ห้องสมุดคณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ชุดทดลองเพื่อการศึกษา
การทำงานของมอเตอร์ไฟฟ้ากระแสตรงแบบไม่มีแปรงถ่าน

BRUSHLESS D.C. MOTOR EDUCATION 'S KIT

โดย

นาย เรวัต ขันธรรม

นาย วิบูลย์ พิชัยยา

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

ภาควิชา เทคโนโลยีการวัดคุมทางอุตสหกรรม

สาขาวิชา เทคโนโลยีคอมพิวเตอร์อุตสาหกรรม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหาร ลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032735

เรื่อง ชุดทดลองเพื่อการศึกษา
การทำงานของมอเตอร์ไฟฟ้ากระแสตรงแบบไม่มีแปรงถ่าน

BRUSHLESS D.C. MOTOR EDUCATION 'S KIT

ผู้จัดทำ 1 นาย เรวัต ขันธรรม รหัสประจำตัว 34162223 ห้อง 2Q

2 นาย วิบูลย์ พิชัยยา รหัสประจำตัว 34162225 ห้อง 2Q

อาจารย์ที่ปรึกษา

อาจารย์ สิงห์ทอง พัฒนเศรษฐานนท์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

032735

ภาควิชา เทคโนโลยีการวัดคุมทางอุตสาหกรรม

สาขาวิชา เทคโนโลยีคอมพิวเตอร์อุตสาหกรรม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้า เจ้าคุณทหารลาดกระบัง

โครงงานเรื่อง ชุดทดลองเพื่อการศึกษา การทำงานของมอเตอร์ไฟฟ้ากระแสตรง
แบบไม่มีแปรงถ่าน

รายนามผู้ร่วมทำโครงงาน

1. นาย เรวัต ขันธรรม รหัส 34162223
2. นาย วิบูลย์ พิชัยยา รหัส 34162225

อาจารย์ที่ปรึกษา

1. อาจารย์ สิงห์ทอง พัฒนเศรษฐานนท์

..... ลายเซ็นอาจารย์ที่ปรึกษา

(สิงห์ทอง พัฒนเศรษฐานนท์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทคัดย่อ

โครงงานนี้ ได้เสนอชุดทดลองเพื่อการศึกษาในเรื่อง การทำงานของมอเตอร์ไฟฟ้ากระแสตรงแบบไม่มีแปรงถ่าน ซึ่งเป็นความรู้พื้นฐานที่จะได้ทำความเข้าใจและสามารถนำไปประยุกต์ใช้งานได้ต่อไป

ชุดทดลองนี้ประกอบด้วย ไอ.ซี.เบอร์ LM621 ซึ่งทำหน้าที่เป็นตัวควบคุมการทำงานของคอมมิวเตเตอร์ ไอ.ซี. เบอร์ LM629 ที่ทำหน้าที่เป็นตัวรับข้อมูลจากเครื่องไมโครค่อมพิวเตอร์ เพื่อทำการควบคุมความเร็วและทิศทางการหมุนของมอเตอร์ วงจรไดร์เวอร์แบบต่างๆ วงจรป้องกันการไหลของกระแสเกินขนาด และวงจรสังเคราะห์สัญญาณพัลส์วิทมอดูเลเตอร์ เพื่อป้อนสัญญาณให้กับ ไอ.ซี. LM621 ที่ใช้ในเป็นตัวควบคุมมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ABSTRACT

This project is the education's kit for studying the function and operation of D.C. Brushless Motor. This is a fundamental knowledge that we are able to get understanding and applying for the other concerns.

The project are contain the I.C. chip LM621 for Brushless Motor Commutator, LM629 which performs an interfacing with the Microcomputer and control the motor's speed and, also, rotation direction. Furthermore, the project are also provided the driver circuits, a Current Protection circuit, and a Pulse Width Modula tor circuit for supplying the control signal to the LM621 by manually.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้าที่
บทนำ	6
วัตถุประสงค์ของปริญญานิพนธ์	7
ขอบเขตของปริญญานิพนธ์	7
บทที่ 1 โครงสร้างพื้นฐานของมอเตอร์ไฟฟ้ากระแสตรงแบบไม่มีแปรงถ่าน	8
1.1. ดีซี มอเตอร์แบบเก่า ชนิดที่ใช้คอมมิวเตเตอร์และแปรงถ่าน	8
1.2. โครงสร้างพื้นฐานของ ดีซี มอเตอร์ ชนิดไม่มีแปรงถ่าน	10
1.3. อุปกรณ์อิเล็คทรอนิคส์ ที่ใช้ในการตรวจสอบตำแหน่งขั้วแม่เหล็ก	12
1.3.1. โฟโต้ทรานซิสเตอร์ และออปติคัลสวิทช์	12
1.3.2. ฮอลล์อิลีเมนท์	15
1.4. การนำเอาฮอลล์อิลีเมนท์ มาประยุกต์ใช้งานในการตรวจสอบตำแหน่งของโรเตอร์	17
1.5. การต่อใช้งาน ดีซี มอเตอร์ ชนิดไม่มีแปรงถ่านแบบไบโพล่าร์	18
บทที่ 2 LM621 BRUSHLESS MOTOR COMMUTATOR	23
2.1. รายละเอียดทั่วไป	23
2.2. รายละเอียดของขาอินพุทและเอาท์พุท	24
2.3. รายละเอียดคุณสมบัติทางไฟฟ้าของ LM621	27

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่

2.4. ตัวอย่างคุณลักษณะความสามารถของ LM621 29

2.5. รายละเอียดการทำงานของ ไอ.ซี. LM621 29

2.6. การควบคุมการใช้งานของมอเตอร์แบบสามเฟส 30

2.7. การใช้งานมอเตอร์แบบสี่เฟส 33

2.8. ลักษณะของ DEAD-TIME 33

บทที่ 3 การทำงานของ ไอ.ซี. เบอร์ LM629 36

3.0 รายละเอียดเบื้องต้น 36

3.1 คุณลักษณะสำคัญ 37

3.2 รายละเอียดการใช้งานในแต่ละขาของ LM629 37

3.3 ทฤษฎีการทำงาน 40

ระบบเชื่อมต่อในการป้อนกลับของตำแหน่งโรเตอร์ 42

การสร้าง Velocity Profile 43

PID (Proportional Integral Derivation 46

การทำงานในสภาวะอ่าน/เขียน ของ LM629 48

สัญญาณเอ้าท์พุทของ LM629 51

สรุปคำสั่งที่ใช้ใน LM629 51

บทที่ 4 การควบคุมการทำงานของ LM629 โดยใช้ไมโครคอนโทล 53
เลอร์

4.1 แนวความคิดพื้นฐาน 53

4.2 การทำงานของชุดเชื่อมต่อ 8031 53

4.3 การกำหนดแอดเดรส 54

4.4 คำสั่งที่ใช้ควบคุมการทำงานของ LM629 54

4.5 การจัดหน่วยความจำภายในของ 8031 55

โปรแกรมที่ใช้สำหรับ 8031 58

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หน้าที่

บทที่ 5 การทดลองประกอบการศึกษาการทำงานของมอเตอร์ไฟฟ้ากระแสตรง แบบไม่มีแปรงถ่าน 76

5.1. การทดลองที่ 1 ศึกษาการทำงานของมอเตอร์ที่สภาวะ OPEN-LOOP 76

5.2. การทดลองที่ 2 ศึกษาการทำงานของมอเตอร์ที่สภาวะ CLOSE-LOOP 78

5.3. การทดลองที่ 3 การควบคุมการทำงานของ IC เบอร์ LM629 โดยไมโครชิพ 81

5.4. การทดลองที่ 4 การควบคุมการทำงานของ IC เบอร์ LM629 โดยไมโครคอมพิวเตอร์ผ่านทางปริ้นเตอร์พอร์ท 82

5.5. คำถามท้ายการทดลอง 83

วงจรที่ใช้ในการทดลองโครงงานแผ่นที่ 1 84

วงจรที่ใช้ในการทดลองโครงงานแผ่นที่ 2 85

วงจร DRIVER ชนิด 3 เฟส แบบที่ 1 86

วงจร DRIVER ชนิด 3 เฟส แบบที่ 2 87

วงจร DRIVER ชนิด 4 เฟส 88

วงจรสำหรับต่อเชื่อมการควบคุมจากภายนอก 89

วงจรเปลี่ยนระดับของสัญญาณฮอล์ลอีลีเมนท์ให้เป็นระดับของสัญญาณทีทีแอล (TTL Level) 90

บทสรุป 91

เอกสารอ้างอิง 92

ภาคผนวก โปรแกรมคอมพิวเตอร์ 93

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทนำ

โดยทั่วไปแล้ว โครงสร้างของมอเตอร์ไฟฟ้ากระแสตรง หรือ ดีซี มอเตอร์ จะประกอบด้วย **คอมมิวเตเตอร์** (COMMUTATOR) และ**แปรงถ่าน** (BRUSHES) ซึ่งในการใช้งานของมอเตอร์ชนิดนี้ จะมีการสึกหรอเกิดขึ้นระหว่างหน้าสัมผัสของชี่คอมมิวเตเตอร์ กับแปรงถ่าน จึงจำเป็นต้องมีการทำความสะอาด ซ่อมแซม และบำรุงรักษาอยู่เสมอ ปกติแล้วมอเตอร์ชนิดนี้เหมาะสำหรับการใช้งานในลักษณะงานที่มีระบบการควบคุมและตรวจสอบแบบย้อนกลับ (FEED BACK CONTROL) หรือในลักษณะของ **เซอร์โวมอเตอร์** นั่นเอง

ส่วนการใช้งานในลักษณะที่ต้องการความสะอาด และต้องการลด หรือขจัดปัญหาในการบำรุงรักษา (MAINTENANCE FREE) เช่น ฟล๊อปปี้ดิสก์ (FLOPPY DISK) ฮาร์ดดิสก์ (HARD DISK) เครื่องเล่นวีดิโอเทป หรือแม้กระทั่ง เครื่องเล่นคอมแพคดิสก์ เป็นต้น บริษัทผู้ผลิตจึงจำเป็นที่จะต้องหลีกเลี่ยงการใช้ ดีซี มอเตอร์แบบเก่า แล้วหันมาพัฒนา ดีซี มอเตอร์ชนิดใหม่ เพื่อหลีกเลี่ยงความเสียหายที่จะเกิดจาก เศษผง หรือวัสดุ ที่สึกหรอของแปรงถ่านและคอมมิวเตเตอร์ มาเป็น ดีซี มอเตอร์ ชนิดที่ไม่มีแปรงถ่าน (D.C. BRUSH LESS MOTOR) โดยอาศัยอุปกรณ์อิเล็คทรอนิคส์มาทำหน้าที่เป็นสวิทช์ในการตรวจสอบตำแหน่งของโรเตอร์

ชุดทดลองที่ได้จัดทำในโครงงานนี้ เป็นชุดทดลองที่ช่วยให้นักศึกษารวมทั้งผู้ที่สนใจสามารถที่จะศึกษาให้ทราบถึง วิธีการควบคุมการทำงานของมอเตอร์ไฟฟ้ากระแสตรง ชนิดที่ไม่มีแปรงถ่าน และการประยุกต์ใช้มอเตอร์ชนิดนี้ ชุดทดลองนี้ได้ปรรจุวงจรที่ใช้ควบคุมการทำงานของคอมมิวเตเตอร์ วงจรไดร์เวอร์ที่ใช้ขับมอเตอร์ให้ทำงาน ทั้งยังรวมถึงส่วนที่ใช้ในการติดต่อ กับชุดคอนโทรลเลอร์ ซึ่งอาจจะเป็น ชิงเกิลบอร์ดไมโครโปรเซสเซอร์ และไมโครคอมพิวเตอร์ ทำให้ผู้ทดลองสามารถที่จะขยายขอบเขตการใช้งานมอเตอร์แบบนี้ได้กว้างขวางยิ่งขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วัตถุประสงค์ของปริญญานิพนธ์

ปริญญานิพนธ์ฉบับนี้ เขียนขึ้นเพื่ออธิบายถึงพื้นฐาน การทำงานของมอเตอร์ไฟฟ้ากระแสตรงแบบไม่มีแปรงถ่าน การควบคุมการทำงานของมอเตอร์ โดยอาศัยอุปกรณ์ไอ.ซี. เบอร์ LM621 ซึ่งเป็นผลิตภัณฑ์ของบริษัท เนชั่นแนล เซมิคอนดัคเตอร์ รวมทั้งอุปกรณ์สนับสนุนต่างๆ ทำให้ผู้ที่ใช้งานชุดทดลองของโครงงานนี้สามารถประยุกต์ใช้งานมอเตอร์ชนิดนี้ได้อย่างง่ายดาย

ขอบเขตของปริญญานิพนธ์

เนื้อหาของปริญญานิพนธ์ฉบับนี้ได้ครอบคลุมถึง พื้นฐานการทำงานของ มอเตอร์ไฟฟ้ากระแสตรงชนิดไม่มีแปรงถ่าน วิธีการควบคุมการทำงานของมอเตอร์ชนิดนี้ด้วยมือ และการทำงานของวงจรควบคุมที่มีอยู่ในชุดทดลองของโครงงานนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

โครงสร้างพื้นฐานของ มอเตอร์ไฟฟ้ากระแสตรงแบบไม่มีแปลงถ่าน (D.C. BRUSHLESS MOTOR)

เพื่อที่จะให้เข้าใจถึงการทำงานและความแตกต่างของ ดีซี มอเตอร์ ทั้งแบบที่ใช้แปรงถ่านและแบบที่ไม่มีแปรงถ่าน ให้ดียิ่งขึ้น ในที่นี้จะเปรียบเทียบกันได้ดังนี้

1.1. ดีซีมอเตอร์แบบเก่าชนิดที่ใช้คอมมิวเตเตอร์และแปรงถ่าน

โครงสร้างพื้นฐานจะประกอบด้วยส่วนสำคัญ ดังนี้

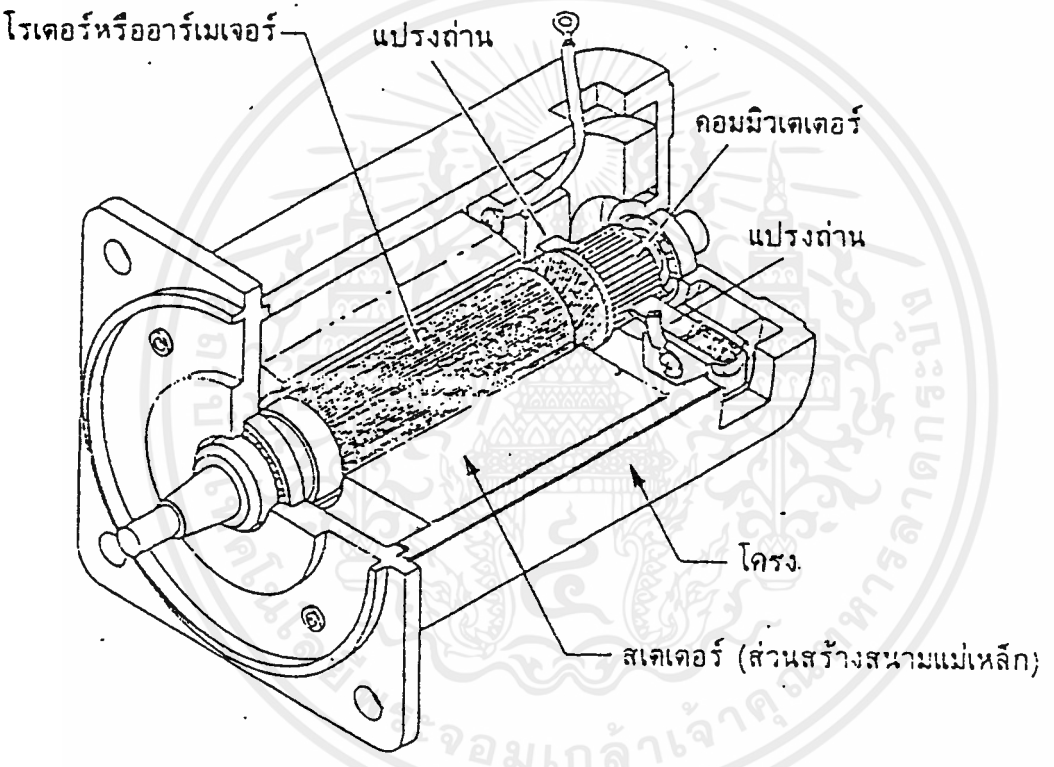
1. โรเตอร์ (ROTOR) จะเป็นส่วนที่เคลื่อนที่หรือหมุน ของมอเตอร์
2. สเตเตอร์ (STATOR) เป็นส่วนที่อยู่กับที่ โดยจะติดอยู่กับตัวมอเตอร์

สเตเตอร์ เป็นส่วนที่สร้างสนามแม่เหล็ก ซึ่งโดยทั่วไปแล้วจะเป็นแม่เหล็กถาวรสองชิ้นประกอบติดอยู่กับโครงโลหะ สนามแม่เหล็กที่เกิดขึ้นจากแม่เหล็กถาวรทั้งสองชิ้นนี้จะเป็นตัวสร้างแรงบิด ส่วนอาร์เมเจอร์ (ARMATURE) หรือ โรเตอร์ (ROTOR) ซึ่งเป็นส่วนที่เคลื่อนที่นั้น จะประกอบด้วยขดลวดพันอยู่รอบๆ แกนเหล็กกลมที่แบ่งเป็นชุดๆ และปลายของขดลวดแต่ละชุดจะถูกต่อเข้ากับแท่งทองแดงที่เป็น คอมมิวเตเตอร์ ซึ่งอยู่รอบๆ แกนของอาร์เมเจอร์ ในการสร้างสนามแม่เหล็กจะอาศัยกระแสไฟฟ้าที่ไหลผ่านเข้ามาทางแปรงถ่าน ซึ่งเป็นส่วนที่อยู่กับที่และสัมผัสกับคอมมิวเตเตอร์ไปยังขดลวด สนามแม่เหล็กที่เกิดขึ้นในขดลวดของอาร์เมเจอร์จะดูดหรือผลักกับสนามแม่เหล็กในสเตเตอร์ ทำให้โรเตอร์ซึ่งเป็นแกนเดียวกับอาร์เมเจอร์หมุนไปได้ และเมื่อโรเตอร์หุนไปก็จะทำให้แปรงถ่านสัมผัสกับซี่คอมมิวเตเตอร์ของขดลวดชุดต่อไป จึงทำให้ขดลาดชุดใหม่สร้างสนามแม่เหล็กต่อไปเรื่อยๆ ซึ่งจะทำให้มอเตอร์หมุนได้อย่างต่อเนื่อง

สำหรับแปรงถ่านนั้นส่วนใหญ่จะทำมาจากแกรไฟต์ (GRAPHITE) ซึ่งเป็นส่วนที่ทำหน้าที่ผ่านกระแสไฟจากแหล่งจ่ายไปยังคอมมิวเตเตอร์และขดลวด ตัวแปรงถ่านจะติดตั้งอยู่ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระบอกมีสปริงดันให้สัมผัสกับคอมมิวเตเตอร์ตลอดเวลา โดยปกติในดีซีมอเตอร์ตัวหนึ่งจะมีแปรงถ่านอยู่สองอัน



รูปที่ 1.1 โครงสร้างของดีซีมอเตอร์แบบมีแปรงถ่าน

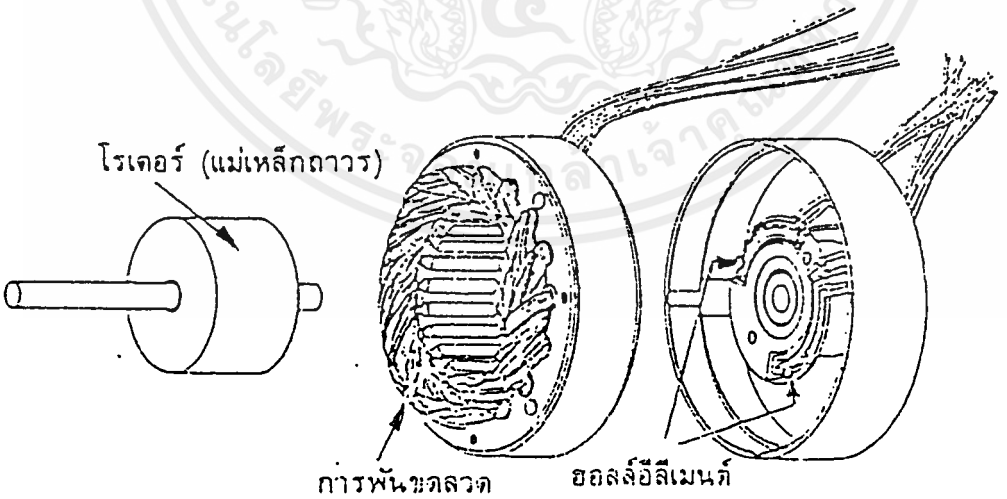
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้
032735

1.2. โครงสร้างพื้นฐานของดีซีมอเตอร์ชนิดที่ไม่มีแปรงถ่าน

โครงสร้างของมอเตอร์ชนิดนี้มีความสลับซับซ้อนมากกว่า ดีซีมอเตอร์แบบมีแปรงถ่านเล็กน้อย โครงสร้างโดยทั่วไปแล้วจะคล้ายคลึงกับ เอซีมอเตอร์ กล่าวคือ ขดลวดจะถูกพันอยู่กับสเตเตอร์ ซึ่งขดลวดนี้จะไม่มีการหมุน ส่วนตัวโรเตอร์จะประกอบด้วยแม่เหล็กถาวร โดยที่แม่เหล็กถาวรดังกล่าวนั้น อาจจะติดตั้งอยู่ด้านในของตัวสเตเตอร์เช่นเดียวกับโรเตอร์ของมอเตอร์ทั่วๆ ไป หรืออาจจะอยู่ภายนอกโดยครอบขดลวดอาร์เมเจอร์ก็ได้ ทั้งนี้ขึ้นอยู่กับวัตถุประสงค์ของการใช้งานโครงสร้างทั้งสองแบบนี้ ดังแสดงในรูป 1.2 และรูป 1.3 ส่วนประกอบที่สำคัญของมอเตอร์ชนิดนี้อีกอันหนึ่งก็คือ อุปกรณ์ที่ใช้ในการตรวจสอบตำแหน่งของขั้วแม่เหล็กของโรเตอร์ อุปกรณ์ชนิดนี้จะใช้เพื่อทำการสร้างสัญญาณควบคุมเพื่อส่งไปให้วงจรขับเคลื่อน โดยทั่วไปแล้วจะเป็นอุปกรณ์ทางอิเล็คทรอนิคส์ที่ทำหน้าที่เป็นสวิทช์ ซึ่งนิยมใช้กันอยู่ทั่วไปมีอยู่ 3 ชนิด คือ

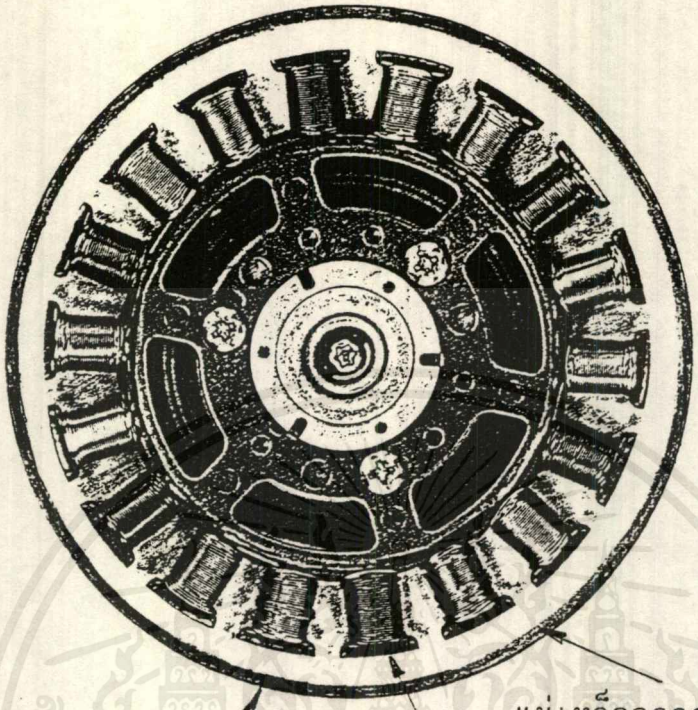
1. โฟโต้ทรานซิสเตอร์ (PHOTO TRANSISTOR)
2. ออปติคอล สวิทช์ (OPTICAL SWITCH)
3. ฮอลอิลลิเมนท์ (HALL ELEMENT)

โดยทั่วไปจะนิยมใช้ ฮอลอิลลิเมนท์ มากที่สุด



รูปที่ 1.2 แบบโรเตอร์อยู่ด้านใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.3 แบบโรเตอร์อยู่ด้านนอก

ดีซีมอเตอร์ แบบที่มีโรเตอร์อยู่ข้างในนั้น เหมาะกับงานที่ต้องการแรงบิดสูง และใช้กับงานที่ต้องการความเร็วไม่สูงมากนัก ส่วนแบบที่มีโรเตอร์อยู่ด้านนอกอาร์เมเจอร์จะเหมาะกับงานที่ต้องการแรงบิดต่ำ แต่ความเร็วสูง เช่น สปิลเด็ลมอเตอร์ (SPINDLE MOTOR) ที่ใช้ใน ฟล็อปปี้ดิสต์ และฮาร์ทดิสต์ เป็นต้น

ข้อแตกต่างอีกอันหนึ่งของดีซีมอเตอร์แบบไม่มีแปรงถ่านที่ต่างจากดีซีมอเตอร์ธรรมดา คือ มีขดลวดสร้างสนามแม่เหล็กของอาร์เมเจอร์ถึง 3 ชุด (หรือที่เรียกว่า 3 เฟส) ทั้งนี้ก็เพื่อเพิ่มประสิทธิภาพการทำงานของมอเตอร์

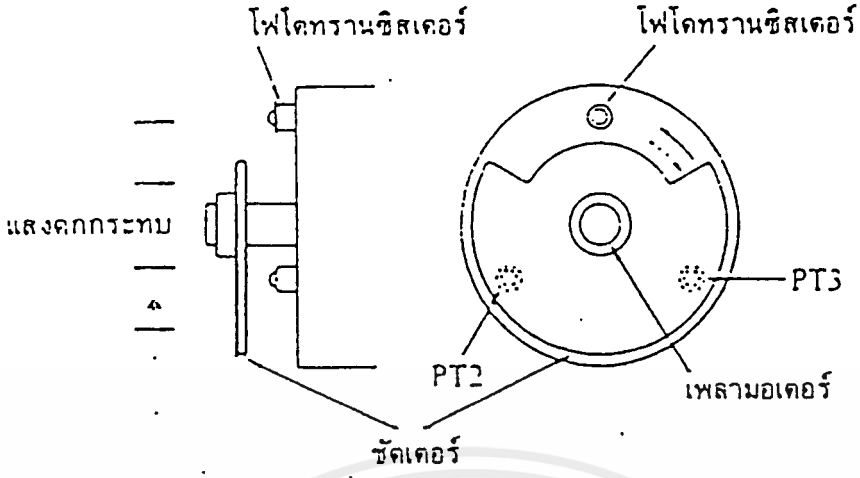
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3. อุปกรณ์อิเล็คทรอนิคส์ที่ใช้ในการตรวจสอบตำแหน่งขั้วของแม่เหล็ก

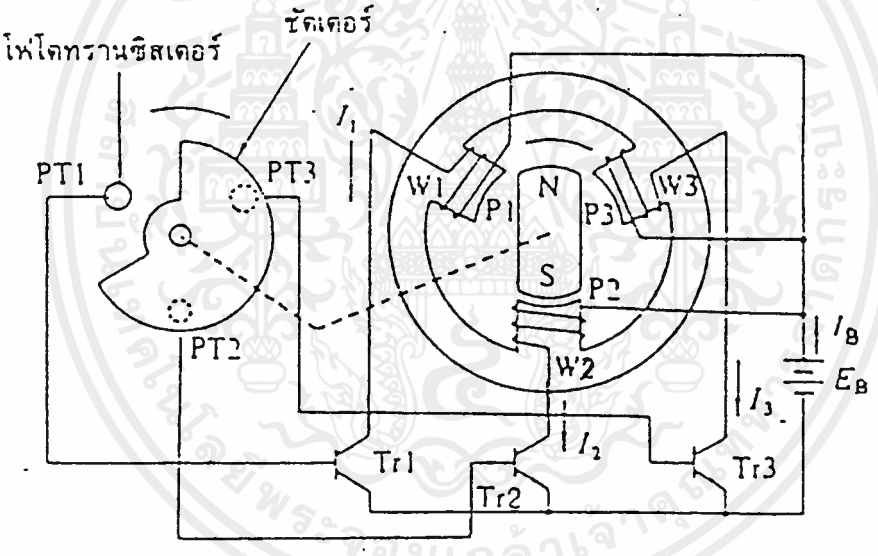
1.3.1 โฟโต้ทรานซิสเตอร์ (PHOTO-TRANSISTOR) และออปติคอล สวิทช์ (OPTICAL SWITCH)

เนื่องจากอุปกรณ์ทั้งสองนี้ เป็นอุปกรณ์ที่อาศัยการทำงานของแสงเพื่อไบแอสให้กับตัวทรานซิสเตอร์ ดังนั้นในการใช้งานจึงต้องอาศัยการติดตั้งแผ่นชัตเตอร์ไว้ที่ตัวโรเตอร์ เพื่อใช้เป็นตัว ปิดและเปิด แสงให้กับโฟโต้ทรานซิสเตอร์เมื่อโรเตอร์หมุนเปลี่ยนตำแหน่งไป ในที่นี้จะขอยกตัวอย่างการใช้โฟโต้ทรานซิสเตอร์ ในการตรวจสอบตำแหน่งของขั้วแม่เหล็กของโรเตอร์ชนิดที่ขดลวดอาร์เมเจอร์เป็นแบบสามเฟส ดังแสดงในรูป 1.4. ซึ่งจะใช้โฟโต้ทรานซิสเตอร์ 3 ตัว การวางตำแหน่งของโฟโตทรานซิสเตอร์ทั้งสามห่างกัน 120° และที่แกนของโรเตอร์มีแผ่นชัตเตอร์ตัดเป็นช่องเปิดไว้เท่ากัน คือ 120° เมื่อโรเตอร์หมุนไปก็จะทำให้โฟโต้ทรานซิสเตอร์นำกระแสเรียงตามลำดับกันไป ในรูป 1.5. แสดงการทำงานของโฟโต้ทรานซิสเตอร์ที่ต่ออยู่กับดีซีมอเตอร์ชนิดไม่มีแปรงถ่านแบบสามเฟส จากในรูปจะเห็นว่าขั้วแม่เหล็กของโรเตอร์ขั้วเหนือ (N) อยู่ระหว่างขดลวดที่ตำแหน่ง P1 และ P3 ในตำแหน่งนี้ โฟโตทรานซิสเตอร์ TR1 อยู่ในตำแหน่งที่ได้รับแสง เมื่อเราจ่ายกระแสไฟให้กับดีซีมอเตอร์ ผ่านเข้าทางขดลวดไปยังทรานซิสเตอร์ ก็จะทำให้ TR1 เกิดการนำกระแสและทำให้ขดลวด W1 ที่ P1 สร้างสนามแม่เหล็กขั้วใต้ขึ้นมา จึงทำให้เกิดการดูดขั้วแม่เหล็กเหนือ (N) ของโรเตอร์เข้าหาตัว ในขณะเดียวกันแผ่นชัตเตอร์ที่อยู่กับแกนโรเตอร์ก็จะหมุนปบัง TR1 ทำให้ TR1 หยุดทำงาน และขณะเดียวกัน TR2 ก็จะทำงานแทนเพราะมีแสงไปตกกระทบ เมื่อมีกระแสไฟไหลผ่านขดลวด W2 ของ P2 ทำให้เกิดสนามแม่เหล็กขั้วใต้ (S) ขึ้นมาดูดขั้วเหนือ (N) ของโรเตอร์ให้หมุนต่อไปในลักษณะทวนเข็มนาฬิกา ซึ่งโรเตอร์จะหมุนต่อเนื่องไปเรื่อยๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.4. แสดงการวางตำแหน่งของโฟโต้ทรานซิสเตอร์

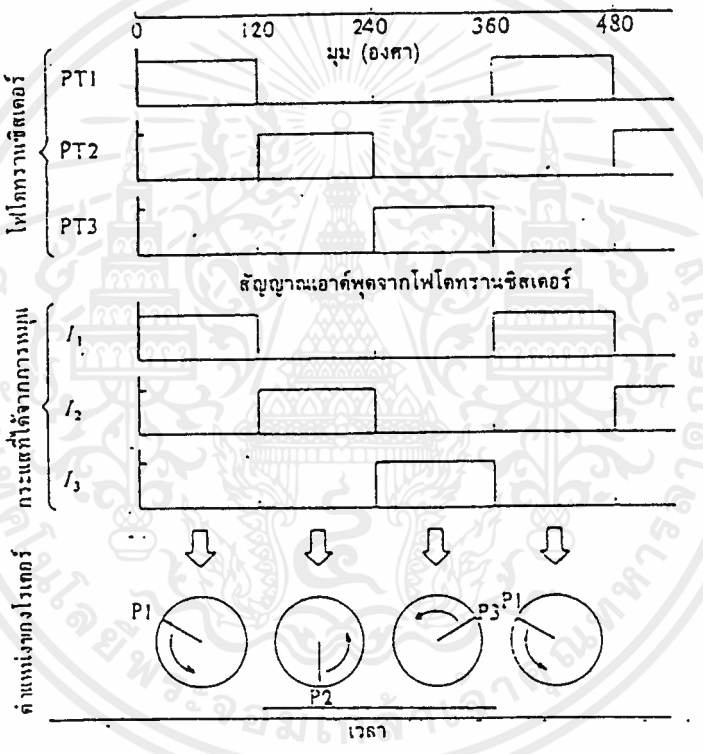


รูปที่ 1.5. แสดงการต่อโฟโต้ทรานซิสเตอร์

ส่วนการที่จะทำให้ของมอเตอร์หมุนกลับทิศได้นั้น ไม่สามารถทำได้โดยใช้วิธีสลับขั้วของแหล่งจ่ายไฟเหมือนกับดี.ซี.มอเตอร์แบบธรรมดาทั่วไป ทั้งนี้เพราะเราใช้อุปกรณ์ทางอิเล็คทรอนิคส์เป็นตัวตรวจสอบขั้วของโรเตอร์ ดังนั้นในการกลับทิศทางการหมุนของมอเตอร์แบบไม่มีแปรงถ่านดังในรูปที่ 1.6. นั้น สามารถทำได้โดยการกลับลำดับการทำงานของโฟโต้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทรานซิสเตอร์ จากเดิมนั้น PT1 ต่ออยู่กับ TR1, PT2 ต่ออยู่กับ TR2 และ PT3 ต่ออยู่กับ TR3 ซึ่งทรานซิสเตอร์ทั้งสามตัวทำหน้าที่สร้างสนามแม่เหล็กที่ขดลวด W_1, W_2, และ W_3 มาเป็น PT_1 ต่อกับ TR_3, PT_2 ต่อกับ TR_2 และ PT_3 ต่อกับ TR_1 ก็จะทำให้การเกิดสนามแม่เหล็กเปลี่ยนมาเริ่มต้นจาก W_3 มา W_2 และ W_1 ตามลำดับ ก็จะทำให้มอเตอร์หมุนกลับทิศได้



รูปที่ 1.6. แสดงลำดับการทำงานของ โฟโต้ทรานซิสเตอร์ กระแสที่ไหลผ่านขดลวด และ ตำแหน่งของมุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.3.2. ฮอลล์อิลีเมนท์ (HALL ELEMENT)

ฮอลล์อิลีเมนท์เป็นอุปกรณ์จำพวกสารกึ่งตัวนำชนิดหนึ่งที่มีคุณสมบัติเหมาะสม ในการนำมาใช้ในการตรวจสอบชิ้วแม่เหล็กของโรเตอร์

ฮอลล์อิลีเมนท์สร้างจากสารกึ่งตัวนำที่มีคุณสมบัติพิเศษ กล่าวคือเมื่อมีกระแสไหลผ่านสารกึ่งตัวนำนี้ในทิศทางหนึ่ง แล้วมีสนามแม่เหล็ก B มาตัดผ่านในลักษณะและทิศทางที่ตั้งฉากกับทิศทางการไหลของกระแส จะทำให้เกิดการเหนี่ยวนำสนามแม่เหล็กไฟฟ้าขึ้นมาในทิศทางที่ตั้งฉากกับสนามแม่เหล็ก B ที่มาตัด ปรากฏการณ์นี้เรา เราเรียกว่าฮอลล์เอฟเฟค (HALL-EFFECT) ซึ่งปรากฏการณ์นี้จะมีผลทำให้ โฮล "HOLD" หรือ อิเล็คตรอน "ELECTRON" เคลื่อนที่ไปยังด้านข้างของสารกึ่งตัวนำ ทำให้มีแรงดันไฟฟ้าเกิดขึ้นมา แรงดันที่ได้นี้เป็นแรงดันขนาดเล็กและเราให้ชื่อว่า VH ซึ่งค่าของ VH นี้จะขึ้นอยู่กับชนิดของสารกึ่งตัวนำที่เอามาทำเป็นฮอลล์อิลีเมนท์ ว่า เป็นชนิด "พี-ไทพ์ (P-TYPE) หรือ "เอ็น-ไทพ์ (N-TYPE)" ซึ่งค่าของ VH นี้สามารถหาได้จาก

$$VH = \frac{IB.IC.RH}{d}$$

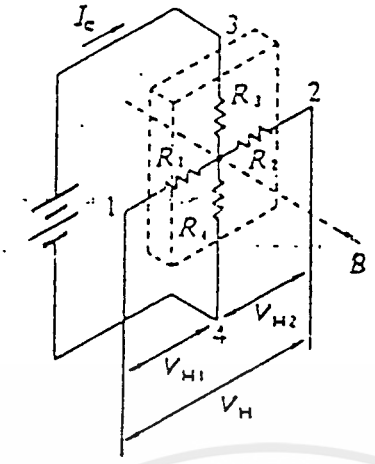
โดยเมื่อ RH เป็นค่าคงที่ของตัวฮอลล์ มีค่าเป็นโอมห์ (OHM)

IC เป็นค่ากระแสไฟฟ้าที่ไหลผ่านฮอลล์อิลีเมนท์ มีหน่วยเป็นแอมป์ (A)

IB เป็นค่าความเข้มของสนามแม่เหล็ก มีหน่วยเป็นทอร์ (T)

d เป็นความหนาของฮอลล์อิลีเมนท์ มีหน่วยเป็นมิลล์ (m)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.7. แสดงถึงปรากฏการณ์ของฮอลล์

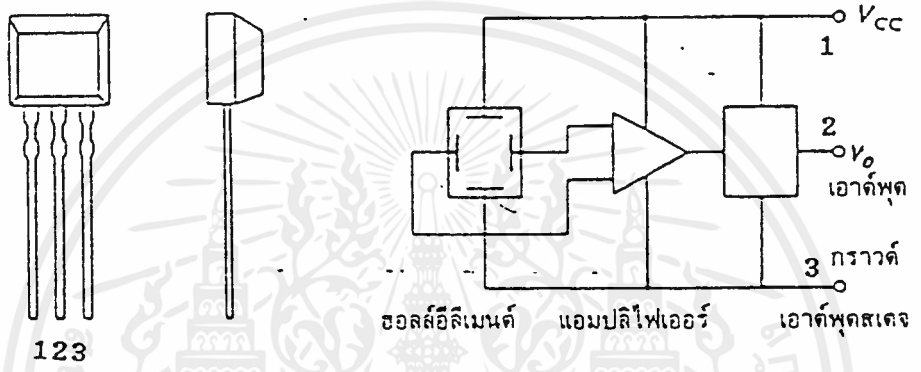
N-TYPE P-TYPE

รูปที่ 1.8. แสดงถึงปรากฏการณ์ของฮอลล์ เอฟเฟค

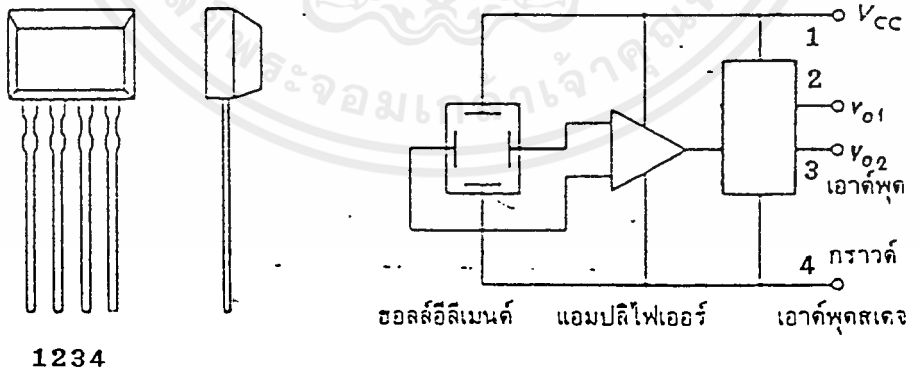
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4. การนำเอาฮอลล์อิลีเมนท์มาประยุกต์ใช้งาน ในการตรวจสอบตำแหน่งของโรเตอร์

เนื่องจากค่าแรงดัน VH ที่เราได้จากฮอลล์อิลีเมนท์นั้นมีขนาดเล็ก ดังนั้นในการนำไปใช้จึงต้องมีการขยายสัญญาณขึ้น เพื่อให้เหมาะสมกับอุปกรณ์ที่ใช้ในการควบคุม ซึ่งในปัจจุบันก็มีหลายบริษัทได้ ผลิตออกมาในรูปที่เป็นตัวไอ.ซี. วึ่งเรียกว่า "ฮอลล์ไอ.ซี." โดยจะมีโครงสร้างทั่วๆ ไปดังแสดงในรูปที่ 1.9.



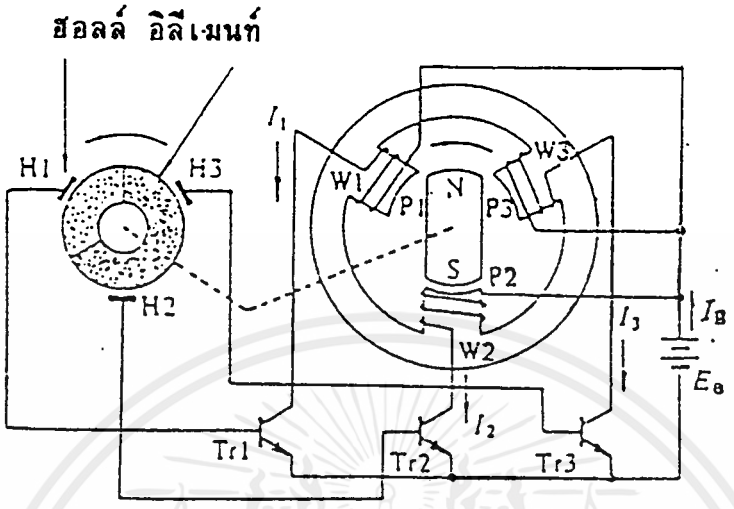
รูปที่ 1.9. แสดงถึงโครงสร้างของ ฮอลล์-ไอ.ซี. แบบ 3 ขา



รูปที่ 1.10. แสดงถึงโครงสร้างฮอลล์ไอ.ซี. แบบ 4 ขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 1.11. เป็นการต่อใช้งานของฮอลล์อิลีเมนท์ ซึ่งต่อเข้าไปแทนที่โฟโต้ทรานซิสเตอร์จากในรูปที่ 1.5.

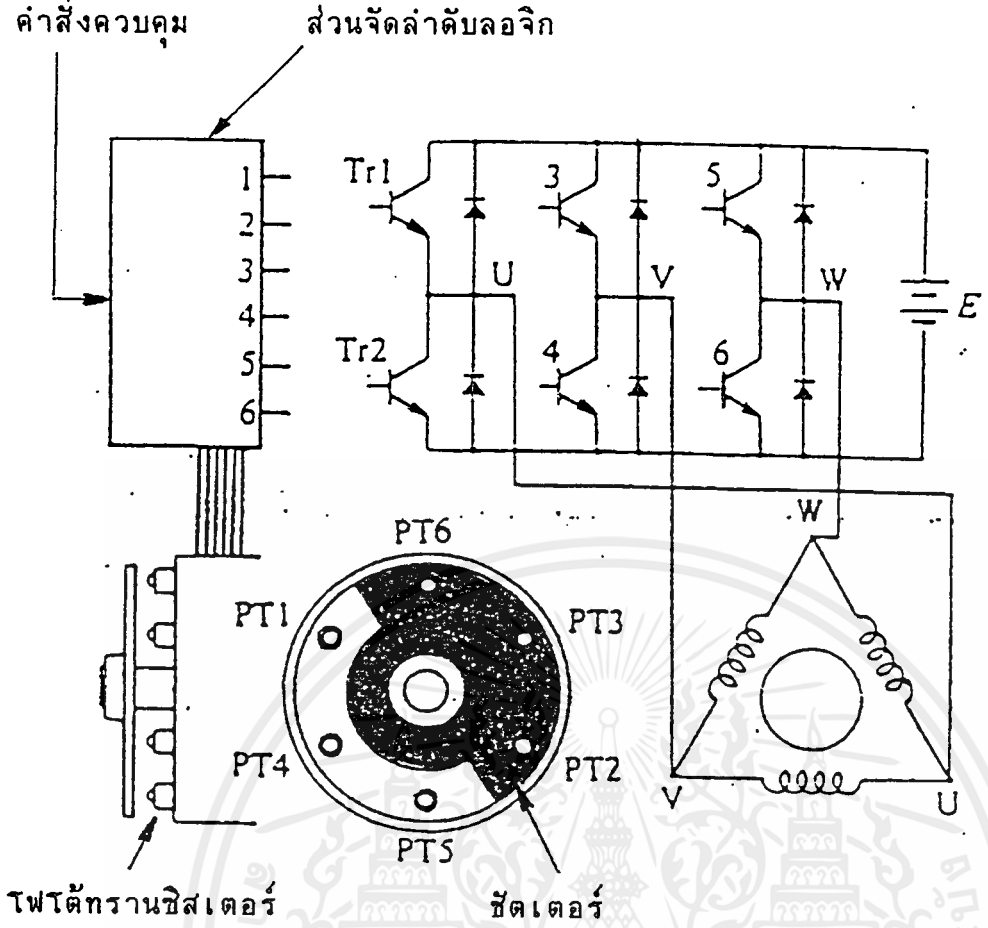


รูปที่ 1.11.
แสดงการต่อใช้งานฮอลล์อิลีเมนท์

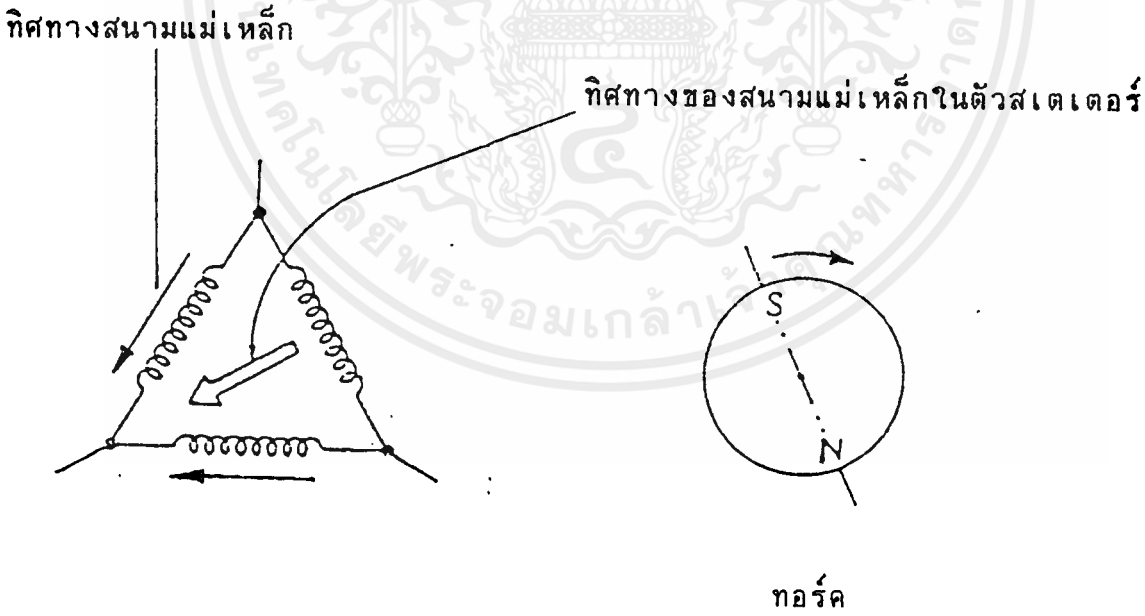
1.5. การต่อใช้งานของ ดี.ซี.มอเตอร์ ชนิดไม่มีแปรงถ่านแบบสามเฟส กระแสไหลสองทิศทาง หรือแบบ ไบโพลาร์ (BI-POLAR)

เนื่องจากเป็นไบโพลาร์ ดังนั้นการต่อวงจรขับเคลื่อนจะต้องอยู่ในรูปของวงจรบริดจ์ การต่อในลักษณะให้กระแสไหลได้สองทิศทางนี้ จะทำให้มอเตอร์ทำงานได้อย่างมีประสิทธิภาพ เพราะการขับเคลื่อนจะใกล้เคียงกับ เอ.ซี.มอเตอร์ วงจรขับจะประกอบด้วยทรานซิสเตอร์สองตังต่อหนึ่งเฟสโดยเป็นทรานซิสเตอร์ชนิด PNP หนึ่งตัวและ NPN หนึ่งตัว รวมทั้งหมดจะมีทรานซิสเตอร์ 6 ตัว ดังนั้นในการที่จะทำให้สามารถควบคุมการทำงานของทรานซิสเตอร์ซิสเตอร์ทั้ง 6 ตัวนี้ได้ จึงจำเป็นต้องมี โฟโต้ทรานซิสเตอร์สำหรับตรวจสอบขั้วแม่เหล็กของโรเตอร์เป็นจำนวน 6 ตัวเช่นเดียวกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 1.12. แสดงการต่อวงจรขับแบบไบโพลาร์ 3 เฟส



รูปที่ 1.13. แสดงทิศทางสนามแม่เหล็กและทอร์คในสเตเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

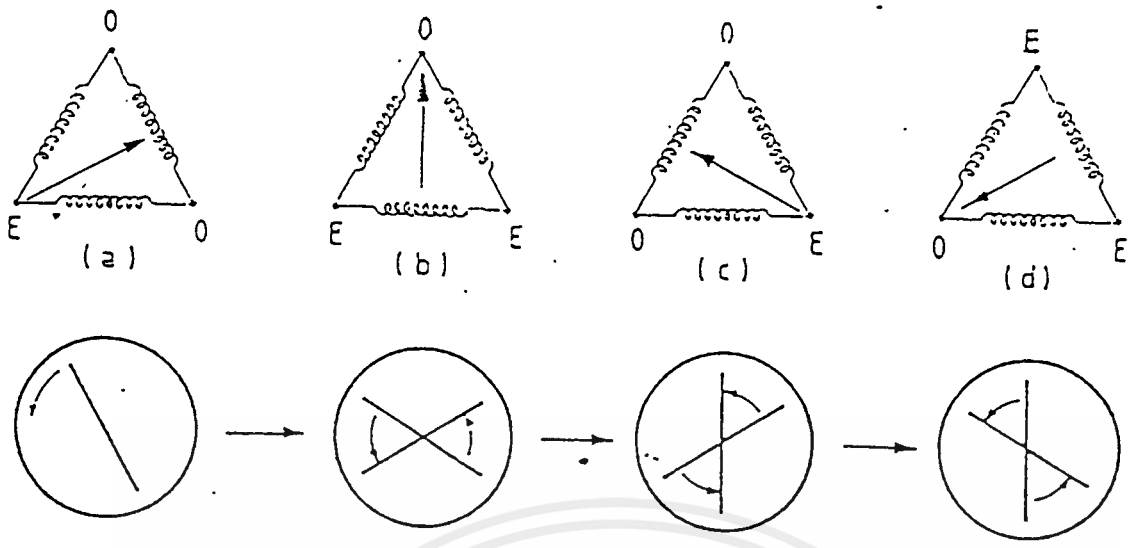
การทำงานของวงจรในรูปที่ 1.12. มีโฟโต้ทรานซิสเตอร์ 3 ตัวคือ PT_1, PT_4, และ PT_5 ถูกไบอัสและทำให้ TR_1, TR_4, และ TR_5 นำกระแส ทำให้มีกระแสไหลผ่านขดลวด U ไปยังขดลวด V และอีกชุดหนึ่งจาก W ไป V ดังแสดงด้วยลูกศรบอกทิศทางของสนามแม่เหล็กที่เกิดขึ้นในสเตเตอร์ ขณะเดียวกันโรเตอร์ซึ่งอยู่ในตำแหน่งที่ทำมุม 90° กับสนามแม่เหล็กในรูปที่ 1.13. ซึ่งจะทำให้เกิดมีแรงบิดที่ทำให้โรเตอร์หมุนไปในทิศทางตามเข็มนาฬิกา และเมื่อโรเตอร์หมุนไปได้ 30° แล้ว โฟโต้ทรานซิสเตอร์จะถูกบัง ทำให้หยุดทำงานและขณะเดียวกัน PT_6 จะได้รับไบอัสทำให้ TR_6 นำกระแสและทำงานต่อไป จึงสามารถดูดให้โรเตอร์หมุนไปได้อีก 60° ในขณะที่ขั้วใต้ของสเตเตอร์จะถูกผลักให้หมุนต่อไป

ส่วนการที่จะให้มอเตอร์หมุนกลับทิศทาง ก็สามารถทำได้เช่นเดียวกันกับทิศทางของมอเตอร์ 3 เฟสแบบกระแสไหลทางเดียว กล่าวคือ ทำการควบคุมให้ลำดับการนำกระแสของทรานซิสเตอร์เปลี่ยนไปโดยการย้ายโฟโต้ทรานซิสเตอร์ PT_1 ไปต่อกับ TR_5 PT_2 ต่อกับ TR_6 PT_3 ต่อกับ TR_3 PT_4 ต่อกับ TR_3 และ PT_5 ต่อกับ TR_2 ก็จะทำให้มอเตอร์หมุนกลับทิศทางได้ตามต้องการ

ลำดับการหมุนของชัตเตอร์		ทวนเข็มนาฬิกา				ตามเข็มนาฬิกา			
		1	2	3	4	1	2	3	4
โฟโต้ทรานซิสเตอร์	PT1	1	0	0	1	1	0	0	1
	PT2	0	1	0	0	0	0	1	0
	PT3	0	0	1	0	0	1	0	0
ทรานซิสเตอร์	TR1	1	0	0	1	0	0	1	0
	TR2	0	1	0	0	0	1	0	0
	TR3	0	0	1	0	1	0	0	1

รูปที่ 1.14 ตารางแสดงการทำงานของโฟโต้ทรานซิสเตอร์และทรานซิสเตอร์แบบกระแสไหลทางเดียว เมื่อหมุนตามเข็มและทวนเข็มนาฬิกา

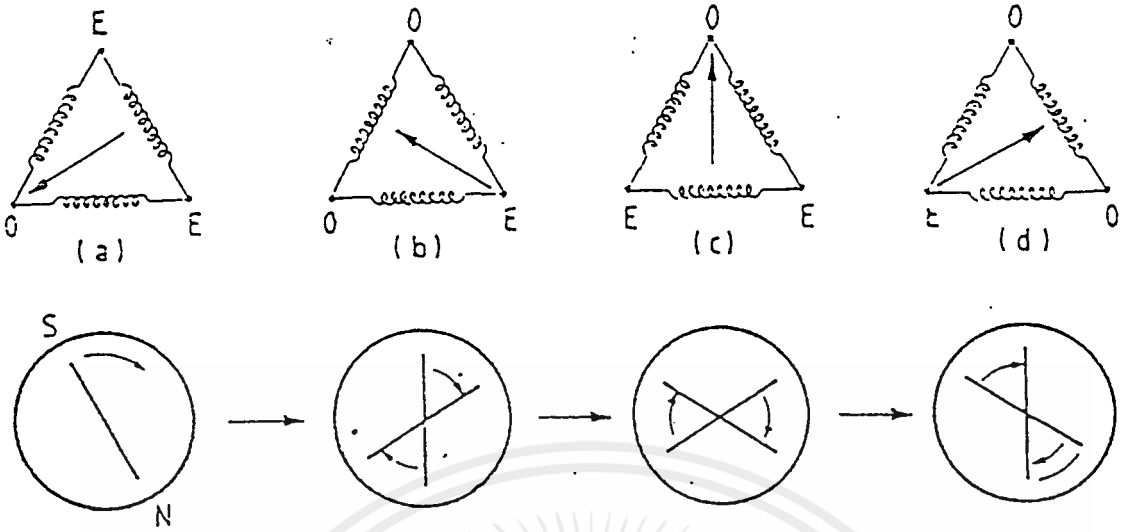
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ลำดับการ ON-OFF ของทรานซิสเตอร์	PT1	PT2	PT3	PT4	PT5	PT6
Tr1	1	1	1	0	0	0
Tr2	0	0	0	1	1	1
Tr3	0	0	1	1	1	0
Tr4	1	1	0	0	0	1
Tr5	1	0	0	0	1	1
Tr6	0	1	1	1	0	0

รูปที่ 1.15 ตารางแสดงการทำงานของโฟโต้ทรานซิสเตอร์และทรานซิสเตอร์แบบกระแสไหลได้สองทิศทาง เมื่อหมุนตามเข็มนาฬิกา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



ลำดับการ ON-OFF ของทรานซิสเตอร์	PT1	PT2	PT3	PT4	PT5	PT6
Tr1	0	1	1	1	0	0
Tr2	1	0	0	0	1	1
Tr3	1	1	0	0	0	1
Tr4	0	0	1	1	1	0
Tr5	0	0	0	1	1	1
Tr6	1	1	1	0	0	0

รูปที่ 1.16. **ตารางแสดงการทำงานของโฟโต้ทรานซิสเตอร์และทรานซิสเตอร์แบบกระแสไหลได้สองทิศทาง เมื่อหมุนทวนเข็มนาฬิกา**

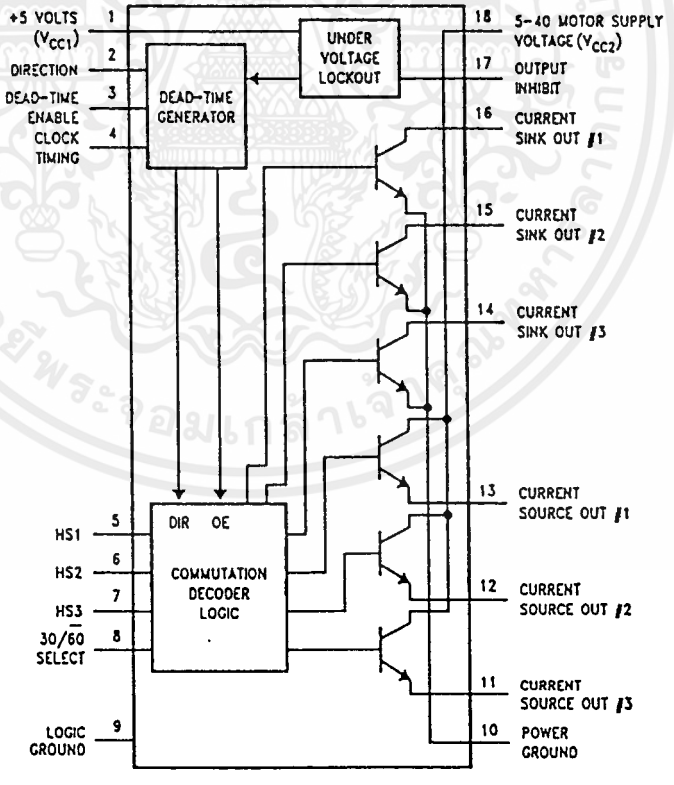
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

LM621 Brushless Motor Commutator:

2.1. รายละเอียดทั่วไป :

ไอ.ซี. เบอร์ LM621 เป็น ไอ.ซี. ที่ออกแบบโดยใช้เทคโนโลยีไบโพล่าที่ถูกออกแบบสำหรับใช้ในการแปลงพลังงานของ BRUSHLESS DC MOTOR ไอ.ซี. เบอร์นี้สามารถใช้ได้กับทั้งมอเตอร์ 3-เฟส และ 4-เฟส โดยสามารถที่จะขับอุปกรณ์ Power switching ที่ใช้ในการขับตัวมอเตอร์ ภายในไอ.ซี. เบอร์นี้ยังมีวงจร DEAD-TIME CIRCUIT ที่ปรับค่าได้ ซึ่งวงจรนี้ใช้เพื่อขจัดกระแส "SHOOT-THROUGH" ที่เกิดขึ้นในวงจร POWER SWITCHING ไอ.ซี.เบอร์นี้จะทำงานภายใต้แหล่งจ่ายไฟตั้งแต่บวก 5 โวลต์ขึ้นไปและก็สามารถที่จะทำให้สัญญาณเอาท์พุทเปลี่ยนแปลงได้ถึง 40 โวลต์ ตัวถังของไอ.ซี.เบอร์ LM621 นี้จะเป็นแบบ DIP (Dual-in-Line) 18 ขา



รูปที่ 2.1. บล็อคไดอะแกรมภายในของ ไอ.ซี. LM621

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2. รายละเอียดของขาอินพุท และเอาท์พุท :

ขา 1: VCC1 (+5V) : เป็นขาไฟเลี้ยงสำหรับวงจรลอจิก และวงจรที่ทำงานเกี่ยวกับสัญญาณนาฬิกาภายในตัว LM621

ขา 2: DIRECTION : เป็นขาอินพุท ที่เป็นตัวกำหนดทิศทางการหมุนของโรเตอร์ว่าหมุนตามเข็ม หรือทวนเข็มนาฬิกา ทั้งนี้ดูได้จากตารางในรูปที่ 2.6

ขา 3: DEAD-TIME : เป็นขาอินพุท ซึ่งสัญญาณที่ป้อนเข้ามาที่ขานี้จะเป็นตัว "เอเนเบิล" หรือ ENABLE "ดิสเอเบิล" การทำงานของวงจร "เดดไทม์ เจนเนอเรเตอร์" เมื่อต่อขานี้เข้ากับไฟ +5 โวลต์จะเป็นการเอเนเบิลให้วงจรเดดไทม์ เจนเนอเรเตอร์ทำงาน แต่ถ้าต่อลงกราวน์จะเป็นการดิสเอเบิลไม่ให้วงจรเดดไทม์-เจนเนอเรเตอร์ทำงาน อย่างไรก็ตามจะต้องไม่ปล่อยให้ขา 3 นี้ลอยไว้

ขา 4: CLOCK TIMER: ขานี้ใช้ในการต่อกับ ตัวต้านทานและตัวเก็บประจุภายนอกเพื่อใช้ในการควบคุมช่วง "พีเรียด" (PERIOD) ของสัญญาณนาฬิกาที่สร้างโดยวงจรออสซิลเลเตอร์ ซึ่งก็จะเป็นตัวกำหนดค่าช่วงเวลา "เดดไทม์" ด้วยเช่นกัน

ขา 5, 6, และ 7: HS1, HS2, และ HS3:
ทั้งสามขานี้เป็นขาอินพุทที่รับสัญญาณจาก "ฮอลล์-เอฟเฟ็ค" เซ็นเซ่อร์ที่กำหนดตำแหน่งของโรเตอร์ จากตัวมอเตอร์เข้ามา สำหรับมอเตอร์แบบสามเฟสแล้ว จะมีสัญญาณเข้ามา 3 สัญญาณ ส่วนมอเตอร์แบบสี่เฟสจะมีสัญญาณออกมาเพียงสองสัญญาณ ต้องเอาสัญญาณอันใดอันหนึ่งมาต่อเข้าทั้งที่ขา 6 (HS2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และขา 7 (HS3)

ขา 8: 30/60 SELECT: ขาอินพุทขานี้ ใช้เป็นตัวเลือกลักษณะของการถอดรหัสของมอเตอร์แบบสามเฟส กล่าวคือ ถ้าต่อขานี้เข้ากับไฟ +5 โวลท์ ก็จะเป็นการถอดรหัสในแบบ 30 องศา หรือถ้าต่อขานี้ลงกราวน์ ก็จะเป็นแบบ 60 องศา แต่ถ้าไปใช้ในการควบคุมมอเตอร์แบบสี่เฟส ก็ให้ต่อขานี้เข้ากับ + 5 โวลท์

ขา 9: LOGIC GROUND: ขากราวด์นสำหรับวงจรลอจิกใน LM621

ขา 10: POWER GROUND: ขาต่อกราวด์นสำหรับวงจร "เอาท์พุท บัฟเฟอร์"

ขา 11, 12, และ 13 : SOURCE OUTPUTS:
ทั้งสามขานี้เป็นขาเอาท์พุทที่เป็นตัวจ่ายกระแส "ซอสส์" (SOURCE CURRENT) เพื่อใช้ในการขับวงจรขับสวิทช์ชิ่งที่ขับมอเตอร์

ขา 14, 15, และ 16 : SINK OUTPUTS :
ทั้งสามขานี้ เป็นขาเอาท์พุทเช่นเดียวกัน ซึ่งจะรับกระแส "ซิงค์" (SINK CURRENT) ในการขับวงจรขับสวิทช์ชิ่งที่เป็นตัวขับมอเตอร์

ขา 17 : OUTPUT INHIBIT :
เป็นขาอินพุท ที่รับสัญญาณซึ่งเป็นตัวดิสเอเบิลการทำงานของวงจรขับที่เอาท์พุทของ LM621 เช่น ถูกป้อนด้วยสัญญาณ "แมคนิจ๊ด" จากวงจร "พัลส์วิด-ม็อดดูเลเตอร์" (PULSE WIDTH MODULATOR) ซึ่งหากขา 17 นี้มีลอจิกเป็น "1"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หรือ +5 โวลท์แล้ว วงจรขับที่เอาท์พุทจะถูกควบคุมให้หยุดทำงาน

ขา 18 : VCC2 (+5 ถึง +40 โวลท์):

ขานี้ ใช้ต่อไฟเลี้ยงให้กับขาคอลเลคเตอร์ของวงจรจ่ายกระแสทั้งสามชุด (ขา 11 ถึง 13) หากเรานำไอ.ซี. LM621 ไปขับวงจรสวิทช์ชิ่งที่เป็น "มอสเฟท" (MOSFET)แล้ว เราอาจจะต่อขา 18 นี้เข้ากับไฟเลี้ยงได้สูงถึง +40 โวลท์ ทั้งนี้ ก็เพื่อให้มีแรงดันสูงพอที่จะไปขับวงจรเกจของมอสเฟท แต่ถ้าวงจรสวิทช์ชิ่งนั้นเป็นวงจรที่ใช้อุปกรณ์ชนิด ไบโพลาร์ แล้วขา 18 นี้ควรจะต่อกับไฟ +5 โวลท์ เพื่อที่จะลดความสูญเสียพลังงานในตัว ไอ.ซี. อย่างไรก็ตาม วงจร "อันเดอร์โวลท์เตจลุคเอาท์" (UNDERVOLTAGE LOOKOUT) จะหยุดการทำงานของวงจรเอาท์พุทโดยอัตโนมัติ ถ้าหากว่าไฟเลี้ยง VCC1 มีค่าต่ำเกินไป คือต่ำกว่าค่าแรงดันที่กำหนดไว้ในวงจร UNDER VOLTAGE LOOKOUT

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3. รายละเอียดคุณสมบัติทางไฟฟ้าของ ไอ.ซี. LM621 :

Parameter	Conditions	Typ	Tested Limits	Design Limits	Units
DEAD-TIME SECTION					
High Level Input Voltage					
DIRECTION:	Pin 3 = 0V		2.0	2.0	V min
OUTPUT INHIBIT:			2.0	2.0	V min
DEAD-TIME ENABLE:	Pin 17 = 0V		2.0	2.0	V min
High Level Input Current	V_{in} = 5V				
DIRECTION:	Pin 3 = 0V		100	150	μA max
OUTPUT INHIBIT:			60	100	μA max
DEAD-TIME ENABLE:			200	300	μA max
Low Level Input Voltage					
DIRECTION:	Pin 3 = 0V		0.6	0.4	V max
OUTPUT INHIBIT:			0.6	0.4	V max
DEAD-TIME ENABLE:			0.3	0.2	V max
Low Level Input Current					
DIRECTION:	V_{in} = 0.6V		−100	−150	μA max
OUTPUT INHIBIT:	V_{in} = 0.6V		−60	−100	μA max
DEAD-TIME ENABLE:	V_{in} = 0V		−200	−300	μA max
Propagation Delays (Inputs to Outputs)	Dead-Time Off, (Pin 3 = 0V)				
OUTPUT INHIBIT		200			ns
DIRECTION		200			ns
Minimum Clock Period, T_{CLK} (Note 3)	R = 11 kΩ C = 200 pF	2.2			μs
Clock Accuracy f = 100 kHz	R = 30k C = 400 pF	±3			%
Minimum Dead-Time	Dead-Time Off	15			ns
Minimum Dead-Time	Dead-Time On	2			T_{CLK}
COMPLETE CIRCUIT					
Total Current Drains	Outputs Off				
I_{CC1}			10		mA min
I_{CC1}		15	22	30	mA max
I_{CC2}	V_{CC2} = 40V		2		mA min
I_{CC2}		3	6	9	mA max
Undervoltage Lockout V_{CC1}		3.6	3.0		V_{MAX}

Note 1. Unless otherwise noted ambient temperature (T_A) = 25°C.

Note 2. Unless otherwise noted: V_{CC1} = +5.0V, "recommended operating range V_{CC} = 4.5V to 5.5V" V_{CC2} = +10.0V, ambient temperature = 25°C.

Note 3. Clock oscillator period, T_{CLK} = RC, where T_{CLK} is in μs, R is in kΩ, and C is pF. Also see selection graph in Typical Characteristics for determining values of R and C. Note that the value of R should be no less than 11 kΩ and C no less than 200 pF.

Note 4. Tested limits are guaranteed and 100% production tested.

Note 5. Design limits are guaranteed (but not 100% production tested) at the indicated temperature and supply voltages. These limits are not used to calculate outgoing quality levels.

Note 6. Specifications in boldface apply over junction temperature range of −40°C to +85°C.

Note 7. Typical Thermal Resistances O_{JA} (see Note 8):

N pkg, board mounted 110°C/W

N pkg, socketed 118°C/W

Note 8. Package thermal resistance indicates the ability of the package to dissipate heat generated on the die. Given ambient temperature and power dissipation, the thermal resistance parameter can be used to determine the approximate operating junction temperature. Operating junction temperature directly effects product performance and reliability.

Note 9. This part specifically does not have thermal shutdown protection to avoid safety problems related to an unintentional restart due to thermal time constant variations. Care should be taken to prevent excessive power dissipation on the die.

Note 10: Human body model, 100 pF, discharged through a 1500Ω resistor.

รูปที่ 2.2. แสดงคุณสมบัติทางไฟฟ้าของ LM621

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Absolute Maximum Ratings (See Notes)

If Military/Aerospace specified devices are required, please contact the National Semiconductor Sales Office/Distributors for availability and specifications.

V_{CC1}	+7V
V_{CC2}	+45V
Logic Inputs (Note 1)	V_{CC1} +0.5V, −0.5V
Logic Input Clamp Current	20 mA
Output Voltages	+45V, −0.5V
Output Currents	Internally current limited

Operating Ambient Temperature Range LM621	−40°C to +85°C
Storage Temperature Range	−65°C to +150°C
Junction Temperature	150°C
ESD Susceptibility (Note 10)	2000V
Lead Temperature, N pkg. (Soldering, 4 sec.)	260°C

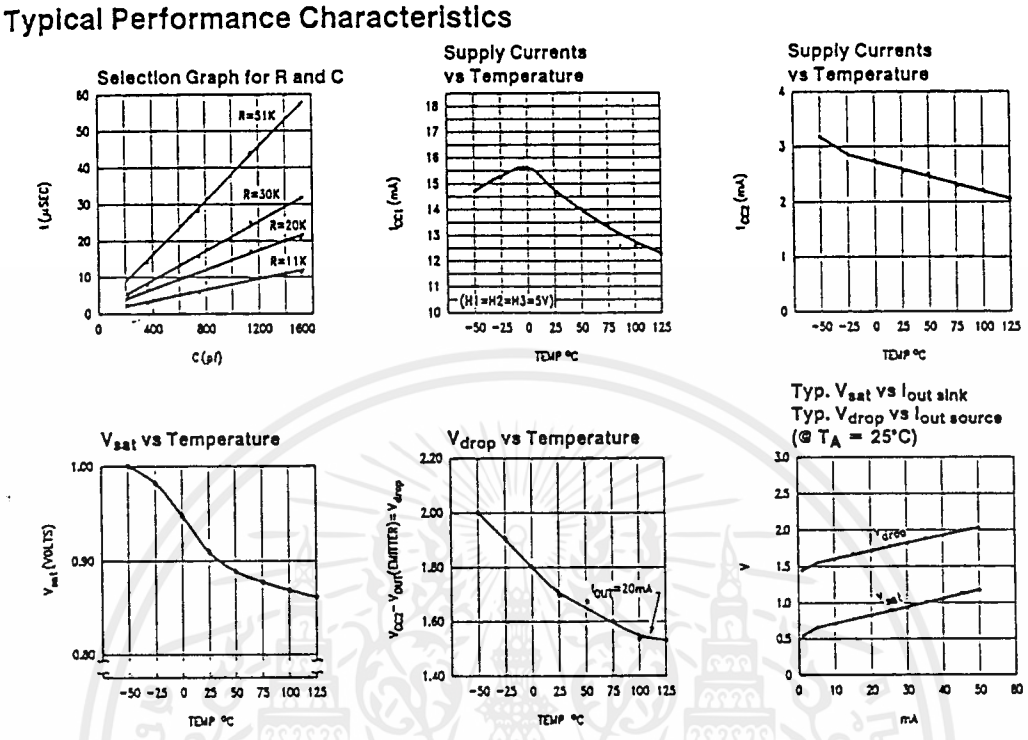
Electrical Characteristics (See Notes)

Parameter	Conditions	Typ	Tested Limits	Design Limits	Units
DECODER SECTION					
High Level Input Voltage HS1, HS2, HS3: 30/$\overline{60}$ SELECT:			 2.0 2.0	 **2.0** **2.0**	 V min V min
High Level Input Current HS1, HS2, HS3: 30/$\overline{60}$ SELECT:	 $V_{IH} = V_{CC1}$ $V_{IH} = V_{CC1}$	 	 100 120	 **200** **240**	 µA max µA max
Low Level Input Voltage HS1, HS3 and HS2 HS1, HS3 and HS2 30/$\overline{60}$ Select	 30/$\overline{60}$ = 5V 30/$\overline{60}$ = 0V $H_{S1} = H_{S3} = 5V$	 	 0.6 0.6 0.6	 **0.4** **0.4** **0.4**	 V max V max V_{max}
Low Level Input Current HS1 and HS3: HS2: 30/$\overline{60}$ SELECT	 $V_{IL} = 0.35V$ $V_{IL} = 0.4V$ $V_{IL} = 0.0V$	 	 −400 −100 −700	 **−600** **−200** **−1000**	 µA max µA max µA max
Input Clamp Voltage (Pins 2, 3, 5, 6, 7, 8, 17)	$I_{in} = 1$ mA $I_{in} = -1$ mA	$(V_{CC1} + 0.7)$ (−0.6)			V V
Output Leakage Current Sinking Outputs Sourcing Outputs	Outputs Off $V_{CC2} = 40V$, $V_{OUT} = 40V$ $V_{OUT} = 0V$	 0.2 −0.2	 	 1.0 −1.0	 µA µA
Short-Circuit Current Sinking Outputs Sourcing Outputs	$V_{CC2} = 10V$, $V_{OUT} = 10V$ $V_{OUT} = 0V$	50 −50	35 −35	 	mA min mA min
V_{sat} (sinking) V_{drop} (sourcing) = $(V_{CC2} - V_{OUT})$	$I = 20$ mA $I = -20$ mA	0.83 1.7	 	1.00 2.00	V max V max
Output Rise Time	(sourcing) $C_L < 10$ pF	50			ns
Output Fall Time	(sinking) $C_L \leq 10$ pF	50			ns
Propagation Delay (Hall Input to Output)	Dead-Time Off	200			ns

รูปที่ 2.2. (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4. ตัวอย่างคุณลักษณะความสามารถของ ไอ.ซี. LM621 :



รูปที่ 2.3. กราฟแสดงคุณสมบัติและความสามารถของ LM621

2.5. รายละเอียดการทำงานของ ไอ.ซี. LM621 :

ไอ.ซี. คอมมิวเตชั่นดีโค๊ดเดอร์ตัวนี้จะรับสัญญาณจาก "ฮอล-เซ็นเซอร์" ที่มาเข้าที่ขา HS1, HS2, และ HS3 พร้อมทั้งสัญญาณอินพุทที่ขา "30/60 SELECT" ซึ่งวจรภายในจะทำการดีโค๊ดสัญญาณที่เข้ามาเป็นลักษณะ "เกรย์–โค๊ด" (GRAY-CODE) ให้เป็นลำดับของสัญญาณที่จะไปขับมอเตอร์ตามที่ต้องการ

วงจร "เด็ดไทม์" เจนเนอเรเตอร์ ทำหน้าที่ตรวจสอบสัญญาณอินพุทที่เข้ามาที่ขา "ไดเร็คชั่น" (DIRECTION) และเป็นตัว "อินฮิบิท" (INHIBIT) สัญญาณเอาท์พุทที่จะปรากฏ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ที่ขา 11 ถึง 16 ในช่วงเวลานานพอที่จะสามารถป้องกันการเกิด "กระแสกระชาก" ในวงจรขับแบบสวิทช์ชิ่งซึ่งต่ออยู่ภายนอกเมื่อมีการเปลี่ยนทิศทางการหมุนของมอเตอร์

วงจรขับด้านเอาท์พุท 6 ชุดภายในไอ.ซี. ตัวนี้จะเป็นตัวจ่ายสัญญาณไปขับวงจรขับแบบสวิทช์ชิ่งที่ต่อเพิ่มเติมภายนอก ซึ่งทำหน้าที่ขับมอเตอร์ วงจรขับดังกล่าวนั้นแบ่งเป็น วงจรจ่ายกระแสเอาท์พุท (SOURCE CURRENT) 3 วงจรและวงจรรับกระแส (SINK CURRENT) อีก 3 วงจร วงจรขับทั้งหมดสามารถที่จ่ายกระแสได้สูงสุดถึง 50 มิลลิแอมป์ หรือจ่ายแรงดันปลี่ยนแปลงที่เอาท์พุทได้ถึง 40 โวลต์ เพื่อให้สามารถขับวงจรที่เป็น "ม่อสเฟท" (MOSFET) โดยที่วงจรลอจิกภายในตัวไอ.ซี. เองใช้ไฟเลี้ยงขนาด 5 โวลต์

วงจรตรวจสอบค่าแรงดันต่ำ (UNDERVOLTAGE LOOKOUT SECTION) ทำหน้าที่ตรวจสอบค่าแรงดันไฟเลี้ยง ถ้าค่าแรงดันที่วัดได้นั้นต่ำเกินกว่าที่จะทำให้วงจรลอจิกภายในทำงานได้อย่างมีประสิทธิภาพแล้ว วงจรนี้จะหยุดการทำงานของวงจรขับด้านเอาท์พุททันที

2.6. <u>การควบคุมการใช้งานของมอเตอร์แบบสามเฟส</u> :

มีวิธีการอยู่สองวิธี ที่ทั่วๆไปนิยมใช้ในการควบคุมความสัมพันธ์ของเฟสของตำแหน่งของโรเตอร์ ก็คือ การใช้วิธีติดตั้งตัวเซ็นเซอร์ควบคุมที่ตำแหน่ง 30 องศา และ 60 องศา ซึ่งตำแหน่งองศาดังกล่าวนั้นเป็นตำแหน่งที่ติดตั้งเซ็นเซ่อร์ ไม่ใช่ตำแหน่งองศาทางไฟฟ้า (เช่น เฟสของสัญญาณควบคุม เป็นต้น) ส่วนตำแหน่งองศาทางไฟฟ้าจะใช้ที่ 60 องศาในทั้งสองกรณี ตำแหน่งเฟสของสัญญาณควบคุมที่แตกต่างกันนั้น สามารถดูได้จากการเปรียบเทียบลำดับของสัญญาณ HS1 ถึง HS3 ดังแสดงในรูปที่ 2.4. ซึ่งแสดงถึง "ทรูธ เทเบิล" (TRUTH TABLE) ของวงจร LM621 คอมมิวเตชั่น ดีโค็ดเดอร์ มีทั้งแบบเฟส 30 และ 60 องศา รวมทั้งที่ 90 องศาสำหรับการใช้งานกับมอเต่อร์แบบ 4 เฟส

จากรูปที่ 2.4. แสดงถึงการกำหนดเฟส (หรือรหัส) ของเซ็นเซ่อร์ฮอลล์เอ็ฟเฟคในแต่ละ 60 องศา (ทางไฟฟ้า) ของย่านการหมุนของมอเตอร์ และความเกี่ยวพันกับสัญญาณชิงค์และซอสส์ ที่ใช้ในการขับวงขับแบบสวิทช์ชิ่งที่ต่อภายนอก ค่าเฟสต่างๆ ที่แสดงนี้สามารถใช้ได้กับมอเตอร์ของผู้ผลิตหลายราย จะเห็นว่าตำแหน่งการหมุนของโรเตอร์ใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แบบ 60 องศาจะเหมือนกับแบบ 30 องศา แต่ในแบบ 60 องศานั้นจะมีความปลอดภัยมากกว่า เนื่องจากไม่มีสัญญาณที่เป็น "ศูนย์" หรือ "หนึ่ง" ทุกบิทออกมาจากเซ็นเซ่อร์ โดยปกติสัญญาณหรือรหัสที่เป็น "ศูนย์" หรือ "หนึ่ง" ทุกบิทอาจสามารถเกิดจากการที่เซ็นเซ่อร์ไม่ถูกต่อกับวงจร หรือ ในทางกลับกันอาจเกิดจากเซ็นเซ่อร์เกิดลัดวงจร อย่างไรก็ตามรหัสของแต่ละเฟสตามที่แสดงในรูปที่ 2.4. นั้นอาจไม่สามารถใช้ได้กับมอเตอร์ของผู้ผลิตทุกราย ดังนั้นจึงมีรหัสอีกชุดหนึ่งถูกกำหนดขึ้นมาเพื่อที่จะครอบคลุมการใช้งานกับมอเตอร์ได้ทุกๆ แบบ ดังแสดงในรูปที่ 2.5. ซึ่งในตารางนี้จะแสดงถึงความแตกต่างของการกำหนดเฟสทั้งแบบ 60,120, 240, และ 300 องศา

TABLE I. LM621 Commutation Decoder Truth Table

Sensor Phasing	Position Range	Sensor Inputs			Sink Outputs			Source Outputs		
		HS1	HS2	HS3	1	2	3	1	2	3
30 deg	0–60	0	0	0	ON	off	off	off	ON	off
	60–120	0	0	1	ON	off	off	off	off	ON
	120–180	0	1	1	off	ON	off	off	off	ON
	180–240	1	1	1	off	ON	off	ON	off	off
	240–300	1	1	0	off	off	ON	ON	off	off
	300–360	1	0	0	off	off	ON	off	ON	off
60 deg	0–60	1	0	1	ON	off	off	off	ON	off
	60–120	1	0	0	ON	off	off	off	off	ON
	120–180	1	1	0	off	ON	off	off	off	ON
	180–240	0	1	0	off	ON	off	ON	off	off
	240–300	0	1	1	off	off	ON	ON	off	off
	300–360	0	0	1	off	off	ON	off	ON	off
90 deg	0–90	0	1	HS2	off	na	off	off	na	ON
	90–180	0	0	HS2	ON	na	off	off	na	off
	180–270	1	0	HS2	off	na	ON	off	na	off
	270–360	1	1	HS2	off	na	off	ON	na	off
Pin Numbers:		5	6	7	16	15	14	13	12	11

Note 1: The above outputs are generated when the Direction input, pin 2, is logic high. For reverse rotation (pin 2 logic low), the above sink and source output states become exchanged.

Note 2: For four-phase motors sink and source outputs number two (pins 15 and 12) are not used; hense the "na" (not applicable) in the appropriate columns above. *Figure 6* shows how the required sink and source outputs for four-phase motors are derived.

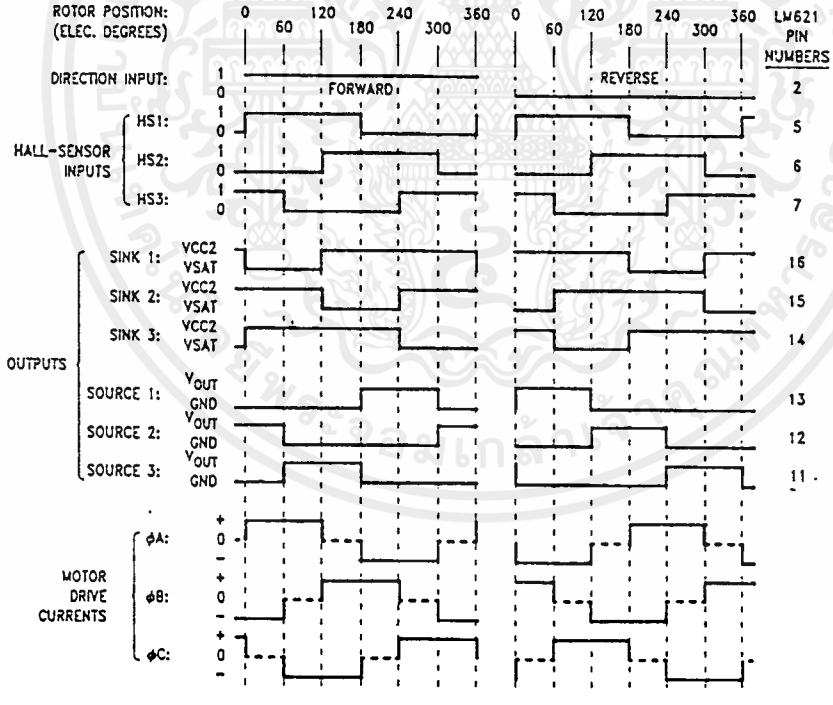
รูปที่ 2.4. ทรูธ เทเบิล ของ LM621 คอมมิวเตชั่น ดีโค็ดเดอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE II. Alternative Sensor-Phasing Names

Alternate Phasing	Position Range	Sensor Inputs			Corresponding LM621 Position Range and/or Comments
		HS1	HS2	HS3	
"60 deg"	0–60	0	0	0	Same as 30-degree phasing, but in reverse order; i.e., only change is relative direction.
	60–120	1	0	0	
	120–180	1	1	0	
	180–240	1	1	1	
	240–300	0	1	1	
	300–360	0	0	1	
"120 deg"	0–60	0	0	1	Same as 60-degree phasing, but with shifted order of position ranges; i.e., only change is relative phasing of sensor signals.
	60–120	1	0	1	
	120–180	1	0	0	
	180–240	1	1	0	
	240–300	0	1	0	
	300–360	0	1	1	
"240 deg"	0–60	0	1	0	Same comment as above for "120 deg" phasing.
	60–120	1	1	0	
	120–180	1	0	0	
	180–240	1	0	1	
	240–300	0	0	1	
	300–360	0	1	1	
"300 deg"	0–60	0	1	1	Same as 30-degree phasing, but with shifted order of position ranges, i.e., only change is relative phasing of sensor signals.
	60–120	1	1	1	
	120–180	1	1	0	
	180–240	1	0	0	
	240–300	0	0	0	
	300–360	0	0	1	

รูปที่ 2.5. ทรูธ เทเบิล ของคอมมิวเตชั่น ดีโค็ดเดอร์ แบบอื่น



รูปที่ 2.6. แสดงรูปคลื่นความสัมพันธ์ของสัญญาณอินพุท และ เอาท์พุท ของวงจรคอมมิวเตชั่น ดีโค็ดเดอร์ ใน LM621

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในรูปที่ 2.6. เป็นรูปคลื่นที่แสดงถึงความสัมพันธ์ของสัญญาณจากฮอลล์เอ็ฟเฟคเซ็นเว่อร์กับสัญญาณเอาท์พุทที่ได้จากวงจรคอมมิวเตเตอร์ ดีโค้ดเดอร์ ของมอเตอร์ที่มีเฟสการหมุนของโรเตอร์แบบ 60 องศา รวมทั้งสัญญาณที่ใช้ขับวงจรขับแบบสวิทช์ชิ่ง จะเห็นว่าสัญญาณจากฮอลล์เอฟเฟ็ค HS1 ถึง HS3 จะแตกต่างกันอยู่ 60 องศาทางไฟฟ้า ซึ่งอาจต้องการการแก้มุมการหมุนหากว่าใช้กับมอเตอร์แบบ 3 เฟส

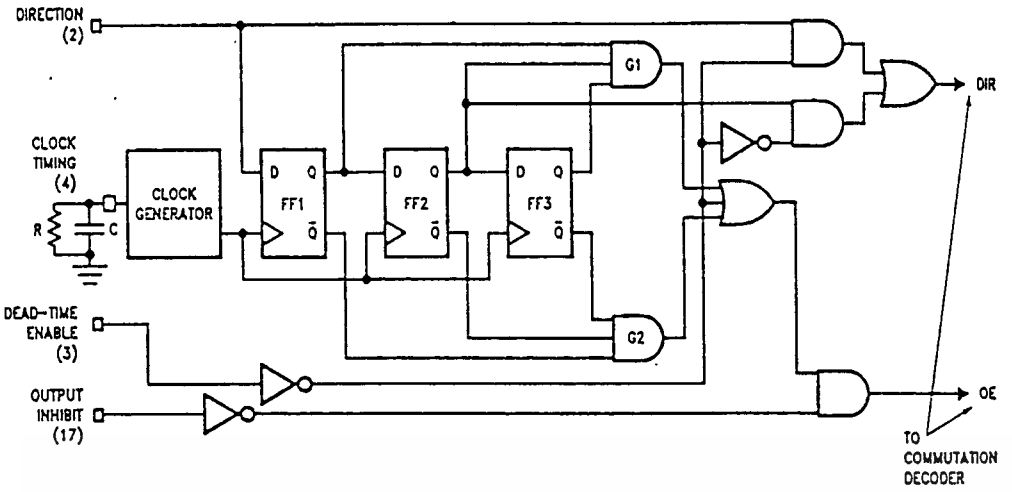
2.7. การใช้งานมอเตอร์แบบสี่เฟส :

โดยปกติแล้ว การควบคุมการทำงานของมอเตอร์แบบสี่เฟสนั้นจะใช้รหัสควบคุตำแหน่งโรเตอร์ที่เป็นแบบ 90 องศา ดังแสดงในรูปที่ 2.4. ซึ่งจะเห็นว่าสัญญาณจากฮอลล์เอฟเฟ็ค เซ็นเซอร์จะใช้เพียงสัญญาณจาก HS1 และ HS2 เท่านั้น เมื่อใช้ไอ.ซี. LM621 ควบคุมการทำงานของมอเตอร์แบบสี่เฟสแล้ว เราจะต่อสัญญาณ HS2 จากฮอลล์เอฟเฟ็คในตัวมอเตอร์เข้าทั้งขาอินพุท HS2 และ HS3 ของ LM621

2.8. ลักษณะของ Dead-Time :

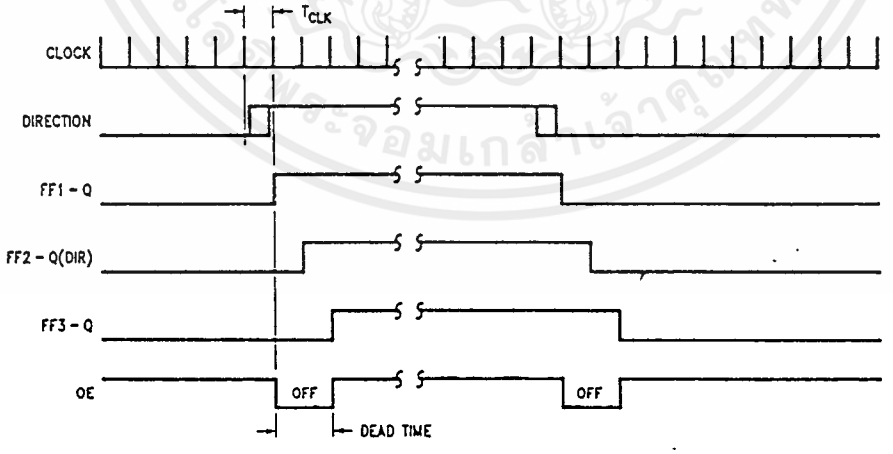
ขา Dead-Time enable จะถูกใช้อีเนเบิลการทำงานในแบบนี้ (โดยการต่อขา 3 เข้ากับไฟ +5 โวลต์) เหตุผลที่ต้องมีลักษณะการทำงานแบบนี้ก็เนื่องจากว่า โครงสร้างโดยทั่วไปของวงจรสวิทชิ่งกำลัง (Power Switching) นั้นเป็นแบบ Totem-pole ซึ่งโครงสร้างแบบนี้ ทำให้กระแสที่สวิทช์มีค่าสูงถ้าหากว่าอุปกรณ์ที่ประกอบอยู่ในวงจร Totem-pole ตัวใดตัวหนึ่งไม่สามารถหยุดทำงานได้อย่างสมบูรณ์ในขณะที่อุปกรณ์ที่อยู่คู่กันนั้นทำงานอย่างเต็มที่แล้ว ปรากฏการ์ณดังกล่าวจะทำให้เกิดกระแส Shoot-through จำนวนมากขึ้นในวงจร ทั้งนี้โดยปกติจะเกิดเหตุการ์ณดังกล่าวจะเกิดขึ้นเมื่อมีการเปลี่ยนแปลงลอจิกของสัญญาณที่ขา Direction (เมื่อสัญญาณเอาท์พุททุกๆ ขาถูกกลับชั้ว) ซึ่งช่วงเวลาดีเลย์ในการเปลี่ยนสภาวะจากการนำกระแสเป็นหยุดนำกระแสนี้ ก็สามารถทำให้เกิดกระแสที่ไม่ต้องการนี้ขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7. แสดงถึงลอจิกของวงจร เด็ดไทม์ เจนเนอเรเตอร์

จากรูปที่ 2.7. แสดงถึงวงจรลอจิกของวงจรเดดไทม์เจนเนอร์เรเตอร์ (DEAD-TIME GENERATOR) ซึ่งประกอบด้วยวงจร อาร์-ซี ออสซิลเลเตอร์ ที่ใช้ในการกำเนิดสัญญาณนาฬิกาที่เราต้องการ จากรูปที่ 2.7. นั้น ขา 4 (CLOCK TIMING) จะต่อกับตัวเก็บประจุและตัวต้านทานภายนอก เพื่อใช้ควบคุมความถี่ของวงจรกำเนิดสัญญาณนาฬิกา ซึ่งความถี่ของสัญญาณนาฬิกาดังกล่าว จะถูกปรับให้ความกว้างของสัญญาณในช่วงเวลาสองพีเรียด (ของสัญญาณนาฬิกา) มีค่ามากกว่าช่วงเวลา เทินออฟไทม์ (TURN-OFF TIME) ของอุปกรณ์สวิทชชิ่งเล็กน้อย



รูปที่ 2.8. แสดงรูปคลื่นที่จุดต่างๆ ในวงจร เดดไทม์ เจนเนอเรเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปกราฟที่แสดงไว้ในส่วนของ "ตัวอย่างของลักษณะคุณสมบัติ" จะเห็นว่าช่วงเวลาในหนึ่งพีเรียดของสัญญาณนาฬิกา (หน่วยเป็นไมโครเซคคั่น) จะมีค่าโดยประมาณเท่ากับค่าไทม์คอนสแตนท์ของค่า อาร์-ซี (R-C TIME CONSTANT) เมื่อ R มีค่าเป็น กิโลโอห์ม และ C มีค่าเป็น พิโคฟาราด) ซึ่งเราสามารถวัดค่าพีเรียดที่ขา 4 ด้วยออสซิลโลสโคป โดยทั่วไป วงจร เดดไทม์ เจนเนอเรเตอร์ จะเป็นตัวตรวจสอบการเปลี่ยนแปลงของสัญญาณที่ขา ไดเร็คชั่น (DIRECTION) ทำการซิงค์โครไนซ์ (SYNCHRONIZE) ทิศทางการหมุนที่เปลี่ยนแปลงกับสัญญาณนาฬิกาภายใน และอินอิบิท (INHIBIT) สัญญาณเอาท์พุทไม่ให้ปรากฏในช่วงเวลาสองพีเรียดของสัญญาณนาฬิกา ฟลิปฟลอป FF1 ถึง FF3 ประกอบกันเป็น ดีเลย์ไลน์ชิปรีจิสเตอร์ขนาด 3 บิท โดยรับสัญญาณอินพุทจากขาไดเร็คชั่น (ขา 2) วงจรฟลิปฟลอปนั้น เป็นส่วนที่ถูกกระตุ้นด้วยสัญญาณจากวงจรกำเนิดสัญญาณนาฬิกาภายในในการที่จะอิเนเบิลเกจ G1 หรือ G2 ให้ทำงาน สัญญาณเอาท์พุทที่ออกมาจากฟลิปฟลอปทั้งสามตัวจะต้องมีลอจิกที่เหมือนกัน เกจ G1 หรือ G2 ตัวใดตัวหนึ่งจะต้องถูกอิเนเบิลก่อนจึงจะมีสัญญาณ OE ไปอิเนเบิลให้มีสัญญาณเอาท์พุทออกจากตัวชิปได้ ทันทีที่สัญญาณการเปลี่ยนทิศทางการหมุนปรากฏที่ขา Q ของฟลิปฟลอป FF1 จะทำให้เกจ G1 และ G2 ถูกดิสเอเบิล ซึ่งเป็นผลให้วงจรขับแบบสวิทช์ซึ่งถูกดิสเอเบิลเป็นเวลาสองคร็อคพีเรียด

ช่วงเวลา "เดดไทม์" ถูกกำหนดให้เป็นช่วงเวลาที่สัญญาณเอาท์พุทของไอ.ซี. LM621 ไม่แอ็คทีฟ เพื่อที่จะป้องกันการเกิดกระแส "โอเวอร์ชูท" หลังจากที่มีการเปลี่ยนแปลงลอจิกของสัญญาณอินพุทที่ขาไดเร็คชั่น จากรูปที่ 3 จะเห็นได้ว่าช่วงเวลาเดดไทม์จะใช้เวลาถึงสองพีเรียดของสัญญาณนาฬิกา และเป็นผลทำให้เกิดการดีเลย์ขึ้นในลูปของระบบควบคุมการป้อนกลับ ซึ่งมีผลกระทบต่อขีดความสามารถและเสถียรภาพของวงจร ดังนั้นจึงเป็นสิ่งจำเป็นที่ต้องทำให้ช่วงเวลาเดดไทม์มีค่าน้อยที่สุด โดยเราจะเห็นได้ว่าช่วงเวลาตรงจุดที่สัญญาณไดเร็คชั่นเปลี่ยนแปลงจนถึงจุดที่สัญญาณ OE เปลี่ยนจากลอจิก "หนึ่ง" เป็นลอจิก "ศูนย์" นั้น สามารถที่จะเปลี่ยนแปลงได้ในช่วง 1 พีเรียด ทั้งนี้เนื่องจากธรรมชาติของสัญญาณที่เป็นแบบ "อะซิงโครนัส"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3.

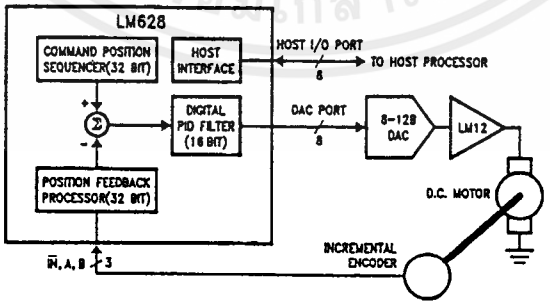
LM629 PRECISION MOTION CONTROLLER

3.0. รายละเอียดเบื้องต้น :

ไอ.ซี. เบอร์ LM629 เป็นโปรเซสเซอร์ที่ถูกออกแบบมาเพื่อใช้ในงานควบคุมการทำงานของมอเตอร์ไฟฟ้าหลายๆ แบบ ไม่ว่าจะเป็น ดี.ซี.เซอร์โวมอเตอร์ (D.C. Servo Motor) ดี.ซี.มอเตอร์แบบไม่มีแปรงถ่าน (Brushless D.C. Motor) หรืออุปกรณ์ทางด้านเซอร์โวแมคคานิคต่างๆ ที่มีการสร้างสัญญาณป้อนกลับของการเปลี่ยนตำแหน่งของโรเตอร์ในทุกๆ 90 องศา ไอ.ซี.เบอร์นี้จะทำหน้าที่ในการคำนวณในแบบ "เรียล-ไทม์ (Real-Time)" ซึ่งจำเป็นสำหรับงานควบคุมการเคลื่อนที่แบบดิจิตอลที่ต้องการประสิทธิภาพสูง

ซอร์ฟแวร์ที่ใช้ในการควบคุมการทำงานนั้นจะอาศัยคำสั่งที่จัดอยู่ในรูปของภาษาชั้นสูง (High-level Command Set) สัญญาณเอาท์พุทที่ได้จากชิปตัวนี้จะเป็นสัญญาณ PWM (Pulse Width Modulation) ที่มีขนาด 8-บิต ซึ่งเราสามารถที่จะนำเอาสัญญาณเอาท์พุทที่ได้นี้ไปป้อนให้กับวงจรขับแบบสวิทช์ชิ่งได้โดยตรง

ไอ.ซี.เบอร์นี้ถูกสร้างขึ้นโดยใช้เทคโนโลยีแบบ NMOS และจะบรรจุอยู่ในตัวถังแบบ DIP (Dual-in-Line Package) 28 ขา ซึ่งจะมีอยู่สองรุ่น คือ รุ่นที่ใช้ความถี่หลักเป็น 6 เมกกะเฮริท์ และรุ่น 8 เมกกะเฮริทซ์



รูปที่ 1. แสดงบล็อคไดอะแกรมของ LM629

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1. **คุณลักษณะที่สำคัญ** :

- o ภายในมีรีจิสเตอร์ขนาด 32 บิตอยู่หลายตัว เช่น Position register, Velocity register และ Acceleration register เป็นต้น
- o มีวงจร Digital PID Filter ที่โปรแกรมได้ขนาด 16 บิต
- o มีวงจร Programmable Derivative Sampling Interval
- o สามารถจ่ายสัญญาณเอาท์พุทเป็นสัญญาณแบบ PWM ขนาด 8 บิต
- o มีวงจรสร้างสัญญาณ Internal Trapezoidal Velocity Profile
- o สามารถเปลี่ยนแปลงค่าของ Velocity, Target Position, และ Filter parameter ได้ในขณะที่มอเตอร์กำลังทำงาน (หมุน) อยู่ได้
- o มีโหมดการทำงานทั้งในแบบ Position Mode และ Velocity Mode
- o สามารถโปรแกรมการอินเตอร์รัพท์ไปที่ Host Commputer ในแบบ Real-Time ได้
- o เชื่อมต่อกับ Host Computer หรือ Microprocessor ได้ในแบบ อะซิงค์โครนัส ขนาด 8 บิต
- o มีวงจร Quadrature Incremental Encoder Interface ซึ่งใช้เชื่อมต่อกับสัญญาณ Index Pulse

3.2. **รายละเอียดการใช้งานในแต่ละขาของ LM629** :

ขา	ชื่อ	รายละเอียด
1	Index Input (IN/)	เป็นขาอินพุทที่รับสัญญาณพัลส์ "อินเด็กซ์ (Index)" จากชุด Encoder สัญญาณ Index Position จะถูกอ่านเข้ามาเมื่อสัญญาณที่ขา 1,2, และ 3 เป็นลอจิก "ศูนย์" ขา IN/ นี้หากไม่ได้ใช้งานจะต้องต่อเข้ากับ VCC
2,3	Encoder Signal(A,	เป็นขาอินพุทที่ทำหน้าที่รับสัญญาณ Two-phase Quadrature ที่ถูกส่งมาจากส่วนของ "Incremental Encoder" เมื่อมอเตอร์หมุน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		ในทิศทางที่เป็นบวก (เดินหน้า) จะทำให้สัญญาณที่ขา 2 นำหน้าสัญญาณที่ขา 3 อยู่ 90 องศา โดยที่สัญญาณทั้งคู่จะต้องคงสภาวะในแต่ละสเตจของ Encoder อยู่อย่างน้อย 8 คาบสัญญาณนาฬิกา เพื่อที่จะสามารถรับรู้ตำแหน่งของโรเตอร์ ทั้งนี้เนื่องจากค่าความละเอียดที่เป็นแบบ 4 ต่อ 1 ที่เกิดจากวิธีการดีโค๊ดสัญญาณ Quadrature Encoder ซึ่งจะสัมพันธ์กับค่าสูงสุดของ Encoder State ที่ตรวจจับได้โดยมีค่าเป็น 1.0 เม็กกะเฮริท์ (ที่ความถี่ f_{clk}=8.0 เม็กกะเฮิรท์) หรือ 750 กิโลเฮิรท์ (ที่ f_{clk}=6.0 เม็กกะเฮิรท์) ซึ่งในกรณีที่ใช้ความถี่ของ f_{clk} แตกต่างไปจากนี้ สัญญาณจาก Encoder ก็จะต้องคงที่อย่างน้อย 8 คาบ เช่นกัน
4-11	Host I/O Port (D0-D7)	เป็นพอร์ทข้อมูลแบบ 2 ทิศทาง ซึ่งจะต่อเข้ากับ Host Computer หรือ Microprocessor พอร์ทเหล่านี้เป็นทางผ่านของทั้งข้อมูลและคำสั่งเข้า/ออก จากตัว LM629 ทั้งยังเป็นทางผ่านของข้อมูลและค่าสภาวะ (Status Byte) ที่อ่านจากรีจิสเตอร์ภายในของ LM629 โดยการควบคุมของสัญญาณที่ขา 12 (CS/) ขา 16 (PS/) ขา 13 (RD/) และขา 15 (WR/)
12	Chip Select(CS/	เป็นขาอินพุท ที่ใช้ในการเลือกว่าจะเขียน/อ่านข้อมูล/คำสั่งกับตัว LM629
13	Read(RD/ Input	เป็นขาอินพุท ใช้ในการอ่านข้อมูลและค่าสภาวะจากรีจิสเตอร์ภายในของ LM629
14	Ground	ขากราวด์นของไอ.ซี.
15	Write	เป็นขาอินพุท ใช้ควบคุมการเขียนคำสั่งและข้อมูลเข้าสู่ LM629

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

16	Port Select (PS/)	เป็นขาอินพุท ที่ใช้ในการเลือกพอร์ตว่าจะใช้ พอร์ตคำสั่งหรือพอร์ตข้อมูล หากสัญญาณที่ขานี้เป็นลอจิก "ศูนย์" ก็จะเป็นการเลือกใช้พอร์ตคำสั่ง หากเป็น "หนึ่ง" ก็เลือกพอร์ตข้อมูล โหมดของการเลือกพอร์ตมีดังนี้ 1. คำสั่งถูกเขียนเข้าไปที่พอร์ตคำสั่ง เมื่อขา 16 เป็น "ศูนย์" 2. ค่าสภาวะจะถูกอ่านออกมาจาก พอร์ตคำสั่งเมื่อขา 16 เป็น "ศูนย์" เช่นกัน และ 3. ข้อมูลจะถูกเขียนหรืออ่านผ่านทางพอร์ตข้อมูล เมื่อขา 16 เป็น "หนึ่ง"
17	Host Interrupt (HI) Output	เป็นขาเอาท์พุท แบบ Active "High" ที่จะถูกส่งไปยัง Host Computer (โดยทาง Host Interrupt Service Routine) หรือ Microprocessor (ที่ขา Interrupt Request) เพื่อบอกว่าเกิดการอินเตอร์รัพท์ขึ้น
18	Sign	เป็นขาเอาท์พุท ที่กำหนดทิศทางการหมุนของมอเตอร์
19	Magnitude	เป็นขาเอาท์พุท ที่กำหนดค่า อัตราความเร็ว (Velocity) และอัตราเร่ง (Acceleration) ของมอเตอร์
26	Clock	เป็นขาอินพุทที่รับสัญญาณนาฬิกาของระบบ (6 หรือ 8 เม็กกะเฮิรท์
27	Reset	เป็นขาอินพุทที่รับสัญญาณแบบ Active "LOW" Positive-edge Trigger โดยที่สัญญาณรีเซ็ทนี้จะต้องเป็นลอจิก "ศูนย์" เป็นเวลาอย่างน้อย 8 คาบสัญญาณนาฬิกา ซึ่งสัญญาณรีเซ็ทจะทำให้เกิด 1. ค่าพารามิเตอร์ของ Filter Coefficient กับ Trajectory จะมีค่าเป็น "ศูนย์"

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

		2. ค่า Position Error Threshold จะถูกเซ็ทให้มีค่าสูงสุด $(7FFF)_{10}$ และจะมีผลหากมีการเอ็กซิคิวส์ค่าสั่ง LPEI 3. อินเตอร์รัพท์ SBPA/SBPR จะถูกยกเลิก (Disable) 4. การอินเตอร์รัพท์ที่เหลืออีก 5 แบบจะถูกเอเนเบิลแทน 5. กำหนดตำแหน่งของโรเตอร์ที่เป็นอยู่ปัจจุบันเป็น "ศูนย์" หรือตำแหน่ง "โฮม"(Home position) 6. กำหนดค่า Derivative Sampling Internal เป็น $2048/f_{clk}$ หรือ 256 ไมโครเซ็กกั่น (สำหรับสัญญาณ f_{clk} = 8 เม็กกะเฮิรท์ หลังจากที่สัญญาณรีเซ็ทถูกปลดออกไปแล้ว ค่าจากพอร์ตสภาวะควรจะอ่านได้ "00_{10}" ถ้าการรีเซ็ทนั้นเสร็จสมบูรณ์แล้ว ค่าสภาวะจะเปลี่ยนไปเป็น "$(84)_{10}$" หรือ $(C4)_{10}$ ภายในเวลา 1.5 มิลลิเซ็คคั่น แต่ถ้าค่าสภาวะไม่เป็นไปตามนี้ ก็ให้รีเซ็ทซ้ำอีกครั้งหนึ่ง หรือเพื่อที่จะให้แน่ใจว่าการรีเซ็ททำได้ถูกต้อง ให้เอ็กซิคิวส์ค่าสั่ง RSTI ซึ่งถ้าการรีเซ็ทนั้นถูกต้อง ค่าสภาวะจะเปลี่ยนจาก $(84)_{10}$ หรือ $(C4)_{10}$ ไปเป็น $(80)_{10}$ แต่ถ้าไม่เป็นไปตามนี้ ก็ให้ทำการรีเซ็ทซ้ำ
28	Supply (VDD)	เป็นขาไฟเลี้ยง โดยมีแรงดัน +5 โวลต์

หมายเหตุ : สำหรับขา 20-25 นั้น ไม่ได้ใช้งาน

3.3. **ทฤษฎีการทำงาน :**

จาก Block Diagram ดังในรูปที่ 1 นั้นแสดงระบบเซอร์โวที่สร้างขึ้นจากโดยใช้ ไอ.ซี. LM629 เป็นตัวควบคุม โดย Host Computer จะติดต่อกับ LM629 ผ่านทาง I/O Port ในการที่จะโปรแกรมค่า Trapezoidal Velocity Profile และค่า Digital Compensation Filter สัญญาณเอาท์พุท PWM (Pulse Width Modulat

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ion) Sign กับ PWM Magnitude จะถูกส่งไปจ่ายเป็นอินพุทให้กับไอ.ซี. LM621 เพื่อทำการสร้างสัญญาณพัลส์ให้กับวงจรขับก่อนที่จะป้อนไปขับมอเตอร์

สัญญาณจากชุด Incremental Encoder จะเป็นสัญญาณป้อนกลับเพื่อให้ครบวงจรของ Position Servo Loop วงจร Trapezoidal Velocity Generator ทำหน้าที่คำนวณแนวทางเดินของโรเตอร์ที่ต้องการทั้งตำแหน่ง (Position) และอัตราความเร็วของการหมุน จากนั้น LM629 จะนำค่าตำแหน่งที่เป็นจริง (ตำแหน่งที่มีการป้อนกลับ) ไปลบกับตำแหน่งที่กำหนดโดยวงจร Profile Generator ซึ่งจะได้ค่าตำแหน่งที่ผิดพลาด (Position Error) ออกมาแล้วจึงนำไปจัดการต่อโดยวงจร Digital Filter เพื่อที่สร้างเป็นสัญญาณขับมอเตอร์ให้หมุนไปยังตำแหน่งที่ต้องการต่อไป

ตารางที่ 1 แสดงถึงคุณสมบัติโดยสรุปของไอ.ซี. LM628 และ LM629

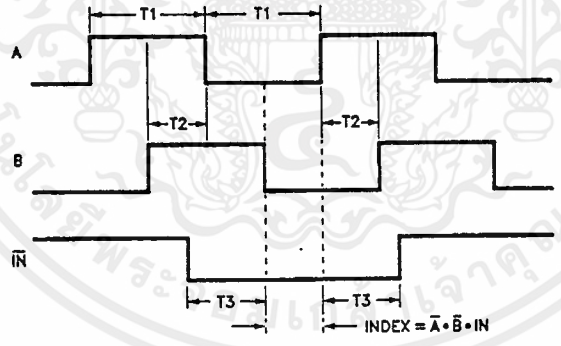
Position Range	−1,073,741,824 to 1,073,741,823 counts
Velocity Range	0 to 1,073,741,823/2^{16} counts/sample; ie, 0 to 16,383 counts/sample, with a resolution of $1/2^{16}$ counts/sample
Acceleration Range	0 to 1,073,741,823/2^{16} counts/sample/sample; ie, 0 to 16,383 counts/sample/sample, with a resolution of $1/2^{16}$ counts/sample/sample
Motor Drive Output	LM628: 8-bit parallel output to DAC, or 12-bit multiplexed output to DAC LM629: 8-bit PWM sign/magnitude signals
Operating Modes	Position and Velocity
Feedback Device	Incremental Encoder (quadrature signals; support for index pulse)
Control Algorithm	Proportional Integral Derivative (PID) (plus programmable integration limit)
Sample Intervals	Derivative Term: Programmable from $2048/f_{CLK}$ to $(2048 * 256)/f_{CLK}$ in steps of $2048/f_{CLK}$ (256 to 65,536 μs for an 8.0 MHz clock). Proportional and Integral: $2048/f_{CLK}$

ตารางที่ 1. สรุปคุณสมบัติของระบบที่ใช้ไอ.ซี. LM628/629

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1. <u>ระบบเชื่อมต่อในการป้อนกลับของตำแหน่งโรเตอร์ (Position Feedback Interface</u> :

ไอ.ซี. เบอร์ LM629 นี้จะทำการเชื่อมต่อกับตัวมอเตอร์ได้ โดยผ่านทางวงจร Incremental Encoder ซึ่งในการนี้จะมีขาอินพุทอยู่สามขาเตรียมเอาไว้ คือขา 1 (Index Pulse) ขา 2 กับ 3 (Quadrature Signal Input หรือ Encoder A/B) สัญญาณ Quadrature จะใช้ในการติดตามตำแหน่งที่แน่นอนของโรเตอร์ โดยที่แต่ละครั้งที่มีการเปลี่ยนแปลงทางด้านลอจิกในแต่ละด้านของสัญญาณที่ ขา 2 และ 3 แล้ว รีจิสเตอร์ Position ภายใน LM629 จะเพิ่มค่าหรือลดค่าตามการเปลี่ยนแปลงนั้น วิธีการนี้สามารถเพิ่มความละเอียดได้เป็นสี่เท่าของจำนวนสัญญาณที่ส่งออกมาจากชุด Encoder จากรูปที่ 9 นั้นแสดงถึงการที่สัญญาณ Encoder ซึ่งป้อนเข้าที่อินพุทของ LM629 จะซิงค์โครไนซ์กับสัญญาณนาฬิกาของ LM629 เอง



รูปที่ 2. ลักษณะรูปคลื่นของสัญญาณจาก Quadrature Encoder

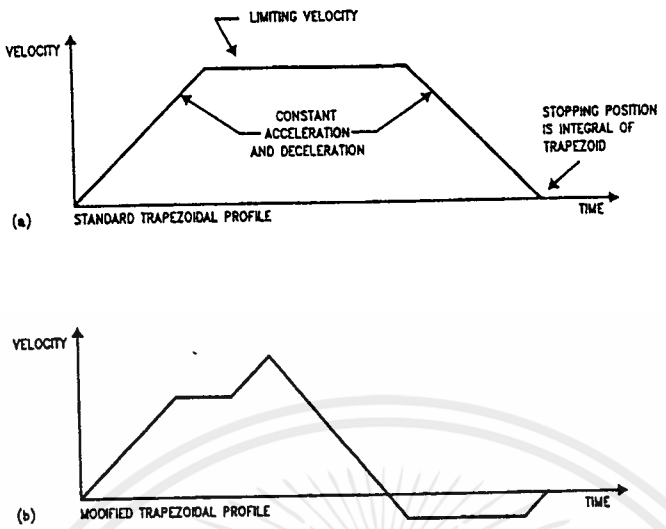
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สัญญาณ Index Pulse นั้นจะมีเฉพาะในชุด Encoder บางแบบเท่านั้นโดยกำหนดให้สถานะทางลอจิก "ศูนย์" ที่เกิดขึ้นในแต่ละครั้งแทนการหมุนครบแต่ละรอบการหมุนของโรเตอร์ แต่ถ้า LM629 ถูกโปรแกรมโดยผู้ใช้แล้ว ก็จะมีการบันทึกตำแหน่งที่เป็นจริงของมอเตอร์ไว้ในรีจิสเตอร์ Index เมื่อสัญญาณจาก Encoder ทั้งสามสัญญาณเป็นลอจิก "ศูนย์" พร้อมกันหมด ในกรณีที่ชุด Encoder ไม่มีสัญญาณ Index Pulse มาให้ เราก็สามารถใช้ขาอินพุท Index (ขา 1)ในการบันทึกตำแหน่งตำแหน่ง "Home" ของมอเตอร์ได้ โดยให้การหมุนของมอเตอร์นั้นไปทำให้สวิทช์บางอย่างทำงาน (หรืออาจใช้เซ็นเซอร์) ในอันที่จะทำให้เกิดมีสัญญาณลอจิก "ศูนย์" ป้อนที่ขา Index ในช่วงที่มอเตอร์อยู่ที่ตำแหน่ง Home ซึ่ง LM629 จะบันทึกตำแหน่งการหมุนของมอเตอร์เอาไว้ แล้วส่งสัญญาณไปทำการอินเตอร์รัพท์ Host Processor การใช้งานในแบบนี้จะต้องแน่ใจว่าสัญญาณ Index Pulse จะไม่เป็นลอจิก "ศูนย์" เมื่อมอเตอร์ไม่อยู่ในตำแหน่ง Home ทั้งนี้ก็เพราะว่าภายในตัว LM629 จะมีการสร้างสัญญาณอินเตอร์รัพท์ทุกๆ ครั้งที่สัญญาณที่มาเข้าที่ ขา 1,2 และ3 เป็นลอจิก "ศูนย์" ทั้งหมด ซึ่งจะทำให้ LM629 ไม่สามารถทำงานได้อย่างถูกต้อง หากว่า อัตราความเร็วในการหมุนของมอเตอร์นั้นมากกว่า 15,000 Count/$_{second}$ (สำหรับสัญญาณนาฬิกา 6.0 เม็กกะเฮิรท์) และ 20,000 Count/$_{second}$ (สำหรับสัญญาณนาฬิกา 8.0 เม็กกะเฮิรทซ์)

3.3.2. <u>การสร้าง Velocity Profile</u>
<u>[Velocity Profile (Trajectory) Generation]</u> :

ส่วนที่ทำหน้าที่สร้าง Trapezoidal Velocity Profile นั้นจะทำการคำนวณตำแหน่งของมอเตอร์ที่ต้องการกับค่าของเวลา ซึ่งเมื่อ LM629 ทำงานอยู่ในโหมด Position แล้ว ตัว Host Computer จะเป็นผู้กำหนด อัตราเร่ง, อัตราความเร็วสูงสุด, และ ตำแหน่งสุดท้ายของมอเตอร์ LM629 จะอาศัยข้อมูลดังกล่าวในการควบคุมการหมุนโดยทำการเร่งการหมุนของมอเตอร์จนกระทั่งถึงอัตราความเร็วสูงสุด หรือจนกระทั่งถึงจุดที่จะต้องเริ่มลดอัตราเร่งเพื่อให้มอเตอร์หยุดที่ตำแหน่งที่กำหนด อัตราส่วนในการลดอัตราเร่งจะเท่าๆ กับอัตราการเพิ่มอัตราเร่ง ณ เวลาใดๆ ก็ตามที่มอเตอร์กำลังหมุนอยู่ ค่าอัตราความเร็วสูงสุด หรือแม้แต่ตำแหน่งของการการหมุนที่ต้องการอาจเปลี่ยนแปลงได้ ซึ่งมอเตอร์จะถูกเพิ่ม/ลดอัตราเร่งตามนั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3. แสดงถึง Velocity Profiles

เมื่อทำงานในโหมด Velocity มอเตอร์จะมีอัตราเร่งเพิ่มขึ้นจนถึงอัตราความเร็วที่กำหนด เมื่ออัตราเร่งเพิ่มขึ้นจนถึงค่าที่กำหนดเช่นกัน และจะรักษาระดับความเร็วนั้นไปตลอดจนกว่าจะมีคำสั่งให้หยุด อัตราความเร็วนี้จะรักษาให้คงที่ได้ด้วยการเปลี่ยนตำแหน่งของมอเตอร์ให้เลื่อนออกไป โดยมีอัตราการเลื่อนคงที่ ซึ่งหากมีการรบกวนการหมุนของมอเตอร์ในขณะที่อยู่ในโหมด Velocity แล้ว ค่าเฉลี่ยในระยะยาวของอัตราความเร็วก็ยังคงที่อยู่ แต่ถ้าในกรณีที่มอเตอร์ไม่สามารถรักษาระดับความเร็วให้คงที่ ตามที่กำหนด (อาจมีสาเหตุจากแกนโรเตอร์ถูกล็อคอยู่ เป็นต้น) ทำให้ค่าตำแหน่งที่ต้องการมีการเพิ่มขึ้นอยู่ตลอด ทำให้มีค่าผิดพลาดเกิดขึ้นมาก หากสภาวะดังกล่าวหมดไป และ แรงที่หน่วงมอเตอร์เริ่มลดลงเรื่อยๆ มอเตอร์จะถูกเพิ่มอัตราความเร็วขึ้นสูงมากเพื่อที่จะไปให้ถึงตำแหน่งที่กำหนด (โดยที่ตำแหน่งดังกล่าวยังคงเลื่อนไปเรื่อยๆ ตามการทำงานของโหมดนั้น) อย่างไรก็ตาม สภาวะที่ว่านี้ก็สามารถตรวจสอบได้ไม่ยาก (ดูรายละเอียดของคำสั่ง LPEI และ LPES)

ค่าพารามิเตอร์ทั้งหมดที่เกี่ยวกับแนวทางการหมุนของมอเตอร์นั้น จะมีขนาด 32 บิต ซึ่งตำแหน่งของมอเตอร์จะกำหนดโดยค่าตัวเลขที่คิดเครื่องหมายด้วย ส่วนอัตราเร่งและอัตราความเร็วจะถูกกำหนดด้วยข้อมูลขนาด 16 บิต (ค่าจำนวนเต็มเป็นบวก และ มีเศษ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นค่าขนาด 16 บิต) โดยค่าจำนวนเต็มเป็นตัวกำหนดจำนวนนับ (Count) ในช่วงเวลาสุ่ม (Sampling) ที่มอเตอร์ทำงาน และค่าเศษเป็นตัวบอกถึงค่าเศษของการนับในช่วงการสุ่มนั้น แม้ว่าค่าความละเอียดของการกำหนดตำแหน่งโดย LM629 จะถูกจำกัดอยู่ด้วยค่านับที่เป็นจำนวนเต็มก็ตาม ยังมีค่านับที่เหลือเศษเป็นส่วนช่วยเพิ่มความละเอียดของอัตราความเร็วโดยเฉลี่ย ส่วนอัตราเร่งนั้นก็มีหลักการกำหนดเช่นเดียวกัน ซึ่งในแต่ละช่วงการสุ่มนั้น ค่าอัตราเร่งที่ถูกกำหนดไว้จะถูกนำไปรวมกับค่าอัตราความเร็วปัจจุบันเพื่อใช้ในการสร้างค่าอัตราความเร็วที่ต้องการใหม่ (เว้นแต่ว่า จะถึงอัตราความเร็วที่โปรแกรมกำหนดเสียก่อน)

สูตรที่ใช้ในการคำนวณเพื่อหาค่าพารามิเตอร์ที่เหมาะสม เพื่อป้อนให้กับ LM629 ในการควบคุมการทำงานของมอเตอร์มีดังนี้

การหาพารามิเตอร์ของตำแหน่งที่ต้องการให้มอเตอร์หยุดหมุน

P = R * จำนวนรอบของการหมุนที่ต้องการ

เมื่อ P = ตำแหน่งที่ต้องการให้มอเตอร์หยุดหมุน

R = System Resolution = Encoder Lines * 4

การหาพารามิเตอร์ของอัตราความเร็ว

V = R * T * C * ความเร็วรอบที่ต้องการ (RPM)

เมื่อ V = Velocity

R = System Resolution

T = Sample Time = 341 uS (สำหรับความถี่คร็อค 6 เมกกะเฮิรท์)

C = Conversion Factor = 1 นาที/60 วินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การหาค่าพารามิเตอร์ของอัตราเร่ง

A = R * T * T * อัตราเร่งที่ต้องการ (Rev/Sec/Sec)

เมื่อ R = System Resolution

T = Sample Time

3.3.3. PID (Proportinal Integral Derivation) Compensation Filter

ไอ.ซี. LM629 วิธีการของดิจิตอลฟิลเตอร์แบบ PID ในการชดเชยในวงจรคอลโทรลลูป มอเตอร์จะถูกตรึงไว้ตรงตำแหน่งที่ต้องการ โดยการจ่าย "แรงรักษาสภาพ" (Restoring Force) ให้กับมอเตอร์ โดยให้เป็นสัดส่วนกับตำแหน่งที่ผิดพลาดไป บวกกับค่าผิดพลาดรวม ค่าเบี่ยงเบนของการผิดพลาด ดังแสดงในสมการ Discrete-time เป็นการควบคุมโดยใช้ LM629

$$u(n) = kp * e(n) + ld \sum_{N=0}^{n} e(n) + kd[e(n') - e(n' - 1)]$$

เมื่อ u(n) คือ สัญญาณเอาท์พุทควบคุมมอเตอร์ในช่วงเวลาสุ่ม (n)

e(n) คือ ตำแหน่งที่ผิดพลาด ณ ช่วงเวลาสุ่ม (n)

n' แสดงถึงการสุ่ม ณ จุดที่มีการเปลี่ยนแปลงอัตราการสุ่ม

และ kp, ki, และ kd เป็นค่าพารามิเตอร์ของ Discrete-time Filter ที่ถูกโหลดโดยผู้ใช้

จากสมการ ในเทอมแรก (เทอม Proportional) เป็นการจัดค่าแรงดงสภาพที่เป็นสัดส่วน กับตำแหน่งที่ผิดพลาด (เหมือนกับการทำงานของขดลวดสปริง ตามกฎของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Hooke) ส่วนเทอมที่สอง (เทอม Integration) เป็นการจัดค่าแรงคงสภาพที่เพิ่มขึ้นตามเวลาและเพื่อให้แน่ใจว่าค่า Static Position Error นั้นเป็นศูนย์ หากว่ามีแรงบิดมากระทำคงที่แล้วมอเตอร์จะยังคงรักษาสภาพที่ค่าความผิดพลาดเป็นศูนย์ได้

เทอมที่สาม (เทอม Derivative) เป็นการจัดแรงที่เป็นอัตราส่วนกับอัตราการเปลี่ยนแปลงของตำแหน่งที่ผิดพลาด ซึ่งมันจะทำงานเหมือนกับอุปกรณ์กันกระเทือนในรถยนตร์ ซึ่งช่วงของการสุ่มจะสัมพันธ์กับค่าในเทอม Derivative ที่ผู้ใช้สามารถจะเลือกได้ ความสามารถอันนี้ทำให้ LM629 สามารถควบคุมอุปกรณ์พวก System Mechanical Time-condtant ได้ โดยการจัดหาการประมาณการของการการเปลี่ยนแปลงอย่างต่อเนื่องที่ดีกว่า โดยทั่วไปแล้ว ช่วงของการสุ่มที่ยาวมากขึ้นจะมีประโยชน์สำหรับการทำงานที่ใช้อัตราความเร็วต่ำ

ในการทำงานจริง โครงสร้างแบบฟิลเตอร์จะรับสัญญาณบอกความผิดพลาดขนาด 16 บิต จากจุดรวมของวงจรลูป โดยที่สัญญาณบอกความผิดพลาดนั้นจะมีจำนวนบิตเต็มที่ได้แค่ 16 บิต ทั้งนี้ก็เพื่อให้แน่ใจว่าจะสามารถคาดการณ์ถึงพฤติกรรมที่จะเกิดขึ้นได้ นอกจากจะถูกคูณด้วยค่าสัมประสิทธิ์การฟิลเตอร์ (kp)แล้ว ค่าผิดพลาดยังจะถูกบวกเข้ากับค่าผิดพลาดสะสมที่เกิดขึ้นก่อนหน้านั้น (เพื่อที่ว่าจะทำให้อยู่ในรูปของสัญญาณ Integral) และค่าผิดก่อนนั้นจะถูกนำมาลบออก โดยมีอัตราส่วนตามที่กำหนดจากช่วงเวลาการสุ่ม Derivative ตามที่กำหนด การคูณของฟิลเตอร์ทั้งหมดจะกระทำโดยมีขนาด 16 บิต และจะเอาผลลัพธ์ 16 บิตล่างมาใช้เท่านั้น

สัญญาณอินทิกัลจะมีทั้งหมด 24 บิต แต่จะนำมาใช้แค่ 16 บิตบนเท่านั้น เทคนิคการปรับแบบนี้ทำเราสามารถนำค่าสัมประสิทธิ์ ki มาใช้งานได้ (โดยมีความไวน้อยลง) ค่า 16 บิตดังกล่าวจะถูกเลื่อนไปทางขวาแปดตำแหน่งแล้วถูกคูณด้วยค่าสัมประสิทธิ์ของฟิลเตอร์ ki เพื่อจัดรูปสมการให้สามารถนำไปสร้างเป็นสัญญาณเอาท์พุท เพื่อควบคุมมอเตอร์ต่อไป ค่า แอ็ปโซลูท แม็คนิจูดของผลลัพท์ที่ได้จากการคูณนั้น จะถูกนำไปเปรียบเทียบกับค่าสัมประสิทธิ์ (il) และ จะได้ค่าแม็คนิจูด (คิดเครื่องหมาย) ที่เหมาะสมจ่ายเป็นสัญญาณควบคุมมอเตอร์

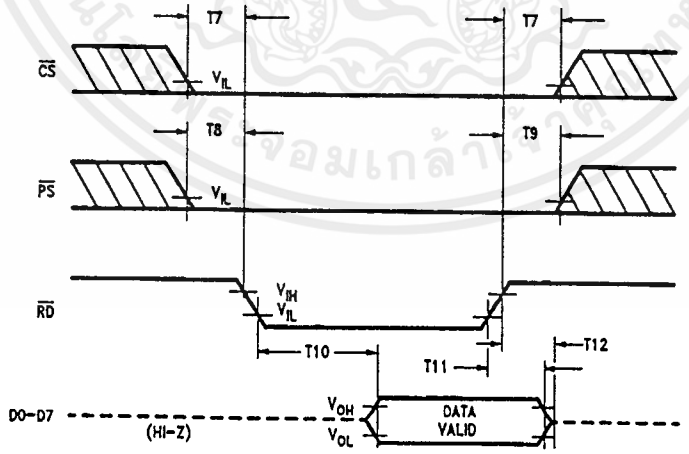
ค่าสัญญาณ Derivative ก็จะถูกคูณด้วยค่าสัมประสิทธิ์ (kd) ในแต่ละช่วงสุ่มแบบ Derivative ผลลัพท์ที่ได้ จะเป็นสัญญาณเอาท์พุทควบคุมมอเตอร์ในทุกๆ ช่วงการสุ่มโดยไม่ขึ้นอยู่กับช่วงการสุ่มที่ผู้ใช้กำหนด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่า kp, ki, และ kd จะถูกรวมเข้าด้วยกันเพื่อทำให้อยู่ในรูปของข้อมูลขนาด 16 บิต ทั้งนี้ขึ้นอยู่กับโหมดของการกำหนดค่าเอาท์พุทว่าจะใช้ข้อมูล 8 บิตบน หรือ 12 บิตบน ไปเป็นสัญญาณเอาท์พุท

3.3.4. การทำงานในสภาวะ อ่าน/เขียน ของ LM629 :

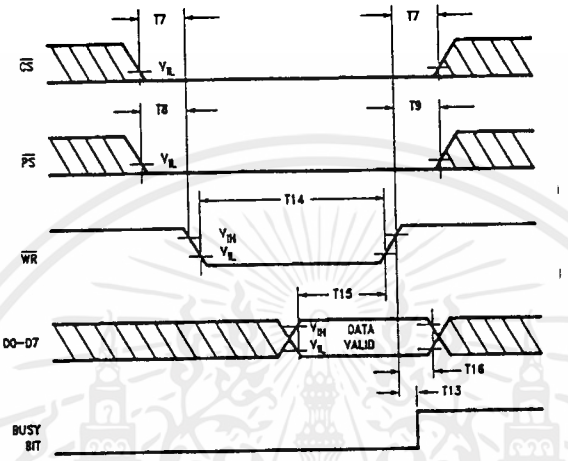
Host Computer จะเขียนคำสั่งเข้าไปยัง LM629 ได้โดยผ่านทางพอร์ท I/O ของ Host เมื่อสัญญาณ Post Select (PS/) ที่ขา 16 เป็นลอจิก "ศูนย์" ระหัสของคำสั่งที่ต้องการจะถูกส่งเข้าทางพอร์ทแบบขนาน โดยมีสัญญาณ Write (WR/) เป็นสโตรบ คำสั่งไบต์นั้นถูกแล็ทช์ไว้ใน LM629 ตรงช่วงขอบขาขึ้นของสัญญาณ WR/ ก่อนที่จะมีการเขียนคำสั่งเข้าสู่ LM629 จำเป็นต้องอ่านค่าสภาวะ เพื่อตรวจสอบสถานะของแฟ็ลกตัวหนึ่ง คือ "Busy bit (Bit 0)" หาก Busy bit เป็นลอจิก "หนึ่ง" ก็ไม่สามารถเขียนคำสั่งเข้าไปได้ Busy bit นี้จะเป็นลอจิก "หนึ่ง" อยู่ไม่เกิน 100 ไมโครเซ็กกั่น แต่โดยปกติจะอยู่ระหว่าง 15 ถึง 25 ไมโครเซ็กกั่น



รูปที่ 4. แสดงไทม์มิ่งของการอ่านค่าสภาวะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

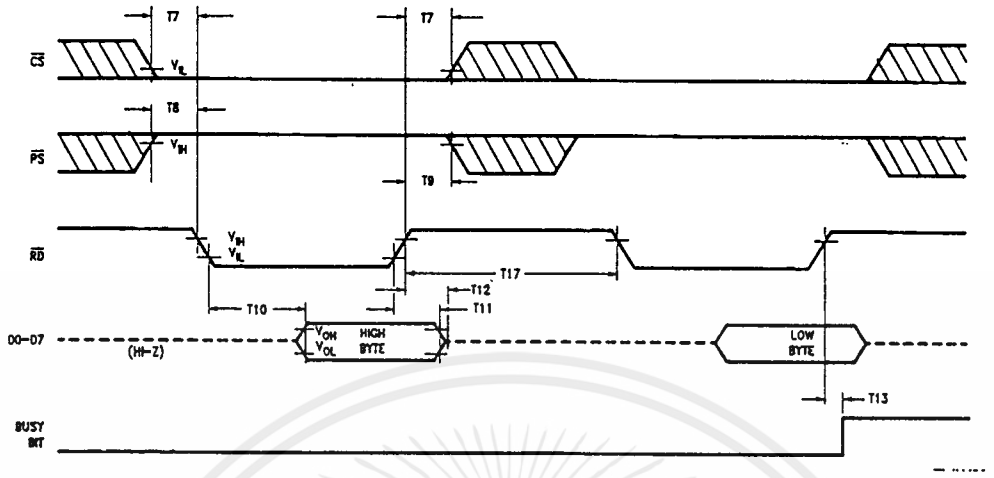
ในตอนอ่านค่าสภาวะก็จะเหมือนกัน คือ จ่ายสัญญาณสโตรปเข้าที่ขา 13 Read (RD/) เมื่อสัญญาณ PS/ ที่ขา 16 เป็นลอจิก "ศูนย์" ข้อมูลของค่าสภาวะจะยังคงมีอยู่ตราบเท่าที่สัญญาณ RD/ เป็นลอจิก "ศูนย์"



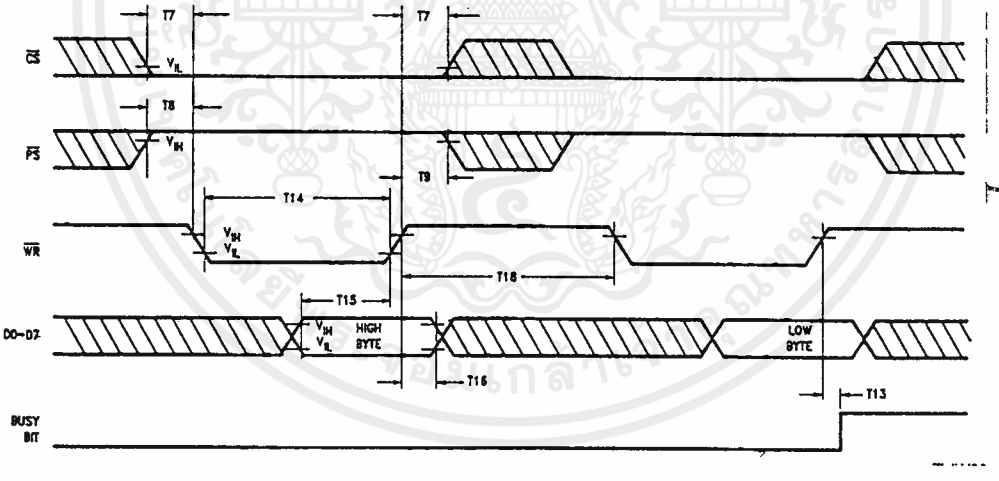
รูปที่ 5. แสดงช่วงเวลาในการเขียนคำสั่งเข้าสู่ตัว LM629

การ เขียน/อ่าน ข้อมูลเข้า/ออก กับ LM629 (ไม่ใช่การเขียน/อ่าน คำสั่งและค่าสภาวะ) จะทำได้โดยให้สัญญาณ PS/ ที่ขา 16 เป็นลอจิก "หนึ่ง" การเขียน/อ่าน ข้อมูลนี้จะเป็นข้อมูลที่มีจำนวน "อินทิกัล" (หนึ่ง ถึง เจ็ด) ของข้อมูลที่เป็น Two-byte Word โดยที่ข้อมูลในไบต์แรกของแต่ละเวิรด์จะมีนัยสำคัญมากกว่า ข้อมูลแต่ละไบต์จะต้องมีสัญญาณ Write (WR/) หรือ Read (RD/) เป็นสโตรป ก่อนที่จะส่งผ่านข้อมูลในแต่ละเวิรด์ (2 ไบต์) จำเป็นต้องมีการตรวจสอบค่าสภาวะและสถานะของ Busy bit ก่อนเสมอ ถ้า Busy bit เป็นลอจิก "ศูนย์" จึงจะสามารถเขียนข้อมูลเข้าสู่ LM629 ได้ ซึ่งหากมีการเขียนข้อมูลเข้าไปในขณะที่ Busy bit เป็นลอจิก "หนึ่ง" จะทำให้ข้อมูลนั้นถูกยกเลิก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 6. แสดงช่วงเวลาในการอ่านข้อมูลจากตัว LM629



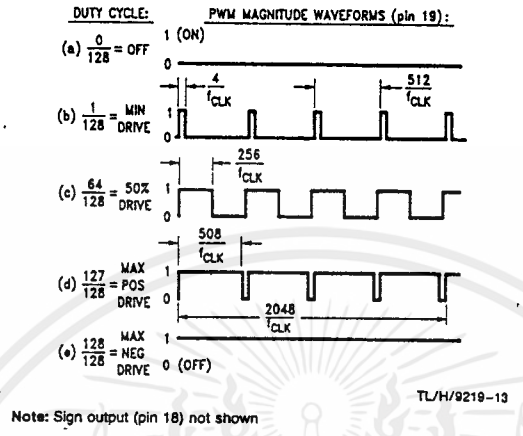
รูปที่ 7. แสดงช่วงเวลาในการเขียนข้อมูลเข้าสู่ตัว LM629

Busy bit จะถูกเซ็ทให้เป็นลอจิก "หนึ่ง" ทันทีที่มีการเขียนค่าสั่ง หรือมีการเขียน/อ่านข้อมูลไบต์ที่สองของข้อมูลแต่ละเวิรด์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.5. สัญญาณเอาท์พุทของ LM629 :

ไอ.ซี. LM629 จะให้สัญญาณเอาท์พุทเป็นสัญญาณ PWM (Pulse Width Modulation) สองสัญญาณ คือ SIGN และ MAGNITUDE ขนาด 8 บิต ซึ่งนำไปจ่ายให้กับวงจรขับมอเตอร์ แบบสวิทช์ชิ่งได้โดยตรง



รูปที่ 8. แสดงลักษณะของสัญญาณเอาท์พุทของ LM629

3.3.6. สรุปคำสั่งที่ใช้กับ LM629 :

คำสั่ง	ประเภท	รายละเอียด	หมายเหตุ
RESET	Initialize	รีเซ็ท LM629	1
PORT8	Initialize	เลือกเอาท์พุทขนาด 8 บิต (ของ LM628)	2
PORT12	Initialize	เลือกเอาท์พุทขนาด 12 บิต (ของ LM628)	2
DFH	Initialize	กำหนดตำแหน่ง Home ของมอเตอร์	1
SIP	Interrupt	เซ็ทตำแหน่งของ Index	1
LPEI	Interrupt	สร้างสัญญาณอินเตอรรัพท์เมื่อมี Error	1
LPES	Interrupt	ให้หยุดเมื่อมี Error	1
SBPA	Interrupt	กำหนด Breakpoint ด้วยค่าแอ็ปโซลุท	1
SBPR	Interrupt	กำหนด Breakpoint ด้วยค่าสัมพัทธ์	1
MSKI	Interrupt	ยกเลิกการเกิดอินเตอร์รัพท์	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

RSTI	Interrupt	รีเซ็ทการเกิดอินเตอร์รัพท์	1
LFIL	Filter	โหลดค่า "ฟิลเตอร์ พารามิเตอร์"	1
UDF	Filter	แก้ไขค่าฟิลเตอร์ให้เป็นค่าปัจจุบัน	1
LTRJ	Trajectory	โหลดค่า Trajectory	1
STT	Trajectory	เริ่มต้นการทำงานของมอเตอร์	3
RDSTAT	Report	อ่านค่าสภาวะ	1,4
RDSIGS	Report	อ่านค่าจากรีจิสเตอร์ Signal	1
RDIP	Report	อ่านค่าตำแหน่งของ Index	1
RDDP	Report	อ่านค่าตำแหน่งของโรเตอร์ที่ต้องการ	1
RDRP	Report	อ่านค่าตำแหน่งของโรเตอร์ที่เป็นจริง	1
RDDV	Report	อ่านค่าอัตราความเร็วที่กำหนดไว้	1
RDRV	Report	อ่านค่าอัตราความเร็วที่เป็นจริง	1
RDSUM	Report	อ่านค่าผลรวมของการอินทิเกรท	1

หมายเหตุ :

1. คำสั่งที่สามารถเอ็กซิคิวส์ในขณะที่มอเตอร์กำลังทำงานอยู่
2. คำสั่งที่ไม่สามารถเอ็กซิคิวส์ได้ในขณะที่มอเตอร์กำลังทำงานอยู่
3. คำสั่งที่อาจจะเอ็กซิคิวส์ได้ในขณะที่มอเตอร์กำลังทำงานอยู่ **แต่ค่าอัตราเร่งจะต้องไม่มีการเปลี่ยนแปลง**
4. คำสั่งที่ไม่ต้องมีระหัส เนื่องจากการอ่านค่าสภาวะนั้นสนับสนุนโดย Hardware โดยตรง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การควบคุมการทำงานของ LM629 โดยใช้ไมโครคอนโทรลเลอร์

4.1. แนวความคิดพื้นฐาน :

เนื่องจากการใช้งานตัวไอ.ซี. LM629 นั้น จะต้องมีการป้อนคำสั่งและข้อมูลต่างๆ หลายประการ ดังนั้นการที่จะควบคุมให้ LM629 ทำงานได้โดยตัวของมันเองจึงเป็นไปได้ยาก ในระบบควบคุมมอเตอร์ขนาดเล็กนั้น เราสามารถที่จะนำเอาไมโครคอนโทลเลอร์ หรือไมโครโปรเซ็สเซอร์ หรือไมโครคอมพิวเตอร์ มาควบคุมการทำงานของไอ.ซี.ตัวนี้ได้

โครงงานนี้ก็ใช้ 8031 ไมโครคอนโทลเลอร์มาเป็นตัวส่งผ่านทั้งข้อมูลและคำสั่งที่ป้อนเข้ามาทางไมโครคอมพิวเตอร์ คำสั่ง (Command) และข้อมูล (Data) ที่จะใช้ในการควบคุมการทำงานของ LM629 นั้น จะถูกส่งมาจากไมโครคอมพิวเตอร์ โดยผ่านทางพรินเตอร์พอร์ท เข้าสู่วงจรเชื่อมต่อซึ่งประกอบด้วย 8031 ไมโครคอนโทลเลอร์ และ 8255 PPI (Programmable Peripheral Interface) จากนั้น 8031 ก็จะส่งผ่านข้อมูล/คำสั่งนั้นไปให้กับ LM629 ต่อไป นอกจากนั้นแล้ว 8031 จะยังเป็นตัวควบคุมและติดตามสภาพการทำงานของ LM629 ตลอดเวลา ทั้งยังเป็นตัวกำหนดค่าพารามิเตอร์ต่างๆ ของ LM629 ในสภาวะเริ่มต้นอีกด้วย

4.2. การทำงานของชุดเชื่อมต่อ 8031 :

จากวงจรในหน้าที่ 65 นั้น ข้อมูล/คำสั่งจะถูกส่งมาเข้าที่ J1 โดยจะมีสัญญาณ STROBE/ เป็นสัญญาณบอกแก่ 8031ว่ามีข้อมูลมาปรากฏที่อินพุท เราใช้สัญญาณ STROBE/ นี้ไปเป็นสัญญาณ "อินเตอร์รัพท์" ให้แก่ 8031 เพื่อทำการอ่านข้อมูลเข้ามา โดยผ่านทาง 8255 ซึ่งถูกกำหนดให้ พอร์ท A เป็น "อินพุท พอร์ท" และพอร์ท B กับ C เป็น "เอาท์พุท พอร์ท" ข้อมูลดังกล่าวจะถูกอ่านผ่านทางพอร์ท A เข้าสู่บัสระบบ (System Bus) จากนั้นข้อมูลก็จะถูกโหลดเข้าสู่ LM629 ผ่านทางบัสระบบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3. การกำหนดแอดเดรส :

แอดเดรสของอุปกรณ์ หรือไอ.ซี. ต่างๆ ที่ประกอบอยู่ในระบบเชื่อมต่อ เป็นดังนี้

LM629	รีจิสเตอร์สภาวะ (Status Regiater)	2000H
	รีจิสเตอร์คำสั่ง (Command Register)	2000H
	รีจิสเตอร์ข้อมูล (Data Register)	2001H
8255	พอร์ท A	4000H
	พอร์ท B	4001H
	พอร์ท C	4002H
	พอร์ทควบคุม (Control Port)	4003H

4.4. คำสั่งที่ใช้ควบคุมการทำงานของ LM629 :

คำสั่งที่เตรียมเอาไว้ในโครงงานนี้มีดังนี้

คำสั่ง	รายละเอียด	ระหัส	ข้อมูล-ติดตาม
START	ใช้ควบคุมให้มอเตอร์เริ่มต้นทำงาน	01H	ไม่ต้องการ
VELOCITY	กำหนดอัตราความเร็วในการหมุนของ	02H	4 Byte
ACCELERATION	กำหนดอัตราเร่งของมอเตอร์	03H	4 Byte
POSITIONING	กำหนดตำแหน่งที่จะให้มอเตอร์หยุดหมุน	04H	4 Byte
STOP	ให้มอเตอร์หยุดทำงาน	05H	ไม่ต้องการ
REVERSE	กำหนดทิศทางการหมุนของมอเตอร์	06H	ไม่ต้องการ
FORWARD	กำหนดทิศทางการหมุนของมอเตอร์	07H	ไม่ต้องการ

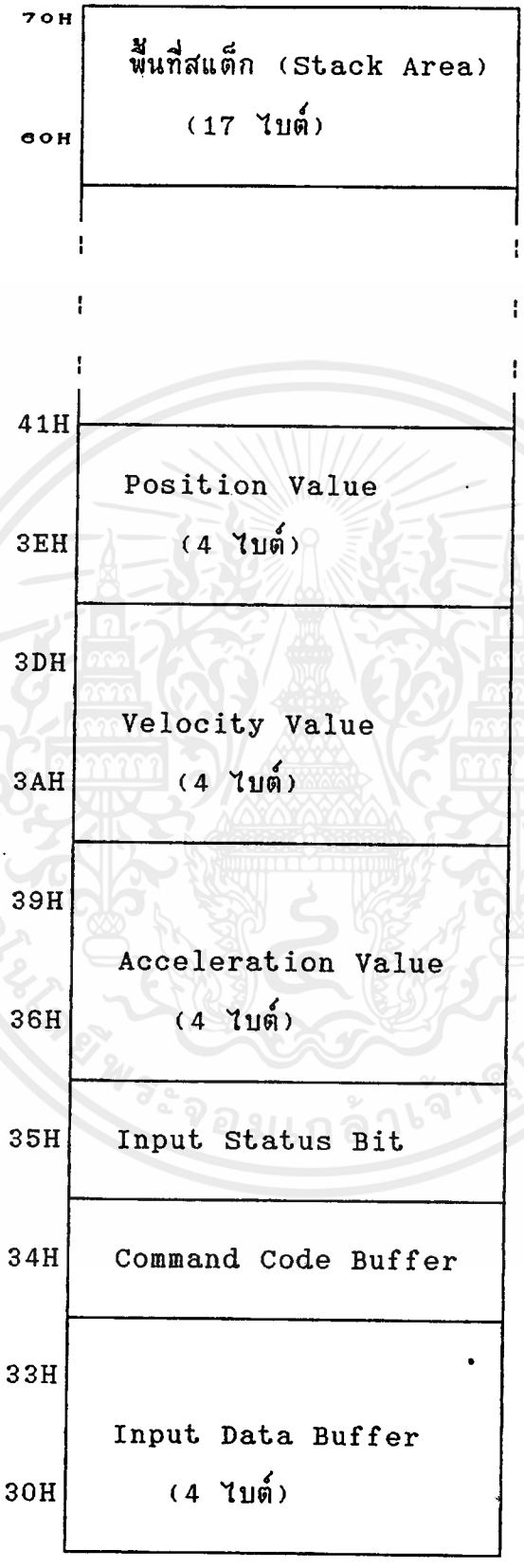
ในกรณีที่คำสั่งนั้นต้องมีข้อมูลติดตาม ก็จะมีรูปแบบดังนี้

ระหัสคำสั่ง	ไบต์ที่มีนัยสูงสุด	ไบต์ที่สอง	ไบต์ที่สาม	ไบต์ที่มีนัยต่ำสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.5. การจัดหน่วยความจำภายในของ 8031 :

การจัดหน่วยความจำ Internal RAM ที่อยู่ภายในตัว 8031 นั้นเป็นดังนี้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Input Data Buffer

เป็นพื้นที่ที่ใช้เก็บข้อมูลชั่วคราวขนาด 4 ไบต์ ในกรณีที่คำสั่งนั้นๆ จำเป็นต้องมีข้อมูลเพิ่มเติม โดยไบต์ที่มีนัยสูงสุดจะอยู่ที่แอดเดรส 30H และไบต์ที่มีนัยต่ำสุดอยู่ที่แอดเดรส 33H

Command Code Buffer

เป็นพื้นที่ขนาด 1 ไบต์ ใช้เก็บระหัสคำสั่งที่ส่งมาจากไมโครคอมพิวเตอร์

Input Status Bit

มีขนาด 1 ไบต์ ใช้เก็บค่าสภาวะต่างๆ ของการส่งผ่านข้อมูลกันระหว่างไมโครคอมพิวเตอร์กับคอนโทรลเลอร์ ซึ่งมีรูปแบบการจัดเก็บดังนี้

X	X	4th Data	3rd Data	2nd Data	1st Data	Direction	Command Valid
บิตที่ 7	6	5	4	3	2	1	0

บิต "0" (Command Valid) เป็น "1" หากข้อมูลที่ส่งมาจากไมโครคอมพิวเตอร์ไบต์นั้นเป็นคำสั่งควบคุมของ LM629 และ เป็นตัวบอกว่ามีคำสั่งเข้ามารอการโปรเซ็สแล้ว

บิต "1" (Direction) เป็น "1" เมื่อทิศทางการหมุนของมอเตอร์ในขณะนั้นเป็น "Forward Direction" และมีสภาวะเป็น "0" เมื่อทิศทางการหมุนเป็น "Reverse Direction"

บิต "2" (1st Data) มีสภาวะเป็น "1" เมื่อมีข้อมูลที่เป็นค่าพารามิเตอร์ที่ป้อนให้กับ LM629 นำไปใช้ และจะมีสภาวะเป็น "0" หากไม่มีข้อมูลพารามิเตอร์ส่งเข้ามา

บิต "3" (2nd Data), บิต "4" (3rd Data), บิต "5" (4th Data) จะมีสภาวะเป็น "1" เมื่อมีข้อมูลของค่าพารามิเตอร์ไบต์ที่ 2 หรือ 3 หรือ 4 ถูกส่งเข้ามา และจะมีสภาวะเป็น "0" หากไม่มีค่าพารามิเตอร์ไบต์ที่กำหนดส่งเข้ามา

บิต "6" และ "7" ไม่ได้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Accelaration Value (แอดเดรส 36H - 39H)

เป็นที่เก็บค่าพารามิเตอร์ที่ใช้ในการกำหนดอัตราเร่งของการหมุนของมอเตอร์ ข้อมูลในส่วนนี้จะถูกลบทิ้งเมื่อคำสั่ง Acceleration ถูกส่งไปให้กับ LM629 เรียบร้อยแล้ว

Velocity Value (แอดเดรส 3AH - 3DH)

ใช้เก็บค่าพารามิเตอร์ของการกำหนดอัตราความเร็วการหมุนของมอเตอร์

Position Value (แอดเดรส 3EH - 41H)

ใช้เก็บค่าพารามิเตอร์ของการกำหนดตำแหน่งที่จะให้มอเตอร์หยุดหมุน ค่านี้จะถูกไปใช้ก็ต่อเมื่อมีการกำหนดให้ LM629 ทำงานในโหมด Position หากกำหนดให้ LM 629 ทำงานในโหมด Velocity ก็จะไม่มีผล

หมายเหตุ: สำหรับโครงงานที่ทำอยู่นี้ ได้กำหนดให้ LM629 ทำงานในโหมด Velocity ดังนั้นจึงไม่ได้ใช้ค่านี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;***** Program LM629 Controller ******** Rev. 1.01/1993 *************
;***** This program is initially created by:
;       Rawat Khantrum and Wiboon Pichaiya
;***** which is used to communicate with IBM PC and control the operation
;***** of an I.C. LM629 Precision Motion Controller per details in
;***** "D.C. Brushless Motor Education's Kit" project (SGT11).
;
;-------- 8255's Port Addresses -----------------------------------
;
CONTROL     EQU     4003H       ;8255 Control Port Address
PORT_A      EQU     4000H       ;8255 Port "A" Address
PORT_B      EQU     4001H       ;8255 Port "B" Address
PORT_C      EQU     4002H       ;8255 Port "C" Address
CON_WD      EQU     82H         ;8255 Control word=Port A,C-> Input
                                ;                Port B  -> Output
;
;-------- LM629's Port Addresses ----------------------------------
;
COMMD       EQU     2000H       ;LM629 Command Port address
STATUS      EQU     2000H       ;LM629 Status Byte address (same as
                                ;command port address)
DATA        EQU     2001H       ;LM629 Data Port address
;
;-------- LM629's Command Code ------------------------------------
;
RESET       EQU     00H         ;Software RESET.
DFH         EQU     02H         ;Define Home position.
SIP         EQU     03H         ;Set INDEX Position.
LPEI        EQU     1BH         ;Load Position Error for Interrupt
                                ;**Need two consequence Data Bytes
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                                ;  range 0000-7FFF.*****
LPES        EQU     1AH         ;Load Position Error for Stopping
                                ;**Need two Data Bytes 0000-7FFFH
SBPA        EQU     20H         ;Set Break Point (Absolute) .Need 4 Data
                                ;Bytes 0C0000000 - 3FFFFFFF.
SBPR        EQU     21H         ;Set Break Point (Relative). Need as SBPA.
MSKI        EQU     1CH         ;Mask Interrupt. Need 2 Data bytes.
RSTI        EQU     1DH         ;Reset Interrupt. Need 2 Data bytes.
LFIL        EQU     1EH         ;Load Filter Parameter. Need 2-10 data
                                ;bytes.
UDF         EQU     04H         ;Update Filter.
LTRJ        EQU     1FH         ;Load Trajectory parameter. Need 2-14
                                ;data bytes.
STT         EQU     01H         ;Start Motion Control
RDSIGS      EQU     0CH         ;Read Signals register.
RDIP        EQU     09H         ;Read Index Position.
RDDP        EQU     08H         ;Read Desired Position.
RDRP        EQU     0AH         ;Read Real Position.
RDDV        EQU     07H         ;Read Desired Velocity.
RDRV        EQU     08H         ;Read Real Velocity.
RDSUM       EQU     0DH         ;Read Integration-Term Summation value.
;
;-------------- Variable declaration --------------------------------
;
START_BF    EQU     30H          ;Start address of Input Data Buffer.
START_AC    EQU     36H          ;Start address of Accelation Data Storage.
START_PS    EQU     3EH          ;Start address of Positioning Data Storage.
START_VL    EQU     3AH          ;Start address of Velocity Data Storage.
INPUTBIT    EQU     35H          ;Status bit of Command Code and Previous
                                 ;direction.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CODEAREA    EQU     34H             ;Command Code storage area.
;
;--------------- Command Code declaration ----------------------------
;
START_M     EQU     01H             ;Motor starting command code.
VELOCITY    EQU     02H             ;Velocity values entry.
ACC_CODE    EQU     03H             ;Acceleration values entry.
POS_CODE    EQU     04H             ;Positioning values entry.
STOP_M      EQU     05H             ;Motor stopping command code.
RDIR_CODE   EQU     06H             ;Reverse direction command code.
FDIR_CODE   EQU     07H             ;Forward direction command code.
;
;--------------- Program Start ----------------------------------------
;
                org     0000H
                ljmp    INITIAL
                org     0003H
                ljmp    Host_int
                org     0013H
                ljmp    PC_link

                org     0024H

INITIAL:
                mov     sp,#60H             ;Set Stack Pointer.
                mov     dptr,#CONTROL       ;Initialise 8255
                mov     a,#CON_WD
                movx    @dptr,a
                sjmp    over
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
BUSY:       mov     dptr,#STATUS        ;Initialise LM629
            movx    a,@dptr
            jb      acc.0,BUSY          ;Check Bit 0 whether LM629
                                        ;busy or not.
            RET

OVER:
            acall   busy                ;Check Busy flag.

            mov     dptr,#COMMD
            mov     a,#RESET            ;Reset LM629
            movx    @dptr,a
            acall   busy

            mov     a,#DFH              ;Define "Home" position
            movx    @dptr,a
            acall   busy

            mov     a,#MSKI             ;Define "Mask interrupt" command.
            movx    @dptr,a
            mov     dptr,#DATA
            mov     a,#00H
            acall   BUSY
            movx    @dptr,a
            mov     a,#04H              ;Enable Trajectory-Complete interrupt
            acall   busy
            movx    @dptr,a

            mov     dptr,#COMMD
;           mov     a,#LFIL             ;Define Filter parameters.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        mov     a,#LTRJ         ;Setting trajectory control.
        lcall   BUSY
        movx    @dptr,a
        mov     dptr,#DATA      ;Prepare to write the Trajectory
                                ;parameter.
        mov     a,#1DH          ;Sets  Mode = Velocity
                                ;      Direction = Forward
                                ;      Motor is stopped
        lcall   busy
        movx    @dptr,a
        setb    35.1H           ;Mark that the default direction is
                                ;"Forward".
        mov     a,#2AH          ;Define to load Velocity, Acceleratio
                                ;and Position data's.
        lcall   BUSY
        movx    @dptr,a
        mov     r0,#03H         ;Prepare to load Acceleration data.
        mov     a,#00H
LPSEND:
        lcall   BUSY
        movx    @dptr,a
        djnz    r0,LPSEND       ;Loop for sending first 3 bytes
        mov     a,0FH
        lcall   BUSY
        movx    @dptr,a         ;Send last byte of Acceleration data.
    ;
    ;Loading 'Velocity data'.
    ;
        mov     a,#00H          ;Sending 1st byte of Velocity data.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        lcall   BUSY
        movx    @dptr,a
        mov     a,#06H          ;2nd byte
        lcall   BUSY
        movx    @dptr,a
        mov     a,#0D1H         ;3rd byte
        lcall   BUSY
        movx    @dptr,a
        mov     a,#0ECH         ;Last byte.
        lcall   BUSY
        movx    @dptr,a
;
;Loading 'Position data'.
;
        mov     a,#00H
        lcall   busy
        movx    @dptr,a
        mov     a,#03H
        lcall   busy
        movx    @dptr,a
        mov     a,#0DH
        lcall   busy
        movx    @dptr,a
        mov     a,#40H
        lcall   busy
        movx    @dptr,a
;
;Starting Motor.\
;
        mov     dptr,#commd
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                mov     a,#01H
                movx    @dptr,a
        ;
        ;Now! motor is working and CPU is waiting for any command from
        ;PC to load in.
        ;
        ;Clearing 4-bytes Input Buffer (Int. RAM 30H-34H) and Input Bit
        ;(Int. RAM 35H).
        ;
CHECKBIT:       mov     a,35H           ;Get contents of Input Bit.
        ;
        ;Check bit '0' of Internal RAM address 20H. If the bit is set, means
        ;command exist.
        ;
                jb      acc.0,READ_BUF   ;If command exist, then go to proces
                sjmp    checkbit
        ;
READ_BUF:       mov     a,#34H
                cjne    a,#01H,VELO_CMD ;Looking for STT command.
                mov     dptr,#commd
                mov     a,#STT
                lcall   BUSY
                movx    @dptr,a
        ;
                lcall   CLRBUF          ;Clear Input Buffer and Input Bit
                                        ;before exit.
                RET
        ;
VELO_CMD:       cjne    a,#02H,ACC_CMD  ;Looking for Velocity changing comman
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        ;If a Velocity command exist, need to transfer the data's in Input
        ;Buffer to Velocity Buffer.
        ;
                mov     r0,#START_BF
                mov     r1,#START_VL        ;Start address of Velocity Buffe
                mov     r2,#04H
X_FER:          mov     a,@r0
                mov     @r1,a
                inc     r0
                inc     r1
                djnz    r2,X_FER
        ;
        ;Loading Velocity data.
        ;
                mov     dptr,#COMMD
                mov     a,#LTRJ
                lcall   busy
                movx    @dptr,a
                mov     dptr,#DATA
                mov     a,35H
                jnb     acc.1,VEL_REV   ;If Acc bit 1 is "0" means previous
                                        ;direction is "reverse"
                mov     a,#1CH          ;Loading  Trajectory Control Word
                                        ;       -Forward Direction
                                        ;       -Velocity mode
                                        ;       -Smoothly stopping
                                        ;       -Only Velocity will be loaded
                lcall   BUSY
                movx    @dptr,a
                sjmp    BOTH_VEL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
VEL_REV:
                mov     a,#08H          ;Mode=Velocity, Motion=Reverse
                lcall    BUSY
                movx    @dptr,a
BOTH_VEL:
                mov     a,#08H          ;Define Velocity values will be loade
                lcall   BUSY
                movx    @dptr,a
        ;
        ;Starting to load Velocity values to LM629
        ;
                mov     r2,#04H
                mov     r0,#36H
VELO_LD:
                mov     a,@r0
                lcall   BUSY
                movx    @dptr,a
                inc     r0
                djnz    r2,VELO_LD

                lcall   CLRBUF
                RET
        ;
        ;Next! check whether it's Acceleration loading command or not.
        ;
ACC_CMD:
                cjne    a,#03H, POS_CMD ;If not Acceleration command, skip to
                                        ;Positioning command.
        ;
        ;Preparring to x-fer data's in Input buffer to Acceleration area.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;
        mov     r0,Start_bf
        mov     r1,Start_ac
        mov     r2,#04H
FILL_ACC:
        mov     a,@r0
        mov     @r1,a
        inc     r0
        inc     r1
        djnz    r2,FILL_ACC

        lcall   CLRBUF          ;Clear Input buffer after transferred.

        mov     dptr,#COMMD
        mov     a,#LTRJ
        lcall   BUSY
        movx    @dptr,a         ;Send a LTRJ command to LM629.
;
        mov     dptr,#DATA
        mov     a,35H
        jnb     acc.1,ACC_REV
        mov     a,#19H          ;Mode=Velocity, Direction=Forward,
                                ;Motor=Off.
        lcall   BUSY
        movx    @dptr,a
        sjmp    BOTH_ACC
ACC_REV:
        mov     a,#09H          ;Mode=Velocity, Direction=Reverse,
                                ;Motion=Turn off.
        lcall   BUSY
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                movx    @dptr,a
BOTH_ACC:
                mov     a,#20H          ;Define that Acceleration value will be
                                        ;loaded.
                movx    @dptr,a
                mov     r0,#3AH         ;Point to starting of Acceleration
                                        ;value area.
                mov     r2,#04H         ;Define numbers of data.
TX_ACC:
                mov     a,@r0
                lcall   BUSY
                movx    @dptr,a
                inc     r0
                djnz    r2,TX_ACC

                mov     dptr,#COMMD     ;Start motor after Acceleration value
                                        ;has been loaded.
                mov     a,#STT
                lcall   BUSY
                movx    @dptr,a

                RET

        ;Positioning command service routine.
        ;
POS_CMD:
                cjne    a,#04H,STOP_CMD ;Go to Stopping command if not position.

                mov     r0,START_BF
                mov     r1,START_PS
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                mov     r2,#04H
FILL_POS:                                       ;Moving the Position values
                                                ;from Input buffer to Position
                                                ;area.
                mov     a,@r0
                mov     @r1,a
                inc     r0
                inc     r1
                djnz    r2,FILL_POS

                lcall   CLRBUF

        ;Sending Positioning command to LM629
        ;
                mov     dptr,#COMMD
                mov     a,#LTRJ
                lcall   BUSY
                movx    @dptr,a
        ;
        ;Sending Position value to LM629
        ;
                mov     dptr,#DATA
                mov     a,35H
                jnb     acc.1,POS_REV
        ;
                mov     a,#18H                  ;   Mode=Velocity
                                                ;   Direction=Forward
                                                ;   Motor keep moving
                lcall   BUSY
                movx    @dptr,a
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                sjmp    BOTH_POS
POS_REV:
                mov     a,#08H
                lcall   BUSY
                movx    @dptr,a
BOTH_POS:
                mov     a,#02H                  ;Define Position value will be
                                                ;loaded.
                mov     r0,START_PS
                mov     r2,#04H
        ;
        ;Load the Positioning values to LM629.
        ;
TX_POS:
                mov     a,@r0
                lcall   BUSY
                movx    @dptr,a
                inc     r0
                djnz    r2,TX_POS

                RET
        ;
        ;Motor Stopping command service routine.
        ;
STOP_CMD:
                cjne    a,#05H,DIR_CMD
                mov     dptr,#COMMD
                mov     a,#LTRJ
                lcall   BUSY
                movx    @dptr,a
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                mov     dptr,#DATA
                mov     a,35H
                jnb     acc.1,STOP_REV          ;'Forward'or 'Reverse'
                mov     a,#19H                  ;Mode=Velocity, Motion=Stop
                                                ;Direction=Forward.
                lcall   BUSY
                movx    @dptr,a
                sjmp    BOTHSTOP
STOP_REV:
                mov     a,#09H                  ;Mode=Velocity, Motion=Stop,
                                                ;Direction=Reverse.
                lcall   BUSY
                movx    @dptr,a
BOTHSTOP:
                mov     a,#00H                  ;Define no data will be loaded
                lcall   BUSY
                movx    @dptr,a
                lcall   CLRBUF

                RET
        ;
        ;Rotating direction command service routine.
        ;
DIR_CMD:
                cjne    a,#06H,FDIR_CMD         ;If code=#05H means Reverse
                                                ;direction.
                mov     dptr,#COMMD
                mov     a,#LTRJ
                lcall   BUSY
                movx    @dptr,a
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                mov     dptr,#DATA
                mov     a,#08H                  ;Mode=Velocity, Motion=Reverse
                lcall   BUSY
                movx    @dptr,a
                clr     35.1H                   ;Clear "Direction bit" to define
                                                ;"reverse" motion.
                sjmp    BOTH_DIR
FDIR_CMD:
                cjne    a,#07H,END_CMD          ;If code=#06H means Forward
                                                ;direction.
                mov     dptr,#COMMD
                mov     a,#LTRJ
                lcall   BUSY
                movx    @dptr,a
                mov     dptr,#DATA
                mov     a,#18H                  ;Mode=Velocity, Motion=Forward
                lcall   BUSY
                movx    @dptr,a
                setb    35.1H                   ;Set "Direction bit" to "1" to
                                                ;define "Forward" motion.
BOTH_DIR:
                mov     a,#00H                  ;Define no data will be loaded
                lcall   BUSY
                movx    @dptr,a
                lcall   CLRBUF

                RET
        ;
        ;End of the command involving to LM629.
        ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
END_CMD:
                nop
CLRBUF:
                mov     r0,#06H
                mov     a,#00H
                mov     r1,#30H
CLEAR:
                mov     @r1,a
                inc     r1
                djnz    r0,CLEAR
                RET
;
;*** Interrupt service routine to communicate with PC ****
;
PC_LINK:
                setb    p1.1            ;Activiate BUSY signal to tell PC not
                                        ;to send any more data at this moment.
                mov     a,P1            ;Get data from 8255. Then keep in
                                        ;Command code area.
                jb      35.0H,DATA_1    ;Check if Command Code has been received
                                        ;or not.
                mov     34H,a           ;If not, assume that byte is Command Code
                                        ;then keep in Command code area.
                setb    35.0H           ; Set Command code status bit to "1"
                                        ;identify that Command code has been
                                        ;loaded from PC.
        ;
        ;If command has been transferred, assume that the subsequence data's
        ;would be data values.
        ;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DATA_1:
        jb      35.2H,DATA_2    ; See if the 1st byte has gotten.
        mov     30H,a           ; If not, then keep in the Input Data
                                ;Buffer.
        setb    35.2H           ; Set bit 2 of Status byte to identify
                                ;that 1st byte has been transferred.
DATA_2:
        jb      35.3H,DATA_3    ; Checking for 2nd byte.
        mov     31H,a
        setb    35.3H
DATA_3:
        jb      35.4H,DATA_4    ; Checking for 3rd byte.
        mov     32H,a
        setb    35.4H
DATA_4:
        jb      35.5H,DATA_LD   ; Checking for 4th byte.
        mov     33H,a
        setb    35.5H

DATA_LD:
        clr     P1.0            ;Activiate "Acknowledge" signal in
                                ;responding to PC that data has been
                                ;accepted.
        nop
        nop
        setb    P1.0
        clr     P1.1            ;Deactiviate "BUSY" signal.
        RETI
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;
;***** Interrupt service routine for interruption from LM629 ******
;
HOST_INT:
                nop
                RETI

                END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

การทดลองประกอบการศึกษาการทำงานของมอเตอร์ไฟฟ้ากระแสตรงแบบไม่มีแปรงถ่าน

5.1 การทดลองที่ 1 ศึกษาการทำงานของมอเตอร์ที่สภาวะ OPEN-LOOP

ลำดับขั้นการทดลอง:

1. ทำการต่อวงจรตามรูปที่ 1 และ 2 โดยการใช้สายต่อที่ขะวต่อบนแผงหน้าปัทม์ของชุดทดลอง ให้เลือกใช้ DRIVER module ที่ 1 ซึ่งเป็น DRIVER ชนิด 3 เฟส และใสายต่อสัญญาน MAG และ DIR มาจาก ชุดการควบคุมด้วยมือ
2. ให้ปลดสายของ ฮอลล์ อิลีเมนท์ ออก
3. ใช้ออสซิลโลสโคปวัดสัญญาณที่ขั้วต่อสัญญาน MAG ของ ไอ.ซี. U3 (LM621)
4. ที่วงจร PWM ในรูปที่ 2 ปรับ VR2 (PERIOD) ให้ความถี่ของสัญญาณ CLOCK มีความถี่ต่ำสุด
5. ปรับ VR3 (DURATION) ให้ค่า DURATION ของสัญญาณ CLOCK อยู่ที่ประมาณ 80 เปอร์เซ็น แล้วสังเกตุว่ามอเตอร์หมุนได้หรือไม่ จากนั้นให้ทดลองปรับ VR3 จนค่า DURATION ลดลงมาเป็น 50 เปอร์เซ็น ถ้ามอเตอร์ยังหมุนได้อยู่ ก็ให้ค่อยๆ ปรับ VR2 (PERIOD) เพิ่มความถี่ของสัญญาณ CLOCK สูงขึ้นอีก สังเกตุว่ามอเตอร์เพิ่มความเร็วขึ้นอีกหรือไม่ ถ้ามอเตอร์ยังทำงานได้ ก็ให้รักษาระดับของ DURATION ไว้ที่ 50% แล้วค่อยปรับทั้ง VR2 (PERIOD) และ VR3 (DURATION) ไปเรื่อยๆ จนถึงจุดที่มอเตอร์ไม่สามารถทำงานได้ หรือเกิดการออสซิลเลท
6. เมื่อรู้ค่าสูงสุดของความถี่ที่มอเตอร์สามารถทำงานได้แล้ว ให้กลับมาเริ่มที่ขั้นตอนที่ 5ใหม่ ไปจนถึงจุดที่ความถี่สูงสุดที่มอเตอร์ยังคงสามารถทำงานได้อยู่ แล้วทดลองปรับ VR3 ให้ DURATION ลดลงมาที่ประมาณ 20% ให้สังเกตุการทำงานของมอเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. บันทึกผลการทดลองลงในตารางที่ 5.1.
8. กดปุ่ม DIR (DIRECTION) เพื่อบังคับให้มอเตอร์หมุนกลับทิศทาง
9. ทำการทดลองในขั้นตอนที่ 5,6, และ 7 ซ้ำอีกแล้วบันทึกผลการทดลองลงในตารางที่ 5.2.
10. ถ้าคราวนี้มอเตอร์ไม่หมุน ให้อธิบายสาเหตุเพราะอะไร

ตารางที่ 5.1. ทิศทางการหมุน _____

ความถี่สัญญาณ CLOCK	DURATION	การทำงานของมอเตอร์		
		หมุน	ไม่หมุน	อื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.2. ทิศทางการหมุน

ความถี่สัญญาณ CLOCK	DURATION	การทำงานของมอเตอร์		
		หมุน	ไม่หมุน	อื่นๆ

5.2 <u>การทดลองที่ 2 ศึกษาการทำงานของมอเตอร์ที่สภาวะ CLOSE-LOOP</u>

ลำดับขั้นการทดลอง:

1. ทำการต่อวงจรตามรูปที่ 1 และ 2 โดยการใช้สายต่อที่ขะวต่อบนแผงหน้าปัทม์ของชุดทดลอง ให้เลือกใช้ DRIVER module ที่ 1 ซึ่งเป็น DRIVER ชนิด 3 เฟส และใสายต่อสัญญาน MAG และ DIR มาจาก ชุดการควบคุมด้วยมือ
2. ให้ต่อสายโรเตอร์ โปชิชั่น เซ็นเซ่อร์ หรือ ฮอลล์อิลีเมนท์ ทั้งสามเข้าที่ขา 5 (HS1), 6 (HS2) และ 7 (HS3) ของไอ.ซี. LM621
3. ใช้ออสซิลโลสโคปวัดสัญญาณที่ขะวต่อสัญญาน MAG ของ ไอ.ซี. U3 (LM621)
4. ที่วงจร PWM ในรูปที่ 2 ปรับ VR2 ให้ความถี่ของสัญญาณ CLOCK มีความถี่ต่ำสุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. ปรับ VR3 (DURATION) ให้ค่า DURATION ของสัญญาณ CLOCK อยู่ที่ประมาณ 80 เปอร์เซ็น แล้วสังเกตุว่ามอเตอร์หมุนได้หรือไม่ จากนั้นให้ทดลองปรับ VR3 จนค่า DURATION ลดลงมาเป็น 50 เปอร์เซ็น ถ้ามอเตอร์ยังหมุนได้อยู่ ก็ให้ค่อยๆ ปรับ VR2 (PERIOD) เพิ่มความถี่ของสัญญาณ CLOCK สูงขึ้นอีก สังเกตุว่ามอเตอร์เพิ่มความเร็วขึ้นอีกหรือไม่ ถ้ามอเตอร์ยังทำงานได้ ก็ให้รักษาระดับของ DURATION ไว้ที่ 50% แล้วค่อยปรับทั้ง VR2 (PERIOD) และ VR3 (DURATION) ไปเรื่อยๆ จนถึงจุดที่มอเตอร์ไม่สามารถทำงานได้ หรือเกิดการออสซิลเลท
6. เมื่อรู้ค่าสูงสุดของความถี่ที่มอเตอร์สามารถทำงานได้แล้ว ให้กลับมาเริ่มที่ขั้นตอนที่ 5ใหม่ ไปจนถึงจุดที่ความถี่สูงสุดที่มอเตอร์ยังคงสามารถทำงานได้อยู่ แล้วทดลองปรับ VR3 ให้ DURATION ลดลงมาที่ประมาณ 20% ให้สังเกตุการทำงานของมอเตอร์
7. บันทึกผลการทดลองลงในตารางที่ 6.3.
8. กดปุ่ม DIR (DIRECTION) เพื่อบังคับให้มอเตอร์หมุนกลับทิศทาง
9. ทำการทดลองในขั้นตอนที่ 5,6, และ 7 ซ้ำอีกแล้วบันทึกผลการทดลองลงในตารางที่ 5.4.
10. ถ้าคราวนี้มอเตอร์ไม่หมุน ให้อธิบายสาเหตุเพราะอะไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 5.3. ทิศทางการหมุน _____

ความถี่สัญญาณ CLOCK	DURATION	การทำงานของมอเตอร์		
		หมุน	ไม่หมุน	อื่นๆ

ตารางที่ 5.4. ทิศทางการหมุน _____

ความถี่สัญญาณ CLOCK	DURATION	การทำงานของมอเตอร์		
		หมุน	ไม่หมุน	อื่นๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3. การทดลองการควบคุมการทำงานของ IC # LM629 โดยไมโครชิพ

1. ทำการต่อวงจรตามรูปที่ 1 และ 2 โดยการใช้สายต่อที่ขะวต่อบนแผงหน้าปัทม์ของชุดทดลอง ให้เลือกใช้ DRIVER module ที่ 1 ซึ่งเป็น DRIVER ชนิด 3 เฟส และใสายต่อสัญญาน MAG และ DIR มาจาก ชุดการควบคุมด้วยไมโครชิพแทน (ดูรายละเอียดจากวงจรของชุดควบคุม 8031)
2. ให้ต่อสายโรเตอร์ โปชิชั่น เซ็นเซ่อร์ หรือ ฮอลล์อิลีเมนท์ ทั้งสามเข้าที่ขา 5 (HS1), 6 (HS2) และ 7 (HS3) ของไอ.ซี. LM621
3. ใช้สายต่อเอ้าท์พุทของฮอล์ลอีลีเมนท์ตัวที่หนึ่งและตัวที่สองไปยังจุดต่อของ IC LM629
4. ปรับสวิทช์ตัวที่ 1,2,3 และ 4 มาที่ตำแหน่ง OFF หรือผลักสวิทชีเข้าหาตัว แล้วเปิดเพาเวอร์สวิทช์จ่ายไฟให้กลับเครื่อง
5. แล้วทดลองผลักครั้งละตัว สวิทช์ตัวที่ 1 ON ขณะที่ตัวอื่นๆ OFF จะแทนการ ON ที่ SPEED ต่ำสุด สวิทช์, สวิทช์ตัวที่ 2 ON ขณะที่ตัวอื่นๆ OFF จะแทนการ ON ที่ SPEED ระดับกลาง และถ้าสวิทช์ตัวที่ 3 ON ขณะที่ตัวอื่น OFF จะแทนการ ON ที่ SPEED สูงสุด ส่วนสวิทว์ตัวที่ 4 คือสวิทช์สำหรับเปลี่ยนทิศทางการหมุน ให้ทดลองปรับสวิทช์ต่างๆเหล่านี้แล้วเปรียบผลการทดลองกับการทดลองที่ 5.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.4. การทดลองการควบคุมการทำงานของ IC # LM629 โดยไมโครคอมพิวเตอร์ผ่านทางปริ้นเตอร์พอร์ท

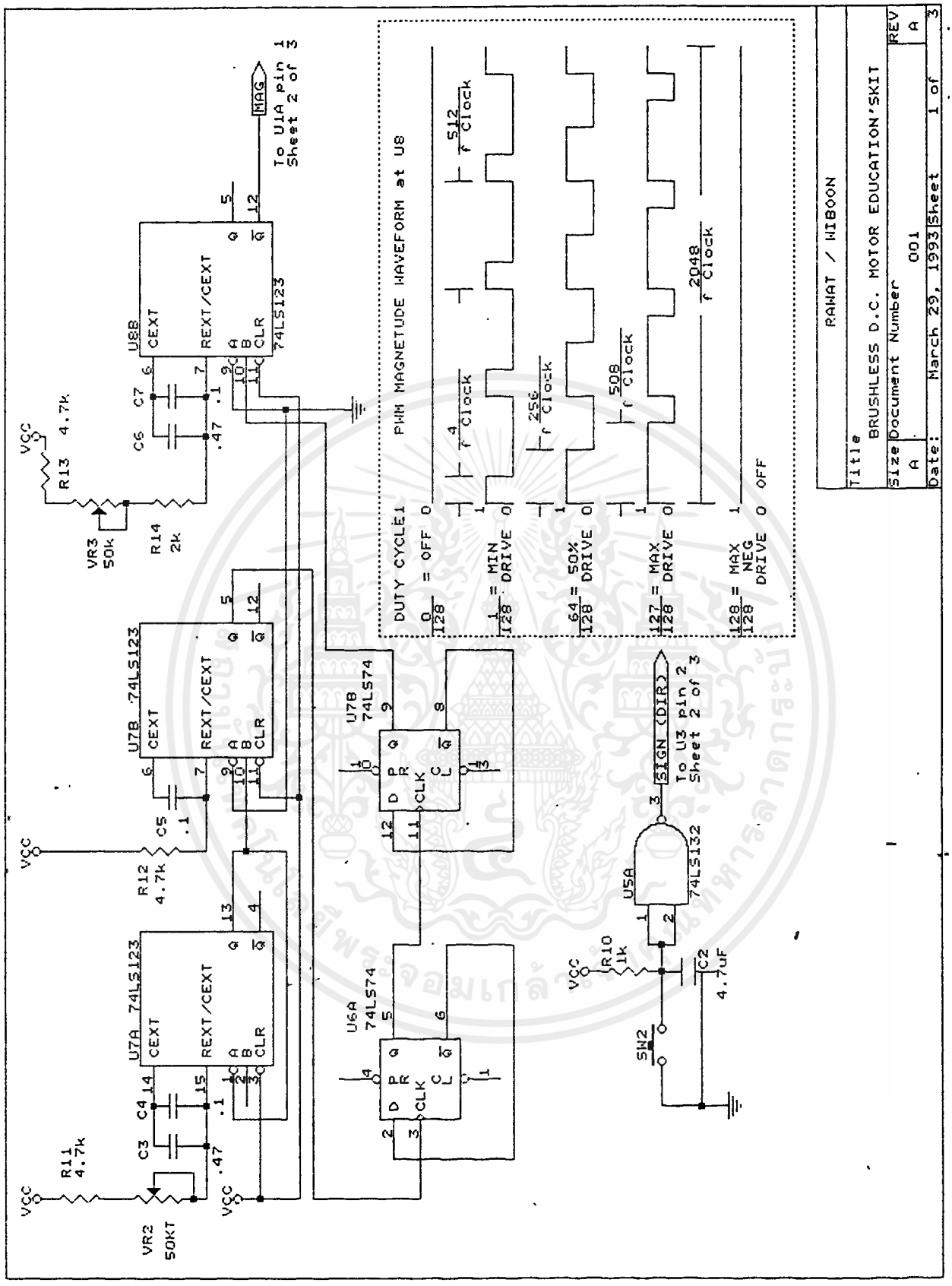
1. ทำการต่อวงจรตามรูปที่ 1 และ 2 โดยการใช้สายต่อที่ขะวต่อบนแผงหน้าปัทม์ของชุดทดลอง ให้เลือกใช้ DRIVER module ที่ 1 ซึ่งเป็น DRIVER ชนิด 3 เฟส และใสายต่อสัญญาน MAG และ DIR มาจาก ชุดการควบคุมด้วยไมโครชิพแทน (ดูรายละเอียดจากวงจรของชุดควบคุม 8031)
2. ให้ต่อสายโรเตอร์ โปชิชั่น เซ็นเซ่อร์ หรือ ฮอลล์อิลีเมนท์ ทั้งสามเข้าที่ขา 5 (HS1), 6 (HS2) และ 7 (HS3) ของไอ.ซี. LM621
3. ใช้สายต่อเอ้าท์พุทของฮอล์ลอีลีเมนท์ตัวที่หนึ่งและตัวที่สองไปยังจุดต่อของ IC LM629
4. ต่อสายจากปริ้นเตอร์พอร์ทมายังคอนเนคเตอร์ของชุดทดลอง
5. โหลดโปรแกรมแล้วรัน หลังจากนั้นให้ทำการเปลี่ยน PERIOD และ DURATION แล้วสังเกตุว่าการทำงานที่ จุดต่างๆเป็นอย่างไร
6. ทำการทดลองเช่นเดียวกับข้อ 5 แต่ให้มอเตอร์มีโหลดบ้างเล็กน้อย แล้วสังเกตุว่ามอเตอร์สามารถทำความเร็ววิ่งเข้าใกล้ความเร็วที่ไม่มีโหลดได้หรือไม่ ให้อธิบายว่ามีความแตกต่างกันอย่างไร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5. คำถามท้ายการทดลอง

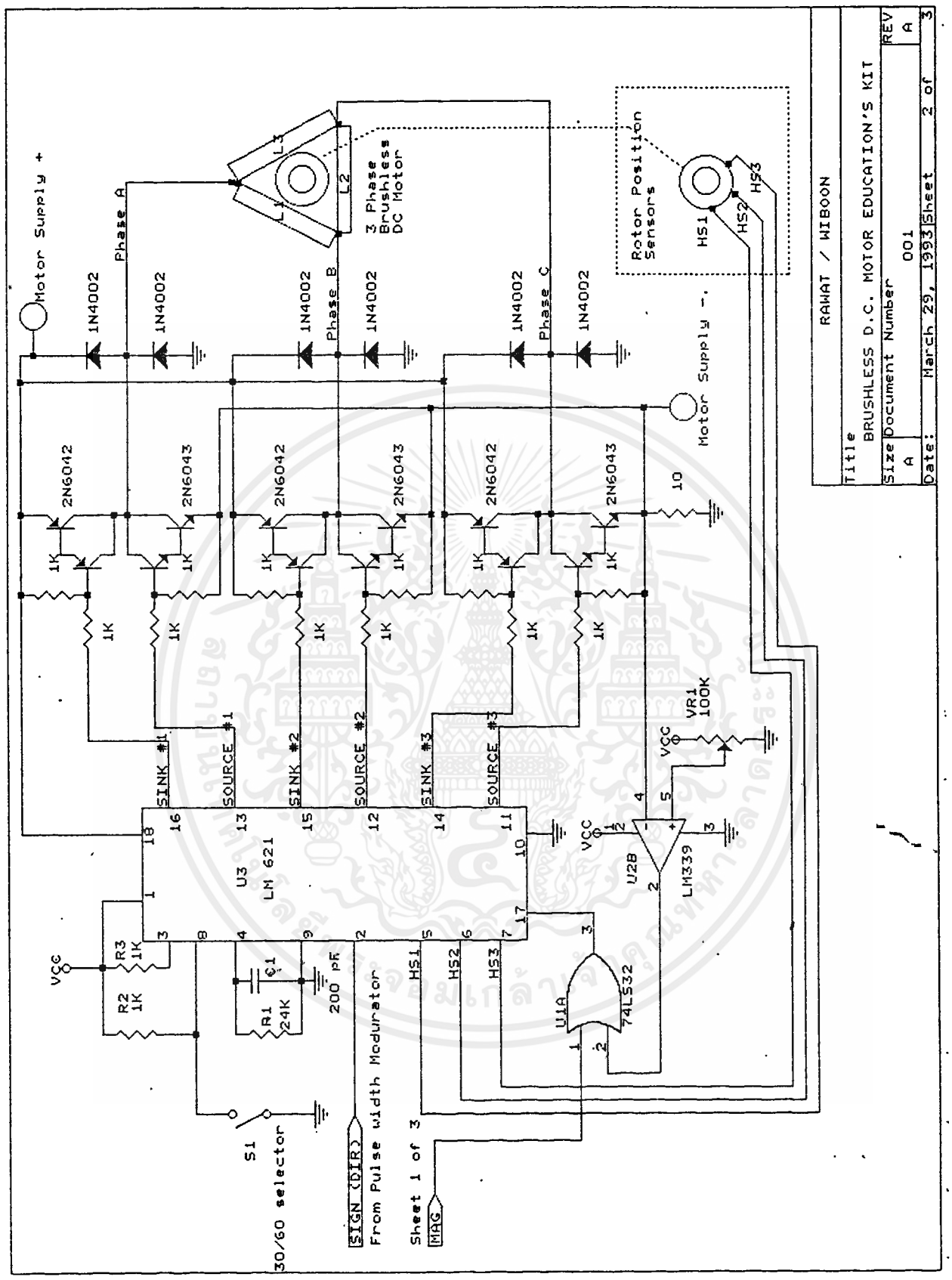
1. ในกรณีการทดลองแบบ OPEN-LOOP ถ้ามอเตอร์หมุนได้จงอธิบายเหตุผล
2. ในกรณีการทดลองแบบ OPEN-LOOP ถ้ามอเตอร์ไม่หมุน จงอธิบายว่าเพราะเหตุใดจึงไม่หมุน
3. จงเปรียบเทียบความแตกต่าง การให้ DURATION มากกว่า 50% และน้อยกว่า 50%
4. จงบอกความแตกต่าง เมื่อเปลี่ยนสวิทช์ 30/60 องศาไปข้างหน้า (ตำแหน่ง 60 องศา) และผลักเข้าหาตัว (ตำแหน่ง 30 องศา)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



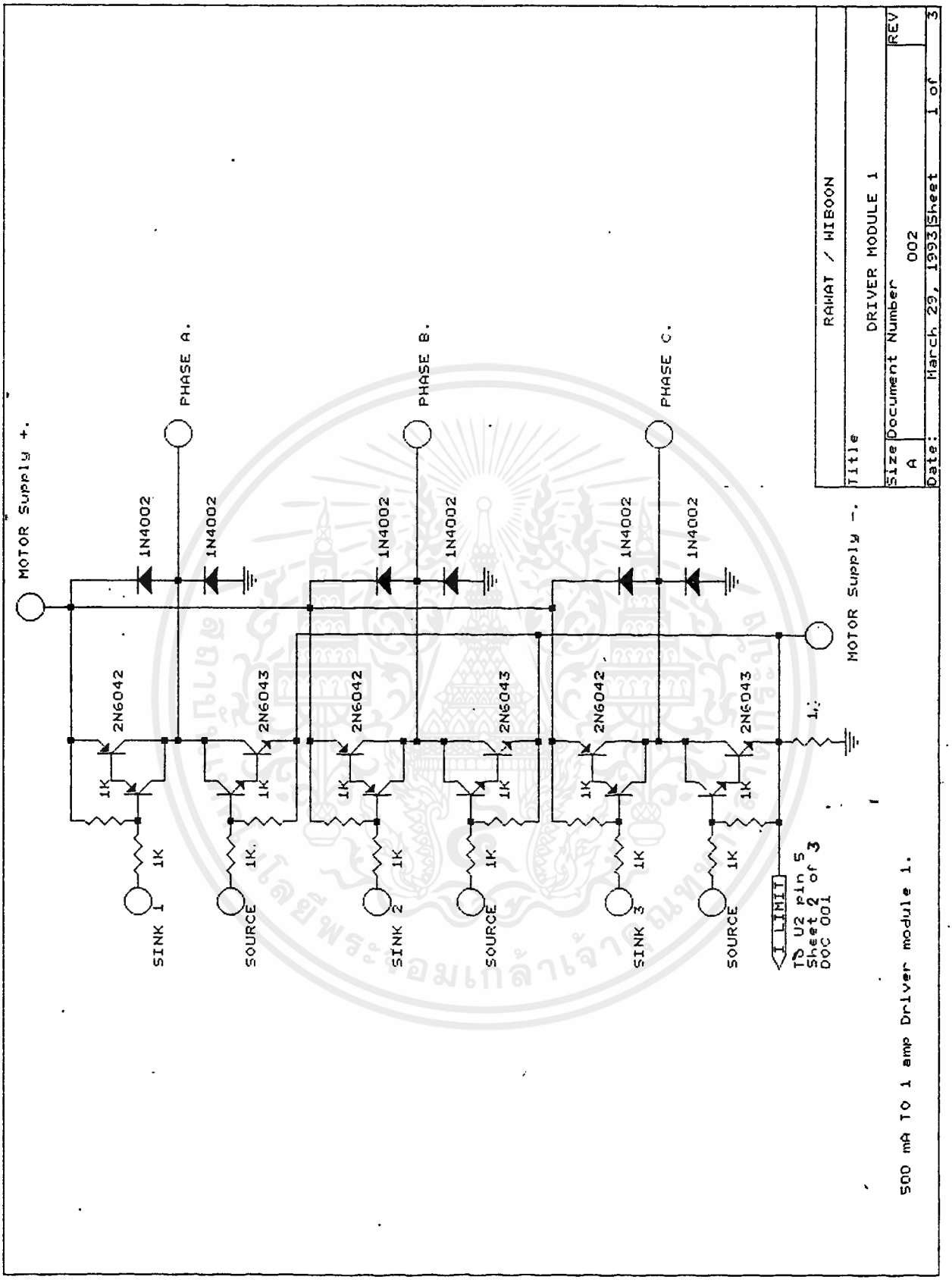
รูปที่ 1 วงจรที่ใช้ในการทดลองโครงงาน (แผ่นที่ 1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



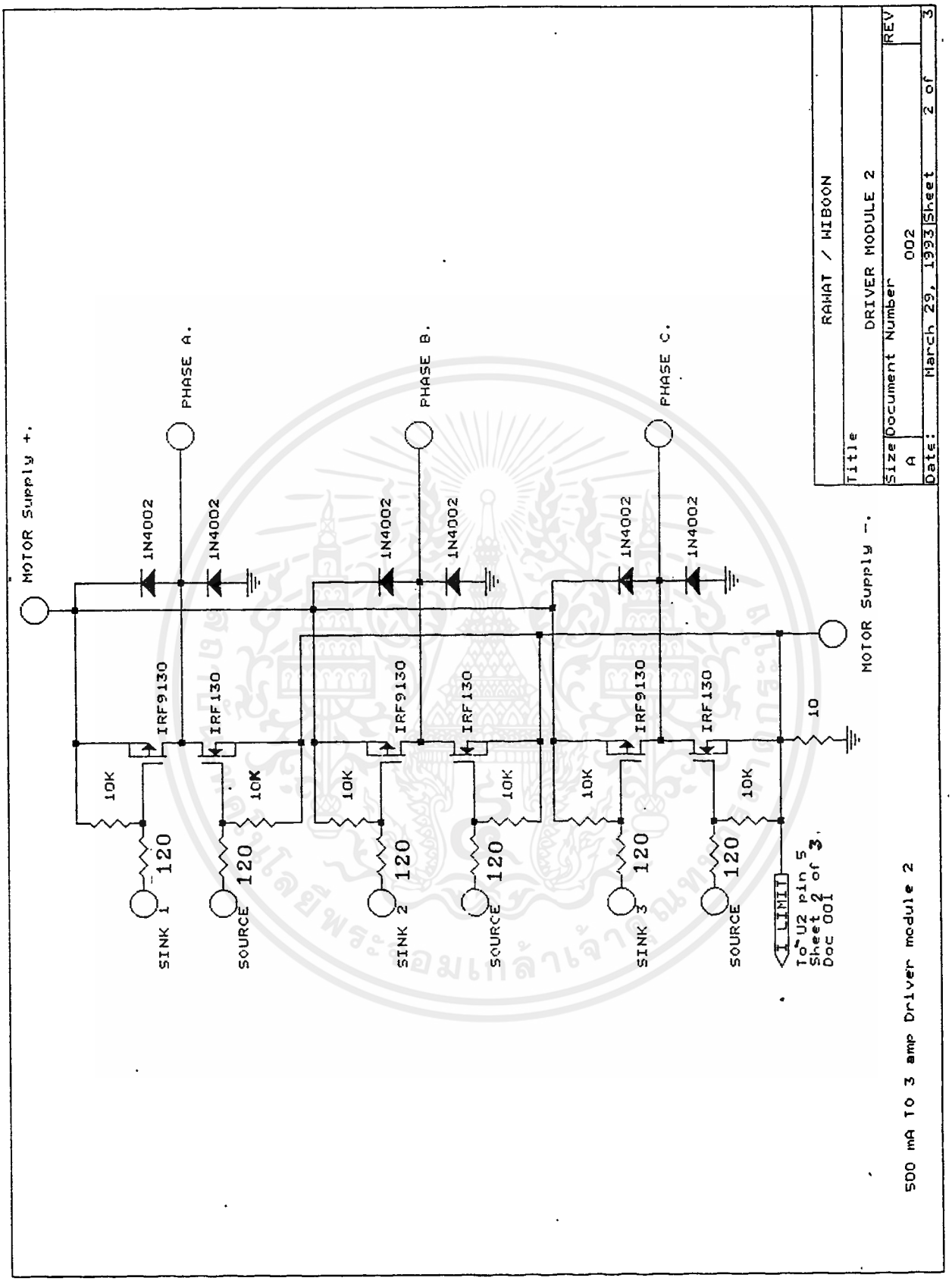
รูปที่ 2 วงจรที่ใช้ในการทดลองโครงงาน (แผ่นที่ 2)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



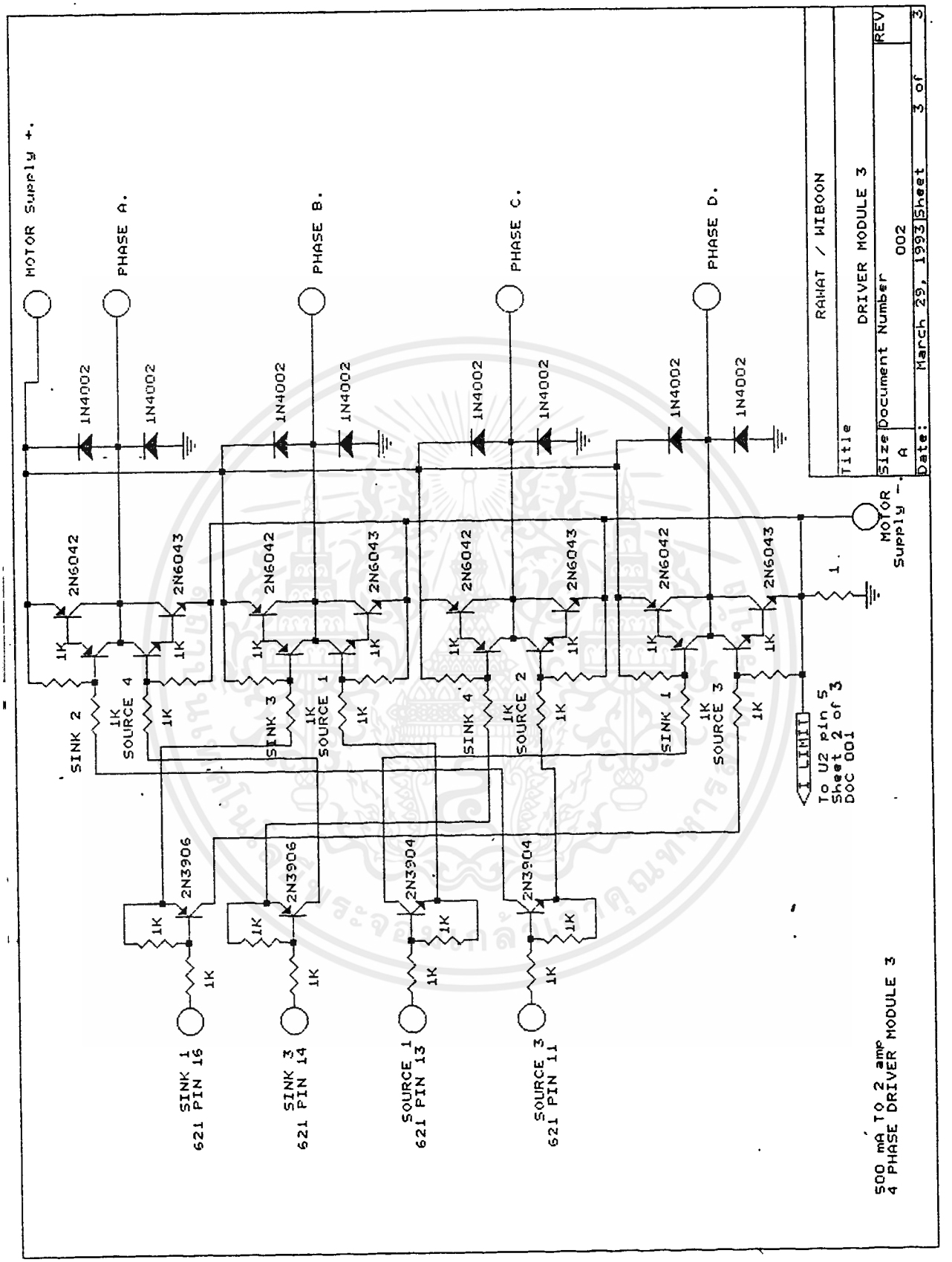
วงจร DRIVER ชนิด 3 เฟส แบบที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



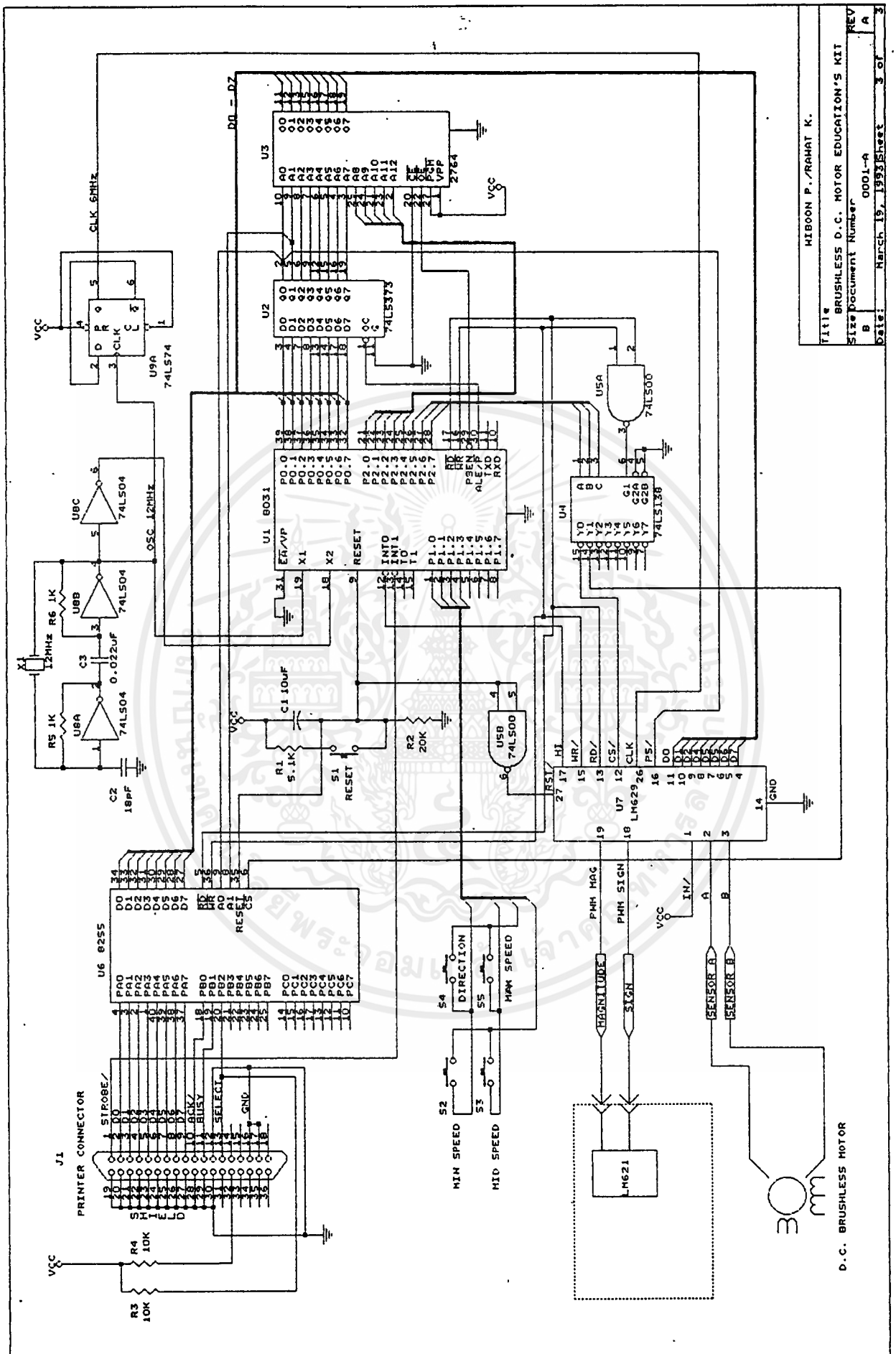
วงจร DRIVER ชนิด 3 เฟส แบบที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



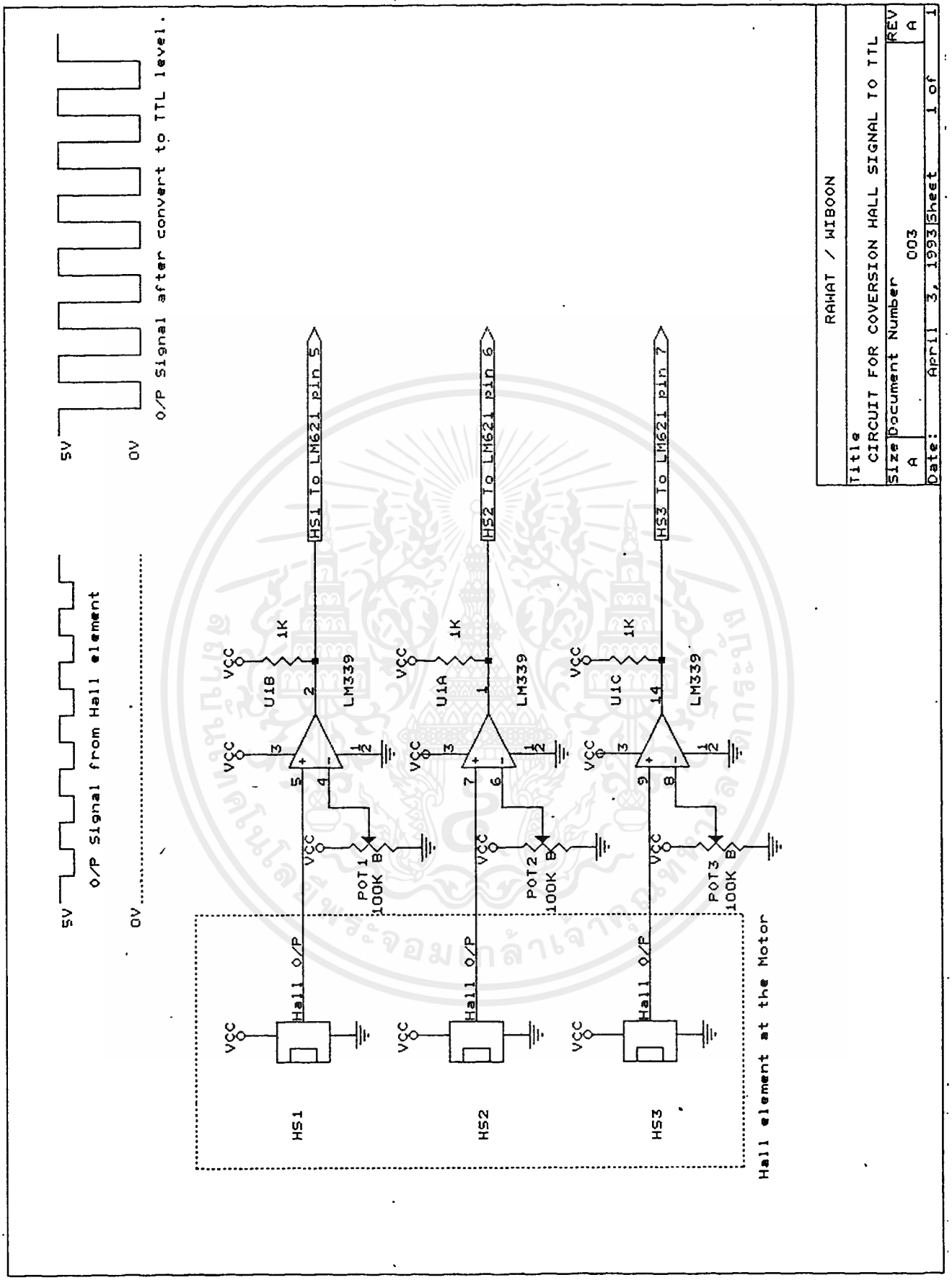
วงจร DRIVER ชนิด 4 เฟส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



วงจรสำหรับต่อเชื่อมการควบคุมจากภายนอก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



วงจรสำหรับเปลี่ยนระดับสัญญาณฮอล์ลอีลีเมนท์ให้เป็นระดับสัญญาณทีทีแอล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สรุป

การทดลองประสพผลสำเร็จไปได้ระดับหนึ่ง แต่ก็ยังไม่เป็นที่พอใจ ซึ่งทั้งนี้ก็เพราะว่าอุปสรรคอันหนึ่งก็คือการสรรหาอุปกรณ์ที่จะนำมาประกอบใช้ในโครงงานนี้ซึ่งหมายรวมถึง ตัวชิพ, มอเตอร์, ฮอล์ลอีลีเมนท์ และเอกสารอ้างอิงต่างๆค่อนข้างจะหาได้ยากภายในประเทศเราเอง โดยเฉพาะอย่างยิ่งในโครงการนี้ได้ใช้ชิพเบอร์ LM629 ทำหน้าที่เป็นตัวควบคุมการทำงานของคอมมิวเตเตอร์ชิพ และในระหว่างการพัฒนาได้ได้เกิดความเสียหายขึ้นกับชิพเบอร์ LM629 ซึ่งการแก้ไขก็คือการสั่งซื้อชิพตัวใหม่เข้ามาเปลี่ยนแต่ปรากฏว่าชิพที่ได้รับมาใหม่นั้นเป็นชิพเบอร์ LM628 ซึ่งใช้กับโครงงานนี้ไม่ได้ จำเป็นที่จะต้องทำการสั่งซื้อใหม่ ซึ่งก็ยังไม่สามารถบออกว่าจะได้รับเมื่อไร อย่างไรก็ตามโครงงานนี้ได้ค้นหาและรวบรวมข้อมูลต่างๆได้มากพอสมควร ข้าพเจ้าหวังว่าคงจะเป็นประโยชน์บ้างสำหรับผู้ที่สนใจงานทางด้านนี้ และหวังว่าจะมีการพัฒนางานลักษณะนี้ต่อไปจนสามารถนำไปประยุกต์ใช้กับการควบคุมขนาดใหญ่ๆได้ ในอนาคตอันใกล้นี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. LINEAR APPLICATION HANDBOOK BOOK 3, 1985.
NATIONAL SEMICONDUCTOR CORPORATION.
2. สมชาย ยรรยงพาณิชย์ "บลัชเลส ดีซีมอเตอร์"
วรสาร เซมิคอนดัคเตอร์ ฉบับที่ 118 ประจำเดือน กรกฎาคม 2535
3. APPLICATION NOTE LIA NOTE 11
"MEASUREMENT OF THE HALL EFFECT USING A LOCK-IN AMPLIFIER"
อภินันทนาการจากบริษัท READ-RITE (BANGKOK) CO.,LTD.
4. ALLEN NUSSBAUM "SEMICONDUCTOR DEVICES PHYSICS"
PRENTRICE HALL, INC. 1962.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวกโปรแกรมคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Program Xfer_cmd;
Uses Crt,Printer,Dos;
Type  Command = String[15];
      Velocity = longInt;
      Resolution = Integer;
      Direct  = String[7];
Const Factor   = 65536;
      T        = 341e-6;
      C        = 1/60;
Var   Cmmd : Command;
      Regs : Registers;
         V : Velocity;
         A : longint;
         R : Resolution;
      Dir  : Direct;
      RPM  : Integer;
      Acc  : Integer;
Procedure Checkey (Cmmd:String);
   Begin
      If (Cmmd ='Start')or (Cmmd='S') or (Cmmd='s') then
         Begin
               Regs.AH := $05;
               Regs.DL := $01;
               MsDos (Regs);
         End;
      If (Cmmd='Stop') or (Cmmd='Sp') then
         Begin
               Regs.AH := $05;
               Regs.DL := $05;
               MsDos (Regs);
         End;
      If (Cmmd='Velocity') or (Cmmd='Vel') or (Cmmd='v') then
         Begin
              Write('Enter the desired number of RPM >');
              Readln(RPM);
              V:=Trunc(R * T * C * RPM * Factor)
         end;
      If (Cmmd='Acceleration') or (Cmmd='Acc') or (Cmmd='acc')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      or (Cmmd='a') then
         Begin
              Write('Enter the desired number of Acceleration > ');
              Readln(Acc);
              A:=Trunc(R * T * T * Acc * Factor)
         end;
      If (Cmmd='Direction') or (Cmmd='Dir') or (Cmmd='dir')
      or (Cmmd='d') then
         Begin
              Write('Enter a desired direction (Forward/Reverse) > ');
              Readln(Dir);
                If (Dir='Forward') or (Dir='F') or (Dir='f') then
                Begin
                     Regs.AH := $05;
                     Regs.DL := $07;
                     MsDos (Regs);
                end;
                If (Dir='Reverse') or (Dir='R') or (Dir='r') then
                Begin
                     Regs.AH := $05;
                     Regs.DL := $06;
                     MsDos (Regs);
                end;
        End;
   End;


Begin {Main}
      Write('Please enter the number of Encoder Lines > ');
      Readln(R);
      R:=R*4;
      Writeln ('Enter a command (Start, Stop, Velocity, Accelerator, or Di
      Readln (Cmmd);
      Checkey (Cmmd);
End.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;Program Frequency_Duration Control.....Revision 1.0.
;This program was created by Rawat Khantrum and Wiboon Pichaiya
;in order to support a project named "Brushless D.C. Motor
;Education's Kit" (SGT11).
;
;.
;------------- 8255's Port Address ----------------------------
;
CONTROL         EQU     4003H
PORT_A          EQU     4000H
PORT_B          EQU     4001H
PORT_C          EQU     4002H
;
;-------------- MC1408 Port's Address -------------------------
;
FREQ_P          EQU     6000H   ;Port address for Period control DAC
PULSE_WD        EQU     8000H   ;Port address for Pulse width control
                                ;DAC
;
;-------------- Variable Declaration --------------------------
;
CON_WD          EQU     82H     ;8255 control word => Port A,C =Input.
                                ;                     Port B   =Output.
F_CODE          EQU     0AH     ;Forward direction command code
R_CODE          EQU     0BH     ;Reverse direction command code
MODE_CD1        EQU     0CH     ;Mode "Frequency"
MODE_CD2        EQU     0DH     ;Mode "Pulse width"
INC_CODE        EQU     0EH     ;Increment speed
DEC_CODE        EQU     0FH     ;Decrement speed
;
STEP            EQU     05H     ;Bit counts change per step
HI_LIMIT        EQU     0D0H    ;Maximum Input value to DAC
LO_LIMIT        EQU     61H     ;Mininum Input value to DAC
;
;-------------- Program Start !!! -----------------------------
;
                org     0000H
                ljmp    INITIAL
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;
                org     0013H
                ljmp    PC_LINK
;
                org     0024H
INITIAL:
        ;Initial Stack Pointer to start at address 60H.
                mov     sp,#60H
        ;Initialize 8255's Ports to the desired function.
                mov     dptr,#CONTROL
                mov     a,#CON_WD
                movx    @dptr,a
                mov     dptr,#PORT_B
                mov     a,#05H          ;Busy='0', Acknowledge='1'
                                        ;Select='1'.
                movx    @dptr,a
       ;Initialize the MC1408 DAC's to supply half range output.
                mov     dptr,#FREQ_P
                mov     a,#0A0H         ;Per experimental, this value is su
                                        ;for "Half-scale" value.
                movx    @dptr,a
                mov     35H,a
       ;
                mov     dptr,#PULSE_WD
                mov     a,#90H
                movx    @dptr,a
                mov     36H,a
      ;Initialize 8031 Port P1 to have P1.0-P1.3 as Input pins and
      ;P1.4-P1.7 as Output pins.
                mov     p1,#0FH
                setb    p1.4            ;Set motion direction to 'Forward'.
      ;
      ;Clear all buffer area.
      ;
      ;Set first start up condition. Mode=Freq, Direction=Forward.
                mov     31H,#03H
      ;
                mov     32H,#00H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                mov     33H,#0FH
                mov     34H,#00H
                mov     35H,#00H
                mov     36H,#00H
        ;
        ;Enable External Interrupt #1.
        ;
                setb    IE.2
                setb    IP.2
        ;Reading Input Key.
CHECKEY:
                mov     a,p1
                anl     a,#0FH
                mov     32H,a                   ;Keep present switch status
                jb      acc.0,F_DIR
KEY1:           jnb     acc.0,R_DIR
KEY2:           jb      acc.1,FREQ
KEY3:           jnb     acc.1,PW
KEY4:           jnb     acc.2,INCxx
                orl     33H,#04H
KEY5:           jnb     acc.3,DECxx
                orl     33H,#08H
        ;
        ;Remote command checking
        ;
                mov     a,34H
                jnb     acc.7,CHECKEY           ;See whether command is va
                                                ;or not
        ;
        ;If valid (bit-7 is logical "1") then go to read command
        ;
                mov     a,34H
                cjne    a,#8AH,COMND2
        ;If command code = 0AH, then do Forward direction.
                lcall   FORWARD
                sjmp    END_COMD                ;Clear Command code buffer
COMND2:
                cjne    a,#8BH,COMND3
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                lcall   REVERSE                 ;Do 'Reverse' direction.
                sjmp    END_COMD
COMND3:
                cjne    a,#8CH,COMND4
                lcall   MODE1                   ;Do 'FREQ' mode
                sjmp    END_COMD
COMND4:
                cjne    a,#8DH,COMND5
                lcall   MODE2                   ;Do 'PULE-WIDTH' mode
                sjmp    END_COMD
COMND5:
                cjne    a,#8EH,COMND6
                lcall   INCREASE                ;Do 'Increment speed'
                sjmp    END_COMD
COMND6:
                cjne    a,#8FH,END_COMD
                lcall   DECREMET                ;Do 'Increment speed'
END_COMD:
                anl     34H,#00H
                sjmp    CHECKEY
;
;This instruction is used for support the "jb" command that is too far
;from the target.
;
DECxx:          ljmp    DEC
INCxx:          ljmp    INC
;

        ;
        ;Manual Key-switches checking.
        ;
F_DIR:
                push    acc
                lcall   FORWARD
                pop     acc
                sjmp    KEY1
R_DIR:
                push    acc
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                lcall   REVERSE
                pop     acc
                sjmp    KEY2
        ;
        ;Mode selection "Frequency (FREQ)/Pulse Width(PW)".
        ;
FREQ:
                acall   MODE1
                sjmp    KEY3
PW:
                acall   MODE2
                sjmp    KEY4
;
MODE1:
                push    acc
                mov     a,31H
                jb      acc.1,FREQ_OUT
                setb    p1.5
                orl     33H,#02H                ;Set MODE to FREQ.
                orl     31H,#02H
FREQ_OUT:
                pop     acc
                ret
        ;
        ;Mode selection "Pulse width".
        ;
MODE2:
                push    acc
                mov     a,31H
                jnb     acc.1,PULSE_O
                clr     p1.5
                anl     33H,#0FDH
                anl     31H,#0FDH
PULSE_O:
                pop     acc
                ret
        ;
INC:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                push    acc
                mov     a,33H
                jnb     acc.2,INC_OUT
                lcall   INCREASE
                setb    p1.6
                anl     33H,#0FBH
INC_OUT:
                setb    p1.7
                pop     acc
                ljmp    KEY5
DEC:
                push    acc
                mov     a,33H
                jnb     acc.3,DEC_OUT
                lcall   DECREMET
                clr     p1.6
                anl     33H,#0F7H
DEC_OUT:
                clr     p1.7
                pop     acc
                ljmp    CHECKEY
FORWARD:
                mov     a,31H
                jb      acc.0,F_EXIT
                setb    p1.4
                setb    acc.0
                mov     31H,a
F_EXIT:
                ret
;
REVERSE:
                mov     a,31H
                jnb     acc.0,R_EXIT
                clr     p1.4
                anl     a,#0FEH
                mov     31H,a
R_EXIT:         ret
;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
INCREASE:
                mov     a,31H
                jnb     acc.1,PW_INC
                mov     a,35H
                cjne    a,#HI_LIMIT,CONT
                ret
;
CONT:
                mov     r0,#STEP
ADD_STEP:
                inc     a
                cjne    a,#HI_LIMIT,ADD_MORE
                mov     35H,a
                ret
ADD_MORE:
                djnz    r0,ADD_STEP
                mov     35H,a
                mov     dptr,#FREQ_P
                movx    @dptr,a
                ret
;
PW_INC:
                mov     a,36H
                cjne    a,#HI_LIMIT,CONT1
                ret
CONT1:
                mov     r0,#STEP
ADD_STEP1:
                inc     a
                cjne    a,#HI_LIMIT,ADD_MRE
                mov     36H,a
                mov     dptr,#PULSE_WD
                movx    @dptr,a
                ret
ADD_MRE:
                djnz    r0,ADD_STEP1
                mov     36H,a
                mov     dptr,#PULSE_WD
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                movx    @dptr,a
                ret
;
DECREMET:
                mov     a,31H
                jnb     acc.1,DEC_WD
                mov     a,35H
                cjne    a,#LO_LIMIT,CONT1
                ret
        ;
CONT2:
                mov     r0,#STEP
SUB_STEP:
                dec     a
                cjne    a,#LO_LIMIT,SUB_MORE
        ;
                mov     35H,a
                mov     dptr,#FREQ_P
                movx    @dptr,a
                ret
        ;
SUB_MORE:
                djnz    r0,SUB_STEP
                mov     35H,a
                mov     dptr,#FREQ_P
                movx    @dptr,a
                ret
DEC_WD:
                mov     a,36H
                cjne    a,#LO_LIMIT,CONT3
                ret
        ;
CONT3:
                mov     r0,#STEP
SUBSTEP:
                dec     a
                cjne    a,#LO_LIMIT,SUBMORE
        ;
```

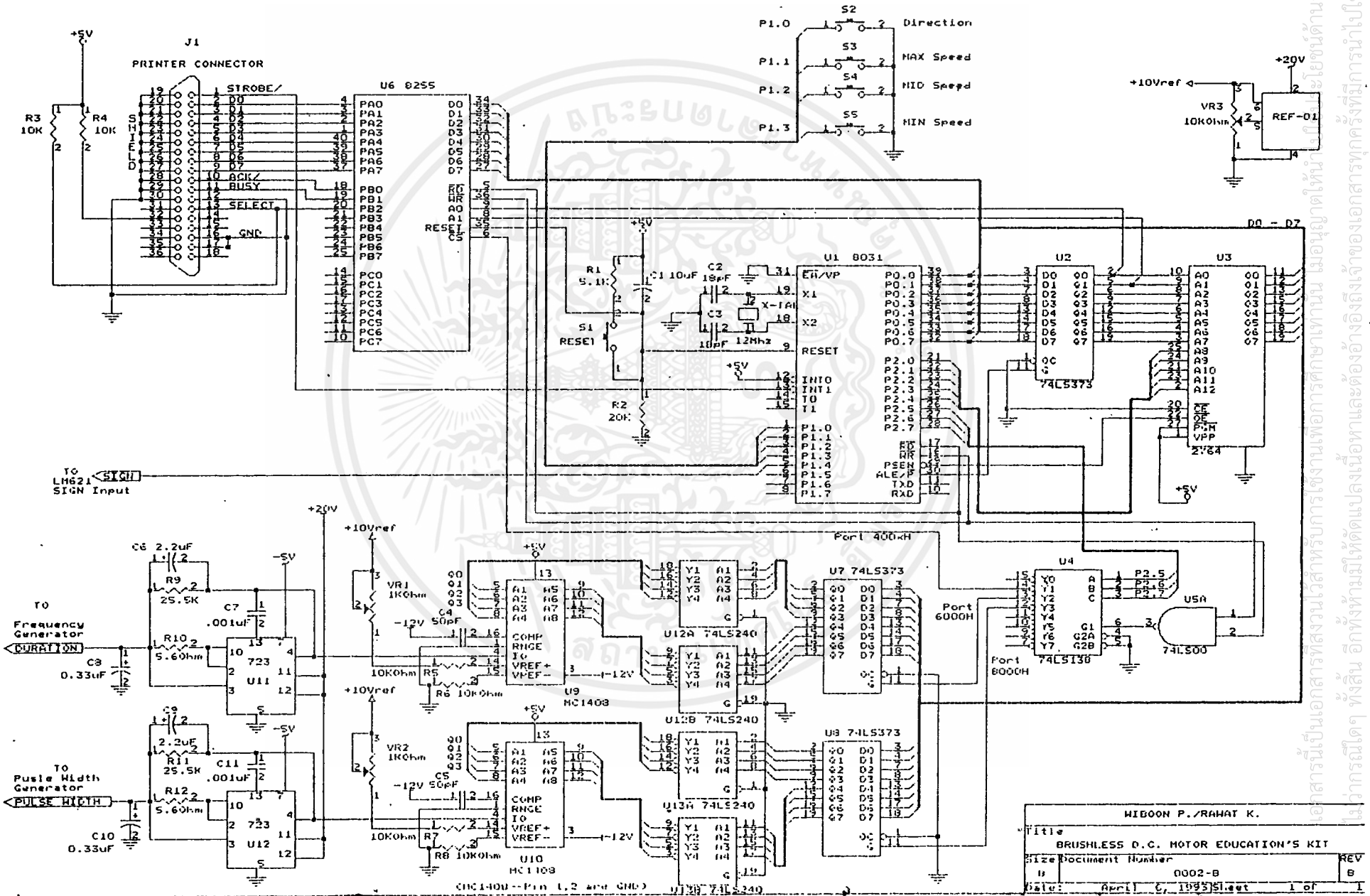
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                mov     36H,a
                mov     dptr,#PULSE_WD
                movx    @dptr,a
                ret
        ;
SUBMORE:
                djnz    r0,SUBSTEP
                mov     36H,a
                mov     dptr,#PULSE_WD
                movx    @dptr,a
                ret
        ;
        ;****** Interrupt service routine for printer port ******
        ;
PC_LINK:
        ;Save working registers.
                push    acc
                push    psw
                push    dpl
                push    dph
                mov     dptr,#PORT_B
                mov     a,#07H          ;Activiate "Busy" signal.
                movx    @dptr,a
                mov     dptr,#PORT_A
                movx    a,@dptr         ;Read data from printer port.
                orl     a,#80H          ;Set bit-7 of imported data
                                        ;to identify command is exist
                mov     34H,a
                mov     dptr,#PORT_B
                mov     a,#06H          ;Activiate an active low "Acknow-
                                        ;ledge" signal.
                nop                     ;Delay for signal stable.
                nop
                mov     a,#07H          ;Deactiviate acknowledge signal.
                movx    @dptr,a
                mov     a,#05H          ;Deactiviate busy signal.
                movx    @dptr,a
                pop     dph
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        pop     dpl
        pop     psw
        pop     acc
;
        reti
;
        END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


J1
PRINTER CONNECTOR
STROBE/
ACK/
BUSY
SELECT
GND
U6 8255
S2
P1.0
Direction
S3
P1.1
MAX Speed
S4
P1.2
MID Speed
S5
P1.3
MIN Speed
+10Vref
VR3
10K0hm
REF-01
+20V
U1 8031
RESET
S1
R1
5.1K
C1 10uF
C2
18pF
C3
18pF
12Mhz
R2
20K
U2
74LS373
U3
2764
D0 - D7
TO
LM621
SIGN Input
SIGN
Port 400xH
U4
74LS138
U5A
74LS00
Port
6000H
Port
8000H
U7 74LS373
U8 74LS373
U12A 74LS240
U12B 74LS240
U13A 74LS240
U13B 74LS240
U9
MC1408
U10
MC1408
VR1
1K0hm
VR2
1K0hm
C4
-12V 50pF
C5
-12V 50pF
R5
10K0hm
R6 10K0hm
R7
10K0hm
R8 10K0hm
-12V
TO
Frequency
Generator
DURATION
C6 2.2uF
R9
25.5K
C7
.001uF
R10
5.60hm
U11
C8
0.33uF
C9
2.2uF
R11
25.5K
C11
.001uF
R12
5.60hm
U12
C10
0.33uF
TO
Pusle Width
Generator
PULSE WIDTH
MC1408--Pin 1,2 and GND
WIBOON P./RAHAT K.
Title
BRUSHLESS D.C. MOTOR EDUCATION'S KIT
Size
B
Document Number
0002-B
REV
B
Date:
April 6, 1993
Sheet
1 of
1