั้งสามส่วนจะแยกพิจารณา ในหัวข้อนี้จะอธิบายเกี่ยวกับส่วนของการรับรู้การสอน

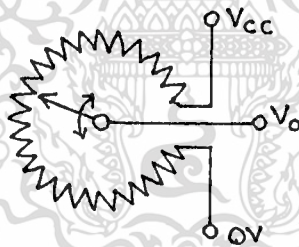
การรับรู้การสอนนั้น สิ่งสำคัญก็คือ อุปกรณ์และวิธีการรับรู้ สำหรับอุปกรณ์รับรู้นั้น อาจจะเป็นตัวอุปกรณ์ตรวจจับตำแหน่ง เป็นตัวอุปกรณ์ตรวจจับการขจัด หรืออาจจะเป็นอุปกรณ์ตรวจจับการเคลื่อนที่ก็ได้ ซึ่งก็ขึ้นอยู่กับวิธีการประยุกต์ใช้งาน จะแยกการรับรู้การสอนตามอุปกรณ์รับรู้อย่างนี้

3.4.1 การใช้ความต้านทานปรับค่าได้เป็นอุปกรณ์รับรู้

ความต้านทานปรับค่าได้ เมื่อนำมาต่อเป็นตัวหารศักดาไฟฟ้าดังรูปที่ 3.13 จะได้ว่าค่าศักดาไฟฟ้าที่ได้ออกมาจากแท็ปกลางความต้านทาน ขึ้นอยู่กับตำแหน่งหรือมุมที่แกนตัวความต้านทาน ถ้าทราบค่าศักดาก็จะทราบค่าตำแหน่งโดยจากสูตร

$$V_o = V_{cc} * \text{Angle} / 360$$

โดย Angle เป็นมุมที่แกนตัวความต้านทานหมุนไป และ V_{cc} เป็นค่าไฟเลี้ยงวงจร และ V_o เป็นศักดาไฟฟ้าที่ได้



รูปที่ 3.13

การใช้ตัวต้านทานปรับค่าได้เป็นอุปกรณ์รับรู้ตำแหน่ง

เมื่อได้ศักดาไฟฟ้าที่แปรตามตำแหน่งแล้ว การประยุกต์ใช้งานก็ขึ้นกับวิธีการของแต่ละรูปแบบ การประยุกต์ใช้งานกับการจับสอนนั้น เนื่องจากการจดจำตำแหน่ง จำเป็นต้องใช้เครื่องคอมพิวเตอร์ ซึ่งการประมวลผลของเครื่องคอมพิวเตอร์ จะใช้ข้อมูลที่เป็นข้อมูลดิจิทัล ดังนั้น จึงต้องแปลงศักดาไฟฟ้าที่ได้เป็นข้อมูลทางดิจิทัล โดยผ่านวงจรเปลี่ยนอนาลอกเป็นดิจิทัล ก่อนแล้วจึงป้อนข้อมูลที่ได้อุปกรณ์ประมวลผลตามวิธีต่อไป ซึ่งวิธีการประมวลผล จะกล่าวโดยละเอียดในหัวข้อในบทต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

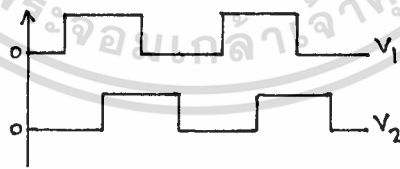
3.4.2 การใช้เอนโคเดอร์เป็นอุปกรณ์รับรู้

เอนโคเดอร์เป็นอุปกรณ์รับรู้ตำแหน่งชนิดหนึ่ง ซึ่งเป็นอุปกรณ์ที่จะให้สัญญาณรหัสออกมาค่าต่าง ๆ ตามตำแหน่งของแกนหมุนของตัวเอนโคเดอร์ขณะนั้น ๆ ตัวอย่าง เช่น เอนโคเดอร์ชนิดสองเฟส จะให้รหัสออกมาเป็นค่า 00 01 11 10 00 ... ตามลำดับเมื่อหมุนไปในทิศหนึ่ง และค่ารหัสจะเรียงกลับกันเมื่อหมุนไปในทิศตรงกันข้าม

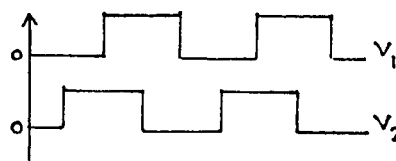
นอกจากเอนโคเดอร์ชนิดดังกล่าวแล้ว ยังมีเอนโคเดอร์อีกหลายชนิด เช่น เอนโคเดอร์ที่ให้รหัสออกมาเป็นรหัสไบนารี 4 บิต ซึ่งสะดวกในการใช้งานมาก หรือเอนโคเดอร์ที่ให้รหัสเป็นรหัสเกรย์โคด 4 บิต

ในการทดลอง จะใช้เอนโคเดอร์แบบสองเฟส ซึ่งให้รหัสออกมาดังรูปที่ 3.14

หมุนตามเข็มนาฬิกา



หมุนทวนเข็มนาฬิกา



รูปที่ 3.14

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รหัสที่ได้จากเอนโคเดอร์เมื่อหมุนแกนไปในทิศต่าง ๆ
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และเนื่องจากในทางปฏิบัติจริง ๆ ค่าที่ได้ออกมาจะมีลักษณะแตกต่างกันเพียง 4 รูปแบบคือ 00 , 01 , 11 , 10 เท่านั้น ซึ่งทำให้ทราบเพียงแค่ว่าทิศทางการเคลื่อนที่ไปทางใดเท่านั้น ดังนั้นค่าตำแหน่งจะต้องใช้คอมพิวเตอร์หรือวงจรดิจิทัลอิเล็กทรอนิกส์ ทำการนับค่าตำแหน่ง เช่นถ้าใช้รหัสที่ได้จากเอนโคเดอร์ได้ว่าเกิดการหมุนไปทางขวา ก็จะบวกค่าตำแหน่งด้วย 1 แต่ถ้าใช้ได้ว่าหมุนไปทางซ้าย ก็จะลบค่าตำแหน่งด้วย 1 คล้าย ๆ กับการอินทิเกรต

เนื่องจากถ้าให้คอมพิวเตอร์ใช้รหัสโดยตรง ซึ่งมีเพียงแค่สองเฟสเท่านั้น ความเร็วในการประมวลผลของคอมพิวเตอร์ยังช้าอยู่เมื่อเทียบกับความเร็วในการเปลี่ยนแปลงรหัสของตัวเอนโคเดอร์เมื่อถูกหมุน ทำให้เกิดความผิดพลาดขึ้นได้ เช่น ถ้ารหัสที่ได้ออกมาเทียบกับเวลาเป็น

เวลามิลลิวินาที	0	1	2	3	4	5	6	7
รหัสที่ได้จากเอนโคเดอร์	00	01	11	10	00	01	11	10
เวลาที่คอมพิวเตอร์รับข้อมูล		↑			↑			↑
ข้อมูลที่คอมพิวเตอร์รับรู้		01			00			10

จากตัวอย่างข้างต้น ข้อมูลที่ได้จากเอนโคเดอร์มีลักษณะเป็น 00 01 11 10 ... แต่ข้อมูลที่คอมพิวเตอร์รับรู้เป็น 01 00 10 ... ซึ่งเป็นรูปแบบข้อมูลของการเคลื่อนที่ตรงกันข้ามกัน ดังนั้นจึงเกิดข้อผิดพลาดขึ้นในการคำนวณตำแหน่ง หรืออีกนัยหนึ่งก็คือ อัตราความเร็วในการหมุนตัวเอนโคเดอร์ถูกจำกัดอยู่ที่อัตราเร็วค่า ๆ หนึ่ง โดยถ้าอัตราความเร็วการหมุนมีค่าสูงเกินไป คอมพิวเตอร์ก็จะไม่สามารถรับรู้ทิศทางการเคลื่อนที่ได้อย่างถูกต้อง

การเพิ่มความสามารถในการรับรู้การเคลื่อนที่ของคอมพิวเตอร์นั้น จำเป็นต้องใช้อุปกรณ์ทางดิจิทัลอิเล็กทรอนิกส์ช่วย เพราะวงจรทางดิจิทัลสามารถประมวลผลได้เร็วและถูกต้อง หลักการก็คือ นำรหัสที่ได้จากตัวเอนโคเดอร์มาผ่านวงจรดิจิทัลก่อน โดยวงจรจะทำหน้าที่เป็นวงจรนับ 4 บิต แล้ว

จึงส่งข้อมูลไบนารี 4 บิตที่ได้ไปให้คอมพิวเตอร์ประมวลผล นั่นก็คือ คอมพิวเตอร์สามารถแยกแยะรหัสที่รับเข้ามาได้ถึง 16 รูปแบบ คือ

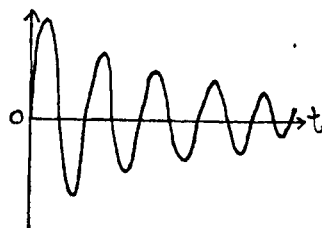
O 1 2 3 4 5 6 7 8 9 A B C D E F

โอกาสที่จะผิดพลาดเหมือนการรับรหัสสองบิตในตัวอย่างแรก ย่อมน้อยกว่า และสามารถเพิ่มความเร็วของอัตราการหมุนเอนโคเดอร์ได้เป็น 8 เท่าของการรับรู้แบบสองบิตในตัวอย่างแรก

วงจรดิจิทัลที่ใช้ร่วมกับการประมวลผลดังกล่าว แสดงไว้ในภาคผนวก สำหรับวงจรนี้ ได้ออกแบบให้มีลักษณะประหยัดอุปกรณ์และค่าใช้จ่าย โดยสามารถทำงานได้ดีเท่ากับการใช้ตัวนับขึ้นนับลงแบบไบนารีซึ่งนิยมใช้กัน

3.4.3 การใช้การตรวจจับสัญญาณในการรับรู้

การใช้การตรวจจับสัญญาณ ที่เกิดจากการหมุนสเทปมอเตอร์นั้น เป็นวิธีการที่ค่อนข้างจะยุ่งยากและซับซ้อนมาก จากหลักการที่ว่า ในสเทปมอเตอร์ที่มีแม่เหล็กถาวรอยู่ภายใน เมื่อหมุนแกนของสเทปมอเตอร์ จะทำให้ขดลวดภายในสเทปมอเตอร์ ไปตัดกับสนามแม่เหล็กจากแม่เหล็กถาวร ทำให้เกิดการสร้างศักดาไฟฟ้าขึ้นในขดลวดในแต่ละเฟสของสเทปมอเตอร์ ศักดาไฟฟ้าที่ได้จะมีลักษณะรูปคลื่นดังรูปที่ 3.15

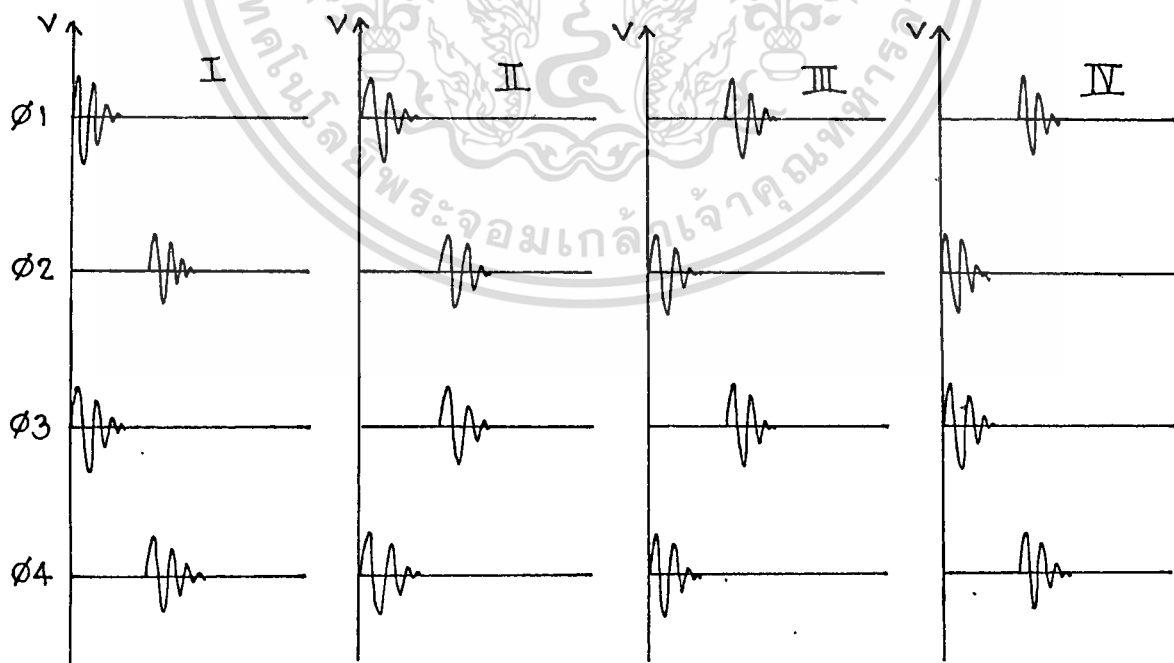


รูปที่ 3.15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับครูใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ศักดาไฟฟ้าที่เกิดขึ้นขณะหมุนแกนสเทปมอเตอร์ไป 1 สเทป
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ศักดาไฟฟ้าที่เกิดขึ้นนั้น มีค่าสูงสุดขึ้นกับความแรงและความเร็วของการหมุนแกน ค่าสนามแม่เหล็กถาวรภายใน และขนาดของสเตปมอเตอร์ ในการทดลอง ค่าศักดาสูงสุดที่ได้มีค่าเพียง 20 มิลลิโวลต์ขณะหมุนเบาและช้า ค่าศักดาที่ต่ำมากเช่นนี้ ทำให้เกิดการรบกวนจากสัญญาณรบกวนภายนอกได้ง่าย และต้องใช้วงจรซึ่งค่อนข้างละเอียดอ่อนในการตรวจจับสัญญาณที่ได้

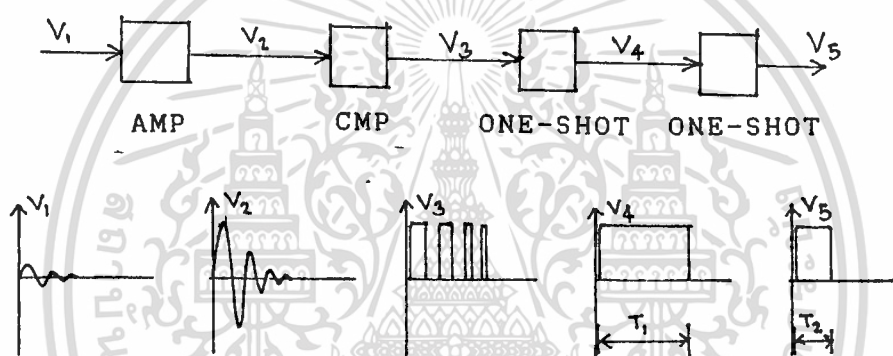
ในการหมุนแกนสเตปมอเตอร์ที่ละสเตปในแต่ละครั้งนั้น ศักดาที่ได้จากขดลวดภายในในแต่ละเฟสนั้น มีรูปคลื่นและขนาดที่เหมือนกันในทุก ๆ เฟส จากการทดลอง ได้ใช้สเตปมอเตอร์ชนิด 4 เฟส และตรวจจับรูปคลื่นของศักดาไฟฟ้าที่ได้ในแต่ละเฟสที่เกิดจากการหมุนแกนมอเตอร์ไป 1 สเตปนั้น เมื่อเปรียบเทียบรูปคลื่นที่ได้ พร้อม ๆ กันทั้ง 4 เฟส จะสังเกตเห็นว่ารูปคลื่นที่ได้นั้นมีลักษณะเหมือน ๆ กัน แต่จุดที่รูปคลื่นเริ่มต้นนั้น ไม่ตรงกัน และมีลักษณะแตกต่างกันไปถึง 4 รูปแบบดังรูปที่ 3.16



รูปที่ 3.16

เอกสารนี้เป็นเอกสารที่งานวิจัยสำหรับงานใช้ภายในเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
รูปแบบทั้ง 4 รูปแบบที่ได้จากการหมุนแกนสเตปมอเตอร์ 1 สเตป
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การนำเอาผลการทดลองดังกล่าว มาประยุกต์ใช้งานกับการรับรู้ การเคลื่อนที่ของแกนสเตปมอเตอร์นั้น เนื่องจากถ้าสามารถหาคจุดเริ่มต้นของ รูปคลื่นในแต่ละเฟสได้ ก็จะสามารถรับรู้ถึงทิศทาง การเคลื่อนที่ได้เช่นเดียวกับ ตัวเอนโคเดอร์



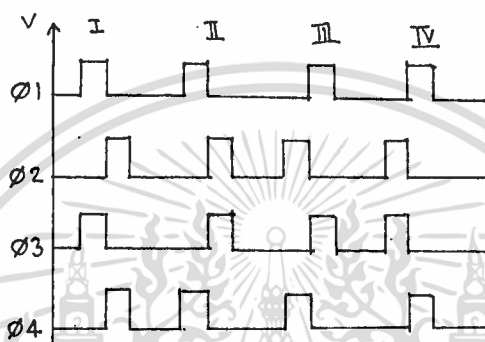
รูปที่ 3.18

การทำงานของวงจรตรวจจับและผลที่ได้ในแต่ละขั้นตอน

ในการทดลอง ได้ออกแบบวงจรตรวจจับจุดเริ่มต้นของรูปคลื่น ซึ่งรูปวงจรแสดงไว้ในภาคผนวก หลักการสำคัญของวงจรก็คือ นำสัญญาณรูป คลื่นที่ได้มาทำการขยายให้ได้คักดาที่มีค่าสูงที่สุดมากขึ้น เพื่อ่ายในการตรวจจับ จากนั้น ก็นำเอาสัญญาณที่ขยายแล้วมาผ่านวงจรเปรียบเทียบ ซึ่งจะทำให้การ เปรียบเทียบค่าสัญญาณกับค่าคักดาคงที่ค่าหนึ่ง ซึ่งทำให้สัญญาณที่ผ่านวงจร เปรียบเทียบแล้ว มีลักษณะเป็นพัลส์ใหญ่ 1 ลูก และพัลส์เล็ก ๆ ตามมาอีกจำ นวนหนึ่ง สัญญาณพัลส์ทั้งหมดที่ได้นี้ นำมาผ่านวงจรยัดเวลาหรือวงจรวนซ้ำสอง ครั้ง ซึ่งวงจรวนซ้ำทั้งสองวงจรมีค่าหน่วงไม่เท่ากัน สุดท้ายจะทำให้สัญญาณที่ได้ไม่มีพัลส์เล็ก ๆ คงเหลือแต่พัลส์กว้างที่ต้องการ ลักษณะการทำงานของ วงจรและผลที่ได้ แสดงไว้ในรูปที่ 3.18

เอกสารนี้เป็นเอกสารลิขสิทธิ์ของมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าธนบุรี ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาตจากทางมหาวิทยาลัย หากฝ่าฝืนจะดำเนินการตามกฎหมายที่เกี่ยวข้องต่อไป

เมื่อนำสัญญาณที่เกิดจากการหมุนมอเตอร์อย่างต่อเนื่อง มาผ่านวงจรดังกล่าวข้างต้นแล้ว สัญญาณในรูปที่ 3.17 ก็จะเปลี่ยนรูปเป็นรูปที่ 3.19



รูปที่ 3.19
สัญญาณเมื่อผ่านวงจรตรวจจับแล้ว

ในการทดลอง ได้ออกแบบโปรแกรมซึ่งสามารถตรวจจับสัญญาณสุดท้ายที่ได้และแสดงผลทางจอภาพ โดยแยกแสดงสัญญาณของแต่ละเฟส ตัวโปรแกรมได้แสดงไว้ดังนี้

```
program proj1.pas
const
  port_a : integer = 768 ;
  port_b : integer = 769 ;
  port_c : integer = 770 ;
  control_port : integer = 771 ;
  control_word : byte = 217 ;
var
  i,j,a1,b1,c1,d1,a2,b2,c2,d2 : integer ;
  word_in : array [1..639] of byte ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 procedure title ;
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

begin
  draw(0,5,639,5,15) ;
  draw(639,5,639,199,15) ;
  draw(639,199,0,199,15) ;
  draw(0,199,0,5,15) ;
  for i := 1 to 79 do
    for j := 1 to 39 do
      plot(i*8,j*5,15) ;
    gotoxy(32,1) ;
    write('STEP MOTOR OUTPUT') ;
    gotoxy(75,25) ;
    write('TIME') ;
    gotoxy(1,1) ;
  end ;
{ main() }
begin
  port[control_port] := control_word ;
  hires ; title ;
  sound(1000) ;
  for i := 1 to 639 do begin
    word_in[i] := port[port_a] ;
    delay(5) ;
  end ;
  nosound ;
  for i := 1 to 639 do begin
    a2 := 0 ; b2 := 0 ; c2 := 0 ; d2 := 0 ;
    word_in[i] := trunc((word_in[i] and 240)/16) ;
    if word_in[i] > 7 then begin

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 word_in[i] := word_in[i] - 8 ; a2 := 1 ;
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

end ;
if word_in[i] > 3 then begin
    word_in[i] := word_in[i] - 4 ; b2 := 1 ;
end ;
if word_in[i] > 1 then begin
    word_in[i] := word_in[i] - 2 ; c2 := 1 ;
end ;
d2 := word_in[i] ;
draw(i, 40-a1*30, i, 40-a2*30, 15) ;
draw(i, 80-b1*30, i, 80-b2*30, 15) ;
draw(i, 120-c1*30, i, 120-c2*30, 15) ;
draw(i, 160-d1*30, i, 160-d2*30, 15) ;
a1 := a2 ; b1 := b2 ;
c1 := c2 ; d1 := d2 ;
end ;
end.

```

อธิบายการทำงานของโปรแกรมได้ว่า เริ่มจากโปรแกรมทำการเซตอุปกรณ์ติดต่อ เคลียร์จอภาพ กำหนดค่าและลักษณะของตัวแปรที่ใช้ในโปรแกรม จากนั้นก็จะรอรับคีย์สั่งงาน เมื่อได้รับคีย์สั่งแล้ว จะทำการรับสัญญาณด้วยวิธีส่มตัวอย่างเข้ามา โดยมีตัวแปรกำหนดคาบเวลาหรือความถี่การส่มตัวอย่าง ข้อมูลที่ได้ทั้งหมดจะแสดงออกทางจอภาพในลักษณะเป็นกราฟและสามารถพิมพ์ออกมาทางเครื่องพิมพ์ได้

สำหรับสัญญาณที่ได้ในรูปที่ 3.19 ซึ่งเป็นสัญญาณสุดท้ายที่จะป้อนให้คอมพิวเตอร์วิเคราะห์ออกมา ซึ่งจะต้องทำให้คอมพิวเตอร์สามารถวิเคราะห์การนำและตามของพัลส์ในแต่ละเฟส ซึ่งมีลักษณะที่แตกต่างกันของรูปแบบอยู่ 4 รูปแบบ กำหนดชื่อรูปแบบแต่ละรูปแบบดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาค้นคว้าเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปแบบ :	A	B	C	D
เฟลที่ 1:	น้ำ	น้ำ	ตาม	ตาม
เฟลที่ 2:	ตาม	ตาม	น้ำ	น้ำ
เฟลที่ 3:	น้ำ	ตาม	ตาม	น้ำ
เฟลที่ 4:	ตาม	น้ำ	น้ำ	ตาม

เมื่อคอมพิวเตอร์รับข้อมูลไป ก็จะทำให้การเช็คข้อมูลที่ได้ว่าเป็นไปตามรูปแบบใด จากนั้นก็สนใจเฉพาะชื่อรูปแบบคือ A B C D เท่านั้น และจากการทดลอง จะได้เหตุการณ์ที่ว่า ถ้าหมุนแกนสเตปมอเตอร์ไปในทิศทางหนึ่ง จะได้รูปแบบเป็น A,B,C,D,A,... ตามลำดับ และถ้าหมุนไปในอีกทิศทางหนึ่งก็จะได้รูปแบบที่ตรงจับได้เป็น A,D,C,B,A,... ตามลำดับ ซึ่งก็เหมือนกับตัวเอนโคดเดอร์ แต่ถ้าเกิดการหมุนกลับทิศกัน ในช่วงการกลับทิศนั้นจะไม่เหมือนตัวเอนโคดเดอร์ คือจะมีลักษณะดังตัวอย่างคือ

A, B, C, D, A, B, C, D, A, C, B, A, D, C, B, A, ...

^ คือจุดที่มีการหมุนกลับทิศ

นั่นคือผลที่เกิดจากการตัดสินใจแม่เหล็ก ทำให้ได้รูปแบบซึ่งเป็นรูปแบบตรงข้าม ไม่ใช่รูปแบบที่ต่อเนื่อง (รูปแบบ A ตรงข้ามรูปแบบ C) การรับรู้ว่าการเปลี่ยนทิศทางการหมุน จึงเป็นหน้าที่ของหน่วยประมวลผลหรือคอมพิวเตอร์ ซึ่งจะต้องรับทราบรูปแบบ และทำการคำนวณค่าตำแหน่ง ในลักษณะคล้าย ๆ กับการคำนวณรหัสจากตัวเอนโคดเดอร์

ในการทดลอง ได้ออกแบบโปรแกรมซึ่งสามารถตรวจเช็คได้ว่าสัญญาณที่เข้ามานั้นอยู่ในรูปแบบใด โปรแกรมแสดงไว้ดังนี้

```
program proj2.pas
```

```
const
```

```
  port_a : integer = 768 ;
```

```
  port_b : integer = 769 ;
```

```
  port_c : integer = 770 ;
```

```
  control_port : integer = 771 ;
```

```
  control_word : byte = 217 ;
```

```
  f1std1 : integer = 200 ;
```

```
  f2std1 : integer = 50 ;
```

```
  f3std1 : integer = 250 ;
```

```
  f4std1 : integer = 50 ;
```

```
var
```

```
  i,j,k,j1234,state:integer;
```

```
  var p1,p2,p3,p4,p12,p34,p1234:integer;
```

```
  a:char;
```

```
procedure phin;
```

```
begin
```

```
  inline
```

```
  (
```

```
    $50/      {PUSH  AX      }
```

```
    $53/      {PUSH  BX      }
```

```
    $51/      {PUSH  CX      }
```

```
    $52/      {PUSH  DX      }
```

```
    $BA/$00/$03/ {MOV   DX,0300 }
```

```
    $EC/      {IN     AL,DX    }
```

```
    $D0/$E8/   {SHR    AL,1    }
```

```
    $D0/$E8/   {SHR    AL,1    }
```

```
    $D0/$E8/   {SHR    AL,1    }
```

```
    $D0/$E8/   {SHR    AL,1    }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

$B4/$00/      {MOV   AH,00   }
$A3/P1234/    {MOV   P1234,AX}
$50/          {PUSH  AX      }
$24/$03/      {AND   AL,03   }
$A3/P34/      {MOV   P34,AX  }
$58/          {POP   AX      }
$50/          {PUSH  AX      }
$D0/$E8/      {SHR   AL,1    }
$D0/$E8/      {SHR   AL,1    }
$A3/P12/      {MOV   P12,AX  }
$58/          {POP   AX      }
$89/$C3/      {MOV   BX,AX   }
$80/$E3/$01/  {AND   BL,01   }
$89/$1E/P4/   {MOV   P4,BX   }
$D0/$E8/      {SHR   AL,1    }
$89/$C3/      {MOV   BX,AX   }
$80/$E3/$01/  {AND   BL,01   }
$89/$1E/P3/   {MOV   P3,BX   }
$D0/$E8/      {SHR   AL,1    }
$89/$C3/      {MOV   BX,AX   }
$80/$E3/$01/  {AND   BL,01   }
$89/$1E/P2/   {MOV   P2,BX   }
$D0/$E8/      {SHR   AL,1    }
$A3/P1/       {MOV   P1,AX   }
$5A/          {POP   DX      }
$59/          {POP   CX      }
$5B/          {POP   BX      }

```

\$58/ {POP AX }
 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
); end;
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure beep;
begin
    sound(1000);
    delay(f4std1);
    nosound;
end;
procedure set_v;
begin
    port[control_port]:=control_word;
    state:=1;
end;
procedure title;
begin
    hires;
    writeln('STEP MOTOR REAL TIME POSITION');
end;
procedure won_a;
begin
    state:=1;
    write('A');
    beep;
end;
procedure won_b;
begin
    state:=2;
    write('B');
    beep;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure won_c;
begin
  state:=3;
  write('C');
  beep;
end;
procedure won_d;
begin
  state:=4;
  write('D');
  beep;
end;
procedure won_ab;
begin
  case state of
    2,4 : won_a;
    1,3 : won_b;
  end;
end;
procedure won_cd;
begin
  case state of
    2,4 : won_c;
    1,3 : won_d;
  end;
end;
procedure won_bc;

```

```
begin
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    2,4 : won_c;
    1,3 : won_b;
end;
end;
procedure won_da;
begin
    case state of
        2,4 : won_a;
        1,3 : won_d;
    end;
end;
{-----}
procedure pstn_10xx_3;
begin
    i:=f3std1;
    repeat
        phin;
        if p34<>0 then i:=1;
        inline(*50/*A1/i/*48/*A3/i/*58); {i:=i-1;}
    until i=0;
    case p34 of
        1 : won_a;
        2 : won_b;
        0 : won_ab;
    end;
end;
end;
procedure pstn_10xx_2;
begin

```

```

j1234:=8;
repeat
  phin;
  inline($50/$A1/j1234/$0B/$06/p1234/$A3/j1234/$58);
  {j1234:=j1234 or p1234;}
case j1234 of
  9,10,11,13,14,15 : i:=1;
end;
inline($50/$A1/i/$48/$A3/i/$58); {i:=i-1;}
until i=0;
case j1234 of
  9,13 : won_a;
  10,14 : won_b;
  12 : won_ab;
end;
end;
procedure pstn_10xx_sub;
begin
  repeat
    'phin;
    if (p34<>0) or (p2=1) then i:=1;
    inline($50/$A1/i/$48/$A3/i/$58); {i:=i-1;}
  until i=0;
end;
procedure pstn_10xx_1;
begin
  i:=f2std1;

```

```

else if p4=1 then won_b
else if p2=1 then pstn_10xx_3
else pstn_10xx_2;
end;
procedure pstn_10xx;
begin
  i:=f1std1;
  pstn_10xx_sub;
  if      p3=1 then won_a
  else if p4=1 then won_b
  else if p2=0 then pstn_10xx_1;
end;
{-----}
procedure pstn_01xx_3;
begin
  i:=f3std1;
  repeat
    phin;
    if p34<>0 then i:=1;
    * inline($50/$A1/i/$48/$A3/i/$58); {i:=i-1;}
  until i=0;
  case p34 of
    1 : won_d;
    2 : won_c;
    0 : won_cd;
  end;
end;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure pstn_01xx_2;
begin
  i:=f3std1;
  j1234:=4;
  repeat
    phin;
    inline($50/$A1/j1234/$0B/$06/p1234/$A3/j1234/$58);
    {j1234:=j1234 or p1234;}
    case j1234 of
      5,6,7,13,14,15 : i:=1;
    end;
    inline($50/$A1/i/$48/$A3/i/$58); {i:=i-1;}
  until i=0;
  case j1234 of
    5,13 : won_d;
    6,14 : won_c;
    12   : won_cd;
  end;
end;

procedure pstn_01xx_sub;
begin
  repeat
    phin;
    if (p34<>0) or (p1=1) then i:=1;
    inline($50/$A1/i/$48/$A3/i/$58); {i:=i-1;}
  until i=0;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure pstn_01xx_1;
begin
  i:=f2std1;
  pstn_01xx_sub;
  if      p3=1 then won_d
  else if p4=1 then won_c
  else if p1=1 then pstn_01xx_3
  else pstn_01xx_2;
end;
procedure pstn_01xx;
begin
  i:=f1std1;
  pstn_01xx_sub;
  if      p3=1 then won_d
  else if p4=1 then won_c
  else if p1=0 then pstn_01xx_1;
end;
{-----}
procedure pstn_xx10_3;
begin
  i:=f3std1;
  repeat
    phin;
    if p12<>0 then i:=1;
    inline($50/$A1/i/$48/$A3/i/$58); {i:=i-1;}
  until i=0;
  case p12 of
    1 : won_a;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    0 : won_da;
end;
end;
procedure pstn_xx10_2;
begin
    i:=f3std1;
    j1234:=2;
    repeat
        phin;
        inline($50/$A1/j1234/$0B/$06/p1234/$A3/j1234/$58);
        {j1234:=j1234 or p1234;}
        case j1234 of
            6,7,10,11,14,15 : i:=1;
        end;
        inline($50/$A1/i/$48/$A3/i/$58); {i:=i-1;}
    until i=0;
    case j1234 of
        6,7 : won_a;
        10,11 : won_d;
        3 : won_da;
    end;
end;
end;
procedure pstn_xx10_sub;
begin
    repeat
        phin;
        if (p12<>0) or (p4=1) then i:=1;
        inline($50/$A1/i/$48/$A3/i/$58); {i:=i-1;}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure pstn_xx10_1;
begin
  i:=f2std1;
  pstn_xx10_sub;
  if      p1=1 then won_a
  else if p2=1 then won_d
  else if p4=1 then pstn_xx10_3
  else pstn_xx10_2;
end;
procedure pstn_xx10;
begin
  i:=f1std1;
  pstn_xx10_sub;
  if      p1=1 then won_a
  else if p2=1 then won_d
  else if p4=0 then pstn_xx10_1;
end;
{-----}
procedure pstn_xx01_3;
begin
  i:=f3std1;
  repeat
    phin;
    if p12<>0 then i:=1;
    inline($50/$A1/i/$48/$A3/i/$58); {i:=i-1;}
  until i=0;
  case p12 of

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    0 : won_bc;
end;
end;
procedure pstn_xx01_2;
begin
    i:=f3std1;
    j1234:=1;
    repeat
        phin;
        inline($50/$A1/j1234/$0B/$06/p1234/$A3/j1234/$58);
        {j1234:=j1234 or p1234;}
        case j1234 of
            5,7,9,11,13,15 : i:=1;
        end;
        inline($50/$A1/i/$48/$A3/i/$58); {i:=i-1;}
    until i=0;
    case j1234 of
        5,7 : won_b;
        6,9 : won_c;
        3   : won_bc;
    end;
end;
end;
procedure pstn_xx01_sub;
begin
    repeat
        phin;
        if (p12<>0) or (p3=1) then i:=1;
        inline($50/$A1/i/$48/$A3/i/$58); {i:=i-1;}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure pstn_xx01_1;
begin
  i:=f2std1;
  pstn_xx01_sub;
  if      p1=1 then won_b
  else if p2=1 then won_c
  else if p3=1 then pstn_xx01_3
  else pstn_xx01_2;
end;
procedure pstn_xx01;
begin
  i:=f1std1;
  pstn_xx01_sub;
  if      p1=1 then won_b
  else if p2=1 then won_c
  else if p3=0 then pstn_xx01_1;
end;
{-----}
procedure pstn;
begin
  repeat phin; until p1234=0;
  repeat phin; until p1234<>0;
  if      p12=2 then pstn_10xx
  else if p12=1 then pstn_01xx
  else if p34=2 then pstn_xx10
  else if p34=1 then pstn_xx01;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{main()}
begin
  set_v; title; beep;
  repeat . pstn; until keypressed;
end.
```

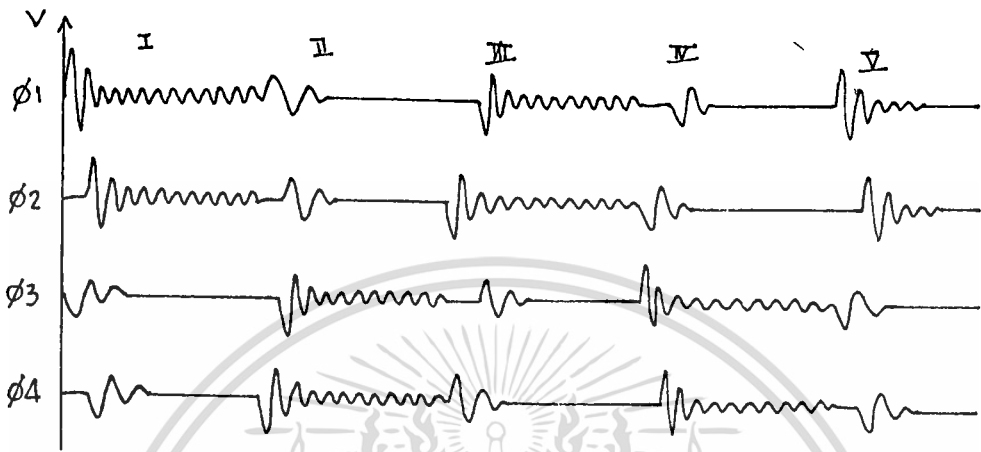
ตัวโปรแกรมมีลักษณะที่ค่อนข้างจะซับซ้อนมาก และต้องใช้ภาษาเครื่องช่วยด้วย เพราะการตรวจจับสัญญาณนั้น จำเป็นต้องใช้ความเร็วในการประมวลผลที่สูงมาก หลักการของโปรแกรมพอสังเขปอธิบายได้ดังนี้

เริ่มจากโปรแกรมจะจัดการกับตัวแปรต่าง ๆ และค่าคงที่ต่าง ๆ ที่จำเป็นในการประมวลผล ต่อจากนั้น ก็จะเริ่มรอรับสัญญาณข้อมูลที่เข้ามา เมื่อมีข้อมูลเข้ามาชุดแรก ก็จะทำการทดสอบว่าข้อมูลที่ได้นั้น มีลักษณะจัดอยู่ในรูปแบบชื่ออะไร A, B, C, หรือ D เมื่อทราบรูปแบบแล้ว ก็จะแสดงชื่อรูปแบบบนจอภาพ แล้วก็ตรวจจับสัญญาณข้อมูลตัวต่อไป

หลักการจัดรูปแบบของสัญญาณนั้น โปรแกรมจะรับข้อมูลที่เข้ามาชุดแรก คือ ชุดที่เฟลไดเฟลหนึ่งเป็น 1 ก่อน จากนั้น ก็ตรวจเช็คดูว่า ในขณะที่นั้น ยังมีสัญญาณเฟลไดเป็น 1 อีกบ้าง แล้วก็ห้วงเวลาไประยะหนึ่งซึ่งเป็นเวลาที่ข้อมูลของเฟลที่นำห่างจากข้อมูลของเฟลตัวที่ตาม จากนั้นก็ตรวจข้อมูลดูว่าเฟลไดตามบ้าง เมื่อทราบว่าเฟลไดตามและเฟลไดนำ ก็จะทราบรูปแบบได้ แต่ถ้าการตรวจจับเกิดปัญหา เช่นข้อมูลได้มาไม่ครบทุกเฟล หายไปบางเฟล โปรแกรมที่ออกแบบไว้ ก็จะสามารถคาดเดาได้ว่ารูปแบบควรจะเช่นไร โดยดูจากรูปแบบก่อนหน้านั้น 1 สเตป และข้อมูลที่ตรวจสามารถตรวจจับได้บางเฟลเท่านั้น ไม่จำเป็นต้องได้ข้อมูลมาครบทุกเฟล

ในการทดลอง ได้ใช้ออสซิลโลสโคปตรวจจับสัญญาณที่ได้จากการหมุนสเตปมอเตอร์ไป และได้พบว่า สัญญาณที่เข้ามารบกวนขดลวดในตัวสเตปมอเตอร์นั้น เข้ามารบกวนไม่ทุกเฟล ดังรูปที่ 3.20

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.20

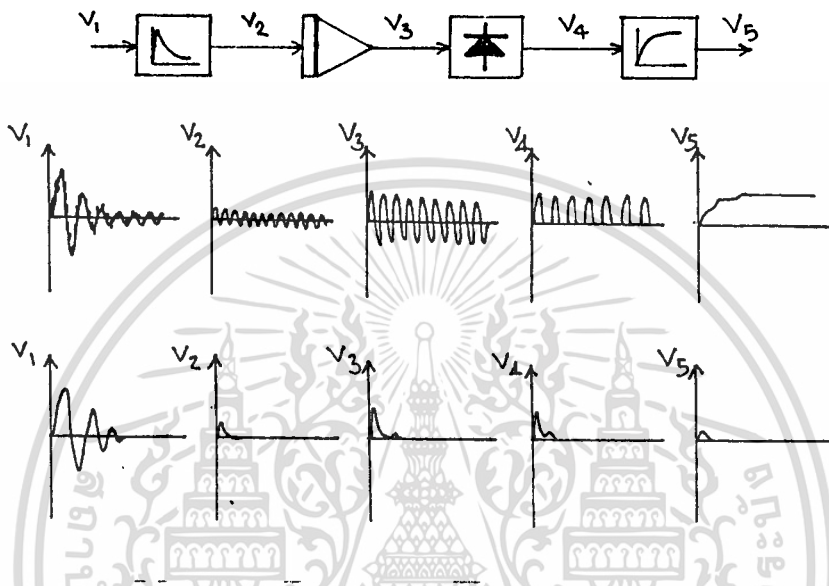
สัญญาณรบกวนในขดลวดของสเตปของมอเตอร์

อธิบายได้ว่า การเกิดการรบกวนของสัญญาณรบกวนที่ขดลวดนั้น ขึ้นอยู่กับตำแหน่งของขดลวดว่าจะอยู่ที่ตำแหน่งที่สัญญาณรบกวนเข้ามารบกวนได้หรือไม่ และจากการที่ได้ทดลองพบว่า สัญญาณรบกวนจะเข้ามารบกวนได้เพียง 2 เฟสเท่านั้น และจะสลับไปยังคู่เฟสอื่นในสเตปต่อไป

ในการทดลอง ได้ออกแบบวงจรอิเล็กทรอนิกส์เพื่อตรวจจับสัญญาณรบกวน โดยวงจรมีหลักการดังรูปที่ 3.21

ในวงจรประกอบด้วยวงจรตอบสนองความถี่สูง หรือวงจรกรองความถี่ต่ำออก จะทำหน้าที่แยกเอาเฉพาะความสูงหรือความถี่ของสัญญาณรบกวนเท่านั้น สัญญาณรบกวนที่แยกเอามาได้นั้นจะถูกนำไปขยายสัญญาณให้มีค่าสูงขึ้น จากนั้นก็จะถูกนำไปผ่านวงจรเรกติไฟร์ โดยจะเอาเฉพาะค่าที่เป็นบวกเท่านั้น ดังรูปที่ 3.21 ผลที่ได้ก็จะเป็นสัญญาณแบบฮาล์ฟเวฟ ต่อจากนั้นก็จะถูกนำไปผ่านวงจรกรองความถี่สูงออก ทำให้ได้สัญญาณออกมาเป็นค่าคิกคากระแสดตรง และเมื่อนำคิกคาสุดท้ายดังกล่าว เข้าไปยังเกทของวงจรดิจิทัล ผลสุดท้ายก็คือ ถ้าพบสัญญาณรบกวน วงจรก็จะให้เอาท์พุทเป็น 1 และถ้าไม่พบสัญญาณรบกวน เอาท์พุทจะเป็น 0

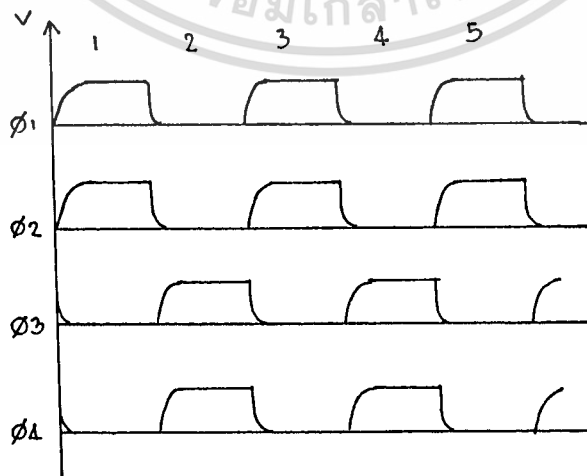
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปเผยแพร่โดยไม่ได้รับอนุญาต
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.21

หลักการตรวจจับสัญญาณรบกวน

จากการทดลอง ทำให้ได้ผลการทดลองดังรูปที่ 3.22



รูปที่ 3.22

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
เอาทบทของวงจรเมื่อหมุนสแตปมอเตอร์ไป 5 สเตป
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามแก้ไขเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากผลการทดลองดังกล่าว จะสรุปได้ว่าจะสามารถใช้วิธีตรวจ
จับสัญญาณรบกวนที่เกิดขึ้นในขดลวดของสเตปมอเตอร์ ในขณะที่หมุนไปในมุมหรือ
ตำแหน่งต่าง ๆ แต่วิธีการนี้ ยังไม่สามารถหาทิศทางของการเคลื่อนที่ได้ว่า
เคลื่อนที่ไปในทิศทางใด เพราะการเคลื่อนที่ไปในทิศทางหนึ่งจะให้สัญญาณออกมา
เหมือนกันกับที่เคลื่อนที่ไปในทิศตรงกันข้าม แต่ถ้าทราบว่ามีการเคลื่อนที่ไปใน
ทิศทางเดียวก็สามารถทราบจำนวนสเตปที่เคลื่อนที่ไปได้ เช่นเดียวกับที่ใช้ตัว
เอนโคเดอร์

เมื่อได้ข้อมูลเข้ามาแล้ว ต่อจากนั้นจะเป็นวิธีการประมวลผล
เก็บข้อมูล และนำเอาข้อมูลที่ได้ ไปใช้ประโยชน์อื่น ๆ เช่นการสอน ซึ่งจะ
เสนอในบทต่อไป

4. วิธีการประมวลผลและเก็บข้อมูล

เมื่อมีการรับรู้ข้อมูลเข้ามาแล้ว หากสามารถเข้าใจข้อมูลและหาวิธีเก็บข้อมูลนั้นไว้ได้ และสามารถนำเอาข้อมูลนั้น ๆ มาใช้ให้เกิดประโยชน์ใหม่ได้ โดยอาจจะให้เครื่องจักรทำงานตามข้อมูลที่รับรู้มา ทั้งหมดที่กล่าวมาก็คือการสอน

จากที่ได้เสนอวิธีการรับรู้ข้อมูลด้วยเทคนิคต่าง ๆ ไว้ในบทที่ 3 และในบทนี้ จะกล่าวถึงการนำข้อมูลที่ได้ มาประมวลผลในลักษณะต่าง ๆ เช่นการย่อขยาย การเปลี่ยนสเกล และจะกล่าวถึงวิธีการเก็บข้อมูลลงในหน่วยความจำในลักษณะต่าง ๆ ให้ได้ประสิทธิภาพ หรือเปลืองหน่วยความจำน้อยที่สุด แต่ได้ปริมาณข้อมูลมากที่สุด ในขั้นแรกนั้น ขอกล่าวถึงรูปแบบการรับข้อมูลก่อน

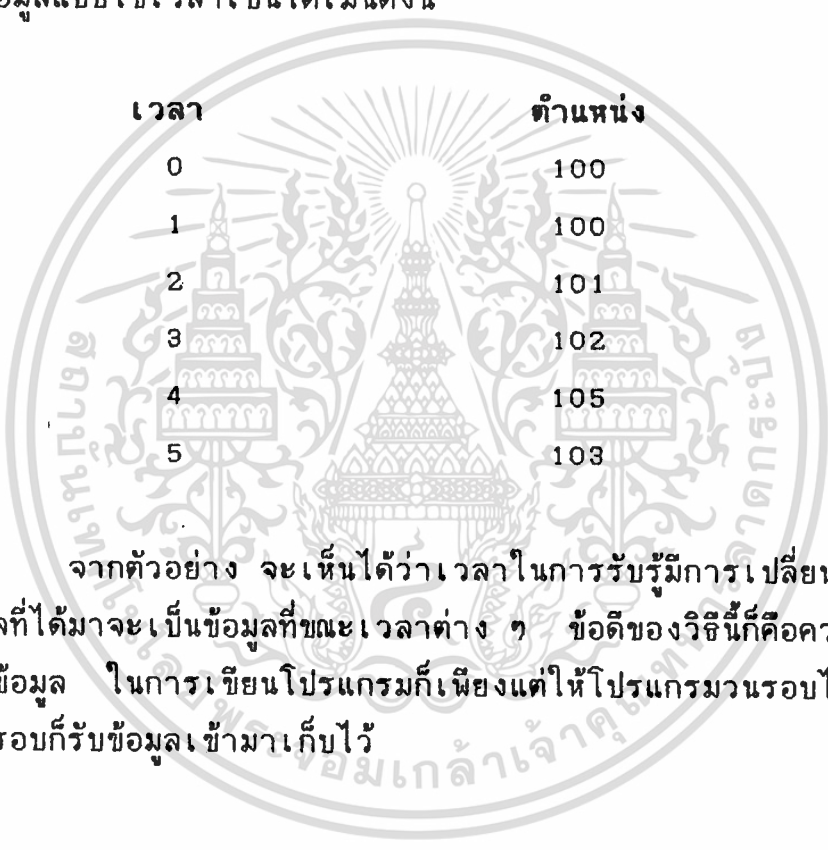
4.1 วิธีการรับข้อมูล

การรับข้อมูลเข้ามานั้น เราจะแบ่งออกเป็นสามประเภทใหญ่ ๆ ตามลักษณะของโดเมน คือ การรับข้อมูลโดยใช้เวลาเป็นโดเมน การรับข้อมูลโดยใช้ตำแหน่งเป็นโดเมน และการรับข้อมูลโดยใช้การจัดเป็นโดเมน รูปแบบการรับข้อมูลนี้ จะเกี่ยวพันกับการเก็บข้อมูลลงในหน่วยความจำด้วย แต่อาจจะมีเทคนิคอื่น ๆ ได้ เช่นรับข้อมูลมาโดยใช้เวลาเป็นโดเมน แต่เวลาเก็บข้อมูลลงในหน่วยความจำ จะใช้โปรแกรมคอมพิวเตอร์แปลงโดเมนจากเวลาเป็นตำแหน่งหรือการจัดและเก็บข้อมูลนั้นไว้ในโดเมนใหม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
จะแยกอธิบายการรับข้อมูลโดยใช้โดเมนแบบต่าง ๆ ดังต่อไปนี้
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมีเหตุดเบี่ยงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.1 การรับข้อมูลแบบใช้เวลาเป็นโดเมน

เป็นวิธีการที่นิยมกันมากที่สุด เพราะง่ายที่สุด ไม่มีอะไรซับซ้อน ในการเขียนโปรแกรม หลักการก็คล้ายคลึงกับการสุ่มตัวอย่าง ขอยกตัวอย่าง การรับข้อมูลแบบใช้เวลาเป็นโดเมนดังนี้



เวลา	ตำแหน่ง
0	100
1	100
2	101
3	102
4	105
5	103

จากตัวอย่าง จะเห็นได้ว่าเวลาในการรับรู้การเปลี่ยนแปลงที่คงที่ ข้อมูลที่ได้มาจะเป็นข้อมูลที่ขณะเวลาต่าง ๆ ข้อดีของวิธีนี้ก็คือความง่ายในการรับรู้ข้อมูล ในการเขียนโปรแกรมก็เพียงแค่ให้โปรแกรมวนรอบไปเรื่อย ๆ ในแต่ละรอบก็รับข้อมูลเข้ามาเก็บไว้

4.1.2 การรับข้อมูลแบบใช้ตำแหน่งเป็นโดเมน

ตรงกันข้ามกับหัวข้อแรก การรับข้อมูลแบบนี้จะมีข้อยุ่งยากในการเขียนโปรแกรมขึ้นมาอีกเล็กน้อย หลักการก็คือจะนับเวลาที่มีการหยุดอยู่ ณ ตำแหน่งใด ๆ เมื่อมีการเปลี่ยนตำแหน่ง ก็จะเริ่มต้นนับเวลาใหม่ ข้อมูลที่ได้ก็คือเวลา โดยมีตำแหน่งและทิศทางเป็นโดเมน ตัวอย่างแสดงไว้ดังนี้

ตำแหน่ง	ทิศทางการเคลื่อนที่	เวลาที่หยุดที่ตำแหน่งนี้
100	-	10
101	+1	12
102	+1	1
103	+1	200
102	-1	2

เนื่องจากในทางปฏิบัตินั้น ข้อมูลที่ได้มาจะเป็นข้อมูลของตำแหน่งที่เวลาใด ๆ ดังนั้น โปรแกรมคอมพิวเตอร์จึงต้องเปลี่ยนการรับรู้โดยเปลี่ยนโดเมนเวลาเป็นโดเมนตำแหน่ง ก็คือ โปรแกรมจะคอยเช็คตำแหน่งตลอดเวลา และนับเวลาที่ตำแหน่งหยุดอยู่กับที่ จากนั้นเมื่อเกิดการเปลี่ยนแปลงตำแหน่ง จึงนับเวลาใหม่ และให้เวลาที่นับได้ในตำแหน่งเก่าเป็นตัวข้อมูลสุดท้ายที่จะนำไปประมวลผลต่อไป วิธีการนี้ มีความซับซ้อนของโปรแกรมบ้าง

4.1.3 การรับข้อมูลแบบใช้การจัดเป็นโดเมน

การรับข้อมูลแบบนี้ คล้ายกับการใช้ตำแหน่งเป็นโดเมนมาก แต่ต่างกันที่ การรับข้อมูลแบบนี้ จะไม่สนใจการเปลี่ยนแปลงตำแหน่งทุก ๆ ครั้ง แต่จะสนใจเมื่อการเปลี่ยนแปลงตำแหน่งนั้นมีการจัดมากกว่าค่าที่กำหนด เพื่อความเข้าใจจะขอยกตัวอย่างดังนี้

ตำแหน่ง	การจัด	เวลาที่หยุด	เวลาที่เก็บข้อมูล
100	0	5	-
101	1	10	
102	2	7	
103	0	4	$10+7+4 = 21$
104	1	5	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตัวอย่าง เมื่อการจัดเป็น 3 ก็ะบันทึกข้อมูลครึ่งหนึ่ง ลักษณะการรับข้อมูลแบบนี้ จะสนใจเฉพาะเมื่อตำแหน่งเปลี่ยนไปตามจำนวนกำหนดเท่านั้น ข้อดีของวิธีนี้ก็คือ การลดความละเอียดเกินไปของตัวอุปกรณ์รับรู้ข้อมูล ซึ่งการที่รับรู้ข้อมูลมากเกินไปจนความจำเป็น จะทำให้เปลืองเวลาคำนวณ และเปลืองหน่วยความจำที่ใช้มาก

4.2 วิธีการประมวลผลข้อมูล

ในการใช้งานจริงของระบบการสอนนั้น เมื่อรับรู้ข้อมูลเข้ามา ข้อมูลที่ได้มาอาจจะต้องมีการเปลี่ยนแปลง ตัวอย่างเช่น ในการสอนให้สแตปมอเตอร์หมุนในทิศทางต่าง ๆ โดยใช้ตัวเอนโคดเดอร์ต่อแกนร่วมกับมอเตอร์ แต่เกิดปัญหาที่จำนวนพัลส์ต่อรอบของตัวเอนโคดเดอร์กับจำนวนสแตปต่อรอบของสแตปมอเตอร์ไม่เท่ากัน เช่น ในการทดลอง จำนวนพัลส์ต่อรอบของเอนโคดเดอร์เท่ากับ 390 พัลส์ แต่จำนวนสแตปต่อรอบของสแตปมอเตอร์เท่ากับ 400 ดังนั้น การที่จะนำเอาข้อมูลตำแหน่งที่ได้จากตัวเอนโคดเดอร์มาใช้เป็นข้อมูลขับเคลื่อนสแตปมอเตอร์ จะทำให้ได้การเคลื่อนที่ที่ผิดเพี้ยนไปจากความเป็นจริง ดังนั้นจึงต้องมีการประมวลผลข้อมูล เพื่อปรับสเกลต่าง ๆ ให้ตรงกัน หรือให้สามารถเข้ากันได้

อีกตัวอย่างหนึ่งก็คือ การใช้เครื่องอ่านภาพทำการอ่านภาพต้นแบบ และนำข้อมูลมาบังคับให้เครื่องตัดแผ่นเหล็ก ตัดตามรูปต้นแบบ ในบางครั้ง ผู้ใช้มีความต้องการขยายหรือย่อชิ้นงานจากรูปต้นแบบ อาจจะต้องการให้ชิ้นงานใหญ่กว่าภาพต้นแบบอีก 1 เท่า หรืออาจให้ย่อขนาดลงเหลือ 75 เปอร์เซ็นต์ โปรแกรมคอมพิวเตอร์จึงมีบทบาทมากในการประมวลผล

4.2.1 การขยายหรือย่อข้อมูลด้วยอัตราส่วนลงตัว

หากจำเป็นต้องขยายหรือย่อสเกลต่าง ๆ ด้วยตัวเลขลงตัว สามารถเขียนโปรแกรมจัดการได้ไม่ยาก หลักการแสดงไว้กับตัวอย่าง

ข้อมูลดิบ	ข้อมูลเมื่อขยาย 100 %	ข้อมูลเมื่อย่อ 50 %
0	0	0
1	2	0
2	4	1
3	6	1
4	8	2
5	10	2

ดังนั้นการรับรู้ข้อมูลจึงทำได้ไม่ยาก และการประมวลผลในทันทีทันใดที่รับข้อมูลเข้ามาจึงไม่เป็นปัญหาซับซ้อน เพราะไม่ต้องมีการคำนวณอะไรมากมาย ตัวเลขต่าง ๆ ก็ลงตัว

4.2.2 การขยายหรือย่อข้อมูลด้วยอัตราส่วนจำนวนจริง

ในบางครั้ง มีความจำเป็นที่จะต้องย่อขยายข้อมูลด้วยอัตราส่วนที่ไม่เป็นเลขลงตัว อาจเป็นเลขที่มีจุดทศนิยม หรือเป็นเลขจำนวนจริง เช่นค่าพาย ค่าคงที่ต่าง ๆ เป็นต้น กรณีที่ไม่ต้องการความเร็ว เช่นมีข้อมูลอยู่ในหน่วยความจำแล้ว จากนั้น ก็ประมวลผลข้อมูลและเก็บในหน่วยความจำ กรณีไม่ซับซ้อนอะไร เพราะสามารถเขียนโปรแกรมจัดการข้อมูลได้ไม่ยากนัก แต่ถ้ามีความจำเป็นต้องคำนวณด้วยความรวดเร็ว เช่น ในการสอนที่มีตัวรับรู้กับอุปกรณ์ทำงานแยกออกจากกัน และต่างสเกลกันด้วยอัตราส่วนไม่ลงตัว ขณะที่ทำการสอนนั้น ต้องการให้อุปกรณ์ทำงานให้ดูด้วยพร้อม ๆ กัน เป็นการตรวจสอบขณะสอน ดังนั้น ข้อมูลที่ได้จากการสอนจากอุปกรณ์รับรู้ จึงต้องนำมาประมวลผลในทันทีขณะนั้น และส่งข้อมูลไปควบคุมให้อุปกรณ์ที่ทำงาน ทำงานตามที่ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้คิดแปลงเนื้อหา และต้องอ้างอิงเงาของเอกสารทุกครั้งที่มีการนำไปใช้

สอนขณะนั้นด้วย ดังนั้น โปรแกรมจึงต้องมีประสิทธิภาพและมีความเร็วสูง
สามารถคำนวณสเกลได้ด้วยความเร็วและคำนวณตลอดเวลา

ขอยกตัวอย่างการย่อขยายด้วยอัตราส่วนไม่ลงตัวดังนี้

ข้อมูลดิบ	ขยาย 3 ต่อ 5	ย่อ 5 ต่อ 3
0	0	0
1	2	1
2	3	2
3	5	2
4	7	3
5	8	3
6	10	4
7	12	5
8	13	5
9	15	6

ลักษณะการคำนวณก็คล้ายกับการคำนวณที่มีการปัดเศษ โดยอาจจะ
ตั้งหลักไว้ว่าปัดขึ้นหมด หรือปัดลงหมด หรือ ถ้าเศษเกิน 50 เปอร์เซนต์ก็ทำ
การปัดขึ้น เป็นต้น

ในการทดลอง ได้ออกแบบโปรแกรมซึ่งเป็นโปรแกรมทำหน้าที่รับ
ข้อมูลจากตัวเอนโคดเดอร์ทำการคำนวณสเกล เปลี่ยนจากจำนวนฟิลล์ 390
ฟิลล์ต่อรอบ เป็น 360 องศาต่อรอบ และแสดงค่าองศาทางจอภาพด้วยเทคนิค
4 หลัก ซึ่งอัตราส่วน 390 ต่อ 360 เป็นอัตราส่วนที่ไม่ลงตัว และจำนวน
ฟิลล์ของตัวเอนโคดเดอร์สามารถเปลี่ยนค่าได้ตามพอใจ โปรแกรมแสดงได้
ดังต่อไปนี้

```

program proj4.pas
const
  port_a : integer = 768;
  port_b : integer = 769;
  port_c : integer = 770;
  control_port : integer = 771;
  control_word : byte = $41;
var
  a,b : char;
  r,p : real;
  l : byte;
  st : string[10];
  i,j,k : integer;
  state,state1,position : integer;
  data : array[0..6399] of byte;
  vd_addr,pls_stp: integer;
procedure ppjj;
begin
  inline
  (
    $50/      {PUSH  AX   }
    $8A/$26/b/ {MOV   AH,[b]}
    $80/$E4/$0F/{AND   AH,$0F }
    $D0/$E4/   {SHL   AH,1  }
    $D0/$E4/   {SHL   AH,1  }
    $D0/$E4/   {SHL   AH,1  }
    $D0/$E4/   {SHL   AH,1  }
    $A0/a/     {MOV   AL,[a]}
    $24/$0F/   {AND   AL,$0F }

```

```

$08/$E0/      {OR      AL,AH }
$A2/1/        {MOV      [1],AL}
$58           {POP      AX   }

);

end;

procedure beep;

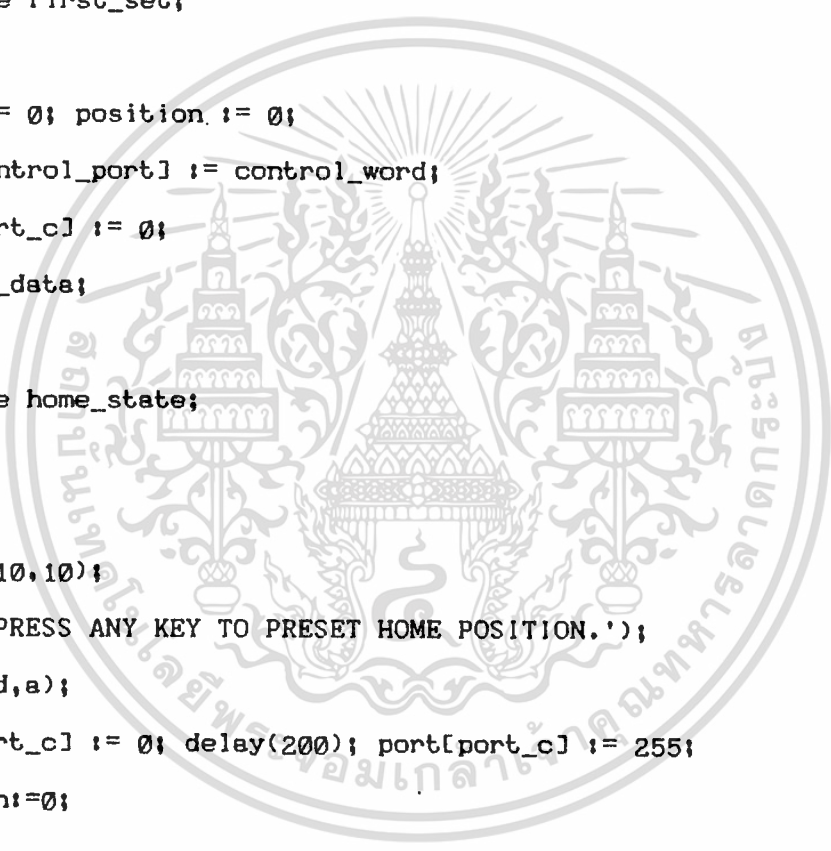
begin
    sound(1000); delay(100); nosound;
end;

procedure prepare_data;
begin
    clrscr;
    gotoxy(1,1); write('MEMORY CLEARING');
    vd_addr:= $B800;
    pls_stp:= 400; {max=400}
    p:=360/(pls_stp*4);
    for i:=0 to pls_stp*4-1 do begin
        r:=i*p;
        str(r:9:5,st);
        a:=copy(st,1,1); b:=copy(st,2,1);
        ppjj;
        data[i*4]:=1;
        a:=copy(st,3,1); b:=copy(st,5,1);
        ppjj;
        data[i*4+1]:=1;
        a:=copy(st,6,1); b:=copy(st,7,1);
        ppjj;
        data[i*4+2]:=1;
        a:=copy(st,8,1); b:=copy(st,9,1);

```

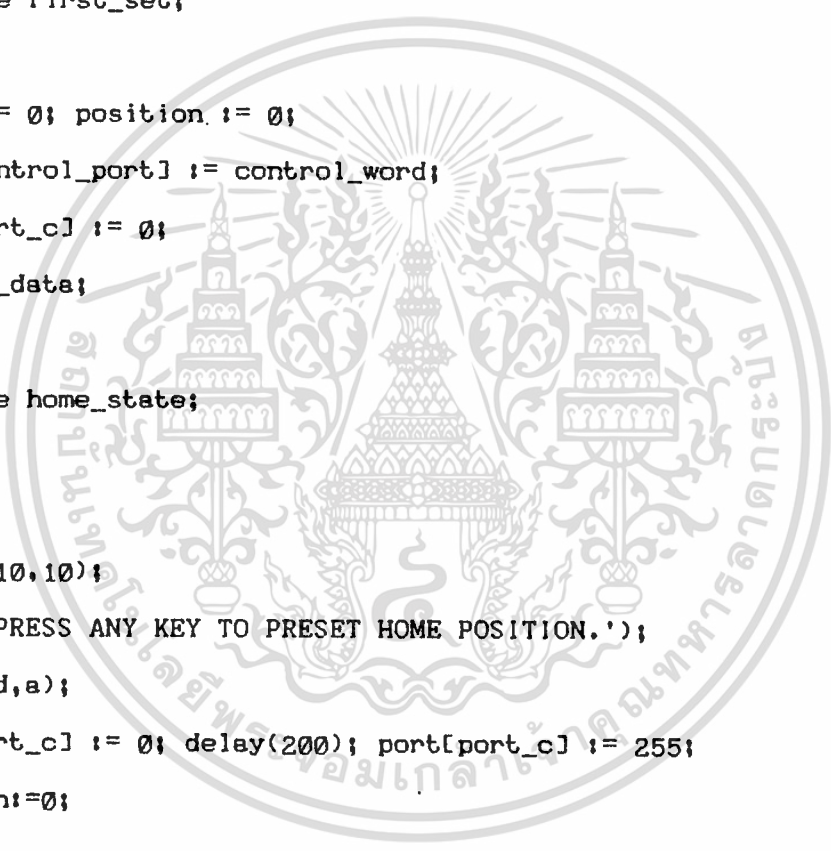
```

    ppjj;
    data[i*4+3]:=1;
end;
end;
procedure first_set;
begin
    state := 0; position := 0;
    port[control_port] := control_word;
    port[port_c] := 0;
    prepare_data;
end;
procedure home_state;
begin
    clrscr;
    gotoxy(10,10);
    write('PRESS ANY KEY TO PRESET HOME POSITION.');
```



```

    read(kbd,a);
    port[port_c] := 0; delay(200); port[port_c] := 255;
    position:=0;
end;
procedure title;
begin
    clrscr; gotoxy(40,11);
    writeln
    ('POSITION : ARM ANGLE IS NOW 000.000000 DEGREE.');
```



```

    writeln;
    writeln('THE NUMBER OF PULSE IN 1 CYCLE : ',PLS_STP);
    gotoxy(1,1);
    end;

```

```

procedure er;
begin
  beep;
  gotoxy(1,15); writeln('ERROR : over speed !');
  write('Press any key to continue. ');
  read(kbd,a);
  home_state; title;
end;
procedure cal;
begin
  repeat
    inline
  (
    $50/      {PUSH  AX      }
    $52/      {PUSH  DX      }
    $BA/$01/$03/ {MOV   DX,0301 }
    $EC/      {IN     AL,DX   }
    $24/$F0/    {AND    AL,F0   }
    $A3/state1/ {MOV    state1,AX}
    $89/$C2/    {MOV    DX,AX   }
    $A1/state/  {MOV    AX,state}
    $D0/$E8/    {SHR    AL,1    }
    $D0/$E8/    {SHR    AL,1    }
    $D0/$E8/    {SHR    AL,1    }
    $D0/$E8/    {SHR    AL,1    }
    $08/$D0/    {OR     AL,DL   }
    $24/$CC/    {AND    AL,CC   }
    $A3/state/  {MOV    state,AX}
    $5A/      {POP   DX      }
  )

```

```

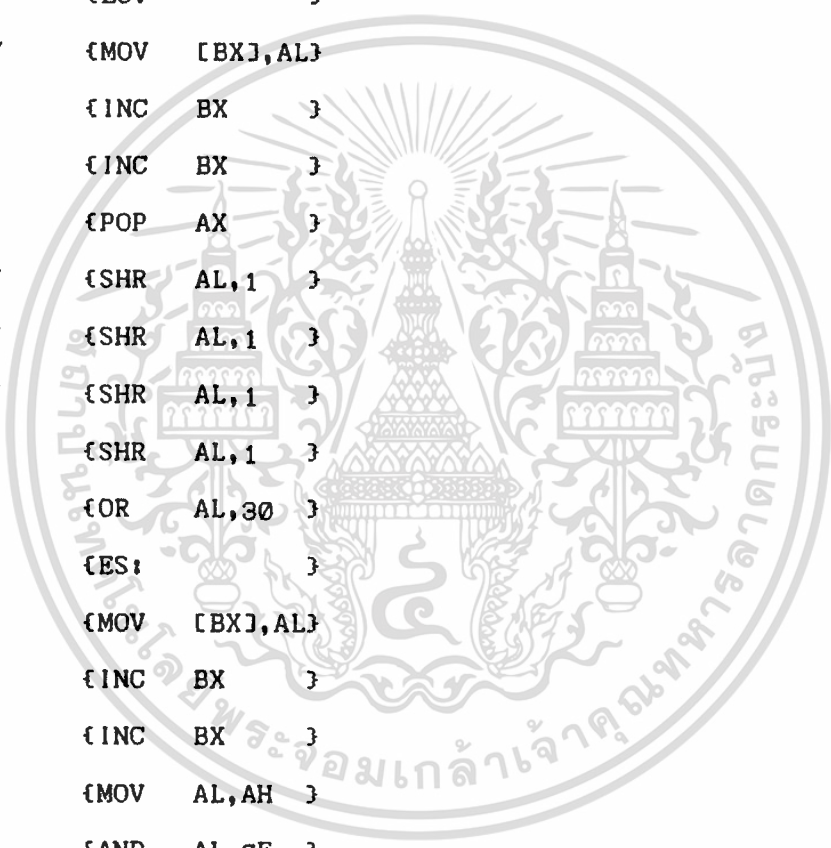
$58      {POP   AX      }
);
case state of
$40,$84,$C8,$0C :
begin
inline
(
$50/      {PUSH  AX      }
$A1/position/{MOV   AX,position}
$40/      {INC   AX      }
$3B/$06/pls_stp/
          {CMP   AX,pls_stp}
$75/$03/  {JNZ   3       }
$B8/$00/$00/{MOV   AX,0000}
$A3/position/{MOV   position,AX}
$58      {POP   AX      }
); end;
$04,$48,$8C,$C0 :
begin
inline
(
$50/      {PUSH  AX      }
$A1/position/{MOV   AX,position}
$3D/$00/$00/{CMP   AX,0000}
$75/$03/  {JNZ   3       }
$A1/pls_stp/{MOV   AX,pls_stp}
$48/      {DEC   AX      }
$A3/position/{MOV   position,AX}
$58      {POP   AX      }

```

```

);
end;
$08,$08,$4C,$C4 : er;
end;
inline
(
$50/      {PUSH  AX      }
$53/      {PUSH  BX      }
$06/      {PUSH  ES      }
$56/      {PUSH  SI      }
$A1/vd_addr/ {MOV    AX,vd_addr}
$8E/$C0/    {MOV    ES,AX  }
$A1/position/ {MOV    AX,position}
$D1/$E0/    {SHL    AX,1   }
$D1/$E0/    {SHL    AX,1   }
$8B/$1E/state1/
            {MOV    BX,state1}
$80/$E3/$30/ {AND    BL,30  }
$D0/$EB/     {SHR    BL,1   }
$D0/$EB/     {SHR    BL,1   }
$D0/$EB/     {SHR    BL,1   }
$D0/$EB/     {SHR    BL,1   }
$D0/$EB/     {SHR    BL,1   }
$B7/$00/     {MOV    BH,00   }
$01/$D8/     {ADD    AX,BX   }
$D1/$E0/     {SHL    AX,1   }
$D1/$E0/     {SHL    AX,1   }
$BE/data/    {MOV    SI,data }
$01/$C6/     {ADD    SI,AX   }
$BB/$92/$06/ {MOV    BX,0692}

```



```

$8B/$04/      {MOV    AX,[SI]}
$50/          {PUSH   AX    }
$24/$0F/      {AND    AL,0F  }
$0C/$30/      {OR     AL,30  }
$26/          {ES:      }
$88/$07/      {MOV    [BX],AL}
$43/          {INC     BX    }
$43/          {INC     BX    }
$58/          {POP     AX    }
$D0/$E8/      {SHR     AL,1  }
$D0/$E8/      {SHR     AL,1  }
$D0/$E8/      {SHR     AL,1  }
$D0/$E8/      {SHR     AL,1  }
$0C/$30/      {OR     AL,30  }
$26/          {ES:      }
$88/$07/      {MOV    [BX],AL}
$43/          {INC     BX    }
$43/          {INC     BX    }
$88/$E0/      {MOV     AL,AH  }
$24/$0F/      {AND     AL,0F  }
$0C/$30/      {OR     AL,30  }
$26/          {ES:      }
$88/$07/      {MOV    [BX],AL}
$43/          {INC     BX    }
$43/          {INC     BX    }
$43/          {INC     BX    }
$43/          {INC     BX    }
$D0/$EC/      {SHR     AH,1  }
$D0/$EC/      {SHR     AH,1  }

```

\$D0/\$EC/ {SHR AH,1 }
 \$D0/\$EC/ {SHR AH,1 }
 \$80/\$CC/\$30/ {OR AH,30 }
 \$26/ {ES: }
 \$88/\$27/ {MOV [BX],AH}
 \$43/ {INC BX }
 \$43/ {INC BX }
 \$46/ {INC SI }
 \$46/ {INC SI }
 \$8B/\$04/ {MOV AX,[SI]}
 \$50/ {PUSH AX }
 \$24/\$0F/ {AND AL,0F }
 \$0C/\$30/ {OR AL,30 }
 \$26/ {ES: }
 \$88/\$07/ {MOV [BX],AL}
 \$43/ {INC BX }
 \$43/ {INC BX }
 \$58/ {POP AX }
 \$D0/\$E8/ {SHR AL,1 }
 \$D0/\$E8/ {SHR AL,1 }
 \$D0/\$E8/ {SHR AL,1 }
 \$D0/\$E8/ {SHR AL,1 }
 \$0C/\$30/ {OR AL,30 }
 \$26/ {ES: }
 \$88/\$07/ {MOV [BX],AL}
 \$43/ {INC BX }
 \$43/ {INC BX }

\$88/\$E0/ {MOV AL,AH }

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

$0C/$30/      {OR    AL,30  }
$26/          {ES:      }
$88/$07/      {MOV    [BX],AL}
$43/          {INC    BX    }
$43/          {INC    BX    }
$D0/$EC/      {SHR    AH,1  }
$D0/$EC/      {SHR    AH,1  }
$D0/$EC/      {SHR    AH,1  }
$D0/$EC/      {SHR    AH,1  }
$80/$CC/$30/ {OR     AH,30  }
$26/          {ES:      }
$88/$27/      {MOV    [BX],AH}
$5E/          {POP     SI    }
$07/          {POP     ES    }
$5B/          {POP     BX    }
$58           {POP     AX    }
);

```

```
until keypressed;
```

```
port[port_c]:=0;
```

```
beep;
```

```
end;
```

```
{main() }
```

```
begin
```

```
  first_set;
```

```
  home_state;
```

```
  title;
```

```
  cal;
```

```
end.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผลจากการทดลองโปรแกรมนี้ สามารถแสดงผลค่าองศาของตัว เอนโคเดอร์ได้ทันที และอัตราการหมุนของเอนโคเดอร์ที่คอมพิวเตอร์สามารถรับรู้ได้ทัน อยู่ในอัตราที่สูงมาก

เนื่องจากต้องการความเร็วสูง เทคนิคการคำนวณก็ทำได้โดยใช้ ภาษาเครื่องช่วยในการประมวลผล และอีกเทคนิค คือ แทนที่ทำการคำนวณในขณะที่รับรู้ ซึ่งจะต้องคำนวณตลอดเวลา และมีโอกาสที่จะต้องคำนวณซ้ำตัวเดิม อีก ดังนั้น จึงใช้วิธีคำนวณก่อน และเก็บผลคำนวณไว้ในหน่วยความจำ เมื่อต้องประมวลผล ก็เพียงแต่ดึงผลการคำนวณซึ่งได้คำนวณไว้ล่วงหน้ามาแล้ว เป็นการลดเวลาในการคำนวณลงได้มาก จะได้ใช้เวลาที่เหลือเพิ่มความเร็ว ในการรับรู้ได้อีกมาก เทคนิคดังกล่าวข้างต้นนี้ใช้อยู่ในโปรแกรมด้วย

4.3 รูปแบบการเก็บข้อมูลในหน่วยความจำ

ในการสอน เมื่อมีการรับรู้ข้อมูลที่สอนด้วยวิธีการใด ๆ ก็ตาม ข้อมูลดังกล่าว จะต้องเก็บไว้ในหน่วยความจำที่สามารถเรียกใช้ได้ในภายหลัง วิธีการเก็บข้อมูลในหน่วยความจำ ก็มีความสำคัญไม่น้อย เพราะการเก็บที่เป็นมาตรฐานตามที่กำหนดไว้ จะทำให้สามารถเรียกการเรียกใช้ข้อมูลทำได้กว้างขวางขึ้น วิธีการเก็บข้อมูลจะมีผลต่อความสะดวกในการอ่าน รวมทั้งประสิทธิภาพในการใช้หน่วยความจำ รูปแบบการเก็บข้อมูลสามารถอธิบายแยกแยะเป็นข้อ ๆ ได้ดังนี้

4.3.1 การเก็บแบบโตเมนเวลา

หลักการเก็บก็คือ จะเก็บข้อมูลซึ่งอาจเป็นข้อมูลตำแหน่งหรือข้อมูลการจัด หรือข้อมูลอื่น ๆ ที่เวลาใด ๆ โดยเวลาจะมีคาบเวลาที่แน่นอน วิธีการนี้ไม่เหมาะสมอย่างยิ่งที่จะนำไปใช้กับข้อมูลที่เปลี่ยนแปลงกับเวลาน้อย หรือข้อมูลที่หยุดนิ่งอยู่กับที่นาน ๆ เช่นการสอนให้แขนกลพ่นสี หากเก็บข้อมูลการไม่ว่าการใด ๆ ที่สอน อีกหนึ่งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มาเก็บใช้

เคลื่อนที่แบบเก็บตำแหน่งโดยมีเวลาเป็นโดเมน ถ้ามีการหยุด ณ ตำแหน่งใดนาน ๆ ค่าตำแหน่งจะคงที่ในขณะที่เวลาก็ยังเปลี่ยนอยู่ ทำให้เปลืองหน่วยความจำมาก ที่จะต้องเก็บค่าตำแหน่งเดิมซ้ำ ๆ ตัวอย่างข้อมูลที่เก็บคือ

100 101 102 103 105 105 104 104 ...

ลักษณะการเก็บรูปแบบนี้ จะมีอัตราการใช้นหน่วยความจำต่อเวลาคงที่แน่นอน เช่น ความจำ 10 กิโลไบต์สามารถเก็บข้อมูลได้ 1 นาที เป็นต้น

4.3.2 การเก็บแบบโดเมนตำแหน่งหรือโดเมนการขจัด

ลักษณะข้อมูลที่เก็บจะเป็นข้อมูลเวลา และทิศทางการเคลื่อนที่ การเก็บข้อมูลลักษณะนี้มีข้อดีที่ การเปลืองของหน่วยความจำขึ้นอยู่กับว่ามีการเคลื่อนที่ไปมากน้อยเพียงไร ไม่ขึ้นกับเวลา จึงเหมาะกับข้อมูลการเคลื่อนที่ที่มีลักษณะเดิน ๆ หยุด ๆ สำหรับข้อเสียของวิธีการนี้ก็คือ ความไม่แน่นอนของเวลาที่หยุด ณ ตำแหน่งใด ๆ สมมติให้ข้อมูลเวลามีหน่วยเป็นมิลลิวินาที เมื่อมีการเคลื่อนที่ไป ข้อมูลเวลาที่หยุด ณ ตำแหน่งใด ๆ จะมีค่าน้อย อาจจะ เป็น 10 หรือ 20 มิลลิวินาที แต่เมื่อหยุดเคลื่อนที่ที่ตำแหน่งใด ๆ นาน ๆ ค่าเวลาอาจจะ เป็น 10000 มิลลิวินาที ทำให้มีความแตกต่างของข้อมูลมาก การจะจองเนื้อที่หน่วยความจำเพื่อเก็บค่าเวลานั้นต้องคำนึงถึงปัจจัยหลายอย่าง ถ้าจองไว้ 1 ไบต์ คือข้อมูลมีค่า 0 ถึง 255 ก็จะไม่เพียงพอหากมีการหยุดที่ใดนาน ๆ หรือจองไว้ 2 ไบต์ ค่าตั้งแต่ 0 ถึง 65000 อาจจะมากเกินไป เพราะส่วนใหญ่แล้ว ข้อมูลมีค่าไม่มาก จะมีบางตำแหน่งเท่านั้นที่มีค่ามาก การเลือกจองพื้นที่ 1 ไบต์ หรือ 2 ไบต์นั้น เป็นผลให้หน่วยความจำเปลืองแตกต่างกันถึง 2 เท่าตัวที่ข้อมูลเท่า ๆ กัน ลักษณะปัญหาดังกล่าว จะเป็นปัญหา มากสำหรับการสอนของเครื่องที่ต้องหยุดที่ตำแหน่งใดนาน ๆ แต่ไม่บ่อย เช่น เครื่องขั่นนอตอัตโนมัติ สอนด้วยวิธีจับสอน การขั่นนอต เมื่อเคลื่อนที่ไป ณ ตำแหน่งที่จะขั่นนอต ก็จะต้องหยุด ณ ตำแหน่งนั้น ๆ นาน อาจเป็นจำนวน เวลาหลายล้านเท่าเมื่อเทียบกับเวลาที่หยุด ณ ตำแหน่งที่เคลื่อนที่ผ่านไปเฉย ๆ ยิ่งถ้ามีจำนวนนอตที่ต้องการขั่นมาก ๆ ยิ่งต้องใช้การสอนที่นานและเคลื่อนที่ มาก การจะเพิ่มหน่วยความจำเพื่อให้สามารถจดจำการสอนได้หมดเป็นเรื่อง ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีให้คิดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สิ้นเปลืองมาก เทคนิคที่จะแก้ปัญหาดังกล่าวอยู่ในหัวข้อต่อไป ตัวอย่างข้อมูลที่เก็บในหน่วยความจำด้วยวิธีนี้คือ

1 05 1 10 1 0E 1 07 0 3A 0 5C 0 02 ...

โดยตัวเลขหน้าหมายถึงทิศทางที่เคลื่อนที่ที่เคลื่อนมาถึงที่ตำแหน่งนี้ และตัวเลขท้ายคือ จำนวนเวลาที่หยุดที่ตำแหน่งนั้น ๆ

ตัวอย่างข้างต้นเป็นการเก็บข้อมูลเวลา โดยใช้ขนาดข้อมูลเวลาเท่ากับ 1 ไบต์ ตัวอย่างต่อไปเป็นการใช้ขนาด 2 ไบต์

1 0002 1 0004 1 000A 1 ABCD 0 010E ...

ในการทดลอง ได้ออกแบบโปรแกรมขึ้นมาเพื่อศึกษาวิธีการเก็บข้อมูลแบบนี้ ดังแสดงโปรแกรมดังนี้

proj7.pas

const

port_a : integer = 768;

port_b : integer = 769;

port_c : integer = 770;

control_port : integer = 771;

control_word : byte = \$41;

var

a : char;

i,j,k,rltm : integer;

state,position_m : integer;

vd_addr,pls_stp_m: integer;

pm :array [0..7] of byte;

tmps :array [0..1000] of byte;

tmpt :array [0..1000] of integer;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure first_set;
begin
    k:=0; state := 0; position_m:=0;
    port[control_port] := control_word;
    pm[0]:=$10; pm[1]:=$50; pm[2]:=$40; pm[3]:=$60;
    pm[4]:=$20; pm[5]:=$A0; pm[6]:=$80; pm[7]:=$90;
    port[port_a]:=0; port[port_c]:=255;
    vd_addr:= $B800; pls_stp_m:= 8;
end;

procedure home_state;
begin
    clrscr; gotoxy(10,10);
    write('PRESS ANY KEY TO PRESET HOME POSITION.');
```



```

    read(kbd,a);
    port[port_c] := 0; delay(200);
end;

procedure title;
begin
    clrscr;
    gotoxy(40,10); writeln('Teaching mode.');
```



```

    sound(1000); delay(100); nosound;
end;

procedure er;
begin
    sound(1000); delay(100); nosound;
    gotoxy(1,15); writeln('ERROR : over speed !');
```



```

    write('Press ctrl-c to exit.');
```



```

    repeat read(kbd,a); until a='#';
end;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
procedure cal;
```

```
begin
```

```
repeat
```

```
inline
```

```
(
```

```
$50/      {PUSH  AX      }
```

```
$52/      {PUSH  DX      }
```

```
$BA/$01/$03/ {MOV    DX,$0301 }
```

```
$EC/      {IN     AL,DX    }
```

```
$24/$F0/   {AND    AL,F0    }
```

```
$89/$C2/   {MOV    DX,AX    }
```

```
$A1/state/ {MOV    AX,state }
```

```
$D0/$E8/   {SHR    AL,1     }
```

```
$D0/$E8/   {SHR    AL,1     }
```

```
$D0/$E8/   {SHR    AL,1     }
```

```
$D0/$E8/   {SHR    AL,1     }
```

```
$08/$D0/   {OR     AL,DL    }
```

```
$24/$CC/   {AND    AL,CC    }
```

```
$A3/state/ {MOV    state,AX }
```

```
$5A/      {POP    DX      }
```

```
$58       {POP    AX      }
```

```
);
```

```
case state of
```

```
$40,$84,$C8,$0C :
```

```
begin inline
```

```
(
```

```
$50/      {PUSH  AX      }
```

```
$A1/position_m/{MOV    AX,position_m}
```

```
$40/      {INC    AX      }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

$3B/$06/pls_stp_m/
        {CMP    AX,pls_stp_m}
$75/$03/    {JNZ    3        }
$B8/$00/$00/ {MOV    AX,0000}
$A3/position_m/{MOV    position_m,AX}
$58        {POP    AX        }
);
tmps[k]:=position_m; tmppt[k]:=rltm;
port[port_a]:=pm[tmps[k]];
inline(
$50/        {PUSH    AX        }
$A1/k/      {MOV     AX,k      }
$40/        {INC     AX        }
$A3/k/      {MOV     k,AX      }
$B8/$00/$00/ {MOV     AX,0000}
$A3/rltm/   {MOV     rltm,AX}
$58        {POP     AX        }
);
end;
$04,$48,$8C,$C0 :
begin inline
(
$50/        {PUSH    AX        }
$A1/position_m/{MOV     AX,position_m}
$3D/$00/$00/ {CMP     AX,0000}
$75/$03/     {JNZ     3        }
$A1/pls_stp_m/{MOV     AX,pls_stp_m}
$48/        {DEC     AX        }
$A3/position_m/{MOV     position_m,AX}

```

```

$58      {POP   AX      }
);
tmps[k]:=position_m; tmpt[k]:=rltm;
port[port_a]:=pm[tmps[k]];
inline(
$50/      {PUSH  AX      }
$A1/k/     {MOV   AX,k    }
$40/      {INC   AX      }
$A3/k/     {MOV   k,AX    }
$B8/$00/$00/ {MOV   AX,0000}
$A3/rltm/  {MOV   rltm,AX}
$58      {POP   AX      }
);
end;
$08,$08,$4C,$C4 : er;
end;
inline(
$50/      {PUSH  AX      }
$A1/rltm/  {MOV   AX,rltm}
$40/      {INC   AX      }
$A3/rltm/  {MOV   rltm,AX}
$58      {POP   AX      }
);
if rltm=0 then er;
until keypressed;
port[port_a]:=0; port[port_c]:=0;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure dsp;
begin
  for i:=0 to k do begin
    writeln(tmp[1]);
    port[port_a]:=pm[tmp[1]];
    for j:=1 to tmp[1] do begin end;
  end;
end;
{main()}
begin
  first_set;
  home_state;
  port[port_c] := 255;
  title;
  cal;
  home_state;
  dsp;
  port[port_a] := 0;
end.

```

เป็นโปรแกรมที่ตรวจจับสัญญาณที่ได้จากการหมุนของเอนโคเดอร์ และทำการบันทึกค่าข้อมูลเวลาที่หยุดที่ตำแหน่งใด ๆ คือเวลาที่เอนโคเดอร์ไม่หมุน ในขณะที่บันทึกค่าและรับค่าต่อไป ก็จะบังคับให้สเตปมอเตอร์เคลื่อนที่ เหมือนกับที่เอนโคเดอร์เคลื่อนที่ พร้อม ๆ กันไปเพื่อเป็นการเช็คข้อมูลของผู้สอนไปในตัว การบันทึกข้อมูลจะให้ตัวแปรที่บันทึกมีขนาดเท่ากับ 2 ไบต์ คือ ตัวแปร tmp[] ในโปรแกรม

4.3.3 การเก็บแบบใช้เทคนิคพิเศษ

ปัญหาการเปลี่ยนแปลงหน่วยความจำที่ใช้เก็บข้อมูลที่สอนนั้น แก้ไขได้โดย เปลี่ยนแปลงวิธีการเก็บที่ใช้ตำแหน่งเป็นโดเมน โดยแทนที่จะจองพื้นที่เก็บข้อมูลของเวลาให้คงที่ในทุก ๆ ตำแหน่ง ก็เปลี่ยนเป็นว่า จำนวนข้อมูลของเวลาที่จะเก็บนั้น ขึ้นอยู่กับว่าเวลาที่จะเก็บในตำแหน่งนั้น ๆ มากหรือน้อย ถ้ามีค่าเวลาน้อย ก็เก็บเพียง 1 ไบท์ก็พอ แต่ถ้าเวลามากเกินกว่าที่จะเก็บใน 1 ไบท์ ก็เพิ่มจำนวนไบท์ที่ตำแหน่งนั้น ๆ เป็น 2 ไบท์ ถ้ายังไม่พอก็เพิ่มจำนวนไบท์ขึ้นเรื่อย ๆ ไม่จำกัด จะลดปัญหาการเปลี่ยนแปลงหน่วยความจำได้ แต่ปัญหาที่จะตามมาก็คือ ความยุ่งยากและซับซ้อนในการเขียนโปรแกรม เนื่องจากรูปแบบในการเก็บไม่คงที่เลย เก็บไบท์เดียวบ้าง เก็บหลายไบท์บ้าง ตัวโปรแกรมก็ต้องรู้ว่า ตอนนี้นับเวลาถึงค่าเท่าไรแล้ว ต้องใช้หน่วยความจำกี่ไบท์ในการเก็บ ซึ่งยุ่งยากพอสมควร แต่ก็ประหยัดหน่วยความจำไปได้มาก และสามารถให้หน่วยความจำที่จำกัด เก็บข้อมูลที่มาก ๆ ได้อย่างมีประสิทธิภาพ ตัวอย่างการเก็บข้อมูลลงในหน่วยความจำโดยการใช้วิธีนี้ แสดงดังนี้

1 01 1 04 1 4E 1 0100 0 020000 0 02 ...

หมายถึงขนาดของข้อมูลของเวลาที่จะเก็บนั้น ไม่คงที่แน่นอน ขึ้นอยู่กับว่าขนาดเวลาจะมากหรือน้อย เช่นในตัวอย่าง มีทั้งการเก็บแบบใช้ 1 ไบท์ แบบใช้ 2 ไบท์ และแบบใช้ 3 ไบท์ ในการเก็บข้อมูลของเวลา

ในการทดลอง ได้ออกแบบโปรแกรมขึ้นมาเพื่อศึกษาวิธีการเก็บแบบวิธีการพิเศษนี้ ตัวโปรแกรมมีดังนี้

```
cseg segment para public 'code'
org 100h
assume cs:cseg,ds:cseg
```

```
proj5 proc near
```

```
mov si,0
```

```
ps: mov bx,0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    push  bx
    push  bx
    pop   cx
    pop   dx
p1  mov   ah,state
    shr   ah,1
    shr   ah,1
    and   ah,30h
    push  dx
    mov   dx,port
    in    al,dx
    pop   dx
    and   al,0c0h
    or    al,ah
    mov   state,al
    cmp   al,00
    jz    nw
    cmp   al,50h
    jz    nw
    cmp   al,0a0h
    jz    nw
    cmp   al,0f0h
    jz    nw
    cmp   al,10h
    jz    fw
    cmp   al,60h
    jz    fw
    cmp   al,0b0h
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        cmp     al,0c0h
        jz      fw
        cmp     al,30h
        jz      bw
        cmp     al,40h
        jz      bw
        cmp     al,90h
        jz      bw
        cmp     al,0e0h
        jz      bw
        jmp     final_p
nw:     inc     bx
        jnz     nw1
        inc     cx
        jnz     nw1
        inc     dx
        jz      fwf
nw1:    push    cx
        mov     cx,100h
nw2:    loop    nw2
        pop     cx
        mov     ah,1
        int     16h
        jz      p1
        jmp     final_p
fw:     mov     al,10h
        cmp     dl,0
        jnz     fwa
        cmp     ch,0

```

```
    jnz    fwb
    cmp    cl,0
    jnz    fwc
    cmp    bh,0
    jnz    fwd
    cmp    bl,40h
    jl     fw1
    jmp     fw2
bw:   mov    al,10h
    cmp    dl,0
    jnz    bwa
    cmp    ch,0
    jnz    bwb
    cmp    cl,0
    jnz    bwc
    cmp    bh,0
    jnz    bwd
    cmp    bl,40h
    jl     bw1
    jmp     bw2
fwd:   cmp    bh,al
    jl     fw2
    jmp     fw3
fwc:   cmp    cl,al
    jl     fw3
    jmp     fw4
fwb:   cmp    ch,al
    jl     fw4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

fwa:  cmp    dl,al
      j1     fw5
fwf:  jmp     final_p
bwd:  cmp    bh,al
      j1     bw2
      jmp    bw3
bwc:  cmp    cl,al
      j1     bw3
      jmp    bw4
bwb:  cmp    ch,al
      j1     bw4
      jmp    bw5
bwa:  cmp    dl,al
      j1     bw5
      jmp    final_p
fw1:  or     bl,40h
      mov    ax,bx
      push   bx
      mov    bx,OFFSET pdata
      jmp    fw51
fw2:  or     bh,0c0h
      mov    ax,bx
      push   bx
      mov    bx,OFFSET pdata
      jmp    fw52
fw3:  or     cl,0d0h
      push   bx
      mov    bx,OFFSET pdata
      jmp    fw53

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

fw4:  or    ch,0e0h
      push  bx
      mov   bx,OFFSET pdata
      jmp   fw54

bw1:  mov   ax,bx
      push  bx
      mov   bx,OFFSET pdata
      jmp   fw51

bw2:  or    bh,80h
      mov   ax,bx
      push  bx
      mov   bx,OFFSET pdata
      jmp   fw52

bw3:  or    cl,90h
      push  bx
      mov   bx,OFFSET pdata
      jmp   fw53

bw4:  or    ch,0a0h
      push  bx
      mov   bx,OFFSET pdata
      jmp   fw54

fw5:  or    dl,0f0h
      push  bx
      mov   bx,OFFSET pdata
      mov   [bx+si],dl
      inc   si

fw54: mov   [bx+si],ch
      inc   si

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า.
 ไม่ควรแก้ไขสิ่งอื่น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        inc     si
fw52:   mov     [bx+si],ah
        inc     si
fw51:   mov     [bx+si],al
        inc     si
        pop     bx
        cmp     si,pdmax
        jnl     final_p
        jmp     ps
bw5:    or      dl,0b0h
        push    bx
        mov     bx,OFFSET pdata
        mov     [bx+si],dl
        inc     si
        jmp     fw54
final_p:
        mov     pdnum,si
port    dw      300h
pdmax   dw      1000
pdnum   dw      0
state   db      0
pdata   db      1007 dup(0)
proj5   endp
cseg    ends
        end
```

โปรแกรมดังกล่าวเป็นเพียงแค่โปรแกรมน้อยเท่านั้น ยังต้องมีโปรแกรมส่วนอื่นทำงานร่วมด้วย โปรแกรมนี้ ทำหน้าที่รับข้อมูลตำแหน่งเข้ามา และนับเวลาที่หยุดอยู่ที่ตำแหน่งนั้น ๆ เมื่อมีการเปลี่ยนแปลงตำแหน่งก็จะหยุดนับเวลาและตรวจสอบเวลาที่นับได้ว่าต้องใช้จำนวนไบนารีในการเก็บเท่าไร จากนั้นก็จะเป็นการบันทึกข้อมูลลงในหน่วยความจำ โดยข้อมูลจะเริ่มต้นด้วยข้อมูลทิศทางการเคลื่อนที่จำนวน 1 บิต และข้อมูลจำนวนไบนารีทั้งหมดที่เก็บอีก 3 บิต ดังนั้นข้อมูลเวลาที่ใช้เก็บต้องน้อยกว่า 8 ไบนารี ซึ่งมากเกินไป ข้อมูลไบนารีต่อไปก็จะเป็นข้อมูลของเวลา ดังนั้น วิธีการในโปรแกรมนี้นี้จึงเป็นวิธีการเก็บข้อมูลเวลาที่ประหยัดมาก แต่การอ่านข้อมูลกลับทีหลัง จะต้องเข้าใจวิธีการเก็บข้อมูลด้วยจึงจะสามารถเข้าใจข้อมูลและนำไปใช้ได้

โปรแกรมนี้นี้เขียนขึ้นด้วยภาษาเครื่อง เพราะการตัดสินใจว่าจะเก็บในลักษณะที่ไบนารีนั้น ต้องตัดสินใจด้วยความรวดเร็ว หากใช้ภาษาสูง จะทำให้เวลาที่ใช้ไปกับการตัดสินใจจะไปทำให้การนับเวลารวมของข้อมูล เกิดความผิดพลาดของเวลานับขึ้นได้

4.4 การเรียกใช้ข้อมูลภายหลัง

เมื่อสามารถรับรู้การสอนได้ และสามารถเก็บข้อมูลที่ได้จากการสอนได้ ก็ต้องสามารถที่จะอ่านข้อมูลนั้นกลับออกมา และทำงานตามข้อมูลนั้นได้ จึงจะเป็นการสอนซึ่งสมบูรณ์ได้

หากว่าการเก็บข้อมูลอยู่ในลักษณะภาพ ซึ่งไม่เกี่ยวข้องกับเวลา เช่นการเก็บข้อมูลของภาพต้นแบบในการตัดแผ่นเหล็ก ข้อมูลภาพจะไม่ขึ้นกับเวลา การเรียกใช้ข้อมูลก็เพียงแต่นำข้อมูลภาพที่ได้ มาเปลี่ยนเป็นเส้นแนวทางการเคลื่อนที่แทน หรือถ้ามีข้อมูลการเคลื่อนที่ในลักษณะที่ไม่ขึ้นกับเวลา

เช่นการวาดรูปของพลอตเตอร์ ข้อมูลการเคลื่อนที่ดังกล่าว ก็จะสามารถนำมาใช้งานได้เลย

ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในงานบางงาน ข้อมูลที่ได้มีลักษณะที่แปรตามเวลา เช่น การผันสีของแขนกล บางจุดต้องพ่นนาน บางจุดต้องพ่นแบบให้ผ่านไปเร็ว ๆ หรือผันสีให้บาง ลักษณะการเคลื่อนที่จึงต้องขึ้นกับเวลา การเรียกใช้ข้อมูลจึงต้องคำนึงถึงเวลาด้วย การจะเรียกใช้ข้อมูลลักษณะนี้ ต้องมีการติดต่อกับหรือมีมาตรฐานอันเดียวกันของตัวสอนและตัวทำงาน คือนอกจากต้องมีสเกลที่อ้างอิงกันได้แล้วยังต้องมีคาบเวลาอ้างอิงกันได้ด้วย มิฉะนั้น จะเกิดความผิดพลาดขึ้นได้เพราะสเกลตำแหน่งหรือสเกลคาบเวลาไม่สอดคล้องกัน

4.5 โปรแกรมประมวลผลที่สมบูรณ์

เป็นโปรแกรมที่จะรับข้อมูลจากตัวเอนโคเดอร์เข้ามา โดยตัวเอนโคเดอร์จะมีจำนวนพัลส์เท่าใดก็ได้ ไม่ได้กำหนดแน่นอน ในขณะที่รับข้อมูลจากตัวเอนโคเดอร์เข้ามา ก็จะบังคับให้สเตปมอเตอร์เคลื่อนที่เหมือนกับเอนโคเดอร์ แต่สามารถบอกอัตราส่วนของรอบของเอนโคเดอร์กับรอบของสเตปมอเตอร์ได้ด้วย จะมีการบันทึกข้อมูลการเคลื่อนที่ เก็บไว้ในหน่วยความจำ สามารถเรียกกลับมาทำงานใหม่ได้ โดยสามารถเปลี่ยนอัตราส่วนหรือสเกลได้ตามต้องการ และสามารถเร่ง ลด เวลาการเคลื่อนที่ให้เร็วขึ้นหรือช้าลงก็ได้ โปรแกรมแสดงดังนี้

Program proj9.pas

Manual control, Teaching mode and Ratio scale.

By Jaroonsakdi Wangviwattana 281045 4E

const

port_a : integer = \$300;

port_b : integer = \$301;

port_c : integer = \$302;

control_word : byte = \$41;

control_port : integer = \$303;

เอกสารนี้เป็นเอกสารสงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

var
  a,ass : char;
  b1,b2 : byte;
  vd_addr : integer;
  p,q,r,ratio : real;
  position : integer;
  act_count : integer;
  i,j,k,m,n : integer;
  delay_time : integer;
  phase_step_num : integer;
  state_step_num : integer;
  pm : array [0..16] of byte;
  state_encoder_num : integer;
  state_encoder_pos : integer;
  state,state1,state2 : integer;
  p_data : array [0..6399] of byte;
  motion_data_pos : array [0..1000] of byte;
  motion_data_time : array [0..1000] of integer;

```

```

procedure first_set;

```

```

begin

```

```

  clrscr;

```

```

  port[control_port] := control_word;

```

```

  port[port_a]:=0;

```

```

  port[port_c]:=0;

```

```

  vd_addr:=$B800;

```

```

  write('Encoder      : pulse per cycle = ');

```

```

  read(m);

```

```

  state_encoder_num:=4*m;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามแก้ไขเพิ่มเติมเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

writeln;
write('Step motor : number of phase (max=8) = ');
read(phase_step_num);
writeln;
write('          (half step) step per cycle = ');
read(state_step_num);
writeln;
write('Cycle ratio : (motor:encoder) = 1:');
read(ratio);
writeln;
write('Step motor run when teaching (Y/N) ?');
read(kbd,ass);
writeln(ass);
writeln;
end;

procedure prepare_data;
begin
  case phase_step_num of
    3:
      begin
        pm[0]:=$10;
        pm[1]:=$50;
        pm[2]:=$40;
        pm[3]:=$60;
        pm[4]:=$20;
        pm[5]:=$30;
        pm[6]:=$10;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

4:
begin
  pm[0]:=$10;
  pm[1]:=$50;
  pm[2]:=$40;
  pm[3]:=$60;
  pm[4]:=$20;
  pm[5]:=$A0;
  pm[6]:=$80;
  pm[7]:=$90;
  pm[8]:=$10;
end;
end;
m:=2*round(m*ratio/2);
j:=2*round(state_encoder_num*ratio/2);
r:=state_step_num/j;
k:=phase_step_num*2;
p:=r/k;
for i:=0 to j-1 do begin
  p_data[i]:=round(k*(p*i-trunc(p*i)));
end;
act_count:=0;
position :=0;
state :=0;
state1:=0;
state2:=0;
b1:=0;
b2:=0;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่าในรูปแบบใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure home_state;
begin
  writeln('PRESS ANY KEY TO PRESET HOME POSITION.');
```



```

  read(kbd,a);
  port[port_c] := 0;
  delay(200);
end;
```

```

procedure er;
begin
  port[port_a]:=0;
  port[port_c]:=0;
  sound(1000);
  delay(100);
  nosound;
  gotoxy(1,15);
  writeln('ERROR : over speed !');
```



```

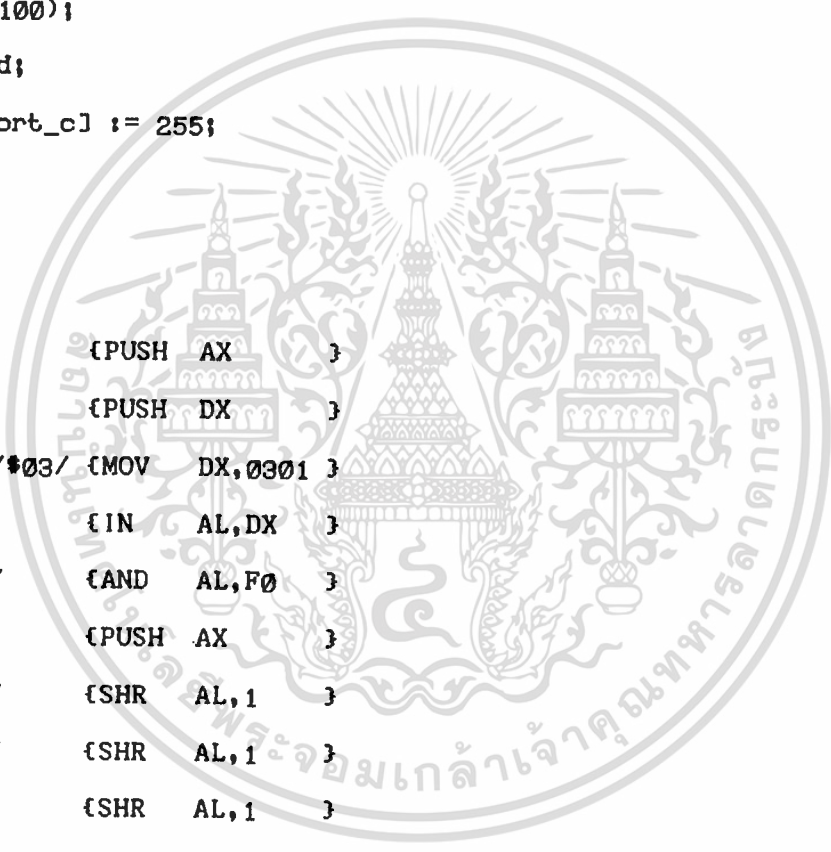
  write('Press ctrl-c to exit.');
```

```

  repeat
    read(kbd,a);
  until a='#';
end;
```

```

procedure cal;
begin
  writeln('Teaching mode.');
```



```

  sound(1000);
  delay(100);
  nosound;
  port[port_c] := 255;
  repeat
    inline
  (
    $50/      {PUSH  AX      }
    $52/      {PUSH  DX      }
    $BA/$01/$03/ {MOV    DX,0301 }
    $EC/      {IN     AL,DX   }
    $24/$F0/   {AND    AL,F0   }
    $50/      {PUSH  AX      }
    $D0/$E8/   {SHR    AL,1   }
    $D0/$E8/   {SHR    AL,1   }
    $D0/$E8/   {SHR    AL,1   }
    $D0/$E8/   {SHR    AL,1   }
    $24/$03/   {AND    AL,03   }
    $A3/state1/ {MOV    state1,AX}
    $5A/      {POP    DX      }
    $A1/state/  {MOV    AX,state}
    $D0/$E8/   {SHR    AL,1   }
    $D0/$E8/   {SHR    AL,1   }
    $D0/$E8/   {SHR    AL,1   }
    $D0/$E8/   {SHR    AL,1   }
    $08/$D0/   {OR     AL,DL   }
  )

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

$24/$CC/      {AND    AL,CC    }
$A3/state/    {MOV     state,AX}
$5A/          {POP     DX      }
$58           {POP     AX      }
};

```

case state of

\$40,\$84,\$C8,\$0C :

begin inline

```

(
$50/          {PUSH    AX      }
$A1/position/{MOV     AX,position}
$40/          {INC     AX      }
$3B/$06/m/    {CMP     AX,m    }
$75/$03/      {JNZ     3       }
$B8/$00/$00/  {MOV     AX,0000}
$A3/position/
               {MOV     position,AX}
$58           {POP     AX      }
);
end;

```

\$04,\$48,\$8C,\$C0 :

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

(
$50/      {PUSH  AX      }
$A1/position/
          {MOV   AX,position}
$3D/$00/$00/ {CMP   AX,0000}
$75/$03/     {JNZ   3      }
$A1/m/       {MOV   AX,m    }
$48/         {DEC   AX      }
$A3/position/
          {MOV   position,AX}
$58         {POP   AX      }
);
end;
$08,$08,$4C,$C4 : er;
end;
if state1<>state2 then
  b1:=p_data[position*4+state1];
  if b1<>b2 then
    begin
      i:=b1-b2;
    case i of
      {For 4 phases step motor}
      3,4,5,-3,-4,-5 : er;
    end;
    b2:=b1;
    motion_data_pos[act_count]:=b1;
    motion_data_time[act_count]:=delay_time;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    delay(5);
end;

inline(
$50/      {PUSH  AX      }
$A1/state1/ {MOV    AX,state1}
$A3/state2/ {MOV    state2,AX}
$A1/act_count/
            {MOV    AX,act_count}
$40/      {INC    AX      }
$A3/act_count/
            {MOV    act_count,AX}
$B8/$00/$00/ {MOV    AX,0000}
$A3/delay_time/
            {MOV    delay_time,AX}
$58      {POP    AX      }
);
end;

if delay_time>$7FF0 then
begin
    motion_data_pos[act_count]:=b1;
    motion_data_time[act_count]:=delay_time;
    inline
    (
$50/      {PUSH  AX      }
$A1/act_count/
            {MOV    AX,act_count}
$40/      {INC    AX      }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งยังมีเหตุเปลี่ยนแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

$B8/$00/$00/ {MOV    AX,0000}
$A3/delay_time/
                {MOV    delay_time,AX}
$58            {POP     AX      }
);
end;
inline(
$50/           {PUSH    AX      }
$A1/delay_time/
                {MOV     AX,delay_time}
$40/           {INC     AX      }
$A3/delay_time/
                {MOV     delay_time,AX}
$58            {POP     AX      }
);
until keypressed;
port[port_a]:=0;
port[port_c]:=0;
end;
procedure dsp;
begin
  for i:=0 to act_count do begin
    port[port_a]:=pm[motion_data_pos[i]];
    writeln(motion_data_time[i]);
    for j:=1 to motion_data_time[i] do
      begin
        if keypressed then read(kbd,a);
        p:=sin(5)*cos(5)*exp(1);
        port[port_a]:=p;
      end;
    end;
  end;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end;  
end;  
{main()}  
begin  
  first_set;  
  prepare_data;  
  home_state;  
  cal;  
  home_state;  
  dsp;  
  port[port_a]:=0;  
end.
```

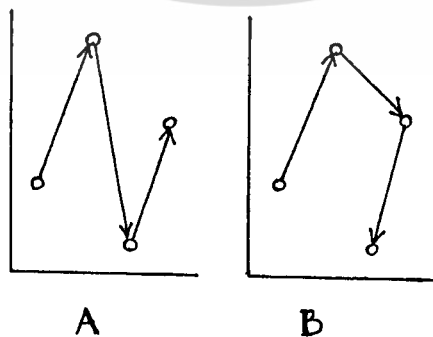


5. การประยุกต์ใช้งาน

จากที่ได้กล่าวรายละเอียดทั้งหมดของวิธีการต่าง ๆ ต่อไปจะเป็นการนำเอาวิธีการและเทคนิคทั้งหมด มาหาความเป็นไปได้ที่จะนำมาใช้ในอุตสาหกรรมจริง ๆ

5.1 เครื่องเจาะแผ่นวงจรอัตโนมัติ

ในอุตสาหกรรมการผลิตแผ่นวงจร การเจาะรูเลียบอุปกรณ์บนแผ่นวงจรเป็นขั้นตอนหนึ่งในการผลิต เมื่อมีการผลิตจำนวนมาก การเจาะโดยใช้พนักงานย่อมเสียเวลา อีกทั้งยังได้งานที่ไม่ดี มีการหล่นลิ่มเจาะไปบางจุดบ้าง หรือเจาะไม่ตรงบ้าง



รูปที่ 5.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้เฉพาะที่ควรศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
การเคลื่อนที่ของหัวเจาะลักษณะต่าง ๆ
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการแรกที่จะเสนอก็คือ การใช้การเขียนโปรแกรมการทำงานเป็นขั้นตอนตามลำดับ โดยการเขียนโปรแกรมบังคับให้หัวเจาะเคลื่อนที่ไปยังจุดที่จะเจาะ แล้วสั่งให้เจาะ จากนั้นก็สั่งให้เคลื่อนที่ไปยังจุดต่อไป ในการใช้เครื่องคอมพิวเตอร์ประมวลผลนั้น สามารถทำให้เครื่องคอมพิวเตอร์มีความสามารถมากยิ่งขึ้น เช่น ไม่จำเป็นต้องสั่งให้เคลื่อนที่ตามลำดับ เพียงแต่บอกพิกัดของจุดที่จะเจาะทั้งหมดเท่านั้น จากนั้น เครื่องคอมพิวเตอร์ก็จะหาทิศทาง การเคลื่อนที่ระหว่างจุดเจาะหนึ่งไปยังอีกจุดที่ดีที่สุด โดยจะหาระยะทางที่เคลื่อนที่รวม น้อยที่สุด เพื่อเสียเวลาในการเจาะต่อแผ่นให้น้อยที่สุด ตัวอย่างแสดงไว้ในรูปที่ 5.13 จะเห็นว่าทิศทางแบบ B ใช้เวลาในการเจาะรวมน้อยกว่าแบบ A เป็นต้น

จากที่กล่าวมา จะเห็นได้ว่า วิธีการที่จะสอนให้เครื่องเจาะแผ่นวงจรที่ดีที่สุดคือ การเขียนโปรแกรมสั่งงานตามลำดับ หรือการป้อนพิกัดจุดเจาะทั้งหมดให้ สำหรับการควบคุมด้วยมือ นั้น เกิดประโยชน์น้อยกว่า หรือการสอนด้วยการจับสอนนั้น อาจจะได้ผลงานที่ไม่แม่นยำ เพราะเครื่องสามารถรับรู้พิกัดจากตัวเลขได้แน่นอนกว่าพิกัดที่คนจับสอน

5.2 เครื่องตัดแผ่นเหล็กอัตโนมัติ

ในอุตสาหกรรมที่ต้องมีการตัดแผ่นเหล็กนั้น บ้างครั้ง จำเป็นต้องตัดแผ่นเหล็กให้ได้ตามแบบที่ต้องการจำนวนมาก ๆ การใช้คนควบคุมย่อมทำให้ชิ้นงานที่ออกมาไม่แน่นอน วิธีการที่จะเสนอวิธีการแรกคือ การใช้การเขียนโปรแกรมสั่งงาน สั่งให้หัวตัดแผ่นเหล็ก เคลื่อนที่ไปในแนวกรอบรูปต้นแบบ เพื่อตัดแผ่นเหล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วิธีการเขียนโปรแกรมสั่งงานการเคลื่อนที่ของเครื่องตัดแผ่นเหล็กนั้น จะไม่เหมาะสมถ้ารูปแบบที่ได้มา มีลักษณะของเส้นโค้ง หรือลักษณะของเส้นที่ไม่เป็นรูปแบบทางเรขาคณิต ผู้ที่เขียนโปรแกรมสั่งงานหรือโปรแกรมการเคลื่อนที่ จะต้องพยายามเขียนให้การเคลื่อนที่อยู่ในแนวเส้นกรอรูปแบบ ซึ่งค่อนข้างจะต้องใช้เวลาในการเขียน ยิ่งถ้าหากว่ามีการขยายหรือลดรูป อาจจะทำให้เกิดปัญหาของสเกลได้

ดังนั้นวิธีการที่เหมาะสมสำหรับกรณีข้างต้นก็คือ ใช้เทคนิคการอ่านภาพต้นแบบ วิธีการอ่านภาพอาจจะใช้การกวาดภาพทั้งหมด แล้วนำไปประมวลผลภาพ หรือ การใช้หัวอ่านภาพเคลื่อนที่ไปตามแนวเส้นกรอรูป และบันทึกการเคลื่อนที่ไว้ในหน่วยความจำ หรือ ชิปซ้อนขึ้นมา ก็ใช้กล้องถ่ายภาพ หรือ กล้องวิดีโอ เพื่อเก็บข้อมูลภาพนำมาประมวลผลด้วยคอมพิวเตอร์ สุดท้ายก็จะได้ข้อมูลการเคลื่อนที่ และนำไปให้เครื่องตัดทำงานตามข้อมูลการเคลื่อนที่ต่อไป

คอมพิวเตอร์สามารถจะออกแบบโปรแกรมให้มีความสามารถขยายหรือย่อภาพได้โดยใช้วิธีการในหัวข้อข้างต้น ดังนั้น รูปแบบก็ไม่จำเป็นที่จะต้องเท่ากับขนาดของชิ้นงาน และอัตราส่วนการย่อขยายก็ไม่จำเป็นที่จะต้องเป็นอัตราส่วนลงตัว

5.3 แขนกลในอุตสาหกรรม

อุตสาหกรรมบางอย่าง ต้องใช้ความปราณีตและฝีมือของพนักงานผู้ชำนาญงาน เช่นการฟันลิ่วสตุ การขัดเกลารูปทรงที่ต้องการความสวยงาม การวาดรูปของนักศิลปะ เหล่านี้หากต้องทำชิ้นงานจำนวนมาก ๆ เหมือน ๆ กันแล้ว การใช้ผู้ชำนาญงานทำกับมีย่อมได้ผลงานน้อย แขนกลจึงมีบทบาทขึ้นมา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้แขนกลทำงานที่ปราณีตหรือที่ต้องมีศิลปนั้น แขนกลจะทำได้ดีหรือไม่ดีขึ้นอยู่กับผู้เขียนโปรแกรมสั่งงานแขนกล เช่นการพ่นสีวัสดุ ผู้เขียนโปรแกรมต้องพยายามเขียนโปรแกรมให้แขนกลพ่นสีได้ และได้ดีเท่าที่ผู้ชำนาญงานทำเองกับมือ ดังนั้น จึงเป็นการยุ่งยากมาก เพราะผู้เขียนโปรแกรมต้องลงไปศึกษาวิธีการพ่นสีเอง จึงสามารถเขียนโปรแกรมให้แขนกลพ่นสีได้ แต่อย่างไรก็ตาม เทคนิควิธีการพ่นสีให้สวยงามนั้น ผู้ชำนาญงานเท่านั้นที่รู้ เพราะต้องอาศัยการเรียนรู้จากประสบการณ์ แต่ถ้าจะให้ผู้ชำนาญงานการพ่นสีมาเขียนโปรแกรมเองก็ย่อมเป็นไปได้

การจับแขนกลสอนให้พ่นสีในตัวอย่างดังกล่าว ย่อมเป็นทางออกที่ดี อุปกรณ์ที่ใช้ก็อาจจะเป็นส่วนหนึ่งของข้อต่อที่ทำเลียนแบบแขนกลด้วยขนาดเดียวกัน และติดตัวเอนโคเดอร์หรือตัวความต้านทานปรับค่าได้ไว้ตามข้อต่อต่าง ๆ จากนั้นผู้ชำนาญงานก็จะจับข้อต่อนี้ทำงาน เหตุที่ไม่ติดตัวอุปกรณ์รับรู้ไว้กับแขนกลจริง ๆ เลยก็เพราะ ในแขนกลนั้นมีความผิดของมอเตอร์ที่ติดอยู่และมีน้ำหนักค่อนข้างมาก ดังนั้นเมื่อผู้ชำนาญงานจับสอน อาจจะทำให้ผู้ชำนาญงานทำงานไม่ได้ดี เพราะต้องยกของหนักและต้องออกแรงมากในการเคลื่อนที่แขนกล จึงใช้ข้อต่อที่มีขนาดเท่าแขนกลจริงในการจับสอนจะทำให้ผู้ชำนาญงานได้ใช้ความชำนาญของตัวเองอย่างเต็มที่

เมื่อผู้ชำนาญงานสอนให้แขนกลทำงานแล้ว ก็จะให้แขนกลลองทำงานดู ถ้าไม่ดีก็สอนใหม่ แล้วเลือกการทำงานครั้งที่ดีที่สุดเก็บไว้ นั่นคือ เราสามารถเก็บข้อมูลการเคลื่อนที่ของผู้ชำนาญงานในขณะทำงานได้ผลดีที่สุดไว้ และเก็บข้อมูลที่ดีที่สุดนี้เป็นมาตรฐานการทำงาน ดังนั้น ถ้าอุตสาหกรรมใหญ่ ๆ ที่มีแขนกลมาก แขนกลทุกเครื่องก็จะทำงานได้ดีที่สุดตลอดไป ผู้ชำนาญงานก็ไม่มี ความจำเป็นอีกต่อไป หรือไม่ต้องจ้างผู้ชำนาญงานมามาก ๆ ก็ได้ จ้างมาเพียงคนเดียว และทำงานเพียงไม่กี่ครั้ง เพื่อหาครั้งที่ดีที่สุด จากนั้นก็ไม่จำเป็นต้องจ้างอีกจนกว่าจะมีการเปลี่ยนแปลงข้อมูล

6. บทสรุป

จากการศึกษาถึงวิธีการในการสอนให้เครื่องจักรอัตโนมัติสามารถทำงานได้ตามต้องการนั้น ได้ค้นคว้าและทดลองจริงในวิธีการต่าง ๆ หลาย ๆ วิธี ทำให้รู้ถึงข้อดีและข้อเสียของแต่ละวิธีการ ตั้งแต่การเลือกใช้อุปกรณ์รับรู้ การเลือกวิธีการสอน เทคนิคการเก็บข้อมูลในหน่วยความจำ และเทคนิคการประมวลผลข้อมูล

วิธีการรับรู้และอุปกรณ์รับรู้ สามารถแบ่งเป็น การรับรู้แบบมือหรือแบบแมนวล ซึ่งจะเหมาะในงานไม่ซับซ้อน การรับรู้แบบข้อมูลโปรแกรมสั่งงาน จะเหมาะกับงานที่ค่อนข้างง่ายและแน่นอน การรับรู้แบบข้อมูลภาพ จะเหมาะกับงานที่มีรูปแบบไม่เป็นเรขาคณิต หรือไม่สามารถเขียนโปรแกรมสั่งงานตามลำดับได้ การรับรู้ข้อมูลแบบการจับสอน เหมาะกับงานที่ต้องใช้ฝีมือ ความชำนาญ หรือเป็นงานที่เป็นศิลปะ

การประมวลผล สามารถใช้คอมพิวเตอร์ช่วยในการประมวลผล ซึ่งอาจจะเป็นการย่อขยายสเกล หรือการทำความเข้าใจรูปแบบ และแปลงรูปแบบเป็นข้อมูลการเคลื่อนที่ การประมวลผลนี้ขึ้นอยู่กับความต้องการของผู้ใช้เป็นหลัก

การเก็บข้อมูลที่สอนไว้ในหน่วยความจำ และสามารถนำกลับมาใช้ในครั้งต่อ ๆ ไปได้ วิธีการเก็บข้อมูลมีหลายรูปแบบ รูปแบบการเก็บแบบใช้เวลาเป็นโดเมนข้อมูล จะเหมาะสำหรับงานที่ไม่ซับซ้อนและไม่คำนึงถึงขนาดของข้อมูล รูปแบบการเก็บแบบใช้ตำแหน่งหรือการจัดเป็นโดเมนก็จะเหมาะกับงานที่ต้องการประหยัดหน่วยความจำ และยังมีรูปแบบพิเศษอื่น ๆ ไม่ว่าจะเป็นวิธีใด ๆ ทั้งสิ้น อีกทั้งยังมีให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งหาก นำไปใช้

ขึ้นอยู่กับความต้องการของผู้ใช้ เช่นต้องการใช้หน่วยความจำที่มีอยู่อย่างจำกัด ให้สามารถเก็บข้อมูลให้ได้มากที่สุด

ในโครงงานนี้ ได้ทำการสรุปวิธีการต่าง ๆ และได้ทดลองจริง เพื่อให้ได้ข้อมูลของวิธีการต่าง ๆ อย่างละเอียด เพื่อเป็นประโยชน์ในการศึกษาค้นคว้าในเรื่องการสอน การรับรู้ และการประมวลผลต่าง ๆ ของเครื่องจักรอัตโนมัติต่อไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

๗. หนังสืออ้างอิง

๗.๑ หนังสืออ้างอิงภาษาไทย

[ท๑] รศ.ดร.โยธิน เปรมปรางค์รัตน์ ;

"สแตมป์มอเตอร์ในระบบควบคุมตำแหน่ง" ;

วารสารคอมพิวเตอร์ อิเล็กทรอนิกส์ เวิลด์ ฉบับที่ ๙๙ ถึง ๑๐๓ , ๒๕๒๙.

[ท๒] ยืน ภู่วรรณ ;

"ไอซีทีทีแอล" ;

ซีเอ็ดยูเคชั่น, กรุงเทพฯ, ๒๕๒๑.

[ท๓] นนทวัฒน์ จุลเดช, ประสิทธิ์ จุลเสวีวงศ์, รศ.ดร.โยธิน เปรมปรางค์รัตน์ ;

"เทคนิคการวัดความเร็วของมอเตอร์

ในระบบการควบคุมมอเตอร์โดยไมโครคอมพิวเตอร์" ;

การสัมมนาทางวิชาการสาขาวิศวกรรมไฟฟ้า ครั้งที่ ๖, สงขลา นครินทร์.

7.2 หนังสืออ้างอิงภาษาอังกฤษ

[1] Alan R. Miller ;

"Assembly Language Techniques for the IBM PC" ;

SYBEX inc.,Berkeley,1986.

[2] Andrew C.Staugaard,Jr. ;

"Robotics and AI ;

And introduction to applied machine intelligence" ;

Prentice-Hall inc.,1987.

[3] B.C.Kuo, A.Cassat ;

"Step Motors and Control Systems ;

Closed-Loop Control By Current Sensing" ;

[4] Billy K.Walker ;

"A Structured Approach to PASCAL" ;

Irwin inc.,Illinois,1983.

[5] Borland ;

"TURBO PASCAL version 3.0 reference manual" ; USA,1985.

[6] Bryan Norris ;

"Digital Integrated Circuits and Operational-Amplifier
and Optoelectronic Circuit Design"

McGraw-Hill inc.,1976.

[7] Carl Helmers ;

"Robotics age : in the beginning" ;Hayden book inc.,1983.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[8] Chan S.Park ;
"Interactive Microcomputer Graphics" ;
USA, september 1985.

[9] IBM ;
"Technical Reference" ;
Personal Computer Hardware Reference Library. ;
USA, April 1983.

[10] Jacob Millman ;
"Microelectronics ;
Digital and Analog Circuits and Systems" ;
McGraw-Hill inc., 1983.

[11] K.S.Fu, R.C.Gonzalez, C.S.G.Lee ;
"Robotics : Control, Sensing, Vision, and Intelligent" ;
McGraw-Hill inc., 1987.

[12] Leo J. Scanlon ;
"IBM PC Assembly Language : A Guide for Programmers" ;
Robert J.Brady Co., Bowie, 1983.

[13] Lewis C.Eggebrecht ;
"Interfacing to the IBM personal computer" ;
USA, 1983.

[14] Martin Bradley Winston ;
"Opto 'Whiskers' for a Mobile Robot" ;

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
Robotics Age, Volume 3, Number 1, January/February 1981.
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

[15] Rafael C.Gonzalez, Paul Wintz ;

"Digital Image Processing" ;

Addison-Wesley Publishing Co.,November,1987.

[16] Richard Oliver ;

"Improve Photo Sensors with a Phase Locked Loop IC," ;

Electronics Design News, April 5,1976,p.112.

[17] Richard Tseng ;

"Continuous path control with stepping motors" ;

Robotics Age,Volume 3,Number 5,September/October 1981.

[18] Rodney Zaks, Austin Lesea ;

"Microprocessor Interfacing Techniques" ;

[19] Russell Rector-George Alexy ;

"THE 8086 BOOK ; includes the 8088" ;

Osborne/McGraw-Hill,Berkeley,1980.

[20] Shafi Motiwalla ;

"Continuous path control with stepping motors" ;

Robotics Age,Volume 3,Number 4,July/August 1981.

[21] Takashi Kenjo ;

"Stepping motors and their microprocessor controls" ;

Clarrendon press,Oxford,1984

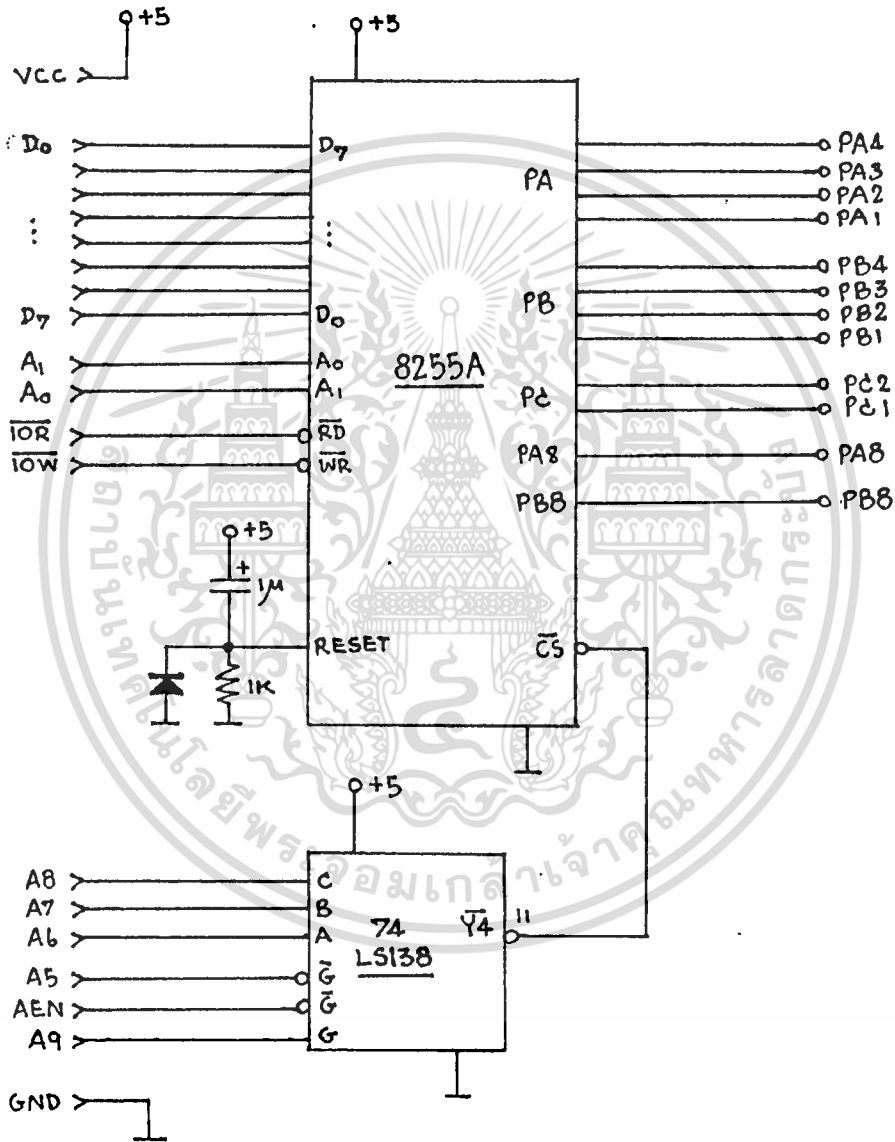
๘. กิตติกรรมประกาศ

ผู้เขียนขอกราบขอบพระคุณครูอาจารย์ทุก ๆ ท่าน ที่ได้ประสิทธิ์ประสาทวิชาความรู้และประสบการณ์ให้ผู้เขียน ขอขอบพระคุณอาจารย์โยธิน เปรมปรางค์ รัชต์ ที่คอยให้คำปรึกษาและข้อเสนอแนะตลอดมา ขอขอบคุณอาจารย์วันชัย รุ่งรุจา ที่คอยดูแลและกระตุ้นอยู่เสมอ ขอขอบคุณภาควิชาระบบควบคุมที่ได้เอื้อเฟื้ออุปกรณ์และสถานที่ในการทดลอง สุดท้ายขอขอบพระคุณคุณพ่อและคุณแม่ของผู้เขียน ที่อุดหนุนทุนในการทำวิทยานิพนธ์ฉบับนี้

9. ภาคผนวก

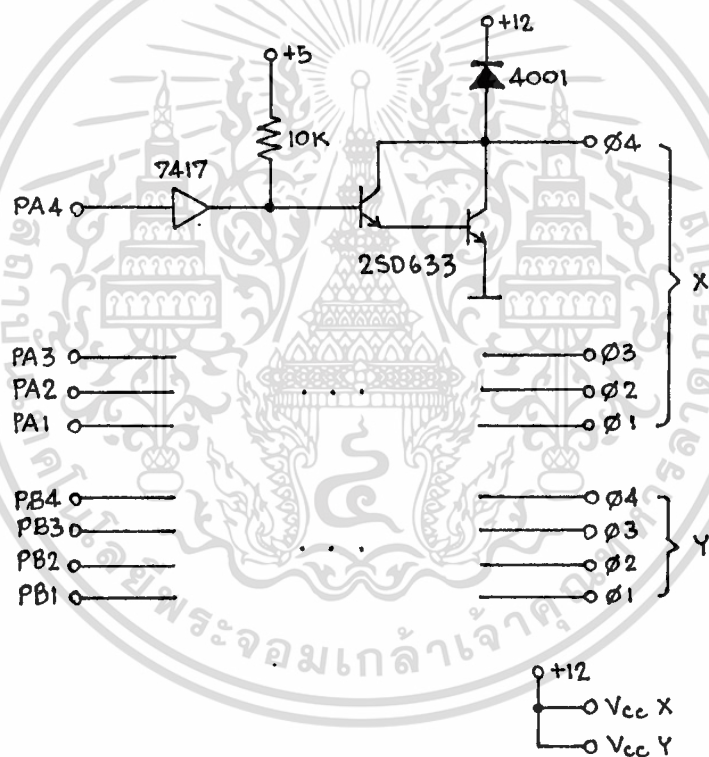
9.1 วงจรที่ใช้ในการทดลอง

- รูปที่ 7.1 : วงจรติดต่ออุปกรณ์ภายนอกกับเครื่อง IBM-XT
- รูปที่ 7.2 : วงจรขับสเตปมอเตอร์ชนิด 4 เฟส 2 ตัว
- รูปที่ 7.3 : วงจรประมวลผลข้อมูลจากหัวอ่านภาพ
- รูปที่ 7.4 : ขี้อสายต่อของตัวเอนโคดเดอร์
- รูปที่ 7.5 : วงจรแปลงรหัสจากเอนโคดเดอร์เป็นรหัสไบนารี 4 บิต
- รูปที่ 7.6 : วงจรตรวจจับสัญญาณที่ได้จากการหมุนสเตปมอเตอร์



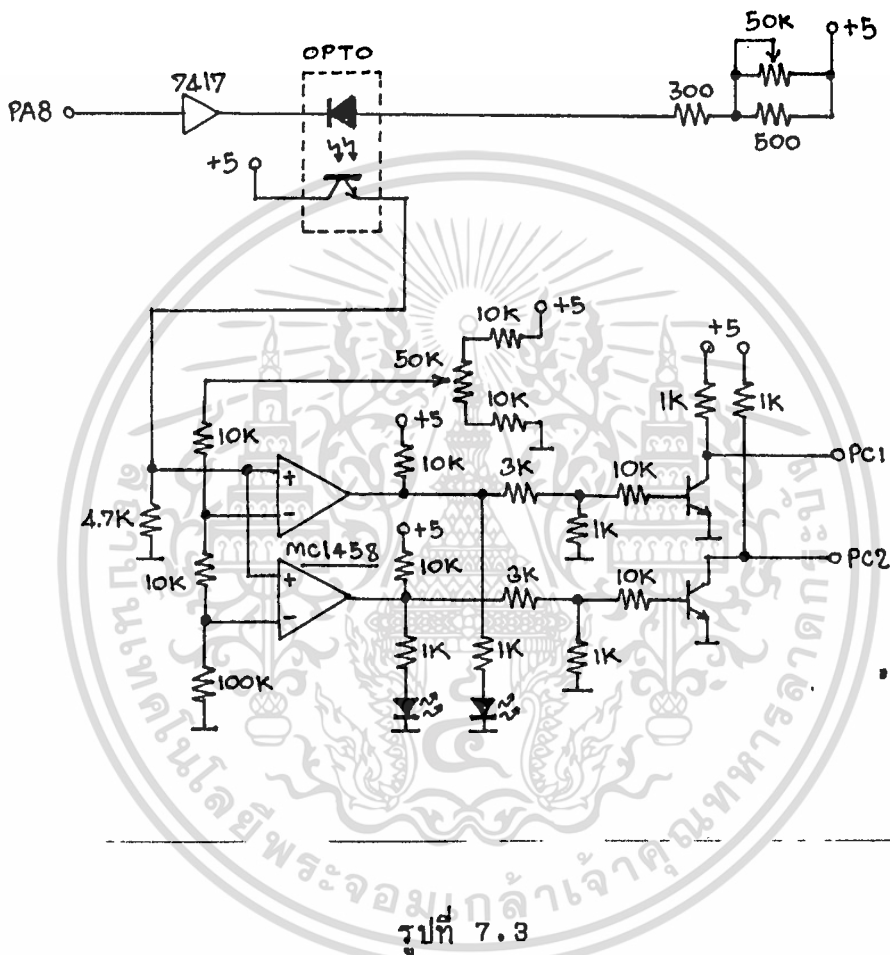
รูปที่ 7.1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 วงจรติดต่อกับอุปกรณ์ภายนอกกับเครื่อง IBM-XT
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



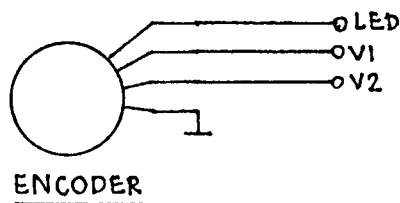
รูปที่ 7.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
วงจรขั้วลบเตปมอเตอรชนิด 4 เฟส 2 ตัว
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



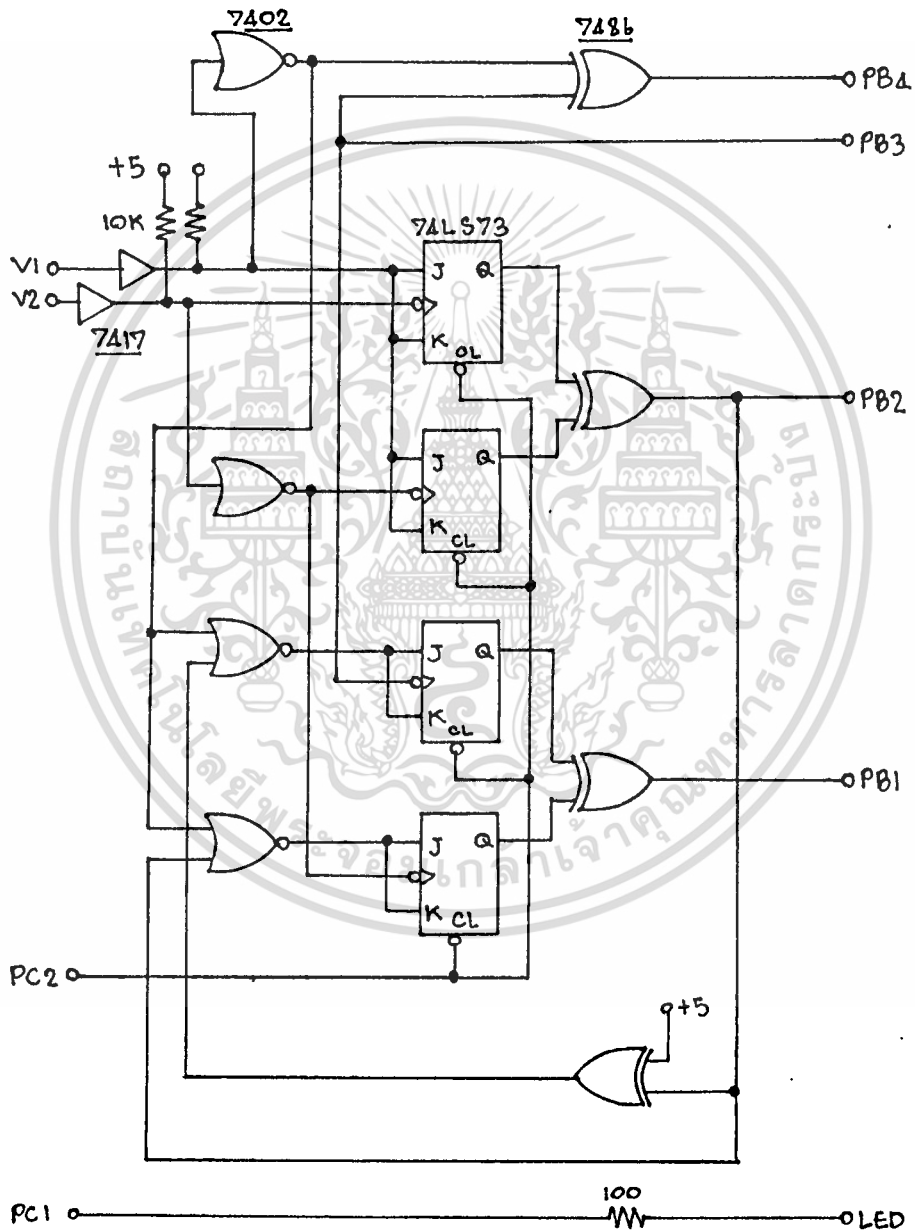
รูปที่ 7.3

วงจรประมวลผลข้อมูลจากหัวอ่านภาพ



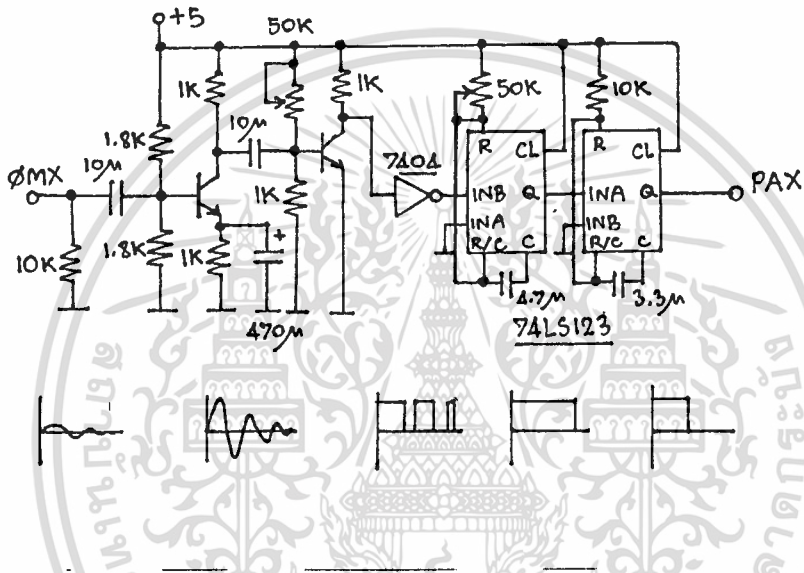
รูปที่ 7.4

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ชื่อสายต่อของตัวเอนโคเดอร์
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
วงจรแปลงรหัสจากเอนโคเดอร์เป็นรหัสไบนารี 4 บิต
 ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7.6

วงจรตรวจจับสัญญาณที่ได้จากการหมุนสแตปมอเตอร์

9.2 เอกสารแสดงรายละเอียดของอุปกรณ์

5400/7400 Quadruple 2-Input Positive-NAND Gate

	Schottky TTL				High-Speed TTL				Low-Power Schottky TTL				Standard TTL				Low-Power TTL			
	Device Type				Device Type				Device Type				Device Type				Device Type			
	C	P	M	CF	C	P	M	CF	C	P	M	CF	C	P	M	CF	C	P	M	CF
T.I.	SN54S00				SN54H00				SN54LS00				SN5400				SN54L00			
FAIRCHILD	FM54S00/FM5400				FM54H00/FM5400				FM54LS00/FM5400				FM5400							
MOTOROLA	MC74S00				MC74H00				MC74LS00				MC7400							
N.S.C.	DM74S00				DM74H00				DM74LS00				DM7400				DM74L00			
PHILIPS	N74S00				N74H00				N74LS00				N7400							
SIGNETICS	N74S00				N74H00				N74LS00				N7400							
SIEMENS																				
FUJITSU					MB501				74LS00				MB400							
HITACHI	HD74S00								HD74LS00				HD7400							
mitsubishi	M55S00								M74LS00				M53200							
NEC	μPB2S00								74LS00				μPB201							
TOSHIBA													TD3400 A							

Electrical Characteristics SN54LS00/SN74LS00

absolute maximum ratings over operating free-air temperature range

Supply voltage, V _{CC}	TV	Operating free-air temperature range	SN54LS	-55°C to 125°C
Input voltage	TV		SN74LS	0°C to 75°C
Interconnect voltage	5.5V	Storage temperature range		-55°C to 150°C

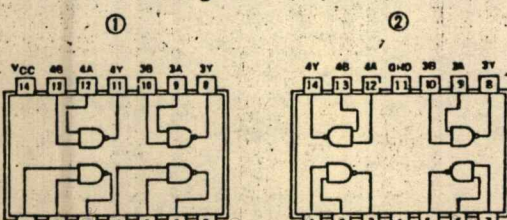
recommended operating conditions

	SN54LS00			SN74LS00			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V _{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I _{OH}			-600			-600	μA
Low-level output current, I _{OL}			4			8	mA
Operating free-air temperature, T _A	-55		125	0		75	°C

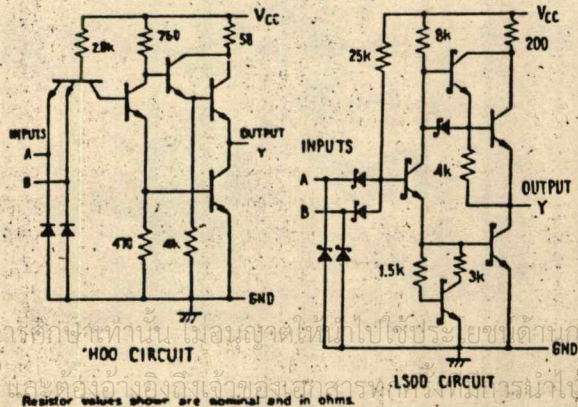
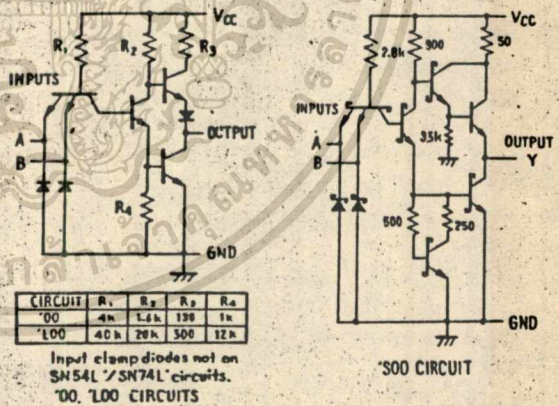
electrical characteristics over recommended operating free-air temperature range

PARAMETER	TEST CONDITIONS†	MIN	TYP‡	MAX	UNIT
V _{IH}	High-level input voltage		2		V
V _{IL}	Low-level input voltage			0.8	V
V _I	Input clamp voltage	V _{CC} =MIN, I _I =-18 mA		-1.5	V
V _{OH}	High-level output voltage	V _{CC} =MIN, V _{IL} =V _{IL} max, I _{OH} =MAX	2.7	3.4	V
V _{OL}	Low-level output voltage	V _{CC} =MIN, V _{IH} =2V, I _{OL} =4mA	0.2	0.4	V
I _I	Input current at maximum input voltage	V _{CC} =MAX, V _I =TV		0.1	mA
I _{IH}	High-level input current	V _{CC} =MAX, V _{IH} =2.7V		20	μA
I _{IL}	Low-level input current	V _{CC} =MAX, V _{IL} =0.4V		-0.4	mA
I _{OS}	Short-circuit output current	V _{CC} =MAX	54LS Family -20 74LS Family -18	-100	mA
I _{CCM}	Supply current	V _{CC} =MAX	Total, outputs high	4	mA
I _{CCL}	Supply current	V _{CC} =MAX	Total, outputs low	52	mA
I _{CC}	Supply current	V _{CC} =5V	Average per gate (50% duty cycle)	8.4	mA
t _{PLH}	Propagation delay time, low-to-high-level output	V _{CC} =5V, T _A =25°C, C _L =15pF, R _L =2kΩ		9	ns
t _{PHL}	Propagation delay time, high-to-low-level output			10	ns

Pin Assignments (Top View)



Schematics (each gate)



† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.
‡ All typical values are at V_{CC}=5V, T_A=25°C.
§ Not more than one output should be shorted at a time, and for SN54H/SN74H and SN54S/SN74S, duration of short-circuit should not exceed 1 second.

5402/7402 Quadruple 2-Input Positive-NOR Gate

	Schottky TTL					High-Speed TTL					Low-Power Schottky TTL					Standard TTL					Low-Power TTL									
	Device Type		Package			Device Type		Package			Device Type		Package			Device Type		Package			Device Type		Package							
			C	P	M			CF	C	P			M	CF	C			P	M	CF			C	P	M	CF	C	P	M	CF
T. I.	SN54S02	J	1		WD						SN54LS02	J	1		WD			SN5402	J	1		WD	SN54L02	J	1	N	I	T	Z	
	SN74S02	J	1	N	1						SN74LS02	J	1	N	1			SN7402	J	1	N	1		SN74L02	J	1	N	1	T	Z
FAIRCHILD	MC5402 / MC7402	DD			FD						MC54LS02 / MC74LS02	DD			FD			FM5402 / FM7402	DD			FD								
	PC1402 / PC7402	DD	PD		FD						PC14LS02 / PC74LS02	DD	PD		FD			FC7402 / FC9402	DD	PD		FD								
MOTOROLA																		MC5402	L	1		FD								
																		UC7402	L	1	PD		FD							
N. S. C.																		DM5402	J	1	N	1	WD	DM54L02	J	1	N	1		FD
																		DM7402	J	1	N	1		DM74L02	J	1	N	1		FD
PHILIPS	N74S02			1							N74LS02			1				FJH221/7402				1								
																		SS402	F	1	A	1	WD							
SIGNETICS	N74S02			A	1						N74LS02			A	1			N7402	F	1	A	1								
SIEMENS																		FLH191				1								
FUJITSU																														
																		74LS02	AK	1				MB417	1	AK	1			
HITACHI	HD74S02			P	1						HD74LS02			P	1			HD7402/HD2511	1		PD									
MITSUBISHI																														
																		M74LS02	P	1				MS3202		P	1			
NEC																														
TOSHIBA																														

Electrical Characteristics SN54LS02/SN74LS02

absolute maximum ratings over operating free-air temperature range

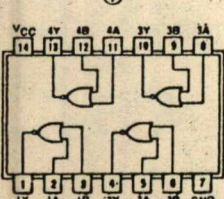
Supply voltage, V_{CC}	TV	Operating free-air temperature range	SN54LS	-55°C to 125°C					
Input voltage	TV	Storage temperature range	SN74LS	-65°C to 150°C					
recommended operating conditions									
			SN54LS02	SN74LS02	UNIT				
			MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}			4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			-400			-400			mA
Low-level output current, I_{OL}			-400			-400			mA
Operating free-air temperature, T_A			-55	125		0	75		°C

electrical characteristics over recommended operating free-air temperature range

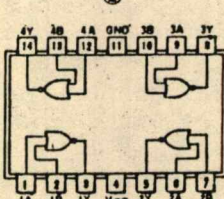
PARAMETER	TEST CONDITIONS†	MIN	TYP‡	MAX	UNIT
V_{IH} High-level input voltage		2			V
V_{IL} Low-level input voltage			0.8		V
V_I Input clamp voltage	$V_{CC} = \text{MIN.}$, $I_I = -18 \text{ mA}$			-1.5	V
V_{OH} High-level output voltage	$V_{CC} = \text{MIN.}$, $V_{IL} = V_{IL \text{ max.}}$, $I_{OH} = \text{MAX.}$	2.7	3.4		V
V_{OL} Low-level output voltage	$V_{CC} = \text{MIN.}$, $V_{IH} = 2 \text{ V}$, $I_{OL} = 4 \text{ mA}$	0.25	0.4		V
I_I Input current at maximum input voltage	$V_{CC} = \text{MAX.}$, $V_I = 2 \text{ V}$		0.1		mA
I_{IH} High-level input current	Data inputs, $V_{CC} = \text{MAX.}$, $V_{IH} = 2.7 \text{ V}$		20		μA
I_{IL} Low-level input current	Data inputs, $V_{CC} = \text{MAX.}$, $V_{IL} = 0.4 \text{ V}$		0.4		mA
I_{OS} Short-circuit output current‡	$V_{CC} = \text{MAX.}$	54LS Family	-20	-100	mA
		74LS Family	-20	-100	mA
I_{CCH} Supply current	$V_{CC} = \text{MAX.}$	Total, outputs high	1.6	3.2	mA
I_{CCL} Supply current	$V_{CC} = \text{MAX.}$	Total, outputs low	2.8	5.4	mA
I_{CCP} Supply current	$V_{CC} = 5 \text{ V}$	Average per gate (50% duty cycle)	0.58		mA
t_{PLH} Propagation delay time, low-to-high-level output	$V_{CC} = 5 \text{ V}$, $T_A = 25^\circ \text{C}$		10	15	ns
t_{PHL} Propagation delay time, high-to-low-level output	$C_L = 150 \text{ pF}$, $R_L = 2 \text{ k}\Omega$		10	15	ns

Pin Assignments (Top View)

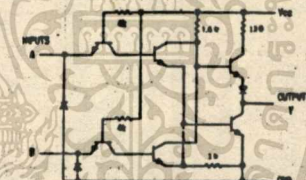
①



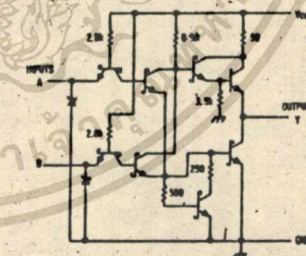
②

positive logic:
 $V = A \cdot B$

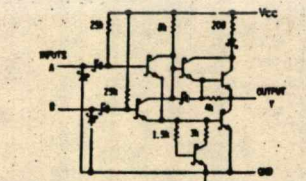
Schematics (each gate)



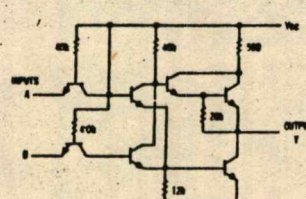
'02' CIRCUIT



'S02' CIRCUIT



'LS02' CIRCUIT



'L02' CIRCUIT

Resistor values shown are nominal and in ohms.

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

‡ All typical values are at $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ \text{C}$.

§ Not more than one output should be shorted at a time, and for SN54S/74S, duration of output short-circuit should not exceed one second.

เอกสารนี้เป็นเอกสารที่... (เอกสารที่...)

5404/7404 Hex Inverter

	Schottky TTL					High-Speed TTL					Low-Power Schottky TTL					Standard TTL					Low-Power TTL				
	Package					Package					Package					Package					Package				
	Device Type	C	P	M	CF	Device Type	C	P	M	CF	Device Type	C	P	M	CF	Device Type	C	P	M	CF	Device Type	C	P	M	CF
T.I.	SN54S04	J	D		W1	SN54H04	J	D		W1	SN54LS04	J	D		W2	SN5404	J	D		W2	SN54L04	J	D		T2
	SN74S04	J	D		W1	SN74H04	J	D		W1	SN74LS04	J	D		W2	SN7404	J	D		W2	SN74L04	J	D		T2
FAIRCHILD	FMS4S04/FMS04	D				FMS4H04/FMSH04	D				FMS4LS04/FMSLS04	D				FMS404/FMSN04	D				FMS4L04/FMSLN04	D			
	FC74S04/FC9S04	D				FC74H04/FC9H04	D				FC74LS04/FC9LS04	D				FC7404/FC9N04	D				FC74L04/FC9LN04	D			
MOTOROLA																									
N.S.C.	DM74S04					DM74H04	J	D			DM74LS04					DM7404	J	D			DM74L04	J	D		F2
PHILIPS	N74S04					N74H04					N74LS04					FH241/7404									
SIGNETICS	S54S04	F	D	A	W1	S54H04	F	D	A	W2						S5404	F	D	A	W2					
	N74S04	F	D	A	W1	N74H04	F	D	A	W2	N74LS04					N7404	F	D	A	W2					
SIEMENS																FLH211									
FUJITSU											74LS04					MB418									
HITACHI	HD74S04										HD74LS04					HD7404/HD2522									
MITSUBISHI	M55004										M74LS04					M53204									
NEC	74S04										74LS04					μPB235									
TOSHIBA																TO3404A									

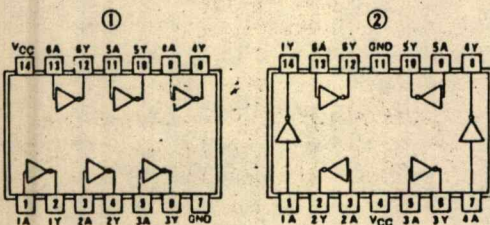
Electrical Characteristics SN54LS04/SN74LS04

absolute maximum ratings over operating free-air temperature range				
Supply voltage, V_{CC}	7V	Operating free-air temperature range	SN54LS	-55°C to 125°C
Input voltage	7V		SN74LS	0°C to 70°C
		Storage temperature range		-55°C to 150°C
recommended operating conditions				
	SN54LS04			UNIT
	MIN	NOM	MAX	
Supply voltage, V_{CC}	4.5	5	5.5	V
High-level output current, I_{OH}			-600	mA
Low-level output current, I_{OL}			4	mA
Operating free-air temperature, T_A	-55	125	8	°C

electrical characteristics over recommended operating free-air temperature range

PARAMETER	TEST CONDITIONS†	MIN	TYP‡	MAX	UNIT
V_{IH} High-level input voltage		2			V
V_{IL} Low-level input voltage				0.8	V
V_I Input clamp voltage	$V_{CC} = \text{MIN}$, $I_I = -18 \text{ mA}$			-1.5	V
V_{OH} High-level output voltage	$V_{CC} = \text{MIN}$, $V_{IL} = V_{IL \text{ max}}$, $I_{OH} = \text{MAX}$	2.7	3.4		V
V_{OL} Low-level output voltage	$V_{CC} = \text{MAX}$, $V_{IH} = 2 \text{ V}$, $I_{OL} = 4 \text{ mA}$			0.4	V
I_I Input current at maximum input voltage	$V_{CC} = \text{MAX}$, $V_I = 7 \text{ V}$		0.1		mA
I_{IH} High-level input current	$V_{CC} = \text{MAX}$, $V_{IH} = 2.7 \text{ V}$		20		μA
I_{IL} Low-level input current	$V_{CC} = \text{MAX}$, $V_{IL} = 0.4 \text{ V}$		-0.6		mA
I_{OS} Short-circuit output current*	$V_{CC} = \text{MAX}$	54LS Family	-20	-100	mA
		74LS Family	-20	-100	mA
I_{CCH} Supply current	$V_{CC} = \text{MAX}$	Total, outputs high	1.2	2.4	mA
I_{CCL} Supply current		Total, outputs low	3.6	6.6	mA
I_{CC} Supply current	$V_{CC} = 5 \text{ V}$	Average per gate (50% duty cycle)	0.4		mA
t_{PLH} Propagation delay time, low-to-high-level output	$V_{CC} = 5 \text{ V}$, $T_A = 25^\circ \text{C}$, $C_L = 15 \text{ pF}$, $R_L = 2 \text{ k}\Omega$		9	15	ns
t_{PHL} Propagation delay time, high-to-low-level output			10	15	ns

Pin Assignments (Top View)



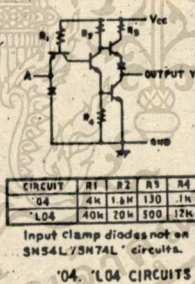
positive logic:

†For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

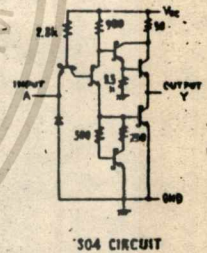
‡All typical values are at $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ \text{C}$.

*Not more than one output should be shorted at a time, and for SN54H/SN74H and SN54S/SN74S, duration of short-circuit should not exceed 1 sec.

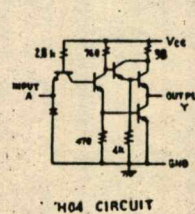
Schematics (each gate)



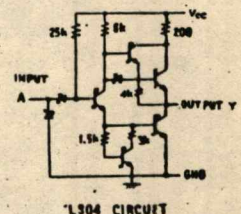
04, L04 CIRCUITS



S04 CIRCUIT



704 CIRCUIT



LS04 CIRCUIT

Resistor values shown are nominal and in ohms.

5417/7417 Hex Buffer/Driver with Open-Collector High-Voltage Output

	Schottky TTL				High-Speed TTL				Low-Power Schottky TTL				Standard TTL				Low-Power TTL			
	Device Type	Package			Device Type	Package			Device Type	Package			Device Type	Package			Device Type	Package		
T.I.													SN5417	J	D		WD			
FAIRCHILD													SN7417	J	D	ND				
MOTOROLA													FM5417/FM7417	D	D		FD			
N.S.C.													FC7417/FC9417	D	D	PD				
PHILIPS													SN7417			P	D			
SIGNETICS													DM5417	J	D	ND		WD		
SIEMENS													DM7417	J	D	ND				
FUJITSU													N7417				D			
HITACHI													SS417	F	D	A	D		WD	
MITSUBISHI													N7417	F	D	A	D			
NEC													FLH4917				D			
TOSHIBA													HD7417			P	D			
													M7417				P	D		
													TD7417				P	D		

Electrical Characteristics SN5417/SN7417

absolute maximum ratings over operating free-air temperature range			
Supply voltage, VCC	7V	Operating free-air temperature range	SN5417: -55°C to 125°C SN7417: 0°C to 70°C
Input voltage	5.5V	Storage temperature range	-65°C to 150°C
Output-state (high-level) voltage applied to open-collector outputs	15V		

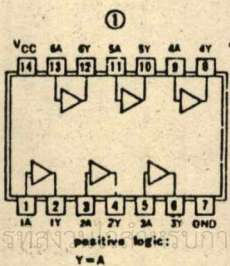
recommended operating conditions

	SN5417			SN7417			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, VCC	4.5	5	5.5	4.75	5	5.25	V
High-level output voltage, VOH			15			15	V
Low-level output current, IOL			30			40	mA
Operating free-air temperature, TA	-55		125	0		70	°C

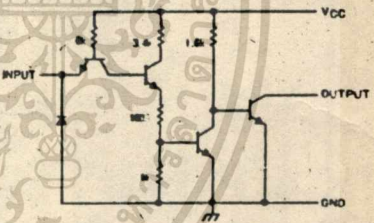
electrical characteristics over recommended operating free-air temperature range

PARAMETER	TEST CONDITIONS 1	MIN	TYP 2	MAX	UNIT
VIH	High-level input voltage		2		V
VIL	Low-level input voltage			0.8	V
VI	Input clamp voltage	VCC=MIN, II=-12mA		-1.5	V
IOH	High-level output current	VCC=MIN, VIH=2V, VOH=MAX		250	μA
VOL	Low-level output voltage	VCC=MIN, VIL=VIL max, IOL=16mA		0.4	V
		VCC=MIN, VIL=VIL max, IOL=MAX		0.7	
II	Input current at maximum input voltage	VCC=MAX, VI=5.5V		1	mA
IIH	High-level input current	VCC=MAX, VIH=2.4V		40	μA
IIL	Low-level input current	VCC=MAX, VIL=0.4V		-1.6	mA
ICCH	Supply current	VCC=MAX	Total, outputs high	29	mA
ICCL	Supply current	VCC=MAX	Total, outputs low	21	mA
ICC	Supply current	VCC=5V	Average per gate (50% duty cycle)	4.17	mA
tdPLH	Propagation delay time, low-to-high-level output	VCC=5V, TA=25°C, CL=15pF, RL=110Ω	6	10	ns
tdPHL	Propagation delay time, high-to-low-level output		20	30	ns

Pin Assignment (Top View)



Schematic (each gate)



Resistor values shown are nominal and in ohms.

1For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions. 2All typical values are at VCC=5V, TA=25°C

5432/7432 Quadruple 2-Input Positive-OR Gate

	Schottky TTL					High-Speed TTL					Low-Power Schottky TTL					Standard TTL					Low-Power TTL						
	Device Type	Package				Device Type	Package				Device Type	Package				Device Type	Package				Device Type	Package					
		C	P	M	CF		C	P	M	CF		C	P	M	CF		C	P	M	CF		C	P	M	CF		
T.I.	SN54S32	J	0		W						SN54LS32	J	0		W												
FAIRCHILD	SN74S32	J	0	N	0						SN74LS32	J	0	N	0												
	FM5432 / FM7432	W	0		F	0					FM54LS32 / FM74LS32	W	0		F	0											
	FC7432 / FC5432	W	0	F	0						FC74LS32 / FC54LS32	W	0	F	0												
MOTOROLA																											
N.S.C.											SN74LS32	P	0														
											DM74LS32	0															
											DM54LS32	0															
PHILIPS																											
	N74S32			0							N74LS32			0													
SIGNETICS																											
SIEMENS																											
FUJITSU																											
HITACHI																											
											HD74LS32	P	0														
MITSUBISHI																											
NEC																											
TOSHIBA																											

Electrical Characteristics SN54LS32/SN74LS32

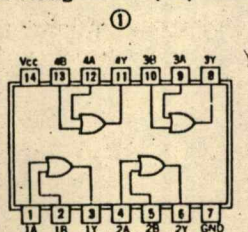
absolute maximum ratings over operating free-air temperature range

Supply voltage, V _{CC}	7V	Operating free-air temperature range	SN54LS	-55°C to 125°C					
Input voltage	7V	Storage temperature range	SN74LS	0°C to 100°C					
				-55°C to 150°C					
recommended operating conditions									
			1000	1000					
			MIN	NOM	MAX	MIN	NOM	MAX	UNIT
Supply voltage, V _{CC}	4.5	5	5.5	4.75	5	5.25			V
High-level output current, I _{OH}			-400			-400			mA
Low-level output current, I _{OL}			4			8			mA
Operating free-air temperature, T _A	-55	125	0			70			°C

electrical characteristics over recommended operating free-air temperature range

PARAMETER	TEST CONDITIONS †	MIN	TYP ‡	MAX	UNIT
V _{IH} High-level input voltage		2			V
V _{IL} Low-level input voltage			0.8		V
V _I Input clamp voltage	V _{CC} = MIN, I _I = -18 mA		-1.5		V
V _{OH} High-level output voltage	V _{CC} = MIN, V _{IH} = 2V, I _{OH} = MAX	2.7	3.4		V
V _{OL} Low-level output voltage	V _{CC} = MIN, V _{IL} = V _{IL} max, I _{OL} = 4 mA	0.25	0.4		V
I _I Input current at maximum input voltage	V _{CC} = MAX, V _I = 7V		0.1		mA
I _{IH} High-level input current	V _{CC} = MAX, V _{IH} = 2.7V		20		µA
I _{IL} Low-level input current	V _{CC} = MAX, V _{IL} = 0.4V		-0.4		mA
I _{OS} Short-circuit output current †	V _{CC} = MAX	54LS Family	-20	-100	mA
		74LS Family	-20	-100	mA
I _{CC} Supply current	V _{CC} = MAX	Total, outputs high	3.1	6.2	mA
I _{CC} Supply current	V _{CC} = MAX	Total, outputs low	4.9	9.8	mA
I _{CC} Supply current	V _{CC} = 5V	Average per gate (50% duty cycle)	1.0		mA
t _{PLH} Propagation delay time, low-to-high-level output	V _{CC} = 5V, T _A = 25°C, C _L = 15 pF, R _L = 2 kΩ		14	22	ns
t _{PHL} Propagation delay time, high-to-low-level output			14	22	ns

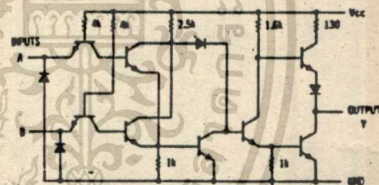
Pin Assignment (Top View)



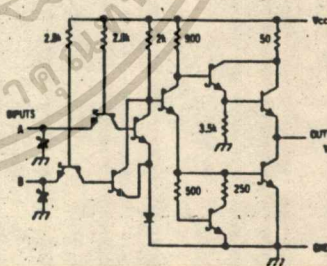
positive logic:

Y = A + B

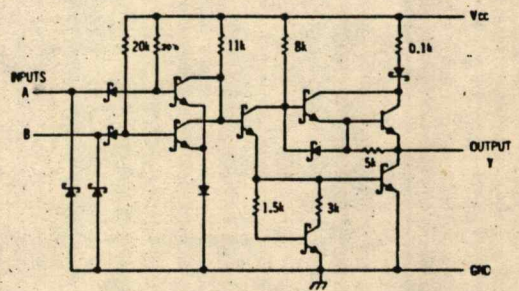
Schematics (each gate)



22 CIRCUIT



33 CIRCUIT



LS32 CIRCUIT

Resistor values shown are nominal and in ohms

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

‡ All typical values are at V_{CC} = 5V, T_A = 25°C.

• Not more than one output should be shorted at a time.

	Schottky TTL					High-Speed TTL					Low-Power Schottky TTL					Standard TTL					Low-Power TTL				
	Device Type	Package				Device Type	Package				Device Type	Package				Device Type	Package				Device Type	Package			
		C	P	M	CF		C	P	M	CF		C	P	M	CF		C	P	M	CF		C	P	M	CF
T.I.					SN54H73	J	Ⓛ		W	SN54LS73	J	Ⓛ		W	SN5473	J	Ⓛ		W	SN54L73	J	Ⓛ		W	
					SN74H73	J	Ⓛ		ND	SN74LS73	J	Ⓛ		ND	SN7473	J	Ⓛ		ND	SN74L73	J	Ⓛ		ND	
FAIRCHILD					FMS4H73/FM9H73	D	Ⓛ		F	PM54LS73/PM5LS73	D	Ⓛ		F	FMS473/FM9H73	D	Ⓛ		F						
					FC74H73/FC9H73	D	Ⓛ		P	FO4LS73/FO4LS73	D	Ⓛ		P	FC7473/FC9H73	D	Ⓛ		P						
MOTOROLA					MC3163	L	Ⓛ		F						MC5473	L	Ⓛ		F						
					MC3063	L	Ⓛ		P	SN74LS73				P	MC7473	L	Ⓛ		P						
N.S.C.					DM54H73	J	Ⓛ		ND	DM54LS73				P	DM5473	J	Ⓛ		ND	DM54L73	J	Ⓛ		ND	
					DM74H73	J	Ⓛ		ND	DM74LS73				P	DM7473	J	Ⓛ		ND	DM74L73	J	Ⓛ		ND	
PHILIPS							Ⓛ			N74LS73			Ⓛ		FJ1121/7473			Ⓛ							
SIGNETICS					S54H73	F	Ⓛ		AD				W	S5473	F	Ⓛ		AD		W					
					N74H73	F	Ⓛ		AD	N74LS73			A		N7473	F	Ⓛ		AD						
SIEMENS															FL1121			Ⓛ							
FUJITSU																									
HITACHI										HD74LS73			P		HD7473/HD2515			Ⓛ	P						
MITSUBISHI										M74LS73			P		M53273			P							
NEC															μPB225			D	Ⓛ						
TOSHIBA															TD3473A			P							

operating temperature ranges over operating free-air temperature range						
Supply voltage, V _{CC}	TV	Operating free-air temperature range	SN54LS	-55°C to 125°C		
Input voltage	5.0V	temperature range	SN74LS	0°C to 70°C		
		Storage temperature range		-55°C to 150°C		

recommended operating conditions

	SN54LS73A			SN74LS73A			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V _{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I _{OH}	-80			-400			mA
Low-level output current, I _{OL}	4			8			mA
Pulse width, t _p	Slow edge	20	20				ns
	Fast or slow edge	20	20				
Input setup time, t _{su}	High-level data	20	25				ns
	Low-level data	20	20				
Input hold time, t _{hd}	0	0	5				ns
Operating free-air temperature T _a	-55	125	150	0	70	125	°C

PARAMETER *		TEST CONDITIONS †	MIN	TYP	MAX	UNIT
V _{IH}	High-level input voltage			2		V
V _{IL}	Low-level input voltage				0.8	V
V _I	Input clamp voltage	V _{CC} = MIN., I _I = -18 mA			-1.5	V
V _{OH}	High-level output voltage	V _{CC} = MIN., V _{IH} = 2V, V _{IL} = 0.8V, I _{OH} = MAX	2.7	3.4		V
V _{OL}	Low-level output voltage	V _{CC} = MIN., V _{IH} = 2V, V _{IL} = 0.8V, I _{OL} = 4 mA		0.25	0.4	V
I _I	Input current at maximum input voltage	D, J, K, Clear			0.1	mA
		V _{CC} = MAX., V _I = 7V			0.3	
		Preset			0.3	
		Clock			0.4	
I _{IH}	High-level input current	D, J, K, Clear			20	μA
		V _{CC} = MAX., V _I = 2.7V			50	
		Preset			60	
		Clock			80	
I _{IL}	Low-level input current	D, J, K, Clear			-0.4	mA
		V _{CC} = MAX., V _I = 0.4 V			-0.8	
		Preset			-2.8	
		Clock			-0.8	
I _{OS}	Short-circuit output current *	Series 54LS		-20	-100	mA
		Series 74LS		-20	-100	
I _{CC}	Supply current (Average per flip-flop)	V _{CC} = MAX. See Note 1		4	5	mA
f _{max}	Maximum clock frequency	V _{CC} = 5V, T _A = 25°C, C _L = 15pF, R _L = 2kΩ	30	45		MHz
1PLH	from clear, preset or clock (as appropriate) to 0 or 1			15	20	ns
1PHL	from clear, preset or clock (as appropriate) to 0 or 1			15	20	

'73 H.73. L.73 (See Note 2) L573 (See Note 2)

INPUTS				OUTPUTS	
CLEAR	CLOCK	J	K	Q	$\bar{Q}$
L	X	X	X	L	H
H	$\downarrow$	L	L	Q_0	$\bar{Q}_0$
H	$\downarrow$	L	H	L	H
H	$\downarrow$	L	H	L	H
H	$\downarrow$	H	H	TOGGLE	

NOTES: 1. with all outputs open, I_{CC} is measured with the 0 and 0 outputs high in turn.
At the same time of measurement, the clock input is grounded.
2. M=high level (steady state), L=low level (steady state), X=irrelevant
1=transition from high to low level
1=high-level pulse; data inputs should be held constant while clock is high;
data is transferred to output on the falling edge of the clock pulse.
Over the level of the clock, the associated input conditions were established.
TOGGLE: Each output changes to the complement of its previous level on
each active transition (pulse) of the clock.

†For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions.

†All typical values are at $V_{CC}=5V$, $T_A=25^\circ C$.

• Not more than one output should be shorted at time.

* t_{PLH} = propagation delay time, low-to-high-level output :

t_{PHL} = propagation delay time, high-to-low-level output

§. If the arrow indicates the edge of the clock pulse used for reference: ↑ for the rising edge, ↓ for the falling edge.

5486/7486 Quadrate 2-Input Exclusive-OR Gate

	Schottky TTL					High-Speed TTL					Low-Power Schottky TTL					Standard TTL					Low-Power TTL							
	Device Type	Package				Device Type	Package				Device Type	Package				Device Type	Package				Device Type	Package						
		C	P	M	CF		C	P	M	CF		C	P	M	CF		C	P	M	CF		C	P	M	CF			
T.I.	SN54S86	J	1		WT							SN54LS86	J	1		WT	SN5486	J	1		WT	SN54LS86	J	2	N	2	TD	
	SN74S86	J	1	N	1							SN74LS86	J	1	N	1		SN7486	J	1	N	1	SN74LS86	J	2	N	2	TD
FAIRCHILD	FMS486L/FMS486	ND			TD							FMS486L/FMS486	ND			TD	FMS486/FMS486	ND			TD							
	FC7486L/FC7486	ND			TD							FC7486L/FC7486	ND			TD	FC7486/FC7486	ND			TD							
MOTOROLA																												
												SN74LS86	P	2			MC7486				P	TD						
N-S-C	DM74S86				WT							DM54LS86	P	2			DM5486	J	1	N	1	WT	DM54LS86	J	2	N	2	F
												DM74LS86	P	2			DM7486	J	1	N	1		DM74LS86	J	2	N	2	F
PHILIPS	N74S86											N74LS86					FH27V7486											
	154S86																5486	F	1	N	1	WT						
SIGNETICS	87S86											N74LS86	A	2			N7486	F	1	N	1	WT						
SIEMENS																	FLH341											
FUJITSU												74LS86	M	2			MB449											
HITACHI	HD74S86				TD							HD74LS86	P	2			HD7486/HD2526				TD							
MISUBISHI												N74LS86	P	2			M53286				P	TD						
NEC												74LS86	C	2			μPB7086				TD							
TOSHIBA																	TD7486				P	TD						

Electrical Characteristics SN54LS86/SN74LS86

absolute maximum ratings over operating free-air temperature range

Supply voltage, V_{CC}	7V	Operating free-air temperature range	SN54LS	-55°C to 125°C
Input voltage	7V	temperature range	SN74LS	0°C to 70°C
		Storage temperature range		-65°C to 150°C

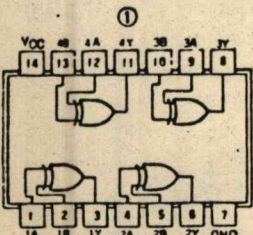
recommended operating conditions

	SN54LS86			SN74LS86			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			-800			-800	μA
Low-level output current, I_{OL}			4			8	mA
Operating free-air temperature, T_A	-55		125	0		70	°C

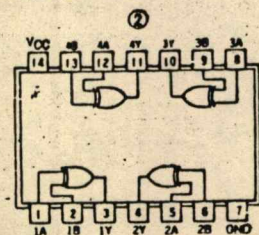
electrical characteristics over recommended operating free-air temperature range

PARAMETER	TEST CONDITIONS	SN74LS86	UNIT
V_{IH} High-level input voltage		2	V
V_{IL} Low-level input voltage		0.8	V
V_T Input clamp voltage	$V_{CC} = \text{MIN.}$ $I_I = -18\text{mA}$	-1.5	V
V_{OH} High-level output voltage	$V_{CC} = \text{MIN.}$ $V_{IH} = 2V$ $V_{IL} = V_{IL} \text{ max.}$ $I_{OH} = -800\mu A$	2.7 3.4	V
V_{OL} Low-level output voltage	$V_{CC} = \text{MIN.}$ $V_{IH} = 2V$ $V_{IL} = V_{IL} \text{ max.}$ $I_{OL} = 8\text{mA}$	0.25 0.4	V
I_I Input current at maximum input voltage	$V_{CC} = \text{MAX.}$ $V_I = 2V$ $V_{IL} = V_{IL} \text{ max.}$ $I_{OL} = 8\text{mA}$	0.35 0.5	mA
I_{IH} High-level input current	$V_{CC} = \text{MAX.}$ $V_I = 2V$	0.2	mA
I_{IL} Low-level input current	$V_{CC} = \text{MAX.}$ $V_I = 0.4V$	-0.8	mA
I_{OS} Short-circuit output current	$V_{CC} = \text{MAX.}$	-5 -42	mA
I_{CC} Supply current	$V_{CC} = \text{MAX.}$ See Note 2	6.1 10	mA
t_{PLH} from A or B input	Other input low. $V_{CC} = 5V$ $T_A = 25^\circ C$ $C_L = 15\text{pF}$ $R_L = 2k\Omega$	12 23	ns
t_{PLH} from A or B input	Other input high	10 17	ns
t_{PLH} from A or B input		20 30	ns
t_{PLH} from A or B input		13 22	ns

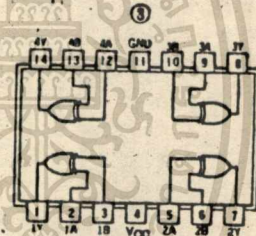
Pin Assignments (Top View)



positive logic: $Y = A \oplus B = \bar{A}B + A\bar{B}$



positive logic: $Y = A \oplus B = \bar{A}B + A\bar{B}$



positive logic: $Y = A \oplus B = \bar{A}B + A\bar{B}$

Function Table

INPUTS	OUTPUT
A B	Y
L L	L
L H	H
H L	H
H H	L

H = High-level L = Low-level

NOTE: t_{CC} is measured with the inputs grounded and the outputs open.

† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable type.
 ‡ All typical values are at $V_{CC} = 5V$, $T_A = 25^\circ C$.
 § Not more than one output should be shorted at a time.
 ¶ t_{PLH} = Propagation delay time, low-to-high-level output
 †† t_{PLH} = Propagation delay time, high-to-low-level output

54123/74123 Dual Retriggerable Monostable Multivibrator with Clear

	Schottky TTL					High-Speed TTL					Low-Power Schottky TTL					Standard TTL					Low-Power TTL				
	Device Type	Package				Device Type	Package				Device Type	Package				Device Type	Package				Device Type	Package			
		C	P	M	CF		C	P	M	CF		C	P	M	CF		C	P	M	CF		C	P	M	CF
T.I.																									
FAIRCHILD																									
MOTOROLA																									
N.S.C.																									
PHILIPS																									
SIGNETICS																									
SIEMENS																									
FUJITSU																									
HITACHI																									
mitsubishi																									
NEC																									
TOSHIBA																									

Electrical Characteristics SN54LS123/SN74LS123

absolute maximum ratings operating free-air temperature range

Supply voltage, V_{CC}	7V	Operating free-air temperature range	SN54LS123	-55°C to 125°C
Input voltage	7V	Storage temperature range	SN74LS123	-55°C to 150°C

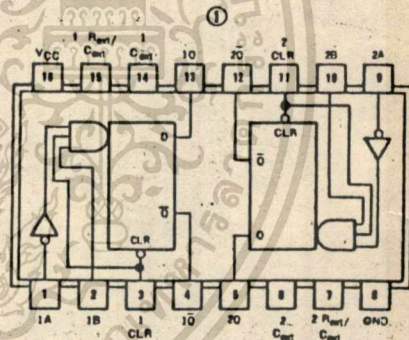
recommended operating conditions

	SN54LS123			SN74LS123			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			-800			-600	μ A
Low-level output current, I_{OL}			4			8	mA
Pulse width, t_p	45			45			ns
External timing resistance, R_{ext}	5		180	5		250	Ω
External timing capacitance, C_{ext}			10			50	pF
Wiring capacitance at first Clear terminal			50			50	pF
Operating free-air temperature, T_A	-55		125	-55		150	°C

electrical characteristics over recommended operating free-air temperature range

PARAMETER	TEST CONDITIONS	MIN	TYP	MAX	UNIT
V_{IH} High-level input voltage		2			V
V_{IL} Low-level input voltage			0.8		V
V_I Input clamp voltage	$V_{CC} = \text{MIN.}$, $I_I = -18 \text{ mA}$			-1.5	V
V_{OH} High-level output voltage	$V_{CC} = \text{MIN.}$, $I_{OH} = \text{MAX.}$, See Note 1	2.7	3.5		V
V_{OL} Low-level output voltage	$V_{CC} = \text{MIN.}$, $I_{OL} = 8 \text{ mA}$, See Note 1	0.35	0.5		V
I_I Input current at maximum input voltage	$V_{CC} = \text{MAX.}$, $V_I = 7 \text{ V}$		0.1		mA
I_{IH} High-level input current	$V_{CC} = \text{MAX.}$, $V_I = 2.7 \text{ V}$		20		μ A
I_{IL} Low-level input current	$V_{CC} = \text{MAX.}$, $V_I = 0.4 \text{ V}$		0.4		mA
I_{OS} Short-circuit output current	$V_{CC} = \text{MAX.}$, See Note 1	20		150	mA
I_{CC} Supply current (quiescent or triggered)	$V_{CC} = \text{MAX.}$, See Notes 2 and 3		12	20	mA
t_{PLH} from A to output Q	$V_{CC} = 5 \text{ V}$, $T_A = 25^\circ \text{C}$, $C_L = 150 \text{ pF}$, $R_L = 2 \text{ k}\Omega$	$C_{ext} = 0$, $R_{ext} = 5 \text{ k}\Omega$	22	33	ns
t_{PHL} from B to output Q			23	44	ns
t_{PLH} from A to output Q			32	45	ns
t_{PHL} from B to output Q			34	56	ns
t_{PHL} from Clear to output Q			25	27	ns
t_{PLH} from Clear to output Q			28	45	ns
$t_{WQ}(\text{min})$ from A or B to output Q	$C_{ext} = 1000 \text{ pF}$, $R_{ext} = 10 \text{ k}\Omega$	4	4.5	5	μ s
t_{WQ} from A or B to output Q					

Pin Assignment (Top View)



FUNCTION TABLE

123 LS123, L123 (See Note 4)

INPUTS		OUTPUTS	
CLEAR	A	B	Q
L	X	X	L
X	H	X	L
X	X	L	H
X	X	H	L
H	L	↑	L
L	H	↑	L
↑	L	H	↑

- NOTES: 1. Ground C_{ext} to measure V_{OH} at 0, V_{OL} at 0, or I_{OS} at 0.
 C_{ext} is open to measure V_{OH} at 0, V_{OL} at 0, or I_{OS} at 0.
 2. Quiescent I_{CC} is measured (after clearing) with 2.4V applied to all clear and A inputs. B inputs grounded, all outputs open, $C_{ext} = 0.02 \mu\text{F}$, and $R_{ext} = 25 \text{ k}\Omega$.
 3. I_{CC} is measured in the triggered state with 2.4V applied to all clear and B inputs. A inputs grounded, all outputs open, $C_{ext} = 0.02 \mu\text{F}$, and $R_{ext} = 25 \text{ k}\Omega$.
 4. A: H=high level (steady state), L=low level (steady state), ↑=transition from low to high level, ↓=transition from high to low level, J=one high-level pulse, U=one low-level pulse, X=relevant (any input, including transitions).
 B: An external timing capacitor may be connected between and R_{ext}/C_{ext} (positive).
 C: For accurate repeatable pulse widths, connect an external resistor between R_{ext}/C_{ext} and V_{CC} with R_{ext} open-circuited.
 D: To obtain variable pulse widths, connect external resistance between R_{ext} or R_{ext}/C_{ext} and V_{CC} .

† For conditions shown as MIN or MAX, use the value specified under recommended operating conditions.

‡ All typical values are at $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ \text{C}$.

§ Not more than one output should be shorted at a time.

• t_{PLH} = propagation delay time, low-to-high-level output

† t_{PHL} = propagation delay time, high-to-low-level output

54138/74138 3-Line-to-8-Line Decoder

	Schottky TTL				High-Speed TTL				Low-Power Schottky TTL				Standard TTL				Low-Power TTL			
	Device Type	Package			Device Type	Package			Device Type	Package			Device Type	Package			Device Type	Package		
		C	P	M/CF		C	P	M/CF		C	P	M/CF		C	P	M/CF		C	P	M/CF
T.I.	SN54S138	J (L)		W (L)					SN54LS138	J (L)		W (L)								
FAIRCHILD	SN74S138	J (L)	N (D)						SN74LS138	J (L)	N (D)									
	/ 74ALS138	D (D)							74ALS138/ 74M138	D (D)		D (D)								
	PC74S138/ PC75S138	D (D)							PC74LS138/ PC75LS138	D (D)	D (D)	D (D)								
MOTOROLA									SN74LS138		P (D)									
N.S.C.									DM74LS138		(L)									
	DM74S138		(L)						DM54LS138		(L)									
PHILIPS	N74S138		(L)						N74LS138		(L)									
SIGNETICS	SS4S138	F (L)	B (D)	W (L)																
	N74S138	F (L)	B (D)						N74LS138		A (L)									
SIEMENS																				
FUJITSU									74LS138		M (D)									
HTACHI									HD74LS138		P (D)									
MTSUBISHI	M74S138		P (D)						M74LS138		P (D)									
NEC									74LS138		C (D)									
TOSHIBA																				

Electrical Characteristics - SN54LS138/SN74LS138

absolute maximum ratings over operating
free-air temperature range

Supply voltage, V_{CC}	TV	Operating free-air temperature range	SN54LS	-55°C to 125°C
Input voltage	TV	Storage temperature range	SN74LS	0°C to 70°C
				-65°C to 150°C

recommended operating conditions

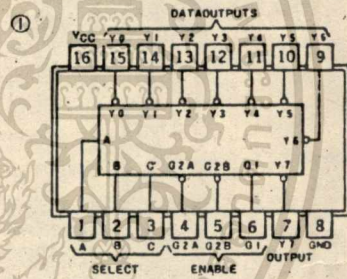
	SN54LS138			SN74LS138			UNIT
	MIN	NOM	MAX	MIN	NOM	MAX	
Supply voltage, V_{CC}	4.5	5	5.5	4.75	5	5.25	V
High-level output current, I_{OH}			800			400	mA
Low-level output current, I_{OL}			4			8	mA
Operating free-air temperature, T_A	-55	125	9	70			°C

electrical characteristics over recommended operating
free-air temperature range

PARAMETER #	TEST CONDITIONS†	MIN	TYP‡	MAX	UNIT
V_{IH}	High-level input voltage		2		V
V_{IL}	Low-level input voltage			0.8	V
V_I	Input clamp voltage	$V_{CC} = \text{MIN.}$, $I_I = -18 \text{ mA}$		1.5	V
V_{OH}	High-level output voltage	$V_{CC} = \text{MIN.}$, $V_{IH} = 2 \text{ V}$, $V_{IL} = 0.8 \text{ V}$, $I_{OH} = 400 \text{ } \mu\text{A}$	2.5	3.4	V
V_{OL}	Low-level output voltage	$V_{CC} = \text{MIN.}$, $V_{IH} = 2 \text{ V}$, $V_{IL} = 0.8 \text{ V}$, $I_{OL} = 8 \text{ mA}$	0.35	0.5	V
I_I	Input current at maximum input voltage	$V_{CC} = \text{MAX.}$, $V_I = \text{TV}$		0.1	mA
I_{IH}	High-level input current	$V_{CC} = \text{MAX.}$, $V_I = 2.7 \text{ V}$		20	μA
I_{IL}	Low-level input current	$V_{CC} = \text{MAX.}$, $V_I = 0.4 \text{ V}$		0.4	mA
I_{OS}	Short-circuit output current	$V_{CC} = \text{MAX.}$	-20	-100	mA
I_{CC}	Supply current	$V_{CC} = \text{MAX.}$, Outputs enabled and open	6.3	10	mA

t_{PLH}	from Binary select		2	13	20	ns
t_{PHL}	to Any output		3	18	27	ns
t_{PLH}	from Enable		2	21	32	ns
t_{PHL}	to Any output		3	17	26	ns

Pin Assignment (Top View)

positive logic:
see function table

Function Table

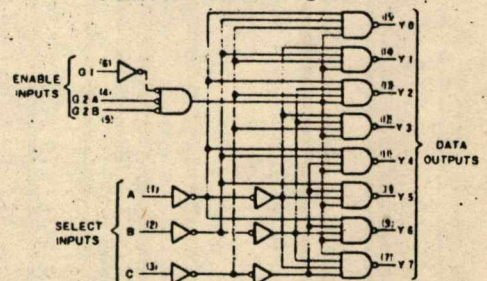
*S138*LS138

ENABLE		INPUTS			OUTPUTS										
G1	G2*	C	B	A	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7			
X	H	X	X	X	H	H	H	H	H	H	H	H			
L	X	X	X	X	H	H	H	H	H	H	H	H			
H	L	L	L	L	L	H	H	H	H	H	H	H			
H	L	L	L	H	H	L	H	H	H	H	H	H			
H	L	L	H	L	H	H	L	H	H	H	H	H			
H	L	L	H	H	H	H	L	H	H	H	H	H			
H	L	H	L	L	H	H	H	L	H	H	H	H			
H	L	H	L	H	H	H	H	L	H	H	H	H			
H	L	H	H	L	H	H	H	H	L	H	H	H			
H	L	H	H	H	H	H	H	H	H	L	H	H			
H	L	H	H	H	H	H	H	H	H	H	L	H			

* G2 = G2A + G2B

H = high level, L = low level, X = irrelevant

Functional Block Diagram



*S138*LS138 DECODER/DEMULPLEXER

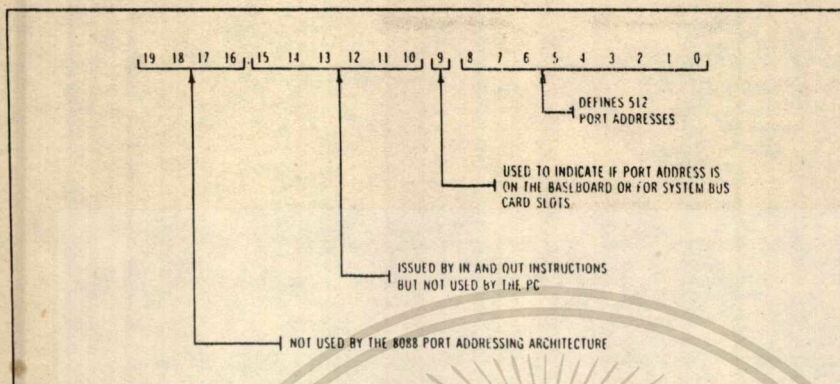
† For conditions shown as MIN or MAX, use the appropriate value specified under recommended operating conditions for the applicable device type.

‡ All typical values are at $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ \text{C}$.

* Not more than one output should be shorted at a time, and duration of the short-circuit test should not exceed one second.

† t_{PLH} = propagation delay time, low-to-high-level output† t_{PHL} = propagation delay time, high-to-low-level output

มีให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



I/O port addresses of the 8088 MPU.

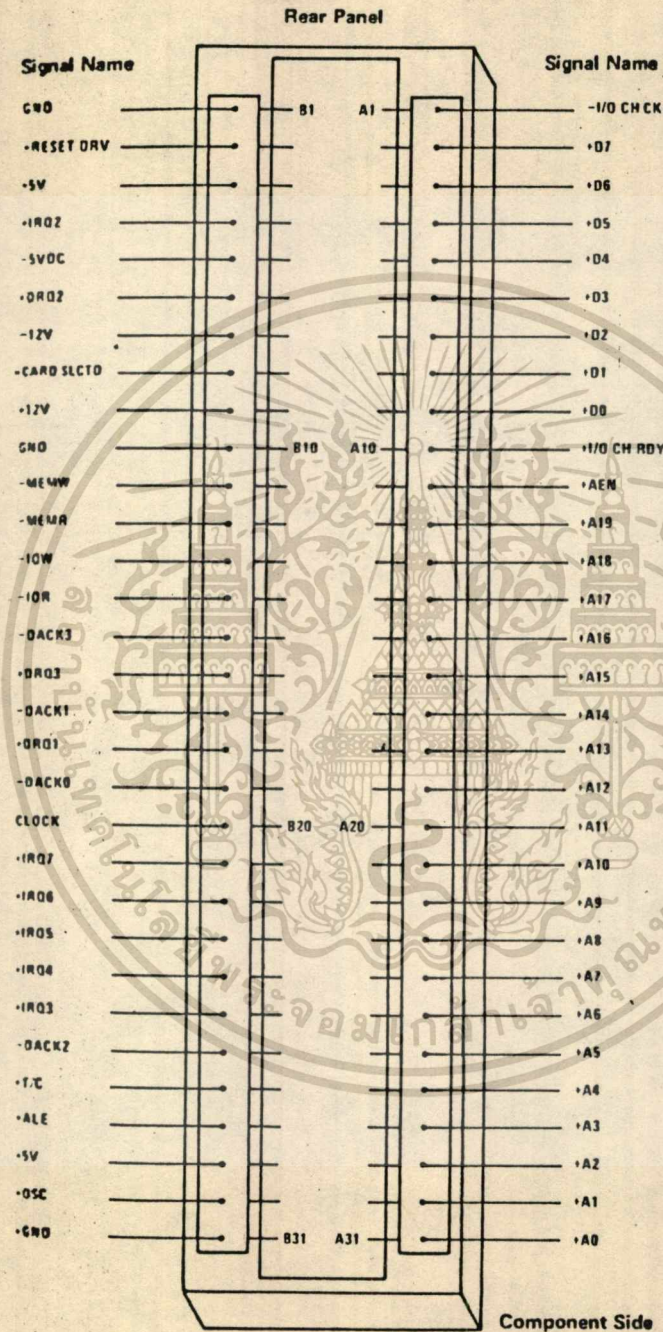
System Memory, I/O Map, and Decoding Techniques

HEX ADDRESS	USES
0200H	NOT USED
0201H	GAME CONTROL ADAPTER
0202H	NOT USED
0277H	NOT USED
0278H	SECOND PRINTER PORT ADAPTER
0279H	NOT USED
0280H	NOT USED
0281H	NOT USED
0282H	NOT USED
0283H	NOT USED
0284H	NOT USED
0285H	NOT USED
0286H	NOT USED
0287H	NOT USED
0288H	NOT USED
0289H	NOT USED
028AH	NOT USED
028BH	NOT USED
028CH	NOT USED
028DH	NOT USED
028EH	NOT USED
028FH	NOT USED
0290H	NOT USED
0291H	NOT USED
0292H	NOT USED
0293H	NOT USED
0294H	NOT USED
0295H	NOT USED
0296H	NOT USED
0297H	NOT USED
0298H	NOT USED
0299H	NOT USED
029AH	NOT USED
029BH	NOT USED
029CH	NOT USED
029DH	NOT USED
029EH	NOT USED
029FH	NOT USED
02A0H	NOT USED
02A1H	NOT USED
02A2H	NOT USED
02A3H	NOT USED
02A4H	NOT USED
02A5H	NOT USED
02A6H	NOT USED
02A7H	NOT USED
02A8H	NOT USED
02A9H	NOT USED
02AAH	NOT USED
02ABH	NOT USED
02ACH	NOT USED
02ADH	NOT USED
02AEH	NOT USED
02AFH	NOT USED
02B0H	NOT USED
02B1H	NOT USED
02B2H	NOT USED
02B3H	NOT USED
02B4H	NOT USED
02B5H	NOT USED
02B6H	NOT USED
02B7H	NOT USED
02B8H	NOT USED
02B9H	NOT USED
02BAH	NOT USED
02BBH	NOT USED
02BCH	NOT USED
02BDH	NOT USED
02BEH	NOT USED
02BFH	NOT USED
02C0H	NOT USED
02C1H	NOT USED
02C2H	NOT USED
02C3H	NOT USED
02C4H	NOT USED
02C5H	NOT USED
02C6H	NOT USED
02C7H	NOT USED
02C8H	NOT USED
02C9H	NOT USED
02CAH	NOT USED
02CBH	NOT USED
02CH	NOT USED
02D0H	NOT USED
02D1H	NOT USED
02D2H	NOT USED
02D3H	NOT USED
02D4H	NOT USED
02D5H	NOT USED
02D6H	NOT USED
02D7H	NOT USED
02D8H	NOT USED
02D9H	NOT USED
02DAH	NOT USED
02DBH	NOT USED
02DCH	NOT USED
02DDH	NOT USED
02DEH	NOT USED
02DFH	NOT USED
02E0H	NOT USED
02E1H	NOT USED
02E2H	NOT USED
02E3H	NOT USED
02E4H	NOT USED
02E5H	NOT USED
02E6H	NOT USED
02E7H	NOT USED
02E8H	NOT USED
02E9H	NOT USED
02EAH	NOT USED
02EBH	NOT USED
02ECH	NOT USED
02EDH	NOT USED
02EEH	NOT USED
02EFH	NOT USED
02F0H	NOT USED
02F1H	NOT USED
02F2H	NOT USED
02F3H	NOT USED
02F4H	NOT USED
02F5H	NOT USED
02F6H	NOT USED
02F7H	NOT USED
02F8H	NOT USED
02F9H	NOT USED
02FAH	NOT USED
02FBH	NOT USED
02FCH	NOT USED
02FDH	NOT USED
02FEH	NOT USED
02FFH	NOT USED

NOTE: NEW FEATURES BY IBM AND OTHER MANUFACTURERS MAY USE SOME OF THE SPARE I/O ADDRESS DECODES.

Card slot I/O port address usage.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



I/O Channel Diagram

8255A/8255A-5 PROGRAMMABLE PERIPHERAL INTERFACE

- MCS-85™ Compatible 8255A-5
- 24 Programmable I/O Pins
- Completely TTL Compatible
- Fully Compatible with Intel® Microprocessor Families
- Improved Timing Characteristics
- Direct Bit Set/Reset Capability Easing Control Application Interface
- Reduces System Package Count
- Improved DC Driving Capability
- Available in EXPRESS
 - Standard Temperature Range
 - Extended Temperature Range

The Intel® 8255A is a general purpose programmable I/O device designed for use with Intel® microprocessors. It has 24 I/O pins which may be individually programmed in 2 groups of 12 and used in 3 major modes of operation. In the first mode (MODE 0), each group of 12 I/O pins may be programmed in sets of 4 to be input or output. In MODE 1, the second mode, each group may be programmed to have 8 lines of input or output. Of the remaining 4 pins, 3 are used for handshaking and interrupt control signals. The third mode of operation (MODE 2) is a bidirectional bus mode which uses 8 lines for a bidirectional bus, and 5 lines, borrowing one from the other group, for handshaking.

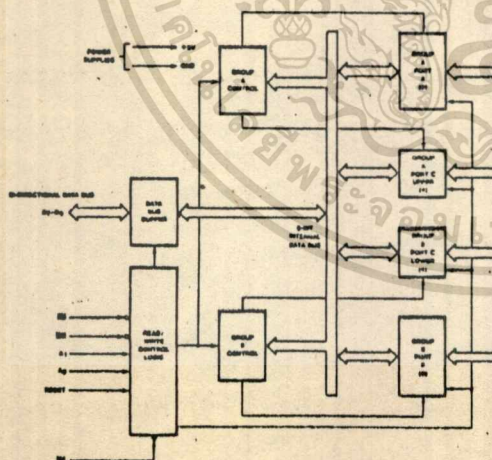


Figure 1. 8255A Block Diagram

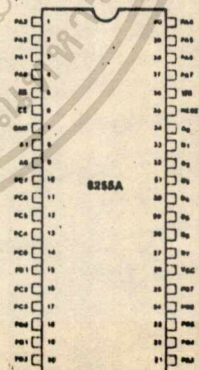


Figure 2. Pin Configuration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

8255A FUNCTIONAL DESCRIPTION

General

The 8255A is a programmable peripheral interface (PPI) device designed for use in Intel® microcomputer systems. Its function is that of a general purpose I/O component to interface peripheral equipment to the microcomputer system bus. The functional configuration of the 8255A is programmed by the system software so that normally no external logic is necessary to interface peripheral devices or structures.

Data Bus Buffer

This 3-state bidirectional 8-bit buffer is used to interface the 8255A to the system data bus. Data is transmitted or received by the buffer upon execution of input or output instructions by the CPU. Control words and status information are also transferred through the data bus buffer.

Read/Write and Control Logic

The function of this block is to manage all of the internal and external transfers of both Data and Control or Status words. It accepts inputs from the CPU Address and Control busses and in turn, issues commands to both of the Control Groups.

$$(\overline{CS})$$

Chip Select. A "low" on this input pin enables the communication between the 8255A and the CPU.

(RD)

Read. A "low" on this input pin enables the 8255A to send the data or status information to the CPU on the data bus. In essence, it allows the CPU to "read from" the 8255A.

(WR)

Write. A "low" on this input pin enables the CPU to write data or control words into the 8255A.

(A₀ and A₁)

Port Select 0 and Port Select 1. These input signals, in conjunction with the RD and WR inputs, control the selection of one of the three ports or the control word registers. They are normally connected to the least significant bits of the address bus (A_0 and A_1).

8255A BASIC OPERATION

A ₁	A ₀	RD	WR	CS	INPUT OPERATION (READ)
0	0	0	1	0	PORT A ← DATA BUS
0	1	0	1	0	PORT B ← DATA BUS
1	0	0	1	0	PORT C ← DATA BUS
					OUTPUT OPERATION (WRITE)
0	0	1	0	0	DATA BUS → PORT A
0	1	1	0	0	DATA BUS → PORT B
1	0	1	0	0	DATA BUS → PORT C
1	1	1	0	0	DATA BUS → CONTROL
					DISABLE FUNCTION
X	X	X	X	1	DATA BUS ← 3-STATE
1	1	0	1	0	ILLEGAL CONDITION
X	X	1	1	0	DATA BUS ← 3-STATE

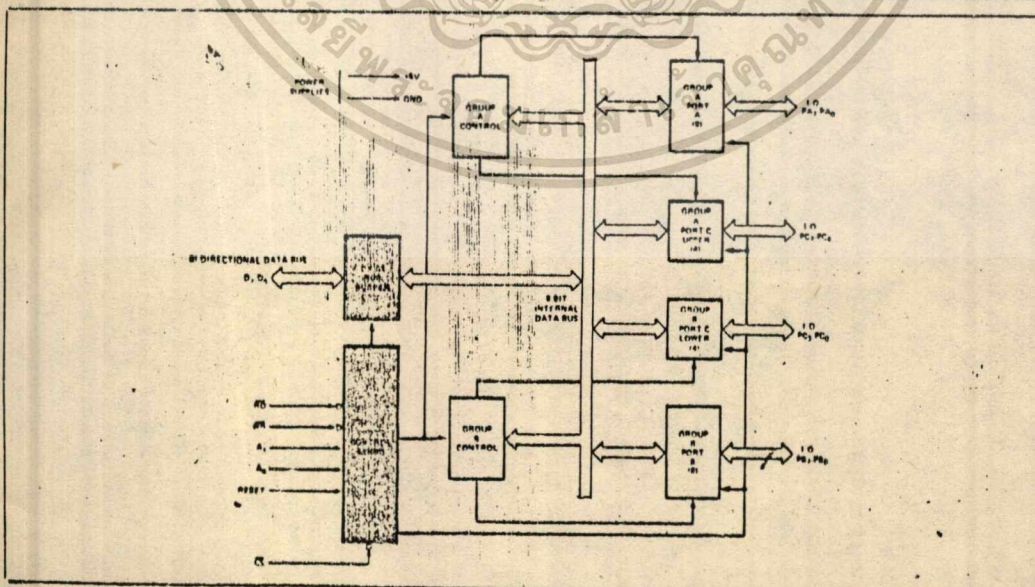


Figure 3. 8255A Block Diagram Showing Data Bus Buffer and Read/Write Control Logic Functions