

ระบบเครื่องจักรป้อนชิ้นงานอัตโนมัติโดยไมโครคอนโทรลเลอร์
AUTOMATIC FEEDING MACHINE SYSTEM WITH
MICROCONTROLLER



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมแมคคาทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2558

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบเครื่องจักรป้อนชิ้นงานอัตโนมัติโดยไมโครคอนโทรลเลอร์
AUTOMATIC FEEDING MACHINE SYSTEM WITH
MICROCONTROLLER



T144388



เลขหมู่.....
เลขทะเบียน 144388
ในเดือนปี 24 พ.ย. 2559

12819659

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมแมคคาทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2558

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

AUTOMATIC FEEDING MACHINE SYSTEM WITH
MICROCONTROLLER



THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN MECHATRONICS ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2015

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาานิพนธ์ปีการศึกษา 2558

ภาควิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง ระบบเครื่องจักรป้อนชิ้นงานอัตโนมัติโดยไมโครคอนโทรลเลอร์
AUTOMATIC FEEDING MACHINE SYSTEM WITH MICROCONTROLLER

ผู้จัดทำ	นางสาวฉัตรสุดา	เพชรช้าง	55010200
	นางสาวพาขวัญ	นพคุณ	55010848
	นายอับดุลฮาฟิซ	เจ๊ะโซะ	55011460



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบเครื่องจักรป้อนชิ้นงานอัตโนมัติโดยไมโครคอนโทรลเลอร์

โดย

นางสาวฉัตรสุดา	เพชรช้าง	55010200
นางสาวพาขวัญ	นพคุณ	55010848
นายอับดุลฮาฟิซ	เจ๊ะโสะ	55011460

อาจารย์ที่ปรึกษา

ผู้ช่วยศาสตราจารย์สุมิตร พนาอุดมทรัพย์

รองศาสตราจารย์ ดร.เกียรติศักดิ์ คมวัชระ

ปีการศึกษา 2558

บทคัดย่อ

การศึกษาระบบหุ่นยนต์และแขนกล มีความจำเป็นต้องพัฒนาอย่างยิ่งสำหรับยุคอุตสาหกรรมที่เจริญก้าวหน้าในปัจจุบัน เพื่อทำงานแทนแรงงานมนุษย์อันเนื่องมาจากเหตุผลทางด้านความปลอดภัย ความรวดเร็วและแม่นยำ และที่สำคัญคือเพื่อเพิ่มผลผลิตให้สามารถก้าวทันคู่แข่งในตลาดได้ ดังนั้นโครงการนี้จึงได้ออกแบบและพัฒนาเครื่องจักรป้อนชิ้นงานสำหรับใช้ในโรงงานอุตสาหกรรม โดยได้จัดทำเป็นหุ่นจำลองตัวต้นแบบเคลื่อนที่สองแกน ประกอบด้วยมอเตอร์ไฟฟ้ากระแสตรง 24 โวลต์ จำนวน 5 ตัว ใช้ในการหมุนเคลื่อนข้อต่อต่างๆ สามารถเคลื่อนที่ขึ้นลง และวิ่งไปตามรางในแนวราบได้ ใช้แม่เหล็กไฟฟ้าขนาด 12 โวลต์ เป็นส่วนจับชิ้นงานบริเวณปลายเครื่องจักรที่ใช้สำหรับจับวัตถุประเภทเหล็ก และควบคุมการทำงานของมอเตอร์โดยใช้ไมโครคอนโทรลเลอร์ ATmega328 ให้สามารถทำงานได้อัตโนมัติ ลักษณะการทำงานของเครื่องจักรป้อนชิ้นงานนี้สามารถเคลื่อนย้ายวัตถุจากตำแหน่งหนึ่งไปยังอีกตำแหน่งหนึ่งได้ตามต้องการ และส่วนปลายมือของเครื่องสามารถหมุนได้เพื่อใช้ในการจับพลิกหงายวัตถุ โดยการทดสอบการทำงาน จะกำหนดตำแหน่งเป้าหมายที่ต้องการวางวัตถุ แล้วเขียนโปรแกรมกำหนดการทำงานของมอเตอร์ทุกตัวให้สัมพันธ์กันตลอดจนการหยิบจับวัตถุ ทำให้สามารถใช้งานได้หลากหลายรูปแบบ สามารถนำไปปรับใช้ให้ตรงกับรูปแบบของเครื่องจักรต่างๆ ในโรงงานได้ต่อไป

AUTOMATIC FEEDING MACHINE WITH MICROCONTROLLER

By

Miss Chatsuda Petchang 55010200

Miss Phakwan Nopkhun 55010848

Mr. Abdulhafiz Chesof 55011460

Advisors

Asst.Prof. Sumit Panaudomsup

Assoc.Prof.Dr. Kiattisak Kumwachara

Academic Year 2015

ABSTRACT

Nowadays, the industry is growing up therefore automatic machine foundations learning especially should be developed. The main robots function can help people working in industry because there is some problem that happened from worker, such as accuracy, speed, human error or safety in human life. The main purpose for this project wants to increased productivity and delivered quality uniform. Consequently, the project developers have designed and developed an Automatic Feeding Machine model for using in industry that consist of 5 of 24 volt DC motors that used for rotate the joints, move vertically upwards and downwards and it can move horizontally on rail track. The end of the machine arm can grab metal objects by using 12 volt electromagnets. This machine can move continuously and automatically by using microcontroller ATmega328. Machine working operations are able to pick up and to move objects from one place to another and the hand can be rotated to turn up the objects. For the functional test, developers have defined the desired position and write the program to control all motors that related to movement and pick up the objects. From the variable function, this Automatic feeding machine can be controlled to move the requiring position and it apply to others type of industrial machinery.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
II
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้ดำเนินการสำเร็จลุล่วงไปได้ด้วยดีด้วยความช่วยเหลือจากอาจารย์ที่ปรึกษา ผู้ช่วยศาสตราจารย์สุมิตร พนาอุดมทรัพย์ ซึ่งท่านได้ให้ข้อคิดเห็นและคำแนะนำต่างๆ อันเป็นประโยชน์อย่างยิ่งในการดำเนินงาน อีกทั้งยังช่วยแก้ปัญหาต่างๆ ที่เกิดขึ้นด้วย

ขอขอบคุณคณะอาจารย์และเจ้าหน้าที่ภาควิชาวิศวกรรมการวัดและควบคุม รวมถึงเพื่อนพี่น้องสถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ให้ความช่วยเหลือและให้คำแนะนำที่ดีมาโดยตลอด

สุดท้ายนี้ คณะผู้จัดทำขอกราบขอบพระคุณผู้ปกครองและครอบครัว ซึ่งให้การสนับสนุนในด้านต่างๆ และคอยเป็นกำลังใจมาโดยตลอดจนสำเร็จการศึกษา คณะผู้จัดทำรู้สึกซาบซึ้งเป็นอย่างยิ่งสำหรับทุกความช่วยเหลือ และขอขอบคุณทุกท่านที่มีส่วนเกี่ยวข้องมา ณ ที่นี้ด้วย



สารบัญ

	หน้า
บทคัดย่อภาษาไทย	I
บทคัดย่อภาษาอังกฤษ	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญรูป	VI
สารบัญตาราง	VIII
บทที่ 1 บทนำ	1
1.1 ความสำคัญของปริญญานิพนธ์	1
1.2 วัตถุประสงค์ในการทำปริญญานิพนธ์	1
1.3 ขอบเขตการศึกษา	2
1.4 รายละเอียดของปริญญานิพนธ์	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
1.6 แผนการดำเนินโครงการ	3
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	4
2.1 มอเตอร์ไฟฟ้ากระแสตรง	4
2.1.1 หลักการทำงานของมอเตอร์กระแสตรง	4
2.1.2 การควบคุมมอเตอร์ขั้นพื้นฐาน	5
2.2 วงจรที่ใช้ในการควบคุมมอเตอร์	8
2.3 ไมโครคอนโทรลเลอร์ (Microcontroller)	11
2.3.1 คุณสมบัติและการต่อใช้งาน Arduino Uno R3	11
2.3.2 คุณสมบัติและการต่อใช้งาน Arduino Nano 3.0	15
2.4 การสื่อสารข้อมูลแบบอนุกรมและโปรโตคอล	16
2.5 ตัวต้านทานแบบเปลี่ยนค่าได้ (Variable Resistor)	16
2.6 Robot Kinematics	18
บทที่ 3 การออกแบบวงจร การเขียนโปรแกรม และการสร้างแบบจำลอง	21
3.1 วงจรควบคุมความเร็วมอเตอร์พื้นฐาน	21
3.2 การออกแบบวงจรขับมอเตอร์ (Drive Motor Circuit)	22

สารบัญ (ต่อ)

	หน้า
3.2.1 การพัฒนางจรขับเคลื่อนมอเตอร์	22
3.2.2 วงจรเรกกูเรเตอร์ (Regulator)	26
3.2.3 วงจรจำกัดกระแส (Limit Current)	27
3.2.4 วงจรลิมิตสวิตช์	29
3.2.5 วงจรเอนโค้ดเดอร์ (Encoder)	29
3.3 หลักการสร้างแบบจำลองของเครื่องจักรป้อนชิ้นงานอัตโนมัติ	30
3.3.1 วัสดุที่ใช้ในการสร้างแบบจำลองเครื่องจักรป้อนชิ้นงานอัตโนมัติ	30
3.3.2 ขั้นตอนการประกอบและติดตั้ง	31
3.4 Inverse Kinematic	33
3.5 การเขียนโปรแกรมควบคุมและการสื่อสารข้อมูล	36
บทที่ 4 การทดลองและผลการทดลอง	41
4.1 การทดสอบวงจรขับเคลื่อนมอเตอร์ด้วยการใช้ Arduino	41
4.1.1 การทดสอบวงจรควบคุมความเร็วมอเตอร์โดยใช้ตัวต้านทานปรับค่าได้	41
4.1.2 การทดลองควบคุมความเร็ว	41
4.1.3 การทดลองควบคุมตำแหน่ง	43
4.1.4 การทดลองใช้วงจรกับการปรับความเร็ว	43
4.1.5 การทดสอบการทำงานของบอร์ดวงจรขับเคลื่อนมอเตอร์	45
4.2 การทดสอบการทำงานของโปรแกรม	45
4.2.1 ทดสอบการทำงานของโปรแกรมโดยใช้ตัวต้านทานปรับค่าได้	45
4.2.2 ทดสอบการทำงานของเอนโค้ดเดอร์	46
4.2.3 ทดสอบความแม่นยำในการเคลื่อนที่เข้าหาเป้าหมาย	48
บทที่ 5 สรุปและวิจารณ์ผลการทดลอง	51
5.1 สรุปผลการทดลอง	51
5.2 ปัญหาและแนวทางแก้ไขปัญหา	51
5.3 ข้อเสนอแนะและแนวทางพัฒนา	52
เอกสารอ้างอิง	53
ภาคผนวก	54

สารบัญรูป

รูปที่	หน้า
2.1 วงจรภายในของมอเตอร์กระแสตรง	4
2.2 ลักษณะของสัญญาณ PWM ที่ %Duty Cycle ต่างๆ	6
2.3 ระบบควบคุมตำแหน่งที่ประกอบด้วย การควบคุมแบบป้อนกลับ	6
2.4 การควบคุมแบบ PI	7
2.5 การสร้าง Pulse ของ Rotary Encoder	7
2.6 วงจร H-Bridge ที่ควบคุมโดยหน้าสัมผัสสี่เหลี่ยม	8
2.7 วงจรการควบคุมแบบลิเนียร์	9
2.8 วงจรการควบคุมความเร็ว DC Motor แบบสวิตชิง หรือ PWM	9
2.9 การเคลื่อนที่กลับทิศของมอเตอร์ในวงจร H-Bridge ที่ควบคุมด้วย MOSFET	10
2.10 โครงสร้างภายนอกของ Arduino Uno R3	11
2.11 ตำแหน่งของขาบน ATmega328	12
2.12 Timing Diagram ในโหมด Fast PWM	13
2.13 บอร์ด Arduino Nano 3.0	15
2.14 ตำแหน่งของพอร์ตการต่อใช้งานของ Arduino Nano 3.0	15
2.15 รูปแบบการส่งสัญญาณสื่อสาร	16
2.16 วงจรแบ่งแรงดัน	17
2.17 การต่อใช้งานตัวต้านทานเปลี่ยนค่าได้ กับ Arduino Nano 3.0	17
3.1 การต่อวงจรควบคุมความเร็วมอเตอร์โดยใช้ตัวต้านทานปรับค่าได้อย่างง่าย	21
3.2 ผังงานโปรแกรม (Flow Chart) ของการเขียนโปรแกรมควบคุมความเร็วมอเตอร์ที่ใช้ตัวต้านทานปรับค่าได้	22
3.3 การต่อวงจร H-Bridge	23
3.4 ผังงานโปรแกรม (Flow Chart) ของการเขียนโปรแกรมควบคุมตำแหน่ง	24
3.5 ขาของไอซี HIP4080	24
3.6 Application Block Diagram ของ HIP4080	25
3.7 การต่อวงจรควบคุมมอเตอร์	26
3.8 การต่อวงจรเรกกูเรเตอร์ (Regulator) และ LM7812	26
3.9 การต่อวงจรป้องกันไฟไหลย้อนกลับของวงจรเรกกูเรเตอร์ (Regulator)	27
3.10 การต่อวงจรเรกกูเรเตอร์ของ LM78L05	27

สารบัญรูป (ต่อ)

รูปที่	หน้า
3.11 วงจรขับมอเตอร์โดยสมบูรณ์	28
3.12 กัดปรินลายวงจร และบัตรกร์ไอซีลงบอร์ด	28
3.13 บอร์ดวงจรขับมอเตอร์ (Drive Motor Circuit) ทั้ง 4 ตัว	28
3.14 การต่อวงจรลิมิตสวิตช์	29
3.15 การต่อวงจรเอนโค้ดเดอร์	29
3.16 ภาพจำลองการใช้งานของเครื่องจักรป้อนชิ้นงานอัตโนมัติกับเครื่องจักรในโรงงาน	30
3.17 วัสดุอุปกรณ์ในการสร้างแบบจำลองแขนกล	30
3.18 แบบจำลองเครื่องจักรป้อนชิ้นงานอัตโนมัติ	32
3.19 Inverse Kinematic	33
3.20 ระบบที่ใช้ในการควบคุม	36
3.21 การสื่อสารแบบอนุกรม	36
3.22 Flow Chart แสดงการทำงานของ Controller ตัว Master	37
3.23 Flow Chart แสดงการทำงานของ Controller ตัว Slave	40
4.1 กราฟผลการทดสอบวงจรควบคุมความเร็วมอเตอร์โดยใช้ตัวต้านทานปรับค่าได้	41
4.2 การต่อวงจรชmittริกเกอร์ (Schmitt Trigger)	42
4.3 สัญญาณกระแสไฟที่ไหลในวงจร	42
4.4 การทดลองควบคุมตำแหน่งด้วยการใช้ Arduino	43
4.5 การต่อวงจร H-Bridge โดยใช้ HIP 4080	44
4.6 กราฟแสดงผลของการปรับความถี่ที่ค่าต่างๆ ตั้งแต่ 1 kHz – 30 kHz	44
4.7 การต่อวงจรทดสอบการทำงานของบอร์ด	45
4.8 การวงจรต่อตัวต้านทานปรับค่าได้	46
4.9 กราฟแสดงระยะทางที่หุ่นเคลื่อนที่ได้	48
4.10 การทดสอบความแม่นยำ	49
4.11 กราฟแสดงผลการเคลื่อนที่และความคลาดเคลื่อนจากตำแหน่งเป้าหมาย	50

สารบัญตาราง

ตารางที่	หน้า
1.1 แผนการดำเนินโครงการระหว่างเดือนสิงหาคม 2558 ถึงเดือนเมษายน 2559	3
2.1 TCCR2A – Timer/Counter Control Register A	13
2.2 Waveform Generation Mode Bit Description	14
2.3 TCCR2B – Timer/Counter Control Register B	14
2.4 Clock Select Bit Description	15
2.5 การถ่ายโอนโครงพิกัด k ไปยังโครงพิกัด k-1	19
3.1 Truth Table ของไอซี HIP 4080	25
4.1 ผลการทดสอบให้เครื่องหยิบจับชิ้นงานวิ่งไปยังตำแหน่งเป้าหมาย	47
4.2 ผลการทดสอบความแม่นยำในการเคลื่อนที่ไปยังตำแหน่งเป้าหมาย	49



บทที่ 1

บทนำ

1.1 ความสำคัญของปัญญาประดิษฐ์

ระบบอุตสาหกรรมในปัจจุบัน มีความจำเป็นต้องพึ่งเครื่องจักรมาทำงานแทนแรงงานมนุษย์ โดยเฉพาะอุตสาหกรรมด้านการผลิตสินค้า อันเนื่องมาจากปัญหาด้านความปลอดภัยในชีวิต งานที่หนักเกินแรงงานมนุษย์ การขาดแคลนแรงงาน รวมทั้งความแม่นยำ และความรวดเร็วในการทำงาน ซึ่งโรงงานต่างก็ต้องแข่งขันกันเพื่อผลกำไรและลดต้นทุนลง ดังนั้นการศึกษาและพัฒนาเครื่องจักรอัตโนมัติที่ใช้อุตสาหกรรม มีความจำเป็นต้องพัฒนาอย่างยิ่งสำหรับยุคอุตสาหกรรมที่เจริญก้าวหน้าไปอย่างรวดเร็วและต่อเนื่อง ทั้งทางด้านอุปกรณ์เครื่องใช้ต่างๆ และกระบวนการผลิตสิ่งต่างๆ เพื่อการตอบสนองความต้องการความสะดวกสบายของมนุษย์ จึงทำให้ภาคอุตสาหกรรมต้องเพิ่มศักยภาพในการผลิต โดยการนำเครื่องจักรอัตโนมัติซึ่งสามารถตอบสนองทางด้าน ความแม่นยำ ความรวดเร็ว ประสิทธิภาพ การลดต้นทุนในการผลิต อีกทั้งเวลาในการทำงานยังมากกว่ามนุษย์ ที่สำคัญสามารถทำงานในสถานะที่อันตรายได้ ดังนั้นเครื่องจักรอัตโนมัติจึงเป็นอีกทางเลือกหนึ่งที่น่าสนใจที่จะเข้ามาแทนที่มนุษย์ ที่อาจเกิดความเสี่ยงเกี่ยวกับเรื่องความปลอดภัยได้

ในปัจจุบัน เครื่องจักรเป็นหนึ่งในเทคโนโลยีที่ได้รับความนิยมเป็นอย่างมาก อีกทั้งยังสามารถเพิ่มประสิทธิภาพในการผลิตได้ คณะผู้จัดทำจึงเล็งเห็นความสำคัญ และประโยชน์ จึงได้จัดทำเครื่องจักรป้อนชิ้นงานอัตโนมัติเพื่อศึกษาการออกแบบ การควบคุม เครื่องจักรให้ทำงานได้ตามต้องการ

1.2 วัตถุประสงค์ในการทำปัญญาประดิษฐ์

1. เพื่อออกแบบวงจรขั้วมอเตอร์
2. เพื่อสร้างโค้ดของโปรแกรมที่ใช้ในการควบคุมตำแหน่งและทิศทางของมอเตอร์ มาใช้ในการควบคุมระบบการทำงานของเครื่องจักรป้อนชิ้นงานอัตโนมัติ ซึ่งเขียนโดยใช้โปรแกรม Arduino
3. เพื่อให้เกิดความรู้ ความเข้าใจในส่วนการออกแบบและการเคลื่อนที่ของเครื่องจักรป้อนชิ้นงานอัตโนมัติ
4. ทำให้ได้ต้นแบบของเครื่องจักรป้อนชิ้นงานอัตโนมัติ และนำต้นแบบที่ได้ไปพัฒนาให้สามารถใช้ในระบบอุตสาหกรรมได้จริง

1.3 ขอบเขตการศึกษา

1. สามารถออกแบบวงจรขับเคลื่อนมอเตอร์ โดยใช้โปรแกรม Altium Designer
2. สามารถใช้โปรแกรม Arduino ในการควบคุมการทำงานของมอเตอร์ให้เคลื่อนที่ไปในตำแหน่งและทิศทางที่ต้องการได้
3. สามารถกำหนดมุมการทำงาน ตำแหน่งของส่วนจับชิ้นงาน ควบคุมเครื่องจักรให้เคลื่อนที่และหยิบชิ้นงานได้

1.4 รายละเอียดของปฏิญานិพนธ์

เนื้อหาที่จะกล่าวในปฏิญานิพนธ์ฉบับนี้ประกอบด้วย

บทที่ 1 บทนำ กล่าวถึง วัตถุประสงค์ ขั้นตอนการศึกษา ขอบเขตของการศึกษา และการจัดทำโครงการ พร้อมทั้งรายละเอียดของปฏิญานิพนธ์ในแต่ละบท

บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้องของหลักการ และทฤษฎีที่เกี่ยวข้องกับการออกแบบเครื่องจักรป้อนชิ้นงานอัตโนมัติ ทฤษฎีระบบควบคุม หลักการทำงานของดีซีมอเตอร์ และวงจรอิเล็กทรอนิกส์ที่ใช้ในการควบคุมมอเตอร์ รวมทั้งการนำความรู้ไปประยุกต์ใช้ในการจัดทำโครงการ

บทที่ 3 หลักการออกแบบวงจรควบคุมมอเตอร์ หลักการออกแบบโปรแกรมที่ใช้ในการควบคุมมอเตอร์ และหลักการสร้างแบบจำลองของเครื่องจักรป้อนชิ้นงานอัตโนมัติ

บทที่ 4 การทดลอง แสดงผลการทดสอบเครื่องจักรป้อนชิ้นงานอัตโนมัติ

บทที่ 5 บทวิจารณ์และสรุป สรุปผลการดำเนินงาน ปัญหาที่เกิดขึ้น และแนวทางการปรับปรุง

1.5 ประโยชน์ที่คาดว่าจะได้รับ

1. เข้าใจการออกแบบวงจรควบคุมมอเตอร์ และการออกแบบลายวงจร โดยใช้โปรแกรม Altium Designer
2. ได้ใช้ความคิดในการทำโปรเจ็ค เพื่อเป็นการฝึกให้มีความรู้ที่เป็นระบบ และมีความรับผิดชอบมากยิ่งขึ้น
3. เข้าใจการเขียนโปรแกรม แก๊ซคำสั่ง และการควบคุมการทำงานของมอเตอร์มากยิ่งขึ้น
4. ทำให้ได้ต้นแบบในการสร้างเครื่องจักรป้อนชิ้นงานอัตโนมัติ
5. รู้ปัญหาและแนวทางในการพัฒนาเครื่องจักรป้อนชิ้นงานอัตโนมัติเพื่อให้สอดคล้องกับการใช้งานจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.6 แผนการดำเนินงาน

ตารางที่ 1.1 แผนการดำเนินงานระหว่างเดือนสิงหาคม 2558 ถึงเดือนเมษายน 2559

ขั้นตอน การดำเนินงาน	2558					2559			
	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.	มี.ค.	เม.ย.
1. ศึกษาข้อมูลและ ทฤษฎีที่เกี่ยวข้อง	←→								
2. ศึกษาวงจรขับ มอเตอร์	←→								
3. ออกแบบวงจรและ ทดสอบการทำงาน	←→								
4. ซึ้ออุปกรณ์สำหรับ ทำวงจรขับมอเตอร์	←→								
5. ศึกษาการเขียน โปรแกรม	←→								
6. ออกแบบหุ่นตัว ต้นแบบ				←→					
7. เตรียมซื้ออุปกรณ์ สำหรับทำวงจรและ ทำหุ่นตัวต้นแบบ				←→					
8. ทำวงจรขับมอเตอร์				←→					
9. เริ่มสร้างและพัฒนา หุ่นตัวต้นแบบ						←→			
10. เขียนโปรแกรมและ ทดสอบการทำงาน						←→			
11. ตรวจสอบและแก้ไข ปัญหา								←→	
12. สรุปผลการ ดำเนินงาน									←→

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

2.1 มอเตอร์ไฟฟ้ากระแสตรง

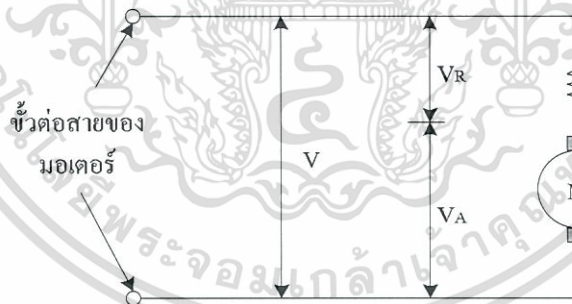
2.1.1 หลักการทำงานของมอเตอร์กระแสตรง

เมื่อมีการผ่านกระแสไฟฟ้าเข้าไปยังขดลวดในสนามแม่เหล็กจะทำให้เกิดแรงแม่เหล็กซึ่งมีสัดส่วนของแรงขึ้นกับกระแสแรงของสนามแม่เหล็ก โดยแรงจะเกิดขึ้นเป็นมุมฉากกับกระแสและสนามแม่เหล็ก ขณะที่ทิศทางของแรงกลับตรงกันข้ามกัน ถ้าหากกระแสของสนามแม่เหล็กไหลย้อนกลับจะทำให้เกิดการเปลี่ยนแปลงของกระแสและสนามแม่เหล็ก เป็นผลทำให้ทิศทางของแรงเปลี่ยนไป ด้วยคุณสมบัตินี้ทำให้มอเตอร์กระแสตรงกลับทิศทางหมุนได้

สนามแม่เหล็กของมอเตอร์ส่วนหนึ่งเกิดขึ้นจากแม่เหล็กถาวร ซึ่งจะถูกยึดติดกับแผ่นเหล็กหรือเหล็กกล้า โดยปกติส่วนนี้จะเป็นส่วนที่ยึดอยู่กับที่ และขดลวดเหนี่ยวนำจะพันอยู่กับส่วนที่เป็นแกนหมุนของมอเตอร์

คุณสมบัติของมอเตอร์กระแสตรง

ในการอธิบายคุณสมบัติของมอเตอร์กระแสตรงให้ละเอียดนั้นต้องพิจารณาแรงดันที่ป้อนและความต้านทานของโรเตอร์ด้วย วงจรภายในของมอเตอร์เขียนได้ดังรูปที่ 2.1



รูปที่ 2.1 วงจรภายในของมอเตอร์กระแสตรง

โดยสมมติให้ฟลักซ์โรเตอร์ไม่มีความต้านทานอยู่เลย อนุกรมกับความต้านทานซึ่งในที่นี้ก็คือความต้านทานของขดลวดนั่นเอง แรงดันที่ขั้วต่อสายของมอเตอร์ก็คือผลบวกระหว่างแรงดันที่ฟลักซ์โรเตอร์ (V_A) และแรงดันตกคร่อมความต้านทานขดลวด (V_R)

แรงดัน V_A ถูกเรียกว่า แรงเคลื่อนเหนี่ยวนำบ่อนกลับ (BACK EMF) ซึ่งเกิดขึ้นในโรเตอร์ ขณะที่หมุน แรงดันที่เกิดขึ้นนี้เป็นไปตามกฎของการเหนี่ยวนำแม่เหล็กไฟฟ้าจากการเคลื่อนที่ของ ตัวนำในสนามแม่เหล็ก สัมพันธ์กับแรงเคลื่อนเหนี่ยวนำแม่เหล็ก และความเร็วในการเคลื่อนที่ของ ตัวนำ แรงดันที่เกิดขึ้นจะมีขั้วตรงกันข้ามกับแรงดันที่ป้อนให้กับมอเตอร์ และแปรผันตรงกับความเร็ว ในการหมุน ผลบวกของแรงดันที่พุนโรเตอร์ (V_A) และแรงดันตกคร่อมขดลวด (V_R) ต้องเท่ากับแรงดัน ที่ป้อนให้กับมอเตอร์ (V)

$$V = V_A + V_R \quad (V) \quad (2.1)$$

เมื่อพิจารณาตั้งแต่มอเตอร์หยุดนิ่ง ความเร็วมีค่าเป็นศูนย์ ดังนั้น $V_A = 0$, $V_R = V$ กระแสที่ ไหลในมอเตอร์หาได้จาก

$$I = V_R / R \quad (A) \quad (2.2)$$

เมื่อมอเตอร์เริ่มหมุนจะมีความเร็ว และ V_A เพิ่มขึ้นเป็นเส้นตรงตามความเร็ว V_R ซึ่งมีค่า เท่ากับความแตกต่างระหว่าง V_A และ V จะเริ่มลดลงกระแส I ก็เริ่มลดลงเช่นกันขณะที่มอเตอร์ยัง มีความเร่งอยู่ ความเร็วจะเพิ่มขึ้น แรงบิดจะลดลงจนกว่าจะถึงจุดซึ่งแรงบิดของมอเตอร์รับภาระโหลด ได้สมดุลพอดีขณะที่มอเตอร์ไม่มีโหลด และหมุนอย่างอิสระจะมีเพียงค่าความถี่ของแบร์ริง และแรง ต้านอากาศทำให้ V_A เกือบเท่ากับค่า V

2.1.2 การควบคุมมอเตอร์ขั้นพื้นฐาน

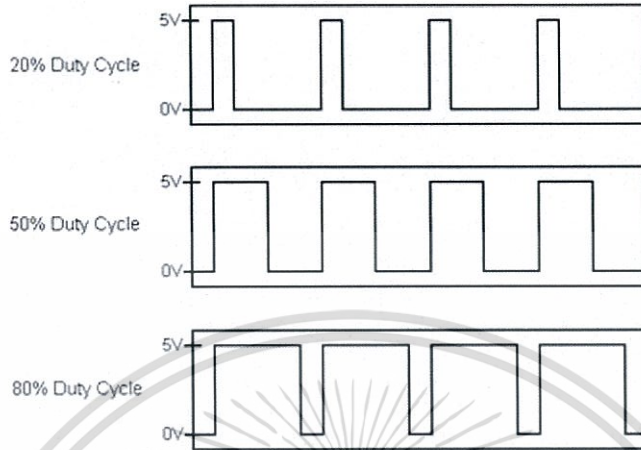
การควบคุมมอเตอร์ประกอบไปด้วยการควบคุมความเร็ว และการควบคุมตำแหน่งหรือมุม ของมอเตอร์ ซึ่งมีรายละเอียดดังนี้

1. การควบคุมความเร็วของมอเตอร์

PWM (Pulse Width Modulation) เป็นการควบคุมความเร็วโดยการปรับความกว้างของ พัลส์ โดยการนำเอาสัญญาณสองสัญญาณมาเปรียบเทียบกัน ประกอบด้วย สัญญาณสามเหลี่ยมและ สัญญาณที่ต้องการปรับความกว้างของพัลส์ ซึ่งสร้างได้ด้วยการใช้วงจร Comparator หรือวงจร เปรียบเทียบ สามารถสร้างได้ด้วยการใช้ Op-Amp ตัวเดียว อีกทั้งไมโครคอนโทรลเลอร์ส่วนมากจะ สามารถสร้างสัญญาณ Output PWM ได้เช่นเดียวกัน

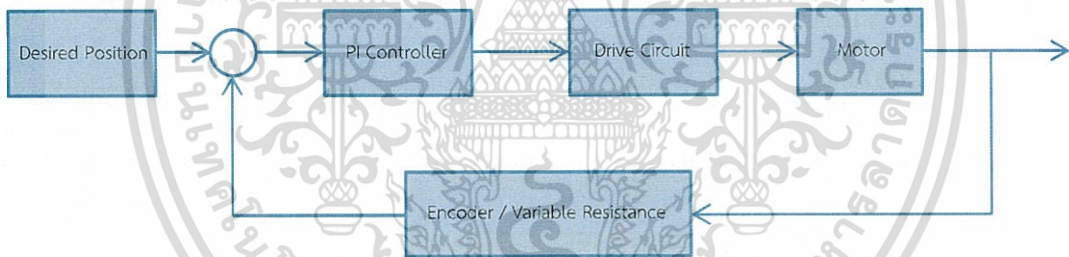
ข้อดีของสัญญาณ PWM คือ มีประสิทธิภาพเพราะ Power Supply จะจ่ายกำลังได้เต็มที่ทั้ง ON และ OFF โดยในโครงงานนี้จะมีการใช้มอสเฟตเป็นตัวตัด-ต่อวงจรของ H-bridge เพื่อสามารถที่

จะควบคุมจังหวะในการจ่ายกระแสด้วยการ ON-OFF ในสัดส่วน (Duty Cycle) ที่ต่างกันด้วยความถี่ที่เหมาะสม หากไม่เช่นนั้นที่ความถี่ต่ำมอเตอร์จะมีการกระตุก แต่ที่ความถี่ที่สูงมากก็จะมีผลสูญเสียพลังงานในวงจรที่มากขึ้นเช่นกัน



รูปที่ 2.2 ลักษณะของสัญญาณ PWM ที่ %Duty Cycle ต่างๆ

2. การควบคุมตำแหน่งและมุมของมอเตอร์

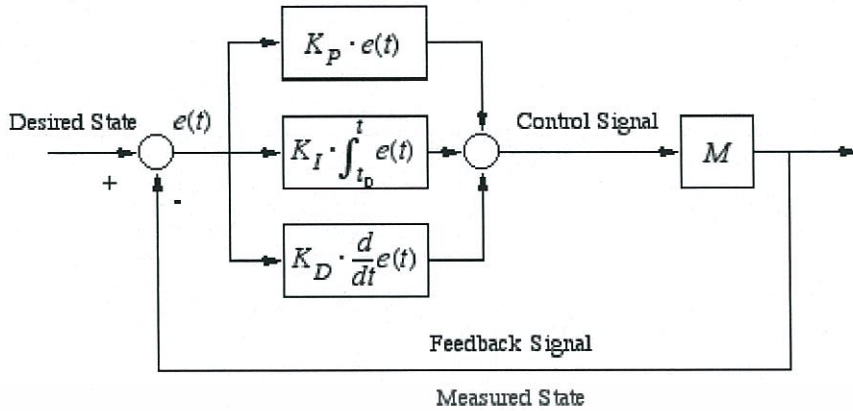


รูปที่ 2.3 ระบบควบคุมตำแหน่งที่ประกอบด้วย การควบคุมแบบป้อนกลับ

จากรูปที่ 2.3 คือ บล็อกไดอะแกรมของการควบคุมตำแหน่งมอเตอร์ DC เมื่อมีการกำหนดตำแหน่งให้กับไมโครคอนโทรลเลอร์ ไมโครคอนโทรลเลอร์จะทำการสร้างสัญญาณควบคุมให้กับวงจรขับมอเตอร์ เมื่อมอเตอร์มีการหมุน Encoder จะทำการตรวจจับตำแหน่งที่มอเตอร์ได้หมุนพร้อมทั้งส่งสัญญาณป้อนกลับให้กับไมโครคอนโทรลเลอร์ สำหรับการคำนวณค่าผิดพลาด (Error) ของตำแหน่งระหว่าง Desired Position กับ Measured Position เพื่อนำค่าที่ได้ไปคำนวณสำหรับการควบคุมแบบ Proportional Plus Integral Plus Derivative (PID)

$$\text{Error (ค่าความผิดพลาดของตำแหน่ง/มุม)} = \text{Desired Position} - \text{Measured Position}$$

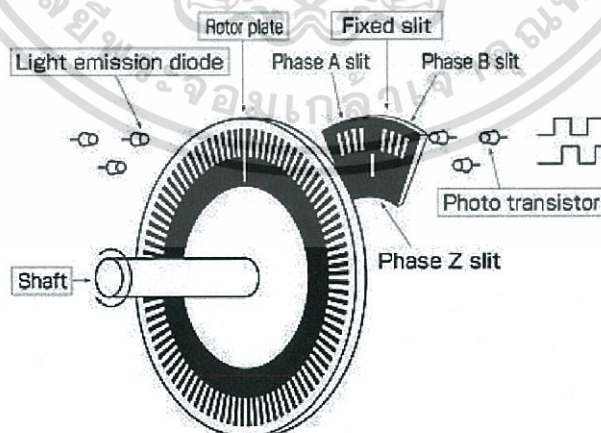
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.4 การควบคุมแบบ PI

เนื่องจากมีระบบควบคุมตำแหน่ง จึงได้มีการนำ Rotary Encoder มาใช้สำหรับในระบบตำแหน่งพร้อมทั้งทิศทางของการเคลื่อนที่ เพื่อที่จะนำมาเปรียบเทียบความแตกต่างระหว่าง Input สำหรับการควบคุมต่อไป

Incremental Encoder หรือโดยทั่วไปเรียกว่า Rotary Encoder จะสร้างสัญญาณพัลส์ (Pulse) ที่แปรผันตรงกับการหมุนของเพลามอเตอร์ หรือจะหมุนด้วยความเร็วเท่ากับเพลาของมอเตอร์นั่นเอง โดย Rotary Encoder จะประกอบด้วยจานหมุน (Rotary Disk) และอุปกรณ์ตรวจจับ (Sensor) โดยจานหมุนจะมีช่องเล็กๆ (Slit) เมื่อเพลาของมอเตอร์หมุนจะทำให้จานหมุนไปตัดลำแสงของ Sensor ทำให้ชุดรับแสงมีการรับสัญญาณเป็นช่วงๆ ส่งผลให้สัญญาณเอาต์พุต (Output) มีลักษณะ Pulse แสดงดังรูปที่ 2.5



รูปที่ 2.5 การสร้าง Pulse ของ Rotary Encoder

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

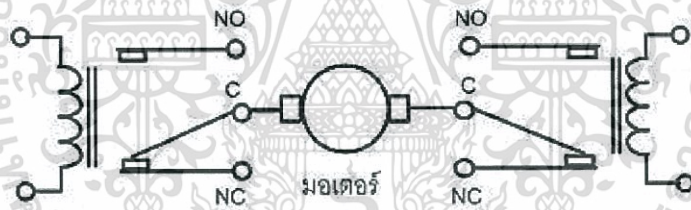
2.2 วงจรที่ใช้ในการควบคุมมอเตอร์

จากหลักการทำงานของมอเตอร์ไฟฟ้ากระแสตรง เมื่อมีกระแสตรงมาตัดผ่านสนามแม่เหล็ก จะเหนี่ยวนำให้เกิดทอร์กขึ้นในทิศทางที่เหมาะสม และทำให้เกิดการหมุนของแกนโรเตอร์ ในการควบคุมมอเตอร์กระแสตรงนั้นมีอยู่ 3 ลักษณะคือ

1. หมุน-หยุดหมุน (ON – OFF Control)
2. ทิศทางการหมุน (Direction Control)
3. ความเร็วการหมุน (Speed Control)

โดยทั่วไปวงจรที่ใช้ควบคุมจะเป็นวงจร H-Bridge ที่สามารถควบคุมการหมุน-หยุดหมุน พร้อมทั้งการกลับทิศทางการหมุนได้ จากการทำ Pulse Width Modulation (PWM) และควบคุมโดยการสั่งงานจากคอนโทรลเลอร์

อุปกรณ์ที่นิยมใช้ควบคุมส่วนมาก ได้แก่ หน้าสัมผัสรีเลย์ นำมาต่อเป็นวงจรในลักษณะ H-Bridge ที่สามารถควบคุมการหมุน-หยุดหมุน พร้อมทั้งการกลับทิศทางการหมุนได้ แต่การใช้หน้าสัมผัสรีเลย์จะไม่สามารถควบคุมความเร็วของการหมุนได้ และต้องระวังในการหมุนกลับทิศทางแบบทันทีทันใด เพราะอาจจะเกิดความเสียหายกับตัวมอเตอร์ได้ แต่ข้อดีของการใช้หน้าสัมผัสรีเลย์คือ ทำให้กระแสที่ไหลผ่านมอเตอร์ไม่ตกและสามารถใช้งานได้เต็มกำลัง

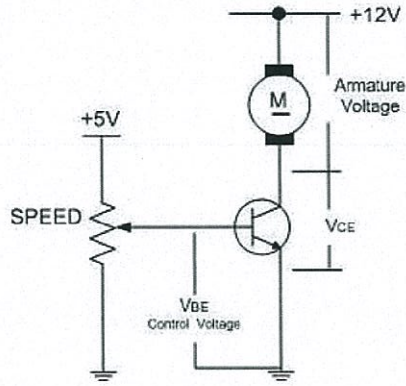


รูปที่ 2.6 วงจร H-Bridge ที่ควบคุมโดยหน้าสัมผัสรีเลย์

ในการควบคุมความเร็วมอเตอร์กระแสตรง (DC Motor) มีการควบคุม 2 แบบ คือ

1. การควบคุมแบบลิเนียร์ (Linear DC Voltage Control)

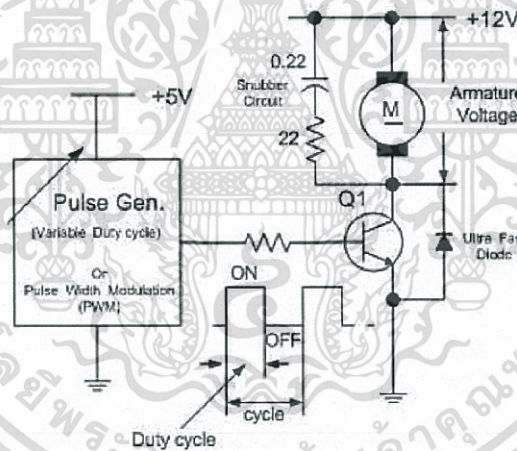
จากรูปที่ 2.7 พบว่าความเร็วของมอเตอร์จะแปรผันไปตามแรงเคลื่อนไฟฟ้าที่ตกคร่อมขดลวดอาเมเจอร์ (Armature Voltage) ดังนั้นหากสามารถปรับค่าแรงเคลื่อน V_{CE} ได้ ก็ทำให้ Armature Voltage เปลี่ยนแปลงตามไปด้วย แต่เนื่องจาก V_{CE} จะแปรผันตามแรงเคลื่อน V_{BE} โดยการปรับค่าตัวต้านทานที่ปรับค่าได้ ดังนั้นจะสามารถควบคุมความเร็วมอเตอร์ได้โดยการปรับค่าตัวต้านทาน Speed ได้ แต่การควบคุมวิธีนี้ไม่ค่อยนิยมใช้กันโดยเฉพาะมอเตอร์ที่มีกำลังมากๆ เนื่องจากจะเกิดพลังงานสูญเสียภายในตัวทรานซิสเตอร์ในรูปของความร้อนสูงมาก ในขณะที่ทรานซิสเตอร์ทำงานอยู่ในช่วง Active



รูปที่ 2.7 วงจรการควบคุมแบบลิเนียร์

2. การควบคุมแบบสวิทชิง (Switching Control)

คือ การป้อนไฟ V_{BE} หรือ Control Voltage ให้ทรานซิสเตอร์ทำงานในลักษณะ Cutoff หรือหยุดทำงาน และ Saturate หรือจุดอิ่มตัว โดยจะหลีกเลี่ยงการทำงานในช่วง Active ซึ่งจะทำให้มีการสูญเสียพลังงาน



รูปที่ 2.8 วงจรการควบคุมความเร็ว DC Motor แบบสวิทชิง หรือ PWM

จากรูปที่ 2.8 ประกอบไปด้วยตัวกำเนิดสัญญาณพัลส์ ส่งไปควบคุมการทำงานของทรานซิสเตอร์ในลักษณะ ON และ OFF ทำให้มีกระแสไหลผ่านมอเตอร์หมุนเป็นช่วงๆ ที่เป็นพัลส์บวก เรียกว่า “Duty Cycle” โดยที่ตัวผลิตพัลส์ (Pulse Generator) จะต้องรักษาช่วง Cycle หรือความถี่ให้คงที่อยู่ค่าใดค่าหนึ่ง โดยปกติจะเลือกค่าคงที่อยู่ระหว่าง 400Hz–2KHz สำหรับงานควบคุมมอเตอร์ ดังนั้นในทุกๆ Cycle หากมีการปรับเปลี่ยนค่าพัลส์บวก หรือค่า Duty Cycle ก็จะมีผลทำให้

ค่าเฉลี่ยแรงเคลื่อน Output ของทรานซิสเตอร์ที่ไปขับมอเตอร์เปลี่ยนแปลงตามไปด้วย ทำให้ความเร็วของมอเตอร์เปลี่ยนแปลงตามสมการที่ (2.3)

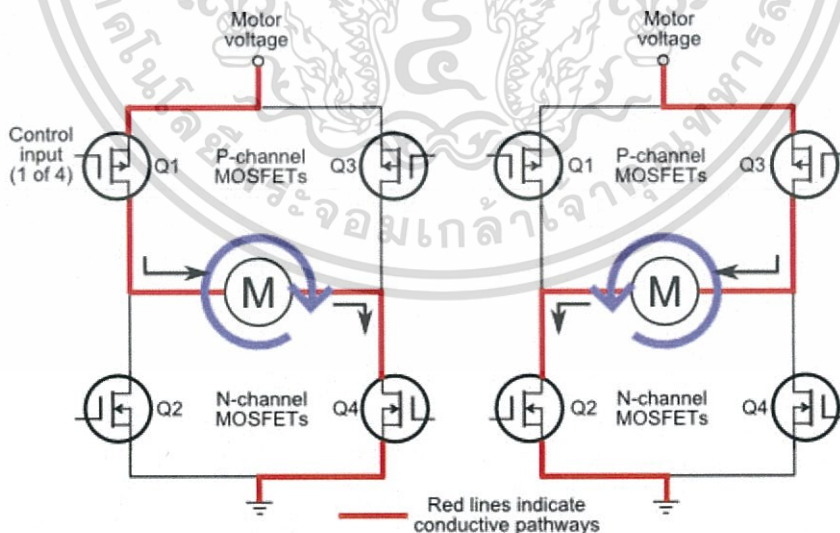
$$\%Duty\ cycle = \frac{Pulse\ Width}{Cycle} \times 100 \quad (2.3)$$

Duty Cycle จะคิดเป็นเปอร์เซ็นต์ของ Cycle และค่าเฉลี่ยของแรงเคลื่อน Output ขับโหลดได้จากสมการที่ (2.4)

$$V_{output} = \%Duty\ cycle \times V_{Supply} \quad (2.4)$$

ตามสมการที่ (2.4) พบว่าแรงเคลื่อน Output ที่จะออกไปขับมอเตอร์ให้หมุนที่ความเร็วเปลี่ยนไปนั้น จะแปรผันโดยตรงกับค่า %Duty Cycle หรือความกว้างของสัญญาณควบคุม PWM นั้นหมายความว่าสามารถเขียนโปรแกรมผลิตสัญญาณ PWM ที่เปลี่ยนค่า Pulse Width ให้ออกมาควบคุมวงจรขับมอเตอร์ได้ ก็จะสามารถควบคุมความเร็วของมอเตอร์ได้

ในกรณีต้องการควบคุมทั้งความเร็วและทิศทางการหมุนของมอเตอร์ไฟฟ้ากระแสตรง จำเป็นต้องใช้วงจร H-Bridge เพื่อให้สามารถเป็นสวิตช์ตัด-ต่อกระแสไฟควบคุมมอเตอร์แทนหน้าสัมผัสรีเลย์ที่เร็วพอที่จะใช้สัญญาณ PWM ควบคุมความเร็วไปพร้อมกับการควบคุมการกลับทิศทางการหมุนของมอเตอร์



รูปที่ 2.9 การเคลื่อนที่กลับทิศของมอเตอร์ในวงจร H-Bridge ที่ควบคุมด้วย MOSFET

จากรูปที่ 2.9 ประกอบด้วยมอสเฟต จำนวน 4 ตัวทำหน้าที่เป็นสวิตช์เปิด-ปิด (Q1-Q4) เมื่อส่งสัญญาณควบคุมให้มอสเฟต Q1 และ Q4 ทำงาน และปิดการทำงานของมอสเฟต Q2 และ Q3 จะทำให้กระแสไหลจากทิศทาง Q1 ไปยัง Q4 ส่งผลให้มอเตอร์เริ่มหมุนไปทางขวา และเมื่อส่งสัญญาณควบคุมให้มอสเฟต Q3 และ Q2 และปิดการทำงานของมอสเฟต Q1 และ Q4 ทำให้กระแสไหลจากจุด Q3 ไป Q2 ส่งผลให้มอเตอร์หมุนกลับทิศไปทางซ้าย

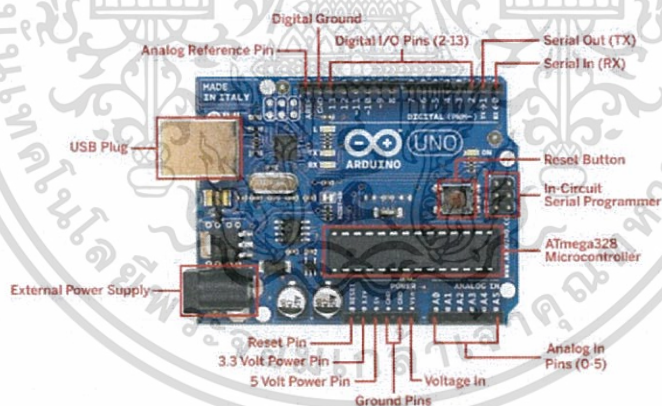
2.3 ไมโครคอนโทรลเลอร์ (Microcontroller)

2.3.1 คุณสมบัติและการใช้งาน Arduino Uno R3

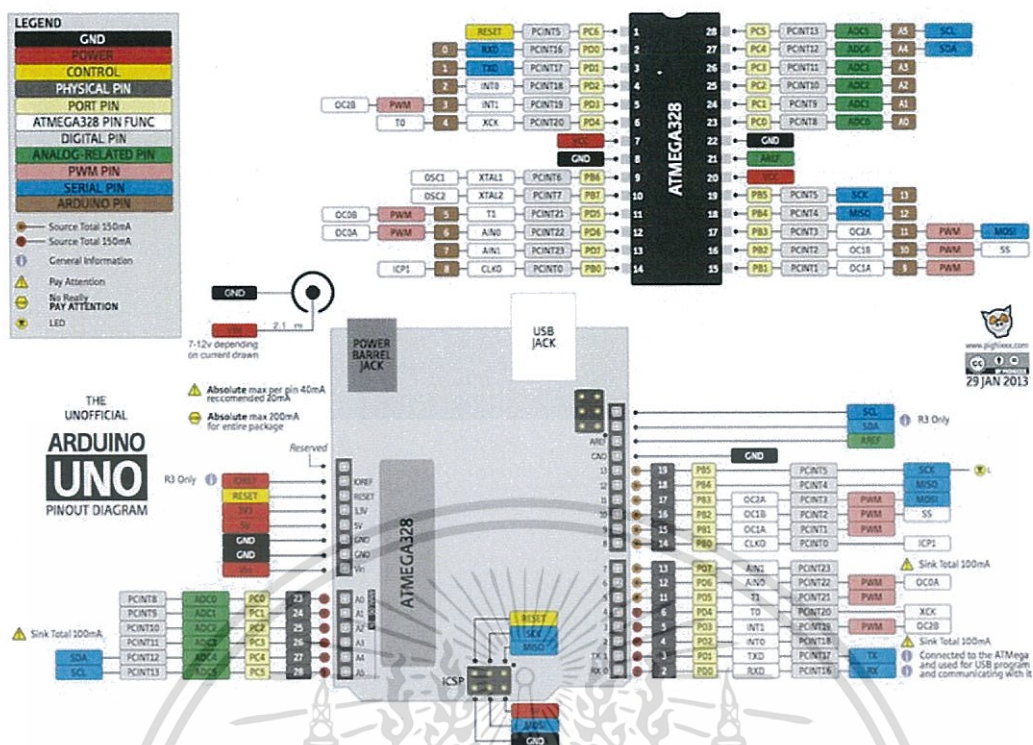
Arduino Uno R3 เป็น Microcontroller Board (MCU) ที่ใช้ ATmega328 เป็น MCU หลักที่รวบรวมอุปกรณ์สนับสนุนการทำงานของ CPU ไว้มากมาย อาทิเช่น Analog to Digital, SPI, UART, Timer, Counter และ PWM ซึ่งอุปกรณ์สนับสนุนการทำงานเหล่านี้ทำให้ MCU สามารถทำงานได้กว้างขวาง

คุณสมบัติของบอร์ดไมโครคอนโทรลเลอร์

บอร์ดไมโครคอนโทรลเลอร์จะมีขา Digital 14 ขา อินพุต/เอาต์พุต (สามารถทำเป็น PWM ได้ถึง 6 ขา) และมีขา Analog อินพุตได้อีก 6 ขา, Run ที่ความถี่ 16 MHz มี USB Connector และ Power Jack DC



รูปที่ 2.10 โครงสร้างภายนอกของ Arduino Uno R3



รูปที่ 2.11 ตำแหน่งของขาบน ATmega328

ฟังก์ชัน ADC

ปกติใน CPU ของ AVR – ATmega328 นั้นจะมีฟังก์ชัน ADC อยู่ภายในตัวไอซี สำหรับฟังก์ชัน ADC นี้สามารถรับสัญญาณอนาล็อกได้สูงสุด 8 Channel โดยรับสัญญาณเข้ามาทางพอร์ต A ซึ่งสามารถเลือกใช้ฟังก์ชันนี้ทำการแปลงสัญญาณอนาล็อกทีละ Channel อย่างต่อเนื่อง หรือจะทำการแปลงสัญญาณเฉพาะ Channel ที่ต้องการได้เช่นกัน โดยสัญญาณดิจิทัลที่แปลงได้จะมีความละเอียด 10 บิต โดยการรับสัญญาณแต่ละขาของพอร์ต A

ฟังก์ชัน PWM

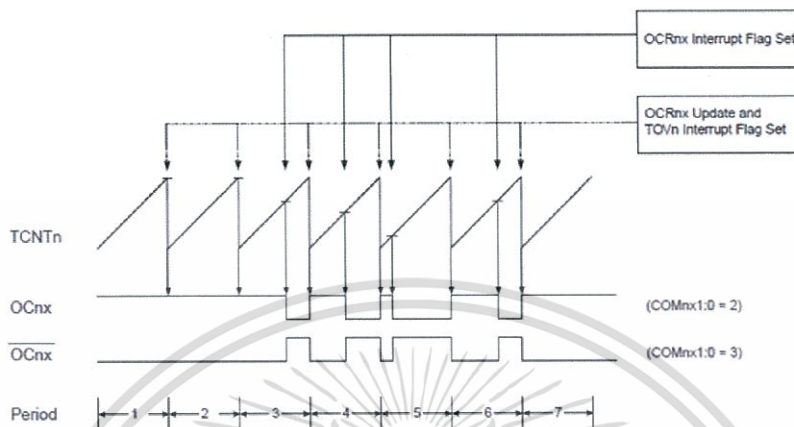
เนื่องจากว่า PWM เป็นฟังก์ชันการทำงานหนึ่งในโหมด PWM ของ Timer/Counter ที่อยู่ใน AVR-ATmega328 ดังนั้นจึงเกี่ยวข้องข้องกับการทำงานของฟังก์ชัน Timer/Counter ของ AVR-ATmega328

Timer/Counter

ภายใน AVR-ATmega328 จัดให้มี Timer/Counter 3 ชุด โดยจัดเป็น Timer/Counter ขนาด 8 บิต 2 ชุด และ Timer/Counter ขนาด 16 บิต 1 ชุด ดังนี้คือ Timer/Counter2 และ Timer/Counter0 และ Timer/Counter1 ซึ่ง Timer/Counter2 สามารถรับสัญญาณ Clock จากภายนอก ซึ่งเป็น Option ที่จะนำ Timer/Counter2 มาทำเป็น RTC โดยใช้ XTAL ที่มีค่าความถี่

เท่ากับ 32.768KHz มาเป็นฐานเวลา และ Timer/Counter0 และ Timer/Counter1 ใช้วงจร Prescaling ขนาด 10 บิตร่วมกัน ส่วน Timer/Counter2 ใช้วงจร Prescaling แยกออกต่างหาก

การใช้งาน Timer/Counter2 ในโหมด Fast PWM



รูปที่ 2.12 Timing Diagram ในโหมด Fast PWM

การทำงานในโหมด PWM ของ Timer/Counter2 โดยเอาต์พุตที่ได้จะออกที่ขา PD (OC2A) และขา PD (OC2B) ในการทำงาน Timer/Counter2 จะนับขึ้นจาก 0x00 จนถึงค่าเท่ากับค่าในรีจิสเตอร์ OCR2A หรือ OCR/B จะทำให้ขา (OC2A)/(OC2B) เปลี่ยนสถานะ และจะอยู่ในสถานะนั้นต่อไปจนกว่า TCNTn จะนับสะสมจนถึงค่า 0xFF

รีจิสเตอร์ที่ใช้งาน

จากที่ได้กล่าวมาแล้วในบทก่อนหน้านี้นี้ว่า ATmega328 ใน Arduino UNO R3 นั้นมี Timer/Counter อยู่ด้วยกัน 3 ตัว ในการทำงานครั้งนี้จะมีการเลือกใช้งาน 8-Bit Timer/Counter2 Prescaling ที่แยกออกต่างหากจาก Timer/Counter0 และ Timer/Counter1 โดยการใช้งานจะอธิบายในลำดับต่อไป

จากโปรแกรมข้างต้นจะสังเกตได้ว่าการ SET Bit ของ TCCR2A ซึ่งประกอบไปด้วย COM2A1, COM2A0, COM2B1, COM2B0, WGM21, WGM20 และ SET bit CS22 สำหรับ TCCR2B

ตารางที่ 2.1 TCCR2A – Timer/Counter Control Register A

Bit (0x80)	7	6	5	4	3	2	1	0	
Read/Write	R/W	R/W	R/W	R/W	R	R	R/W	R/W	TCCR2A
Initial Value	0	0	0	0	0	0	0	0	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Bits 7:6 – COM2A1:0: Compare Match Output A Mode

เป็นบิตสำหรับใช้ในการกำหนดลักษณะของสัญญาณที่เกิดขึ้นที่ขา OC2A เมื่อ Timer/Counter2 เกิด Compare Match

Bits 5:4 – COM2B1:0: Compare Match Output B Mode

เป็นบิตสำหรับใช้ในการกำหนดลักษณะของสัญญาณที่เกิดขึ้นที่ขา OC2B เมื่อ Timer/Counter2 เกิด Compare Match

Bits 3:2-Res: Reserved Bits

ATmega328 จะสงวนบิตในกลุ่มนี้ไว้ จะมีค่าเป็นศูนย์

Bits 1:0 – WGM21:0: Waveform Generation Mode

เป็นบิตสำหรับควบคุมลักษณะ Counting Sequence ของ Counter รวมไปถึงบิต WGM22 ที่อยู่ใน TCCR2B ตามตารางที่ 2.2 ซึ่งในโครงการครั้งนี้ได้มีการเลือกใช้โหมด Fast PWM

ตารางที่ 2.2 Waveform Generation Mode Bit Description

Mode	WGM2	WGM1	WGM0	Timer/Counter Mode of Operation	TOP	Update of OCRx at	TOV Flag Set on ⁽¹⁾⁽²⁾
0	0	0	0	Normal	0xFF	Immediate	MAX
1	0	0	1	PWM, Phase Correct	0xFF	TOP	BOTTOM
2	0	1	0	CTC	OCRA	Immediate	MAX
3	0	1	1	Fast PWM	0xFF	BOTTOM	MAX
4	1	0	0	Reserved	–	–	–
5	1	0	1	PWM, Phase Correct	OCRA	TOP	BOTTOM
6	1	1	0	Reserved	–	–	–
7	1	1	1	Fast PWM	OCRA	BOTTOM	TOP

ตารางที่ 2.3 TCCR2B – Timer/Counter Control Register B

Bit (0xB1)	7	6	5	4	3	2	1	0	TCCR2B
Read/Write	W	W	R	R	R	R	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

Bits 2:0 – CS22:0: Clock Select

เป็นบิตที่ใช้ในการเลือกสัญญาณ Clock ซึ่งมีรายละเอียดดังตารางที่ 2.4

ตารางที่ 2.4 Clock Select Bit Description

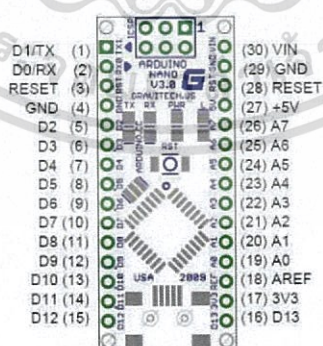
CS22	CS21	CS20	Description
0	0	0	No clock source (Timer/Counter stopped).
0	0	1	clk _{T2S} /(No prescaling)
0	1	0	clk _{T2S} /8 (From prescaler)
0	1	1	clk _{T2S} /32 (From prescaler)
1	0	0	clk _{T2S} /64 (From prescaler)
1	0	1	clk _{T2S} /128 (From prescaler)
1	1	0	clk _{T2S} /256 (From prescaler)
1	1	1	clk _{T2S} /1024 (From prescaler)

2.3.2 คุณสมบัติและการต่อใช้งาน Arduino Nano 3.0

เป็น Arduino ที่ใช้หน่วยประมวลผล ATmega328 เช่นเดียวกับ Arduino Uno R3 ความสามารถจึงเท่าเทียมกัน แต่แตกต่างกันตรงที่ Arduino Nano 3.0 ถูกออกแบบให้มีขนาดเล็ก โดยตัดส่วนของ Socket ที่ไม่จำเป็นออก และยังคงความสามารถในการติดต่อผ่าน USB Port เหมือนบอร์ด Arduino ตัวใหญ่ไว้ อาจจะเรียกได้ว่า Arduino Nano 3.0 ตัวนี้ คือ Arduino Uno ขนาดย่อส่วนลงมา โดยมักใช้เป็นตัว Slave หรือเป็นตัวย่อยที่ใช้ในการควบคุมมอเตอร์แต่ละตัว ก่อนจะนำข้อมูลแต่ละส่วนนี้มาประสานงานกันโดยผ่านตัว Master หรือตัวหลัก



รูปที่ 2.13 บอร์ด Arduino Nano 3.0

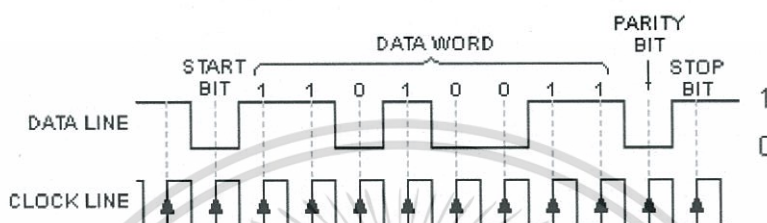


รูปที่ 2.14 ตำแหน่งของพอร์ตการต่อใช้งานของ Arduino Nano 3.0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4 การสื่อสารข้อมูลแบบอนุกรมและโปรโตคอล

การสื่อสารแบบอนุกรม หรือ Serial เป็นส่งข้อมูล โดยใช้เทคนิคการเลื่อนข้อมูล (Shift Bit) ส่งไปที่ละบิตบนสายสัญญาณเส้นเดียว โดยการส่งข้อมูลแบบ Serial จะไม่มีการ Sync สัญญาณนาฬิกา ระหว่างตัวรับและตัวส่ง แต่จะอาศัยวิธีตั้งค่าความเร็วในการรับส่งสัญญาณให้เท่ากัน หรือเรียกว่าการตั้งค่า Baud Rate และส่งสัญญาณ Start และ Stop เพื่อบอกว่าเป็นส่วนต้นของข้อมูล (Start Bit) หรือส่วนท้ายของข้อมูล (Stop Bit) ดังรูปที่ 2.15



รูปที่ 2.15 รูปแบบการส่งสัญญาณสื่อสาร

รูปแบบของ ข้อมูลจากที่ส่งผ่าน Serial จะมีการเพิ่ม Start Bit และ Stop Bit เข้าไปเพิ่มจากข้อมูลเดิม ประกอบด้วย

- บิตเริ่มต้น (Start Bit) จะมีขนาด 1 บิต จะเป็นลอจิก LOW
- บิตข้อมูล (Data Bit) 8 บิต ข้อมูลที่จะส่ง
- บิตภาวะคู่หรือคี่ (Parity Bit) มีขนาด 1 บิต ใช้ตรวจสอบข้อมูล ถ้าข้อมูลที่ได้รับไม่สมบูรณ์ นำเข้าค่ามา Check กับ Parity Bit จะได้ค่าไม่ตรงกัน
- บิตหยุด (Stop Bit) เป็นการระบุถึงขอบเขตของการสิ้นสุดข้อมูล จะเป็นลอจิก HIGH

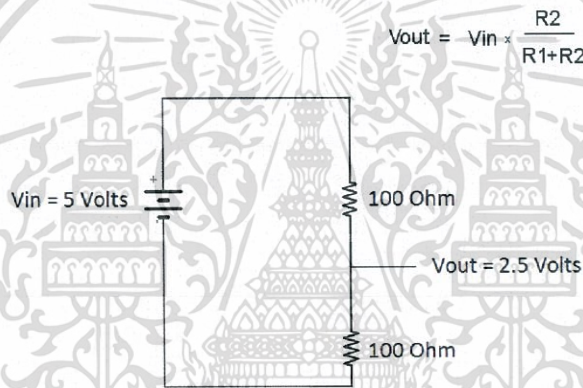
2.5 ตัวต้านทานแบบเปลี่ยนค่าได้ (Variable Resistor)

เป็นตัวต้านทานที่สามารถปรับค่าความต้านทานได้อย่างต่อเนื่องในช่วงค่าความต้านทานที่กำหนดไว้ สำหรับงานที่ต้องการปรับค่าความต้านทานบ่อยๆ โดยการใช้แกนหมุน (แบบวงแหวน) หรือเลื่อนแกน (แบบสไลด์) โครงสร้างภายในทำมาจากคาร์บอนเซรามิค หรือพลาสติกตัวนำ วัสดุที่ใช้ทำตัวความต้านทานชนิดนี้ อาจจะเป็นวัสดุประเภทเดียวกับตัวความต้านทานแบบคงที่ คือ ชนิดคาร์บอน (Carbon) หรือชนิดเส้นลวด (Wire-Wound) ซึ่งแล้วแต่ว่าจะต้องการควบคุมปริมาณของกระแสจำนวนมากน้อยเท่าไร ถ้าใช้กับวงจรที่กระแสสูง วัสดุที่ใช้จะเป็นแบบเส้นลวด ถ้าใช้กับวงจรกระแสต่ำ จะใช้กับวัสดุประเภทคาร์บอน เช่น ในเครื่องรับวิทยุ, โทรทัศน์ เพื่อปรับลดหรือเพิ่มเสียง, ปรับลดหรือเพิ่มแสงในวงจรทีวีไฟ

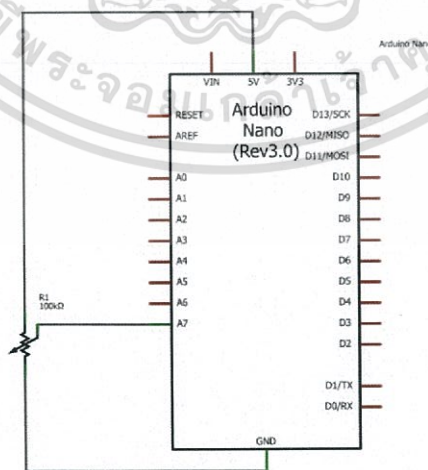
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานของตัวความต้านทานชนิดเปลี่ยนค่าได้ จะมีลักษณะการใช้งานอยู่ 2 ชนิด คือ ชนิดรีโอสตัส (Rheostat) และชนิดโพเทนซิโอมิเตอร์ (Potentiometer)

ตัวต้านทานแบบเปลี่ยนค่าได้นี้ มักใช้ในการปรับค่าสัญญาณอนาล็อกสำหรับส่งค่าให้กับ ไมโครคอนโทรลเลอร์ ซึ่งมีความสามารถรับค่า Analog Input ได้ 5V การอ่านค่าแบบนี้จะแตกต่างจากการอ่านค่าดิจิตอลตรงที่ ดิจิตอลนั้นค่าที่อ่านได้จะมีอยู่ 2 ค่า คือ “1” และ “0” แต่การอ่านค่าอนาล็อกจะแบ่งค่าที่อ่านได้เป็น 1024 ระดับ โดยเทียบตามสัดส่วนของแรงดันอ้างอิง โดยค่าที่อ่านได้เริ่มจาก 0 และเพิ่มขึ้นเรื่อยๆ ไปจนถึง 1023 สำหรับบอร์ด Arduino Nano แรงดันอ้างอิงคือ 5 โวลต์ และสามารถเปลี่ยนแรงดันอ้างอิงเป็นค่าอื่นได้ โดยตัวต้านทานเปลี่ยนค่านี้ส่วนใหญ่จะมี 3 ขา ซึ่งก็คือวงจรแบ่งแรงดันนั่นเอง ซึ่งแสดงดังรูปที่ 2.16 ดังนั้นเมื่อป้อนแรงดันขนาด 5V ให้กับขาทั้งสองของตัวต้านทานเปลี่ยนค่าได้ จะสามารถปรับแรงดันฝั่งขาออกได้ในช่วง 0 – 5V กล่าวคือ ค่าที่อ่าน ADC (Analog-to-Digital Converter) ที่ได้จะมีค่าตั้งแต่ 0-1023 นั่นเอง



รูปที่ 2.16 วงจรแบ่งแรงดัน



รูปที่ 2.17 การต่อใช้งานตัวต้านทานเปลี่ยนค่าได้กับ Arduino Nano 3.0

เรียก Tran (p) ว่าเป็นเมทริกซ์เลื่อนตำแหน่งโฮโมจีเนียสพื้นฐาน (Fundamental Homogeneous Transformation Matrix) จะสังเกตเห็นได้ว่าเวกเตอร์ตำแหน่งใน p ในขณะที่เมทริกซ์การหมุน 3x3 จะเป็นเมทริกซ์เอกลักษณ์ I

สมการแขนกล (The Arm Equation)

เมื่อกำหนดโครงฟัดลงบนแขนได้แล้วทำการย้ายโครงฟัด k ไปยังโครงฟัด k-1 โดยใช้เมทริกซ์ถ่ายโอนโครงฟัดโฮโมจีเนียส เมื่อนำเมทริกซ์ที่ได้มาคูณกันทั้งหมด จะได้เมทริกซ์ที่แสดงการย้ายฟัดของ Tool เทียบกับฟัดฐาน โดยเมทริกซ์ที่ได้นี้เรียกว่า เมทริกซ์แขน (Arm Matrix)

เมทริกซ์แขน (Arm Equation)

การสร้างเมทริกซ์การถ่ายโอนฟัดโฮโมจีเนียสจากโครงฟัด k ไปยังโครงฟัด k-1 จะมี 4 ขั้นตอน แต่ละขั้นตอนจะเกี่ยวข้องกับ Kinematics Parameter 1 ตัว ในการพิจารณาจะทำการหมุนและเลื่อนตำแหน่งโครงฟัด k โดยกระทำตามขั้นตอนดังตารางที่ 2.5

ตารางที่ 2.5 การถ่ายโอนโครงฟัด k ไปยังโครงฟัด k-1

ขั้นตอน (Operation)	คำอธิบาย (Description)
1	หมุน L_{k-1} รอบ z^{k-1} เป็นมุม θ_k
2	เลื่อน L_{k-1} ไปตามแนว z^{k-1} เป็นระยะ d_k
3	เลื่อน L_{k-1} ไปตามแนว x^{k-1} เป็นระยะ a_k
4	หมุน L_{k-1} รอบ x^{k-1} เป็นมุม α_k

การถ่ายโอนโครงฟัด (Link-Coordinate Transformation) ให้ $\{L_0, L_1, \dots, L_n\}$ เป็นเซตของโครงฟัด $[q]^k$ และ $[q]^{k-1}$ เป็นฟัดโฮโมจีเนียสของจุด q เทียบกับโครงฟัด L_k และ L_{k-1} สำหรับ $1 \leq k \leq n$ จะได้ $[q]^{k-1} = T_{k-1}^k [q]^k$ โดยที่ T_{k-1}^k คือ การถ่ายโอนโครงฟัด k ไปยัง k-1

$$T_{k-1}^k = \begin{bmatrix} c\theta_k & -c\alpha_k s\theta_k & s\alpha_k s\theta_k & ac\theta_k \\ s\theta_k & c\alpha_k c\theta_k & -s\alpha_k c\theta_k & as\theta_k \\ 0 & s\alpha_k & c\alpha_k & d_k \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.7)$$

ดังที่กล่าวมาแล้วว่า Kinematics Parameter 3 ตัวที่ปรากฏใน T_{k-1}^k เป็นค่าคงที่ ตัวที่เหลือจะเป็นตัวแปรข้อต่อ ซึ่งอาจเป็นมุม θ_k (สำหรับข้อต่อหมุน) หรือ d_k สำหรับข้อต่อเลื่อน

2. Inverse Kinematics : ความยาวของแต่ละก้านโยงและตำแหน่งเป้าหมายของหุ่นยนต์ ถูกกำหนดให้มา ทำให้สามารถคำนวณมุมแต่ละมุมได้จากตำแหน่งเป้าหมายเหล่านั้นภายในพื้นที่การทำงานของหุ่นยนต์



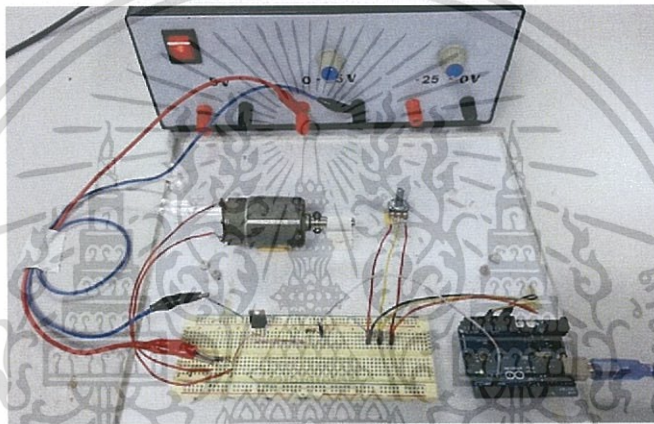
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบวงจร การเขียนโปรแกรม และการสร้างแบบจำลอง

ในการจัดทำโครงงานนี้นอกจากจะอาศัยทฤษฎีข้างต้นดังที่ได้กล่าวมาแล้ว ยังต้องอาศัยวงจรควบคุมการทำงานของเครื่องจักรที่มีประสิทธิภาพ จึงจะส่งผลให้แบบจำลองสามารถทำงานได้ตรงตามเป้าหมาย ซึ่งมีรายละเอียดวงจรที่เกี่ยวข้องกับการควบคุมการทำงานของเครื่องจักรป้อนชิ้นงานอัตโนมัติดังนี้

3.1 วงจรควบคุมความเร็วมอเตอร์พื้นฐาน



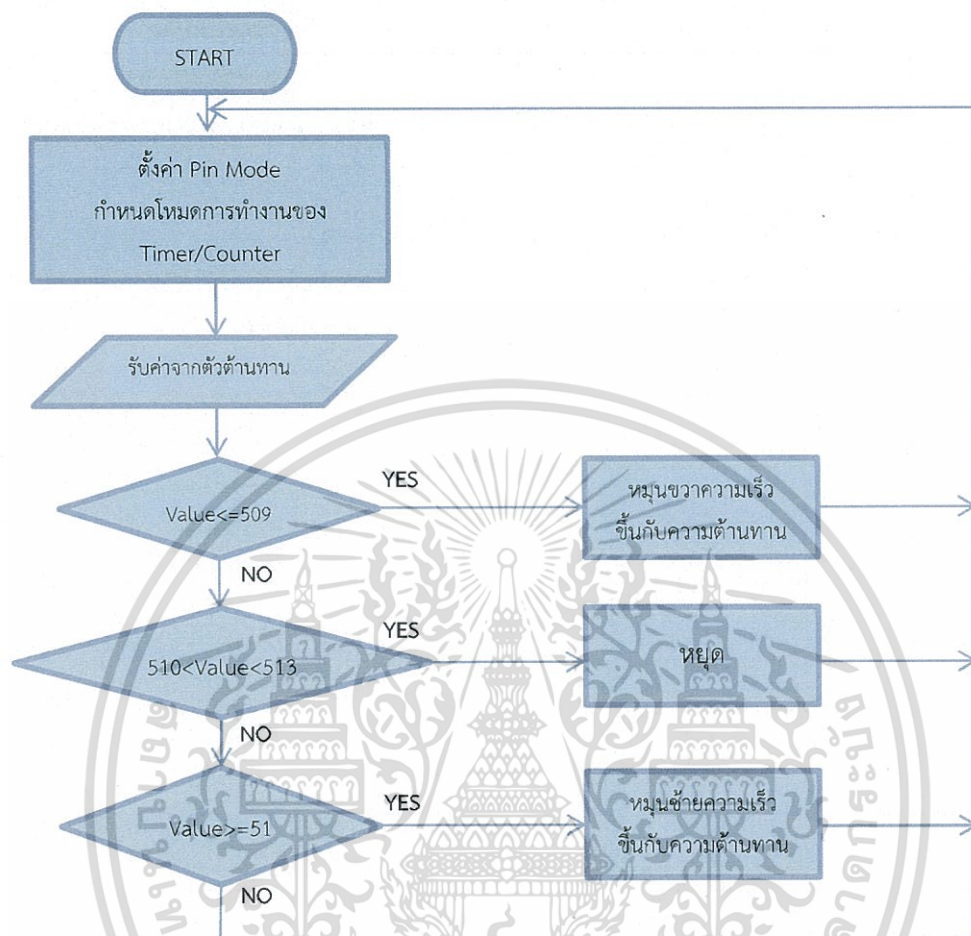
รูปที่ 3.1 การต่อวงจรควบคุมความเร็วมอเตอร์โดยใช้ตัวต้านทานปรับค่าได้อย่างง่าย

ในขั้นตอนแรกของการศึกษาการควบคุมมอเตอร์ จะเริ่มจากการทดลองควบคุมความเร็วมอเตอร์ ซึ่งสามารถต่อวงจรอย่างง่ายได้ดังรูปที่ 3.1 โดยมีตัวต้านทานชนิดปรับค่าได้ เป็นอุปกรณ์หลักที่ใช้ในการควบคุมความเร็วของมอเตอร์ และใช้ Arduino Uno R3 เป็นชุด Microcontroller ที่ใช้ในการเขียนโปรแกรมควบคุม

โปรแกรมควบคุมความเร็วด้วยการใช้ตัวต้านทานแบบปรับค่าได้

ลักษณะของโปรแกรมนั้น ต้องการสัญญาณจะรับค่าความต้านทานจากตัวต้านทานชนิดปรับค่าได้ โดยค่าที่อ่านมาจะเป็นค่าอนาล็อกจากตัวต้านทานปรับค่าได้ ซึ่งจะให้ค่าความละเอียด 10 บิต หรือค่าเท่ากับ 1024 ในระบบฐานสิบ ดังนั้นเมื่อต้องการให้มอเตอร์สามารถหมุนทั้งซ้ายและขวาจึงนำค่าความต้านทานสำหรับการกำหนดตำแหน่งตรงกลาง เมื่อหมุนไปตามเข็มนาฬิกา มอเตอร์ก็จะหมุน

ไปในทิศทางหนึ่งด้วยความเร็วที่เพิ่มขึ้นเรื่อยๆ จนกระทั่งจุดสุดท้าย เมื่อหมุนตัวต้านทานปรับค่าในทิศทางตรงกันข้าม มอเตอร์ก็จะหมุนในทิศทางกลับกันด้วยความเร็วที่เพิ่มขึ้นเช่นกัน



รูปที่ 3.2 ผังงานโปรแกรม (Flow Chart) ของการเขียนโปรแกรมควบคุมความเร็วมอเตอร์ที่ใช้ตัวต้านทานปรับค่าได้

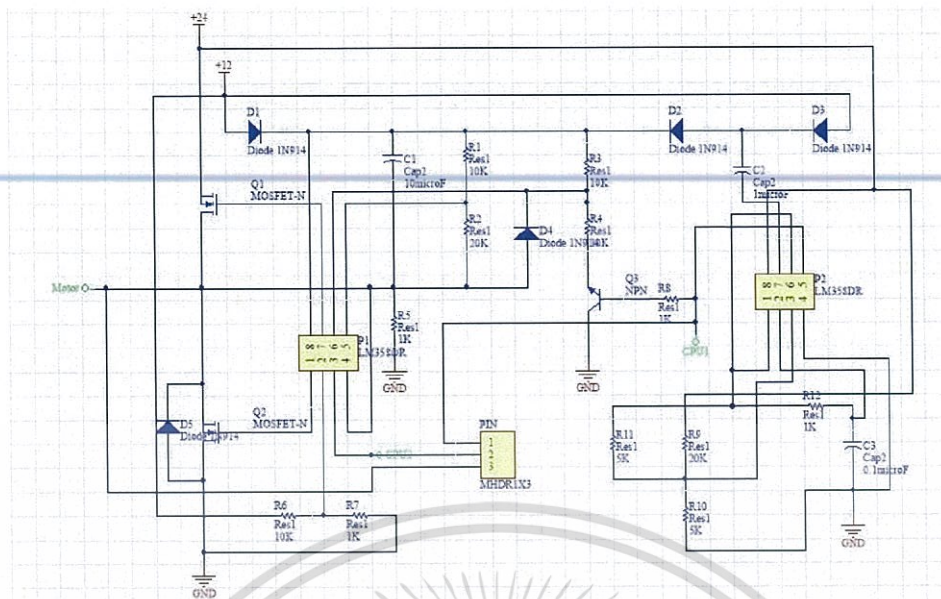
เมื่อศึกษาพื้นฐานของการเขียนโปรแกรมให้กับมอเตอร์ได้แล้ว จึงเริ่มพัฒนาออกแบบวงจรที่ใช้ในการขับมอเตอร์ (Drive Motor Circuit) ดังหัวข้อที่จะกล่าวต่อไป

3.2 การออกแบบวงจรขับมอเตอร์ (Drive Motor Circuit)

3.2.1 การพัฒนางจรขับมอเตอร์

สำหรับวงจรขับมอเตอร์ตามทฤษฎีที่ได้กล่าวมาข้างต้นนั้น ได้เลือกใช้มอสเฟตจำนวน 4 ตัว มาต่อกันแบบวงจร H-Bridge เพื่อใช้ในการควบคุมทิศทางของมอเตอร์ ส่วน Microcontroller ที่ใช้เป็นบอร์ด Arduino Uno R3 ซึ่งมี ATmega328 เป็น MCU หลักในการควบคุม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

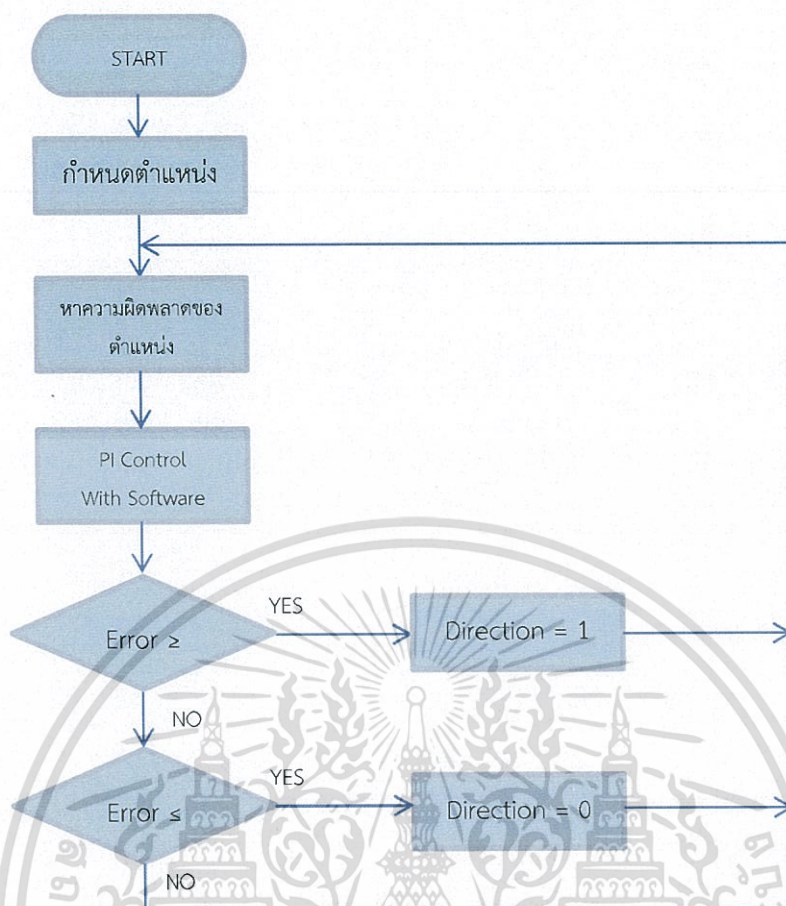


รูปที่ 3.3 การต่อวงจร H-Bridge

จากรูปที่ 3.3 เป็นการออกแบบวงจรจำนวน 1 ข้างของวงจร H-Bridge โดยอุปกรณ์หลักสำหรับวงจรนี้คือ Op-Amp ซึ่งเลือกใช้เบอร์ LM324 จำนวน 2 ตัว และใช้ไฟจาก Power Supply จ่ายมาด้วยขนาด 24V

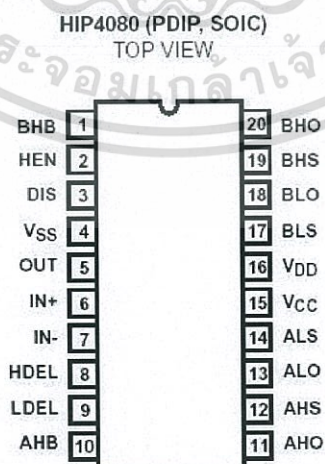
โปรแกรมการควบคุมตำแหน่งแบบ PI โดยคอนโทรลเลอร์

การออกแบบในครั้งแรกนั้นจะมีการใช้คอนโทรลเลอร์ตัวที่เป็น Arduino โดยที่ใช้ตัวต้านทาน R แบบปรับค่าได้ ติดกับปลายของมอเตอร์ เมื่อมอเตอร์หมุนไปที่ตำแหน่งต่างๆ ค่าความต้านทานก็จะมีการเปลี่ยนแปลง จะทำให้เกิดค่าผิดพลาด (Error) ระหว่างตำแหน่งที่กำหนดและตำแหน่งที่ความต้านทานอ่านได้ เพื่อนำค่าผิดพลาด (Error) ที่ได้ไปใช้ในการควบคุมแบบ PI โดยการควบคุมนี้จะควบคุมโดยซอฟต์แวร์ ที่แสดงดังผังงานโปรแกรม (Flow Chart) ในรูปที่ 3.4



รูปที่ 3.4 ผังงานโปรแกรม (Flow Chart) ของการเขียนโปรแกรมควบคุมตำแหน่ง

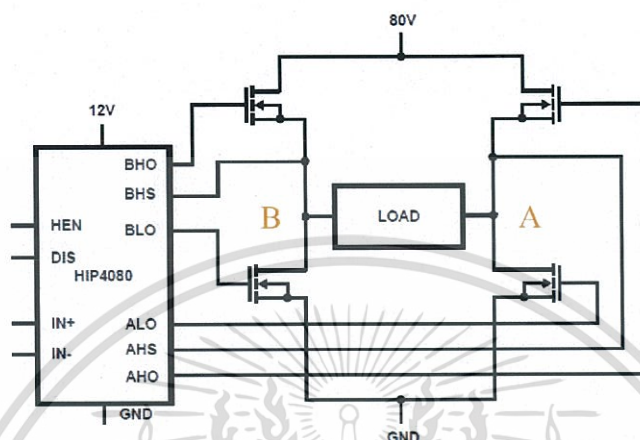
จากการศึกษาเก็บข้อมูลวงจรไฟฟ้าควบคุมทิศทางหมุนของ DC Motor ได้เลือกใช้ ไอซี HIP4080 นำมาใช้ในการควบคุมการเปิด-ปิดมอสเฟต ทำหน้าที่เป็นสวิตซ์ให้กับวงจร H-Bridge



รูปที่ 3.5 ขาของไอซี HIP4080

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.5 แสดงขาของไอซี HIP4080 ซึ่งเป็นไอซีที่สามารถรับสัญญาณ (Requirement) ได้ทั้ง PWM และสัญญาณไคเรกชัน (Direction) สาเหตุที่เลือกใช้ไอซี HIP 4080 เนื่องจากสามารถใช้งานได้ที่มีความถี่สูง และการทำงานของวงจรที่มีความถี่สูงนั้น จะทำให้มอเตอร์หมุนได้เรียบเนียนกว่า อีกทั้งยังมีการดึงกระแสต่ำกว่าการใช้งานที่ความถี่ต่ำ



รูปที่ 3.6 Application Block Diagram ของ HIP4080

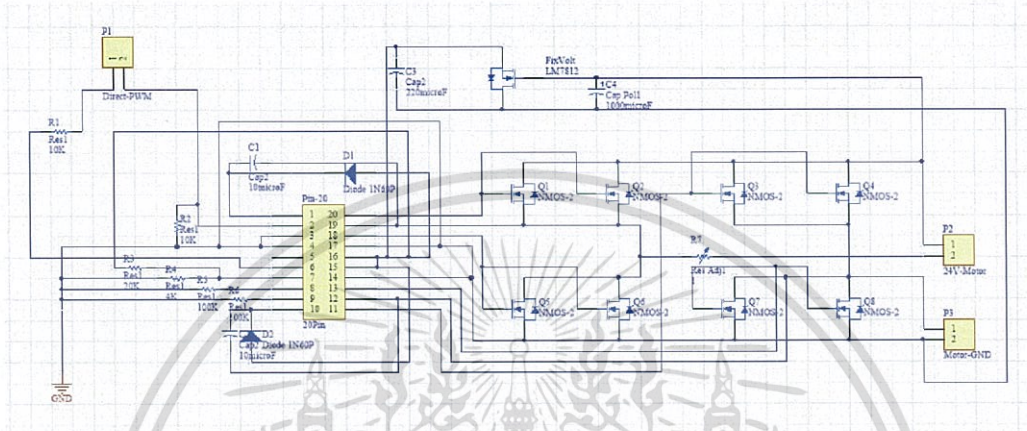
ตารางที่ 3.1 Truth Table ของไอซี HIP 4080

Input		Output				
IN+ > IN-	HEN	DIS	ALO	AHO	BLO	BHO
X	X	1	0	0	0	0
1	1	0	0	1	1	0
0	1	0	1	0	0	1
1	0	0	0	0	1	0
0	0	0	1	0	0	0

การทำงานอย่างง่าย ๆ ของวงจรไดรฟ์ด้วยไอซีชนิดนี้ จะอ้างอิงตามตาราง Truth Table จาก Datasheet ของไอซี HIP4080 ที่ยกมาแสดงดังตารางที่ 3.1 โดยอธิบายได้ว่าจากรูปที่ 3.6 ได้กำหนดให้การควบคุมมอเตอร์ฝั่งขวาเป็น A และการควบคุมมอเตอร์ฝั่งซ้ายเป็น B โดยที่ขา HEN (Enable) คือขาที่รับสัญญาณ PWM มา ดังนั้นถ้าให้ขา Enable เป็น High แล้ว BH กับ AH จะมีผลโดยขึ้นอยู่กับค่า IN ทางด้านอินพุต ซึ่งขา IN นี้กำหนดเป็นสัญญาณลอจิก 0-1 โดยถ้าอินพุต IN+>IN-

จะให้ค่าเท่ากับ 1 และในทางกลับกัน ถ้า $IN+ < IN-$ ลอจิกจะเป็น 0 ดังนั้นถ้าอินพุตเป็น 1 มอสเฟต AH จะ On ส่วนถ้าเป็น 0 มอสเฟต BH จะ On แทน เป็นที่มาของการควบคุมมอเตอร์แบบ H-Bridge นั้นเอง

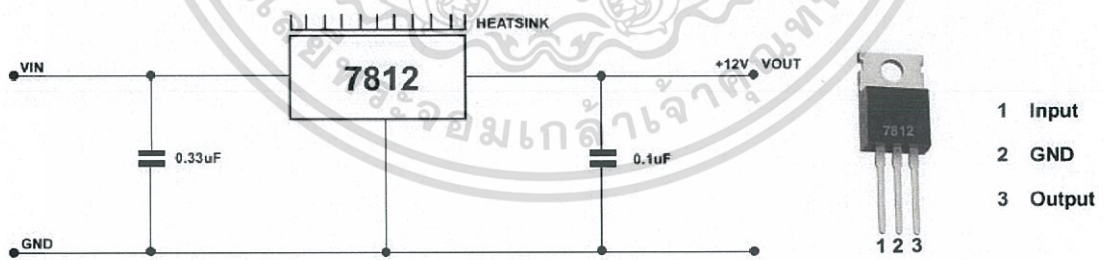
การต่อวงจรลักษณะนี้เรียกว่า Hysteresis Mode Switching หรือวงจรคอนโทรล H-Bridge แสดงดังรูปที่ 3.7



รูปที่ 3.7 การต่อวงจรควบคุมมอเตอร์

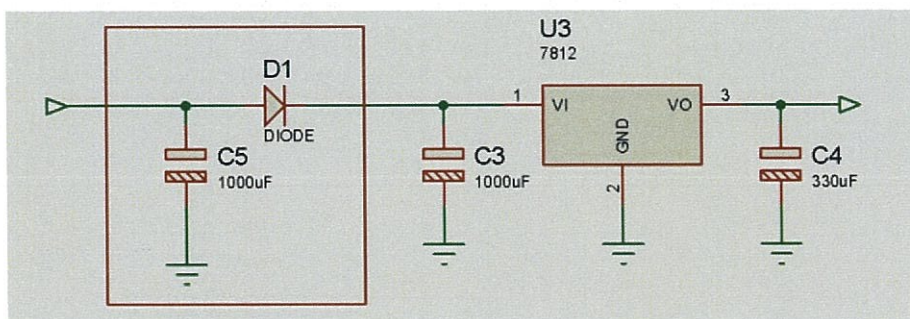
3.2.2 วงจรเรกกูเรเตอร์ (Regulator)

เรกกูเรเตอร์ (Regulator) เป็นอุปกรณ์ที่ใช้ในการแปลงแรงดันเอาต์พุตที่ออกมาจากตัวเรกกูเรเตอร์ให้มีค่าคงที่เสมอ โดย IC ที่เลือกใช้เป็นเบอร์ LM7812 ซึ่งในวงจรนี้ให้แรงดันอินพุตมีค่า 24V จะทำให้แรงดันเอาต์พุตที่ออกมามีค่าเป็น 12V การต่อวงจรแสดงดังรูปที่ 3.8



รูปที่ 3.8 การต่อวงจรเรกกูเรเตอร์ (Regulator) และ LM7812

กรณีมอเตอร์มีการกระชากไฟ โดยเฉพาะขณะที่มีมอเตอร์การหมุนกลับทิศ จะทำให้เกิดกระแสขึ้นสูง เนื่องจากจะทำให้แรงดันไฟที่เลี้ยงตัว HIP4080 เกิดการทรอป ดังนั้นสามารถแก้ได้โดยการต่อไดโอดจากไฟอินพุต 24V ก่อนเข้า LM7812 เพื่อป้องกันไฟไหลย้อนกลับ ดังรูปที่ 3.9



รูปที่ 3.9 การต่อวงจรป้องกันไฟไหลย้อนกลับของวงจรเรกกูเรเตอร์ (Regulator)

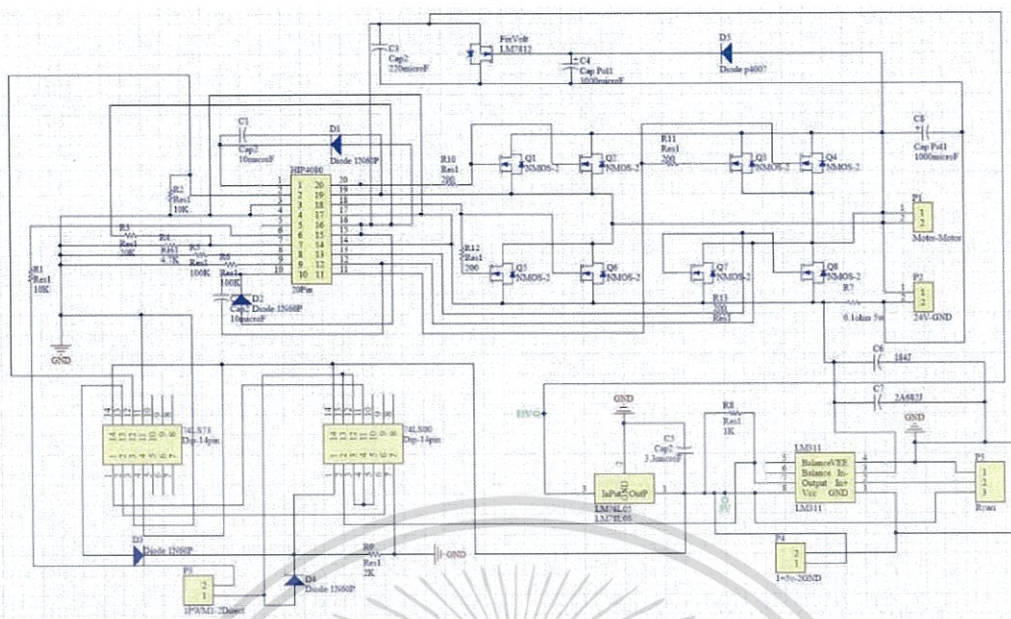
ส่วนในการแปลงสัญญาณแรงดันเอาต์พุตเป็น 5V จะใช้ IC เบอร์ LM78L05 โดยสัญญาณ Input คือ 12V จากวงจรเรกกูเรเตอร์ 12V ในรูปที่ 3.9 ดังกล่าว ดังนั้นการต่อวงจรแสดงดังรูปที่ 3.10



รูปที่ 3.10 การต่อวงจรเรกกูเรเตอร์ของ LM78L05

3.2.3 วงจรจำกัดกระแส (Limit Current)

ต่อมาได้ทำการออกแบบวงจร Limit Current หรือวงจรจำกัดกระแส เพื่อใช้ในการควบคุมสัญญาณ Pulse Width ในกรณีที่กระแสอาจกระชากขึ้นสูง วงจรนี้จะ Off ทันทันทีที่มีสัญญาณกระแสเกิน เพื่อป้องกันวงจรหลักเสียหาย โดยหลักการคือจะใช้สัญญาณ PWM เป็นสัญญาณ Clock ซึ่งได้เลือกใช้ไอซี J-K Flip-Flops 74LS73 เป็นตัวควบคุม และใช้ไอซี 74LS00 (NAND Gate) เป็นตัวหน่วงเวลาสัญญาณ ก่อนเข้า J-K Flip-Flops ดังนั้นการออกแบบวงจรที่เสร็จสมบูรณ์ แสดงดังรูปที่ 3.11



รูปที่ 3.11 วงจรขับมอเตอร์โดยสมบูรณ์



รูปที่ 3.12 กัดปรินลายวงจร และบอร์ดอิเล็กทรอนิกส์

เมื่อออกแบบลายวงจรเสร็จแล้ว จึงทำการกัดปรินลายวงจรจำนวน 4 บอร์ด ใช้ในการคุมมอเตอร์จำนวน 4 ตัว และนำมาประกอบกัน ดังรูปที่ 3.13

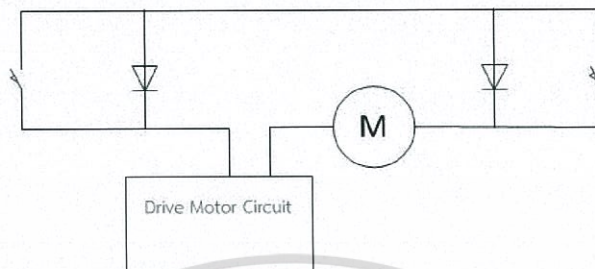


รูปที่ 3.13 บอร์ดวงจรขับมอเตอร์ (Drive Motor Circuit) ทั้ง 4 ตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.4 วงจรลิมิตสวิตช์

นอกจากนี้ ยังมีส่วนของวงจรลิมิตสวิตช์ที่จะนำมาใช้ในกรณีที่โปรแกรมอาจทำงานผิดพลาด เพื่อป้องกันตัวเครื่องเสียหาย หรือเรียกว่าเป็น Safety อย่างหนึ่ง โดยลิมิตสวิตช์จะทำการตัดการทำงานของมอเตอร์ทันทีที่หน้าสัมผัสถูกสัมผัส ซึ่งการต่อวงจรลิมิตสวิตช์แสดงดังรูปที่ 3.14

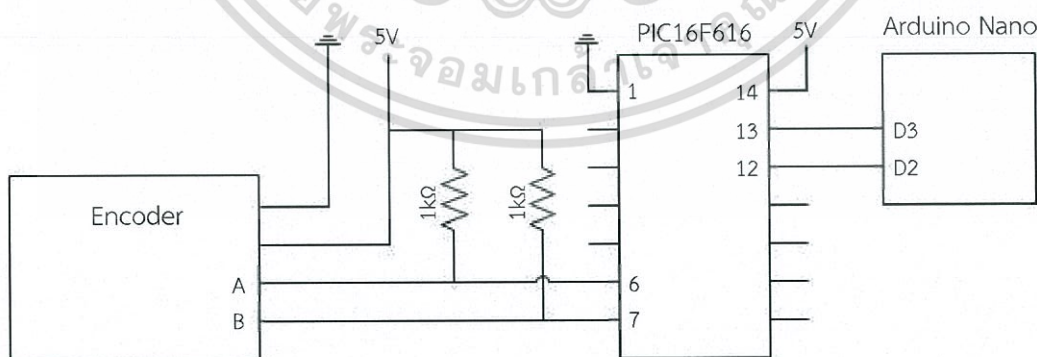


รูปที่ 3.14 การต่อวงจรลิมิตสวิตช์

ลิมิตสวิตช์นี้จะถูกติดตั้งอยู่ 2 ตำแหน่งคือ ตำแหน่งมอเตอร์เคลื่อนที่ขึ้น/ลง และตำแหน่งมอเตอร์ควบคุมการหมุนของส่วนแขน

3.2.5 วงจรเอนโค้ดเดอร์ (Encoder)

ในการควบคุมตำแหน่งและทิศทางของล้อ จะใช้ Encoder เป็นตัวรับส่งสัญญาณการเคลื่อนที่ ในการต่อวงจร จะใช้สัญญาณจากเอนโค้ดเดอร์ 2 สัญญาณคือ A และ B ต่อเข้ากับไอซี PIC 16F616 ซึ่งได้เขียนโค้ดโปรแกรมไว้สำหรับการอ่านสัญญาณทั้งสองเทียบกัน แล้วส่งข้อมูลให้ไมโครคอนโทรลเลอร์ โดยการต่อวงจรเอนโค้ดเดอร์แสดงไว้ดังรูปที่ 3.15 ส่วนการวางตำแหน่งเอนโค้ดเดอร์ จะต้องติดตั้งให้หมุนสัมพันธ์กับการเคลื่อนที่ของล้อ



รูปที่ 3.15 การต่อวงจรเอนโค้ดเดอร์

3.3 หลักการสร้างแบบจำลองของเครื่องจักรป้อนชิ้นงานอัตโนมัติ

ในการออกแบบสร้างแบบจำลองเครื่องจักรป้อนชิ้นงานอัตโนมัติ มีความจำเป็นต้องเข้าใจลักษณะการเคลื่อนที่ของตัวเครื่องจักรก่อน โดยเครื่องจักรนี้ออกแบบมาเพื่อให้ใช้กับเครื่องปั๊มแผ่นเหล็ก ซึ่งตัวเครื่องจะทำหน้าที่แทนมนุษย์ โดยการหยิบชิ้นงานเคลื่อนที่เข้าไปวางในแท่นปั๊ม แล้วจับชิ้นงานที่ถูกปั๊มแล้ว ส่งต่อไปยังเครื่องปั๊มอื่นๆ เพื่อปั๊มในส่วนรายละเอียดอื่นๆ ต่อไป ดังแสดงในรูปที่ 3.16 ดังนั้นผู้ออกแบบจึงได้สร้างเครื่องจักรตัวต้นแบบเป็นชนิดเคลื่อนที่สี่แกน มีการเคลื่อนที่ขึ้น-ลงในแนวตั้ง และมีข้อต่อที่สามารถหมุนได้ 180 องศา ส่วนปลายมือจับจะใช้แม่เหล็กไฟฟ้าเป็นตัวจับชิ้นงานซึ่งสามารถหมุนได้รอบ เพื่อใช้ในการพลิกวัตถุให้ตรงตามรูปแบบที่เครื่องปั๊มต้องการ นอกจากนี้ตัวเครื่องจักรต้องสามารถเคลื่อนที่ไปในแนวราบได้เพื่อส่งต่อชิ้นงานให้กับเครื่องปั๊มที่อยู่เครื่องถัดไป จึงได้ออกแบบให้ตัวแบบจำลองนี้วิ่งไปตามรางที่ได้จัดทำขึ้นเพื่อนำไปประยุกต์ใช้ต่อไป



รูปที่ 3.16 ภาพจำลองการใช้งานของเครื่องจักรป้อนชิ้นงานอัตโนมัติกับเครื่องจักรในโรงงาน

3.3.1 วัสดุที่ใช้ในการสร้างแบบจำลองเครื่องจักรป้อนชิ้นงานอัตโนมัติ

วัสดุหลักที่ใช้ในการสร้างแบบจำลองเครื่องจักรป้อนชิ้นงานอัตโนมัติประกอบด้วย



รูปที่ 3.17 วัสดุอุปกรณ์ในการสร้างแบบจำลองแขนกล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

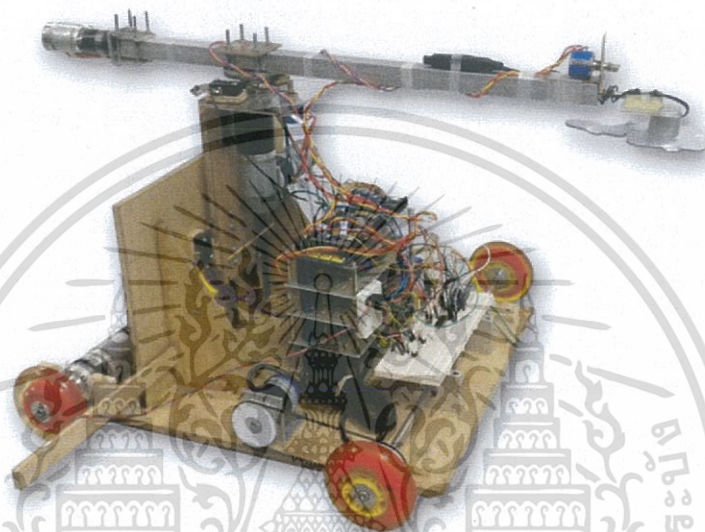
1. แผ่นไม้สำหรับทำฐานขนาด 33cm x 42cm x 0.8cm
2. แผ่นไม้สำหรับเป็นฉากติดตั้งมอเตอร์แนวตั้ง ขนาด 21cm x 30cm x 0.8cm
3. แผ่นไม้ฉากรองรับน้ำหนักรูปสามเหลี่ยม จำนวน 2 แผ่น
4. ท่อกลางอลูมิเนียมสี่เหลี่ยมจัตุรัส ขนาดยาว 47cm หน้าตัด 2x2cm
5. เหล็กเส้นกลม เส้นผ่าศูนย์กลาง 6mm
6. ล้อยาง 4 ล้อ
7. แบริ่งสำหรับเชื่อมชุดขับเคลื่อน จำนวน 2 ชิ้น
8. มอเตอร์ไฟฟ้า 24 VDC 300 rpm จำนวน 2 ตัว
9. มอเตอร์ไฟฟ้า 24 VDC 150 rpm จำนวน 1 ตัว
10. มอเตอร์ไฟฟ้า 24 VDC 60 rpm จำนวน 1 ตัว
11. มอเตอร์ไฟฟ้า 24 VDC 15 rpm จำนวน 1 ตัว
12. แม่เหล็กไฟฟ้า ขนาด 12 VDC
13. Rotary Encoder 1 ตัว
14. ตัวต้านทานเปลี่ยนค่าได้แบบหมุน 25 รอบ จำนวน 2 ตัว
15. ตัวต้านทานเปลี่ยนค่าได้แบบสไลด์ 1 ตัว
16. ลิimitswitch จำนวน 4 ตัว
17. รางเลื่อน (Linear Guide Rail) ขนาด 20cm
18. คัปปลิ่งมอเตอร์ ขนาดเส้นผ่าศูนย์กลาง 6mm และ 8mm อย่างละ 1 ตัว
19. ยางโอริง 1 เส้น ขนาดเส้นผ่าศูนย์กลาง 11cm
20. นี้อตและสกรู
21. เชือก, เทปใยลึบประรด
22. กาวร้อน, กาวอีพ็อกซี่
23. สายไฟแข็ง, สายไฟฝอย

3.3.2 ขั้นตอนการประกอบและติดตั้ง

จากวัสดุดังกล่าว นำมาประกอบติดตั้ง โดยเริ่มจากวัดแผ่นไม้ตามขนาดที่ได้ออกแบบไว้ และประกอบแผ่นไม้ทั้งหมดด้วยกาวอีพ็อกซี่เสริมด้วยใยแก้วเพิ่มความแข็งแรง เนื่องจากต้องแบกรับน้ำหนักของมอเตอร์ โครงสร้างจึงควรมีความแข็งแรงคงทน ต่อมาทำการประกอบรางเลื่อนโดยติดตั้งในแนวตั้ง ประกอบติดด้วยแผ่นไม้ จากนั้นประกอบมอเตอร์ทั้งหมดตามตำแหน่งที่กำหนดไว้

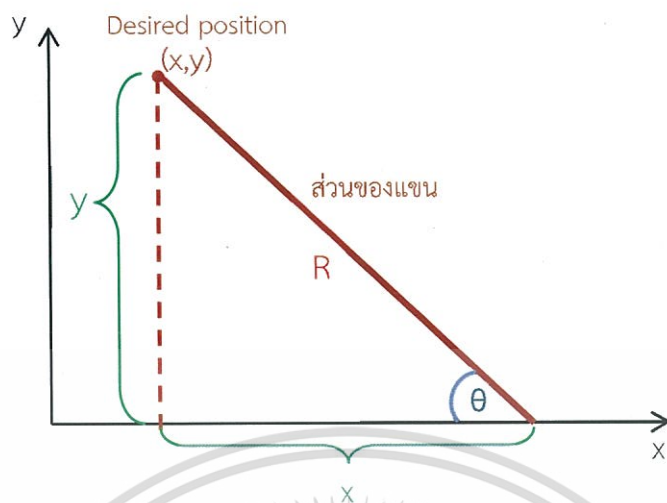
เมื่อติดตั้งส่วนของแขนด้วยน็อตและสกรูแล้ว จึงทำการติดตั้งเอนโค้ดเดอร์โดยพ่วงกับล้อด้วยยางโอริง ส่วนตัวต้านทานปรับค่าจะติดกับขั้วมอเตอร์โดยใช้เชือกพันรั้งไว้ให้เคลื่อนที่ไปโดยสัมพันธ์กัน และที่ส่วนปลายของแขนหุ่นจะทำการติดตั้งแม่เหล็กไฟฟ้ายึดด้วยคัปปลิง

ลำดับสุดท้าย จะทำการติดตั้งลิมิตสวิทช์ 2 ตำแหน่งของหุ่น คือ บริเวณมอเตอร์เคลื่อนที่ขึ้น/ลง และบริเวณมอเตอร์ควบคุมการหมุนของแขน โดยติดตั้งตำแหน่งละ 2 ตัว เพื่อป้องกันการเคลื่อนที่ทั้ง 2 ด้าน และเมื่อประกอบติดตั้งเสร็จเรียบร้อยแล้ว จะได้แบบจำลองโดยสมบูรณ์แสดงดังรูปที่ 3.18



รูปที่ 3.18 แบบจำลองเครื่องจักรป้อนชิ้นงานอัตโนมัติ

3.4 Inverse Kinematic



รูปที่ 3.19 Inverse Kinematic

กำหนดให้

 θ

เป็นมุมของแขน

 X'

ตำแหน่งของตัวเครื่องหรือจุดศูนย์กลางการหมุน

 (x, y)

คอนโทรลเลอร์จะได้รับจากตัว Master

 R

ความยาวของแขน

หามุม θ ;

จาก

$$\sin \theta = \frac{y}{R} \quad (3.1)$$

$$\theta = \sin^{-1} \left(\frac{y}{R} \right) \quad (3.2)$$

หาระยะ X ;

$$X'' = R(\cos(\theta)) \quad ; \quad \theta = \sin^{-1} \left(\frac{y}{R} \right) \quad (3.3)$$

หาระยะ X' ;

$$X' + X'' = X \quad (3.4)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เพราะฉะนั้น ตำแหน่งของตัวเครื่องหรือจุดศูนย์กลางการหมุน คือ

$$X' = X + R \left[\cos \left(\sin^{-1} \left(\frac{Y}{R} \right) \right) \right] \quad (3.5)$$

เนื่องจากการทำงานของมอเตอร์แต่ละตัว คอนโทรลเลอร์จะทำงานแยกกันแต่จะได้รับข้อมูล ตำแหน่งชุดเดียวกันจึงต้องคำนวณค่า Desired Position ของตัวเอง

การคำนวณ Desired Position ของตำแหน่งตัวเครื่อง

$$DP = \frac{\text{ความละเอียดของตำแหน่งเอนโค้ดเดอร์ต่อรอบ}}{\text{ระยะที่เคลื่อนที่ได้ต่อรอบการหมุน}} \times X' \quad (3.6)$$

$$DP = \frac{1440}{800\pi} X' \quad (3.7)$$

โดยที่

X'

DP

ความละเอียดของตำแหน่งเอนโค้ดเดอร์ต่อรอบ

$$\text{ความละเอียดของตำแหน่งต่อรอบ} = PPR \times 4 \quad (3.8)$$

โดยที่

PPR

จำนวนพัลส์ของเอนโค้ดเดอร์ต่อรอบ

เอนโค้ดเดอร์ที่มีการใช้งานมีความละเอียดที่ 360 PPR (พัลส์ต่อรอบ) แต่เลือกการใช้งานที่ความละเอียด 4X นั้นหมายความว่า การหมุนหนึ่งรอบจะสามารถให้ความละเอียดของตำแหน่งเป็น 4 เท่าของความละเอียดเอนโค้ดเดอร์ คือ 1440 ตำแหน่งต่อการหมุน 1 รอบ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระยะทางที่สามารถเคลื่อนที่ได้ต่อรอบการหมุน

$$L = \pi D \quad (3.9)$$

โดยที่

- L ระยะทางที่สามารถเคลื่อนที่ได้ต่อรอบการหมุน
D เส้นผ่าศูนย์กลางล้อ

การคำนวณ Desired Angle ของแขน

$$R = \left(\frac{R_{\max} - R_{\min}}{\theta_{\max} - \theta_{\min}} \right) (\theta - \theta_{\min}) + R_{\min} \quad (3.10)$$

โดยที่

- R_{\max} สัญญาณ ADC มากที่สุด
 R_{\min} สัญญาณ ADC น้อยที่สุด
R ค่ามุมสำหรับคำนวณ PI Control
 θ_{\max} ค่ามุมมากที่สุด
 θ_{\min} ค่ามุมน้อยที่สุด
 θ ค่ามุมที่คำนวณได้จากสมการที่ (3.2)

เนื่องจากการบิดของแขนจะอยู่ในช่วง $0 \leq \theta \leq \pi$ ดังนั้น $\theta_{\min} = 0$, $\theta_{\max} = \frac{\pi}{2}$ ในส่วน
ของ R_{\max} , R_{\min} จะสามารถปรับตั้งได้จากการปรับเทียบ

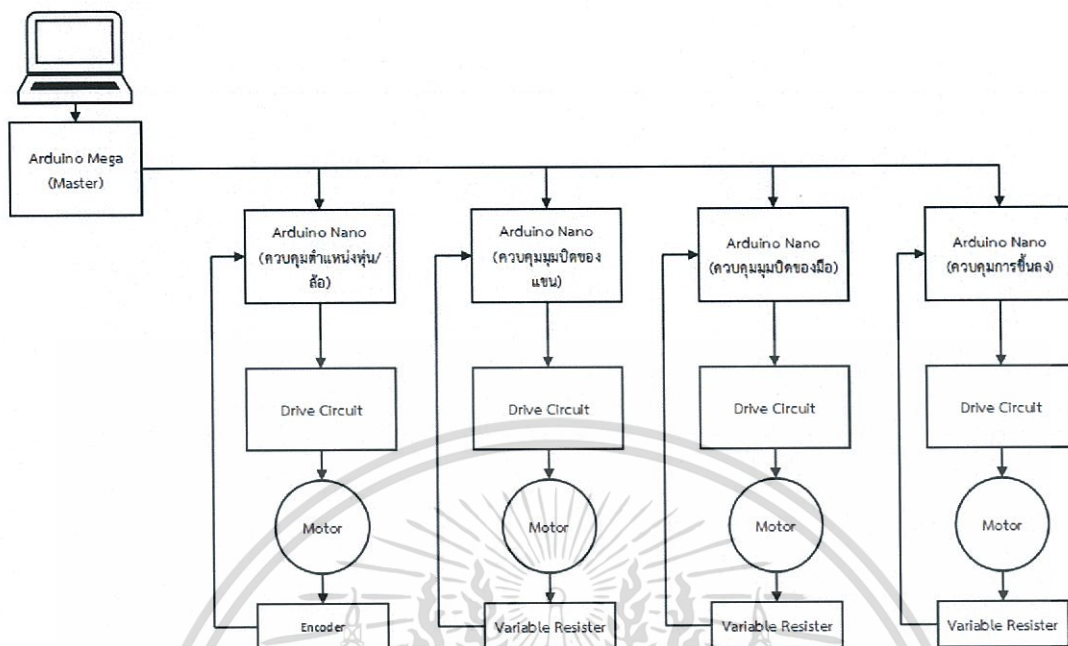
จากการปรับเทียบได้

$$R_{\max} = 510, R_{\min} = 470$$

ดังนั้น

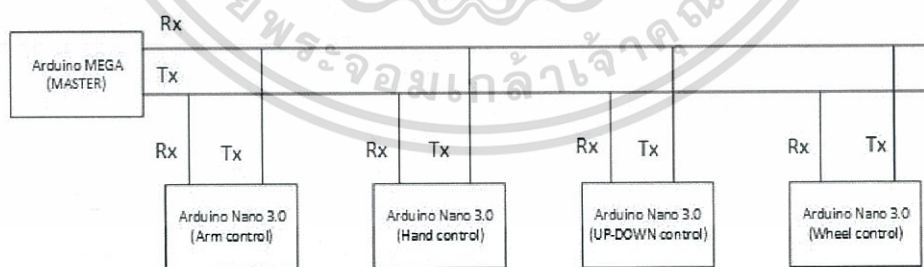
$$R = \left(\frac{510 - 470}{\pi/2} \right) (\theta) + 470 \quad (3.11)$$

3.5 การเขียนโปรแกรมควบคุมและการสื่อสารข้อมูล



รูปที่ 3.20 ระบบที่ใช้ในการควบคุม

การควบคุมการทำงาน จะใช้โครงข่ายของคอนโทรลเลอร์ทำงานร่วมกัน สื่อสารกันในรูปแบบของการสื่อสารอนุกรม (Serial Communication) โดยที่ตัวมาสเตอร์จะส่งตำแหน่งโคออร์ดิเนต (X,Y) ของเป้าหมาย (ตำแหน่งปลายแขน) ให้กับคอนโทรลเลอร์ทุกตัวที่ควบคุมมอเตอร์ คอนโทรลเลอร์แต่ละตัวจะทำการคำนวณ Inverse Kinematic ตำแหน่งของตัวเอง และมีการควบคุมการป้อนกลับแบบ PI ทำให้ตำแหน่งมีความแม่นยำมากขึ้น

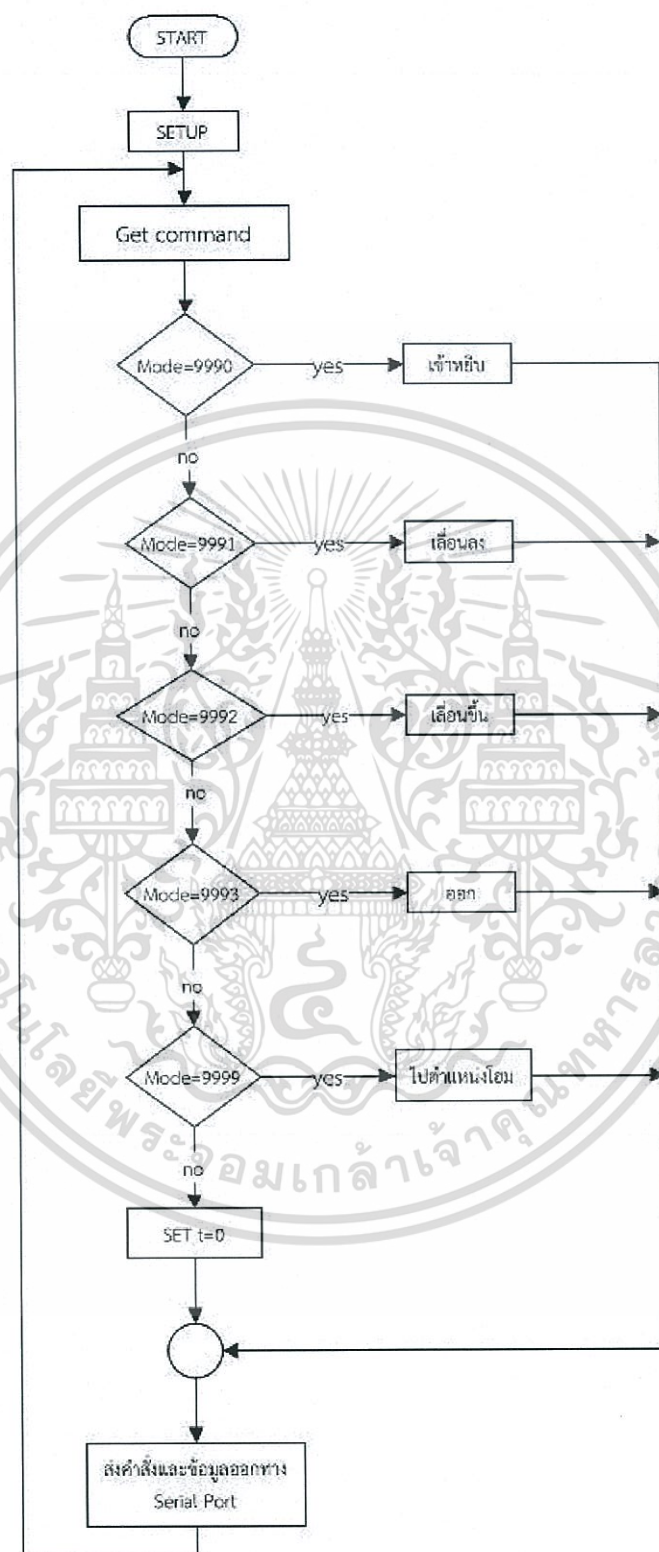


รูปที่ 3.21 การสื่อสารแบบอนุกรม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเขียนโปรแกรม

Flow Chart แสดงการทำงานของ Controller ตัว Master



รูปที่ 3.22 Flow Chart แสดงการทำงานของ Controller ตัว Master

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอินเทอร์รัพท์ (Interrupt) เวลา

```
ISR(__vector_13)
{
  TCNT1=64908;
  t++;
  T2=1;
}
```

ฟังก์ชันย่อยสำหรับการเข้าหยิบ

```
void armCW(){
  currentX=0;
  currentY=0;
  t=0;
  while(currentY<=armLong){
    if(T2==1){
      currentY=V*t*0.01;
      if(currentY>=armLong){
        currentY=armLong;
        intX=8888;
        return;
      }
      sendData();
    }
    T2=0;
  }
```

เมื่อมีการเรียกใช้ฟังก์ชันการเข้าหยิบ ตำแหน่งเริ่มต้นของปลายแขนคือ ตำแหน่งที่มีค่าพิกัด (X,Y) ของปลายมือ คือ (0,0) จากนั้นจะมีการเพิ่มตำแหน่งของ Y ขึ้นเรื่อยๆ ด้วยความเร็วคงที่ V ซึ่งจะมีการคำนวณและส่งข้อมูลทุกๆ 10 ms

ฟังก์ชันย่อยสำหรับการออก

```

void armCCW(){
    t=0;
    currentX=0;
    currentY=armLong;
    while(currentY>=0){
        if(T2==1){
            currentY=armLong-((V*t)*0.01);
            if(currentY<=0){
                currentY=0;
                intX=8888;
                return;
            }
            sendData();
        }
        T2=0;
    }
}

```

เมื่อมีการเรียกใช้ฟังก์ชันการออกจากตำแหน่งเริ่มต้นปลายของปลายแขน คือ ตำแหน่งมีค่าพิกัดสุดท้ายที่แขนทำมุม 90 องศา ซึ่งมีตำแหน่ง (X,Y) ของปลายมือ คือ (0,400) จากนั้นจะมีการลดตำแหน่งของ Y ลงด้วยความเร็วคงที่ V ซึ่งจะมีการคำนวณและส่งข้อมูลทุกๆ 10 ms รูปแบบของข้อมูลที่ Master ส่งให้กับ Slave

ดังที่กล่าวมาแล้วข้างต้น ข้อมูลจากมาสเตอร์ที่จะส่งให้กับคอนโทรลเลอร์ตัวอื่นๆ ผ่านพอร์ตสื่อสารอนุกรมนั้นจะต้องมีการจัดรูปแบบโดยข้อมูลตัวแรกเป็นค่าตำแหน่ง X ข้อมูลตัวที่สองจะเป็นค่าตำแหน่ง Y ข้อมูลตัวที่สามเป็นมุมการบิดของมอเตอร์ และข้อมูลตัวสุดท้าย เป็นระยะขึ้นลงตามแนวแกน Z

โดยมีรูปแบบดังนี้ #AxxxBxxxCxxxDxxx*

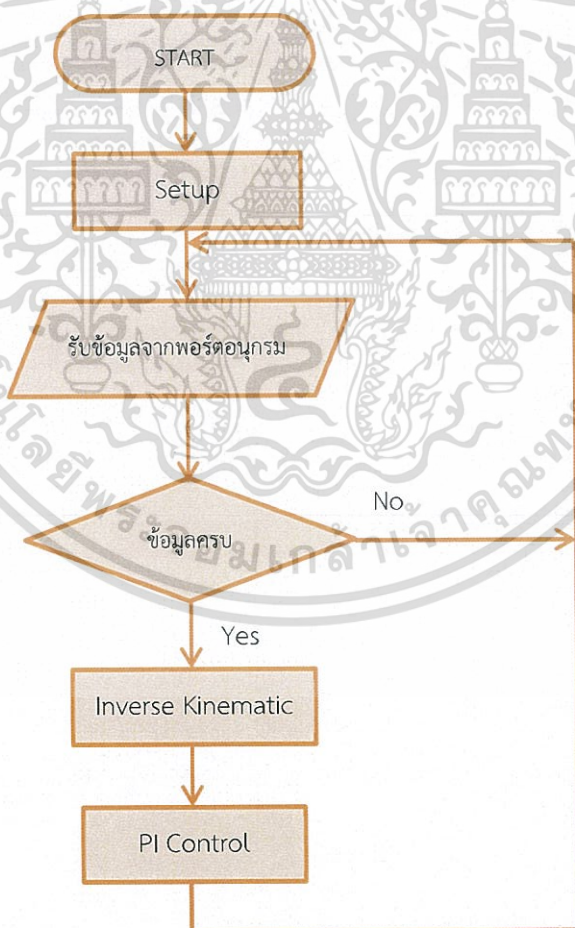
1. # แสดงการเริ่มต้นส่งข้อมูล
2. AxxxBxxxCxxxDxxx ส่วนของข้อมูล

3. *จุดสิ้นสุดข้อมูล

```

void sendData(){
  String string = String("#");
  string += String("A");
  string += String(currentX);
  string += String("B");
  string += String(currentY);
  string += String("C");
  string += String(C);
  string += String("D");
  string += String("4");
  string += String("*");
  Serial.println(string);
}

```



รูปที่ 3.23 Flow Chart แสดงการทำงานของ Controller ตัว Slave

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

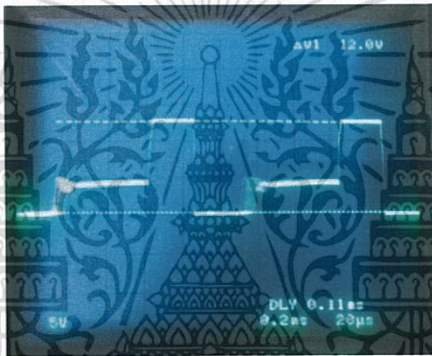
การทดลองและผลการทดลอง

4.1 การทดสอบวงจรขับเคลื่อนมอเตอร์ด้วยการใช้ Arduino

4.1.1 การทดสอบวงจรควบคุมความเร็วมอเตอร์โดยใช้ตัวต้านทานปรับค่าได้

จากวงจรในรูปที่ 3.1 เป็นการต่อวงจรควบคุมความเร็วของมอเตอร์พื้นฐาน แล้วทำการป้อนแรงดันไฟฟ้าขนาด 12V จากแหล่งจ่ายไฟ (Power Supply) และเขียนโปรแกรมควบคุมลงไปที่บอร์ด Arduino UNO R3 โดยให้ Duty Cycle ของสัญญาณ PWM ขึ้นกับแรงดันที่ตกคร่อมตัวต้านทานปรับค่าได้

ผลการทดสอบ



รูปที่ 4.1 กราฟผลการทดสอบวงจรควบคุมความเร็วมอเตอร์โดยใช้ตัวต้านทานปรับค่าได้

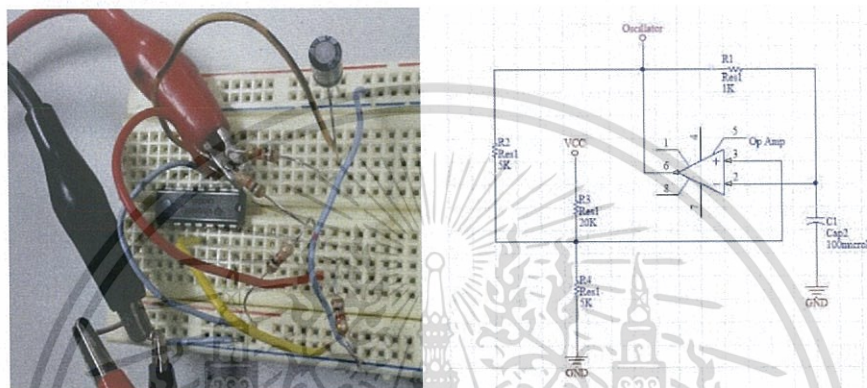
เมื่อหมุนค่าความต้านทานปรับค่าได้ ซึ่งถูกเขียนโปรแกรมไว้ให้อ่านค่าอนาล็อกจากตัวต้านทานปรับค่า มอเตอร์จะเริ่มค่อยๆ หมุนช้าๆ หยุดและหมุนขวาตามค่าที่กำหนดไว้ แต่จากรูปที่ 4.1 จะพบว่าในช่วงนี้เมื่อวัดแรงดันที่ตกคร่อมมอเตอร์พบว่ามี Back EMF เนื่องจากไดโอดไม่ได้ทำงานแต่มอเตอร์ยังหมุนอยู่

4.1.2 การทดลองควบคุมความเร็ว

ทำการปรับเปลี่ยนวงจร โดยใช้วงจรขับเคลื่อนมอเตอร์แบบ H-Bridge (รูปที่ 3.3) และเริ่มทดสอบจากการต่อมอเตอร์เพียงข้างเดียวของวงจร H-Bridge ก่อน ซึ่งใช้มอเตอร์จำนวน 2 ตัว ได้แก่ ตัวบนและตัวล่าง จ่ายไฟจากแหล่งจ่ายขนาด 12V

ผลการทดสอบ

ขณะที่มอเตอร์หมุนในช่วงที่ ON ยาวกว่าช่วง OFF นั้นมีปัญหาแรงดันที่ตัวเก็บประจุเกิดการดรอปลง ทำให้มอเตอร์ OFF ไม่สนิท ส่งผลให้แรงดันไฟที่เข้า Op-Amp ไม่คงที่ มอเตอร์จึงมีการกระตุกเป็นจังหวะ และเกิดกระแสสูงขึ้น วิธีการแก้ไขสำหรับปัญหานี้คือ ทำการสร้างวงจรความถี่ (Oscillator) เพื่อชาร์จประจุป้อนเข้าไปยังตัวเก็บประจุในช่วงที่เกิดการดรอป และใช้วงจรชmittริกเกอร์ (Schmitt Trigger) ในการเปรียบเทียบสัญญาณของ PWM และวงจรความถี่นี้ ผ่าน Op-Amp เบอร์ LM324 อีกตัวหนึ่ง ดังนั้นผลการแก้ไขนี้จะแสดงดังรูปที่ 4.2



รูปที่ 4.2 การต่อวงจรชmittริกเกอร์ (Schmitt Trigger)

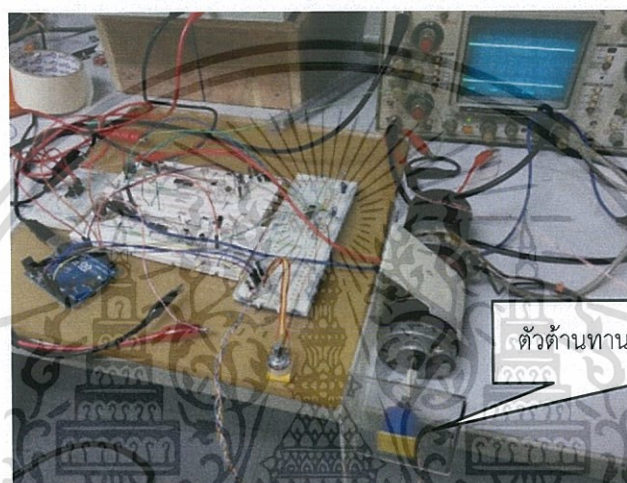


รูปที่ 4.3 สัญญาณกระแสไฟที่ไหลในวงจร

จากรูปที่ 4.3 จะเห็นว่า สัญญาณความถี่ที่ถูกสร้างขึ้นมา จะสัมพันธ์กับการชาร์จและการคายประจุของตัวเก็บประจุอย่างต่อเนื่อง โดยประจุจะถูกชาร์จขึ้นเรื่อยๆ จนไปแตะกับกราฟความถี่ช่วงขาลงก็จะคายประจุออก แล้วเริ่มชาร์จใหม่อีกครั้งตลอดช่วงเวลาที่มอเตอร์ ON เพื่อไม่ให้เกิดการดรอป

4.1.3 การทดลองควบคุมตำแหน่ง

การทดลองในขั้นตอนนี้ ใช้ตัวต้านทานในการส่งสัญญาณป้อนกลับตำแหน่งให้กับคอนโทรลเลอร์ โดยจะมีการติดตั้งตัวต้านทานตรงเพลาของมอเตอร์ ในส่วนของวงจรขับมอเตอร์หรือ H-Bridge นั้นจะมีการใช้วงจรเดียวกับวงจรควบคุมความเร็ว การทำงานของวงจรเมื่อมีการหมุนตัวต้านทานปรับค่า หรือ R_control (อ้างอิงจากโปรแกรม) จะทำให้เกิดค่าความผิดพลาด เป็นเหตุให้คอนโทรลเลอร์สร้างสัญญาณพัลส์เพื่อให้อัตราการหมุน เพื่อให้ค่าของ R_control และ R_motor มีค่าเท่ากันหรือค่าความผิดพลาดเป็นศูนย์นั่นเอง



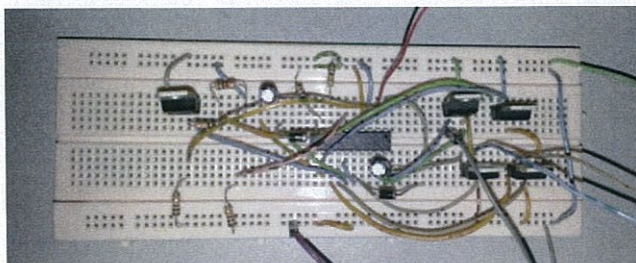
รูปที่ 4.4 การทดลองควบคุมตำแหน่งด้วยการใช้ Arduino

ผลการทดสอบ

เมื่อหมุนตัวต้านทานปรับค่าได้ พบว่ามอเตอร์จะหมุนไปตามตำแหน่งที่อ่านค่าได้จากตัวต้านทานปรับค่าซึ่งถูกกำหนดไว้ในโปรแกรม และเมื่อลองเอามือบิดเพลาของมอเตอร์ เพลาของมอเตอร์จะต้านมือเนื่องจากเป็นผลของการควบคุมแบบ PI ที่ป้อนกลับมาจากตัวต้านทานที่ติดตั้งบริเวณปลายของเพลาของมอเตอร์ และเมื่อปรับค่า K_e ในโปรแกรม จะส่งผลเกี่ยวกับการลื่นไถลตำแหน่งของมอเตอร์ โดยจากการทดลองพบว่า ค่า $K_e = 25$ เหมาะสมที่สุด

4.1.4 การทดลองใช้วงจรกับการปรับความถี่

จากรูปที่ 4.5 เป็นการต่อวงจร H-Bridge โดยใช้ HIP4080 เป็นตัวขับหลักในวงจร และใช้ Function Generator เป็นอุปกรณ์จ่ายสัญญาณ PWM ให้กับวงจร โดยใช้จ่ายไฟจากแหล่งจ่ายขนาด 12V

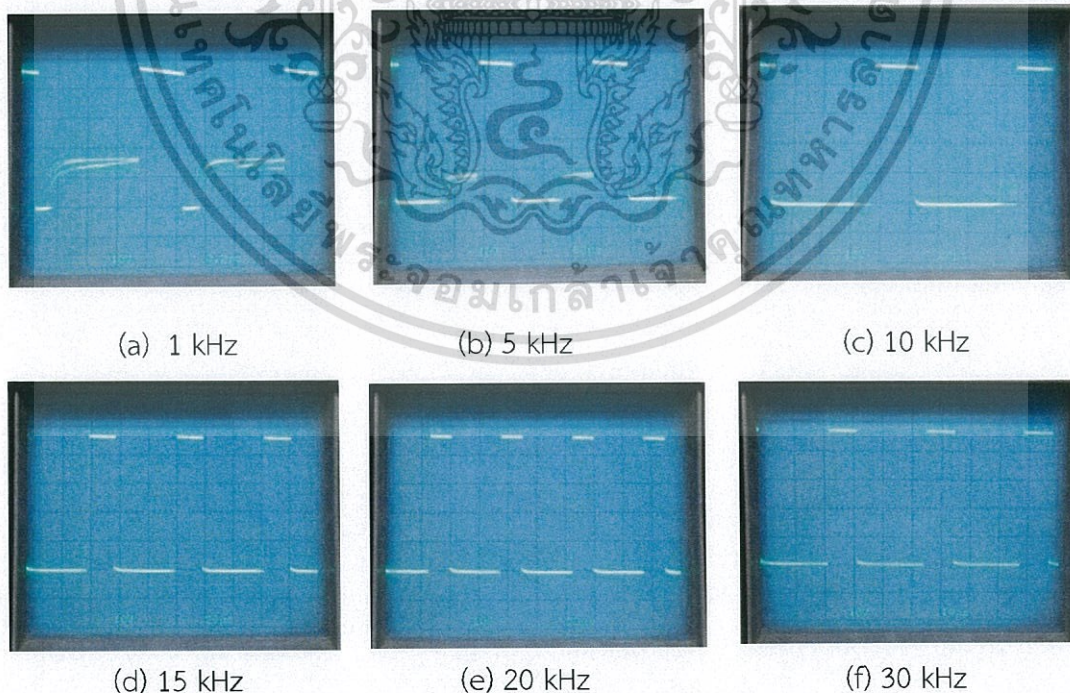


รูปที่ 4.5 การต่อวงจร H-Bridge โดยใช้ HIP 4080

ผลการทดสอบ

เมื่อใช้วงจรนี้กับมอเตอร์ตัวเล็ก (กำลังน้อย) พบว่าไม่มีปัญหาใด สามารถหมุนได้ตามปกติ แต่เมื่อเปลี่ยนมอเตอร์ขนาดใหญ่ขึ้น (กำลังมาก) พบว่าช่วงที่มีการเปลี่ยนลอจิก 0-1 ขณะส่งสัญญาณลอจิกเข้าขาที่ 6 ของ HIP4080 นั้น มอเตอร์มีการดิ่งกระแสมาก เนื่องจากวงจรมอเตอร์นี้ไม่สามารถเบรกได้ และอีกเหตุผลคือ เพราะไม่ได้ต่อชุด Limit Current ทำให้ไม่มีการ Disable ก่อนการเปลี่ยนทิศทางหมุน จึงเกิดการดิ่งกระแสมาก

ต่อมาทำการเพิ่มขนาดไฟจากแหล่งจ่ายไปเป็น 24V และเพิ่มขนาดของตัวเก็บประจุเพื่อลดสัญญาณรบกวนจากการที่ไฟดรอป แล้วทำการปรับค่าความถี่ที่จ่ายมาจาก Function Generator ตั้งแต่ 1kHz – 30 kHz ได้ผลดังรูปที่ 4.6



รูปที่ 4.6 กราฟแสดงผลของการปรับความถี่ที่ค่าต่างๆ ตั้งแต่ 1kHz – 30kHz

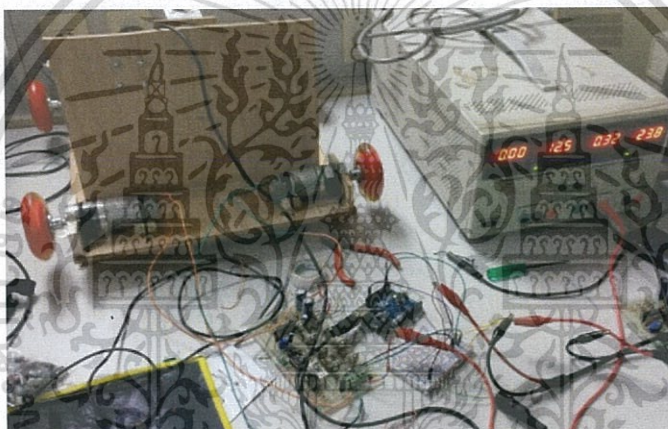
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.6(a) จะเห็นว่ากราฟค่อนข้างสั้น ไม่เรียบ มีการรบกวนสูง ในขณะที่ความถี่สูง เช่น รูปที่ 4.6(e) หรือ 4.6(f) นั้นจะเห็นว่า การใช้งานที่ความถี่สูงนั้น ทำให้มีสัญญาณรบกวนน้อยลง กราฟเรียบสวยขึ้น และมอเตอร์มีการหมุนที่เรียบขึ้น และยังทำให้กระแสกระชากลดลงด้วย เนื่องจาก ช่วง ON สั้น แต่ถี่บ่อยขึ้น

แต่เนื่องจากการใช้งานที่ความถี่สูงนั้น เกิดปัญหาแรงดันไฟฟ้าเกิดการดรอปลงที่ตัวเก็บประจุ ดังนั้นจึงทำการเปลี่ยนชนิดของตัวเก็บประจุเป็นชนิดแทนทาลัม (Tantalum) เพราะเหมาะกับวงจรความถี่สูง เนื่องจากมีความต้านทานน้อย

4.1.5 การทดสอบการทำงานของบอร์ดวงจรขับมอเตอร์

การทดสอบนี้ จะต่อวงจรตามรูปที่ 3.7 เพื่อดูผลการทำงานของมอเตอร์โดยจะต่อเข้ากับมอเตอร์เพื่อดูว่าวงจร H-Bridge นี้สามารถทำการเปิด/ปิดสวิตซ์ให้สั่งงานมอเตอร์ได้ถูกต้องหรือไม่



รูปที่ 4.7 การต่อวงจรทดสอบการทำงานของบอร์ด

ผลการทดสอบ

การทดลองนี้ได้เขียนโปรแกรมโดยใช้ Arduino UNO R3 เป็นไมโครคอนโทรลเลอร์ และเมื่อจ่ายไฟ 24V พบว่ามอเตอร์สามารถหมุนได้ปกติ ต่อมาได้ทำการเปลี่ยนสัญญาณ Direction โดยการส่งสัญญาณจาก 0 เป็น 1 คือจากตอนแรกที่ต่อลงกราวด์ เปลี่ยนเป็นจ่ายสัญญาณไฟให้กับ Direction ผลคือทำให้มอเตอร์เคลื่อนที่กลับทิศทางและหมุนต่อไปได้ปกติ

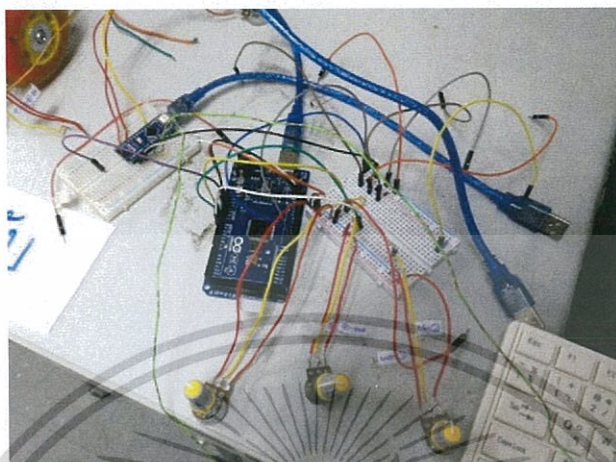
4.2 การทดสอบการทำงานของโปรแกรม

4.2.1 ทดสอบการทำงานของโปรแกรมโดยใช้ตัวต้านทานปรับค่าได้

การทดลองนี้ทำเพื่อทดสอบโปรแกรมส่วนของการส่งข้อมูล ว่าสามารถรับค่าแรงดันตกคร่อมตัวต้านทานปรับค่าได้ แล้วสามารถส่งข้อมูลสั่งให้มอเตอร์เคลื่อนที่ตามที่ผู้ทดลองหมุนปรับได้หรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยมอเตอร์แต่ละตัว จะถูกควบคุมโดย Arduino Nano เป็นตัว Slave และมีตัวต้านทานปรับค่า 1 ตัวต่อข้อมูล 1 ชุดที่ต่อเข้ากับตัวมาสเตอร์ สำหรับเป็นข้อมูลอ้างอิงที่จะส่งให้ Slave ซึ่งในที่นี้ได้เลือกใช้ Arduino ATmega2560 เป็นมาสเตอร์สำหรับรวบรวมส่งข้อมูล



รูปที่ 4.8 การวงจรต่อตัวต้านทานปรับค่าได้

ผลการทดสอบ

เมื่อหมุนตัวต้านทานปรับค่าได้ ตัวมาสเตอร์จะอ่านค่า ADC จากตัวต้านทานปรับค่าแล้ว จัดรูปแบบข้อมูล ส่งข้อมูลให้กับคอนโทรลเลอร์ตัว Slave ที่มีการเขียนโปรแกรมควบคุมแบบ PI ไว้ แล้วนั้น ก็จะทำการคำนวณ Error สำหรับการทำให้ Close Loop Control ต่อไป ซึ่งจากการทดลอง ข้อมูลที่มาสเตอร์ส่งและ Slave รับ เมื่อตั้งค่า Baud Rate ที่ 9600 ข้อมูลจะมีความผิดพลาดอยู่บ้าง จึงแก้ไขด้วยการเพิ่ม Baud Rate เป็น 57600 ข้อมูลที่ได้มีความถูกต้อง

4.2.2 ทดสอบการทำงานของเอนโค้ดเดอร์

การทดลองนี้จะทดสอบการเคลื่อนที่ของมอเตอร์ส่วนล้อ จากการอ่านสัญญาณเอนโค้ดเดอร์ โดยเขียนโปรแกรมสั่งให้หุ่นวิ่งในแนวแกน X แล้ววัดค่าที่ได้เทียบกับค่าจริง โดยต้องการให้หุ่นวิ่งไปที่ ตำแหน่ง 40cm ซึ่งการทดลองมีการเซตค่าโฮม (Home) ของตำแหน่งไว้ ค่าการนับเอนโค้ดเดอร์ที่ 2293 การวิ่งของตัวหุ่นไปหาเป้าหมายจะเป็นการนับแบบลดจนถึงเป้าหมาย 40cm เอนโค้ดเดอร์จะอ่านค่าตำแหน่งได้เท่ากับศูนย์

ผลการทดสอบ

เมื่อสั่งงานให้หุ่นทำงานพบว่าตัวหุ่นวิ่งไปยังตำแหน่งเป้าหมายที่ต้องการ โดยทดสอบการทำงานจำนวน 20 ครั้ง แสดงผลได้ดังตารางที่ 4.1

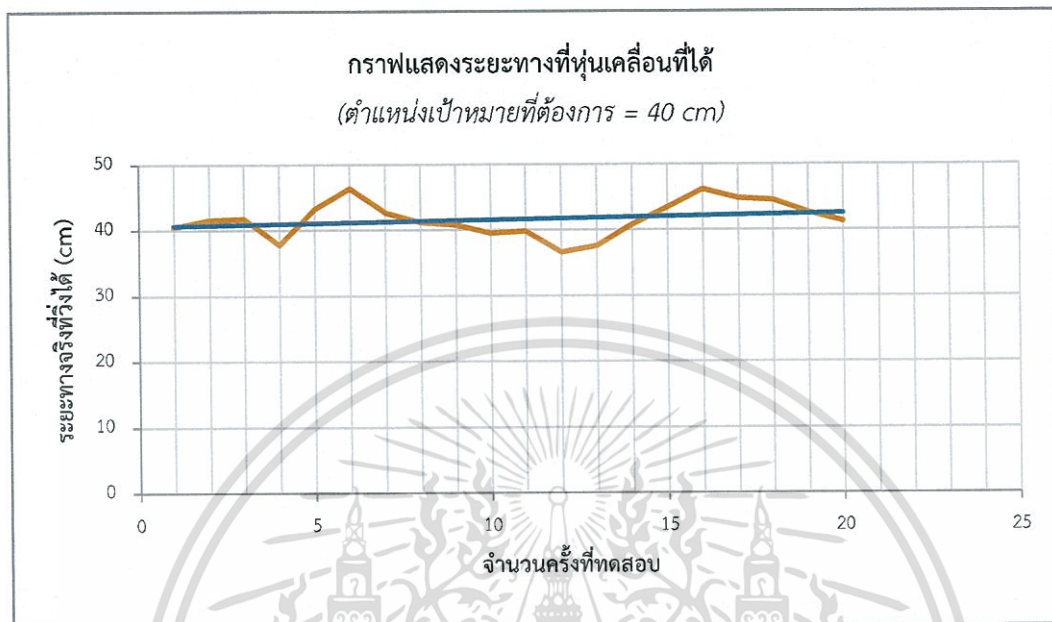
ตารางที่ 4.1 ผลการทดสอบให้เครื่องหยิบจับชิ้นงานวิ่งไปยังตำแหน่งเป้าหมาย

ทดสอบ ครั้งที่	วัตรยะการเคลื่อนที่จากซอฟต์แวร์		วัตรยะการเคลื่อนที่ของหุ่น (cm)	
	ตำแหน่งเป้าหมาย	ค่าที่อ่านได้จาก เอ็นโค้ดเดอร์	ตำแหน่งเป้าหมาย ที่ต้องการ	ตำแหน่งจริงที่วิ่งไป ได้
1	0	14	40	40.5
2	0	16	40	41.6
3	0	9	40	41.8
4	0	13	40	37.8
5	0	12	40	43.2
6	0	6	40	46.4
7	0	12	40	42.7
8	0	12	40	41.2
9	0	14	40	40.8
10	0	8	40	39.5
11	0	12	40	39.8
12	0	13	40	36.6
13	0	12	40	37.6
14	0	16	40	40.8
15	0	13	40	43.4
16	0	12	40	46.2
17	0	14	40	44.8
18	0	10	40	44.5
19	0	12	40	42.7
20	0	15	40	41.3

จากผลการทดสอบ คำนวณหาค่าเฉลี่ย (Average) ของตำแหน่งที่หุ่นวิ่งไปได้เท่ากับ 41.66 cm และหาค่าส่วนเบี่ยงเบนมาตรฐาน (S.D.) = 2.683948388

จากตารางที่ 4.1 พบว่าหุ่นเคลื่อนที่เข้าใกล้ตำแหน่งที่ต้องการ คือที่ระยะ 40 cm โดยมีความคลาดเคลื่อนเล็กน้อย เนื่องจากการเซ็ทโฮมของหุ่นไม่ได้กลับมาที่ตำแหน่งเดิมทุกครั้ง เพราะล้อมีการ

สลิปขณะถึงตำแหน่งเป้าหมาย อีกทั้งโปรแกรมที่ใช้ควบคุมการเคลื่อนที่เป็นการเขียนแบบความเร็วคงที่ ทำให้ไม่มีการลดความเร็วในขณะที่วิ่งเข้าใกล้เป้าหมาย ทำให้มอเตอร์มีการกระชากวิ่งไป/กลับ



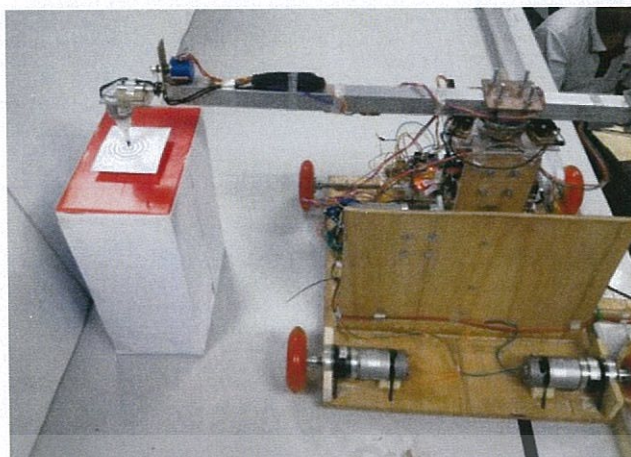
รูปที่ 4.9 กราฟแสดงระยะทางที่หุ่นเคลื่อนที่ได้

จากรูปที่ 4.9 แสดงระยะทางที่หุ่นเคลื่อนที่ได้ไปหาเป้าหมาย จะเห็นว่าเส้นสีส้มคือเส้นที่แสดงระยะทางที่หุ่นเคลื่อนที่ได้ในแต่ละครั้งที่ทดสอบ โดยค่าจะคลาดเคลื่อนจากค่า 40cm ไปทั้งทางบวกและทางลบเล็กน้อย และเส้นสีฟ้าแสดง Trend Line ของผลลัพธ์

4.2.3 ทดสอบความแม่นยำในการเคลื่อนที่เข้าหาเป้าหมาย

ในการทดสอบนี้ จะเขียนโปรแกรมให้ตัวหุ่นทำงานอัตโนมัติโดยกำหนดตำแหน่งปลาย (X,Y) ที่ต้องการให้ปลายมือเคลื่อนไปเป็น (0,400) ส่วนตำแหน่งเริ่มต้นของปลายมือเป็น (0,0) และมีการตั้งค่า Home ให้กับหุ่นเพื่อเป็นการรีเซ็ตจุดเริ่มต้นให้กับหุ่นเพื่อเคลื่อนกลับมายังตำแหน่งเดิมก่อนการเริ่มทำงานครั้งต่อไป

ในส่วนของโปรแกรม ได้กำหนดรหัสสำหรับส่งค่าเคลื่อนที่ออกจากตำแหน่ง (0,0) ไปยัง (0,400) และรหัสสำหรับส่งค่าเคลื่อนที่กลับมายังตำแหน่ง Home โดยที่ตำแหน่งโฮม (Home) ก็คือตำแหน่งที่ปลายมืออยู่ที่จุด (0,0) นั่นเอง ส่วนที่ปลายมือได้ติดตั้งอุปกรณ์ปลายแหลมที่ช่วยเป็นตัวชี้วัดค่าที่ปลายได้เพื่อสะดวกต่อการบันทึกค่า



รูปที่ 4.10 การทดสอบความแม่นยำ

ผลการทดสอบ

จากตำแหน่งที่ต้องการ จะพบว่าการเปลี่ยนแปลงเฉพาะค่าในแกน Y คือจากระยะ 0 ไป 400mm จึงทำให้ค่าความคลาดเคลื่อนที่เกิดขึ้นนั้น เกิดในค่าของ X โดยเกิดทั้งค่าบวกและค่าลบซึ่งแสดงดังตารางที่ 4.2

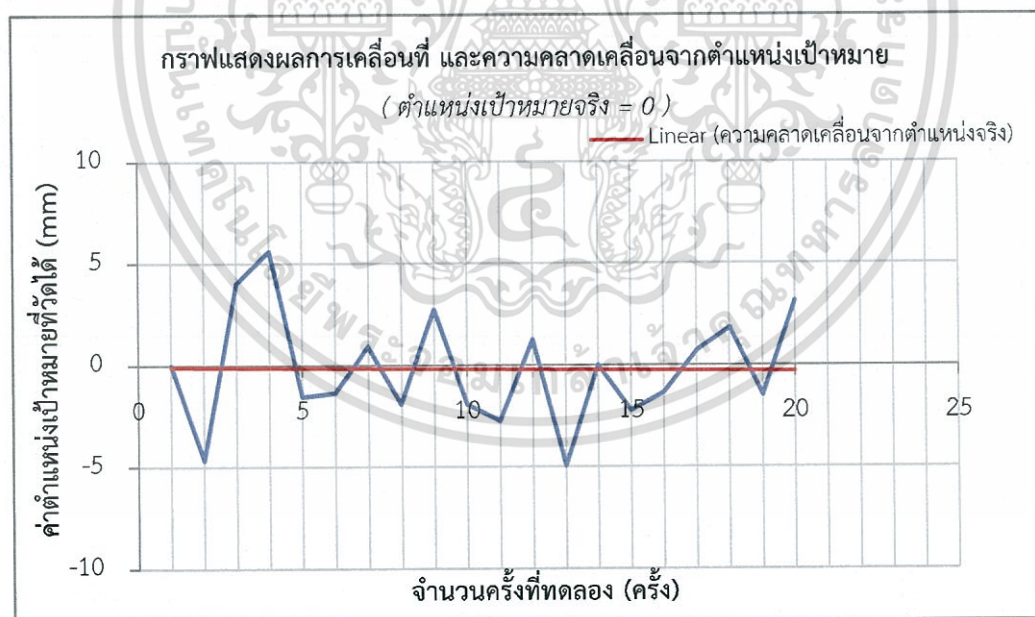
ตารางที่ 4.2 ผลการทดสอบความแม่นยำในการเคลื่อนที่ไปยังตำแหน่งเป้าหมาย

ครั้งที่	ค่าสัญญาณ ADC จากตัว ต้านทานปรับค่า		ตำแหน่ง เริ่มต้น (mm)	ตำแหน่ง เป้าหมายจริง (mm)	ตำแหน่ง เป้าหมายที่ วัดได้ (mm)
	สัญญาณที่ส่ง	สัญญาณที่ อ่านได้			
1	508	507	(0,0)	(0,400)	(0.00,400)
2	508	507	(0,0)	(0,400)	(-4.66,400)
3	508	507	(0,0)	(0,400)	(4.03,400)
4	508	507	(0,0)	(0,400)	(5.60,400)
5	508	507	(0,0)	(0,400)	(-1.54,400)
6	508	507	(0,0)	(0,400)	(-1.36,400)
7	508	507	(0,0)	(0,400)	(0.93,400)
8	508	507	(0,0)	(0,400)	(-1.94,400)
9	508	507	(0,0)	(0,400)	(2.74,400)
10	508	507	(0,0)	(0,400)	(-1.94,400)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 ผลการทดสอบความแม่นยำในการเคลื่อนที่ไปยังตำแหน่งเป้าหมาย (ต่อ)

ครั้งที่	ค่าสัญญาณ ADC จากตัว ต้านทานปรับค่า		ตำแหน่ง เริ่มต้น (mm)	ตำแหน่ง เป้าหมายจริง (mm)	ตำแหน่ง เป้าหมายที่ วัดได้ (mm)
	สัญญาณที่ส่ง	สัญญาณที่ อ่านได้			
11	508	507	(0,0)	(0,400)	(-2.76,400)
12	508	507	(0,0)	(0,400)	(1.25,400)
13	508	507	(0,0)	(0,400)	(-4.97,400)
14	508	507	(0,0)	(0,400)	(0.00,400)
15	508	507	(0,0)	(0,400)	(-2.26,400)
16	508	507	(0,0)	(0,400)	(-1.35,400)
17	508	507	(0,0)	(0,400)	(0.73,400)
18	508	507	(0,0)	(0,400)	(1.82,400)
19	508	507	(0,0)	(0,400)	(-1.5,400)
20	508	507	(0,0)	(0,400)	(3.15,400)



รูปที่ 4.11 กราฟแสดงผลการเคลื่อนที่และความคลาดเคลื่อนจากตำแหน่งเป้าหมาย

จากผลการทดสอบ คำนวณหาค่าเฉลี่ย (Average) ของตำแหน่งที่คลาดเคลื่อนเฉพาะค่า X ได้เท่ากับ -0.2015 mm และหาค่าส่วนเบี่ยงเบนมาตรฐาน (S.D.) = 2.766855093

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปและวิจารณ์ผลการทดลอง

5.1 สรุปผลการทดลอง

จากการทดสอบบอร์ดขับเคลื่อนมอเตอร์ที่ได้ทำการออกแบบ พบว่าสามารถขับเคลื่อนมอเตอร์ที่มีความถี่สูงถึง 30 KHz ที่แรงดัน 24VDC ใช้สำหรับขับเคลื่อนของเครื่องจักรป้อนชิ้นงานอัตโนมัติ โดยใช้บอร์ด Arduino nano 3.0 ในการควบคุมและการป้อนกลับตำแหน่งของมอเตอร์ทดสอบ โดยให้เครื่องป้อนชิ้นงานวิ่งจากตำแหน่งโฮม (Home) ไปยังตำแหน่งหยิบชิ้นงาน ผลปรากฏว่าในการวิ่งเข้าหาตำแหน่งเป้าหมายทดสอบที่ระยะ (0,400) mm โดยทดสอบจำนวน 20 ครั้ง ตำแหน่งปลายมือของเครื่องป้อนชิ้นงานมีความคลาดเคลื่อน -0.2015mm ในแนวแกน X ซึ่งอยู่ในช่วงที่ยอมรับได้เพราะเครื่องป้อนชิ้นงานเองมีจุดลื่นอยู่แล้ว

5.2 ปัญหาและแนวทางแก้ไขปัญหา

1. ปัญหาที่พบ

ปัจจุบันระบบนี้ยังอยู่ในช่วงของการพัฒนา ทำให้ระบบยังมีข้อจำกัดบางประการที่เกิดจากการทำตัวต้นแบบ จากผลการทดสอบจะพบว่าการเคลื่อนตำแหน่งของตัวหุ่นยังมีความผิดพลาดอยู่บ้าง เกิดจากทั้งปัจจัยทางด้านกลไกและการเขียนซอฟต์แวร์ควบคุม ซึ่งสามารถวิเคราะห์สาเหตุเบื้องต้นมาจากการยึดแน่นของชิ้นส่วน ตัวต้านทานปรับค่าได้ที่ใช้สำหรับป้อนกลับตำแหน่งมีจำนวนรอบที่มากไป แต่มีการบิดของแกนอยู่ในช่วงแค่ 90 องศา ทำให้การป้อนกลับไม่แม่นยำ อีกทั้งการเขียนซอฟต์แวร์ส่งตำแหน่งจากคอนโทรลเลอร์ตัวมาสเตอร์ให้กับคอนโทรลเลอร์ตัวสเลฟในแบบความเร็วคงที่ ทำให้มีการกระชากของมอเตอร์เมื่อถึงจุดหมาย

2. แนวทางแก้ไข

ฮาร์ดแวร์

ควรมีการเปลี่ยนการป้อนกลับจากการใช้ตัวต้านทานปรับค่าได้ เป็นเอนโค้ดเดอร์ทุกส่วน เพราะจะมีการให้ตำแหน่งที่มีความละเอียดและความถูกต้องมากกว่า

ซอฟต์แวร์

การเขียนโปรแกรมส่งข้อมูลตำแหน่งจากมาสเตอร์ ควรจะอยู่ในรูปแบบที่มีความเร่งและความหน่วงมีการลดความเร็วลงเมื่อใกล้ถึงจุดหมายด้วย

5.3 ข้อเสนอแนะและแนวทางพัฒนา

ระบบนี้ยังสามารถพัฒนาต่อไปอีกในอนาคตให้มีฟังก์ชันการทำงานที่หลากหลาย โดยการเปลี่ยนรูปแบบการจับยึดที่ปลายมือด้วยการออกแบบให้อยู่ในรูปของโมดูลที่สามารถถอดเปลี่ยนได้ ให้เป็นไปตามจุดประสงค์การใช้งานในรูปแบบที่แตกต่างกันไป ระบบนี้มีความได้เปรียบสูงในอุตสาหกรรมเพราะสามารถนำมาใช้แทนแรงงานในปัจจุบัน สามารถลดเวลาที่ใช้ในการป้อนชิ้นงานให้กับเครื่องปั๊ม อีกทั้งชั่วโมงการทำงานจริงในหนึ่งวันมากกว่าการทำงานด้วยแรงงานมนุษย์ ซึ่งจะ เป็นประโยชน์สำหรับผู้ประกอบการที่จะลดต้นทุนการผลิตภาคอุตสาหกรรม และนอกจากนี้ยังเหมาะสมสำหรับอุตสาหกรรมที่ต้องการความปลอดภัยสูง

ในประเทศไทยได้มีการนำเทคโนโลยีเครื่องจักรอัตโนมัติมาใช้กันมากขึ้น เพื่อทดแทนการขาดแคลนแรงงานในการผลิต ในปัจจุบันการนำเข้าหุ่นยนต์หรือเครื่องจักรอัตโนมัติจากต่างประเทศมีราคาสูง การพัฒนาระบบเครื่องจักรอัตโนมัติภายในประเทศควรที่จะได้รับการส่งเสริมและสนับสนุนจากทางภาครัฐและเอกชน



เอกสารอ้างอิง

- [1] Qian Dong and Jianying Xie, “Designing and Tuning of PID Controller for a Digital DC Position Servo System,” Intelligent Control and Automation, vol. 2, June 2002, pp. 905-908 .
- [2] Thaicontrol, (2011, November 27), “PID Tuning,” [Online], Available: <https://thaicontrol.wordpress.com/2011/11/27/pid-tuning/>.
- [3] Arduino, “Serial,” [Online], Available: <https://www.arduino.cc>.
- [4] Keith Thompson, (2013, February 6), “Programming PID loops in C,” [Online], Available: <http://programmers.stackexchange.com/questions/186124/programming-pid-loops-in-c>.
- [5] Dogan Ibrahim, “Advance PIC Microcontroller Project in C,” Elsevier. MA:Newnes, 2008.
- [6] Arduino, “Arduino ATmega2560,” [Online], 2009, [cited 2015 Sep 12], Available from: <https://www.arduino.cc/en/Main/arduinoBoardMega2560>
- [7] Atmel microcontroller PDF, “ATmel ATmega Datasheet,” Available from: www.atmel.com
- [8] Intersil Americas, “HIP4080 Datasheet,” Available from: www.intersil.com
- [9] Fairchild Semicondustor, “DM74LS73A J-K Flip-flops Datasheet,” August 2000, Available from: www.fairchildsemi.com
- [10] Fairchild Semicondustor, “DM74LS08A Datasheet,” August 2000, Available from: www.fairchildsemi.com
- [11] Fairchild Semicondustor, “LM311 single comparator Datasheet,” 2001, Available from: www.fairchildsemi.com
- [12] National Semiconductor, “LM78XX Series Datasheet,” January 2000, Available from: www.national.com



ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

โค้ดโปรแกรม

1. ทดลองควบคุมความเร็ว

```
int left;
int right;
int pulseWidth;

void setup() {
  // put your setup code here, to run once:
  Serial.begin(9600);
  pinMode(3, OUTPUT);
  pinMode(11, OUTPUT);
;

  TCCR2A = _BV(COM2A1) | _BV(COM2B1) | _BV(WGM21) | _BV(WGM20);
  TCCR2B = _BV(CS22);
  OCR2A = 255;
  OCR2B = 255;
}

void loop() {
  // put your main code here, to run repeatedly:
  pulseWidth=analogRead(A0);
  Serial.print(" pulseWidth :");
  Serial.print(pulseWidth);
  Serial.print(" ");
  if(pulseWidth<511)
  {
    OCR2B=255;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

right=511-pulseWidth;
Serial.print("Right :");
Serial.println(right);
OCR2A=map(right,510,0,0,255);
}
else if(pulseWidth>511)
{
OCR2A=255;
left=pulseWidth-512;
Serial.print("left :");
Serial.println(left);
OCR2B=map(left,511,0,0,255);
}
}

```

2. ทดลองหมุนไปกลับ (PID Control)

```

int left;
int right;
int pulseWidth;
int mapH=509;//แมพสูง
int mapL=0;//แมพต่ำ
int p_control=A0;//analogRead pin
int p_moter=A1;
int gapH=513;//ค่าสูงของช่วงหยุด
int gapL=510;//ค่าต่ำของช่วงหยุด
int Lc=514;
int Rc=509;
int cH=255;
int cL=0;
int Ke=25;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int Ki=0.6;
int R_control;
int R_motor;
int E;
int I=0;

void setup() {
  Serial.begin(19200);
  pinMode(3, OUTPUT);// RotateRight
  pinMode(11, OUTPUT);//RotateLeft
  pinMode(4,OUTPUT);// RotateRight
  pinMode(12,OUTPUT);//RotateLeft
  TCCR2A = _BV(COM2A1) | _BV(COM2A0) | _BV(COM2B1) | _BV(COM2B0) |
  _BV(WGM21) | _BV(WGM20);
  TCCR2B = _BV(CS22);
}

void loop() {
  R_control=analogRead(p_control);
  R_motor=analogRead(p_moter);
  E=R_control-R_motor;
  I=I+E;
  pulseWidth=(Ke*E)+(Ki*I)+512;
  // Serial.print(" pulseWidth :");
  // Serial.println(pulseWidth);
  // Serial.print(" ");
  if(pulseWidth<=Rc)
  {
    digitalWrite(12,LOW);
    digitalWrite(4,HIGH);
    right=Rc-pulseWidth;
  }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

OCR2B=cH;
    // Serial.print("Right :");
    // Serial.println(right);
OCR2A=map(right,mapH,mapL,0,255);

}
else if(pulseWidth>gapL&&pulseWidth<gapH)
{
    // Serial.println("stop");
    digitalWrite(12,LOW);
    digitalWrite(4,LOW);
    OCR2B=255;
    OCR2A=255;
}
else if(pulseWidth>=Lc)
{
    digitalWrite(4,LOW);
    digitalWrite(12,HIGH);
    left=pulseWidth-Lc;
    OCR2A=cH;
    //Serial.print("left :");
    //Serial.println(left);
    OCR2B=map(left,mapH,mapL,0,255);
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



HIP4080

**NOT RECOMMENDED FOR NEW DESIGNS
POSSIBLE SUBSTITUTE PRODUCT
INTERSIL PART NUMBER HIP4080A**

February, 2003

FN3178.11

**80V/2.5A Peak, High Frequency
Full Bridge FET Driver**

The HIP4080 is a high frequency, medium voltage Full Bridge N-Channel FET driver IC, available in 20 lead plastic SOIC and DIP packages. The HIP4080 includes an input comparator, used to facilitate the "hysteresis" and PWM modes of operation. Its HEN (high enable) lead can force current to freewheel in the bottom two external power MOSFETs, maintaining the upper power MOSFETs off. Since it can switch at frequencies up to 1MHz, the HIP4080 is well suited for driving Voice Coil Motors, switching power amplifiers and power supplies.

HIP4080 can also drive medium voltage brush motors, and two HIP4080s can be used to drive high performance stepper motors, since the short minimum "on-time" can provide fine micro-stepping capability.

Short propagation delays of approximately 55ns maximizes control loop crossover frequencies and dead-times which can be adjusted to near zero to minimize distortion, resulting in precise control of the driven load.

The similar HIP4081 IC allows independent control of all 4 FETs in an Full Bridge configuration.

See also, Application Note AN9324 for the HIP4080.

Similar part, HIP4080A, includes under voltage circuitry which doesn't require the circuitry shown in Figure 30 of this data sheet.

Ordering Information

PART NUMBER	TEMP. RANGE (°C)	PACKAGE	PKG. NO.
HIP4080IP	-40 to 85	20 Lead PDIP	E20.3
HIP4080IB	-40 to 85	20 Lead SOIC	M20.3

Features

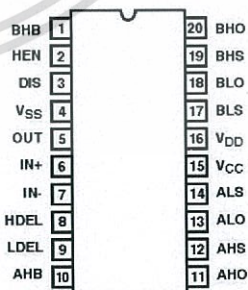
- Drives N-Channel FET Full Bridge Including High Side Chop Capability
- Bootstrap Supply Max Voltage to 95V_{DC}
- Drives 1000pF Load at 1MHz in Free Air at 50°C with Rise and Fall Times of 10ns (Typ)
- User-Programmable Dead Time
- Charge-Pump and Bootstrap Maintain Upper Bias Supplies
- DIS (Disable) Pin Pulls Gates Low
- Input Logic Thresholds Compatible with 5V to 15V Logic Levels
- Very Low Power Consumption

Applications

- Medium/Large Voice Coil Motors
- Full Bridge Power Supplies
- Switching Power Amplifiers
- High Performance Motor Controls
- Noise Cancellation Systems
- Battery Powered Vehicles
- Peripherals
- U.P.S.

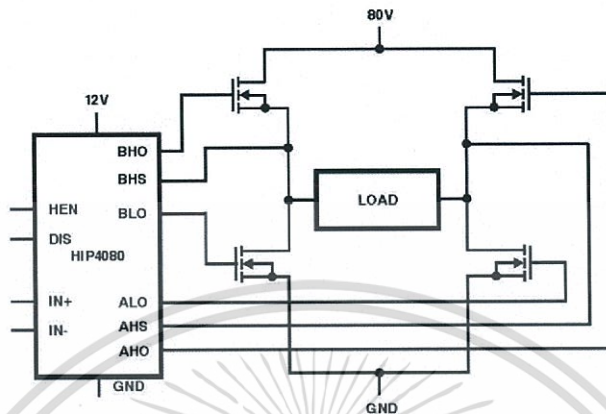
Pinout

HIP4080
(20-LEAD PDIP, SOIC)
TOP VIEW

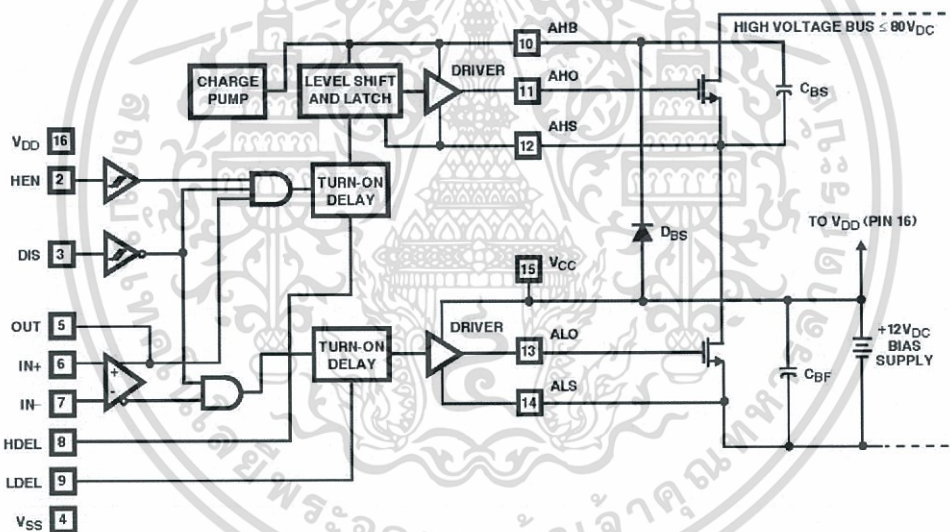


HIP4080

Application Block Diagram



Functional Block Diagram (1/2 HIP4080)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

HIP4080

Absolute Maximum Ratings

Supply Voltage, V_{DD} and V_{CC} -0.3V to 16V
 Logic I/O Voltages -0.3V to V_{DD} +0.3V
 Voltage on AHS, BHS -6.0V (Transient) to 80V (25°C to 125°C)
 Voltage on AHS, BHS -6.0V (Transient) to 70V (-55°C to 125°C)
 Voltage on ALS, BLS -2.0V (Transient) to +2.0V (Transient)
 Voltage on AHB, BHB, VAHS, BHS -0.3V to V_{AHS} , BHS +16V
 Voltage on ALO, BLO V_{ALS} , BLS -0.3V to V_{CC} +0.3V
 Voltage on AHO, BHO V_{AHS} , BHS -0.3V to V_{AHB} , BHB +0.3V
 Input Current, HDEL and LDEL -5mA to 0mA
 Phase Slew Rate 20V/ns
 All Voltages relative to pin 4, V_{SS} , unless otherwise specified.

Thermal Information

Thermal Resistance (Typical, Note 1) θ_{JA} (°C/W)
 SOIC Package 85
 PDIP Package 75
 Maximum Power Dissipation at 85°C
 SOIC Package 470mW
 DIP Package 530mW
 Storage Temperature Range -65°C to 150°C
 Operating Max. Junction Temperature 125°C
 Lead Temperature (Soldering 10s) 300°C
 (SOIC - Lead Tips Only)

Operating Conditions

Supply Voltage, V_{DD} and V_{CC} +8V to +15V
 Voltage on ALS, BLS -1.0V to +1.0V
 Voltage on AHB, BHB V_{AHS} , BHS +5V to V_{AHS} , BHS +15V
 Input Current, HDEL and LDEL -500 μ A to -50 μ A
 Operating Ambient Temperature Range -40°C to 85°C

CAUTION: Stresses above those listed in "Absolute Maximum Ratings" may cause permanent damage to the device. This is a stress only rating and operation of the device at these or any other conditions above those indicated in the operational sections of this specification is not implied.

NOTE:

- θ_{JA} is measured with the component mounted on an evaluation PC board in free air.

Electrical Specifications $V_{DD} = V_{CC} = V_{AHB} = V_{BHB} = 12V$, $V_{SS} = V_{ALS} = V_{BLS} = V_{AHS} = V_{BHS} = 0V$, $R_{HDEL} = R_{LDEL} = 100k\Omega$, and $T_A = 25^\circ C$, Unless Otherwise Specified

PARAMETERS	SYMBOL	TEST CONDITIONS	$T_J = 25^\circ C$			$T_J = -40^\circ C$ TO $125^\circ C$		UNITS
			MIN	TYP	MAX	MIN	MAX	
SUPPLY CURRENTS AND CHARGE PUMPS								
V_{DD} Quiescent Current	I_{DD}	IN- = 2.5V, Other Inputs = 0V	8	10.5	13	7	14	mA
V_{DD} Operating Current	I_{DDO}	Outputs switching $f = 500kHz$	9	11	14	8	15	mA
V_{CC} Quiescent Current	I_{CC}	IN- = 2.5V, Other Inputs = 0V, $I_{ALO} = I_{BLO} = 0$	-	25	80	-	100	μA
V_{CC} Operating Current	I_{CCO}	$f = 500kHz$, No Load	1	1.5	2.0	0.8	3	mA
AHB, BHB Quiescent Current - Qpump Output Current	I_{AHB} , I_{BHB}	IN- = 2.5V, Other Inputs = 0V, $I_{AHO} =$ $I_{BHO} = 0$, $V_{DD} = V_{CC} = V_{AHB} =$ $V_{BHB} = 10V$	-50	-30	-15	-60	-10	μA
AHB, BHB Operating Current	I_{AHBO} , I_{BHBO}	$f = 500kHz$, No Load	0.5	0.9	1.3	0.4	1.7	mA
AHS, BHS, AHB, BHB Leakage Current	I_{HLK}	$V_{AHS} = V_{BHS} = V_{AHB} = V_{BHB} = 95V$	-	0.02	1.0	-	10	μA
AHB-AHS, BHB-BHS Qpump Output Voltage	$V_{AHB} -$ V_{AHS} $V_{BHB} -$ V_{BHS}	$I_{AHB} = I_{AHS} = 0$, No Load	11.5	12.6	14.0	10.5	14.5	V
INPUT COMPARATOR PINS: IN+, IN-, OUT								
Offset Voltage	V_{OS}	Over Common Mode Voltage Range	-10	0	+10	-15	+15	mV
Input Bias Current	I_{IB}		0	0.5	2	0	4	μA
Input Offset Current	I_{OS}		-1	0	+1	-2	+2	μA
Input Common Mode Voltage Range	CMVR		1	-	V_{DD} -1.5	1	V_{DD} -1.5	V
Voltage Gain	AVOL		10	25	-	10	-	V/mV

HIP4080

Electrical Specifications $V_{DD} = V_{CC} = V_{AHB} = V_{BHB} = 12V$, $V_{SS} = V_{ALS} = V_{BLS} = V_{AHS} = V_{BHS} = 0V$, $R_{HDEL} = R_{LDEL} = 100K$, and $T_A = 25^\circ C$, Unless Otherwise Specified (Continued)

PARAMETERS	SYMBOL	TEST CONDITIONS	$T_J = 25^\circ C$			$T_J = -40^\circ C$ TO $125^\circ C$		UNITS
			MIN	TYP	MAX	MIN	MAX	
OUT High Level Output Voltage	V_{OH}	$IN+ > IN-$, $I_{OH} = -300\mu A$	$V_{DD} - 0.4$	-	-	$V_{DD} - 0.5$	-	V
OUT Low Level Output Voltage	V_{OL}	$IN+ < IN-$, $I_{OL} = 300\mu A$	-	-	0.3	-	0.4	V
High Level Output Current	I_{OH}	$V_{OUT} = 6V$	-9	-7	-4	-11	-2	mA
Low Level Output Current	I_{OL}	$V_{OUT} = 6V$	8	10	12	5	14	mA
INPUT PINS: DIS								
Low Level Input Voltage	V_{IL}	Full Operating Conditions	-	-	1.0	-	0.8	V
High Level Input Voltage	V_{IH}	Full Operating Conditions	2.5	-	-	2.7	-	V
Input Voltage Hysteresis			-	35	-	-	-	mV
Low Level Input Current	I_{IL}	$V_{IN} = 0V$, Full Operating Conditions	-130	-100	-75	-135	-65	μA
High Level Input Current	I_{IH}	$V_{IN} = 5V$, Full Operating Conditions	-1	-	+1	-10	+10	μA
INPUT PINS: HEN								
Low Level Input Voltage	V_{IL}	Full Operating Conditions	-	-	1.0	-	0.8	V
High Level Input Voltage	V_{IH}	Full Operating Conditions	2.5	-	-	2.7	-	V
Input Voltage Hysteresis			-	35	-	-	-	mV
Low Level Input Current	I_{IL}	$V_{IN} = 0V$, Full Operating Conditions	-260	-200	-150	-270	-130	μA
High Level Input Current	I_{IH}	$V_{IN} = 5V$, Full Operating Conditions	-1	-	+1	-10	+10	μA
TURN-ON DELAY PINS: LDEL AND HDEL								
LDEL, HDEL Voltage	V_{HDEL}, V	$I_{HDEL} = I_{LDEL} = -100\mu A$	4.9	5.1	5.3	4.8	5.4	V
GATE DRIVER OUTPUT PINS: ALO, BLO, AHO, AND BHO								
Low Level Output Voltage	V_{OL}	$I_{OUT} = 100mA$	0.70	0.85	1.0	0.5	1.1	V
High Level Output Voltage	$V_{CC} - V_{OH}$	$I_{OUT} = -100mA$	0.8	0.95	1.1	0.5	1.2	V
Peak Pull-up Current	I_{O+}	$V_{OUT} = 0V$	1.7	2.6	3.8	1.4	4.1	A
Peak Pull-down Current	I_{O-}	$V_{OUT} = 12V$	1.7	2.4	3.3	1.3	3.6	A

Switching Specifications $V_{DD} = V_{CC} = V_{AHB} = V_{BHB} = 12V$, $V_{SS} = V_{ALS} = V_{BLS} = V_{AHS} = V_{BHS} = 0V$, $R_{HDEL} = R_{LDEL} = 10K$, $C_L = 1000pF$, and $T_A = 25^\circ C$, Unless Otherwise Specified

PARAMETERS	SYMBOL	TEST CONDITIONS	$T_J = 25^\circ C$			$T_J = -40^\circ C$ TO $125^\circ C$		UNITS
			MIN	TYP	MAX	MIN	MAX	
Lower Turn-off Propagation Delay (IN+/IN- to ALO/BLO)	T_{LPHL}		-	40	70	-	90	ns
Upper Turn-off Propagation Delay (IN+/IN- to AHO/BHO)	T_{HPLH}		-	50	80	-	110	ns
Lower Turn-on Propagation Delay (IN+/IN- to ALO/BLO)	T_{LPLH}	$R_{HDEL} = R_{LDEL} = 10K$	-	45	70	-	90	ns
Upper Turn-on Propagation Delay (IN+/IN- to AHO/BHO)	T_{HPLH}	$R_{HDEL} = R_{LDEL} = 10K$	-	70	110	-	140	ns
Rise Time	T_r		-	10	25	-	35	ns
Fall Time	T_f		-	10	25	-	35	ns
Turn-on Input Pulse Width	$T_{PWIN-ON}$	$R_{HDEL} = R_{LDEL} = 10K$	50	-	-	50	-	ns
Turn-off Input Pulse Width	$T_{PWIN-OFF}$	$R_{HDEL} = R_{LDEL} = 10K$	40	-	-	40	-	ns

HIP4080

Switching Specifications $V_{DD} = V_{CC} = V_{AHB} = V_{BHB} = 12V$, $V_{SS} = V_{ALS} = V_{BLS} = V_{AHS} = V_{BHS} = 0V$, $R_{HDEL} = R_{LDEL} = 10K$, $C_L = 1000pF$, and $T_A = 25^\circ C$, Unless Otherwise Specified (Continued)

PARAMETERS	SYMBOL	TEST CONDITIONS	$T_J = 25^\circ C$			$T_J = -40^\circ C$ TO $125^\circ C$		UNITS
			MIN	TYP	MAX	MIN	MAX	
Disable Turn-off Propagation Delay (DIS - Lower Outputs)	T_{DISLOW}		-	45	75	-	95	ns
Disable Turn-off Propagation Delay (DIS - Upper Outputs)	$T_{DISHIGH}$		-	55	85	-	105	ns
Disable to Lower Turn-on Propagation Delay (DIS - ALO and BLO)	T_{DLPLH}		-	35	70	-	90	ns
Refresh Pulse Width (ALO and BLO)	T_{REF-PW}		160	260	380	140	420	ns
Disable to Upper Enable (DIS - AHO and BHO)	T_{UEN}		-	335	500	-	550	ns
HEN-AHO, BHO Turn-off, Propagation Delay	$T_{HEN-PHL}$	$R_{HDEL} = R_{LDEL} = 10K$	-	35	70	-	90	ns
HEN-AHO, BHO Turn-on, Propagation Delay	$T_{HEN-PLH}$	$R_{HDEL} = R_{LDEL} = 10K$	-	60	90	-	110	ns

TRUTH TABLE

INPUT			OUTPUT			
IN+ > IN-	HEN	DIS	ALO	AHO	BLO	BHO
X	X	1	0	0	0	0
1	1	0	0	1	1	0
0	1	0	1	0	0	1
1	0	0	0	0	1	0
0	0	0	1	0	0	0

Pin Descriptions

PIN NUMBER	SYMBOL	DESCRIPTION
1	BHB	B High-side Bootstrap supply. External bootstrap diode and capacitor are required. Connect cathode of bootstrap diode and positive side of bootstrap capacitor to this pin. Internal charge pump supplies 30 μ A out of this pin to maintain bootstrap supply. Internal circuitry clamps the bootstrap supply to approximately 12.8V.
2	HEN	High-side Enable input. Logic level input that when low overrides IN+/IN- (Pins 6 and 7) to put AHO and BHO drivers (Pins 11 and 20) in low output state. When HEN is high AHO and BHO are controlled by IN+/IN- inputs. The pin can be driven by signal levels of 0V to 15V (no greater than V_{DD}).
3	DIS	DiSable input. Logic level input that when taken high sets all four outputs low. DIS high overrides all other inputs. When DIS is taken low the outputs are controlled by the other inputs. The pin can be driven by signal levels of 0V to 15V (no greater than V_{DD}).
4	V_{SS}	Chip negative supply, generally will be ground.
5	OUT	OUTPUT of the input control comparator. This output can be used for feedback and hysteresis.
6	IN+	Non-inverting input of control comparator. If IN+ is greater than IN- (Pin 7) then ALO and BHO are low level outputs and BLO and AHO are high level outputs. If IN+ is less than IN- then ALO and BHO are high level outputs and BLO and AHO are low level outputs. DIS (Pin 3) high level will override IN+/IN- control for all outputs. HEN (Pin 2) low level will override IN+/IN- control of AHO and BHO. When switching in four quadrant mode, dead time in a half bridge leg is controlled by HDEL and LDEL (Pins 8 and 9).
7	IN-	Inverting input of control comparator. See IN+ (Pin 6) description.
8	HDEL	High-side turn-on DELay. Connect resistor from this pin to V_{SS} to set timing current that defines the turn-on delay of both high-side drivers. The low-side drivers turn-off with no adjustable delay, so the HDEL resistor guarantees no shoot-through by delaying the turn-on of the high-side drivers. HDEL reference voltage is approximately 5.1V.

HIP4080

Pin Descriptions (Continued)

PIN NUMBER	SYMBOL	DESCRIPTION
9	LDEL	Low-side turn-on DELay. Connect resistor from this pin to V_{SS} to set timing current that defines the turn-on delay of both low-side drivers. The high-side drivers turn-off with no adjustable delay, so the LDEL resistor guarantees no shoot-through by delaying the turn-on of the low-side drivers. LDEL reference voltage is approximately 5.1V.
10	AHB	A High-side Bootstrap supply. External bootstrap diode and capacitor are required. Connect cathode of bootstrap diode and positive side of bootstrap capacitor to this pin. Internal charge pump supplies 30 μ A out of this pin to maintain bootstrap supply. Internal circuitry clamps the bootstrap supply to approximately 12.8V.
11	AHO	A High-side Output. Connect to gate of A High-side power MOSFET.
12	AHS	A High-side Source connection. Connect to source of A High-side power MOSFET. Connect negative side of bootstrap capacitor to this pin.
13	ALO	A Low-side Output. Connect to gate of A Low-side power MOSFET.
14	ALS	A Low-side Source connection. Connect to source of A Low-side power MOSFET.
15	V_{CC}	Positive supply to gate drivers. Must be same potential as V_{DD} (Pin 16). Connect to anodes of two bootstrap diodes.
16	V_{DD}	Positive supply to lower gate drivers. Must be same potential as V_{CC} (Pin 15). De-couple this pin to V_{SS} (Pin 4).
17	BLS	B Low-side Source connection. Connect to source of B Low-side power MOSFET.
18	BLO	B Low-side Output. Connect to gate of B Low-side power MOSFET.
19	BHS	B High-side Source connection. Connect to source of B High-side power MOSFET. Connect negative side of bootstrap capacitor to this pin.
20	BHO	B High-side Output. Connect to gate of B High-side power MOSFET.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

FAIRCHILD
SEMICONDUCTOR™

August 1986
Revised March 2000

DM74LS73A

Dual Negative-Edge-Triggered Master-Slave J-K Flip-Flops with Clear and Complementary Outputs

General Description

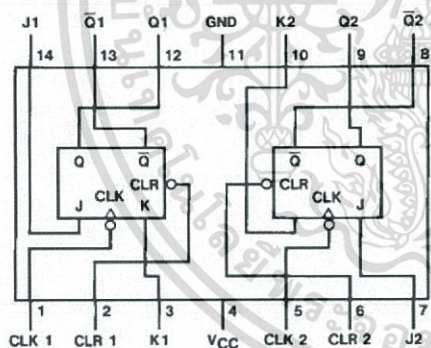
This device contains two independent negative-edge-triggered J-K flip-flops with complementary outputs. The J and K data is processed by the flip-flops on the falling edge of the clock pulse. The clock triggering occurs at a voltage level and is not directly related to the transition time of the negative going edge of the clock pulse. The data on the J and K inputs is allowed to change while the clock is HIGH or LOW without affecting the outputs as long as setup and hold times are not violated. A low logic level on the clear input will reset the outputs regardless of the levels of the other inputs.

Ordering Code:

Order Number	Package Number	Package Description
DM74LS73AM	M14A	14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow
DM74LS73AN	N14A	14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter 'X' to the ordering code.

Connection Diagram



Function Table

		Inputs			Outputs	
	CLR	CLK	J	K	Q	Q̄
	L	X	X	X	L	H
	H	↓	L	L	Q ₀	Q̄ ₀
	H	↓	H	L	H	L
	H	↓	L	H	L	H
	H	↓	H	H	Toggle	
	H	H	X	X	Q ₀	Q̄ ₀

H = HIGH Logic Level

L = LOW Logic Level

X = Either LOW or HIGH Logic Level

↓ = Negative going edge of pulse.

Q₀ = The output logic level before the indicated input conditions were established.

Toggle = Each output changes to the complement of its previous level on each falling edge of the clock pulse.

Absolute Maximum Ratings(Note 1)

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	0°C to +70°C
Storage Temperature Range	-65°C to +150°C

Note 1: The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the Electrical Characteristics tables are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.

Recommended Operating Conditions

Symbol	Parameter	Min	Nom	Max	Units
V _{CC}	Supply Voltage	4.75	5	5.25	V
V _{IH}	HIGH Level Input Voltage	2			V
V _{IL}	LOW Level Input Voltage			0.8	V
I _{OH}	HIGH Level Output Current			-0.4	mA
I _{OL}	LOW Level Output Current			8	mA
f _{CLK}	Clock Frequency (Note 2)	0		30	MHz
f _{CLK}	Clock Frequency (Note 3)	0		25	MHz
t _w	Pulse Width (Note 2)	Clock HIGH	20		ns
		Preset LOW	25		
		Clear LOW	25		
t _w	Pulse Width (Note 3)	Clock HIGH	25		ns
		Preset LOW	30		
		Clear LOW	30		
t _{SU}	Setup Time (Note 2)(Note 4)	20↓			ns
t _{SU}	Setup Time (Note 3)(Note 4)	25↓			ns
t _H	Hold Time (Note 2)(Note 4)	0↓			ns
t _H	Hold Time (Note 3)(Note 4)	5↓			ns
T _A	Free Air Operating Temperature	0		70	°C

Note 2: C_L = 15 pF, R_L = 2 kΩ, T_A = 25°C and V_{CC} = 5V.

Note 3: C_L = 50 pF, R_L = 2 kΩ, T_A = 25°C and V_{CC} = 5V.

Note 4: The symbol (↓) indicates the falling edge of the clock pulse is used for reference.

Electrical Characteristics

over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions	Min	Typ (Note 5)	Max	Units
V_I	Input Clamp Voltage	$V_{CC} = \text{Min}, I_I = -18 \text{ mA}$			-1.5	V
V_{OH}	HIGH Level Output Voltage	$V_{CC} = \text{Min}, I_{OH} = \text{Max}$ $V_{IL} = \text{Max}, V_{IH} = \text{Min}$	2.7	3.4		V
V_{OL}	LOW Level Output Voltage	$V_{CC} = \text{Min}, I_{OL} = \text{Max}$ $V_{IL} = \text{Max}, V_{IH} = \text{Min}$ $I_{OL} = 4 \text{ mA}, V_{CC} = \text{Min}$		0.35 0.25	0.5 0.4	V
I_I	Input Current @ Max Input Voltage	$V_{CC} = \text{Max}$ $V_I = 7 \text{ V}$			0.1 0.3 0.4	mA
I_{IH}	HIGH Level Input Current	$V_{CC} = \text{Max}$ $V_I = 2.7 \text{ V}$			20 60 80	μA
I_{IL}	LOW Level Input Current	$V_{CC} = \text{Max}$ $V_I = 0.4 \text{ V}$			-0.4 -0.8 -0.8	mA
I_{OS}	Short Circuit Output Current	$V_{CC} = \text{Max}$ (Note 6)	-20		-100	mA
I_{CC}	Supply Current	$V_{CC} = \text{Max}$ (Note 7)		4	6	mA

Note 5: All typicals are at $V_{CC} = 5 \text{ V}$, $T_A = 25^\circ\text{C}$.

Note 6: Not more than one output should be shorted at a time, and the duration should not exceed one second. For devices, with feedback from the outputs, where shorting the outputs to ground may cause the outputs to change logic state, an equivalent test may be performed where $V_O = 2.125 \text{ V}$ with the minimum and maximum limits reduced by one half from their stated values. This is very useful when using automatic test equipment.

Note 7: With all outputs OPEN, I_{CC} is measured with the Q and \bar{Q} outputs HIGH in turn. At the time of measurement, the clock is grounded.

Switching Characteristics

at $V_{CC} = 5 \text{ V}$ and $T_A = 25^\circ\text{C}$.

Symbol	Parameter	From (Input) To (Output)	$R_L = 2 \text{ k}\Omega$				Units
			$C_L = 15 \text{ pF}$		$C_L = 50 \text{ pF}$		
			Min	Max	Min	Max	
f_{MAX}	Maximum Clock Frequency		30		25		MHz
t_{PHL}	Propagation Delay Time HIGH-to-LOW Level Output	Clear to Q		20		28	ns
t_{PLH}	Propagation Delay Time LOW-to-HIGH Level Output	Clear to \bar{Q}		20		24	ns
t_{PLH}	Propagation Delay Time LOW-to-HIGH Level Output	Clock to Q or \bar{Q}		20		24	ns
t_{PHL}	Propagation Delay Time HIGH-to-LOW Level Output	Clock to Q or \bar{Q}		20		28	ns

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

LM311

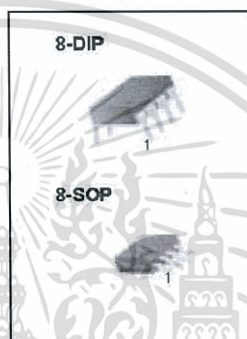
Single Comparator

Features

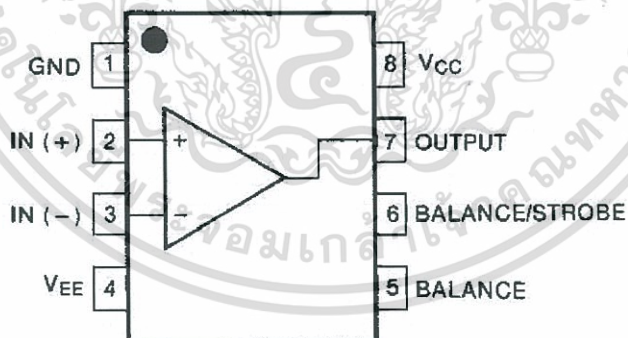
- Low input bias current : 250nA (Max)
- Low input offset current : 50nA (Max)
- Differential Input Voltage : $\pm 30V$
- Power supply voltage : single 5.0V supply to $\pm 15V$.
- Offset voltage null capability.
- Strobe capability.

Description

The LM311 series is a monolithic, low input current voltage comparator. The device is also designed to operate from dual or single supply voltage.



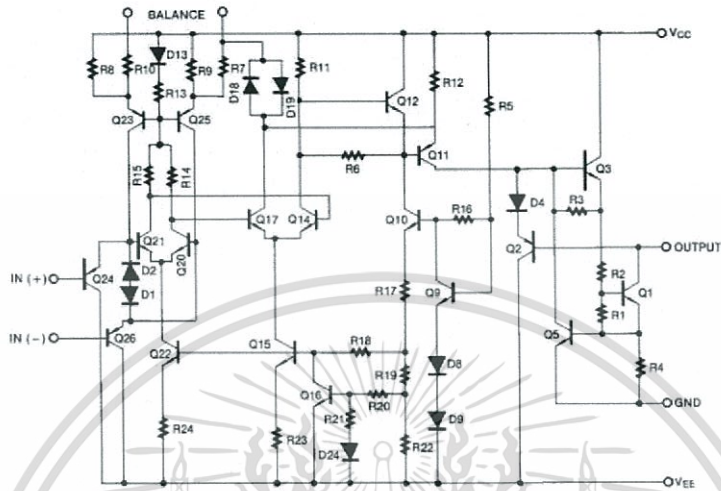
Internal Block Diagram



Rev. 1.0.1

LM311

Schematic Diagram



Absolute Maximum Ratings

Parameter	Symbol	Value	Unit
Total Supply Voltage	VCC	36	V
Output to Negative Supply Voltage LM311	VO - VEE	40	V
Ground to Negative voltage	VEE	-30	V
Differential Input Voltage	VI(DIFF)	30	V
Input Voltage	VI	±15	V
Output Short Circuit Duration	-	10	sec
Power Dissipation	PD	500	mW
Operating Temperature Range	TOPR	0 ~ +70	°C
Storage Temperature Range	TSTG	-65 ~ +150	°C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Electrical Characteristics

($V_{CC} = 15V$, $T_A = 25^\circ C$, unless otherwise specified)

Parameter	Symbol	Conditions	Min.	Typ.	Max.	Unit
Input Offset Voltage	V_{IO}	$R_S \leq 50K\Omega$	-	1.0	7.5	mV
			Note 1	-	-	
Input Offset Current	I_{IO}		-	6	50	nA
			Note 1	-	-	
Input Bias Current	I_{BIAS}		-	100	250	nA
			Note 1	-	-	
Voltage Gain	G_V	-	40	200	-	V/mV
Response Time	T_{RES}	Note 2	-	200	-	ns
Saturation Voltage	V_{SAT}	$I_O = 50mA, V_I \leq -10mV$	-	0.75	1.5	V
		$V_{CC} \geq 4.5V, V_{EE} = 0V$ $I_O = 8mA, V_I \leq -10mV$, Note 1	-	0.23	0.4	
Strobe "ON" Current	$I_{STR(ON)}$	-	-	3	-	mA
Output Leakage Current	I_{SINK}	$I_{STR} = 3mA, V_I \geq 10mV$ $V_O = 15V, V_{CC} = \pm 15V$	-	0.2	50	nA
Input Voltage Range	$V_{I(R)}$	Note 1	-14.5 to 13.0	-14.7 to 13.8	-	V
Positive Supply Current	I_{CC}	-	-	3.0	7.5	mA
Negative Supply Current	I_{EE}	-	-	-2.2	-5.0	mA
Strobe Current	I_{STR}	-	-	3	-	mA

Notes :

- $0 \leq T_A \leq +70^\circ C$
- The response time specified is for a 100mV input step with 5mV over drive.

FAIRCHILD
SEMICONDUCTOR™

August 1986
Revised March 2000

DM74LS08

Quad 2-Input AND Gates

General Description

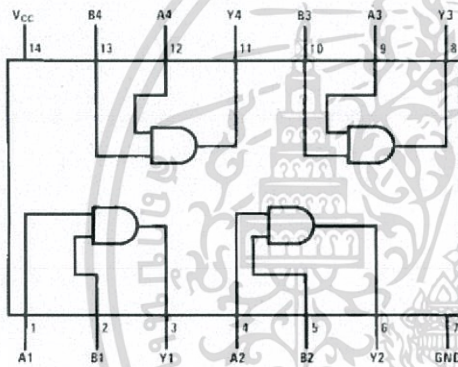
This device contains four independent gates each of which performs the logic AND function.

Ordering Code:

Order Number	Package Number	Package Description
DM74LS08M	M14A	14-Lead Small Outline Integrated Circuit (SOIC), JEDEC MS-120, 0.150 Narrow
DM74LS08SJ	M14D	14-Lead Small Outline Package (SOP), EIAJ TYPE II, 5.3mm Wide
DM74LS08N	N14A	14-Lead Plastic Dual-In-Line Package (PDIP), JEDEC MS-001, 0.300 Wide

Devices also available in Tape and Reel. Specify by appending the suffix letter "X" to the ordering code.

Connection Diagram



Function Table

$Y = AB$

Inputs		Output
A	B	Y
L	L	L
L	H	L
H	L	L
H	H	H

H = HIGH Logic Level
L = LOW Logic Level

Absolute Maximum Ratings (Note 1)

Supply Voltage	7V
Input Voltage	7V
Operating Free Air Temperature Range	0°C to +70°C
Storage Temperature Range	-65°C to +150°C

Note 1: The "Absolute Maximum Ratings" are those values beyond which the safety of the device cannot be guaranteed. The device should not be operated at these limits. The parametric values defined in the Electrical Characteristics tables are not guaranteed at the absolute maximum ratings. The "Recommended Operating Conditions" table will define the conditions for actual device operation.

Recommended Operating Conditions

Symbol	Parameter	Min	Nom	Max	Units
V _{CC}	Supply Voltage	4.75	5	5.25	V
V _{IH}	HIGH Level Input Voltage	2			V
V _{IL}	LOW Level Input Voltage			0.8	V
I _{OH}	HIGH Level Output Current			-0.4	mA
I _{OL}	LOW Level Output Current			8	mA
T _A	Free Air Operating Temperature	0		70	°C

Electrical Characteristics

over recommended operating free air temperature range (unless otherwise noted)

Symbol	Parameter	Conditions	Min	Typ (Note 2)	Max	Units
V _I	Input Clamp Voltage	V _{CC} = Min, I _I = -18 mA			-1.5	V
V _{OH}	HIGH Level Output Voltage	V _{CC} = Min, I _{OH} = Max, V _{IH} = Min	2.7	3.4		V
V _{OL}	LOW Level Output Voltage	V _{CC} = Min, I _{OL} = Max, V _{IL} = Max		0.35	0.5	V
		I _{OL} = 4 mA, V _{CC} = Min		0.25	0.4	
I _I	Input Current @ Max Input Voltage	V _{CC} = Max, V _I = 7V			0.1	mA
I _{IH}	HIGH Level Input Current	V _{CC} = Max, V _I = 2.7V			20	μA
I _{IL}	LOW Level Input Current	V _{CC} = Max, V _I = 0.4V			-0.36	mA
I _{CS}	Short Circuit Output Current	V _{CC} = Max (Note 3)	-20		-100	mA
I _{CCH}	Supply Current with Outputs HIGH	V _{CC} = Max		2.4	4.8	mA
I _{CCL}	Supply Current with Outputs LOW	V _{CC} = Max		4.4	8.8	mA

Switching Characteristics

at V_{CC} = 5V and T_A = 25°C

Symbol	Parameter	R _L = 2 kΩ				Units
		C _L = 15 pF		C _L = 50 pF		
		Min	Max	Min	Max	
t _{PLH}	Propagation Delay Time LOW-to-HIGH Level Output	4	13	6	18	ns
t _{PHL}	Propagation Delay Time HIGH-to-LOW Level Output	3	11	5	18	ns

Note 2: All typicals are at V_{CC} = 5V, T_A = 25°C.

Note 3: Not more than one output should be shorted at a time, and the duration should not exceed one second.