สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

การสร้างชุดขับเคลื่อนแท่นสแกน xy ควบคุมด้วยคอมพิวเตอร์



นางสาวกนกพร โพธิ์เทียนทอง
นางสาวชุติมา เรืองไทย
นางสาวศันสนีย์ ทวีธนประสิทธิ์

โครงงานพิเศษเป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต
ภาควิชา ฟิสิกส์ประยุกต์
คณะวิทยาศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2546

เลขหมู่.....
เลขทะเบียน 51880
วัน,เดือน,ปี 4 ส.ค. 2547

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

b.....
i.....

Construction of computerized xy Scanning System

Miss Kanokporn Potiantong

Miss Chutima Ruangthai

Miss Sansanee Taweetanaprasit

A Special Project Submitted in Partial Fulfillment of the Requirement for the Degree of

Bachelor of Science

Department of Applied Physics

Faculty of Science

King Mongkut's Institute of Technology Ladkrabang

Academic Year 2003

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงงานพิเศษเรื่อง การสร้างชุดขับเคลื่อนแท่นสแกน xy ควบคุมด้วยคอมพิวเตอร์

นักศึกษา นางสาวกนกพร โพธิ์เทียนทอง

นางสาวชุติมา เรืองไทย

นางสาวศันสนีย์ ทวีธนประสิทธิ์

ภาควิชา ฟิสิกส์ประยุกต์

สาขาวิชา โซลิดสเตทอิเล็กทรอนิกส์

อาจารย์ที่ปรึกษา ผศ.ดร.วราวุฒิ เถาลัดดา

ภาควิชา ฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง อนุมัติให้โครงงานพิเศษนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิทยาศาสตรบัณฑิต

คณะกรรมการตรวจสอบ	ลายมือชื่อ
ประธานกรรมการ ผศ.ดร.วราวุฒิ เถาลัดดา	
กรรมการ ผศ.วิชาญ เตชิตธีระ	
กรรมการ ผศ.ดร.ปรีชา เทียนสมประสงค์	
กรรมการ อ.สุรชาติ กมลดิลก	

.....

(ผู้ช่วยศาสตราจารย์วิชาญ เตชิตธีระ)

หัวหน้าภาควิชา

ลิขสิทธิ์ของภาควิชา ฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โครงงานพิเศษ เรื่อง การสร้างชุดขับเคลื่อนแท่นสแกน xy ควบคุมด้วยคอมพิวเตอร์

นักศึกษา	นางสาวกนกพร	โพธิ์เทียนทอง
	นางสาวชุติมา	เรืองไทย
	นางสาวศันสนีย์	ทวีธนประสิทธิ์
ภาควิชา	ฟิสิกส์ประยุกต์	
สาขาวิชา	โซลิดสเตทอิเล็กทรอนิกส์	
ปีการศึกษา	2546	
อาจารย์ที่ปรึกษา	ผศ.ดร.วราวุฒิ เถาลัดดา	

บทคัดย่อ

โครงงานพิเศษนี้เป็นการออกแบบและสร้างชุดควบคุมแท่นสแกน XY แท่นสแกนขนาดพื้นที่การสแกน 13x18 เซนติเมตร ขับเคลื่อนด้วยสเตปชนิด 5 เฟส ระบบควบคุมที่สร้างขึ้นใช้ Super Vexta รุ่น UDK5107N ของบริษัท Oriental Motor เป็นชุดขับเคลื่อนมอเตอร์ ซึ่งถูกควบคุมด้วยไมโครคอนโทรลเลอร์ MSC51 (AT89C51) โปรแกรมที่เขียนด้วยภาษาวิชวลเบสิกจะรับข้อมูลตำแหน่ง ทิศทางการเคลื่อนที่และความเร็วจากผู้ใช้งานและส่งผ่านไปยังชุดควบคุมมอเตอร์ผ่านทางพอร์ตอนุกรม RS232 ของไมโครคอมพิวเตอร์ส่วนบุคคล โปรแกรมที่เขียนด้วยภาษาแอสเซมบลีในไมโครคอนโทรลเลอร์จะประมวลผลและส่งสัญญาณพัลส์แบบอนุกรมไปยังชุดขับเคลื่อนมอเตอร์เพื่อควบคุมการเคลื่อนที่ของแท่นสแกน จากการทดลองพบว่าชุดควบคุมที่สร้างขึ้นสามารถควบคุมตำแหน่งและทิศทางการหมุนของมอเตอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Special Project Title	Construction of computerized xy Scanning System	
Name	Miss Kanokporn	Potiantong
	Miss Chutima	Ruangthai
	Miss Sansanee	Taweetanaprasit
Department	Appied Physics	
Program	Solidstate electronics	
Academic Year	2003	
Special project Advisor	Asst. Prof. Dr. Warawoot Thowladda	

ABSTRACT

This project proposes the design and construction of a computerized XY scanning system. The XY table of 13x18 cm scanning area was driven by two 5-phase step motors. The scanning system in this project used Super Vexta, model UDK5107N from Oriental Motor as a motor driver. The driver was controlled by MSC51 (AT89C52) microcontroller. The position, direction and speed commands for the table movement were received and transferred to the microcontroller via RS232 Serial port of the personal computer by the program which developed by Microsoft Visual Basic compiler language. The assembly program in the microcontroller processed the command and sent series of pulse to the driver to move the XY table. From the experiment, the position, direction and speed of the XY table could be controlled successfully.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานพิเศษนี้ สามารถสำเร็จลุล่วงไปได้ด้วยดีเนื่องจากความสนับสนุน ช่วยเหลือ และความอนุเคราะห์จากบุคคลหลายท่าน ซึ่งต้องขอขอบพระคุณเป็นอย่างสูงไว้ ณ ที่นี้

ผศ.ดร.วราวุฒิ เถาลัดดา	ผู้ซึ่งถ่ายทอดวิชาความรู้ทั้งด้านวิชาการและให้คำปรึกษาในการทำงานเป็นอย่างดีมาตลอดจนโครงงานนี้สามารถสำเร็จลุล่วงไปได้ด้วยดี
บิดา-มารดา	ผู้ซึ่งให้กำเนิด เลี้ยงดู และให้โอกาสในการศึกษามาตลอด จนสามารถสำเร็จการศึกษาระดับชั้นอุดมศึกษา
คุณสมภพ แซ่เจี่ย	ที่ให้คำแนะนำซึ่งมีประโยชน์อย่างยิ่งในการเขียนโปรแกรมและในการทดลองเป็นอย่างดี
คุณมนตรี ผักกระโทก และ คุณชัยสุริยา ประภาศิริ	ที่ให้ความช่วยเหลือและให้คำแนะนำเกี่ยวกับวงจรอิเล็กทรอนิกส์
คุณอุดม เพียรภูงา	ที่คอยให้การช่วยเหลือในการเขียนโปรแกรมทำให้โครงงานนี้สามารถสำเร็จได้

นางสาวกนกพร โพธิ์เทียนทอง
นางสาวชุติมา เรืองไทย
นางสาวศันสนีย์ ทวีธนประสิทธิ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	ฉ
สารบัญรูป	ช
บทที่ 1 บทนำ	
1.1 ความเป็นมาของโครงงานพิเศษ	1
1.2 วัตถุประสงค์	1
1.3 ขอบเขตของการวิจัย	1
1.4 ขั้นตอนของการวิจัยและวิธีการดำเนินงาน	2
1.5 ผลที่คาดว่าจะได้รับ	2
บทที่ 2 ทฤษฎีและหลักการ	
2.1 หลักการทำงานเบื้องต้นของมอเตอร์สเตปเปอร์	3
2.2 การทำงานใน รูปแบบ-ครึ่ง-สเตป	6
2.3 รูปแบบไมโครสเตป	6
2.4 วิธีการหน่วงความแกว่งไกวของโรเตอร์	7
2.5 มอเตอร์สเตปเปอร์แบบแม่เหล็ก-ถาวร	7
2.6 การขับเคลื่อนมอเตอร์สเตปเปอร์	8
2.7 มอเตอร์สเตปเปอร์เชิงเส้น	10
2.8 ชุดขับ	21
2.9 การใช้ Chip สำเร็จรูป	26
2.10Driver	27
2.11วงจร Suppressors	27
2.12การลดเวลาหน่วงกระแสให้กับ Stepping motor	31
2.13การสร้างสัญญาณ Pulse เพื่อขับ Stepping motor ด้วย Microprocessor	34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

หน้า

บทที่ 3 วิธีดำเนินการวิจัย

3.1 ส่วนประกอบของโครงงาน 40

3.2 ส่วนควบคุมระบบขับเคลื่อน 40

3.3 ส่วนควบคุมการทำงาน 42

บทที่ 4 ผลการทดลองและอภิปรายผล

4.1 โปรแกรมภาษาแอสเซมบลี 43

4.2 โปรแกรมวิชวลเบสิก 43

บทที่ 5 สรุปและข้อเสนอแนะ

5.1 ผลการวิจัยและดำเนินงาน 45

5.2 ประสิทธิภาพของชุดขับเคลื่อนแท่นสแกน xy 45

5.3 วิจารณ์ผลการทดลอง 45

5.4 ข้อเสนอแนะและแนวทางในการทดลอง 45

เอกสารอ้างอิง

ภาคผนวก

ภาคผนวก ก โปรแกรมควบคุมชุดขับเคลื่อนแท่นสแกน xy ด้วย 89C51

ภาคผนวก ข โฟลชาร์ทของโปรแกรมในส่วนต่างๆ

ภาคผนวก ค โปรแกรมควบคุมชุดขับเคลื่อนแท่นสแกน xy
บนระบบปฏิบัติการวินโดว์

ภาคผนวก ง คู่มืออุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

หน้า

ตาราง 2.1 แสดงความแตกต่างของสเตปเปอร์มอเตอร์ชนิด PM และ VR 8

ตาราง 2.2 แสดง Function ของ Logic Gate และ Flip-flop 23

ตาราง 2.3 แสดงข้อมูลในแอดเดรส 0000:0411 ที่ใช้แจ้งจำนวนพอร์ตอนุกรม 38

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

รูปที่ 2.1 โครงสร้างพื้นฐานของสเตปเปอร์มอเตอร์ชนิดความต้านทานแม่เหล็กแปรเปลี่ยน 3
รูปที่ 2.2 การกระตุ้นสเตปเปอร์มอเตอร์ชนิดความต้านทานแม่เหล็กแปรเปลี่ยนที่ขั้ว BCDA 5
รูปที่ 2.3 ไดอะแกรมอย่างง่ายของสเตปเปอร์มอเตอร์ชนิดแม่เหล็กถาวร 8
รูปที่ 2.4 (ก) วงจรขับสเตปเปอร์มอเตอร์ชนิด 2 เฟส 9
รูปที่ 2.4 (ข) วงจรสวิตช์จากรูป (ก) 9
รูปที่ 2.5 (ก) สัญญาณลอจิกที่ป้อนให้ทรานซิสเตอร์ 10
รูปที่ 2.5 (ข) กระแสที่ป้อนให้มอเตอร์ 10
รูปที่ 2.5 (ค) การหมุนของโรเตอร์สำหรับมอเตอร์ในรูปที่ 2.3 และการขับในรูปที่ 2.4.ก 10
รูปที่ 2.6 ส่วนประกอบของสเตปเปอร์มอเตอร์เชิงเส้นชนิด 2 เฟส 11
รูปที่ 2.7 การหมุนของสเตปเปอร์มอเตอร์เชิงเส้นชนิด 2 เฟส 13
รูปที่ 2.8 โครงสร้างของสเตปเปอร์มอเตอร์แบบ 2 แกน 15
รูปที่ 2.9 ส่วนโค้งของแรงเทียบกับระยะทางที่เคลื่อนไป 16
รูปที่ 2.10 ตำแหน่งคำสั่งการและตำแหน่งจริงของอาร์เมเจอร์เมื่อเทียบกับเวลาสำหรับรูปแบบความเร็วชนิดสามเหลี่ยม 18
รูปที่ 2.11 มอเตอร์สเตปเปอร์เชิงเส้นที่ขับเคลื่อนด้วยเทคนิค burst 19
รูปที่ 2.12 ลักษณะสมบัติ ความเร็ว-แรง ของมอเตอร์สเตปเปอร์เชิงเส้นทั่วไป 20
รูปที่ 2.13 การขับ Stepping motor 21
รูปที่ 2.14 Block diagram ของระบบการขับ Stepping motor 22
รูปที่ 2.15 unidirection logic สำหรับการกระตุ้นแบบ Full step ของมอเตอร์ 4 เฟส แบบต่างๆ 24
รูปที่ 2.16 Logic sequencer สำหรับการกระตุ้น Stepping motor 4 เฟส แบบ Single Step 25
รูปที่ 2.17 การกระตุ้น Stepping motor 3 เฟส แบบ Half step 26
รูปที่ 2.18 IC ควบคุม Stepping motor ของ Nanotec GmbH 26
รูปที่ 2.19 วงจรขับแบบ Unipolar อย่างง่าย 27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

	หน้า
รูปที่ 2.20 วงจรสมมูลย์ของขดลวด stepping motor	28
รูปที่ 2.21 วงจร Diode Suppressors	28
รูปที่ 2.22 วงจร Diode/Resistor Suppressors	29
รูปที่ 2.23 วงจร Zener diode Suppressors	29
รูปที่ 2.24 เปรียบเทียบผลของวงจร Suppressors ต่าง ๆ	30
รูปที่ 2.25 ตัวอย่างการใช้งาน วงจร Zener diode Suppressors	30
รูปที่ 2.26 Capacitor Suppressors	30
รูปที่ 2.27 การใช้ Series Resistance	31
รูปที่ 2.28 การขับแบบ R – L	32
รูปที่ 2.29 การใช้แหล่งจ่ายคู่	33
รูปที่ 2.30 การใช้แหล่งจ่ายคู่ที่ควบคุมด้วย Microprocessor	33
รูปที่ 2.31 รูปร่างของแรงดันและกระแสเมื่อขับแบบ Dual Power supply	34
รูปที่ 2.32 ลักษณะการทำงานของการใช้ Microprocessor กระตุ้น Stepping motor	34
รูปที่ 3.1 ส่วนประกอบของโครงงาน	40
รูปที่ 3.2 วงจรชุดไมโครคอนโทรลเลอร์	41
รูปที่ 4.1 หน้าตาของโปรแกรม	44

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

1.1 ความเป็นมาของโครงงานพิเศษ

แท่นสแกน X-Y เป็นอุปกรณ์ที่ใช้กันอย่างแพร่หลายในเครื่องมือทางวิทยาศาสตร์หลายประเภท เช่น เครื่องมือสำหรับตัดวัตถุด้วยเลเซอร์ ซึ่งแท่นสแกนนี้จำเป็นสำหรับใช้วางวัตถุให้สามารถเลื่อนไปตามตำแหน่งที่ต้องการได้ในทุกทิศทาง โดยแท่นนี้จะมีมอเตอร์ 2 ตัวควบคุมการเคลื่อนที่ในแนวแกน X และแกน Y อย่างละตัว ซึ่งแท่นสแกนเดิมที่เรามีอยู่นั้นเป็นแบบ manual ไม่สะดวกในการใช้งาน เพราะขาดความแม่นยำและยากต่อการกำหนดตำแหน่งของวัตถุ จากปัญหาที่ผ่านมาจากการตัดโลหะด้วยเลเซอร์นั้น ถ้าเราไม่เลื่อนตำแหน่งของวัตถุแต่ไปเลื่อนตำแหน่งของเลเซอร์แทนนั้น จะเห็นว่ามีความยุ่งยาก และต้องจัดอุปกรณ์อย่างดีมากๆ เพราะถ้าเกิดความผิดพลาดขึ้นมาแล้วอาจมีอันตรายถึงชีวิตได้ เช่นเลเซอร์ลอดผ่านออกมาโดยผู้ทำการทดลอง ทางผู้จัดทำโครงงานพิเศษนี้จึงได้เล็งเห็นถึงความสำคัญในการพัฒนาแท่นสแกนที่มีอยู่แล้วให้มีประสิทธิภาพมากขึ้นเพื่อความปลอดภัยในการทำงาน จึงคิดค้นการ control แท่นสแกน X-Y ให้มีฟังก์ชันในการควบคุมความเร็ว กำหนดตำแหน่งของวัตถุ ได้ตามที่เราต้องการโดยให้มีการสั่งงานผ่านคีย์บอร์ดของคอมพิวเตอร์ เพื่อประโยชน์ในการนำไปใช้ในการพัฒนางานวิจัยทางด้านฟิสิกส์ต่อไปได้

1.2 วัตถุประสงค์

1. เพื่อศึกษาการทำงานและการควบคุมสเตปมอเตอร์ชนิด 5 เฟส
2. เพื่อศึกษาการทำงานของคอนโทรลเลอร์
3. เพื่อศึกษาและสร้างระบบควบคุมชุดขับเคลื่อนแท่นสแกน XY ด้วยคอนโทรลเลอร์
4. เพื่อศึกษาการเขียนโปรแกรมภาษาแอสแซมบลีและวิชวลเบสิก

1.3 ขอบเขตการวิจัย

1. ศึกษาการทำงานและการควบคุมสเตปมอเตอร์ชนิด 5 เฟส
2. สร้างระบบควบคุมชุดขับเคลื่อนแท่นสแกน XY ควบคุมด้วยไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. เขียนโปรแกรมควบคุมการทำงานด้วยภาษาแอสแซมบลีและวิชวลเบสิก
4. ทดสอบการทำงานระบบควบคุมการขับเคลื่อนแท่นสแกน XY

1.4 ขั้นตอนการวิจัยและวิธีการดำเนินงาน

1. ศึกษาการทำงานและการควบคุมสเตปมอเตอร์ชนิด 5 เฟส
2. สร้างระบบควบคุมชุดขับเคลื่อนมอเตอร์ด้วยไมโครคอนโทรลเลอร์
3. การเขียนโปรแกรมภาษาแอสแซมบลีสำหรับระบบควบคุมชุดขับเคลื่อนมอเตอร์
4. การเขียนโปรแกรมภาษาวิชวลเบสิกสำหรับการรับคำสั่งควบคุมจากผู้ใช้และส่งไปยังระบบควบคุมชุดขับเคลื่อนมอเตอร์

1.5 ผลที่คาดว่าจะได้รับ

1. เข้าใจการทำงานและการควบคุมสเตปมอเตอร์ชนิด 5 เฟส
2. ได้ระบบควบคุมชุดขับเคลื่อนมอเตอร์ควบคุมด้วยไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

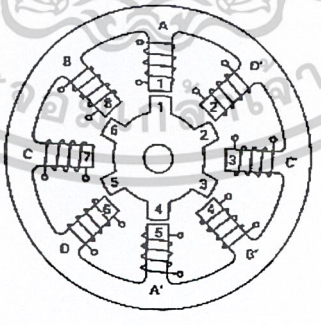
บทที่ 2
ทฤษฎีและหลักการ

2.1 หลักการทำงานเบื้องต้นของมอเตอร์สเตปเปอร์

(Principle operation of stepper motor)

โครงสร้างพื้นฐานของมอเตอร์สเตปเปอร์มีอยู่สองแบบ คือ (1) แบบความต้านทานแม่เหล็กแปรเปลี่ยน (variable reluctance : VR :) และ (2) แบบแม่เหล็กถาวร (permanent magnet : PM :) ถึงแม้ว่า แบบที่สอง (PM) จะเป็นแบบที่ถูกนำมาใช้งานอย่างกว้างขวางกว่า ในปัจจุบันนี้ก็ตามแต่แบบที่หนึ่ง (VR) จะมีการทำงานที่เข้าใจได้ง่ายกว่า เพราะฉะนั้น จึงขอพิจารณาการทำงานของมอเตอร์แบบนี้ก่อนในหัวข้อนี้

โครงสร้างทั่วไปของมอเตอร์สเตปเปอร์แบบ VR แสดงอยู่ในรูปที่ 2.1 จะเห็นได้ว่าไม่เหมือนกับมอเตอร์เซอร์โว ทั้งโรเตอร์และสเตเตอร์มีโครงสร้างแบบเป็นซี่ (toothed) หรือ ขั้ว หลักการทำงานพื้นฐานของมอเตอร์แบบนี้ก็คือ โรเตอร์และสเตเตอร์ จะต้องมีจำนวนซี่ไม่เท่ากัน เป็นต้นว่า สเตเตอร์ ที่แสดงในรูปที่ 1 มีแปดซี่ (อยู่ที่ตำแหน่งทุก 45°) และโรเตอร์มีหกซี่ (อยู่ที่ตำแหน่งทุก 60°) แต่ละซี่ของสเตเตอร์มีขดลวดพันให้เกิดเป็นขั้วตรงข้ามกัน (เช่น $A - A'$) ลักษณะจับกลุ่มเข้าด้วยกันเป็นเฟส (phase) ในตัวอย่างนี้จะเห็นได้ว่ามี สี่เฟส (โดยใช้ตัวอักษร A, B, C และ D)



รูปที่ 2.1 โครงสร้างพื้นฐานของสเตปเปอร์มอเตอร์ชนิดความต้านทานแม่เหล็กแปรเปลี่ยน

การทำงานของมอเตอร์สเตปเปอร์แบบ VR และค่อนข้างง่าย โดยวางอยู่บนพื้นฐานของค่า "ความต้านทานแม่เหล็กที่น้อยที่สุด" (minimum reluctance) ซึ่งเป็นลักษณะของการพยายามจัดแนวเส้นแรงแม่เหล็กของวงจรแม่เหล็กเองที่จะเดินไปในเส้นทางที่สั้นที่สุดมีช่องว่างอากาศ (air gap) น้อยที่สุด จึงทำให้โรเตอร์ต้องหมุนหาจุดที่เกิดค่าความต้านทานแม่เหล็กที่น้อย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่สุด ดังนั้นตามตัวอย่างในรูปที่ 2.1 ที่แสดงอยู่เมื่อจ่ายไฟเข้าที่ เฟส A ซี่โรเตอร์หมายเลข 1 และ 4 (R4,1) จะเลื่อนไปอยู่ตามแนวซี่สเตเตอร์หมายเลข 1 และ 5 (S5,1) และจะคงอยู่ที่ตำแหน่งนี้นานตราบเท่าที่เฟสนี้ยังคงมีการจ่ายไฟเข้ามาลักษณะเช่นนี้เห็นการเข้าสู่จุดสมดุลเป็นการที่มอเตอร์เคลื่อนที่ไปได้ "หนึ่งสเตป"

เป็นเรื่องสำคัญที่จะต้องเข้าใจด้วยว่า ตราบเท่าที่ยังคงมีการกระตุ้นที่ขด $A - A'$ จะเกิดมี *ทอร์คยึดติด (holding torque)* อยู่เสมอ ดังนั้นถ้าทอร์คที่ป้อนเข้ามาจากภายนอกมีค่าน้อยกว่าจะไม่เกิดการหมุนใดๆทั้งสิ้น ถึงแม้ว่าจะเพิ่มค่ากระแสที่ไหลเข้าไปในเฟสๆนี้อีกเท่าใดก็ตาม ก็จะเป็นแต่เพียงการเพิ่มค่าทอร์คยึดติด มอเตอร์ยังคงอยู่ในสภาวะ "lock" เช่นเดิม ลักษณะเช่นนี้ควรนำไปเปรียบเทียบกับมอเตอร์เซอร์โว ซึ่งเมื่อเพิ่มการกระตุ้นจะทำให้โรเตอร์หมุนเร็วขึ้น

คราวนี้สมมติให้การกระตุ้นเคลื่อนย้ายจากเฟส A ไปที่เฟส B ก็จะทำให้ R6,3 ไปเรียงตามแนว S8,4 ดังแสดงในรูปที่ 2.2 (ก)ทำให้เห็นได้ว่าโรเตอร์จะหมุนในทิศทางตามเข็มนาฬิกาไปเป็นมุม 15° (60 – 45 = 15) ลักษณะเช่นนี้เมื่อเกิดขึ้นซ้ำกับเฟส C,D และย้อนกลับมายัง A (ดูรูปที่ 2.2.ข,ค,ง) ในแต่ละกรณีจะไปได้สเตปละ 15° และลำดับขั้นการกระตุ้นจะเสร็จสิ้นสมบูรณ์ครบทุกเฟส (A – B – C – D – A) ทำให้หนึ่งรอบการกระตุ้นเกิดการหมุนไป 60° ในกรณีนี้การทำให้มอเตอร์หมุนครบหนึ่งรอบ ต้องกระตุ้นให้ครบหกรอบ เพราะฉะนั้น มอเตอร์ตัวนี้จึงเป็นมอเตอร์ชนิด 24 สเตป/รอบ (24-step/revolution motor)

ผู้อ่านคงจะสามารถพิสูจน์ได้ด้วยตนเองได้ว่า การกระตุ้นเฟส A,D,C,B,A จะทำให้เกิดการหมุนในทิศทางทวนเข็มนาฬิกาไป 60° ซึ่งเห็นได้ว่า การเปลี่ยนทิศทางการหมุนสามารถกระทำได้โดยไม่ยากนัก และแสดงให้เห็นว่าการเปลี่ยนทิศทางจะทราบ ก็ต่อเมื่อต้องกระตุ้นไปอย่างน้อย 3 เฟส

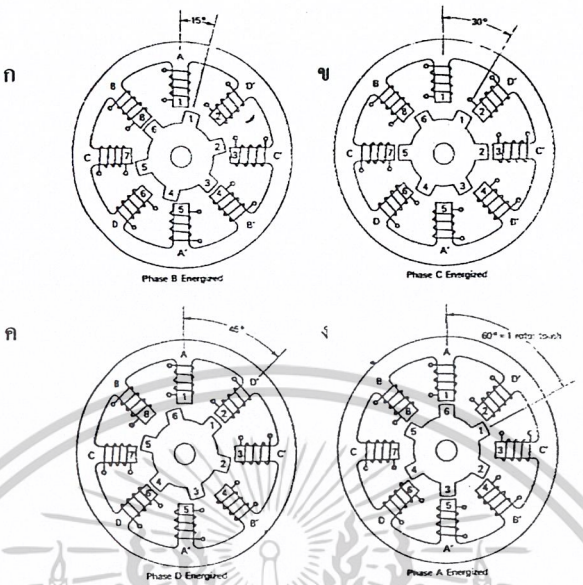
มุมในการเคลื่อนที่แต่ละสเตป (step angle) เป็นความสัมพันธ์กันระหว่าง ซี่ขั้วสเตเตอร์ N_s และซี่ขั้วโรเตอร์ N_r ดังนี้

$$\text{Step angle} = 360^{\circ}\frac{|N_s - N_r|}{N_s N_r} \qquad \text{สมการ 2.1}$$

และจะได้อีกว่า

$$\text{Number step / rev} = \frac{N_r N_s}{|N_s - N_r|} \qquad \text{สมการ 2.2}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 การกระตุ้นสเตปเปอร์มอเตอร์ชนิดความต้านทานแม่เหล็กแปรเปลี่ยนที่ขั้ว BCDA

โดยลักษณะทางกายภาพ โรเตอร์ของมอเตอร์สเตปเปอร์ทุกตัว จะแสดงคุณสมบัติการตอบสนองแบบอัตราหน่วงต่ำ (Underdamped) ในการเคลื่อนที่ จากสเตปหนึ่งไปยังอีกสเตปหนึ่ง จะเห็นได้ว่าในขณะที่การกระตุ้นเปลี่ยนไปยังเฟสข้างเคียง โรเตอร์จะพยายามเคลื่อนไปยังจุดสมดุลจุดใหม่ ถึงแม้ว่าทอร์คอัตราเร่งจะเป็นศูนย์เพราะซี่ขั้วโรเตอร์และสเตเตอร์ไม่ได้อยู่แนวเดียวกัน แต่ความเร็วเชิงมุมของโรเตอร์ยังไม่เป็นศูนย์ผลที่เกิดขึ้นจะเกิดการพุ่งขึ้นสูงกว่า (overshoot) ที่ตำแหน่งสมดุลตำแหน่งใหม่ขณะเข้าสู่ตำแหน่งสมดุลก็จะเกิดทอร์คอัตราเร่งดึงโรเตอร์กลับมายังจุดสมดุลที่เพิ่งผ่านมา (ในทิศทางตรงกันข้าม) ในความเป็นจริงลักษณะเช่นนี้จะเกิดซ้ำกันหลายครั้งก่อนที่โรเตอร์จะหยุดนิ่ง โดยทั่วไปช่วงระยะเวลาของความหน่วงการแกว่งไกวจะขึ้นอยู่กับขนาดของมุมสเตป (step angle) หมายความว่า ขนาดมุมสเตปยิ่งมากการพุ่งขึ้นสูงกว่ายิ่งเกิดขึ้นมาก ในการใช้งานบางแบบลักษณะเช่นนี้ยอมให้เกิดขึ้นไม่ได้ (เป็นต้นว่าในระบบหุ่นยนต์) แต่อาจใช้ได้ในงานบางแบบ(เช่นใน printer)

มอเตอร์สเตปเปอร์สามารถทำให้มีจำนวนสเตป/รอบ (step/rev) ได้ในย่านที่กว้างมากแต่เนื่องด้วยข้อจำกัดของราคา ในทางปฏิบัติจะอยู่ที่ 200 สเตป/รอบ ทำให้ได้มุมสเตปเท่ากับ 1.8° ในการประยุกต์ใช้งานหลายชนิด ค่ามุมสเตปที่ค่อนข้างน้อยขนาดนี้นับว่าเหมาะสม (หมายความว่าการแกว่งไกวของโรเตอร์ไม่รบกวนการทำงานของอุปกรณ์ตัวอื่น) อย่างไรก็ตาม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อต้องการจะได้มุมสเตปที่ละเอียดมากขึ้น (ซึ่งเป็นวิธีหนึ่งที่จะลดปัญหาการ overshoot ที่อธิบายมาแล้ว) ก็สามารถใช้เทคนิคอีกแบบหนึ่งได้

2.2 การทำงานใน รูปแบบ-ครึ่ง-สเตป (Half-step-mode operation)

จากหัวข้อที่แล้วเป็นการกระตุ้นทีละเฟสในแต่ละช่วงเวลา ทำให้เกิดมุมสเตปตามสมการ 2.1 วิธีการนี้เป็น รูปแบบเต็ม-สเตป (full-step mode) คราวนี้สมมติว่าถ้าเฟสใกล้เคียงกันสองเฟสถูกกระตุ้นพร้อมกันในเวลาเดียวกัน (เช่นเฟส A และ B) ก็จะเห็นได้ว่าจะเกิดจุดสมดุลบางแห่งระหว่าง จุดเต็มสเตปทั้งสองจุดนั้น ถ้าคุณสมบัติทางไฟฟ้าของขดลวดทั้งสองขดเหมือนกันและถูกกระตุ้นด้วยแรงเคลื่อนไฟฟ้าที่เท่ากันทั้งสองเฟส จุดสมดุลจุดใหม่จะต้องอยู่โดยประมาณที่ตำแหน่งกึ่งกลางระหว่างจุด-เต็ม-สเตป ทั้งสองจุดนั้น วิธีการนี้เมื่อทำซ้ำกับ BC, CD และ DA จะทำให้เกิดจุดสมดุลกึ่งกลางขึ้นมาและจะเห็นได้อย่างชัดเจนว่าถ้าลำดับขั้นการกระตุ้นเป็นดังนี้ A,AB,B,BC,C,CD,D,DA,A และต่อๆไป จะทำให้โรเตอร์มีจำนวนสเตปการหมุนเป็นสองเท่าของจำนวนสเตปการหมุนเดิม (เมื่อเทียบกับรูปแบบ-เต็ม-สเตป) และมอเตอร์สเตปเปอร์ในรูปที่ 2.1 จะมีจุดสมดุลที่แตกต่างกันได้ถึง 48 จุดในหนึ่งรอบลักษณะการทำงานเช่นนี้คือการทำงานในรูปแบบครึ่ง-สเตป เนื่องจากในการหมุนต่อหนึ่งสเตปลดลงเหลือเพียงครึ่งเดียว การพุ่งขึ้นสูงกว่าของโรเตอร์เนื่องจากการเคลื่อนที่ จากจุดหนึ่งไปอีกจุดหนึ่งก็จะลดลง และการกระตุ้นในลักษณะย้อนกลับ ก็จะทำให้มอเตอร์หมุนกลับทางเป็นทิศทางทวนเข็มนาฬิกาได้เช่นเดียวกันอีกทั้งวงจรการสวิตช์ในการทำให้เกิดการทำงานใน รูปแบบ-ครึ่ง-สเตป แน่นอนว่าต้องซับซ้อนและราคาแพงกว่า วงจรในรูปแบบ-เต็ม-สเตป

2.3 รูปแบบไมโครสเตป (Microstep mode)

การกระตุ้นไม่จำเป็นที่จะต้องกระตุ้นเฟสใกล้เคียงกันด้วยแรงเคลื่อนที่เท่ากันเสมอไป ในความเป็นจริงเป็นไปได้ว่าสามารถกระตุ้นด้วยแรงเคลื่อนระดับใดก็ได้ที่อยู่ระหว่างระดับ 0 โวลต์ จนถึง V โวลต์ ตามข้อกำหนดของตัวมอเตอร์เข้าที่เฟสที่สองในกรณีนี้ จุดสมดุลที่เกิดขึ้นจะเกิดขึ้นที่ตำแหน่งบางตำแหน่งที่ไม่อยู่ตรงกลางพอดี ในระหว่างซี่ขั้วทั้งสองเหมือนกับแบบ-เต็ม-สเตป ลักษณะเช่นนี้ คือรูปแบบการทำงานแบบ *ไมโครสเตป (microstepping)* โดยทั่วไปแล้วขนาดของไมโครสเตป จะเป็นการแบ่งออกเป็นเศษส่วนจากระยะเต็ม-สเตป เป็นเลขจำนวนเต็มของสองเป็นต้นว่า 2,4,8,16,32,... ซึ่งจะทำได้ไม่ยากนักโดยใช้ชุดควบคุมมอเตอร์สเตปเปอร์เป็นตัวจัดการถึงแม้ว่า การทำงานแบบไมโครสเตป จะต้องใช้วงจรที่ซับซ้อนกว่า ซึ่งหมายถึงว่าต้องมีราคาแพงกว่าขึ้นบ้างเมื่อเทียบกับแบบ-เต็ม-สเตป แต่การใช้งานที่ความเร็วรอบต่ำๆ มอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะหมุนได้นุ่มนวลมาก ในการประยุกต์ใช้กับระบบหุ่นยนต์ ลักษณะเช่นนี้สำคัญมากสำหรับหุ่นยนต์เพราะการเคลื่อนไหวจะมีความ smooth และการแกว่งไกวเมื่อเคลื่อนไปยังตำแหน่งที่ต้องการ เป็นสิ่งที่ยอมรับไม่ได้

2.4 วิธีการหน่วงความแกว่งไกวของโรเตอร์

(Methods of damping rotor oscillation)

การหน่วงไม่ให้โรเตอร์เกิดการแกว่งไกวสามารถกระทำได้หลายทาง เช่น การเพิ่มความเฉื่อยเนื่องจากความหนืด (viscous inertia) ซึ่งบางทีก็เรียกว่า Lanchester damper หรืออาจจะใช้ จานความฝืด (friction disk) หรือชุดหน่วงกระแส-ไหลวน (eddy-current damper) ก็ได้ถึงแม้เทคนิคเหล่านี้จะทำให้ได้ผลตามที่ต้องการก็ตามแต่จะเป็นการเพิ่มแรงเฉื่อยเพราะฉะนั้นจึงอาจส่งผลกระทบที่ตรงข้ามกับสิ่งที่ต้องการกับ การตอบสนองแบบฉับพลัน (transient response) ต่อการหมุนของระบบ

เทคนิคทางอิเล็กทรอนิกส์ที่ช่วยขจัดปัญหาเนื่องจากการเพิ่มขึ้นของแรงเฉื่อย คือ เทคนิคการหน่วงแบบ แบง-แบง (bang-bang damping) แนวความคิดก็คือ ให้มอเตอร์ทำงานในลักษณะที่มีอัตราเร่งเหมือนปกติธรรมดา แต่ก่อนที่โรเตอร์จะเคลื่อนไปถึงตำแหน่งที่ต้องการจะกลับลำดับการกระตุ้นเฟสของมอเตอร์ ซึ่งส่งผลให้โรเตอร์หยุดนิ่งอยู่ที่สมดุล โดยไม่มีการพุ่งขึ้นสูงกว่า เป็นที่แน่ชัดว่า ช่วงเวลาในการกลับเฟสเป็นเวลาวิกฤตและช่วงเวลาในการสวิตช์ขึ้นอยู่กับฟังก์ชันของพารามิเตอร์ในระบบด้วย ในระบบหุ่นยนต์ที่แรงเฉื่อยของจุดเชื่อมต่อแต่ละจุดเปลี่ยนแปลงตามตำแหน่งการเคลื่อนที่ของแกนทำให้เพิ่มความซับซ้อนในการกำหนดช่วงเวลาซึ่งจะเปลี่ยนแปลงตลอดเวลาเข้าไปอีก

2.5 มอเตอร์สเตปเปอร์แบบแม่เหล็ก-ถาวร

(Permanent-magnet stepper motor)

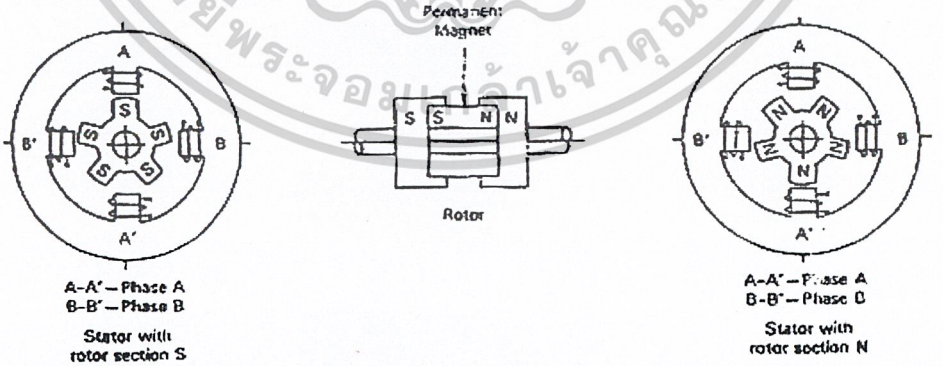
ดังได้กล่าวมาแล้วว่า มอเตอร์สเตปเปอร์แบบ PM เป็นมอเตอร์ที่ถูกนำมาใช้งานมากที่สุดชนิดหนึ่ง โครงสร้างประกอบด้วย สเตเตอร์แบบหลายเฟสและโรเตอร์ซึ่งมีแม่เหล็กถาวรอยู่ สองส่วนลักษณะเหมือนกับมอเตอร์แบบ VR ทั้งสองส่วนจะมีซี่ขั้วเหมือนกัน (ดูรูปที่ 2.3) ที่แตกต่างกันก็คือ ปลายแต่ละด้านที่อยู่ตรงข้ามกันของโรเตอร์จะเป็นสนามแม่เหล็กขั้วเหนือและขั้วใต้ โดยมีซี่ขั้วที่วางเหลื่อมกันในลักษณะเต็มซี่ฟันลักษณะการทำงานของมอเตอร์สเตปเปอร์แบบ PM จะเหมือนกับแบบ VR คือสามารถขับเคลื่อนแบบ เต็ม-สเตป, ครึ่ง-สเตป และไมโครสเตป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตาราง 2.1 แสดงความแตกต่างของสเตปเปอร์มอเตอร์ชนิด PM และ VR

Characteristic	PM motor	VR motor
1. Motor	Magnetized	Not magnetized
2. Rotor position	Depends on stator excitation polarity	Independent of stator excitation polarity
3. Rotor inertia	High due to magnet	Low (no magnet)
4. Mechanical response	Not as good (due to high inertia)	Good (low-inertia device)
5. Inductance	Low due to rotor offset	Generally high for same torque rating
6. Electrical response	Faster current rise (due to low inductance)	Slower current rise (due to higher inductance)

As indicated in Table 2.1 the PM stepper rotor position is dependent on the polarity of the phase excitation. Consequently, a bipolar signal is required to achieve bidirectional control. With respect to Figure 2.3, it is seen that this



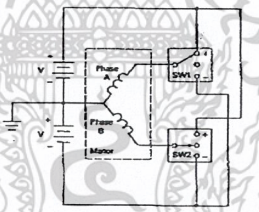
รูปที่ 2.3 ไดอะแกรมอย่างง่ายของสเตปเปอร์มอเตอร์ชนิดแม่เหล็กถาวร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

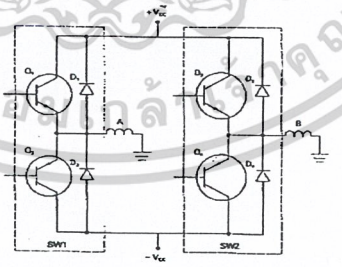
2.6 การขับเคลื่อนมอเตอร์สเตปเปอร์ (Stepper motor drives)

จากตาราง 2.1 ตำแหน่งโรเตอร์ของมอเตอร์สเตปเปอร์แบบ PM ขึ้นอยู่กับขั้วแม่เหล็กของเฟสที่ถูกกระตุ้น ซึ่งหมายความว่า สัญญาณแบบสองขั้ว (bipolar signal) สามารถควบคุมทิศทางการหมุนได้สองทิศทาง จากรูปที่ 2.3 จะเห็นได้ว่า มอเตอร์สามารถหมุนได้ด้วยการกระตุ้นเพียงสองเฟสเท่านั้น ในขณะที่ต้องกระตุ้นมอเตอร์สเตปเปอร์แบบ VR ถึงสี่เฟส (ดูรูปที่ 2.1)

เมื่อใช้แหล่งจ่ายกำลังชนิดปลายบรรจบสองด้าน (double-ended power supply) มอเตอร์ในรูปที่ 2.3 ก็สามารถขับเคลื่อนแบบเต็มสเตปได้ด้วยรูปแบบการสวิตช์ที่แสดงอยู่ในรูปที่ 2.4.ก จะเห็นได้ว่าการสวิตช์จำเป็นจะต้องใช้สวิตช์สามสภาวะ (tristate switch) สองตัวตำแหน่งที่แสดงในรูป คือ ตำแหน่งที่เป็นการกระตุ้นไฟบวกที่เฟส A ในขณะที่ไม่มีการกระตุ้นใดๆที่เฟส B ถ้าใช้วงจรสวิชชิ่ง ในการควบคุมจะเป็นดังที่แสดงอยู่ในรูปที่ 2.4.ข ไดโอดที่ต่อคร่อมระหว่างคอลเลคเตอร์-อิมิตเตอร์ของทรานซิสเตอร์ เป็นไดโอดที่ใช้ป้องกันความเสียหายของทรานซิสเตอร์อันเนื่องมาจากแรงเคลื่อนสูงของ back emf ของขดลวด ในช่วงเวลาการสวิตช์



(ก)



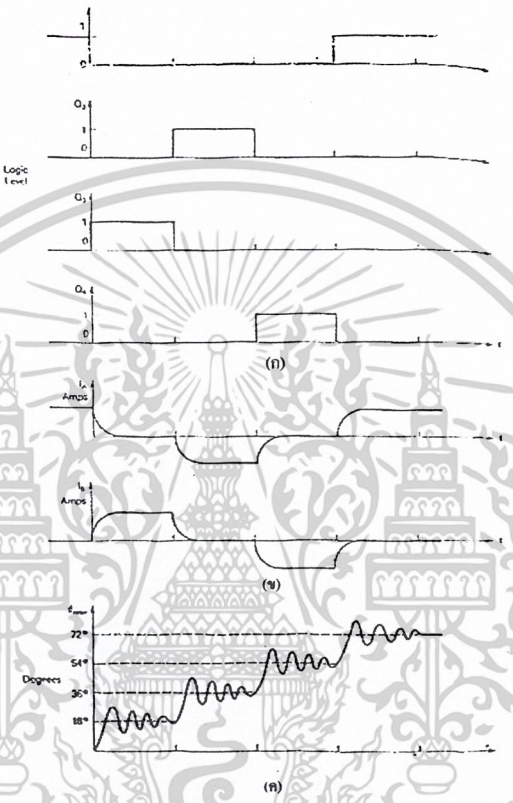
(ข)

รูปที่ 2.4 (ก) วงจรขับสเตปเปอร์มอเตอร์ชนิด 2 เฟส
(ข) วงจรสวิตช์จากรูป (ก)

สำหรับมอเตอร์ที่แสดงอยู่ในรูปที่ 2.3 เป็นมอเตอร์ที่เคลื่อนสเตปละ 18° เพราะฉะนั้นจึงเป็นมอเตอร์ชนิด 20 สเตป/รอบ สัญญาณลอจิกที่ป้อนเข้าที่ทรานซิสเตอร์ซึ่งทำให้โรเตอร์เคลื่อนไปสี่สเตป (72°) แสดงอยู่ในรูปที่ 2.5 เป็นกรณีที่แต่ละสเตปใช้เวลาเท่ากัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายถึงว่า ตัวภาระที่ติดอยู่กับโรเตอร์หมุนด้วยความเร็วคงที่แต่ในช่วงที่มีอัตราเร่ง (acceleration) หรืออัตราหน่วง (deceleration) เป็นที่แน่นอนว่า ระยะห่างของสเตปจะต้องเปลี่ยนแปลงตามระยะเวลา



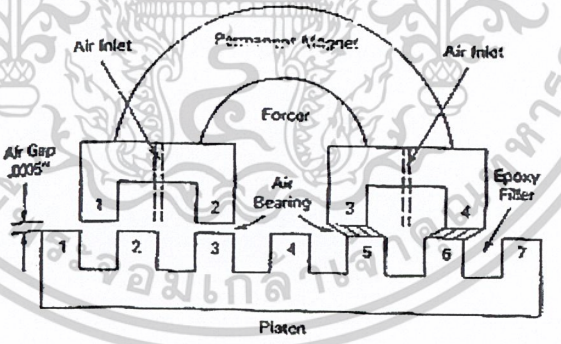
รูปที่ 2.5 (ก) สัญญาณลอจิกที่ป้อนให้ทรานซิสเตอร์ (ข) กระแสที่ป้อนให้มอเตอร์
(ค) การหมุนของโรเตอร์สำหรับมอเตอร์ในรูปที่ 2.3 และการขับในรูปที่ 2.4.ก

ในการทำงานจริง ไมโครโปรเซสเซอร์จะเป็นตัวกำหนดจำนวนสเตปที่ทำให้ตัวภาระเคลื่อนที่ไปในระยะทางระยะหนึ่งโปรเซสเซอร์จะเป็นตัวส่งสัญญาณบอกข้อมูลที่มีทั้งทิศทางและเวลาในการเคลื่อนที่ให้กับชุดฮาร์ดแวร์วงจรดิจิตอล ชุดฮาร์ดแวร์จะรับข้อมูลจำนวนสเตปที่ต้องเคลื่อนที่ไปและทำการสร้างลำดับขั้นการสวิตช์ที่ถูกต้องป้อนให้กับมอเตอร์ต่อไปจะเห็นได้อย่างชัดเจนว่า ลักษณะเช่นนี้เป็นการควบคุมแบบวงรอบปิด (closed-loop control) จะใช้กับมอเตอร์สเตปเปอร์ได้เช่นกัน อย่างไรก็ตามสิ่งหนึ่งที่อาจเกิดขึ้น คือ ค่าใช้จ่ายทั้งหมดอาจสูงขึ้น จนอาจจะมากกว่าค่าใช้จ่ายเมื่อเลือกใช้มอเตอร์เซอร์โวก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.7 มอเตอร์สเตปเปอร์เชิงเส้น (Linear stepper motor)

การเปลี่ยนแปลงที่น่าสนใจในหลักการทำงานของมอเตอร์สเตปเปอร์เชิงหมุนแบบธรรมดา คือ หลักการทำงานของมอเตอร์สเตปเปอร์เชิงเส้นของ Sawyer (Sawyer-principle linear stepper motor) เป็นการค้นพบในปี ค.ศ.1969 และได้รับการจดสิทธิบัตรการผลิตโดย Xynetics Corporation, Santa Clara, California ประกอบด้วยส่วนประกอบหลักทางกลสองส่วน ส่วนแรก คือส่วนอาร์เมเจอร์ซึ่งเคลื่อนที่ได้ (movable armature) เป็นตัวให้แรง (forcer) ที่ลอยอยู่เหนือ ส่วนที่สองซึ่งเป็นสเตเตอร์อยู่กับที่ (fixed stator) ซึ่งบางครั้งจะเรียกว่า ตัวแทน (platen) ดูรูปที่ 2.8 มีการใช้ตัวรองรับ (bearing) เพื่อให้แน่ใจว่า ระยะห่างระหว่างอาร์เมเจอร์และสเตเตอร์คงที่ สิ่งที่ตรงกันข้ามกับมอเตอร์สเตปเปอร์แบบธรรมดาที่มีรูปทรงทางเรขาคณิตใกล้เคียงกัน คือ ความยาวของแท่นจะแปรเปลี่ยนได้และขึ้นอยู่กับว่าต้องการจะเคลื่อนภาระที่ติดอยู่บนตัวให้แรงไปไกลแค่ไหน ลักษณะเช่นนี้เป็นข้อแตกต่างอีกประการหนึ่งจากมอเตอร์สเตปเปอร์ธรรมดาที่เป็นการขับเคลื่อนตัวภาระโดยตรง (direct driven payload) ไม่มีการใช้ระบบการเชื่อมต่อทางกลใดๆ มาช่วยในการส่งถ่ายกำลัง



รูปที่ 2.6 ส่วนประกอบของสเตปเปอร์มอเตอร์เชิงเส้นชนิด 2 เฟส

ดังที่ได้เห็นจากรูปที่ 2.6 ตัวให้แรงประกอบด้วยแม่เหล็กถาวร (permanent magnet : PM :) หนึ่งชุดและแม่เหล็กไฟฟ้า (electromagnet : EM :) สองชุดสี่ขั้ว (สองขั้วต่อหนึ่งชุด) พื้นผิวหน้าของขั้วได้รับการเซาะเป็นร่องเพื่อทำให้เกิดระยะ pitch ของมอเตอร์และมีการเซาะเป็นร่องขนาดเดียวกันที่สเตเตอร์เช่นเดียวกันซึ่งจะได้เห็นต่อไปว่า การทำเป็นร่องลักษณะนี้ทำให้เกิดสเตปความละเอียดในการเคลื่อนที่ อีกทั้งเมื่อช่องว่างระหว่างร่องถูกบรรจุด้วย วัสดุที่ไม่เป็นแม่เหล็กทำให้เกิดพื้นผิวราบเรียบที่สามารถใช้สร้างเป็นเบาะอากาศ (air bearing) รองรับระหว่าง ด้านล่าง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

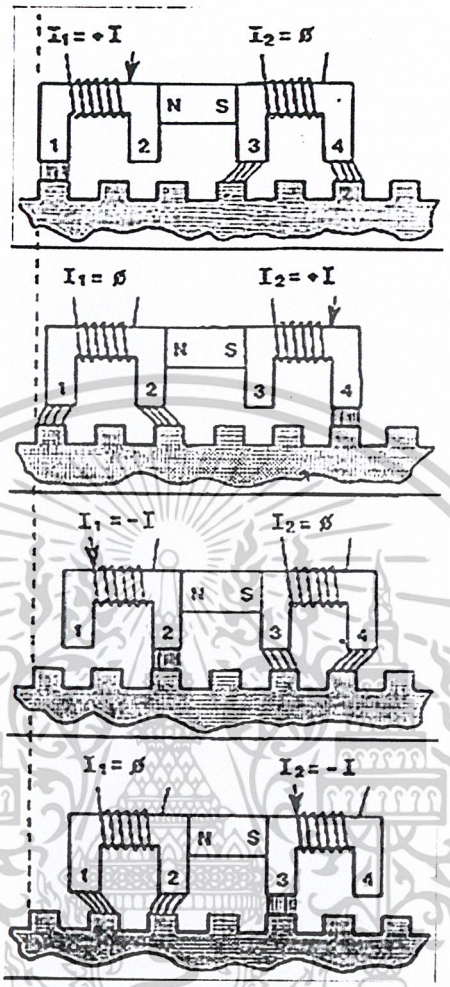
ของตัวให้แรงและด้านบนของแท่นซึ่งทำได้โดย พ่นลมความดันสูงออกจากช่องเล็กๆ ที่อยู่บนตัวให้แรงเบาะอากาศซึ่งเกิดระหว่างพื้นผิวทั้งสอง ทำให้ไม่มีทั้งค่าความฝืดเริ่มต้น (starting friction) และความฝืดเคลื่อนที่ (running friction)

แม่เหล็กถาวรทำให้แท่นและตัวให้แรงเคลื่อนเข้าหากัน (อยู่ห่างกันเท่ากับความหนาของตัวรองรับ) เพราะฉะนั้นจึงเป็นได้ที่จะกำหนดตำแหน่งแท่นเพื่อให้ตัวให้แรงสามารถเคลื่อนที่อยู่เหนือหรือใต้แท่นได้ เมื่อไม่มีกระแสไหลเส้นแรงแม่เหล็กของแม่เหล็กถาวร PM จะครบวงจรโดยผ่านช่องว่างอากาศ (air gap) ระหว่างแท่นและขั้วแม่เหล็กไฟฟ้า EM เส้นแรงแม่เหล็กจะแยกออกเป็นจำนวนเท่าๆกันที่ขั้วแม่เหล็กไฟฟ้า EM ทั้งสองขั้วเนื่องจากวงจรแม่เหล็กมีความต้านทานแม่เหล็ก (reluctance) เท่ากัน (ดูรูปที่ 2.7.ก pole 3 & 4) เมื่อสวิตช์กระแสผ่านแม่เหล็กไฟฟ้าจะเกิดการสลับที่ขึ้น โดยทั่วไปแล้วเส้นแรงแม่เหล็กที่เกิดขึ้นในแม่เหล็กถาวรจะมีขนาดเท่ากับที่เกิดขั้นในแม่เหล็กไฟฟ้าดังนั้นเมื่อกระแสที่ไหลเกิดการเปลี่ยนแปลง เส้นแรงแม่เหล็กจะเกิดการเปลี่ยนแปลงจากค่าสูงสุดและลดลงจนเกือบเป็นศูนย์

การสลับที่ร่วมกับตำแหน่งที่สัมพันธ์กันของตัวให้แรงและตัวแท่นก่อให้เกิดแรงในแนวตั้งฉากกับซี่ขั้วและแนวขนานกับตัวแท่นเนื่องจากซี่ขั้วของแม่เหล็กไฟฟ้าได้รับการจัดให้มีช่องว่างเหลือห่างประมาณเศษหนึ่งส่วนสี่ของซี่ขั้วถัดไป เส้นแรงแม่เหล็กจากแม่เหล็กถาวรจึงสามารถสลับที่กับเส้นแรงแม่เหล็กจากแม่เหล็กไฟฟ้าและพุ่งไปยังพื้นผิวซี่ขั้วตัวแท่นที่อยู่เหลื่อมออกไปได้ผลที่เกิดขึ้นคือจะเกิดแรงเคลื่อนในแนวนอนทำให้ตัวให้แรงสามารถเคลื่อนไปตามแนวซี่ขั้วของตัวแท่นได้โดยมีระยะห่างของช่องว่างอากาศน้อยที่สุด (เป็นการลดค่าความต้านทานแม่เหล็ก) แรงที่เกิดขึ้นนี้ทำให้เกิดการเคลื่อนที่ตามแนวความยาวของตัวแท่น นอกจากนี้ยังมีแรงในแนวตั้งฉากเกิดขึ้นด้วยซึ่งช่วยดึงให้ตัวให้แรงเคลื่อนไปตามซี่ขั้วของตัวแท่นซึ่งเท่ากับเป็นการช่วยการทำงานของเบาะอากาศ

รูปที่ 2.7.ก-ง เป็นภาพที่แสดงหลักการทำงานของมอเตอร์สเตปเปอร์เชิงเส้น ในแต่ละภาพทิศทางของกระแสและเส้นแรงแม่เหล็กแสดงด้วยลูกศร ถ้าป้อนกำลังให้กับแม่เหล็กไฟฟ้าชุด A ความเข้มเส้นแรงแม่เหล็กจะเกิดขึ้นที่ pole 2 และแนวเส้นแรงแม่เหล็กเป็นไป ดังแสดงในรูปที่ 2.7.ก เมื่อหยุดป้อนกำลังให้แม่เหล็กไฟฟ้าชุด A แล้วไปป้อนให้กับแม่เหล็กไฟฟ้าชุด B รูปที่ 2.7.ก-ง เป็นภาพที่แสดงหลักการทำงานของมอเตอร์สเตปเปอร์เชิงเส้น ในแต่ละภาพทิศทางของกระแสและเส้นแรงแม่เหล็กแสดงด้วยลูกศร ถ้าป้อนกำลังให้กับแม่เหล็กไฟฟ้าชุด A ความเข้มเส้นแรงแม่เหล็กจะเกิดขึ้นที่ pole 2 และแนวเส้นแรงแม่เหล็กเป็นไป ดังแสดงในรูปที่ 2.7.ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 การหมุนของสเตปเปอร์มอเตอร์เชิงเส้นชนิด 2 เฟส

เมื่อหยุดป้อนกำลังให้แม่เหล็กไฟฟ้าชุด A แล้วไปป้อนให้กับแม่เหล็กไฟฟ้าชุด B จะเกิดความหนาแน่นเส้นแรงแม่เหล็กมากที่สุดที่ pole 3 และน้อยที่สุดที่ pole 4 แรงดูดที่ pole 3 ทำให้เกิดการเคลื่อนที่ไปทางด้านขวาของตัวแท่นเท่ากับเศษหนึ่งส่วนสี่ของระยะ pitch เป็นความสัมพันธ์ของการเคลื่อนที่ดังแสดงในรูปที่ 2.7.ข ถ้าหยุดป้อนกำลังให้กับแม่เหล็กไฟฟ้าชุด B แล้วป้อนกำลังให้กับแม่เหล็กไฟฟ้าชุด A (ด้วยกระแสที่มีทิศทางตรงข้ามกับกระแสในรูปที่ 2.7.ก) อีกครั้ง จะเกิดการเคลื่อนที่ไปทางขวาเนื่องจาก pole 1 จะมีค่าความหนาแน่นเส้นแรงแม่เหล็กสูงสุด ส่วน pole 2 จะต่ำสุด ในขณะที่ pole 3 และ 4 จะเป็นเส้นแรงแม่เหล็กที่ได้มาจากชุดแม่เหล็กถาวรตัวให้แรงจะอยู่ที่ตำแหน่ง ดังแสดงในรูปที่ 2.7.ค และท้ายสุดเมื่อหยุดป้อนกำลังให้กับแม่เหล็กไฟฟ้าชุด A แล้วป้อนกำลังให้กับแม่เหล็กไฟฟ้าชุด B (ในทิศทางตรงข้ามกับครั้งที่แล้ว pole 4 จะมีความหนาแน่นเส้นแรงแม่เหล็กสูงสุด pole 3 จะน้อยที่สุด ในขณะที่ pole 1 และ 2 เป็นเส้นแรงแม่เหล็กที่ได้มาจาก ชุดแม่เหล็กถาวร เพื่อให้ครบรอบจังหวะการทำงานอีกครั้งจะเริ่ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ป้อนกำลังให้กับแม่เหล็กไฟฟ้าชุด A ดังรูปที่ 2.7.ก ซึ่งจะทำให้ระบบเคลื่อนที่ไปเป็นระยะทางเต็มหนึ่งซี่ขั้ว (หรือ pitch) ของตัวแท่น (เทียบเท่ากับการหมุนในรูปแบบ full step ในมอเตอร์สเตปเปอร์เชิงหมุน) ความถี่ของกระแสในการทำให้ครบรอบการสวิตช์เป็นตัวกำหนดความเร็วในการเคลื่อนที่ของตัวให้แรง

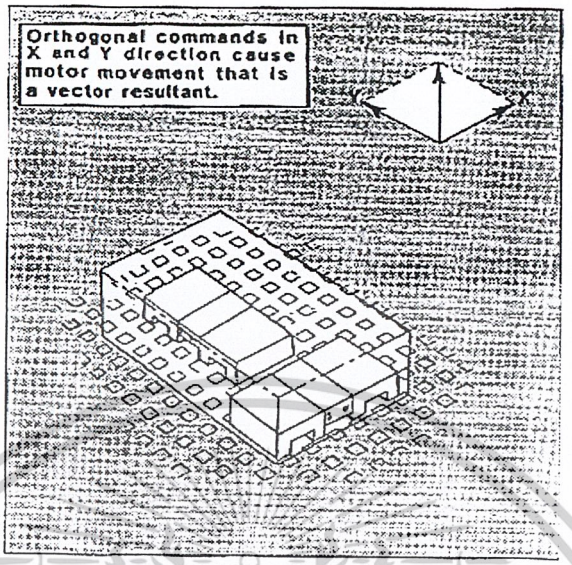
ตามความเป็นจริงโดยธรรมชาติตำแหน่งที่สัมพันธ์กันของตัวให้แรงกับตัวแท่นเป็นตำแหน่งที่ไม่มีความต่อเนื่อง (discrete) แต่ถ้าป้อนกระแสในลักษณะที่มีความถี่ครบรอบการสวิตช์ดังเช่นที่กล่าวมาแล้วมอเตอร์สเตปเปอร์เชิงเส้นจะหมุนแบบ full-step โดยมีความละเอียดซึ่งใช้ระยะห่างซี่ขั้วเป็นตัวกำหนดระยะ pitch ทั่วๆไป คือ 0.040 นิ้ว ดังนั้นการหมุนตามลำดับขั้นที่แสดงในรูปที่ 2.7 จึงทำให้มีความละเอียดในการเคลื่อนที่เท่ากับเศษ-หนึ่งส่วนสี่ pitch หรือ 0.010 นิ้ว ตำแหน่งลักษณะนี้บางครั้งเรียกว่า *cardinal step* ถ้าต้องการความละเอียดมากกว่านี้ ก็สามารถกระทำได้ โดยป้อนกระแสในขนาดที่น้อยกว่ากระแสที่ใช้ในการเคลื่อนที่แบบ full-step ซึ่งทำให้มอเตอร์หมุนแบบ micro-step

เป็นไปได้ที่จะสร้างมอเตอร์ชนิดนี้ให้มีส่วนประกอบที่ทำให้เกิดแรงสองแรงซึ่งตั้งฉากกันในโครงมอเตอร์ตัวเดียวซึ่งจะทำให้โครงสร้างทั้งของตัวให้แรงและตัวแท่นมีซี่ขั้วเป็นรูปสี่เหลี่ยมจัตุรัสในลักษณะเป็นชั้นๆ (waffle pattern) รูปที่ 2.8 แสดงโครงสร้างของมอเตอร์สเตปเปอร์แบบสองแกน (two-axis) รูปแบบลักษณะนี้ทำให้เกิดการเคลื่อนที่ได้ทั้งในแนวทิศทาง x และ y หรือตามแนวในระนาบ x-y (x-y plane)

ดังได้กล่าวมาแล้วว่ามอเตอร์สเตปเปอร์เชิงเส้นเป็นมอเตอร์ชนิดขับเคลื่อนโดยตรง โดยนัย จึงหมายความว่า ความละเอียดในการควบคุมตำแหน่งและแรงที่ต้องใช้ในการเคลื่อนตัวภาพไปยังตำแหน่งที่ต้องการจะถูกกำหนดจากความสามารถของตัวมอเตอร์เองโดยตรง ดังนั้นการประยุกต์ใช้งานใดๆ ที่ต้องการความละเอียดในการเคลื่อนที่มากกว่าระยะห่างของซี่ขั้วที่เป็นอยู่ตัวควบคุมจะต้องสามารถทำให้เกิดการเคลื่อนที่แบบ micro-step ไปยังตำแหน่งที่ต้องการได้ นอกจากนี้แล้วเส้นโค้งแสดงลักษณะสมบัติ ความเร็ว-แรงบิด (speed-torque curve) ในการขับเคลื่อนมอเตอร์ควรต้องได้รับตรวจสอบอย่างละเอียดถี่ถ้วนเพื่อให้แน่ใจว่ามอเตอร์มีความสามารถที่จะขับเคลื่อนตัวภาระไปได้ด้วยความเร็วตามที่ต้องการอย่างแท้จริง

มอเตอร์สเตปเปอร์เชิงเส้นจะแสดงคุณสมบัติการเกิด resonance และอาจสูญเสียการพ้องจังหวะ (synchronization) หรือสูญเสีย step ได้เช่นเดียวกับมอเตอร์สเตปเปอร์เชิงหมุน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

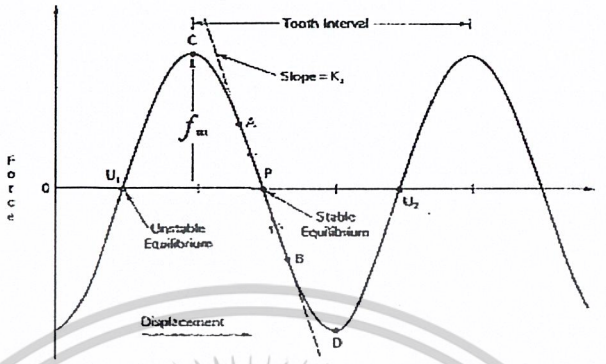


รูปที่ 2.8 โครงสร้างของสเตปเปอร์มอเตอร์แบบ 2 แกน

อย่างไรก็ตามยังมีลักษณะสมบัติอีกสองข้อที่ก่อให้เกิดความซับซ้อนในการควบคุมมอเตอร์ชนิดนี้ ประการแรก อุปกรณ์ชนิดนี้สามารถแสดงลักษณะสมบัติการเป็น "สปริงได้" (springy) อาร์เมเจอร์ที่หยุดอยู่ระหว่างซี่ขั้วเป็นการหยุดอยู่ในแรงหน่วงของซี่ขั้วทั้งสอง มอเตอร์จะมีแรงเกิดขึ้นในตัวมันเองหลายแรงทั้งแรงเฉื่อยและแรงดึงดูด (inertial & drag force) ลักษณะเช่นนี้อาจทำให้มอเตอร์เคลื่อนออกจากตำแหน่งซี่ขั้วได้ จึงเป็นความจำเป็นอย่างยิ่งที่จะต้องรักษาระดับของแรงให้คงที่อยู่ตลอดเวลา ถ้าแรงดึงดูดมีค่ามากกว่าค่าของแรงหน่วงมอเตอร์จะเกิดการลื่นไหลและสูญเสียการพ้องจังหวะไปได้ ลักษณะสมบัติข้อที่สองที่ทำให้การควบคุมทำได้ยากคือปรากฏการณ์รีโซแนนซ์ของอาร์เมเจอร์ซึ่งเป็นผลมาจากการใช้ตัวรองรับ (bearing) แบบพื้นผิวไม่มีความฝืด (frictionless) คือ เบาะอากาศ จึงทำให้มีความหน่วงเกิดขึ้นเพียงเล็กน้อยเท่านั้น ในระบบสปริง-มวล (spring-mass system) ของอาร์เมเจอร์และระบบแรงยืดเหนี่ยวลักษณะเช่นนี้ทำให้มอเตอร์มีความไวต่อการกระตุ้นในช่วงใกล้กับความถี่รีโซแนนซ์ธรรมชาติ (natural resonance frequency) ซึ่งหมายถึงว่า จะต้องตั้งจุดทำงานที่ควรจะอยู่ห่างจากจุดที่ทำให้เกิดปรากฏการณ์การรบกวนลักษณะนี้ให้มากที่สุดเท่าที่จะทำได้

รูปที่ 2.9 แสดงส่วนโค้งของแรงเทียบกับระยะทางที่เคลื่อนไปซึ่งจะใช้อธิบายในเชิงคุณภาพของลักษณะสมบัติเชิงกลศาสตร์ของมอเตอร์สเตปเปอร์เชิงเส้น ในภาพนี้สมมติให้มอเตอร์ได้รับคำสั่งให้เคลื่อนที่ไปยัง ตำแหน่ง P ซึ่งเป็นตำแหน่งสมมูลที่มีเสถียรภาพ (stable equilibrium point) ดังนั้นในสภาวะคงที่ (steady state) จะไม่มีแรงใดๆกระทำบนอาร์เมเจอร์เมื่ออยู่ที่ตำแหน่งนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 ส่วนโค้งของแรงเทียบกับระยะทางที่เคลื่อนไป

ถ้าในขณะนี้อาร์เมเจอร์เคลื่อนไปที่ตำแหน่ง A หรือ B จะเกิดแรงกระทำกลับ (restoring force) ที่ทำให้มีอัตราเร่งเคลื่อนกลับมายังจุด P ยิ่งเคลื่อนห่างออกจากจุด P เท่าใด ยิ่งเกิดแรงกระทำกลับมากขึ้นเท่านั้น จนกระทั่งเมื่อมาถึงตำแหน่ง C หรือ D ที่จุดๆนี้ คือ ตำแหน่ง ¼ ของซี่ขั้ว (tooth pitch) ถ้ายิ่งเคลื่อนออกจาก P ไปใกล้ U_1 หรือ U_2 มากเท่าใด แรงกระทำกลับจะอยู่ในทิศทางที่ถูกต้องมากขึ้น (คือจะเคลื่อนเข้าหาจุดสมดุล) แต่ขนาดจะลดลง

จุด U_1 และ U_2 เป็นจุดสมดุลที่ไม่มีเสถียรภาพ (unstable equilibrium) ดังจะเห็นได้จากรูปที่ 2.9 ไม่มีแรงใดๆกระทำเมื่ออาร์เมเจอร์อยู่ที่ตำแหน่งนี้ อย่างไรก็ตาม ถ้ามอเตอร์เคลื่อนออกจากตำแหน่งนี้เพียงเล็กน้อยจะทำให้เกิดอัตราเร่งให้เคลื่อนห่างออกไป (จนกระทั่งไปถึงตำแหน่งสมดุลตำแหน่งใหม่) ดังนั้นตราบที่อาร์เมเจอร์มิได้อยู่นอกตำแหน่ง(เปิด) ในช่วงระหว่าง U_1 และ U_2 มันจะถูกจับให้ไปที่จุด P ลักษณะเช่นนี้จะเกิดขึ้นเช่นเดียวกันกับอุปกรณ์เชิงหมุน มอเตอร์สเตปเปอร์เชิงเส้นจะแสดงคุณสมบัติเป็นแบบความหน่วงต่ำกว่า (underdamped) ซึ่งทำให้อาร์เมเจอร์เกิดการแกว่งไกว (oscillate) ที่จุด P ก่อนที่จะหยุดนิ่งในที่สุด

เป็นที่แน่ชัดจากที่ได้กล่าวมาจากข้างบนนี้ว่า รูปแบบของสนามแม่เหล็กจะส่งแรงกระทำไปที่อาร์เมเจอร์ก็ต่อเมื่อ *มีการเคลื่อนออกจากตำแหน่งสมดุลสั่งการ (command equilibrium position) เท่านั้น* ดังนั้น การต้องใช้แรงเพื่อชนะ ความเฉื่อย (inertia) และความฝืด (friction) ในความเป็นจริงมอเตอร์จึงจำเป็นจะต้องทำงาน ล้ำหน้า (leading) หรือล้าหลัง (lagging) ตำแหน่งคำสั่งการ เมื่อตรวจสอบกับ รูปที่ 2.9 จะพบว่าฟังก์ชันการขับเคลื่อน (driving function) จะต้องมีความผิดพลาดระหว่างค่าที่ต้องการ (desire position) และค่าจริง (actual

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

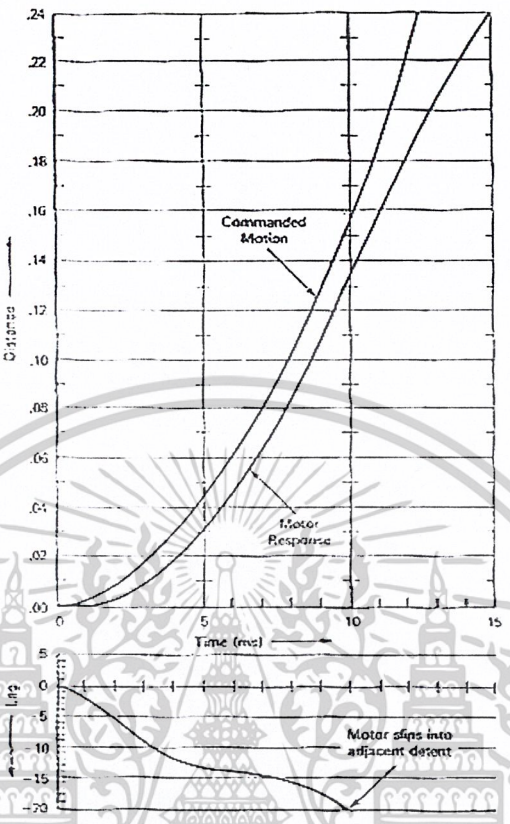
สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

position) ของตำแหน่งอาร์เมเจอร์ไม่เกินกว่าระยะ ¼ ของซี่ขั้ว ถ้ามีความแตกต่างมากกว่านี้ อาจสูญเสียการพ้องจังหวะ (lost of synchronization) ได้

เนื่องด้วยมอเตอร์สเตปเปอร์เชิงเส้นโดยปกติจะทำการควบคุมในแบบวงรอบ-เปิด (open-loop) จึงเป็นสิ่งสำคัญอย่างยิ่งที่ฟังก์ชันการขับเคลื่อนหรือฟังก์ชันบังคับการทำงาน (driving or forcing function) จะต้องไม่กระตุ้นให้มอเตอร์ทำงานที่ความถี่รีโซแนนซ์ ตัวอย่างเช่น อัตราการเคลื่อนที่แบบไมโครสเตป จะต้องได้รับการเลือกอย่างระมัดระวังที่จุดต้องไม่อยู่ที่หรืออยู่ใกล้กับความถี่ธรรมชาติของตัวมอเตอร์สเตปเปอร์เอง นอกจากนี้การสั่น (oscillation) อาจถูกเหนี่ยวนำให้เกิดขึ้นได้ ถึงแม้จะดูเหมือนว่าฟังก์ชันบังคับการทำงานจะเป็นฟังก์ชันที่ราบเรียบก็ตามที่เป็นเช่นนี้เพราะว่าสัญญาณมีพลังงานพอเพียงที่จะไปกระตุ้นจุดความถี่รีโซแนนซ์ให้เกิด “การสั่นระรัว” (ringing) ขึ้นมาได้ รูปที่ 2.10 แสดงตำแหน่งคำสั่งการและตำแหน่งจริงของอาร์เมเจอร์เมื่อเทียบกับเวลาสำหรับรูปแบบความเร็วชนิดสามเหลี่ยม (triangular velocity profile) ในภาพยังแสดงความแตกต่างระหว่างค่าทั้งสองค่าไว้ด้วยคงจะระลึกถึงได้ว่า ค่าความแตกต่างนี้ คือค่าของแรงที่กระทำโดยสนามแม่เหล็กบนตัวอาร์เมเจอร์จะสังเกตเห็นด้วยว่าเมื่อเกิดการล้าหลังถึงจุด -20 mils อาร์เมเจอร์จะอยู่นอกช่วงตำแหน่งเปิด (U_1 , U_2) ทำให้เกิดสูญเสียการพ้องจังหวะ เป็นที่แน่ชัดว่า ถ้าเลือกมอเตอร์, ตัวภาระ (load), และอัตราเร่งได้ถูกต้องก็เป็นไปได้ที่จะใช้รูปแบบความเร็วชนิดสามเหลี่ยมได้โดยไม่มีการตกหล่นของสเตป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

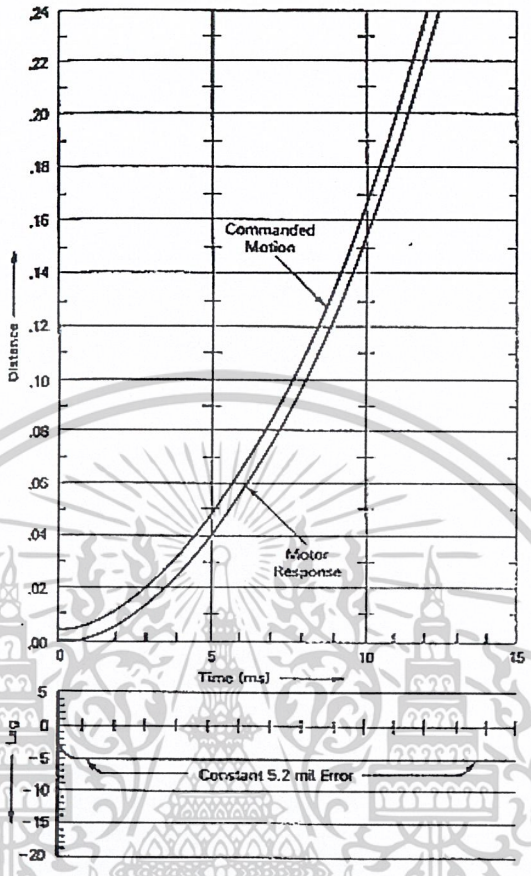
51880



รูปที่ 2.10 ตำแหน่งคำสั่งการและตำแหน่งจริงของอาร์เมเจอร์
เมื่อเทียบกับเวลาสำหรับรูปแบบความเร็วชนิดสามเหลี่ยม

เทคนิคซึ่งพัฒนาขึ้นมาใหม่เมื่อไม่นานมานี้เพื่อป้องกันการสั่นระรัว (ringing) ของมอเตอร์สเตปเปอร์เชิงเส้น คือเทคนิคที่เรียกว่า burst เป็นฟังก์ชันบังคับการทำงานที่กำหนดพื้นฐานการทำงานขนความแตกต่างระหว่างคำสั่งการ (command) และตำแหน่งอาร์เมเจอร์จริง (error) เป็นค่าคงที่ (lag) ขณะการเคลื่อนที่ของเฟสอัตราเร่ง รูปที่ 2.11 แสดงมอเตอร์สเตปเปอร์เชิงเส้นที่ขับเคลื่อนด้วยวิธีนี้จะสังเกตเห็นได้ว่ามีความแตกต่างของสเตปอยู่ 5.2 mils ลักษณะเช่นนี้คือ “burst” ด้วยฟังก์ชันบังคับการทำงานแบบนี้ มอเตอร์จะไม่เกิดการสั่นระรัว จึงไม่เกิดอันตรายจากการตกหล่นของสเตป ตามปกติการทำให้เกิดเทคนิคนี้จะต้องป้อน burst ที่จุดตัด (เช่น จุดเริ่มต้นและจุดสิ้นสุด การเคลื่อนที่ของเฟสอัตราเร่งและอัตราหน่วง รายละเอียดเกี่ยวกับเรื่องนี้หาเพิ่มเติมได้จากเอกสารของ Nordis,J.I., and Smit , P.M., “A Motion Control System for Linear(Stepper) Motors,” Proceedings of the Incremental Motion Society , University of Illinois , Urbana , III. , May 1985.)

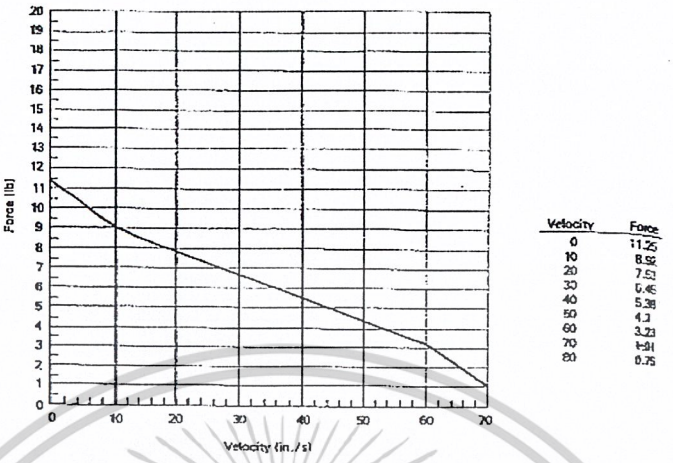
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 มอเตอร์สเตปเปอร์เชิงเส้นที่ขับเคลื่อนด้วยเทคนิค burst

ส่วนโค้งลักษณะสมบัติ ความเร็ว-แรง (speed-fource) ของมอเตอร์สเตปเปอร์เชิงเส้นทั่วไปแสดงอยู่ใน รูปที่ 2.12 จากส่วนโค้งจะเห็นได้ว่าในขณะที่ความเร็วของมอเตอร์ชนิดนี้สูงขึ้น แรงที่เกิดขึ้นจะลดลงอย่างสัมพันธ์กันเหมือนกับทอร์คของมอเตอร์เชิงหมุนลดลงเมื่อความเร็วเชิงมุมเพิ่มขึ้นการนำมาประยุกต์ใช้งานให้เกิดประโยชน์จึงควรพิจารณาจุดที่ต้องการแรงที่ความเร็วสูงสุดเท่าใด ตัวอย่างเช่น เนื่องจากแรงที่ความเร็ว 50 in./s. คือ 4.3 lb. ดังนั้นเพื่อความแน่นอนในการทำงานไม่ควรใช้แรงเกินกว่า 4.3 lb. ในช่วงต้องการอัตราเร่ง หรือ อัตราหน่วง ซึ่งเห็นได้ว่าการกระทำลักษณะนี้จะทำให้สามารถใช้ประโยชน์จากมอเตอร์ได้เต็มตามความสามารถของมัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.12 ลักษณะสมบัติ ความเร็ว-แรง ของมอเตอร์สเตปเปอร์เชิงเส้นทั่วไป

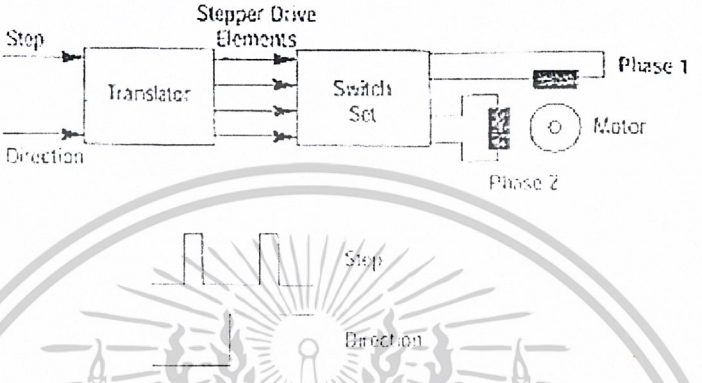
การกระทำอีกลักษณะหนึ่งที่ทำให้ประสิทธิภาพการใช้มอเตอร์เพิ่มขึ้นคือการใช้รูปแบบความเร็ว (velocity profile) ในแบบเส้นตรงเป็นช่วงๆ (piecewise linear) ที่มีหลายความลาดชัน (multiple slopes) ในช่วงที่มีอัตราเร่งหรืออัตราหน่วงเทคนิคนี้จะช่วยลดช่วงเวลาการเคลื่อนที่ทั้งหมดลงได้บ้าง เป็นต้นว่า ถ้าช่วงที่มีอัตราเร่งถูกแบ่งออกเป็นสามช่วง คือ 0 ถึง 20 in./s. , 20 ถึง 40 in./s. และ 40 ถึง 50 in./s. ทำให้ได้แรงสูงสุด 7.53, 5.38 และ 4.3 lb. สำหรับภาระ (load) ขนาด 16 lb. รูปแบบความเร็ว จะเห็นได้ว่าการกระทำลักษณะนี้ทำให้ประสิทธิภาพการทำงานเพิ่มขึ้น ทำให้สามารถลดเวลาอัตราเร่งลงได้ถึง 25 % (เหลือเพียง 0.36 s. แทนที่จะเป็น 0.481 s.)

ความพยายามที่จะทำให้แรงของมอเตอร์ดีกว่าเดิมเป็นวิธีการที่เรียกว่า acceleration roll-off วิธีนี้อาจทำได้โดยใช้รูปแบบที่กล่าวมาแล้วข้างบนนี้ หรืออาจใช้รูปคลื่นอัตราเร่งแบบ exponential ทั้งสองกรณีแนวคิดพื้นฐานคือ การปรับแต่งอัตราเร่งหรืออัตราหน่วงในขณะมีการเคลื่อนที่โดยใช้ประโยชน์จากส่วนโค้งแรง-ความเร็วของมอเตอร์ทำให้ผลรวมของแรงเฉื่อยและแรงฉุดเป็นค่าคงที่ตลอดเวลาสิ่งสำคัญที่จะต้องเข้าใจ คือ การใช้เทคนิคนี้ มิได้ป้องกัน การสั่นระรัว (ringing) ที่เกิดขึ้นในตัวมอเตอร์ดังได้กล่าวมาแล้วจะต้องกระทำอย่างระมัดระวังในการเลือกฟังก์ชัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.8 ชุดขับ (Driver)

ส่วนที่สำคัญในการใช้ Stepping motors คือชุดขับ (Driver)



รูปที่ 2.13 การขับ Stepping motor

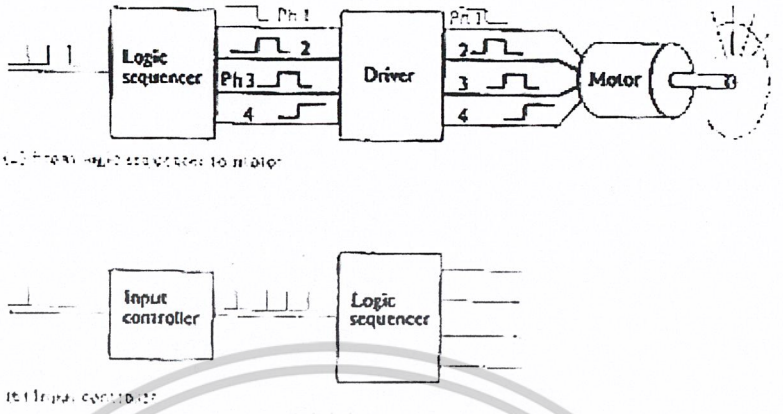
ชุดขับจะจ่ายพลังงานไฟฟ้าให้กับ Stepping motor ตามสัญญาณควบคุม (Controls Signal) ซึ่งจะมีระดับต่ำว่า Stepping motor จะสร้างทอร์ค, แรงขับหมุนและ magneto-motive force (MMF) ขึ้นจากสนามแม่เหล็กซึ่งขึ้นอยู่กับกระแสไฟฟ้าและจำนวนรอบของขดลวดที่ใช้ ดังนั้นจะเห็นว่าชุดขับนั้นมีความสำคัญและทำหน้าที่เป็นแหล่งจ่ายกระแสให้กับ Stepping motor

สัญญาณที่ควบคุมชุดขับนั้นประกอบด้วยพัลส์ (Step pulse) และสัญญาณควบคุมทิศทางการหมุนพัลส์ 1 ลูกจะทำให้ Stepping motor หมุนไป 1 step ดังนั้นถ้าต้องการให้ stepping motor หมุน 1 รอบ และ stepping motor มีขนาด 1.8 องศาต่อสเตป จะต้องใช้พัลส์ถึง 200 พัลส์และในงานด้านอุตสาหกรรมโดยทั่วไปจะกระตุ้น stepping motor แบบ Half step ก็จะต้องใช้ 400 พัลส์ต่อการหมุน 1 รอบ เมื่อ Stepping motor หมุนที่ความเร็ว 1,800 รอบต่อนาทีต้องใช้พัลส์ความถี่ 12 kHz และถ้าหมุน 25,000 สเตปต่อ 1 รอบ จะต้องใช้พัลส์ความถี่ 750 kHz ดังนั้นตัวควบคุมและชุดขับจะต้องตอบสนองที่ความถี่สูงๆ มากได้

การขับแบบลูปเปิด (Open loop)

ระบบการขับ Stepping motor จะมีลักษณะดังรูปที่ 2.14 ซึ่งประกอบด้วย input controller และ Logic sequencer ซึ่งอาจจะใช้วงจรทาง Digital logic หรือใช้ Microprocessor ก็ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



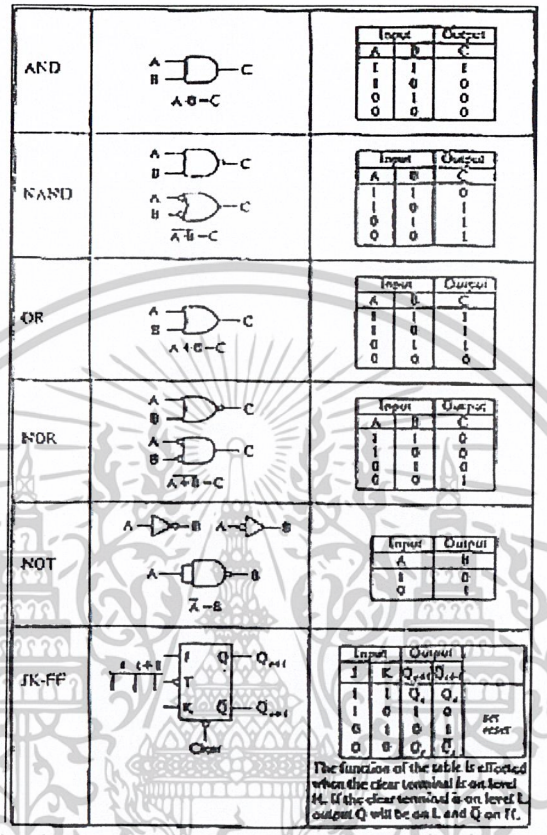
รูปที่ 2.14 Block diagram ของระบบการขับ Stepping motor

Logic sequencer

Logic sequencer เป็นวงจร Logic ที่สร้างสัญญาณเพื่อที่จะนำไปขับ Stepping motor ในรูปแบบการกระตุ้นต่างๆ ตามต้องการ เช่น แบบ Single step, Full step, หรือ Half step วงจร Logic ใช้ Logic gate, Flip-flop ต่างๆ ประกอบเข้าด้วยกัน ซึ่ง Logic gate และ Flip-flop ต่างๆ จะมี Function ดังตารางที่ 2.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2_แสดงฟังก์ชันของ Logic gate และ Flip-flop



การกระตุ้น Stepping motor 4 เฟส แบบ Full step

การขับแบบ Full step นั้นในเวลาเดียวกันจะมีขดลวด 2 เฟส ที่ได้รับการกระตุ้น ดังรูปที่ 2.15 (ก) เป็นการหมุนตามเข็มนาฬิกา และ (ข) เป็นหมุนทวนเข็มนาฬิกา โดยให้

Q1 ต่อกับ Phase 1 ของ Stepping motor

Q2 ต่อกับ Phase 2 ของ Stepping motor

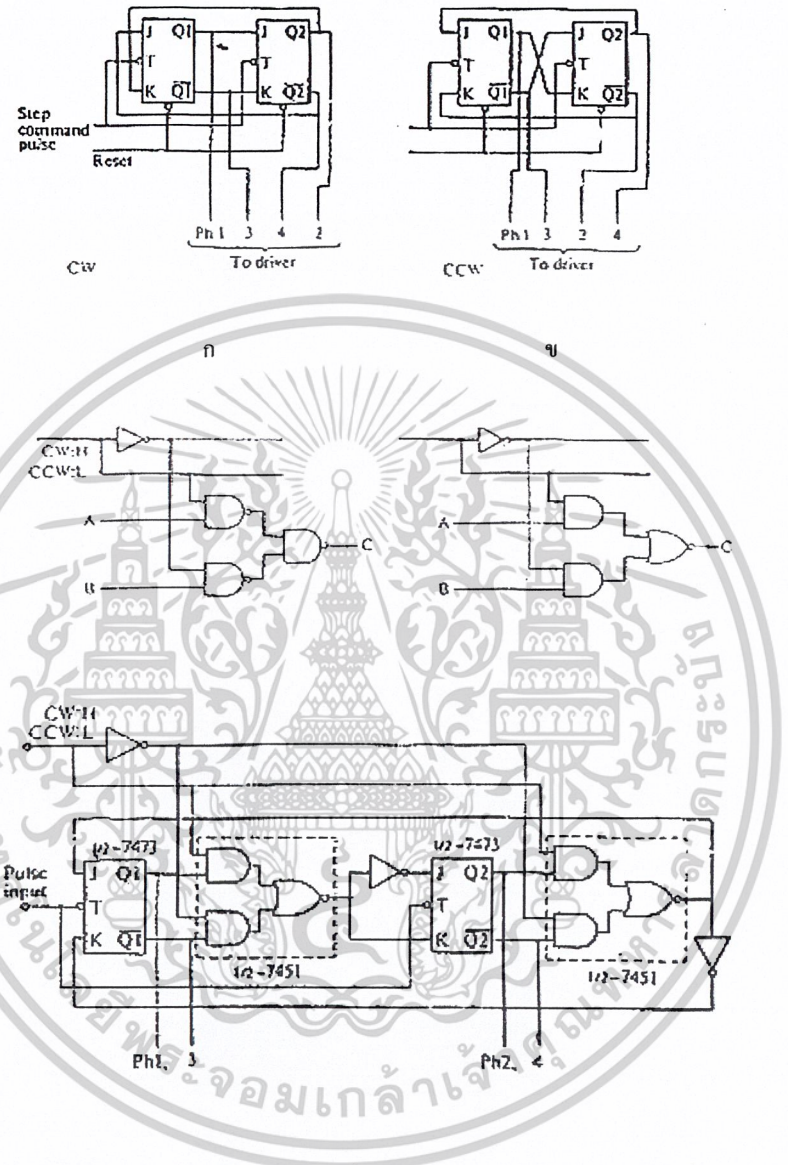
Q3 ต่อกับ Phase 3 ของ Stepping motor

Q4 ต่อกับ Phase 4 ของ Stepping motor

และระดับ Logic “1” คือกระตุ้น stepping motor และ Logic “0” ไม่กระตุ้น Stepping motor

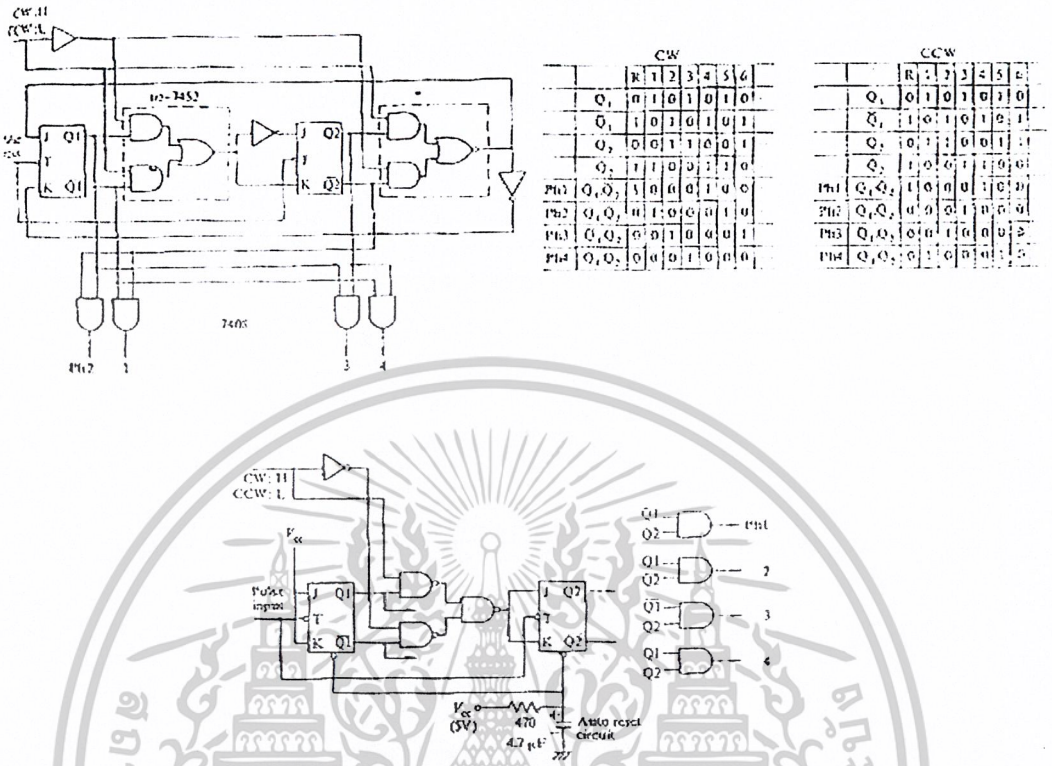
การกระตุ้น Stepping motor 4 เฟส แบบ single step การกระตุ้นแบบนี้ในเวลาหนึ่งจะมีขดลวด Stepping motor เฟสเดียวที่ได้รับการกระตุ้น ซึ่งทำได้โดยการนำเอา AND Gates ต่อที่ Output ของวงจรการกระตุ้น Stepping motor 4 เฟส แบบ Full step ดังแสดงในรูปที่ 2.16

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.15 unidirection logic สำหรับการกระตุ้นแบบ Full step ของมอเตอร์ 4 เฟส แบบต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 Logic sequencer สำหรับการกระตุ้น Stepping motor 4 เฟส แบบSingle Step

การกระตุ้น Stepping motor 3 เฟส แบบ Full step

วงจร Logic sequencer แบบนี้จะใช้ NAND Gates แทนการใช้ AND Gates

การกระตุ้น Stepping motor 3 เฟส แบบ Half step

วงจรกระตุ้นนี้ใช้ J-K Flip-flop ดังแสดงในรูปที่ 2.17

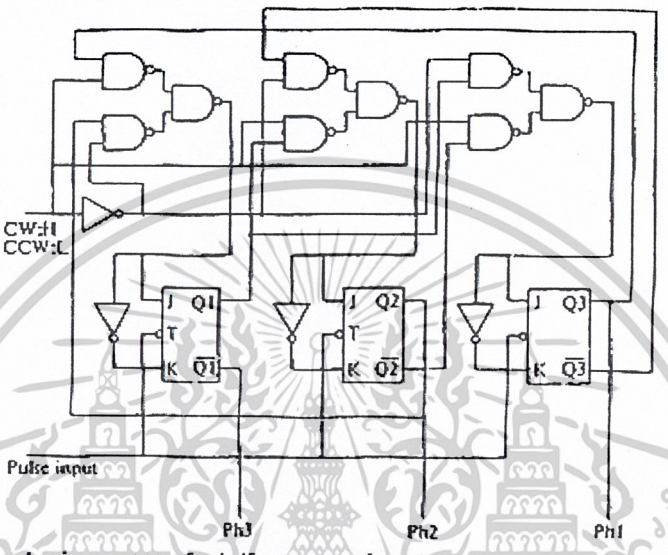
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CW

		R	1	2	3	4	5	6	7
Ph1.	$\overline{Q1}$	1	1	0	0	0	1	1	1
Ph2.	Q2	0	0	0	1	1	1	0	0
Ph3.	Q3	0	1	1	1	0	0	0	1

CCW

		R	1	2	3	4	5	6	7
Ph1.	$\overline{Q1}$	1	1	0	0	0	1	1	1
Ph2.	Q2	0	1	1	1	0	0	0	1
Ph3.	Q3	0	0	0	1	1	1	0	0



Logic sequencer for half-step operation of a three-phase motor.

รูปที่ 2.17 การกระตุ้น Stepping motor 3 เฟส แบบ Half step

2.9 การใช้ Chip สำเร็จรูป

ปัจจุบันเทคโนโลยีในการผลิต Chip ก้าวหน้าขึ้นมากมีการผลิต Chip ซึ่งใช้กระตุ้น stepping motor ได้เลย ดังตัวอย่างเช่น IC เบอร์ PMM8713 ของบริษัท SANYO และเบอร์ UCN-5804B เป็นต้น



รูปที่ 2.18 IC ควบคุม Stepping motor ของ Nanotec GmbH

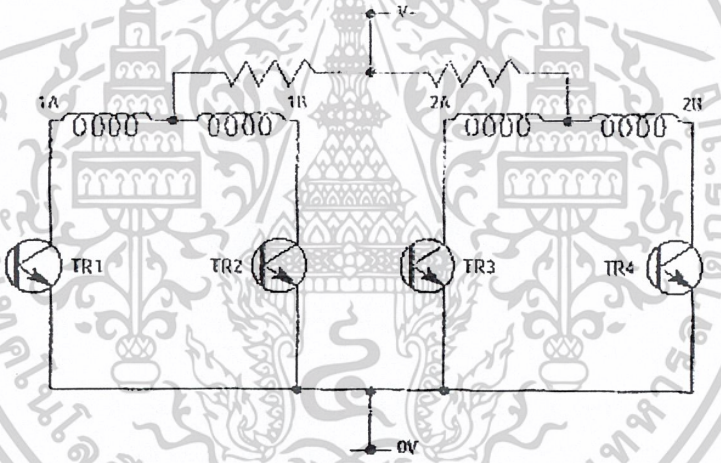
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.10 Driver

Driver หรือชุดขับ Stepping motor จะขึ้นอยู่กับรูปแบบการต่อขดลวดของ Stepping motor ซึ่งโดยทั่วไป Stepping motor จะมีลักษณะของขดลวด 2 แบบคือ Unipolar และ Bipolar ซึ่งแต่ละแบบก็ต้องการชุดขับที่แตกต่างกันออกไป

Basic Unipolar Driver

ชุดขับแบบนี้จะมีกระแสไฟฟ้าไหลผ่านขั้วต่อขดลวดของ Stepping motor ทางเดียว และขนาดของกระแสที่ไหลเข้าไปจะขึ้นอยู่กับ ค่าความต้านทานของขดลวด (Motor winding resistance) นั้นๆ และขนาดแรงดันแหล่งจ่ายที่จ่ายให้กับระบบ ดังรูปที่ 2.19



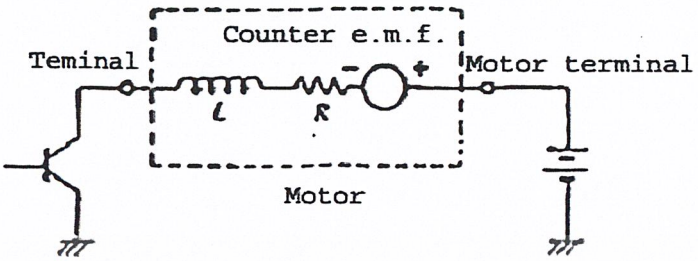
รูปที่ 2.19 วงจรขับแบบ Unipolar อย่างง่าย

การขับแบบนี้จะทำงานได้ดีที่อัตราการหมุนที่ช้าๆ แต่เมื่อให้อัตราหมุนเร็วมากจะทำให้ทอร์ค ตกลงอันเป็นผลมาจากค่าเหนี่ยวนำของขดลวดใน Stepping motor

2.11 วงจร Suppressors

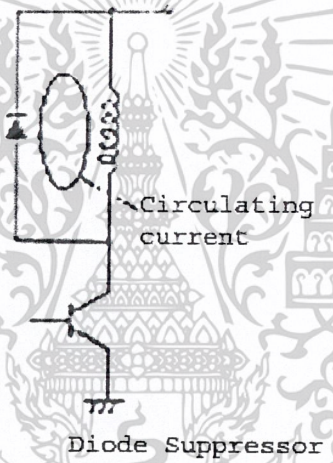
จากรูปที่ 2.20 จะเห็นว่าได้ใช้ทรานซิสเตอร์ซึ่งเป็นสารกึ่งตัวนำเป็นสวิตช์ เมื่อมีการกระตุ้น Stepping motor ก็จะเกิดแรงดันเหนี่ยวนำขึ้นในขดลวดของ Stepping motor อันเนื่องมาจากตัวเหนี่ยวนำในตัว stepping motor และยังเกิดแรงดันเนื่องจากการหมุนของสนามแม่เหล็กตัดกับขวดอีกด้วยซึ่งจะมีผลทำให้กระแสไหลน้อยลงจะส่งผลให้ทอร์คน้อยลงด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



Equivalent circuit to a winding of a stepping motor.

รูปที่ 2.20 วงจรสมมูลของขดลวด stepping motor



รูปที่ 2.21 วงจร Diode Suppressors

จากรูปที่ 2.20 จะเห็นว่าเมื่อกระแสในขดลวดลดลงเป็นศูนย์จะทำให้เกิดแรงดันเหนี่ยวนำขึ้นในขดลวดและถ้าแรงดันนี้สูงกว่าที่ทรานซิสเตอร์จะทนได้ก็จะทำให้เสียหายได้

ดังนั้นเพื่อลดผลดังกล่าวจึงต้องใส่วงจร Suppressors ซึ่งมีหลายลักษณะดังต่อไปนี้

วงจร Diode Suppressors

วงจรนี้ใช้ไดโอดต่อกับอุปกรณ์สวิตช์กำลังดังรูปที่ 2.21 โดยจะทำหน้าที่คายพลังงานที่เก็บสะสมในตัวเหนี่ยวนำออกไปซึ่งวงจรนี้จะใช้เวลาครู่หนึ่งซึ่งจะทำให้เกิด Breaking torque ขึ้น ทำให้ stepping motor หมุนช้าลง โดยมีแรงดันตกคร่อมอุปกรณ์กำลัง (ทรานซิสเตอร์) ดังนี้

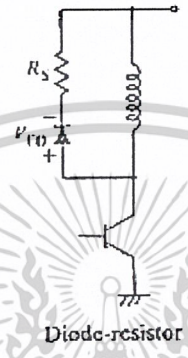
$$VCE = E + VDF \qquad \text{สมการ 2.3}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจร Diode/Resistor Suppressors

วงจรนี้ใช้ตัวต้านทาน (Resistor) ต่ออนุกรมกับไดโอดดังรูปที่ 2.22 โดยจะทำให้คายพลังงานได้เร็วขึ้นกว่าวงจรแรก โดยมีแรงดันตกคร่อมอุปกรณ์กำลัง(ทรานซิสเตอร์) ดังนี้

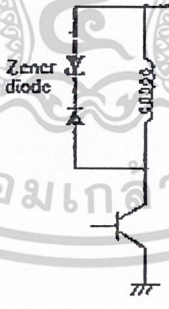
$$VCE = E + I \times RS + VDF \qquad \text{สมการ 2.4}$$



รูปที่ 2.22 วงจร Diode/Resistor Suppressors

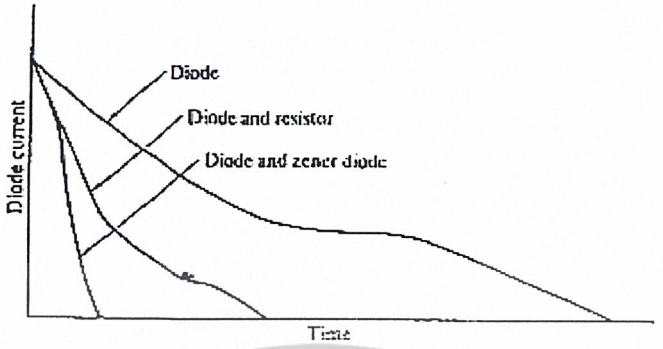
วงจร Zener diode Suppressors

วิธีนี้จะใช้ Zener Diode ต่ออนุกรมกับ Diode ดังรูป 2.23 ทำให้ใช้เวลาคายพลังงานได้เร็วกว่าวงจรแรกๆ ดังแสดงในรูปที่ 2.24



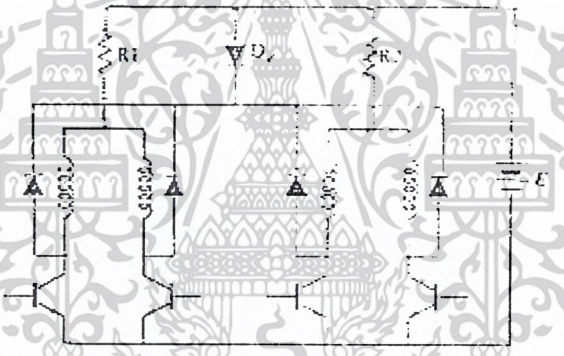
รูปที่ 2.23 วงจร Zener diode Suppressors

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.24 เปรียบเทียบผลของวงจร Suppressors ต่างๆ

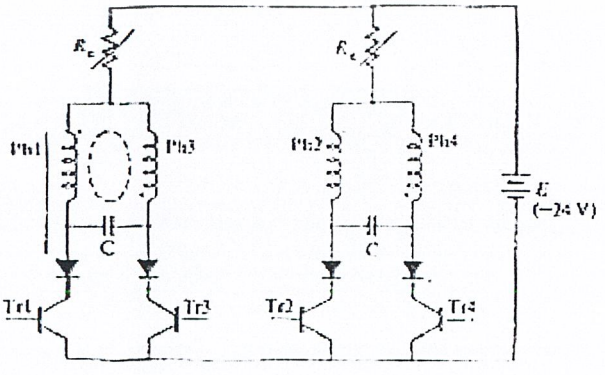
ตัวอย่างการใช้งาน วงจร Zener diode Suppressors กับการขับมอเตอร์แบบ 4 เฟสดังรูปที่ 2.25



รูปที่ 2.25 ตัวอย่างการใช้งาน วงจร Zener diode Suppressors

Capacitor Suppressors

วิธีนี้นิยมใช้กับ Stepping motor ที่มีขดลวดแบบ Unipolar winding โดยจะต่อ Capacitor ไว้ระหว่างเฟสที่ 1 กับ 3 และเฟสที่ 2 กับ 4 ดังรูปที่ 2.26



รูปที่ 2.26 Capacitor Suppressors

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดย Capacitor จะทำหน้าที่เป็นตัวหน่วงทางไฟฟ้า เพื่อลดการแกว่งไกวของ Rotor ใน Stepping motor ได้อีกด้วย

2.12 การลดเวลาหน่วงกระแสให้กับ Stepping motor

เมื่อพิจารณาขดลวดของ Stepping motor ดังที่กล่าวมาข้างต้นทำให้ทราบว่าภายในขดลวดของ Stepping motor สามารถเขียนเป็นวงจรสมมูลย์ได้ดังรูปที่ 2.22 และกระแสจะมีสมการดังสมการที่ (1) ซึ่งเมื่อป้อนแรงดันที่เป็น Unit step Function Pulse กระแสจะค่อยๆ เพิ่มขึ้นจนถึงค่าสูงสุด ตามสมการที่ (2) โดยใช้เวลา 5τ โดย τ คือค่าคงที่เวลา (Tine constant)

$$i = \frac{V}{R}\left(1 - e^{\frac{t}{\tau}}\right) \qquad \text{สมการ 2.5}$$

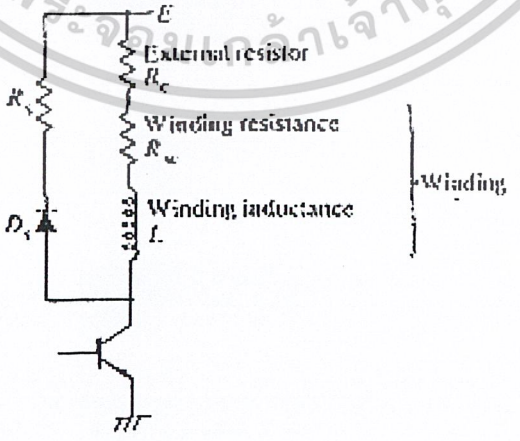
$$I = \frac{V}{R} \qquad \text{สมการ 2.6}$$

$$\tau = \frac{L}{R} \qquad \text{สมการ 2.7}$$

ซึ่งในการแก้ไขปัญหานี้ มี 2 วิธีคือ

(1). การใช้ Series Resistance

วิธีนี้แสดงดังรูปที่ 2.27 โดยการเพิ่มตัวต้านทาน (Re) เข้าไปอีก 1 ตัวโดยต่อแบบอนุกรมกับขดลวดของ Stepping motor ดังนั้นค่าคงที่ของเวลา (Time constant)



รูปที่ 2.27 การใช้ Series Resistance

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

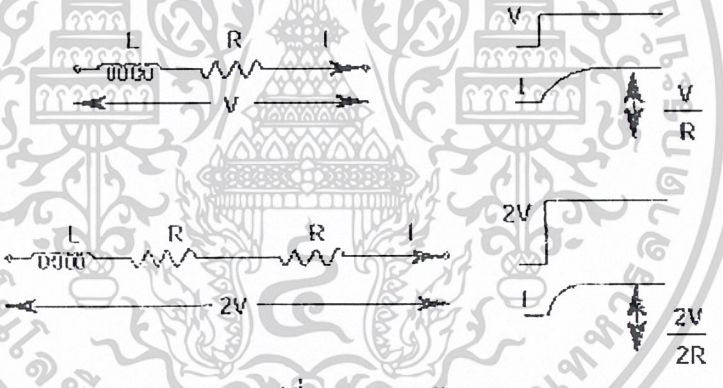
$$\tau = \frac{L}{R_e + R_w}$$ สมการ 2.8

จะทำให้ค่าคงที่ของเวลาลดลง แต่จะทำให้กระแสที่ไหลผ่านขดลวดลดลงด้วย เพื่อให้กระแสไหลเท่าเดิมจึงต้องเพิ่มแรงดันแหล่งจ่ายเพิ่มขึ้นตามสมการที่ (5) ดังรูปที่ 2.28

$$V = I \cdot (R_e + R_w)$$ สมการ 2.9

แต่จะเกิดการสูญเสียกำลังเพิ่มขึ้นที่ตัวต้านทานที่ต่อเข้าไป

$$P = I^2 \cdot R_e$$ สมการ 2.10

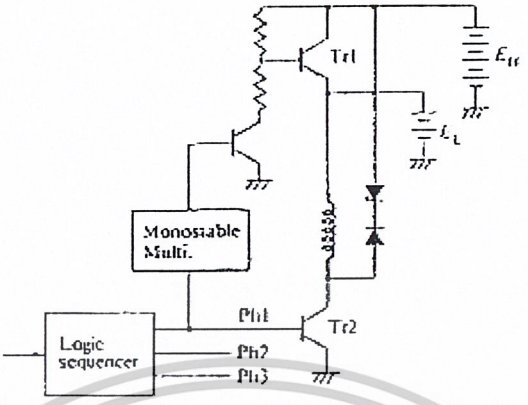


รูปที่ 2.28 การขับแบบ R-L

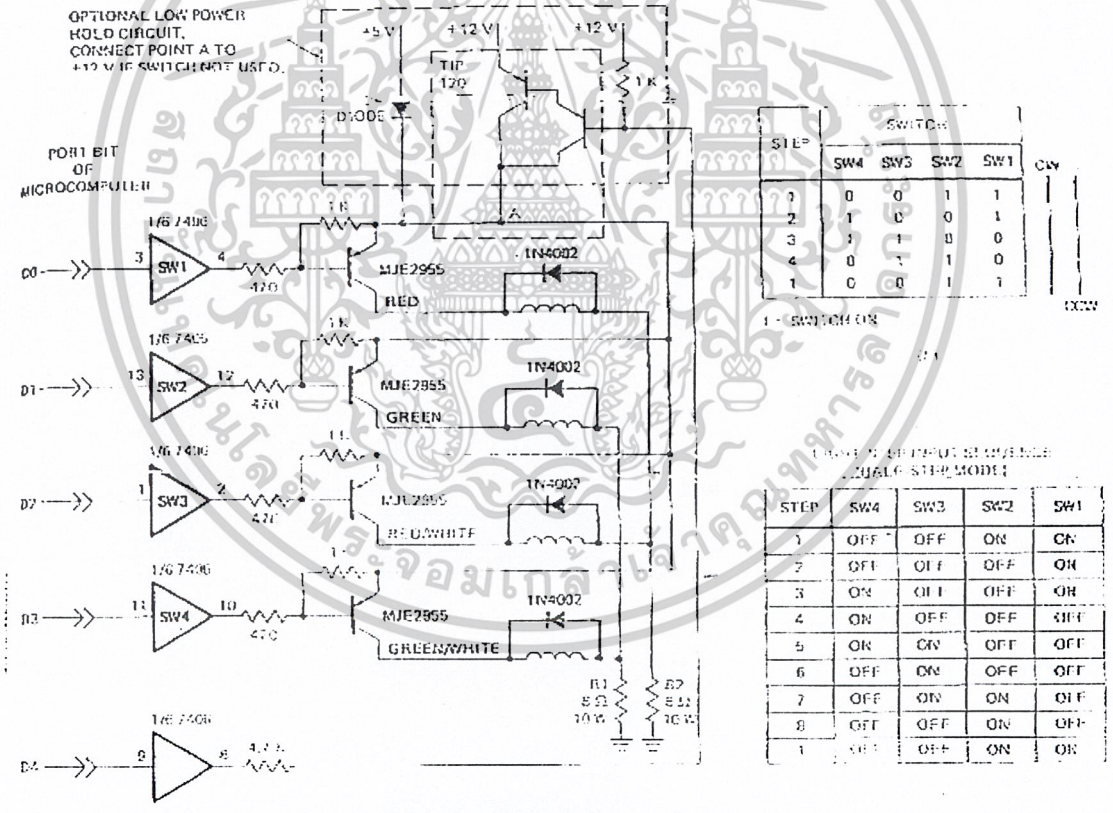
(2). การใช้แหล่งจ่าย (Dual power supply)

เพื่อลดการสูญเสียที่เกิดขึ้นเมื่อใช้ตัวต้านทานภายนอก (Re) ต่อเข้าไปและเพิ่มประสิทธิภาพให้กับ stepping motor ดังรูปที่ 2.29 และรูปที่ 2.30 โดยในช่วงแรกจะให้แหล่งจ่าย EH (High Voltage) ก่อนโดยจะกำหนดเวลาไว้ด้วยวงจร Monostable multivibrator ซึ่งจะให้ stepping motor มีกระแสไหลเข้าไปจนถึงค่าพิกัดของมอเตอร์แล้วก็จะจ่ายแหล่งจ่าย EL (Low Voltage) เข้าไปซึ่งนอกจากจะควบคุมด้วยวงจร Monostable multivibrator แล้วอาจจะควบคุมด้วย Microprocessor ได้เช่นเดียวกัน ดังรูปที่ 2.30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



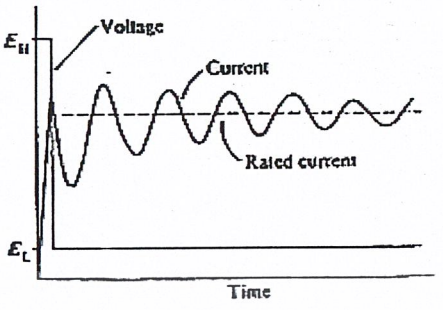
รูปที่ 2.29 การใช้แหล่งจ่ายคู่ (Dual power supply)



รูปที่ 2.30 การใช้แหล่งจ่ายคู่ที่ควบคุมด้วย Microprocessor (Dual power supply)

เนื่องจากผลของค่าตัวเหนี่ยวนำจะเห็นได้จากรูปที่ 2.31 กระแสในวงจรแรกจะช้ากว่า วงจรที่สองซึ่งใช้การลดค่าคงที่ของขดลวดลง โดยการเพิ่มตัวต้านทานเข้าไป แต่เพิ่มขนาดแรงดันไฟฟ้าอีกเพื่อให้กระแสมีขนาดเท่าเดิม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



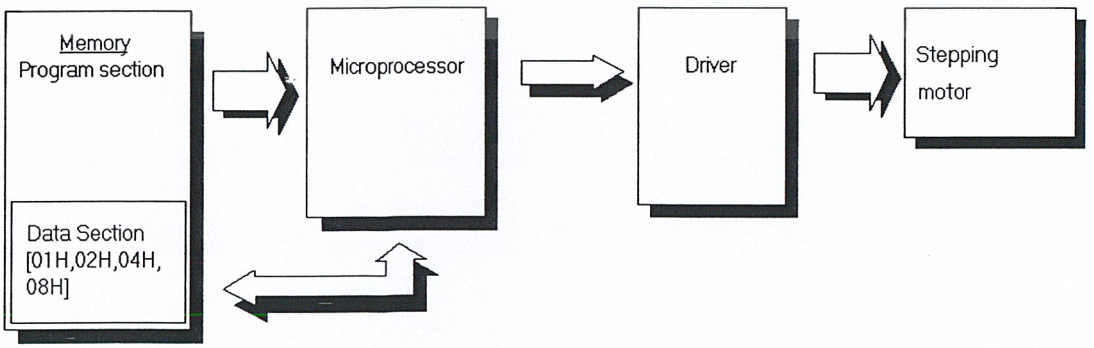
รูปที่ 2.31 รูปร่างของแรงดันและกระแสเมื่อขับแบบ Dual Power supply

2.13 การสร้างสัญญาณ Pulse เพื่อขับ Stepping motor ด้วย Microprocessor

นอกจากการใช้ Pulse sequencer ในการสร้างสัญญาณ Pulse เพื่อนำไปกระตุ้น Stepping motor แล้วการใช้ Microprocessor ก็เป็นอีกวิธีการหนึ่งที่เป็นที่นิยมเป็นอย่างมาก เนื่องจากมีความอ่อนตัวในการแก้ไข ปรับปรุง ไม่ว่าจะเป็นการเปลี่ยนการกระตุ้นแบบต่างๆ หรือการเปลี่ยนความเร็วรอบ การควบคุมจำนวนรอบ ซึ่งราคาถูกกว่าการเปลี่ยนแปลงในวงจร Pulse sequencer มาก เพราะเป็นการแก้ไขเฉพาะที่โปรแกรมเท่านั้นและยังสามารถที่จะทำการควบคุมแบบลูปเปิด (closed loop) ได้ง่ายอีกด้วย

ในปัจจุบันนี้ราคาของ Microprocessor ได้ลดต่ำลงมามากโดยเฉพาะ Microcontroller อีกทั้งมีให้เลือกใช้งานมาก

Microprocessor จะทำงานตามคำสั่งที่บรรจุไว้ในหน่วยความจำ (Memory) ดังนั้นเพื่อที่จะให้ Microprocessor ทำงานได้จึงต้องประกอบด้วยหลายๆ ส่วนและในการขับ stepping motor จะมีลักษณะการทำงานดังรูปที่ 2.32



รูปที่ 2.32 ลักษณะการทำงานของการใช้ Microprocessor กระตุ้น Stepping motor

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความรู้พื้นฐานเกี่ยวกับพอร์ตอนุกรม

ทำความรู้จักกับพอร์ตอนุกรม

มีทางเลือกอยู่ 2 ทางในการเคลื่อนย้ายข้อมูลทางคอมพิวเตอร์ไปยังอุปกรณ์ต่อพ่วงอื่นๆ หรือคอมพิวเตอร์ด้วยกัน นั่นคือการรับส่งข้อมูลแบบขนานและการรับส่งข้อมูลแบบอนุกรม การรับส่งข้อมูลแบบขนาน จะเป็นการรับส่งข้อมูลคราวละ 4 ถึง 8 บิต ในเวลาเดียวกันซึ่งจะทำให้การรับส่งข้อมูลทำได้อย่างรวดเร็วซึ่งก็หมายความว่าจำนวนของสายที่ใช้ในการส่งจะต้องมีเท่ากับจำนวนของบิตที่จะใช้ส่ง นอกจากนี้ยังรวมถึงสายที่ใช้สำหรับการควบคุมและการตรวจสอบการรับส่งข้อมูลด้วย ซึ่งอาจจะต้องใช้สายเป็นจำนวนมากเป็น 2 เท่า ของจำนวนบิตที่จะส่งได้ ซึ่งจะเป็นปัญหาในเรื่องราคาของสายที่ใช้ในการเชื่อมต่อแบบขนานมักมีราคาแพง

ในขณะการรับส่งข้อมูลแบบอนุกรมเป็นการรับส่งข้อมูลครั้งละ 1 บิต แต่ก็สามารถส่งข้อมูลทีละหลายๆ บิตได้ หากแต่ต้องมีการตกลงกันระหว่างตัวรับกับตัวส่ง จะรับส่งข้อมูลคราวละกี่บิต ตัวรับจะต้องรับข้อมูลให้ครบทุกบิตเสียก่อนจึงทำการประมวลผล ส่งผลให้การสื่อสารข้อมูลแบบอนุกรมมีความเร็วต่ำกว่าแบบขนาน ในด้านจำนวนสายสัญญาณการรับส่งข้อมูลแบบอนุกรมจะใช้จำนวนสายน้อยกว่ามาก อย่างน้อยที่สุดใช้ 2 - 3 เส้นเท่านั้น แต่อัตราเร็วในการรังส่งต่ำกว่าแบบขนาน

การสื่อสารแบบอนุกรม

การสื่อสารแบบอนุกรมอาจแบ่งได้เป็น 2 แบบคือการสื่อสารแบบอนุกรมซิงโครนัสและการสื่อสารแบบอนุกรมอะซิงโครนัส การสื่อสารแบบซิงโครนัสจะมีสัญญาณนาฬิการ่วมอยู่กับการรับและส่งสัญญาณด้วย ตัวอย่างการส่งข้อมูลแบบซิงโครนัสก็คือ คีย์บอร์ดของคอมพิวเตอร์ ซึ่งสายเส้นหนึ่งจะเป็นสัญญาณนาฬิกา ส่วนสายอีกเส้นหนึ่งจะเป็นสายของข้อมูล ดังนั้นการติดต่อกันแบบซิงโครนัสจึงต้องใช้สายต่อกันอย่างน้อย 3 เส้น คือ สัญญาณนาฬิกา ข้อมูล และกราวด์

การสื่อสารข้อมูลแบบอะซิงโครนัส

การสื่อสารข้อมูลแบบอะซิงโครนัสคือการรับและส่งข้อมูลไปในสายโดยไม่จำเป็นต้องมีสัญญาณนาฬิการ่วมด้วยเหมือนกับการรับส่งข้อมูลแบบซิงโครนัส แต่จะกำหนดค่าสัญญาณนาฬิกาทั้งภาครับและภาคส่งให้มีค่าเท่ากัน ซึ่งเรียกสัญญาณนาฬิกาที่ใช้กำหนดค่าให้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาครับและภาคส่งนี้ว่า อัตราการถ่ายทอดข้อมูล หรือ บอดเรต (baudrate) มีหน่วยเป็นบิตต่อวินาที (bit per second : bps)

รูปแบบข้อมูลที่ใช้ในการรับส่งแบบอะซิงโครนัส ประกอบด้วย 4 ส่วนคือ

1. บิตเริ่มต้น (Start Bit) ซึ่งจะมีขนาด 1 บิต

1. บิตข้อมูลแบบอนุกรมจะมีขนาด 5, 6, 7 หรือ 8 บิต

2. บิตตรวจสอบพาริตี (Parity Bit) จะมีขนาด 1 บิตหรือไม่มี

3. บิตปิดท้าย (Stop Bit) จะมีขนาด 1, 1.5 หรือ 2 บิต

อุปกรณ์พิเศษได้รับการออกแบบมาสำหรับการรับและส่งข้อมูลแบบอะซิงโครนัส เรียกว่า Universal Asynchronous Redeiver / Transmitter หรือ UART อัตราความเร็วในการรับและส่งข้อมูลแบบอะซิงโครนัสคือค่าบอดเรต ซึ่งก็คือจำนวนบิตต่อวินาทีที่ใช้ในการรับและส่งข้อมูลบอดเรตมาตรฐาน ที่ใช้สำหรับพอร์ตอนุกรม RS-232 ได้แก่ 110, 150, 300, 600, 1200, 2400, 4800, 9600 และ 19200 บิตต่อวินาที และมีค่ามากขึ้นตามเทคโนโลยีของคอมพิวเตอร์ซึ่งการรับส่งแบบอนุกรมโดยไม่ผ่านโมเด็มอาจจะกำหนดค่าบอดเรตได้สูงถึง 115200 บิตต่อวินาที เนื่องจากบอดเรต คือจำนวนบิตของข้อมูลที่สามารถถ่ายทอดได้ภายใน 1 วินาที ยกตัวอย่างข้อมูลแบบอนุกรมถูกส่งในลักษณะ 8 บิต ไม่มีการตรวจสอบพาริตี มีบิตเริ่มต้น 1 บิต และบิตปิดท้าย 1 บิต ความยาวของข้อมูลที่รับส่งนี้เท่ากับ 10 บิต ถ้าใช้บอดเรตในการส่งข้อมูลเท่ากับ 9600 บิตต่อวินาที ก็จะสามารถรับส่งข้อมูลด้วยความเร็ว 960 ไบต์ต่อวินาที และถ้ามีการใช้พาริตีความเร็วในการรับส่งจะเป็น 872 ไบต์ต่อวินาที

การตรวจสอบพาริตีสามารถกำหนดให้เป็นแบบคี่ (Odd) แบบคู่ (Even) หรือไม่มีการตรวจสอบพาริตีก็ได้ การตรวจสอบพาริตีเป็นการตรวจสอบจำนวนรวมของบิตที่เป็นลอจิก “1” ภายในข้อมูลที่ส่งไป 1 ไบต์ มีจำนวนรวมเป็นเลขคู่หรือเลขคี่โดยต้องรวมบิตพาริตีเข้าไปด้วย ยกตัวอย่าง ข้อมูลที่จะทำการส่งมีขนาด 8 บิตและมีค่าเท่ากับ 99 ฐานสิบหก หรือ 1001 1001 ฐานสอง จะเห็นว่าข้อมูลในไบต์นี้มีจำนวน ลอจิก “1” จำนวน 4 ตัวซึ่งเป็นเลขคู่ ค่าในบิตพาริตี จะต้องมี ลอจิกเป็น “0” แต่ถ้าพาริตีเป็นคี่ คาพาริตีจะต้องเป็น “1” เพื่อให้ข้อมูล 1 ไบต์รวมทั้งบิตพาริตีที่มีจำนวนบิตที่เป็น ลอจิก “1” มีจำนวนรวมกันเป็นเลขคี่

บิตพาริตีถูกสร้างขึ้นจากภาคส่งข้อมูลของ UART ซึ่งทางภาครับจะต้องทำการกำหนดคุณสมบัติการตรวจสอบพาริตี ให้ตรงกันว่าจะตรวจสอบพาริตีคู่หรือพาริตีคี่ จากนั้นภาครับของ UART จะทำการตรวจสอบค่าพาริตีที่เกิดขึ้นว่าเป็นคู่หรือเป็นคี่ โดยการนับจำนวนลอจิก “1” ทั้งหมดรวมทั้งบิตพาริตีด้วย ถ้ากำหนดพาริตีไว้เป็นคู่แต่อ่านค่าตัวเลขในการนับออกมาได้เป็นเลขคี่ทางภาครับจะแสดงข้อผิดพลาดออกมาให้ผู้ใช้ทราบ นับเป็นการตรวจสอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความผิดพลาดที่เกิดขึ้นในการถ่ายทอดข้อมูลที่ง่ายที่สุด แต่จะเชื่อถือได้เมื่อมีบิตข้อมูลที่ทำการส่งผิดพลาดเพียงบิตเดียวเท่านั้น ถ้าข้อมูลที่ทำการส่งมีข้อผิดพลาดมากกว่า 1 บิต การตรวจสอบด้วยวิธีนี้จะไม่ได้ผล สำหรับการตั้งพาริตีบิตเป็น NONE นั้นทั้งภาครับและภาคส่ง จะไม่มีการตรวจสอบพาริตี

มาตรฐานพอร์ตอนุกรมแบบ RS-232

มาตรฐานการเชื่อมต่อแบบอนุกรม RS-232 เป็นมาตรฐานอุตสาหกรรมที่ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบอะซิงโครนัส 2 ทิศทาง โดยมาตรฐาน RS-232 ในอดีตนั้นถูกออกแบบมาเพื่อการส่งผ่านข้อมูลจากคอมพิวเตอร์ไปยังโมเด็มเพียงอย่างเดียว เพื่อที่จะนำข้อมูลจากโมเด็มนั้นสื่อสารผ่านสายโทรศัพท์ไปยังคอมพิวเตอร์อีกชุดซึ่งอยู่ไกลกัน โดยคณะกรรมการที่เรียกว่า สมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronic Inductrues Associatuin: EIA) ได้วางมาตรฐานที่มีชื่อเรียกกันว่า EIA RS-232 มาตรฐานนี้ในระยะแรกจะใช้คอนเน็กเตอร์เป็นแบบ DB-25 โดยกำหนดความยาวสูงสุดของสายสัญญาณไว้ที่ 50 ฟุต มีสัญญาณไว้ที่ 50 ฟุต มีสัญญาณตั้งแต่ -3 ถึง -12 V แสดงว่ามีข้อมูล (Mark) และ +3 ถึง +12 V แสดงว่าเป็นช่องว่าง (Space)

มาตรฐาน RS-232 ได้กำหนดรูปแบบอุปกรณ์เชื่อมต่อข้อมูล (Data Terminal Equipment: DTE) กับวงจรข้อมูลปลายทาง (Data Circuit Terminting: DCE) ไว้ว่าอุปกรณ์ DTE จะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัวเช่น ไมโครคอนโทรลเลอร์ หรือไมโครคอมพิวเตอร์ซึ่งมีความสามารถในการสร้างบิตข้อมูลแบบอนุกรม ส่วนอุปกรณ์ DCE จะทำหน้าที่เป็นเพียงตัวรับข้อมูลที่ส่งมาจาก DTE เท่านั้น โดยการรับส่งข้อมูลระหว่างอุปกรณ์ทั้งสองจะกระทำผ่านมาตรฐาน RS-232

ข้อแตกต่างของอุปกรณ์ DTE และอุปกรณ์ DCE อย่างหนึ่งที่เห็นชัดก็คือ คอนเน็กเตอร์ของ DTE จะเป็นตัวเมีย ซึ่งพอร์ตอนุกรมของคอมพิวเตอร์ ที่ใช้กันอยู่ทั่วไปจะเป็นแบบ DTE ส่วนคอนเน็กเตอร์ที่อยู่ที่โมเด็มจะเป็นแบบ DCE สำหรับการใช้งานบนคอมพิวเตอร์ พอร์ตอนุกรม RS-232 มักถูกใช้เชื่อมต่อกับโมเด็มหรือเมาส์ โดยสามารถรับส่งข้อมูลได้ที่ความยาวของสายสัญญาณสูงสุดถึง 20 ฟุต คอนเน็กเตอร์สำหรับพอร์ต RS-232 และการเชื่อมต่อมาตรฐานการเชื่อมต่อแบบ RS-232 จะใช้คอนเน็กแบบ DB-25 ตัวผู้หรือ DB-9 ตัวผู้ ซึ่งคอนเน็กเตอร์แบบ DB-25 จะมีขาต่อใช้งานเพียง 9 เส้น เช่นเดียวกับคอนเน็กเตอร์แบบ DB-9 เนื่องจากขาอื่นๆ ที่เคยใช้ในอดีตปัจจุบันมีการใช้งานไม่มากนักจึงถูกยกเลิกไป

ลักษณะสัญญาณอินพุทและเอาท์พุทของพอร์ต RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณเอาท์พุทที่ใช้ควบคุม (RTS และ DTR) และสัญญาณแสดงสถานะอินพุท(CTS, DSR และ DCD) ของพอร์ตอนุกรม RS-232 จะถูกกลับสถานะภายในตัว UART ส่วนสัญญาณข้อมูลทั้งภาคส่งและรับจะไม่ถูกกลับสถานะ UART จะให้ระดับสัญญาณเอาท์พุทออกมาเป็นแบบทีทีแอลเท่านั้น ดังนั้นเมื่อสัญญาณถูกส่งออกมาจาก UART จึงต้องส่งเข้าสู่วงจรขับเพื่อปรับระดับแรงดันให้ได้ระดับสัญญาณเป็นไปตามมาตรฐาน RS-232 ก่อนส่งออกไปจากคอมพิวเตอร์สำหรับอุปกรณ์ต่อเชื่อมปลายทางก็จะต้องมีวงจรขับในลักษณะนี้เช่นเดียวกัน เพื่อให้ได้ระดับสัญญาณในระดับเดียวกัน แต่วงจรขับที่ใช้ทั้งภายในคอมพิวเตอร์และอุปกรณ์ต่อเชื่อมปลายทางนั้นจะถูกกลับสถานะ

แอดเดรสของพอร์ตอนุกรม

แอดเดรสพื้นฐานของพอร์ตอนุกรมมี 4 ตำแหน่งดังนี้คือ

COM1 : 3F8H

COM2 : 2F8H

COM3 : 3E8H

COM4 : 2E8H

ตารางที่ 2.3 แสดงข้อมูลในแอดเดรส 0000 : 0411H ที่ใช้แจ้งจำนวนพอร์ตอนุกรม

บิต 3	บิต 2	บิต 1	จำนวนพอร์ต
0	0	0	ไม่มีพอร์ตอนุกรม
0	0	1	มีพอร์ตอนุกรม 1 พอร์ต
0	1	0	มีพอร์ตอนุกรม 2 พอร์ต
0	1	1	มีพอร์ตอนุกรม 3 พอร์ต
1	0	0	มีพอร์ตอนุกรม 4 พอร์ต

เมื่อเริ่มเปิดเครื่องเพื่อใช้งานคอมพิวเตอร์ ไบออสภายในคอมพิวเตอร์จะทำการตรวจสอบแอดเดรสของพอร์ตอนุกรมทั้งหมด ถ้าไบออสตรวจพบแอดเดรสของพอร์ตอนุกรม ไบออสจะนำแอดเดรสที่ตรวจพบไปเก็บไว้ในหน่วยความจำขนาด 2 ไบต์ สำหรับพอร์ตอนุกรม COM 1จะเก็บไว้ที่แอดเดรส 0000 : 0400H และ 0000 : 0401H ส่วนตำแหน่งอื่นๆ มีรายละเอียดดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

COM12 = 0000 : 0402H - 0000 : 0403H

COM13 = 0000 : 0404H - 0000 : 0405H

COM14 = 0000 : 0406H - 0000 : 0407H

นอกจากนี้ที่หน่วยความจำแอดเดรส 0000 : 0411H ยังใช้สำหรับแสดงจำนวนของพอร์ตอนุกรมที่มีใช้อยู่ในคอมพิวเตอร์อีกด้วย

การหาค่าตำแหน่งแอดเดรสของพอร์ตอนุกรม

สามารถดูได้จากวินโดวส์ 95/98/ME โดยไปที่ Control Panel เรียก System -> Device Manager -> Ports (COM & LPT) ส่วนวินโดวส์ 2000 เข้าไปดูที่ Control Panel เรียก System -> Hardware -> Device Manager -> Ports (COM & LPT) จากนั้นเลือกพอร์ตอนุกรมที่ต้องการดูค่าแล้วเลือก Properties -> Resource ซึ่งในหน้าต่างนี้จะแสดงทั้งตำแหน่งแอดเดรสของพอร์ตอนุกรมนั้นๆ รวมถึงตำแหน่งของอินเตอร์รัปต์ที่ใช้ด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
วิธีดำเนินการวิจัย

3.1 ส่วนประกอบของโครงงาน

โครงงานนี้ประกอบไปด้วย 2 ส่วน หลักๆ ได้แก่

1. ส่วนควบคุมระบบขับเคลื่อน ได้แก่ ชุดไมโครคอนโทรลเลอร์ ชุดขับ Vexta และแท่นสแกน XY

2. ส่วนควบคุมการทำงาน ได้แก่ โปรแกรมวิชวลเบสิกที่สั่งงานผ่านคอมพิวเตอร์โดยผู้ใช้สามารถสั่งงานผ่านหน้าจอคอมพิวเตอร์ซึ่งเขียนด้วยโปรแกรมวิชวลเบสิก จากโปรแกรมวิชวลเบสิกจะเชื่อมต่อกับชุดไมโครคอนโทรลเลอร์ผ่านทางพอร์ตอนุกรม (MSCOMM1) และจากชุดคอนโทรลเลอร์ไปสั่งงานมอเตอร์ผ่านชุดขับให้ทำงานตามต้องการ ดังแสดงในรูปที่ 3.1



รูปที่ 3.1 ส่วนประกอบของโครงงาน

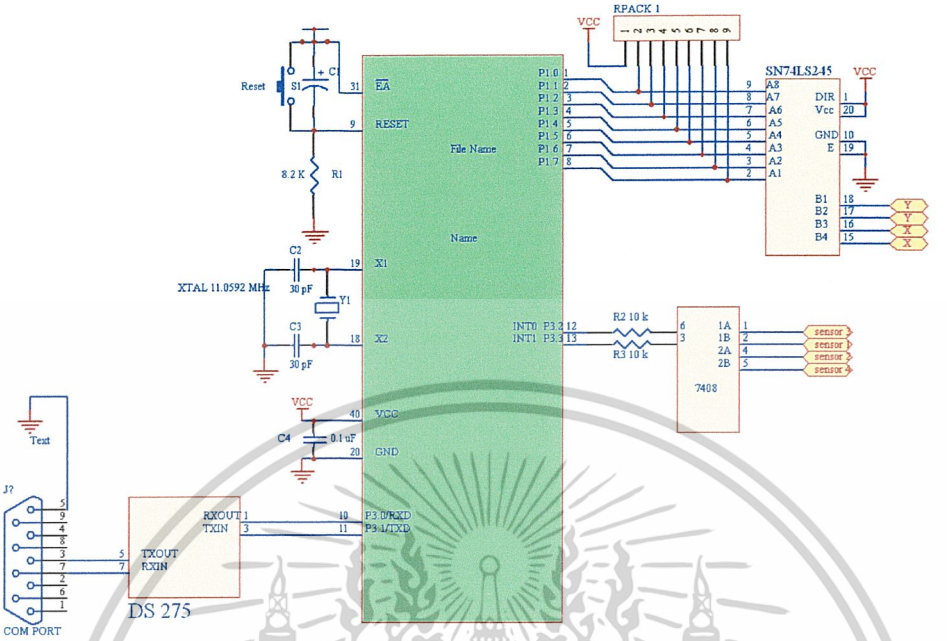
3.2 ส่วนควบคุมระบบขับเคลื่อน

การทำงานของส่วนนี้ควบคุมด้วยชุดคอนโทรลเลอร์ที่เขียนด้วยภาษาแอสเซมบลี ทำหน้าที่ควบคุมการทำงานของสเตปเปอร์มอเตอร์ในการรับส่งข้อมูลกับคอมพิวเตอร์ผ่านทางพอร์ตอนุกรม

3.2.1 ออกแบบวงจรของชุดคอนโทรลเลอร์

การออกแบบชุดคอนโทรลเลอร์มีส่วนประกอบหลักคือ MCS51, วงจรรีเซต, วงจรกำเนิดสัญญาณนาฬิกา, ฟิลเตอร์เพื่อกันสัญญาณรบกวน และส่งข้อมูลออกทางพอร์ตที่ P1 โดยมีการต่ออินเทอร์รัพท์ที่พอร์ต P 3.2 และ P3.3 ดังแสดงในรูปที่ 3.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 วงจรชุดไมโครคอนโทรลเลอร์

หลักการทำงาน

ไมโครคอนโทรลเลอร์เบอร์ AT89C51 เป็นส่วนประกอบที่สำคัญของวงจรทำหน้าที่รับข้อมูลจากคอมพิวเตอร์และไปควบคุมชุดขับเพื่อขับสเตปเปอร์มอเตอร์ผ่านทางพอร์ต P1.0 ถึง P1.7 แต่ในการใช้งานร่วมกับชุดขับ SUPER VEXTA ก่อนขับสเตปเปอร์มอเตอร์จะต้องผ่านบัฟเฟอร์เบอร์ 74LS245N เพื่อขยายกระแสโดยใช้รับอินพุทที่พอร์ต A1 – A8 และเอาท์พุทที่นำมาขับคือพอร์ต B1 – B8 ของบัฟเฟอร์

3.2.1.1 ศึกษาการทำงานของสเตปเปอร์มอเตอร์

เริ่มจากการต่อสเตปเปอร์มอเตอร์เข้ากับเครื่องกำเนิดความถี่ (generator) โดยตรงแล้วทดลองปรับความถี่เพื่อศึกษาว่าสเตปเปอร์มอเตอร์ชนิด 5 เฟส ที่ใช้งานนี้สามารถเริ่มทำงานที่ความถี่ต่ำสุดที่เท่าไร ซึ่งจากการทดลองนี้เราได้ค่าความถี่ต่ำสุดประมาณ 36 เฮิร์ซ ส่วนความถี่สูงสุดที่จะไม่ทำให้มอเตอร์เสียหายคือ 0.5 เมกะเฮิร์ซ แต่ในการใช้งานพบว่าหากป้อนความถี่เกิน 3 กิโลเฮิร์ซจะทำให้มอเตอร์กระตุกในตอนแรกก่อนที่จะหมุนต่อได้ ดังนั้นโครงงานนี้จึงเลือกใช้ความถี่ 5 ความถี่ในการกำหนดความเร็วให้ผู้ใช้เลือกคือ 1 kHz, 1.5 kHz, 2 kHz, 2.5 kHz และ 3 kHz หลังจากนั้นได้เขียนโปรแกรมทดสอบระยะทางที่มอเตอร์จะสามารถวิ่งได้โดยเปรียบเทียบจากระยะทางน้อยที่สุดที่มอเตอร์หมุนได้ 1 รอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.1.2 ทดสอบการทำงานของสเตปเปอร์มอเตอร์กับ Microcontroller

เมื่อทราบข้อจำกัดของสเตปเปอร์มอเตอร์ชนิด 5 เฟสแล้ว จึงเริ่มออกแบบชุดไมโครคอนโทรลเลอร์เพื่อติดต่อกับคอมพิวเตอร์ ซึ่งในขณะนี้ได้เขียนโปรแกรมภาษาแอสเซมบลีที่สามารถกำหนดความถี่ได้ขึ้นเองโดยผ่านพอร์ตอนุกรมเข้าสู่ชุดคอนโทรลเลอร์เพื่อสั่งงานชุดขับเพื่อให้สเตปเปอร์มอเตอร์ชนิด 5 เฟสทำงาน โดยความถี่ที่ป้อนเข้าไปนั้นมีผลต่อความเร็วในการหมุนของสเตปเปอร์มอเตอร์คือ เมื่อป้อนความถี่มากมอเตอร์ก็จะหมุนเร็ว

3.3 ส่วนควบคุมการทำงาน

เราใช้โปรแกรมวิชวลเบสิกในการออกแบบหน้าตาโปรแกรมและสั่งงานเชื่อมต่อระหว่างผู้ใช้กับส่วนควบคุมระบบขับเคลื่อน โดยแบ่งเป็น 2 ส่วนดังนี้

3.3.1 ส่วนติดต่อรับข้อมูลจากผู้ใช้

โดยเขียนโปรแกรม วิชวลเบสิกสำหรับรับข้อมูลจากผู้ใช้ เช่นรับค่าระยะทางที่ต้องการให้แท่นสแกนเคลื่อนที่, รับค่าความเร็วที่ต้องการ, หรือเลือกทิศทางที่ต้องการ

3.3.2 ส่วนที่ติดต่อรับและส่งข้อมูลกับไมโครคอนโทรลเลอร์

มีหน้าที่นำข้อมูลที่รับมาจากผู้ให้ส่งไปให้กับไมโครคอนโทรลเลอร์เพื่อกำหนดความถี่และตำแหน่งเพื่อให้แท่นสแกนทำงานตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4
ผลการทดลองและอภิปราย

1. โปรแกรมภาษาแอสเซมบลี

แท่นสแกน XY เป็นงานที่ต้องอาศัยความแม่นยำสูงในการระบุตำแหน่ง ดังนั้นผู้ทำจึงได้ใช้ ภาษาแอสเซมบลีในการเขียนเพราะเป็นภาษาที่เขียนง่าย ไม่ยุ่งยากเหมือนภาษาอื่น

เริ่มต้นโดยการเขียนโปรแกรมสร้างสัญญาณนาฬิกาขับให้ตัวขับทำงานเพื่อไปสั่งงานมอเตอร์ให้หมุนและสามารถเลือกทิศทางการทำงานตามที่ต้องการได้ และมีคำสั่งหยุดหากต้องการหยุดการทำงานของมอเตอร์ขณะที่เคลื่อนที่อยู่ ในที่นี้ได้กำหนดตัวอักษรแทนทิศทางทั้งสี่ คือ

N เคลื่อนที่ไปในทิศเหนือ
S เคลื่อนที่ไปในทิศใต้
W เคลื่อนที่ไปในทิศตะวันตก
E เคลื่อนที่ไปในทิศตะวันตก
H เคลื่อนที่ไปในทิศตะวันตกเฉียงเหนือ
J เคลื่อนที่ไปในทิศตะวันออกเฉียงเหนือ
K เคลื่อนที่ไปในทิศตะวันตกเฉียงใต้
L เคลื่อนที่ไปในทิศตะวันออกเฉียงใต้

แท่นสแกนมีขนาดกว้าง 65 มิลลิเมตร ยาว 90 มิลลิเมตร และหนึ่งรอบของมอเตอร์มี 500 สเตป เพราะหนึ่งสเตปของมอเตอร์เท่ากับ 0.72 องศา

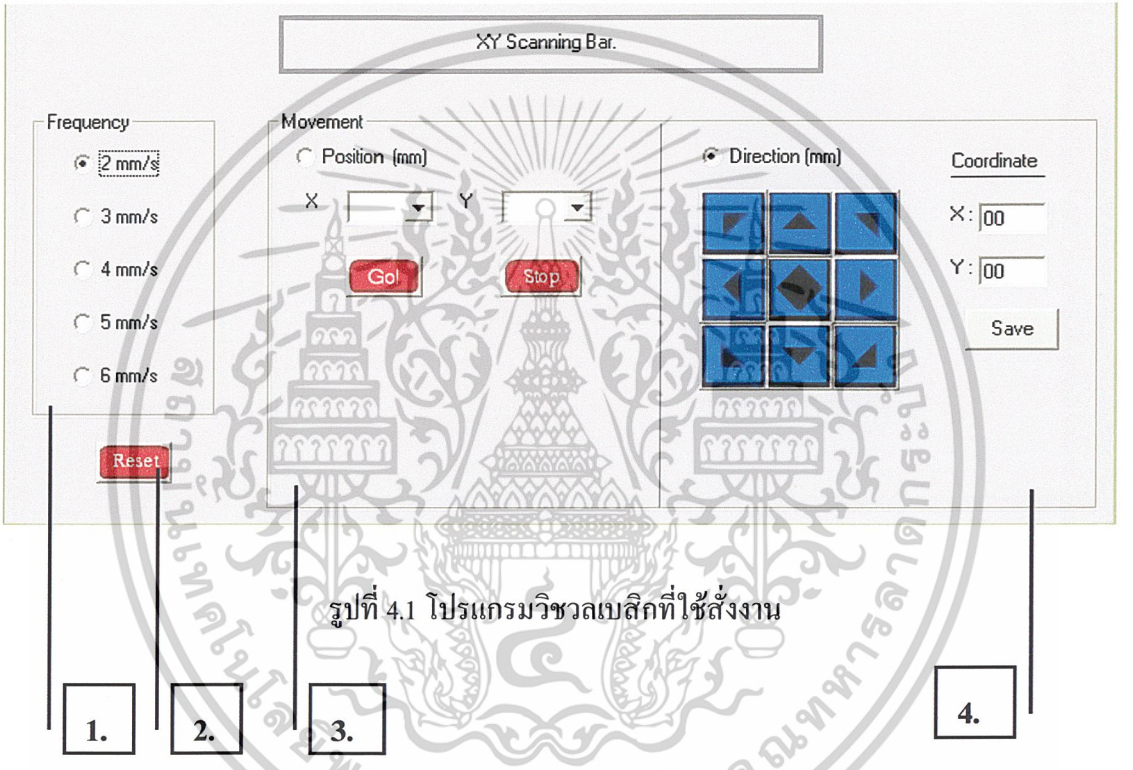
หลังจากนั้นได้เขียนโปรแกรมกำหนดความเร็วให้กับมอเตอร์ โดยป้อนเป็นค่าของความถี่ให้กับมอเตอร์ จากการทดลองพบว่าหากป้อนความถี่เกิน 3 kHz แล้วมอเตอร์จะกระตุกและไม่สามารถทำงานได้ดี ดังนั้นจึงได้เลือกความถี่สูงสุดให้เป็น 3 kHz และความถี่ต่ำสุดคือ 1 kHz และได้เพิ่มค่าระหว่างนั้นอีก 3 ค่า เพื่อให้ผู้ใช้สามารถเลือกความเร็วที่ต้องการได้ โดยคำนวณค่าจากความถี่เป็นความเร็วเพื่อให้ผู้ใช้สามารถเข้าใจและสามารถสั่งงานได้สะดวกมากขึ้น

2. โปรแกรมวิชวลเบสิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.โปรแกรมวิชวลเบสิก

เมื่อทราบถึงความต้องการในการสั่งงานแท่นสแกน จึงได้ออกแบบโปรแกรมวิชวลเบสิกให้ผู้ใช้สามารถสั่งงานได้อย่างสะดวก และง่ายต่อการทำความเข้าใจโปรแกรม โดยรูปที่ 1 ด้านล่างนี้แสดงให้เห็นหน้าตาโปรแกรมที่ออกแบบแล้ว โดยจะมีโหมดอยู่ทั้งหมด 3 โหมดด้วยกันและปุ่มรีเซ็ต ซึ่งแสดงดังข้างล่างนี้



รูปที่ 4.1 โปรแกรมวิชวลเบสิกที่ใช้สั่งงาน

โดยหมายเลข

1. คือช่องสำหรับเลือกความเร็วของมอเตอร์ โดยมีให้เลือก 5 ความเร็วด้วยกัน มีหน่วยเป็นมิลลิเมตรต่อวินาที
2. คือปุ่ม Reset ใช้เมื่อต้องการให้แท่นสแกนเคลื่อนที่ไปอยู่ที่จุด Origin เพื่อเริ่มต้นการทำงานใหม่
3. คือช่องเลือกตำแหน่งของแท่นสแกนที่มีทั้ง แกน X และ Y หลังจากนั้นกดปุ่ม Go! ในการให้แท่นสแกนทำงานและ Stop หากต้องการให้แท่นหยุดการทำงาน โดยระยะทางมีหน่วยเป็นมิลลิเมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. คือปุ่มสั่งงานให้แท่นสแกนเคลื่อนไปในทิศทางที่ต้องการทั้ง ขึ้น ลง ซ้าย ขวา และมุมเฉียงทั้งสี่มุม โดยลักษณะของปุ่มจะทำงานเหมือนสวิสซ์กดติดปล่อยดับ คือต้องกดปุ่มค้างไว้เมื่อต้องการให้แท่นเคลื่อนที่และแท่นสแกนจะหยุดเมื่อปล่อยปุ่มโดยมีช่องแสดงระยะทางที่แท่นสแกนเคลื่อนที่ไปเมื่อผู้ใช้กดปุ่ม มีหน่วยเป็นมิลลิเมตร



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปและข้อเสนอแนะ

5.1 สรุปผลการวิจัยและดำเนินงาน

วิชาฟิสิกส์นั้นจัดว่าเป็นวิชาที่มีบทบาทต่อการพัฒนาเทคโนโลยีต่าง ๆ เป็นอย่างมาก เนื่องจากต้องอาศัยหลักการและทฤษฎีหลาย ๆ ด้าน การจะศึกษาวิชาฟิสิกส์ให้มีความรู้ความเข้าใจและสามารถนำไปประยุกต์ใช้งานได้อย่างมีประสิทธิภาพนั้นจึงควรเริ่มตั้งแต่รากฐานที่ดี ด้วยเหตุนี้คณะผู้จัดทำจึงได้เล็งเห็นถึงความสำคัญของอุปกรณ์ในการทดลอง จึงได้สร้างชุดขับเคลื่อนแท่นสแกน XY ที่สามารถควบคุมผ่านทางเครื่องคอมพิวเตอร์ได้ ทำให้มีความสะดวกต่อการใช้งานหรือเพื่อการพัฒนาให้มีประสิทธิภาพดียิ่งขึ้น

5.2 ประสิทธิภาพของชุดขับเคลื่อนแท่นสแกน XY

5.2.1 สามารถเลือกความถี่ที่ป้อนให้มอเตอร์หมุนได้ 5 ความถี่

5.2.2 สามารถเลือกระยะทางที่ต้องการให้แท่นสแกนเคลื่อนที่ได้

5.2.3 สามารถกำหนดทิศทางให้แท่นสแกนเคลื่อนที่ตามที่ต้องการและสั่งให้หยุดเมื่อต้องการได้

5.2.4 เซนเซอร์สามารถขัดจังหวะการทำงานของโปรแกรมทำให้มอเตอร์ไม่สามารถเคลื่อนไปยังทิศทางที่เกิดการขัดจังหวะของโปรแกรมแต่สามารถทำให้เคลื่อนที่ไปยังทิศทางตรงกันข้ามได้

5.3 ข้อเสนอแนะและแนวทางในการพัฒนา

5.3.1 ควรทำอุปกรณ์ทุกอย่างให้อยู่ในรูปแบบที่ใช้ง่าย ไม่ยุ่งยากเกี่ยวกับการต่อวงจรง่ายต่อการเคลื่อนย้าย

5.3.2 พัฒนาโปรแกรมให้สามารถกำหนดความถี่ได้โดยผู้ใช้งาน โดยการเพิ่มโค้ดส่วนกำหนดความถี่โดยที่ลักษณะการทำงานของโปรแกรมยังคงเดิม

5.3.3 พัฒนาในสามารถนำไปใช้ในการพัฒนาให้เป็น 3 แกน โดยการเพิ่ม driver อีกชุดหนึ่ง โดยคอนโทรลเลอร์ตัวเดิมและเพิ่มโค้ดเพื่อควบคุมแกนที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

วิชิต ศิริโชติ. **การเขียนโปรแกรมภาษาแอสเซมบลี.** สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

ปลื้มใจ อินช่วย, ศิริชัย มินฮายีนุด และอุสา อภิญญานุรักษ์. 2543. **ระบบวัดเส้นสมศักย์ควบคุมด้วยคอมพิวเตอร์.** วิทยานิพนธ์ปริญญาวิทยาศาสตร์บัณฑิต สาขาวิชาฟิสิกส์ประยุกต์ คณะวิทยาศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง.

ปฏิพันธ์ ทวนทอง. 2542. **เอกสารประกอบการฝึกอบรม.** บริษัทฟิลลิปส์ เซมิคอนดัคเตอร์ ประเทศไทย จำกัด. (อัดสำเนา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก
โปรแกรมควบคุมชุดขับเคลื่อนแท่นสแกน xy ด้วย 89C51
(ภาษาแอสแซมบลี)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
$mod52
$noprint

        dseg  at  40h

        tick:ds      1
        count_hi:ds  1
        count_lo:ds  1
        speed:ds     1
        dir:ds       1
        com:ds       1

        flag_n:ds     1
        flag_s:ds     1
        flag_w:ds     1
        flag_e:ds     1

        pulse:ds     1
        count:ds     1
        select:ds    1
        round:ds     1
        round1:ds    1
        ground:ds    1

        get_x:ds     1
        get_y:ds     1

        esc_char equ 27

        cseg  at 0000h
        jmp   start1

        org   0003h
        jmp   external_0

        org   000bh
        jmp   service

        org   0013h
        jmp   external_1

        org   0100h

start1:mov  tmod,#00100000b
       mov  th1,#0fdh
       mov  tl1,#0fdh
       orl  tcon,#01001001b
       orl  scon,#01010010b

       mov  flag_n,#00h
       mov  flag_s,#00h
       mov  flag_w,#00h
       mov  flag_e,#00h

       mov  get_x,#00h
       mov  get_y,#00h

       mov p0,#0ffh

start:mov  tick,#00h
      mov  count_lo,#00h
      mov  count_hi,#00h
      mov  ground,#00h
      mov  count,#00h
      mov  pulse,#00h
;//////////////////////////GET VALUE\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\

      call cin
      mov  speed,a
      mov  a,speed
      cjne a,#'1',hi
      mov  count_lo,#033h
      mov  count_hi,#0feh
      ajmp  main
;//////////////////////////COMPARE\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\\
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 hi:mov  speed,a
    cjne a,#'2',hi1
    mov  count_lo,#0cdh
    mov  count_hi,#0feh
    ajmp  main

 hi1:mov  speed,a
     cjne a,#'3',hi2
     mov  count_lo,#01bh
     mov  count_hi,#0ffh
     ajmp  main

 hi2:mov  speed,a
     cjne a,#'4',hi3
     mov  count_lo,#043h
     mov  count_hi,#0ffh
     ajmp  main

 hi3:mov  speed,a
     cjne a,#'5',reset
     mov  count_lo,#066h
     mov  count_hi,#0ffh
     ajmp  main

 reset:mov  speed,a
       cjne a,#'6',start
       mov  count_lo,#066h
       mov  count_hi,#0ffh

       orl  tmod,#01h
       mov  th0,count_hi
       mov  tl0,count_lo
       setb ea
       setb et0
       setb tr0

 run:mov  a,tick
     cjne a,#0,run1
     mov  com,#00h

 run1:mov  a,tick
      cjne a,#1,run2
      mov  com,#0ffh

 run2:mov  a,tick
      cjne a,#2,tran
      mov  tick,#00h

 tran:mov  a,flag_s
      cjne a,#00h,tran1
      mov  a,com
      anl  a,#01000000b
      mov  p1,a
      jmp  run

 tran1:mov  a,flag_e
      cjne a,#00h,jump
      mov  a,com
      anl  a,#00010000b
      mov  p1,a
      jmp  run

 jump:ajmp start
 ;////////////////////////////////////////////////////

 main:orl  tmod,#01h
      mov  th0,count_hi
      mov  tl0,count_lo
      setb ea
      setb et0
      setb tr0

      setb ex0
      setb ex1
      clr  it0
      clr  it1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;////////////////////////////////////////////////////////////

mode:call cin
     mov  select,a
     cjne a,#'b',mode1
     jmp  direct

mode1:mov  a,select
      cjne a,#'a',new   ;edit
      jmp  get

direct:call cin
       mov  dir,a
       jmp loop

new: jmp start
;////////////////////////////////////////////////////////////////

loop:mov  a,tick
     cjne a,#0,loop1
     mov  com,#00h

loop1:mov  a,tick
      cjne a,#1,loop2
      mov  com,#0ffh

loop2:mov  a,tick
      cjne a,#2,direction
      mov  tick,#00h
      inc  count

direction:mov  a,dir
          cjne a,#'n',direction1                ;int0   N
          mov  a,flag_n
          cjne a,#00h,new
          mov  flag_s,#00h
          mov  a,com
          anl  a,#10000000b
          mov  p1,a

          mov  a,count
          cjne a,#100,loop
          mov  count,#00h
          inc  pulse
          mov  a,pulse
          cjne a,#5,loop
          inc  get_y
          mov  pulse,#00h

                mov a,get_x
                jnb ti,stop1
                clr ti
                mov sbuf,a

                mov a,get_y
                jnb ti,stop1
                clr ti
                mov sbuf,a

stop1:    jnb  ri,loop     ;edit
          clr  ri
          mov  a,sbuf
          cjne a,#'p',loop       ;PAUSE
          jmp  start

direction1:mov  a,dir
           cjne a,#'s',direction2               ;int1   S
           mov  a,flag_s
           cjne a,#00h,new
           mov  flag_n,#00h
           mov  a,com
           anl  a,#01000000b
           mov  p1,a

           mov  a,count
           cjne a,#100,loop
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            mov  count,#00h
            inc  pulse
            mov  a,pulse
            cjne a,#5,loop6
            dec  get_y
            mov  pulse,#00h

                 mov a,get_x
                 jnb ti,stop2
                 clr ti
                 mov sbuf,a

                 mov a,get_y
                 jnb ti,stop2
                 clr ti
                 mov sbuf,a

stop2:      jnb  ri,loop6
            clr  ri
            mov  a,sbuf
            cjne a,#'p',loop6       ;PAUSE
            jmp  start
loop6:  ajmp loop

direction2:mov  a,dir
           cjne a,#'w',direction3                ;int0    W
           mov  a,flag_w
           cjne a,#00h,new11
           mov  flag_e,#00h
           mov  a,com
           anl  a,#00100000b
           mov  p1,a

           mov  a,count
           cjne a,#100,loop4
           mov  count,#00h
           inc  pulse
           mov  a,pulse
           cjne a,#5,loop4
           inc  get_x
           mov  pulse,#00h

                mov a,get_x
                jnb ti,stop3
                clr ti
                mov sbuf,a

                mov a,get_y
                jnb ti,stop3
                clr ti
                mov sbuf,a

stop3:jnb  ri,loop4
      clr  ri
      mov  a,sbuf
      cjne a,#'p',loop4       ;PAUSE
      jmp  start

loop4:ajmp loop
new11:ajmp  start

direction3:mov  a,dir
           cjne a,#'e',direction4                ;int1      E
           mov  a,flag_e
           cjne a,#00h,new11
           mov  flag_w,#00h
           mov  a,com
           anl  a,#00010000b
           mov  p1,a

           mov  a,count
           cjne a,#100,loop4
           mov  count,#00h
           inc  pulse
           mov  a,pulse
           cjne a,#5,loop4
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            dec  get_x
            mov  pulse,#00h

                 mov  a,get_x
                 jnb     ti,stop4
                 clr     ti
                 mov     sbuf,a

                 mov  a,get_y
                 jnb     ti,stop4
                 clr     ti
                 mov     sbuf,a

stop4:jnb  ri,loop4
      clr  ri
      mov  a,sbuf
      cjne a,#'p',loop4      ;PAUSE
      jmp  start

new1:ajmp start
;////////////////////  EDIT ////////////////////////////

direction4:mov  a,dir
           cjne a,#'h',direction5                    ; N W = H
           mov  a,flag_n
           cjne a,#00h,re12
           mov  a,flag_w
           cjne a,#00h,re12
           mov  a,com
           anl  a,#10100000b                 ; 00100000b
           mov  p1,a

           mov  a,count
           cjne a,#100,loop4
           mov  count,#00h
           inc  pulse
           mov  a,pulse
           cjne a,#5,loop4
           inc  get_x
           inc  get_y
           mov  pulse,#00h

                mov  a,get_x
                jnb     ti,stop5
                clr     ti
                mov     sbuf,a

                mov  a,get_y
                jnb     ti,stop5
                clr     ti
                mov     sbuf,a

stop5:jnb  ri,loop7
      clr  ri
      mov  a,sbuf
      cjne a,#'p',loop7      ;PAUSE
      jmp  start

re12:ajmp start
loop7:ajmp loop

direction5:mov  a,dir
           cjne a,#'j',direction6                  ;N E=J
           mov  a,flag_n
           cjne a,#00h,re
           mov  a,flag_e
           cjne a,#00h,re
           mov  a,com
           anl  a,#10010000b
           mov  p1,a

           mov  a,count
           cjne a,#100,loop5
           mov  count,#00h
           inc  pulse
           mov  a,pulse
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
          cjne a,#5,loop5
          inc  get_x
          dec  get_y
          mov  pulse,#00h

               mov  a,get_x
               jnb  ti,stop6
               clr  ti
               mov  sbuf,a

               mov  a,get_y
               jnb  ti,stop6
               clr  ti
               mov  sbuf,a

stop6:jnb  ri,loop5
      clr  ri
      mov  a,sbuf
      cjne a,#'p',loop5        ;PAUSE
      jmp  start

re:ajmp start
loop5:ajmp loop

direction6:mov  a,dir
           cjne a,#'k',direction7                    ;W S = K
           mov  a,flag_w
           cjne a,#00h,re
           mov  a,flag_s
           cjne a,#00h,re
           mov  a,com
           anl  a,#01100000b
           mov  p1,a

           mov  a,count
           cjne a,#100,looop
           mov  count,#00h
           inc  pulse
           mov  a,pulse
           cjne a,#5,looop
           dec  get_x
           inc  get_y
           mov  pulse,#00h

                mov  a,get_x
                jnb     ti,stop7
                clr     ti
                mov     sbuf,a

                mov  a,get_y
                jnb     ti,stop7
                clr     ti
                mov     sbuf,a

stop7:jnb  ri,loop5
     clr  ri
     mov  a,sbuf
     cjne a,#'p',loop5        ;PAUSE
     jmp  start


direction7:mov  a,dir
           cjne a,#'l',stop                  ;S E =L
           mov  a,flag_s
           cjne a,#00h,re1
           mov  a,flag_e
           cjne a,#00h,re1
           mov  a,com
           anl  a,#01010000b
           mov  p1,a

           mov  a,count
           cjne a,#100,looop
           mov  count,#00h
           inc  pulse
           mov  a,pulse
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            cjne a,#5,looop
            dec  get_x
            dec  get_y
            mov  pulse,#00h

                 mov a,get_x
                 jnb ti,stop
                 clr ti
                 mov sbuf,a

                 mov a,get_y
                 jnb ti,stop
                 clr ti
                 mov sbuf,a
                 jmp  stop

re1:ajmp start

stop:jnb  ri,looop
     clr  ri
     mov  a,sbuf
     cjne a,#'p',looop       ;PAUSE
     jmp  start

looop:ajmp loop
;////////////////////////////////////////////////////////////////////////

get:call ghex
    mov  round,a
    call ghex
    mov  round1,a
;////////////////////////////////////////////////////////////////////////

turn:mov  a,tick
     cjne a,#0,turn1
     mov  com,#00h

turn1:mov  a,tick
      cjne a,#1,turn2
      mov  com,#0ffh

turn2:mov  a,tick
      cjne a,#2,increse
      mov  tick,#00h
      mov  com,#00h
      inc  count

increse:jnb  ri,mm1
        clr  ri
        mov  a,sbuf
        cjne a,#'p',mm1
        ajmp start1

mm1:mov  a,com
    anl  a,#10000000b
    mov  p1,a
    mov  a,count                              ;count pulse
    cjne a,#100,back
    mov  count,#00h
    inc  pulse
    mov  a,pulse
    cjne a,#5,back
    inc  ground

    mov  pulse,#00h
    mov  a,ground
    mov  b,round
    cjne a,round,turn
    mov  ground,#00h

    mov count,#00h
    mov p1,#00h
;*****************************************************
turn0:mov  a,tick
      cjne a,#0,turn01
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
       mov  com,#00h

turn01:mov  a,tick
       cjne a,#1,turn02
       mov  com,#0ffh

turn02:mov  a,tick
       cjne a,#2,increse1
       mov  tick,#00h
       mov  com,#00h
       inc  count

increse1:jnb  ri,mm
         clr  ri
         mov  a,sbuf
         cjne a,#'p',mm
         ajmp start1

mm:mov  a,com
   anl  a,#00100000b
   mov  p1,a
   mov  a,count                          ;count pulse
   cjne a,#100,turn0
   mov  count,#00h
   inc  pulse
   mov  a,pulse
   cjne a,#5,turn0
   inc  ground
   mov  pulse,#00h
   mov  a,ground

   cjne a,round1,turn0
   mov  ground,#00h

back:jmp turn
;*************************************************************

service: inc  tick
         mov  th0,count_hi
         mov  tl0,count_lo
         reti

external_0:mov a,p0
           cjne a,#7dh,cc1
           mov flag_e,#0ffh
cc1:       mov a,p0
           cjne a,#77h,cc2
           mov flag_s,#0ffh

cc2:       mov  p1,#10100000b
           call delay
           mov  p1,#00h
           call delay
           mov  p1,#10100000b
           call delay
           mov  p1,#00h
           call delay

           setb p3.2
           reti

external_1:mov a,p0
           cjne a,#7bh,cc3
           mov flag_n,#0ffh
cc3:       mov a,p0
           cjne a,#7eh,cc4
           mov flag_w,#0ffh

cc4:       mov  p1,#1010000b
           call delay
           mov  p1,#00h
           call delay
           mov  p1,#1010000b
           call delay
           mov  p1,#00h
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            setb p3.3
            reti
;*********/////----------END-----------\\\\\\*******

delay:mov r0,#0a0h
mmm:djnz r0,mmm
    ret
;////////////////////////////////////////////////////////////////////
cin_filter:jnb   ri, cinf1
           lcall cin
           cjne  a,#esc_char, cinf_end
        ;if esc was already in sbuf, just ignore it
cinf1:  lcall   cin
        cjne    a,#esc_char,cinf_end
cinf2:  acall   cinf_wait
        jb      ri,cinf4
        mov     a,#esc_char
        ret                         ;an ordinary ESC

cinf4:  ;if we get here, it's a control code, since a character
        ;was received shortly after receiving an ESC character
        lcall   cin
        cjne    a, #'[',cinf_consume
        acall   cinf_wait
        jnb     ri, cin_filter
        lcall   cin
cinf5a: cjne    a,#'A',cinf5b
        mov     a, #11
        ret
cinf5b: cjne    a,#'B',cinf5c
        mov     a,#10
        ret
cinf5c: cjne    a,#'C',cinf5d
        mov     a,#21
        ret
cinf5d: cjne    a,#'D', cinf5e
        mov     a,#8
        ret
cinf5e: cjne    a,#35h,cinf5f
        sjmp    cinf8
cinf5f: cjne    a,#36h,cinf5g
        sjmp    cinf8
cinf5g: sjmp    cinf_consume            ;unknown escape sequence

cinf8:  ;when we get here, we've got the sequence for pageup/pagedown
        ;but there's one more incoming byte to check...
        push    acc
        acall   cinf_wait
        jnb     ri, cinf_restart
        lcall   cin
        cjne    a,#7Eh,cinf_notpg
        pop     acc
        add     a,#228
cinf_end: ret
cinf_restart:
        pop     acc
        sjmp    cin_filter
cinf_notpg:
        pop     acc
cinf_consume:
        acall   cinf_wait
        jnb     ri, cin_filter
        lcall   cin
        cjne    a,#esc_char, cinf_consume
        sjmp    cinf2

        char_delay equ  4
cinf_wait:
        mov     a, r2
        push    acc
        mov     r2, #char_delay*5
cinfw2: mov     a, th0
cinfw3: jb      ri, cinfw4
        inc     a
        jnz     cinfw3
        djnz    r2,cinfw2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
cinfw4: pop     acc
        mov     r2,a
        ret

cin_filter_h:
        ljmp    cin_filter      ;62
        ajmp    asc2hex         ;64
cin:            jnb     ri,cin
                clr     ri
                mov     a,sbuf
                ret
cout:           jnb     ti,cout
                clr     ti              ;clr ti before the mov to sbuf!
                mov     sbuf,a
                ret

pstr:   push    acc
pstr1:clr       a
        movc a,@a+dptr
        inc     dptr
        jz      pstr2
        mov     c,acc.7
        anl     a,#7Fh
        acall   cout
        jc      pstr2
        sjmp    pstr1
pstr2:  pop     acc
                ret

;////////////////////////////////////////////////////////////////////////////
phex:
phex8:
        push    acc
        swap    a
        anl     a,#15
        add     a,#246
        jnc     phex_b
        add     a,#7
phex_b: add     a,#58
        acall   cout
        pop     acc
phex1:  push    acc
        anl     a,#15
        add     a,#246
        jnc     phex_c
        add     a,#7
phex_c: add     a,#58
        acall   cout
        pop     acc
        ret
phex16:
        push    acc
        mov     a,dph
        acall   phex
        mov     a,dpl
        acall   phex
        pop     acc
    ret
;////////////////////////////////////////////////////////////////////////////////////////////
ghex:
ghex8:  clr     psw.5
ghex8c:
        acall   cin_filter_h    ;get first digit
        acall   upper
        cjne    a, #27,ghex8f
ghex8d: setb    c
        clr     a
        ret
ghex8f: cjne    a,#13,ghex8h
        setb    psw.5
        clr     c
        clr     a
        ret
ghex8h: mov     r2,a
        acall   asc2hex
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        jc      ghex8c
        xch     a,r2            ;r2 will hold hex value of 1st digit
        acall   cout
ghex8j:
        acall   cin_filter_h    ;get second digit
        acall   upper
        cjne    a, #27,ghex8k
        sjmp    ghex8d
ghex8k: cjne    a, #13,ghex8m
        mov     a,r2
        clr     c
        ret
ghex8m: cjne    a, #8,ghex8p
ghex8n: acall   cout
        sjmp    ghex8c
ghex8p: cjne    a,#21,ghex8q
        sjmp    ghex8n
ghex8q: mov     r3,a
        acall   asc2hex
        jc      ghex8j
        xch     a,r3
        acall   cout
        mov     a,r2
        swap    a
        orl     a,r3
        clr     c
        ret

        ;carry set if esc pressed
        ;psw.5 set if return pressed w/ no input
ghex16:
        mov     r2,#0           ;start out with 0
        mov     r3,#0
        mov     r4,#4           ;number of digits left
        clr     psw.5

ghex16c:
        acall   cin_filter_h
        acall   upper
        cjne    a,#27,ghex16d
        setb    c               ;handle esc key
        clr     a
        mov     dph,a
        mov     dpl,a
        ret
ghex16d:cjne    a, #8,ghex16f
        sjmp    ghex16k
ghex16f:cjne    a,#127,ghex16g  ;handle backspace
ghex16k:cjne    r4,#4,ghex16e     ;have they entered anything yet?
        sjmp    ghex16c
ghex16e:acall   cout
        acall   ghex16y
        inc     r4
        sjmp    ghex16c
ghex16g:cjne    a,#13,ghex16i     ;return key
        mov     dph,r3
        mov     dpl,r2
        cjne    r4,#4,ghex16h
        clr     a
        mov     dph,a
        mov     dpl,a
        setb    psw.5
ghex16h:clr     c
        ret
ghex16i:mov     r5,a            ;keep copy of original keystroke

        acall   asc2hex
        jc      ghex16c
        xch     a, r5
        lcall   cout
        mov     a, r5
        push    acc
        acall   ghex16x
        pop     acc
        add     a,r2
        mov     r2,a
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        clr     a
        addc    a,r3
        mov     r3,a
        djnz    r4,ghex16c
        clr     c
        mov     dpl,r2
        mov     dph,r3
        ret

ghex16x:  ;multiply r3-r2 by 16 (shift left by 4)
        mov     a,r3
        swap    a
        anl     a,#11110000b
        mov     r3,a
        mov     a,r2
        swap    a
        anl     a,#00001111b
        orl     a,r3
        mov     r3,a
        mov     a,r2
        swap    a
        anl     a,#11110000b
        mov     r2,a
        ret

ghex16y:  ;divide r3-r2 by 16 (shift right by 4)
        mov     a,r2
        swap    a
        anl     a,#00001111b
        mov     r2,a
        mov     a,r3
        swap    a
        anl     a,#11110000b
        orl     a,r2
        mov     r2,a
        mov     a,r3
        swap    a
        anl     a,#00001111b
        mov     r3,a
        ret


        ;carry set if invalid input
asc2hex:
        clr     c
        add     a,#208
        jnc     hex_not
        add     a,#246
        jc      hex_maybe
        add     a,#10
        clr     c
        ret
hex_maybe:
        add     a,#249
        jnc     hex_not
        add     a,#250
        jc      hex_not
        add     a,#16
        clr     c
        ret
hex_not:setb    c
        ret
upper:  ;converts the ascii code in Acc to uppercase, if it is lowercase
        push    acc
        clr     c
        subb    a,#97
        jc      upper2          ;is it a lowercase character
        subb    a,#26
        jnc     upper2
        pop     acc
        add     a,#224          ;convert to uppercase
        ret
upper2: pop     acc             ;don't change anything
        ret
;////////////////////////////////////////////////////////////////
```

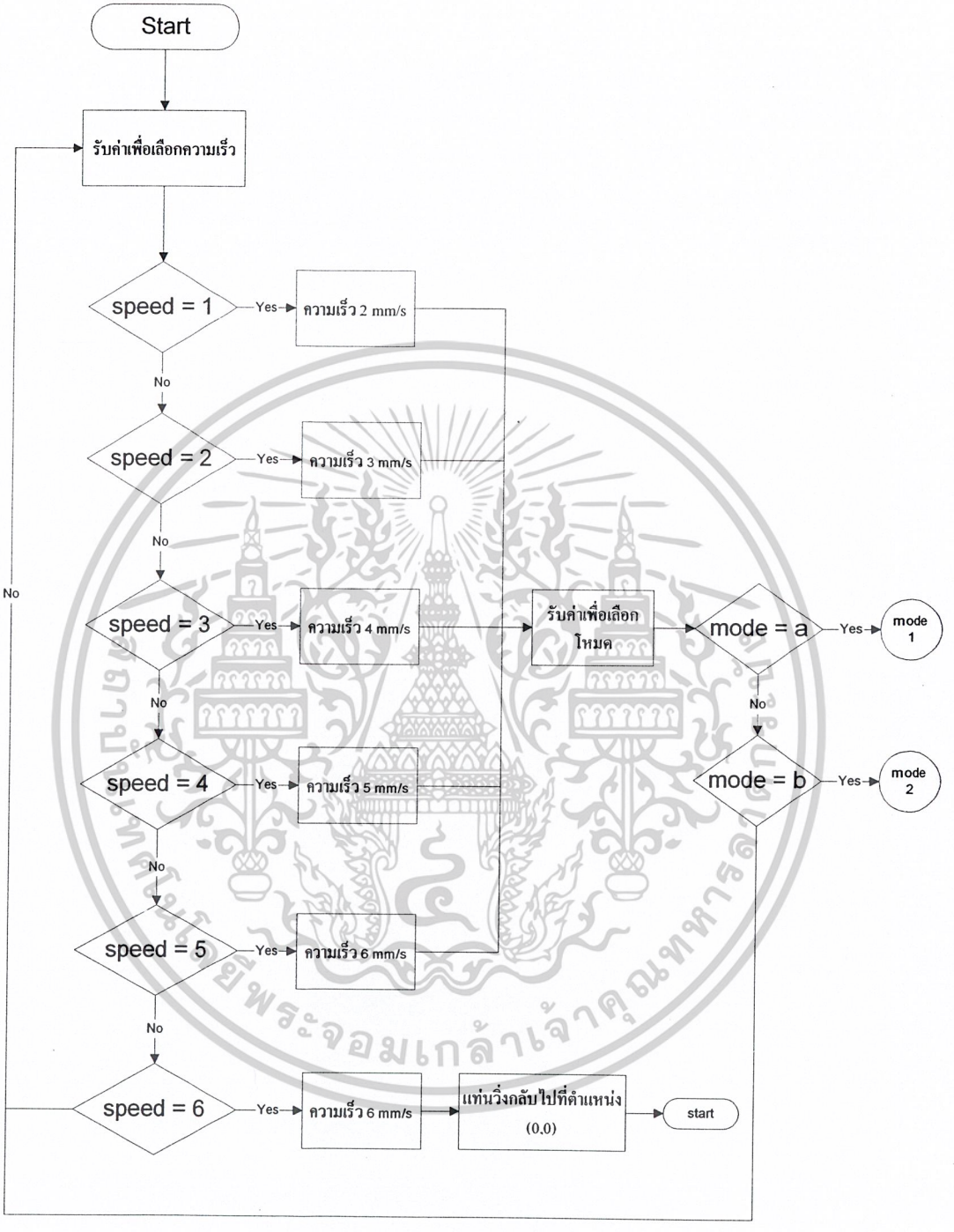
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
end
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข
โฟลชาร์ทของโปรแกรมในส่วนต่าง ๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



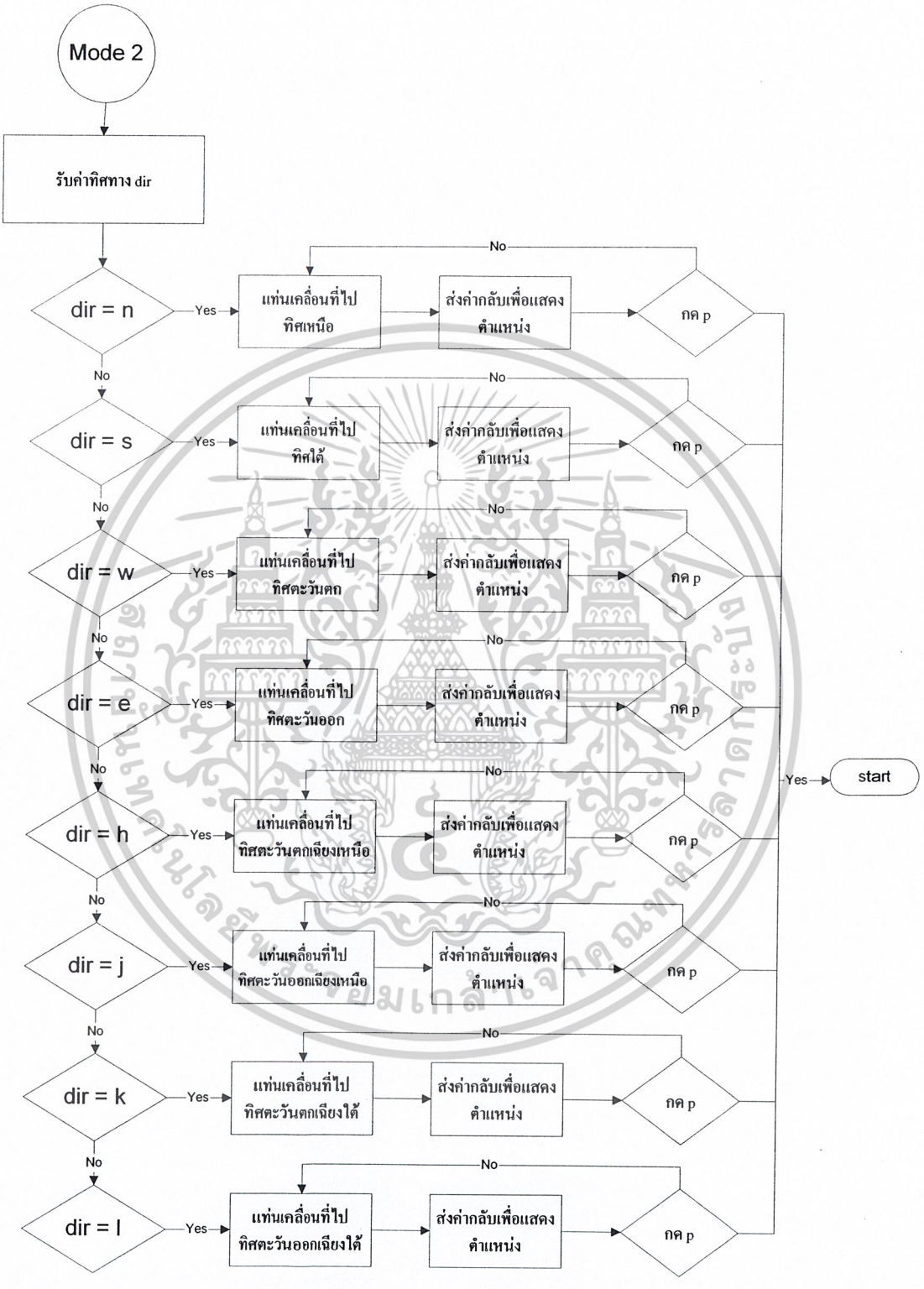
รูปที่ 1 โฟล์ชาร์ทการรับค่าความเร็วของมอเตอร์ก่อนแยกการทำงานในแต่ละโหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2 โฟร์ชาร์ทการทำงานในโหมดระบุตำแหน่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3 โฟร์ชาร์ทการทำงานในโหมดเคลื่อนที่แต่ละทิศเป็นอิสระต่อกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค
โปรแกรมควบคุมชุดขับเคลื่อนแท่นสแกน xy
บนระบบปฏิบัติการวินโดว์
(ภาษาวิชวลเบสิก)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Option Explicit
Dim a
Dim i
Dim st
Dim ipstr
Dim fso As New FileSystemObject
Dim fs As Scripting.TextStream
```

```
Sub CheckFreq()
    If Option1.Value = True Then a = "1"
    If Option2.Value = True Then a = "2"
    If Option3.Value = True Then a = "3"
    If Option4.Value = True Then a = "4"
    If Option5.Value = True Then a = "5"
End Sub
```

```
Private Sub cmdHome_Click()
    MSComm1.Output = "p"
End Sub
```

```
Private Sub cmdNorthEast_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    CheckFreq
    If direction = True Then
        MSComm1.Output = a
        MSComm1.Output = "b"
        MSComm1.Output = "j"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    End If
End Sub
```

```
Private Sub cmdNorthEast_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    MSComm1.Output = "p"
End Sub
```

```
Private Sub cmdSave_Click()
    If Val(Text1.Text) < 10 Then Text1.Text = "0" + Mid(Str(Val(Text1.Text)), 2, 1)
    If Len(Text1.Text) > 2 Then Text1.Text = Mid(Text1.Text, 2, 2)
    If Val(Text2.Text) < 10 Then Text2.Text = "0" + Mid(Str(Val(Text2.Text)), 2, 1)
    If Len(Text2.Text) > 2 Then Text2.Text = Mid(Text2.Text, 2, 2)

    Set fs = fso.OpenTextFile("save.txt", ForWriting, True)
    fs.Write Text1.Text + Text2.Text
    fs.Close
End Sub
```

```
Private Sub cmdSouth_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    CheckFreq
    If direction = True Then
        MSComm1.Output = a
        MSComm1.Output = "b"
        MSComm1.Output = "s"
    End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub
```

```
Private Sub cmdSouth_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
    MSComm1.Output = "p"
End Sub
```

```
Private Sub cmdGo_Click()
    CheckFreq
    If (position = True) And (Combo1.Text <> "") And (Combo2.Text <> "") Then
        MSComm1.Output = a
        MSComm1.Output = "a"
        MSComm1.Output = Combo1.Text
        MSComm1.Output = Combo2.Text
    End If
End Sub
```

```
Private Sub cmdSouthWest_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    CheckFreq
    If direction = True Then
        MSComm1.Output = a
        MSComm1.Output = "b"
        MSComm1.Output = "k"
    End If
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub cmdSouthWest_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As
Single)
   MSComm1.Output = "p"
End Sub
```

```
Private Sub cmdWest_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As
Single)
   CheckFreq
   If direction = True Then
      MSComm1.Output = a
      MSComm1.Output = "b"
      MSComm1.Output = "w"
   End If
End Sub
```

```
Private Sub cmdWest_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
   MSComm1.Output = "p"
End Sub
```

```
Private Sub cmdNorthWest_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As
Single)
   CheckFreq
   If direction = True Then
      MSComm1.Output = a
      MSComm1.Output = "b"
      MSComm1.Output = "h"
   End If
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub
```

```
Private Sub cmdNorthWest_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    MSComm1.Output = "p"
End Sub
```

```
Private Sub cmdSouthEast_mousedown(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    CheckFreq
    If direction = True Then
        MSComm1.Output = a
        MSComm1.Output = "b"
        MSComm1.Output = "l"
    End If
End Sub
```

```
Private Sub cmdSouthEast_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    MSComm1.Output = "p"
End Sub
```

```
Private Sub cmdEast_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    CheckFreq
    If direction = True Then
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        MSComm1.Output = a
        MSComm1.Output = "b"
        MSComm1.Output = "e"
    End If
End Sub
```

```
Private Sub cmdEast_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
    MSComm1.Output = "p"
End Sub
```

```
Private Sub cmdStop_Click()
    MSComm1.Output = "p"
End Sub
```

```
Private Sub cmdNorth_MouseDown(Button As Integer, Shift As Integer, X As Single, Y As
Single)
    CheckFreq
    If direction = True Then
        MSComm1.Output = a
        MSComm1.Output = "b"
        MSComm1.Output = "n"
    End If
End Sub
```

```
Private Sub cmdNorth_MouseUp(Button As Integer, Shift As Integer, X As Single, Y As Single)
    MSComm1.Output = "p"
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Sub
```

```
Private Sub Command4_Click()
    MSComm1.Output = "6"
End Sub
```

```
Private Sub Form_Load()
    MSComm1.Settings = "9600,N,8,1"
    MSComm1.CommPort = 1
    MSComm1.InputLen = 1
    MSComm1.PortOpen = True
    MSComm1.RThreshold = 1

    Set fs = fso.OpenTextFile("save.txt", ForReading, True)
    On Error GoTo er
    st = fs.Read(4)
    fs.Close

    Text1.Text = Mid(st, 1, 2)
    Text2.Text = Mid(st, 3, 2)

er:
End Sub
```

```
Private Sub MSComm1_OnComm()
    ipstr = MSComm1.Input
    Text1.Text = Mid(ipstr, 1, 2)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  Text2.Text = Mid(ipstr, 3, 2)
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ภาคผนวก ง
คู่มืออุปกรณ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Features

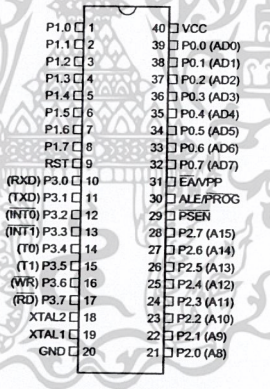
- Compatible with MCS-51™ Products
- 4K Bytes of In-System Reprogrammable Flash Memory
  - Endurance: 1,000 Write/Erase Cycles
- Fully Static Operation: 0 Hz to 24 MHz
- Three-level Program Memory Lock
- 128 x 8-bit Internal RAM
- 32 Programmable I/O Lines
- Two 16-bit Timer/Counters
- Six Interrupt Sources
- Programmable Serial Channel
- Low-power Idle and Power-down Modes

Description

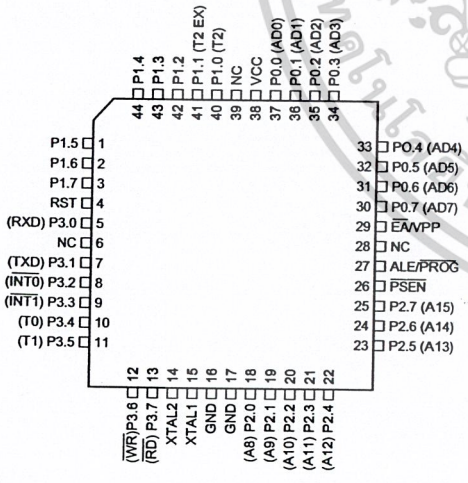
The AT89C51 is a low-power, high-performance CMOS 8-bit microcomputer with 4K bytes of Flash programmable and erasable read only memory (PEROM). The device is manufactured using Atmel's high-density nonvolatile memory technology and is compatible with the industry-standard MCS-51 instruction set and pinout. The on-chip Flash allows the program memory to be reprogrammed in-system or by a conventional nonvolatile memory programmer. By combining a versatile 8-bit CPU with Flash on a monolithic chip, the Atmel AT89C51 is a powerful microcomputer which provides a highly-flexible and cost-effective solution to many embedded control applications.

Pin Configurations

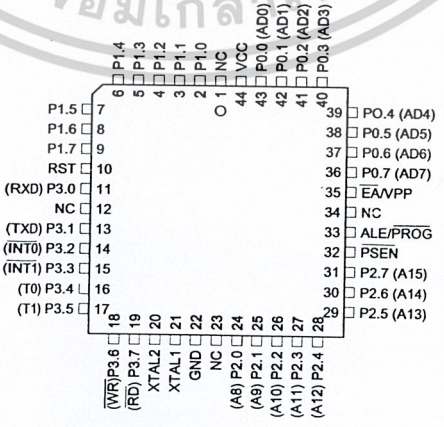
PDIP



PQFP/TQFP



PLCC



8-bit Microcontroller with 4K Bytes Flash

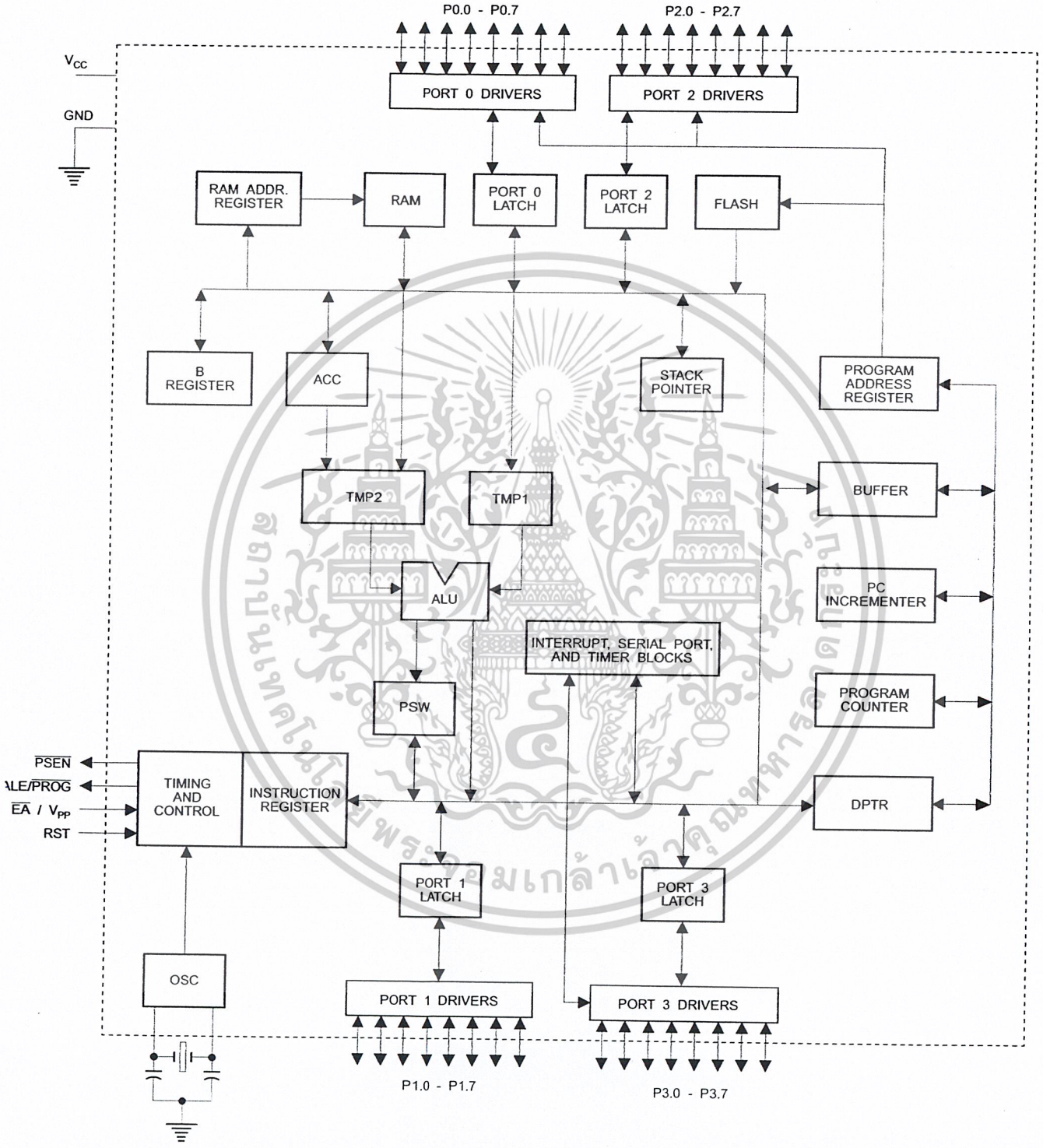
AT89C51

Not Recommended for New Designs. Use AT89S51.

Rev. 0265G–02/00

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Block Diagram



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The AT89C51 provides the following standard features: 4K bytes of Flash, 128 bytes of RAM, 32 I/O lines, two 16-bit timer/counters, a five vector two-level interrupt architecture, a full duplex serial port, on-chip oscillator and clock circuitry. In addition, the AT89C51 is designed with static logic for operation down to zero frequency and supports two software selectable power saving modes. The Idle Mode stops the CPU while allowing the RAM, timer/counters, serial port and interrupt system to continue functioning. The Power-down Mode saves the RAM contents but freezes the oscillator disabling all other chip functions until the next hardware reset.

Pin Description

VCC

Supply voltage.

GND

Ground.

Port 0

Port 0 is an 8-bit open-drain bi-directional I/O port. As an output port, each pin can sink eight TTL inputs. When 1s are written to port 0 pins, the pins can be used as high-impedance inputs.

Port 0 may also be configured to be the multiplexed low-order address/data bus during accesses to external program and data memory. In this mode P0 has internal pullups.

Port 0 also receives the code bytes during Flash programming, and outputs the code bytes during program verification. External pullups are required during program verification.

Port 1

Port 1 is an 8-bit bi-directional I/O port with internal pullups. The Port 1 output buffers can sink/source four TTL inputs. When 1s are written to Port 1 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 1 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 1 also receives the low-order address bytes during Flash programming and verification.

Port 2

Port 2 is an 8-bit bi-directional I/O port with internal pullups. The Port 2 output buffers can sink/source four TTL inputs. When 1s are written to Port 2 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 2 pins that are externally being pulled low will source current (I_{IL}) because of the internal pullups.

Port 2 emits the high-order address byte during fetches from external program memory and during accesses to external data memory that use 16-bit addresses (MOVX @ DPTR). In this application, it uses strong internal pullups when emitting 1s. During accesses to external data memory that use 8-bit addresses (MOVX @ RI), Port 2 emits the contents of the P2 Special Function Register.

Port 2 also receives the high-order address bits and some control signals during Flash programming and verification.

Port 3

Port 3 is an 8-bit bi-directional I/O port with internal pullups. The Port 3 output buffers can sink/source four TTL inputs. When 1s are written to Port 3 pins they are pulled high by the internal pullups and can be used as inputs. As inputs, Port 3 pins that are externally being pulled low will source current (I_{IL}) because of the pullups.

Port 3 also serves the functions of various special features of the AT89C51 as listed below:

Port Pin	Alternate Functions
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt 0)
P3.3	$\overline{INT1}$ (external interrupt 1)
P3.4	T0 (timer 0 external input)
P3.5	T1 (timer 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

Port 3 also receives some control signals for Flash programming and verification.

RST

Reset input. A high on this pin for two machine cycles while the oscillator is running resets the device.

ALE/$\overline{PROG}$

Address Latch Enable output pulse for latching the low byte of the address during accesses to external memory. This pin is also the program pulse input ($\overline{PROG}$) during Flash programming.

In normal operation ALE is emitted at a constant rate of 1/6 the oscillator frequency, and may be used for external timing or clocking purposes. Note, however, that one ALE

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

pulse is skipped during each access to external Data Memory.

If desired, ALE operation can be disabled by setting bit 0 of SFR location 8EH. With the bit set, ALE is active only during a MOVX or MOVC instruction. Otherwise, the pin is weakly pulled high. Setting the ALE-disable bit has no effect if the microcontroller is in external execution mode.

$\overline{\text{PSEN}}$

Program Store Enable is the read strobe to external program memory.

When the AT89C51 is executing code from external program memory, $\overline{\text{PSEN}}$ is activated twice each machine cycle, except that two $\overline{\text{PSEN}}$ activations are skipped during each access to external data memory.

$\overline{\text{EA}}$/VPP

External Access Enable. $\overline{\text{EA}}$ must be strapped to GND in order to enable the device to fetch code from external program memory locations starting at 0000H up to FFFFH. Note, however, that if lock bit 1 is programmed, $\overline{\text{EA}}$ will be internally latched on reset.

$\overline{\text{EA}}$ should be strapped to V_{CC} for internal program executions.

This pin also receives the 12-volt programming enable voltage (V_{PP}) during Flash programming, for parts that require 12-volt V_{PP}.

XTAL1

Input to the inverting oscillator amplifier and input to the internal clock operating circuit.

XTAL2

Output from the inverting oscillator amplifier.

Oscillator Characteristics

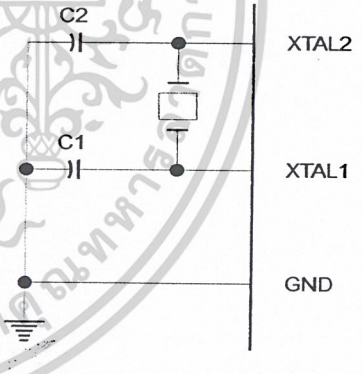
XTAL1 and XTAL2 are the input and output, respectively, of an inverting amplifier which can be configured for use as an on-chip oscillator, as shown in Figure 1. Either a quartz crystal or ceramic resonator may be used. To drive the device from an external clock source, XTAL2 should be left unconnected while XTAL1 is driven as shown in Figure 2. There are no requirements on the duty cycle of the external clock signal, since the input to the internal clocking circuitry is through a divide-by-two flip-flop, but minimum and maximum voltage high and low time specifications must be observed.

Idle Mode

In idle mode, the CPU puts itself to sleep while all the on-chip peripherals remain active. The mode is invoked by software. The content of the on-chip RAM and all the special functions registers remain unchanged during this mode. The idle mode can be terminated by any enabled interrupt or by a hardware reset.

It should be noted that when idle is terminated by a hard ware reset, the device normally resumes program execution, from where it left off, up to two machine cycles before the internal reset algorithm takes control. On-chip hardware inhibits access to internal RAM in this event, but access to the port pins is not inhibited. To eliminate the possibility of an unexpected write to a port pin when Idle is terminated by reset, the instruction following the one that invokes Idle should not be one that writes to a port pin or to external memory.

Figure 1. Oscillator Connections



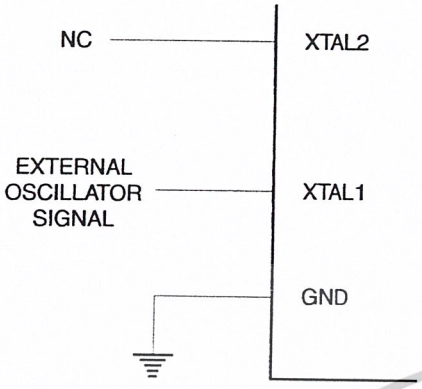
Note: C1, C2 = 30 pF ±10 pF for Crystals
= 40 pF ±10 pF for Ceramic Resonators

Status of External Pins During Idle and Power-down Modes

Mode	Program Memory	ALE	$\overline{\text{PSEN}}$	PORT0	PORT1	PORT2	PORT3
Idle	Internal	1	1	Data	Data	Data	Data
Idle	External	1	1	Float	Data	Address	Data
Power-down	Internal	0	0	Data	Data	Data	Data
Power-down	External	0	0	Float	Data	Data	Data

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Figure 2. External Clock Drive Configuration



Power-down Mode

In the power-down mode, the oscillator is stopped, and the instruction that invokes power-down is the last instruction executed. The on-chip RAM and Special Function Registers retain their values until the power-down mode is terminated. The only exit from power-down is a hardware reset. Reset redefines the SFRs but does not change the on-chip RAM. The reset should not be activated before V_{CC} is restored to its normal operating level and must be held active long enough to allow the oscillator to restart and stabilize.

Program Memory Lock Bits

On the chip are three lock bits which can be left unprogrammed (U) or can be programmed (P) to obtain the additional features listed in the table below.

When lock bit 1 is programmed, the logic level at the $\overline{EA}$ pin is sampled and latched during reset. If the device is powered up without a reset, the latch initializes to a random value, and holds that value until reset is activated. It is necessary that the latched value of $\overline{EA}$ be in agreement with the current logic level at that pin in order for the device to function properly.

Lock Bit Protection Modes

Program Lock Bits				
	LB1	LB2	LB3	Protection Type
1	U	U	U	No program lock features
2	P	U	U	MOVC instructions executed from external program memory are disabled from fetching code bytes from internal memory, $\overline{EA}$ is sampled and latched on reset, and further programming of the Flash is disabled
3	P	P	U	Same as mode 2, also verify is disabled
4	P	P	P	Same as mode 3, also external execution is disabled

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Programming the Flash

The AT89C51 is normally shipped with the on-chip Flash memory array in the erased state (that is, contents = FFH) and ready to be programmed. The programming interface accepts either a high-voltage (12-volt) or a low-voltage (V_{CC}) program enable signal. The low-voltage programming mode provides a convenient way to program the AT89C51 inside the user's system, while the high-voltage programming mode is compatible with conventional third-party Flash or EPROM programmers.

The AT89C51 is shipped with either the high-voltage or low-voltage programming mode enabled. The respective top-side marking and device signature codes are listed in the following table.

	V_{PP} = 12V	V_{PP} = 5V
Top-side Mark	AT89C51 xxxx yyww	AT89C51 xxxx-5 yyww
Signature	(030H) = 1EH (031H) = 51H (032H) =F FH	(030H) = 1EH (031H) = 51H (032H) = 05H

The AT89C51 code memory array is programmed byte-by-byte in either programming mode. *To program any non-blank byte in the on-chip Flash Memory, the entire memory must be erased using the Chip Erase Mode.*

Programming Algorithm: Before programming the AT89C51, the address, data and control signals should be set up according to the Flash programming mode table and Figure 3 and Figure 4. To program the AT89C51, take the following steps.

1. Input the desired memory location on the address lines.
2. Input the appropriate data byte on the data lines.
3. Activate the correct combination of control signals.
4. Raise $\overline{EA}/V_{PP}$ to 12V for the high-voltage programming mode.
5. Pulse ALE/$\overline{PROG}$ once to program a byte in the Flash array or the lock bits. The byte-write cycle is self-timed and typically takes no more than 1.5 ms. Repeat steps 1 through 5, changing the address and data for the entire array or until the end of the object file is reached.

Data Polling: The AT89C51 features $\overline{Data}$ Polling to indicate the end of a write cycle. During a write cycle, an attempted read of the last byte written will result in the complement of the written datum on PO.7. Once the write cycle has been completed, true data are valid on all outputs, and the next cycle may begin. $\overline{Data}$ Polling may begin any time after a write cycle has been initiated.

Ready/$\overline{Busy}$: The progress of byte programming can also be monitored by the RDY/$\overline{BSY}$ output signal. P3.4 is pulled low after ALE goes high during programming to indicate BUSY. P3.4 is pulled high again when programming is done to indicate READY.

Program Verify: If lock bits LB1 and LB2 have not been programmed, the programmed code data can be read back via the address and data lines for verification. The lock bits cannot be verified directly. Verification of the lock bits is achieved by observing that their features are enabled.

Chip Erase: The entire Flash array is erased electrically by using the proper combination of control signals and by holding ALE/$\overline{PROG}$ low for 10 ms. The code array is written with all "1"s. The chip erase operation must be executed before the code memory can be re-programmed.

Reading the Signature Bytes: The signature bytes are read by the same procedure as a normal verification of locations 030H, 031H, and 032H, except that P3.6 and P3.7 must be pulled to a logic low. The values returned are as follows.

(030H) = 1EH indicates manufactured by Atmel
(031H) = 51H indicates 89C51
(032H) = FFH indicates 12V programming
(032H) = 05H indicates 5V programming

Programming Interface

Every code byte in the Flash array can be written and the entire array can be erased by using the appropriate combination of control signals. The write operation cycle is self-timed and once initiated, will automatically time itself to completion.

All major programming vendors offer worldwide support for the Atmel microcontroller series. Please contact your local programming vendor for the appropriate software revision.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash Programming Modes

Mode		RST	PSEN	ALE/PROG	EA/V_{PP}	P2.6	P2.7	P3.6	P3.7
Write Code Data		H	L		H/12V	L	H	H	H
Read Code Data		H	L	H	H	L	L	H	H
Write Lock	Bit - 1	H	L		H/12V	H	H	H	H
	Bit - 2	H	L		H/12V	H	H	L	L
	Bit - 3	H	L		H/12V	H	L	H	L
Chip Erase		H	L	(1)	H/12V	H	L	L	L
Read Signature Byte		H	L	H	H	L	L	L	L

Note: 1. Chip Erase requires a 10 ms PROG pulse.

Figure 3. Programming the Flash

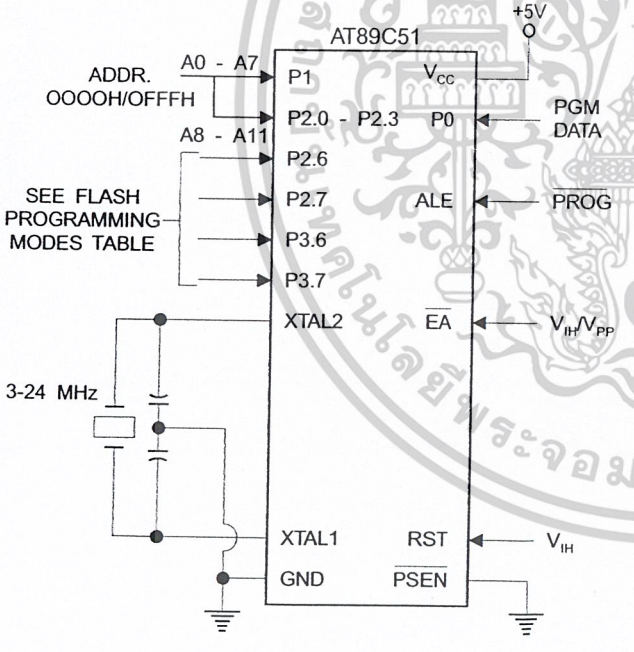
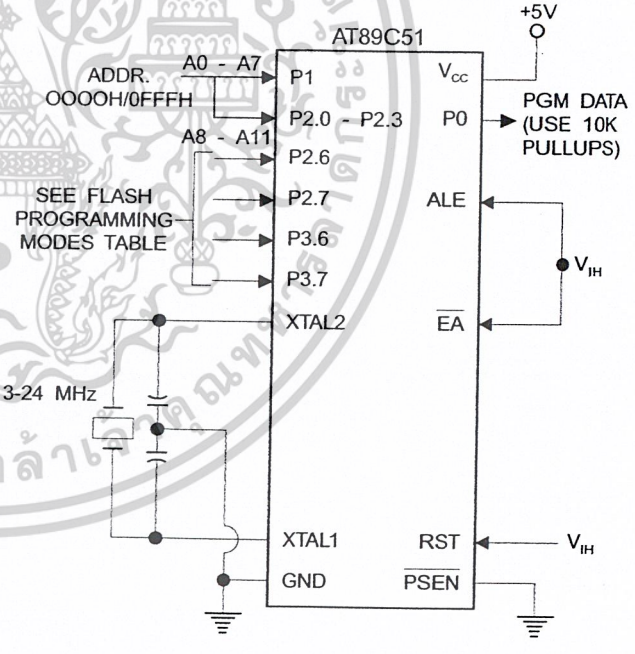
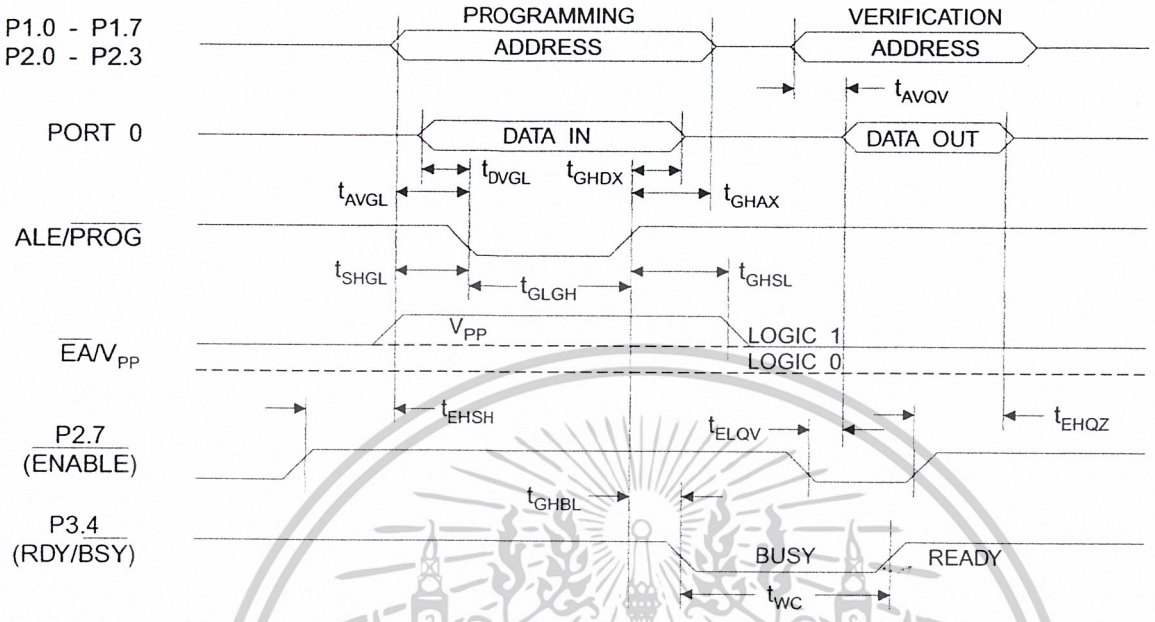


Figure 4. Verifying the Flash

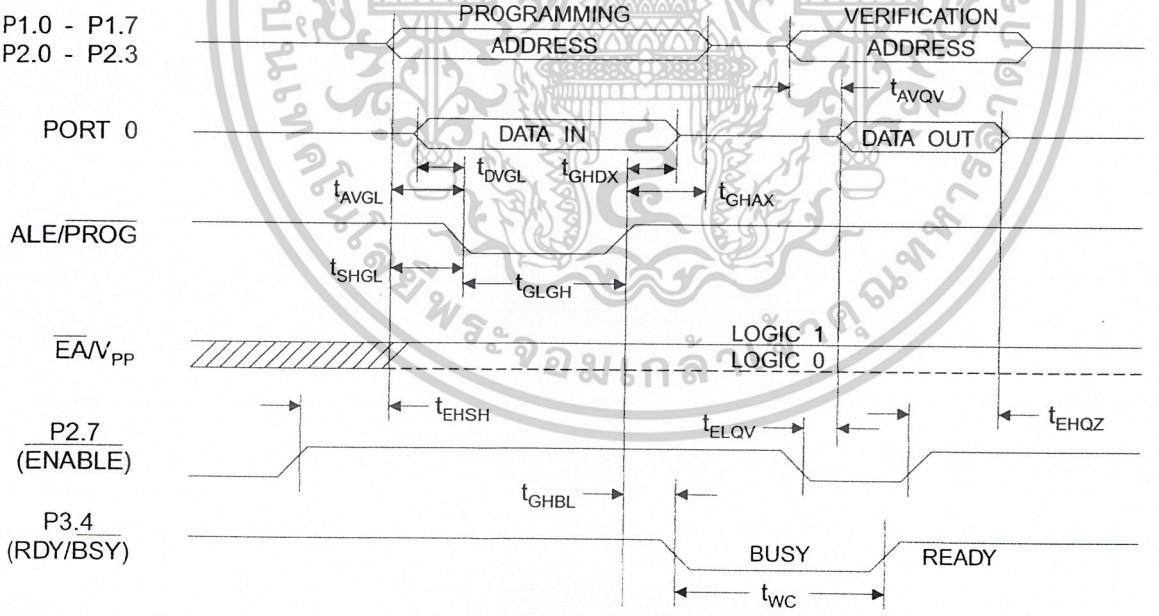


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash Programming and Verification Waveforms - High-voltage Mode (V_{PP} = 12V)



Flash Programming and Verification Waveforms - Low-voltage Mode (V_{PP} = 5V)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Flash Programming and Verification Characteristics

T_A = 0°C to 70°C, V_{CC} = 5.0 ± 10%

Symbol	Parameter	Min	Max	Units
V_{PP}(1)	Programming Enable Voltage	11.5	12.5	V
I_{PP}(1)	Programming Enable Current		1.0	mA
$1/t_{CLCL}$	Oscillator Frequency	3	24	MHz
t_{AVGL}	Address Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHAX}	Address Hold after $\overline{PROG}$	$48t_{CLCL}$		
t_{DVGL}	Data Setup to $\overline{PROG}$ Low	$48t_{CLCL}$		
t_{GHDX}	Data Hold after $\overline{PROG}$	$48t_{CLCL}$		
t_{EHSH}	P2.7 ($\overline{ENABLE}$) High to V_{PP}	$48t_{CLCL}$		
t_{SHGL}	V_{PP} Setup to $\overline{PROG}$ Low	10		µs
t_{GHSL}(1)	V_{PP} Hold after $\overline{PROG}$	10		µs
t_{GLGH}	$\overline{PROG}$ Width	1	110	µs
t_{AVQV}	Address to Data Valid		$48t_{CLCL}$	
t_{ELQV}	$\overline{ENABLE}$ Low to Data Valid		$48t_{CLCL}$	
t_{EHQZ}	Data Float after $\overline{ENABLE}$	0	$48t_{CLCL}$	
t_{GHBL}	$\overline{PROG}$ High to $\overline{BUSY}$ Low		1.0	µs
t_{WC}	Byte Write Cycle Time		2.0	ms

Note: 1. Only used in 12-volt programming mode.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings*

Operating Temperature	-55°C to +125°C
Storage Temperature	-65°C to +150°C
Voltage on Any Pin with Respect to Ground	-1.0V to +7.0V
Maximum Operating Voltage	6.6V
DC Output Current	15.0 mA

*NOTICE: Stresses beyond those listed under "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress rating only and functional operation of the device at these or any other conditions beyond those indicated in the operational sections of this specification is not implied. Exposure to absolute maximum rating conditions for extended periods may affect device reliability.

DC Characteristics

T_A = -40°C to 85°C, V_{CC} = 5.0V ±20% (unless otherwise noted)

Symbol	Parameter	Condition	Min	Max	Units
V_{IL}	Input Low-voltage	(Except $\overline{EA}$)	-0.5	0.2 V_{CC} - 0.1	V
V_{IL1}	Input Low-voltage ($\overline{EA}$)		-0.5	0.2 V_{CC} - 0.3	V
V_{IH}	Input High-voltage	(Except XTAL1, RST)	0.2 V_{CC} + 0.9	V_{CC} + 0.5	V
V_{IH1}	Input High-voltage	(XTAL1, RST)	0.7 V_{CC}	V_{CC} + 0.5	V
V_{OL}	Output Low-voltage[1] (Ports 1,2,3)	I_{OL} = 1.6 mA		0.45	V
V_{OL1}	Output Low-voltage[1] (Port 0, ALE, $\overline{PSEN}$)	I_{OL} = 3.2 mA		0.45	V
V_{OH}	Output High-voltage (Ports 1,2,3, ALE, $\overline{PSEN}$)	I_{OH} = -60 µA, V_{CC} = 5V ±10%	2.4		V
		I_{OH} = -25 µA	0.75 V_{CC}		V
		I_{OH} = -10 µA	0.9 V_{CC}		V
V_{OH1}	Output High-voltage (Port 0 in External Bus Mode)	I_{OH} = -800 µA, V_{CC} = 5V ±10%	2.4		V
		I_{OH} = -300 µA	0.75 V_{CC}		V
		I_{OH} = -80 µA	0.9 V_{CC}		V
I_{IL}	Logical 0 Input Current (Ports 1,2,3)	V_{IN} = 0.45V		-50	µA
I_{TL}	Logical 1 to 0 Transition Current (Ports 1,2,3)	V_{IN} = 2V, VCC = 5V ±10%		-650	µA
I_{LI}	Input Leakage Current (Port 0, $\overline{EA}$)	0.45 < V_{IN} < V_{CC}		±10	µA
RRST	Reset Pull-down Resistor		50	300	KΩ
C_{IO}	Pin Capacitance	Test Freq. = 1 MHz, T_A = 25°C		10	pF
I_{CC}	Power Supply Current	Active Mode, 12 MHz		20	mA
		Idle Mode, 12 MHz		5	mA
	Power-down Mode[2]	V_{CC} = 6V		100	µA
		V_{CC} = 3V		40	µA

Notes:
1. Under steady state (non-transient) conditions, I_{OL} must be externally limited as follows:
   Maximum I_{OL} per port pin: 10 mA
   Maximum I_{OL} per 8-bit port: Port 0: 26 mA
   Ports 1, 2, 3: 15 mA
   Maximum total I_{OL} for all output pins: 71 mA
   If I_{OL} exceeds the test condition, V_{OL} may exceed the related specification. Pins are not guaranteed to sink current greater than the listed test conditions.
2. Minimum V_{CC} for Power-down is 2V.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AC Characteristics

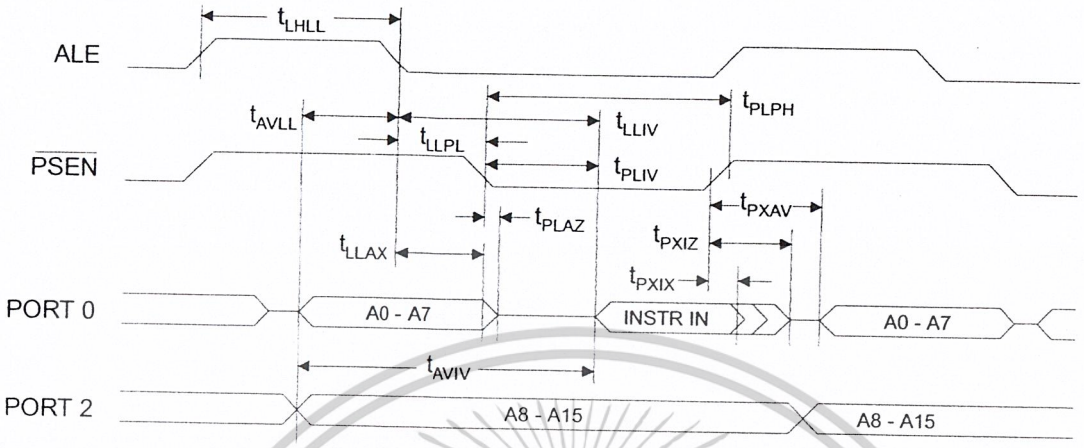
Under operating conditions, load capacitance for Port 0, ALE/$\overline{PROG}$, and $\overline{PSEN}$ = 100 pF; load capacitance for all other outputs = 80 pF.

External Program and Data Memory Characteristics

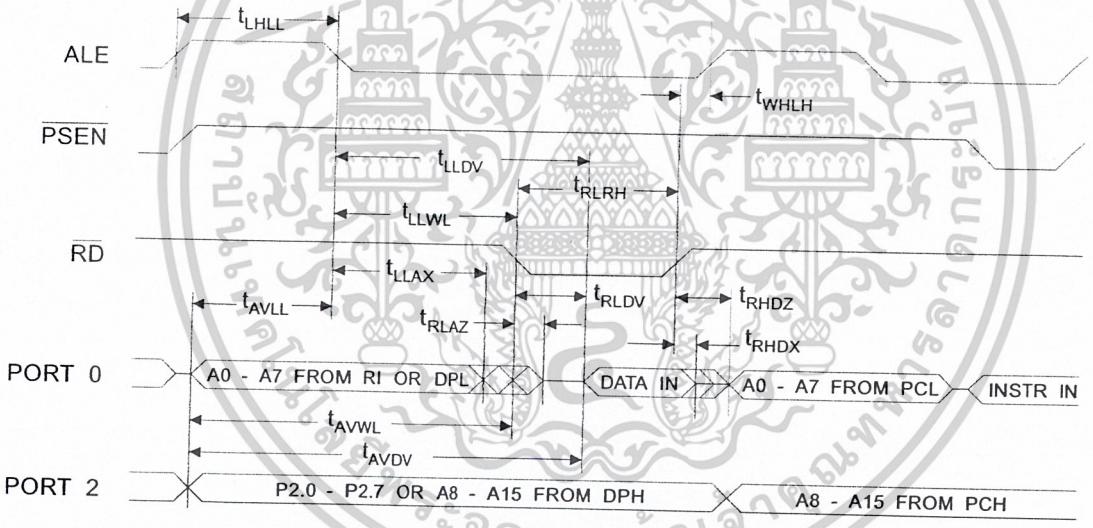
Symbol	Parameter	12 MHz Oscillator		16 to 24 MHz Oscillator		Units
		Min	Max	Min	Max	
$1/t_{CLCL}$	Oscillator Frequency			0	24	MHz
t_{LHLL}	ALE Pulse Width	127		$2t_{CLCL}$-40		ns
t_{AVLL}	Address Valid to ALE Low	43		t_{CLCL}-13		ns
t_{LLAX}	Address Hold after ALE Low	48		t_{CLCL}-20		ns
t_{LLIV}	ALE Low to Valid Instruction In		233		$4t_{CLCL}$-65	ns
t_{LLPL}	ALE Low to $\overline{PSEN}$ Low	43		t_{CLCL}-13		ns
t_{PLPH}	$\overline{PSEN}$ Pulse Width	205		$3t_{CLCL}$-20		ns
t_{PLIV}	$\overline{PSEN}$ Low to Valid Instruction In		145		$3t_{CLCL}$-45	ns
t_{PXIX}	Input Instruction Hold after $\overline{PSEN}$	0		0		ns
t_{PXIZ}	Input Instruction Float after $\overline{PSEN}$		59		t_{CLCL}-10	ns
t_{PXAV}	$\overline{PSEN}$ to Address Valid	75		t_{CLCL}-8		ns
t_{AVIV}	Address to Valid Instruction In		312		$5t_{CLCL}$-55	ns
t_{PLAZ}	$\overline{PSEN}$ Low to Address Float		10		10	ns
t_{RLRH}	$\overline{RD}$ Pulse Width	400		$6t_{CLCL}$-100		ns
t_{WLWH}	$\overline{WR}$ Pulse Width	400		$6t_{CLCL}$-100		ns
t_{RLDV}	$\overline{RD}$ Low to Valid Data In		252		$5t_{CLCL}$-90	ns
t_{RHDX}	Data Hold after $\overline{RD}$	0		0		ns
t_{RHDZ}	Data Float after $\overline{RD}$		97		$2t_{CLCL}$-28	ns
t_{LLDV}	ALE Low to Valid Data In		517		$8t_{CLCL}$-150	ns
t_{AVDV}	Address to Valid Data In		585		$9t_{CLCL}$-165	ns
t_{LLWL}	ALE Low to $\overline{RD}$ or $\overline{WR}$ Low	200	300	$3t_{CLCL}$-50	$3t_{CLCL}$+50	ns
t_{AVWL}	Address to $\overline{RD}$ or $\overline{WR}$ Low	203		$4t_{CLCL}$-75		ns
t_{QVWX}	Data Valid to $\overline{WR}$ Transition	23		t_{CLCL}-20		ns
t_{QVWH}	Data Valid to $\overline{WR}$ High	433		$7t_{CLCL}$-120		ns
t_{WHQX}	Data Hold after $\overline{WR}$	33		t_{CLCL}-20		ns
t_{RLAZ}	$\overline{RD}$ Low to Address Float		0		0	ns
t_{WHLH}	$\overline{RD}$ or $\overline{WR}$ High to ALE High	43	123	t_{CLCL}-20	t_{CLCL}+25	ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Program Memory Read Cycle

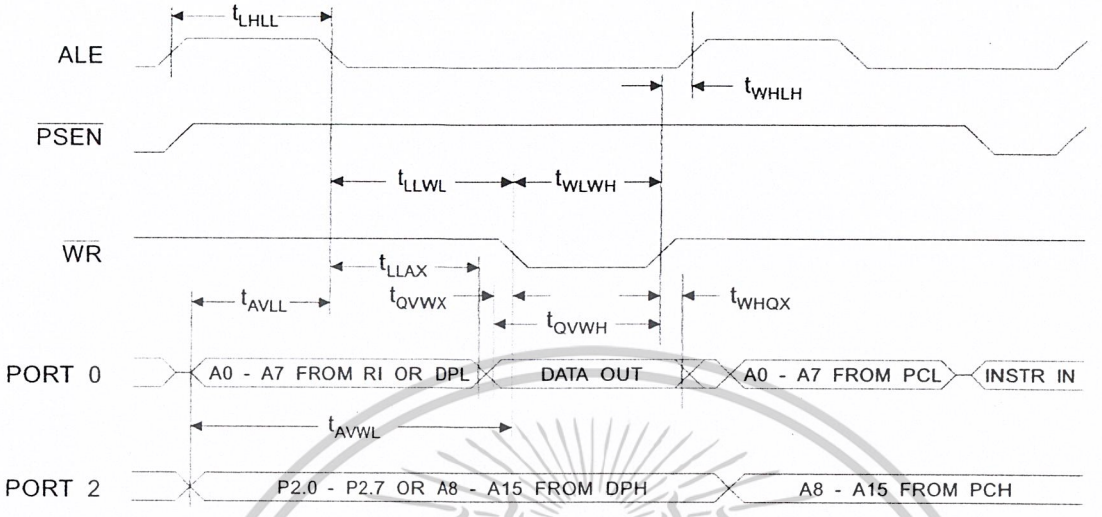


External Data Memory Read Cycle

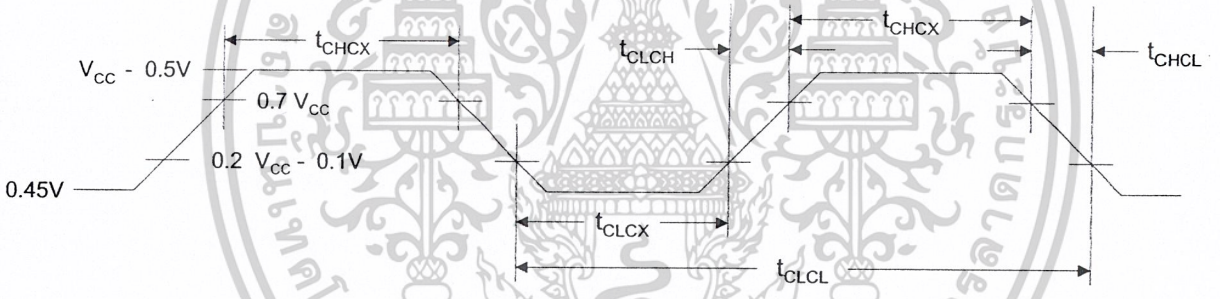


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

External Data Memory Write Cycle



External Clock Drive Waveforms



External Clock Drive

Symbol	Parameter	Min	Max	Units
$1/t_{CLCL}$	Oscillator Frequency	0	24	MHz
t_{CLCL}	Clock Period	41.6		ns
t_{CHCX}	High Time	15		ns
t_{CLCX}	Low Time	15		ns
t_{CLCH}	Rise Time		20	ns
t_{CHCL}	Fall Time		20	ns

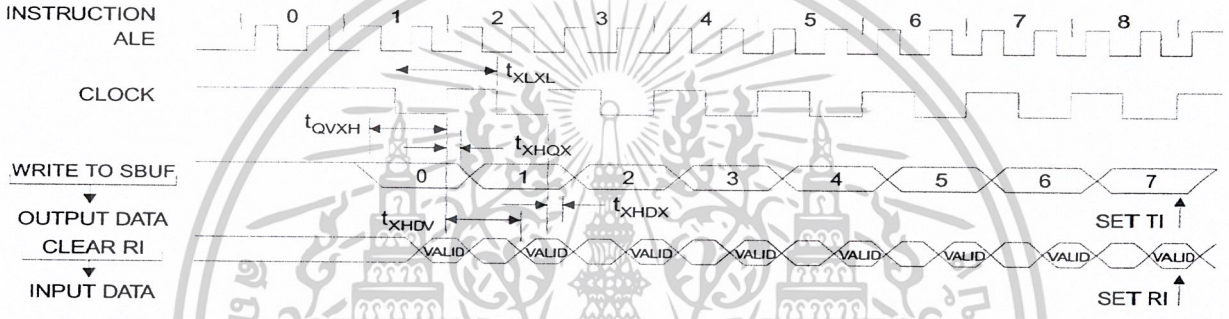
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Serial Port Timing: Shift Register Mode Test Conditions

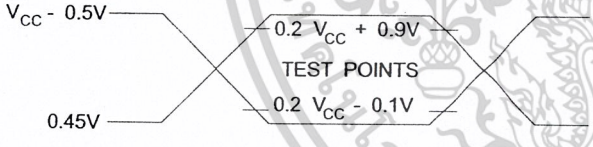
(V_{CC} = 5.0 V ±20%; Load Capacitance = 80 pF)

Symbol	Parameter	12 MHz Osc		Variable Oscillator		Units
		Min	Max	Min	Max	
t_{XLXL}	Serial Port Clock Cycle Time	1.0		$12t_{CLCL}$		µs
t_{QVXH}	Output Data Setup to Clock Rising Edge	700		$10t_{CLCL}$-133		ns
t_{XHQX}	Output Data Hold after Clock Rising Edge	50		$2t_{CLCL}$-117		ns
t_{XHDX}	Input Data Hold after Clock Rising Edge	0		0		ns
t_{XHDV}	Clock Rising Edge to Input Data Valid		700		$10t_{CLCL}$-133	ns

Shift Register Mode Timing Waveforms

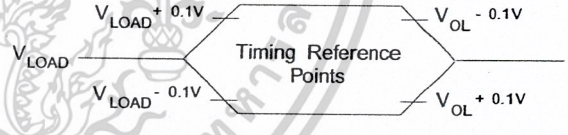


AC Testing Input/Output Waveforms[1]



Note: 1. AC Inputs during testing are driven at V_{CC} - 0.5V for a logic 1 and 0.45V for a logic 0. Timing measurements are made at V_{IH} min. for a logic 1 and V_{IL} max. for a logic 0.

Float Waveforms[1]



Note: 1. For timing purposes, a port pin is no longer floating when a 100 mV change from load voltage occurs. A port pin begins to float when 100 mV change from the loaded V_{OH}/V_{OL} level occurs.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Ordering Information

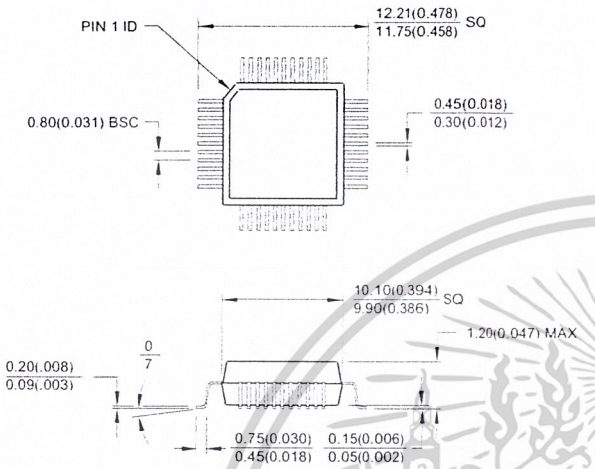
Speed (MHz)	Power Supply	Ordering Code	Package	Operation Range
12	5V ±20%	AT89C51-12AC	44A	Commercial (0° C to 70° C)
		AT89C51-12JC	44J	
		AT89C51-12PC	40P6	
		AT89C51-12QC	44Q	
		AT89C51-12AI	44A	Industrial (-40° C to 85° C)
		AT89C51-12JI	44J	
		AT89C51-12PI	40P6	
		AT89C51-12QI	44Q	
16	5V ±20%	AT89C51-16AC	44A	Commercial (0° C to 70° C)
		AT89C51-16JC	44J	
		AT89C51-16PC	40P6	
		AT89C51-16QC	44Q	
		AT89C51-16AI	44A	Industrial (-40° C to 85° C)
		AT89C51-16JI	44J	
		AT89C51-16PI	40P6	
		AT89C51-16QI	44Q	
20	5V ±20%	AT89C51-20AC	44A	Commercial (0° C to 70° C)
		AT89C51-20JC	44J	
		AT89C51-20PC	40P6	
		AT89C51-20QC	44Q	
		AT89C51-20AI	44A	Industrial (-40° C to 85° C)
		AT89C51-20JI	44J	
		AT89C51-20PI	40P6	
		AT89C51-20QI	44Q	
24	5V ±20%	AT89C51-24AC	44A	Commercial (0° C to 70° C)
		AT89C51-24JC	44J	
		AT89C51-24PC	40P6	
		AT89C51-24QC	44Q	
		AT89C51-24AI	44A	Industrial (-40° C to 85° C)
		AT89C51-24JI	44J	
		AT89C51-24PI	40P6	
		AT89C51-24QI	44Q	

Package Type	
44A	44-lead, Thin Plastic Gull Wing Quad Flatpack (TQFP)
44J	44-lead, Plastic J-leaded Chip Carrier (PLCC)
40P6	40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
44Q	44-lead, Plastic Gull Wing Quad Flatpack (PQFP)

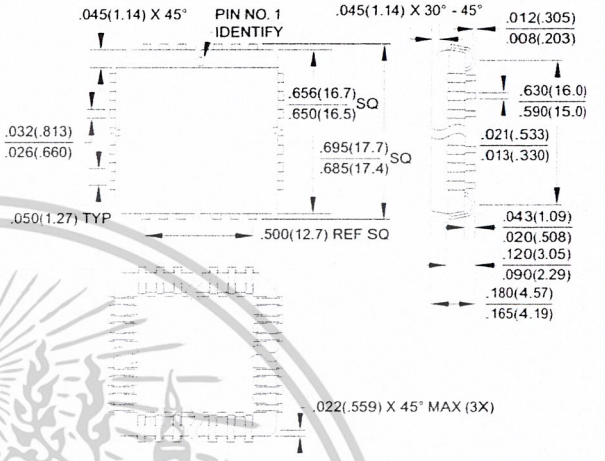
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Packaging Information

44A, 44-lead, Thin (1.0 mm) Plastic Gull Wing Quad Flatpack (TQFP)
Dimensions in Millimeters and (Inches)*
JEDEC STANDARD MS-026 ACB

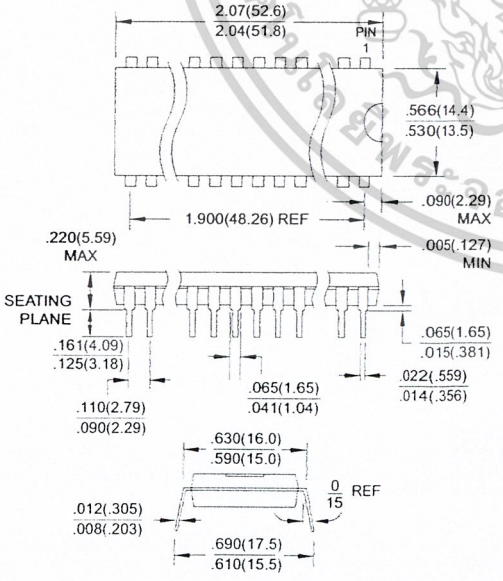


44J, 44-lead, Plastic J-leaded Chip Carrier (PLCC)
Dimensions in Inches and (Millimeters)
JEDEC STANDARD MS-018 AC

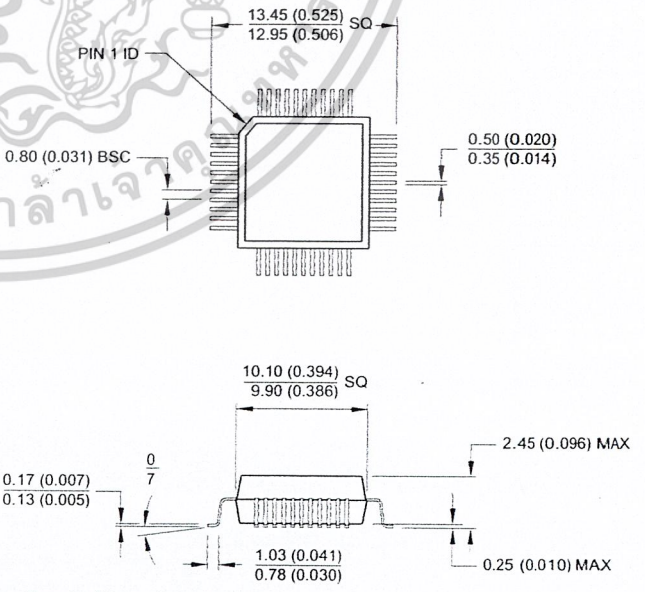


Controlling dimension: millimeters

40P6, 40-lead, 0.600" Wide, Plastic Dual Inline Package (PDIP)
Dimensions in Inches and (Millimeters)



44Q, 44-lead, Plastic Quad Flat Package (PQFP)
Dimensions in Millimeters and (Inches)*
JEDEC STANDARD MS-022 AB



Controlling dimension: millimeters

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Atmel Headquarters

Corporate Headquarters
2325 Orchard Parkway
San Jose, CA 95131
TEL (408) 441-0311
FAX (408) 487-2600

Europe
Atmel U.K., Ltd.
Coliseum Business Centre
Riverside Way
Camberley, Surrey GU15 3YL
England
TEL (44) 1276-686-677
FAX (44) 1276-686-697

Asia
Atmel Asia, Ltd.
Room 1219
Chinachem Golden Plaza
77 Mody Road Tsimhatsui
East Kowloon
Hong Kong
TEL (852) 2721-9778
FAX (852) 2722-1369

Japan
Atmel Japan K.K.
9F, Tonetsu Shinkawa Bldg.
1-24-8 Shinkawa
Chuo-ku, Tokyo 104-0033
Japan
TEL (81) 3-3523-3551
FAX (81) 3-3523-7581

Atmel Operations

Atmel Colorado Springs
1150 E. Cheyenne Mtn. Blvd.
Colorado Springs, CO 80906
TEL (719) 576-3300
FAX (719) 540-1759

Atmel Rousset
Zone Industrielle
13106 Rousset Cedex
France
TEL (33) 4-4253-6000
FAX (33) 4-4253-6001

Fax-on-Demand
North America:
1-(800) 292-8635
International:
1-(408) 441-0732

e-mail
literature@atmel.com

Web Site
http://www.atmel.com

BBS
1-(408) 436-4309

© Atmel Corporation 2000.
Atmel Corporation makes no warranty for the use of its products, other than those expressly contained in the Company's standard warranty which is detailed in Atmel's Terms and Conditions located on the Company's web site. The Company assumes no responsibility for any errors which may appear in this document, reserves the right to change devices or specifications detailed herein at any time without notice, and does not make any commitment to update the information contained herein. No licenses to patents or other intellectual property of Atmel are granted by the Company in connection with the sale of Atmel products, expressly or by implication. Atmel's products are not authorized for use as critical components in life support devices or systems.

Marks bearing ® and/or ™ are registered trademarks and trademarks of Atmel Corporation.

Terms and product names in this document may be trademarks of others.

Printed on recycled paper.

0265G–02/00/xM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MOTOROLA

SN54/74LS245

OCTAL BUS TRANSCEIVER

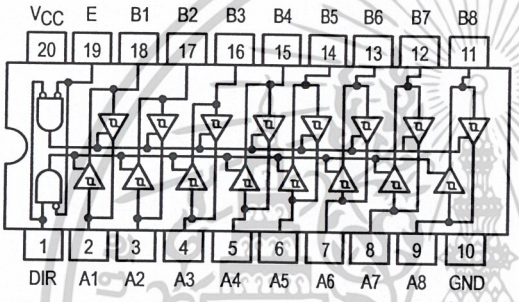
The SN54/74LS245 is an Octal Bus Transmitter/Receiver designed for 8-line asynchronous 2-way data communication between data buses. Direction Input (DR) controls transmission of Data from bus A to bus B or bus B to bus A depending upon its logic level. The Enable input (E) can be used to isolate the buses.

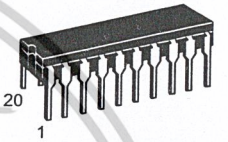
- Hysteresis Inputs to Improve Noise Immunity
- 2-Way Asynchronous Data Bus Communication
- Input Diodes Limit High-Speed Termination Effects
- ESD > 3500 Volts

OCTAL BUS TRANSCEIVER

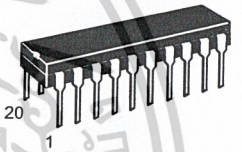
LOW POWER SCHOTTKY

LOGIC AND CONNECTION DIAGRAMS DIP (TOP VIEW)

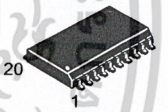




J SUFFIX
CERAMIC
CASE 732-03



N SUFFIX
PLASTIC
CASE 738-03



DW SUFFIX
SOIC
CASE 751D-03

TRUTH TABLE

INPUTS		OUTPUT
E	DIR	
L	L	Bus B Data to Bus A
L	H	Bus A Data to Bus B
H	X	Isolation

H = HIGH Voltage Level
L = LOW Voltage Level
X = Immaterial

ORDERING INFORMATION

SN54LSXXXJ	Ceramic
SN74LSXXXN	Plastic
SN74LSXXXDW	SOIC

GUARANTEED OPERATING RANGES

Symbol	Parameter		Min	Typ	Max	Unit
V_{CC}	Supply Voltage	54 74	4.5 4.75	5.0 5.0	5.5 5.25	V
T_A	Operating Ambient Temperature Range	54 74	−55 0	25 25	125 70	°C
I_{OH}	Output Current — High	54, 74			−3.0	mA
		54 74			−12 −15	mA
I_{OL}	Output Current — Low	54 74			12 24	mA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DC CHARACTERISTICS OVER OPERATING TEMPERATURE RANGE (unless otherwise specified)

Symbol	Parameter		Limits Min	Limits Typ	Limits Max	Unit	Test Conditions	
V_{IH}	Input HIGH Voltage		2.0			V	Guaranteed Input HIGH Voltage for All Inputs	
V_{IL}	Input LOW Voltage	54			0.7	V	Guaranteed Input LOW Voltage for All Inputs	
		74			0.8			
$V_{T+}-V_{T-}$	Hysteresis		0.2	0.4		V	V_{CC} = MIN	
V_{IK}	Input Clamp Diode Voltage			−0.65	−1.5	V	V_{CC} = MIN, I_{IN} = −18 mA	
V_{OH}	Output HIGH Voltage	54, 74	2.4	3.4		V	V_{CC} = MIN, I_{OH} = −3.0 mA	
		54, 74	2.0			V	V_{CC} = MIN, I_{OH} = MAX	
V_{OL}	Output LOW Voltage	54, 74		0.25	0.4	V	I_{OL} = 12 mA	V_{CC} = V_{CC} MIN, V_{IN} = V_{IL} or V_{IH} per Truth Table
		74		0.35	0.5	V	I_{OL} = 24 mA	
I_{OZH}	Output Off Current HIGH				20	μA	V_{CC} = MAX, V_{OUT} = 2.7 V	
I_{OZL}	Output Off Current LOW				−200	μA	V_{CC} = MAX, V_{OUT} = 0.4 V	
I_{IH}	Input HIGH Current	A or B, DR or E			20	μA	V_{CC} = MAX, V_{IN} = 2.7 V	
		DR or E			0.1	mA	V_{CC} = MAX, V_{IN} = 7.0 V	
		A or B			0.1	mA	V_{CC} = MAX, V_{IN} = 5.5 V	
I_{IL}	Input LOW Current				−0.2	mA	V_{CC} = MAX, V_{IN} = 0.4 V	
I_{OS}	Output Short Circuit Current (Note 1)		−40		−225	mA	V_{CC} = MAX	
I_{CC}	Power Supply Current Total, Output HIGH				70	mA	V_{CC} = MAX	
	Total, Output LOW				90			
	Total at HIGH Z				95			

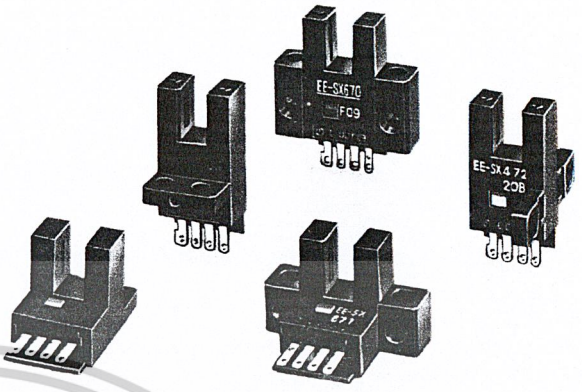
Note 1: Not more than one output should be shorted at a time, nor for more than 1 second.

AC CHARACTERISTICS (T_A = 25°C, V_{CC} = 5.0 V, $T_{RISE}/T_{FALL} \leq$ 6.0 ns)

Symbol	Parameter	Limits Min	Limits Typ	Limits Max	Unit	Test Conditions
t_{PLH} t_{PHL}	Propagation Delay, Data to Output		8.0 8.0	12 12	ns	C_L = 45 pF, R_L = 667 Ω
t_{PZH}	Output Enable Time to HIGH Level		25	40	ns	
t_{PZL}	Output Enable Time to LOW Level		27	40	ns	
t_{PLZ}	Output Disable Time from LOW Level		15	25	ns	C_L = 5.0 pF, R_L = 667 Ω
t_{PHZ}	Output Disable Time from HIGH Level		15	25	ns	

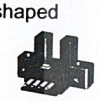
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Photomicrosensor with 100-mA Switching Capacity that can be Built into Equipment



- Standard, L-shaped, T-shaped, and close mounting: 5-series models available.
- Select from fifteen output variations, including Light-ON or Light-OFF/ON models.
- Response frequency as high as 1 kHz.
- Easy operation monitoring with bright light indicator.
- Wide operating voltage range (5 to 24 VDC) makes smooth connection of the photomicrosensor with TTLs, relays, and programmable controllers (PC) possible.
- Dust-proof slit.
- EE-SX670A/671A/672A/673A/674A models come with a Light-OFF indicator, which lights when an object is detected.

Ordering Information

Appearance	Sensing method	Sensing distance	Output configuration	Model	Weight
Standard 	Transmissive type (channel-type)	5 mm (channel width)	Light-OFF/ON (see note 1)	*EE-SX670	Approx. 3.1 g
				*EE-SX670A (see note 2)	
			Light-ON	*EE-SX470	
L-shaped 			Light-OFF/ON (see note 1)	*EE-SX671	Approx. 3.0 g
				*EE-SX671A (see note 2)	
			Light-ON	*EE-SX471	
T-shaped 			Light-OFF/ON (see note 1)	*EE-SX672	Approx. 2.4 g
				*EE-SX672A (see note 2)	
			Light-ON	*EE-SX472	
Close-mounting 			Light-OFF/ON (see note 1)	*EE-SX673	Approx. 2.3 g
				*EE-SX673A (see note 2)	
			Light-ON	*EE-SX473	
Close-mounting NEW 			Light-OFF/ON (see note 1)	EE-SX674	Approx. 3.0 g
				EE-SX674A (see note 2)	
			Light-ON	EE-SX474	

Note: 1. The Light-OFF/ON models can be used as Light-ON models when the L terminal and positive (+) terminal are short-circuited. A L terminal and positive (+) terminal short circuit Connector (EE-1001-1) is available.

2. These models are fitted with a LIGHT OFF indicator light.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

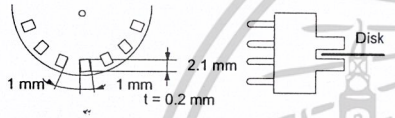
Specifications

■ Ratings

Item	Standard	L-shaped	T-shaped	Close-mounting
	EE-SX670(A), EE-SX470	EE-SX671(A), EE-SX471	EE-SX672(A), EE-SX472	EE-SX673(A), EE-SX473, EE-SX674(A), EE-SX474
Supply voltage	5 to 24 VDC ±10%, ripple (p-p): 10% max.			
Current consumption	35 mA max.			
Standard reference object	Opaque: 2 × 0.8 mm			
Sensing distance	5 mm			
Differentialdistance	0.025 mm			
Control output	At 5 to 24 VDC: 100-mA load current (I_C) with a residual voltage of 0.8 V max. 40-mA load current (I_C) with a residual voltage of 0.4 V max.			
Indicator (see note 1)	Light indicator (red)			
Response frequency (see note 2)	1 kHz max. (3 kHz average)			
Connecting method	Dedicated connectors: EE-1009/1006(A)/1010/1001-01/1001; direct soldering			
Light source	GaAs infrared LED with a peak wavelength of 940 nm			
Receiver	Si phototransistor with a sensing wavelength of 850 nm max.			

Note: 1. The indicator is GaP red LED (peak emission wavelength: 690 nm).

2. The response frequency was measured by detecting the following rotating disks.



■ Characteristics

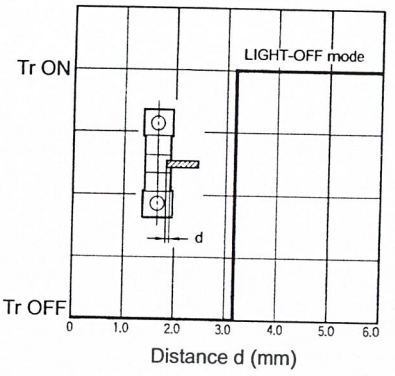
Ambient illumination (see note)	Fluorescent light: 1,000 ℓx max.
Enclosure ratings	IEC IP50
Ambient temperature	Operating: –25° to 55°C Storage: –30° to 80°C
Ambient humidity	Operating: 5% to 85% Storage: 5% to 95%
Vibration resistance	Destruction: 20 to 2,000 Hz, (with a peak acceleration of 10G), 1.5-mm double amplitude for 2 hrs (with 4-minute cycles) each in X, Y, and Z directions
Shock resistance	Destruction: 500 m/s^2 (approx. 50G) for 3 times each in X, Y, and Z directions
Soldering heat resistance	260±5°C when the portion between the tip of the terminals and the position 1.5 mm from the terminal base is dipped into the solder for 10±1 seconds
Material	Case: Polybutylene phthalate (PBT), Cover: Polycarbonate (PC), Emitter/receiver: Polycarbonate (PC)

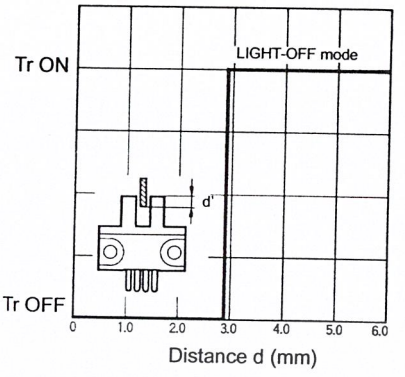
Note: The ambient illuminance is measured on the surface of the receiver.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

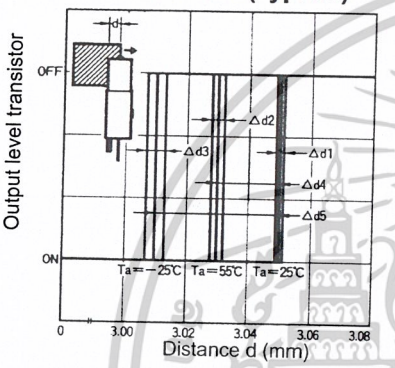
Engineering Data

Sensing Position Characteristics (Typical)





Repeated Sensing Position Characteristics (Typical)



V_{CC} = 12 V
No. of repetitions: 20
Δd1 = 0.002 mm
Δd2 = 0.004 mm
Δd3 = 0.005 mm
Δd4 = 0.02 mm
Δd5 = 0.04 mm

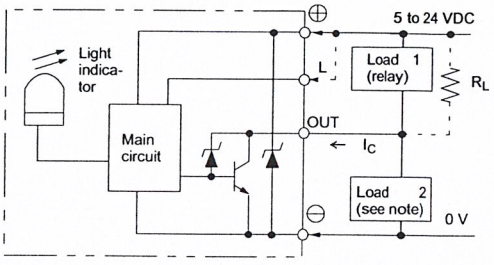
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Operation

■ Output Circuit Diagrams

EE-SX670, -SX671, -SX672, -SX673, -SX674

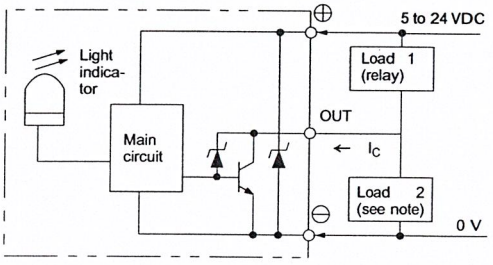
Light ON/OFF



Note: When using on voltage output, always insert a resistor in RL and use load 2.

EE-SX470, -SX471, -SX472, -SX473, -SX474

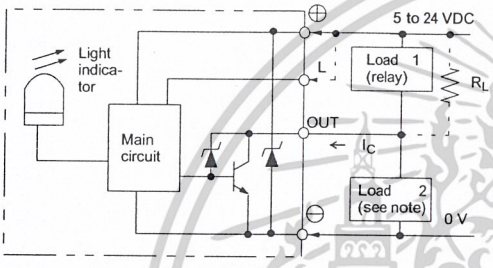
Light ON



Note: Voltage output (when the sensor is connected to a transistor circuit).

EE-SX670A, -SX671A, -SX672A, -SX673A, -SX674A

Light ON/OFF



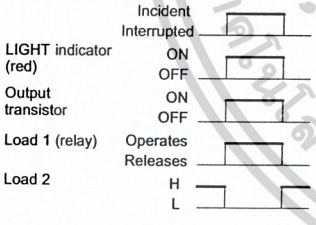
Note: When using on voltage output, always insert a resistor in RL and use load 2.

■ Timing Chart

EE-SX670, -SX671, -SX672, -SX673, -SX674

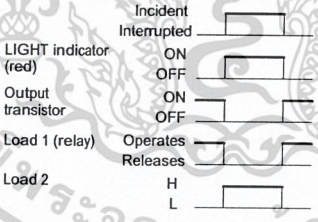
Light ON

(When terminals L and ⊕ are short-circuited)



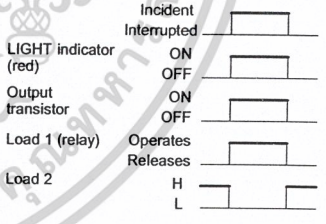
Light OFF

(When terminals L and ⊕ are open)



EE-SX470, -SX471, -SX472, -SX473, -SX474

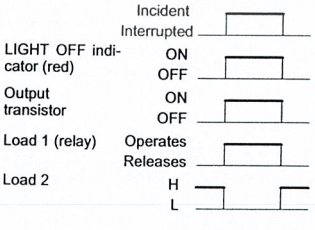
Light ON



EE-SX670A, -SX671A, -SX672A, -SX673A, -SX674A

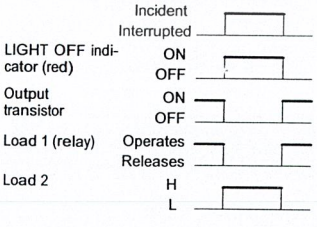
Light ON

(When terminals L and ⊕ are short-circuited)



Light OFF

(When terminals L and ⊕ are open)

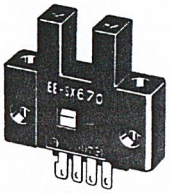


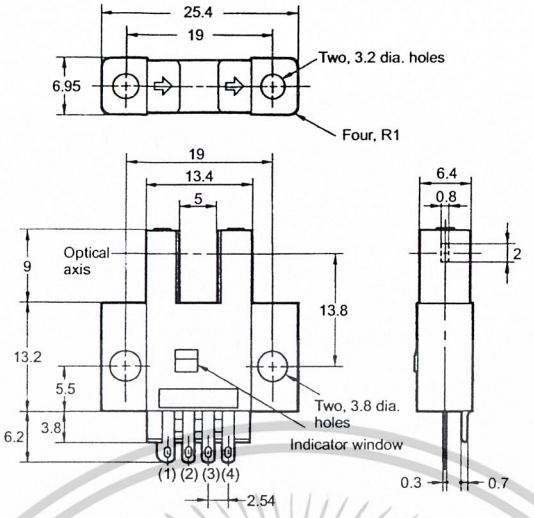
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Dimensions

Note: All units are in millimeters unless otherwise indicated.

EE-SX670
EE-SX670A
EE-SX470



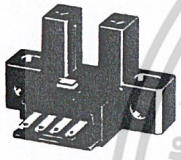


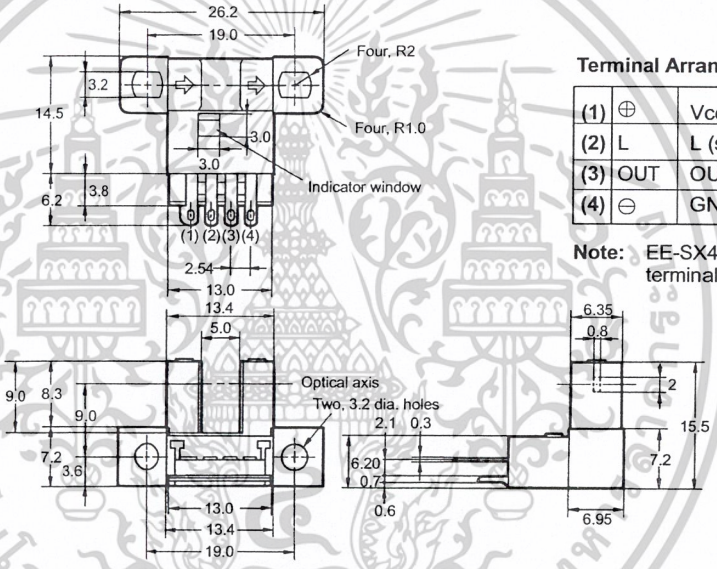
Terminal Arrangement

(1)	⊕	Vcc
(2)	L	L (see note)
(3)	OUT	OUT PUT
(4)	⊖	GND (0 V)

Note: EE-SX470 is a vacant terminal

EE-SX671
EE-SX671A
EE-SX471





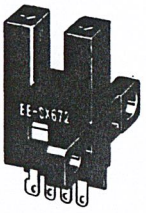
Terminal Arrangement

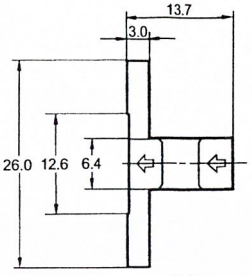
(1)	⊕	Vcc
(2)	L	L (see note)
(3)	OUT	OUT PUT
(4)	⊖	GND (0 V)

Note: EE-SX471 is a vacant terminal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EE-SX672
EE-SX672A
EE-SX472

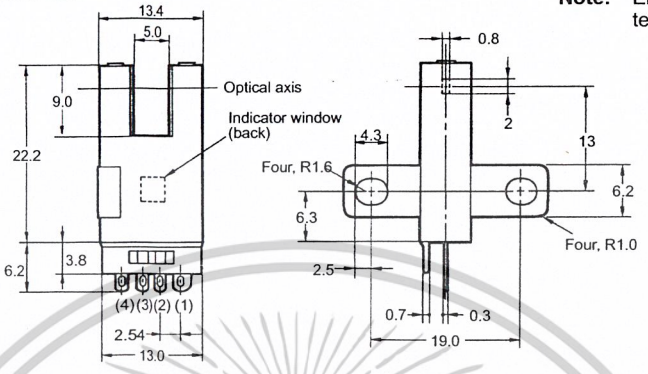




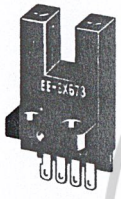
Terminal Arrangement

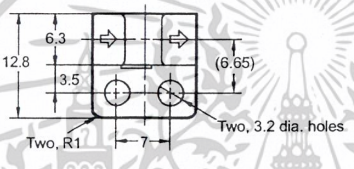
(1)	⊕	Vcc
(2)	L	L (see note)
(3)	OUT	OUT PUT
(4)	⊖	GND (0 V)

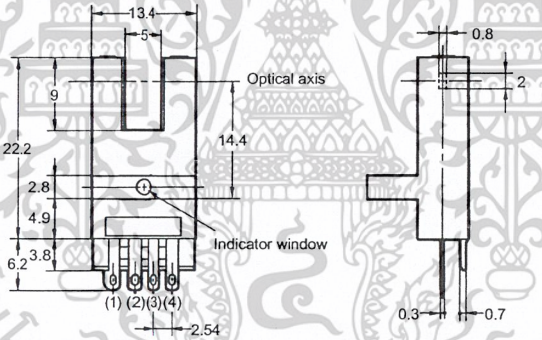
Note: EE-SX472 is a vacant terminal



EE-SX673
EE-SX673A
EE-SX473







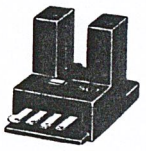
Terminal Arrangement

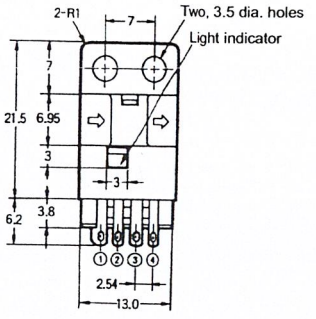
(1)	⊕	Vcc
(2)	L	L (see note)
(3)	OUT	OUT PUT
(4)	⊖	GND (0 V)

Note: EE-SX473 is a vacant terminal

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

EE-SX674
EE-SX674A
EE-SX474

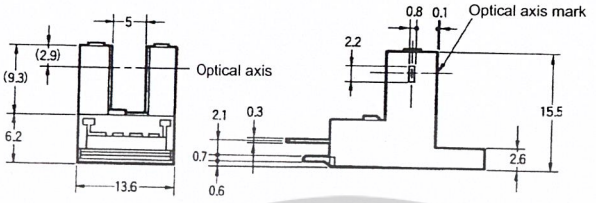




Terminal Arrangement

(1)	⊕	Vcc
(2)	L	L (see note)
(3)	OUT	OUT PUT
(4)	⊖	GND (0 V)

Note: EE-SX474 is a vacant terminal



Applicable Connectors

EE-1009, EE-1010, EE-1001-1, EE-1001, EE-1006, EE-1006A
Refer to page 98 for details.

Precautions

Refer to page 25, *Precautions* in *Technical Information*, for general precautions.

When direct soldering to the terminal, use the following guidelines.

Soldering Conditions

Item	Temprature	Permissible time	Remarks
Soldering Iron	350°C max.	3 sec max.	The portion between the base of the terminals and the position 1.5 mm from the terminal base must not be soldered.

The terminal base uses a polycarbonate resin, which could be deformed by excessive soldering heat.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ALL DIMENSIONS SHOWN ARE IN MILLIMETERS.
To convert millimeters into inches, multiply by 0.03937. To convert grams into ounces, multiply by 0.03527.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้