

วงจรควบคุมมอเตอร์ไฟฟ้ากระแสตรงแบบไร้แปรงถ่าน
CONTROLLER CIRCUIT OF BRUSHLESS DC MOTOR

นายยุทธพงษ์ พรหมสีนอง รหัส 48361721
นายชาญวิทย์ ปารี รหัส 48364302

15080952 e.2

ห้องสมุดคณะวิศวกรรมศาสตร์
วันที่รับ...../...../.....
เลขทะเบียน.....5200061.....
เลขเรียกหนังสือ..... ๒๕.....
มหาวิทยาลัยเกษตรศาสตร์ ๒๕๖๑

๒๕๖๑

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์
คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเกษตรศาสตร์
ปีการศึกษา 2551



ใบรับรองโครงการวิศวกรรม

หัวข้อโครงการ	วงจรควบคุมมอเตอร์ไฟฟ้ากระแสตรงแบบไร้แปรงถ่าน
ผู้ดำเนินโครงการ	นายยุทธพงษ์ พรหมสีนอง รหัส 48361721 นายชาญวิทย์ ปารี รหัส 48364302
อาจารย์ที่ปรึกษา	ดร. สมพร เรืองสินชัยวานิช
สาขาวิชา	วิศวกรรมไฟฟ้า
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์
ปีการศึกษา	2551

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยบรบรัมย์ อนุมัติให้โครงการฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า คณะกรรมการสอบโครงการวิศวกรรม

.....
.....ประธานกรรมการ
(ดร.สมพร เรืองสินชัยวานิช)

.....
.....กรรมการ
(อาจารย์สรารุณี วัฒนวงศ์พิทักษ์)

.....
.....กรรมการ
(อาจารย์แสงชัย มังกรทอง)

หัวข้อโครงการ	วงจรควบคุมมอเตอร์ไฟฟ้ากระแสตรงแบบไร้แปรงถ่าน		
ผู้ดำเนินโครงการ	นายยุทธพงษ์	พรมสินอง	รหัส 48361721
	นายชาญวิทย์	ปารี	รหัส 48364302
อาจารย์ที่ปรึกษา	ดร. สมพร	เรืองสินชัยวานิช	
สาขาวิชา	วิศวกรรมไฟฟ้า		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2551		

บทคัดย่อ

โครงการนี้เป็นการสร้างชุดขับเคลื่อนมอเตอร์ไฟฟ้ากระแสตรงแบบไร้แปรงถ่านโดยมีไมโครคอนโทรลเลอร์เบอร์ ATMEGA8 เพื่อควบคุมการทำงานของอินเวอร์เตอร์ 3 เฟส ไมโครคอนโทรลเลอร์จะรับสัญญาณจากอุปกรณ์ฮอลล์เซนเซอร์ที่ติดตั้งในตัวมอเตอร์กระแสตรงแบบไร้แปรงถ่าน และสร้างสัญญาณไปขับอินเวอร์เตอร์ 3 เฟสในโหมดสวิตชิง เป็นลำดับเฟสเพื่อให้มอเตอร์ทำงานอย่างเหมาะสม

Project title	Controller Circuit of Brushless DC Motor		
Name	Mr.Yuttapong	Promseenong	ID.48361721
	Mr.Chanwit	Paree	ID.48364302
Project Advisor	Somporn	Ruangsinchaiwanich, Ph.D.	
Major	Electrical Engineering		
Department	Electrical and Computer Engineering		
Academic year	2008		

ABSTRACT

This project proposes the driver circuit for Brushless DC Motor (BLDC) with ATmega8 microcontroller, controlling the operating of three phases inverter. Microcontroller receives signals from Hall Sensor devices which are embedded in BLDC. Driving signals from microcontroller will drive Three phases Inverter which operates in switching mode for producing phase orders to drive the BLDC properly.

กิตติกรรมประกาศ

ในการทำปริญญาบัตรในครั้งนี้สำเร็จล่วงไปด้วยดีโดยได้รับความช่วยเหลือและได้รับคำแนะนำจาก คร.สมพร เรืองสินชัยวานิช ซึ่งเป็นอาจารย์ที่ปรึกษาปริญญาบัตรในครั้งนี้ และได้กรุณาให้แนวความคิด ช่วยชี้แนวทางในการทำโครงการ อีกทั้งยังช่วยแนะนำแหล่งข้อมูลในการค้นคว้าข้อมูลเพิ่มเติม ทำให้เป็นประโยชน์ต่อการทำโครงการของผู้จัดทำเป็นอย่างมาก

สุดท้ายต้องขอขอบพระคุณ คุณพ่อ และ คุณแม่ อาจารย์ทุกท่านและเพื่อนๆ พี่ๆ ทุกคน ที่คอยให้คำแนะนำและให้การสนับสนุนผู้จัดทำโครงการให้สามารถทำโครงการชิ้นนี้จนสำเร็จล่วงไปได้อย่างดี

คณะผู้จัดทำ

นาย ยุทธพงษ์ พรมสีนอง
นาย ชาญวิทย์ ปารี



สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
กิตติกรรมประกาศ	ค
สารบัญ	ง
สารบัญตาราง	ฉ
สารบัญรูป	ช
บทที่ 1 บทนำ	
1.1 ความเป็นมาและความสำคัญของ โครงการงาน.....	1
1.2 วัตถุประสงค์ของ โครงการงาน	1
1.3 แนวทางการดำเนิน โครงการงาน	1
1.4 ขอบเขตของ โครงการงาน	2
1.5 ประโยชน์ที่คาดว่าจะได้รับ	2
1.6 แผนการดำเนินงาน	2
1.7 งบประมาณ	3
บทที่ 2 หลักการและทฤษฎีเบื้องต้น	
2.1 มอเตอร์ไฟฟ้ากระแสตรง.....	4
2.2 ไมโครคอนโทรลเลอร์ ATMEGA8.....	8
2.3 ทรานซิสเตอร์สนามไฟฟ้า(Field – Effect Transistor ; FET)	14
2.4 อินเวอร์เตอร์ 3 เฟส.....	17
2.5 เซนเซอร์ตรวจจับกระแส(Hall Sensor).....	20
2.6 วงจรเรียงกระแส(Rectifier).....	23

สารบัญ (ต่อ)

หน้า

บทที่ 3 การออกแบบการทดลอง

3.1	อุปกรณ์การทดลอง	27
3.2	ระบบการทำงานของวงจรควบคุมการทำงานของมอเตอร์	28
3.3	การออกแบบการเขียนโปรแกรม	33
3.4	การโปรแกรมไฟล์ .hex ลงในไมโครคอนโทรลเลอร์	35

บทที่ 4 ผลการทดลอง

4.1	จุดประสงค์การทดลอง	38
4.2	ขั้นตอนการทดลอง	38
4.3	ผลการทดลอง	39
4.4	วิเคราะห์ผลการทดลอง	43

บทที่ 5 บทสรุป

5.1	สรุปผลการทดลอง	44
5.2	ปัญหาในการทำงานและแนวทางแก้ไข	44
5.3	ประเมินผล	45
5.4	แนวทางสำหรับการพัฒนา	45

เอกสารอ้างอิง	46
---------------------	----

ภาคผนวก	47
---------------	----

ภาคผนวก (ก) Source Code ของโปรแกรม	48
--	----

ภาคผนวก (ข) เครื่องโปรแกรมไมโครคอนโทรลเลอร์ AVR แบบ ISP	55
---	----

ภาคผนวก (ค) รายการอุปกรณ์อิเล็กทรอนิกส์	70
---	----

ประวัติผู้เขียนโครงการ	71
------------------------------	----

สารบัญตาราง

ตารางที่	หน้า
2.1	หน้าที่การทำงานของขาสัญญาณทั้งหมดของ ATMEGA8..... 10
2.2	อัตราส่วนระหว่างค่า RMS ของฮามอนิกส์กับ V_s ของอินเวอร์เตอร์ 3 เฟส 20
3.1	แสดงการ ON/OFF ของมอสเฟตในวงจรอินเวอร์เตอร์..... 33



สารบัญรูป

รูปที่	หน้า
2.1	ไดอะแกรมของมอเตอร์แบบBLDC 5
2.2	แผนผังของขดลวด, เซ็นเซอร์ เทียบกับเวลา 7
2.3	การจัดการขาของ ATMEGA8..... 9
2.4	มอเตอร์เฟสแบบ n แชนแนลและสัญลักษณ์ 15
2.5	โครงสร้างคีมอเตอร์เฟส n แชนแนล 15
2.6	การทำงานของมอเตอร์เฟสแบบ n แชนแนล 16
2.7	กราฟแสดงคุณลักษณะการทำงานของมอเตอร์เฟส n แชนแนล 17
2.8	วงจรและรูปคลื่นสัญญาณของอินเวอร์เตอร์ 3 เฟสแบบ PWM 19
2.9	การวิเคราะห์สเปคโดยใช้ฮอลล์เซ็นเซอร์..... 21
2.10	ลักษณะภายในของเซ็นเซอร์ตรวจจับกระแส 21
2.11	เซ็นเซอร์ตรวจจับกระแสเบอร์ ACS706TELC-20A 22
2.12	กราฟแสดงแรงดันเอาต์พุตเมื่อตรวจจับกระแสที่ค่าต่างๆ โดยใช้แรงดันไฟเลี้ยง 5 V 22
2.13	การจัดการและตัวไอซีฮอลล์เซ็นเซอร์เบอร์ UGN 3503 23
2.14	รูปแบบการใช้ฮอลล์เซ็นเซอร์ในการทำงาน 23
2.15	วงจรเรียงกระแสแบบครึ่งคลื่น 24
2.16	วงจรเรียงกระแสแบบเต็มคลื่น 24
2.17	แสดงการทำงานของวงจรเรียงกระแสแบบเซ็นเตอร์แท็ปเมื่ออินพุตเป็นซีกบวก..... 25
2.18	แสดงการทำงานของวงจรเรียงกระแสแบบเซ็นเตอร์แท็ปเมื่ออินพุตเป็นซีกลบ 25
2.19	วงจรเรียงกระแสแบบบริดจ์ 25
2.20	แสดงการทำงานของวงจรเรียงกระแสแบบเต็มคลื่นแบบบริดจ์เมื่ออินพุตเป็นซีกลบ..... 26
2.21	แสดงการทำงานของวงจรเรียงกระแสเต็มคลื่นแบบบริดจ์เมื่ออินพุตเป็นซีกบวก 26
3.1	ตัวต้านทาน 30วัตต์ 1 โอห์ม 27
3.2	Brushless DC Motor 27
3.3	แคลมป์มอเตอร์ 27
3.4	หม้อแปลง 24 V 27

สารบัญรูป(ต่อ)

รูปที่	หน้า
3.5 Oscilloscope	27
3.6 Power & Harmonics Analyzer	27
3.7 Block Diagram	28
3.8 วงจรที่ใช้ขับให้มอเตอร์ทำงาน	29
3.9 รูปแบบการต่อขาของ IR 2112	30
3.10 ชุดขับเคลื่อนมอเตอร์	30
3.11 วงจรเรียงกระแส	31
3.12 แหล่งจ่ายไฟกระแสตรง	32
3.13 วงจรของ Hall Sensor	32
3.14 ตรวจสอบว่ามี MCU เชื่อมต่ออยู่หรือไม่	35
3.15 เลือกโปรแกรม	35
3.16 การตั้งค่า Configure	36
3.17 เลือกไคลเรททอรีที่เก็บไฟล์ .hex	36
3.18 คำนวณโหลดไฟล์ .hex ลงสู่ MCU	37
3.19 ชุดขับมอเตอร์	37
4.1 การต่อตัว R เพื่อวัดกราฟกระแส	38
4.2 กราฟกระแสจากเฟส A	39
4.3 กราฟกระแสจากเฟส B	39
4.4 กราฟกระแสจากเฟส C	40
4.5 กราฟแรงดันระหว่างเฟส A กับ B	40
4.6 กราฟแรงดันระหว่างเฟส A กับ C	41
4.7 กราฟแรงดันระหว่างเฟส B กับ A	41
4.8 กราฟแรงดันระหว่างเฟส B กับ C	42
4.9 กราฟแรงดันระหว่างเฟส C กับ A	42
4.10 กราฟแรงดันระหว่างเฟส C กับ B	43

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของโครงการ

เป็นที่รู้กัน โดยทั่วไปว่ามอเตอร์เป็นอุปกรณ์ไฟฟ้าที่ใช้กันอย่างกว้างขวางในงานต่างๆ การเลือกใช้มอเตอร์ให้เหมาะสมกับงานนับว่าเป็นเรื่องสำคัญ แต่ที่สำคัญกว่านั้นคือการควบคุมการทำงานของมอเตอร์ให้มีประสิทธิภาพ ปัจจุบันนี้ชุดขับเคลื่อนมอเตอร์ได้ถูกผลิตออกมาวางขายมากมาย แต่ถ้าหากต้องการใช้งานในการควบคุมจำเป็นต้องซื้อชุดควบคุมที่มีราคาแพงมากมาต่อเข้าไป และสำหรับชุดขับเคลื่อนแต่ละชนิดนั้นต้องใช้ชุดควบคุมที่แตกต่างกัน ทำให้การซ่อมบำรุงนั้นยุ่งยาก ถ้าเราสามารถสร้างตัวควบคุมที่มีประสิทธิภาพได้ก็คงจะมีประโยชน์อย่างมาก โครงการนี้จึงเป็นการสร้างวงจรควบคุมมอเตอร์ไฟฟ้ากระแสตรงแบบไร้แปรงถ่าน (Brushless DC Motor)

1.2 วัตถุประสงค์ของโครงการ

1. ต้องการศึกษการทำงานของอุปกรณ์อิเล็กทรอนิกส์ซึ่งจะนำมาสร้างเป็นแผงวงจรควบคุมมอเตอร์ไฟฟ้ากระแสตรงแบบไร้แปรงถ่าน
2. ต้องการสร้างวงจรควบคุมการทำงานของมอเตอร์ไฟฟ้ากระแสตรงแบบไร้แปรงถ่าน (Brushless DC Motor) และสามารถสั่งการให้มอเตอร์ทำงานได้
3. สร้างวงจรควบคุม Brushless DC Motor ที่มีราคาไม่แพง

1.3 แนวทางการดำเนินโครงการ

1. หาข้อมูลและทำการศึกษาวงจรและอุปกรณ์อิเล็กทรอนิกส์ในส่วนที่จะนำมาทำโครงการ
2. ศึกษาคุณสมบัติของ Brushless DC Motor
3. ทำการสร้างแผงวงจรอิเล็กทรอนิกส์ที่จะนำมาควบคุม Brushless DC Motor
4. ทดสอบการทำงานของวงจรควบคุมมอเตอร์

1.7 งบประมาณ

ค่าอุปกรณ์อิเล็กทรอนิกส์	3,500	บาท
ค่าจ้างแกะลายแผงวงจร	700	บาท
ค่าจัดทำรายงาน	<u>500</u>	บาท
รวมเป็นเงินทั้งสิ้น	<u>4,700</u>	บาท (สี่พันเจ็ดร้อยบาทถ้วน)

หมายเหตุ ขออนุมัติตัวเฉลี่ยทุกรายการ



บทที่ 2

หลักการและทฤษฎีเบื้องต้น

2.1 มอเตอร์ไฟฟ้ากระแสตรง

มอเตอร์ไฟฟ้ากระแสตรงและเครื่องกำเนิดไฟฟ้ากระแสตรงนั้นมีส่วนประกอบสำคัญที่เหมือนกันซึ่งถ้าดูจากภายนอกแล้วจะมีลักษณะเหมือนกันทุกประการส่วนที่แตกต่างกันก็คือลักษณะการใช้งานเท่านั้น เครื่องกำเนิดไฟฟ้ากระแสตรงได้รับพลังงานกลจากภายนอกมาขับเคลื่อนมอเตอร์ให้หมุนติดกับสนามแม่เหล็กที่เกิดจากขดลวดฟิวส์ค้อยล์ในเครื่องกำเนิดนั้น ส่วนมอเตอร์ไฟฟ้ากระแสตรงได้รับพลังงานจากภายนอกมาทำให้เกิดสนามแม่เหล็กขึ้นในมอเตอร์นั้นไปขับเคลื่อนขดลวดตัวนำในอานมอเตอร์ ดังนั้นเพลลาของมอเตอร์ก็จะหมุน ซึ่งจะได้พลังงานกลออกไปใช้งาน

เมื่อมีขดลวดตัวนำหมุนติดกับสนามแม่เหล็กจะทำให้เกิดแรงดันไฟฟ้าเหนี่ยวนำขึ้นในขดลวดตัวนำนั้นและทิศทางการไหลของกระแสในขดลวดตัวนำจะไหลในทิศทางที่ต้านการหมุนของขดลวดตัวนำนั้น ซึ่งกฎในเรื่องทิศทางการไหลของกระแสไฟฟ้านี้เรียกว่ากฎของเลนซ์ เซอร์เฟลมมิ่งได้ค้นพบวิธีพิจารณาว่าการหมุนของมอเตอร์จะหมุนไปในทางใดนั้น ถ้าทราบทิศทางการไหลของกระแสไฟฟ้า ก็จะพบความสัมพันธ์ดังนี้คือ ถ้าทราบทิศทางของเส้นแรงแม่เหล็ก ทิศทางการไหลของกระแสไฟฟ้าในตัวนำจะทำให้ทราบทิศทางการหมุนของขดลวดตัวนำได้ การค้นพบนี้เรียกว่ากฎมือซ้ายของเฟลมมิ่งซึ่งกระแสที่ไหลในขดลวดตัวนำจะสร้างสนามแม่เหล็กและขดลวดตัวนำจะอยู่ห่างจากศูนย์กลางการหมุน ซึ่งจะได้ความสัมพันธ์ระหว่างแรงบิดของเพลลาและกระแส คือ

$$T = k\phi I \quad (2.1)$$

เมื่อ	T	คือแรงบิดของเพลลา (นิวตัน/เมตร)
	ϕ	คือเส้นแรงแม่เหล็ก (เวเบอร์)
	I	คือกระแส (แอมแปร์)
	k	คือค่าคงที่มอเตอร์ (ตามโครงสร้างการออกแบบมอเตอร์)

ดังนั้นแรงบิดของเพลลาจะเป็นสัดส่วนโดยตรงกับผลคูณเส้นแรงแม่เหล็กและกระแส เมื่อขดลวดตัวนำเคลื่อนที่ในสนามแม่เหล็กก็จะทำให้เกิดแรงดันคคร่อมตัวมันเอง แรงดันนี้จะเป็นสัดส่วนกับความเร็วของเพลลามอเตอร์และด้านการไหลของกระแส จะได้ความสัมพันธ์ระหว่างแรงดันย้อนกลับและความเร็วของเพลลามอเตอร์

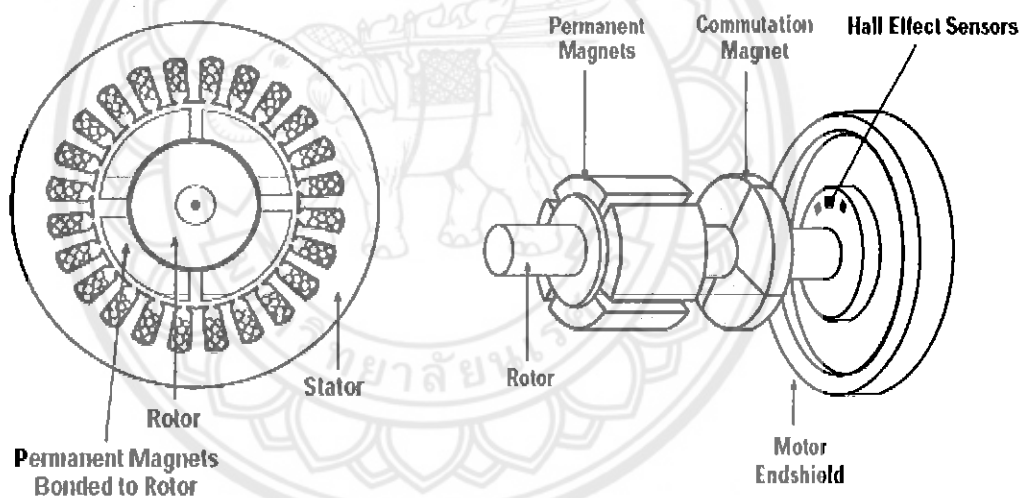
$$E = k\phi\omega \quad (2.2)$$

เมื่อ	E	คือแรงดันย้อนกลับ (back emf) (โวลต์)
	ϕ	คือเส้นแรงแม่เหล็ก (เวเบอร์)
	ω	คือความเร็วเชิงมุมของมอเตอร์ (เรเดียน/วินาที)

2.1.1 มอเตอร์ไฟฟ้ากระแสตรงแบบไม่มีแปรงถ่าน

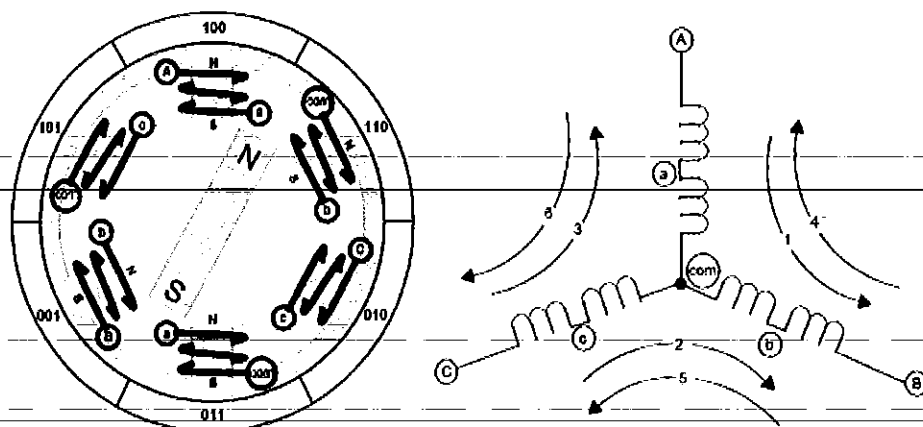
มอเตอร์ไฟฟ้ากระแสตรงแบบไม่มีแปรงถ่าน (Brushless DC หรือ BLDC Motor)

มอเตอร์ไฟฟ้ากระแสตรงแบบไม่มีแปรงถ่าน ประกอบไปด้วย แกนหมุนแม่เหล็กถาวร และขดลวดที่พันอยู่ที่ขั้วของสเตเตอร์ดังแสดงในรูปที่ 2.1(ก) พลังงานไฟฟ้าที่จ่ายให้มอเตอร์จะถูกเปลี่ยนเป็นพลังงานกล โดยแรงดึงดูดระหว่างแกนหมุนแม่เหล็กถาวรและสนามแม่เหล็กที่เกิดจากขดลวดที่พันอยู่รอบขั้วที่สเตเตอร์



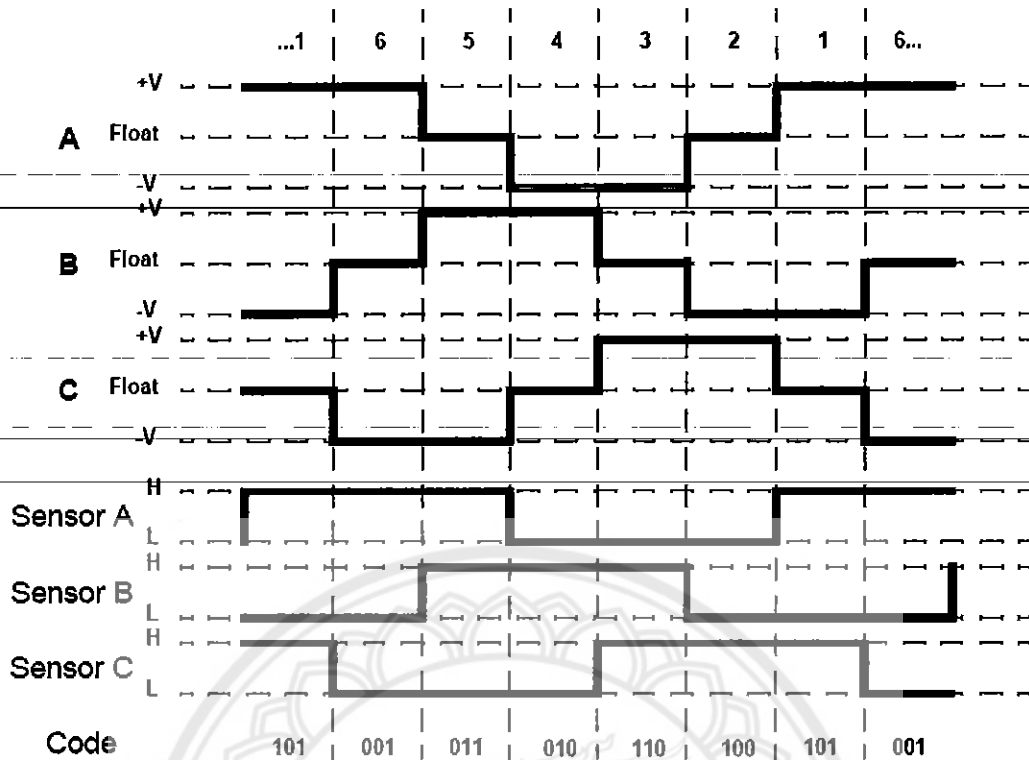
รูปที่ 2.1 โค้ดแกรมของมอเตอร์แบบ BLDC [1]

รูปที่ 2.1 (ก) ส่วนประกอบของมอเตอร์แบบ BLDC



รูปที่ 2.1(ข) โค้ดแกรมการทำงานของมอเตอร์แบบ BLDC

จากรูปที่ 2.1(ข) วงจรสนามแม่เหล็ก 3 วงจรจะต่อเชื่อมกันที่จุด Common (com) โดยแต่ละวงจรจะแยกออกมาจากจุดรวมที่อยู่ตรงกลาง ทำให้แกนหมุนแม่เหล็กถาวรเคลื่อนที่อยู่ระหว่างกลางของสนามแม่เหล็กที่เกิดขึ้น โดยทั่วไป BLDC มอเตอร์มีขดลวด 3 เฟสพันกันซึ่งจะเชื่อมต่อกันแบบ Star(Y) พลังงานไฟฟ้าจะถูกจ่ายครั้งละ 2 เฟสเพื่อขับเคลื่อนมอเตอร์ การไหลของกระแสไฟฟ้ามักมีการจัดเรียงดังรูปที่ 2.2 ซึ่งกระแสไฟฟ้าที่ไหลจากขั้ว A ไป B แสดงถึงเส้นทางที่ 1 ดังในรูปที่ 2.1(ข) แกนหมุนจะถูกทำให้หมุนตามเข็มนาฬิกาไป 60 องศาของการจัดเรียง A ไป B ด้วยการเปลี่ยนให้เส้นทางกระแสไฟฟ้าไหลจาก C ไป B ซึ่งก็คือเส้นทางที่ 2 ในรูปที่ 2.1(ข) การจัดเรียงของแม่เหล็กที่แสดงไว้ในรูปที่ 2.1(ข) เพื่อความสะดวกสำหรับการอธิบายเท่านั้น ในทางปฏิบัติแรงบิดสูงสุดจะเกิดขึ้นเมื่อแกนแม่เหล็กถาวรทำมุม 90 องศากับการจัดเรียงสนามแม่เหล็กที่สเตเตอร์



รูปที่ 2.2 แผนผังของขดลวด , เซ็นเซอร์ เทียบกับเวลา [2]

การทำงานของมอเตอร์แบบ BLDC ขึ้นอยู่กับตำแหน่งของแกนหมุน ซึ่งพลังงานที่ป้อนให้ที่ขดลวดจะทำให้เกิดแรงบิดที่เพียงพอ เมื่อแกนหมุนเคลื่อนที่ไป 60 องศา (ทางไฟฟ้า) ต่อ Commutation 1 สเต็ป เส้นทางกระแสไฟฟ้าที่สเตเตอร์ถูกกระตุ้นเมื่อแกนหมุนทำมุม 120 องศาจากการจัดเรียงที่สอดคล้องกับสนามแม่เหล็กที่สเตเตอร์ และจะไม่ถูกกระตุ้นเมื่อแกนหมุนทำมุม 60 องศากับการจัดเรียง เมื่อวงจรถัดไปถูกกระตุ้นกระบวนการที่กล่าวข้างต้นก็จะเกิดขึ้นซ้ำ จากรูป 2.1 (ข) Commutation สำหรับตำแหน่งของแกนหมุน เมื่อกระแสไฟฟ้าป้อนให้กับเส้นทางที่ 2 เสรีจลัน และมีกระแสไฟฟ้าป้อนไปที่เส้นทางที่ 3 จะทำให้แกนหมุนเคลื่อนที่ตามเข็มนาฬิกา ถ้าหากป้อนกระแสไฟฟ้าไปยังเส้นทางที่ 1 ถึง 6 ดังรูปที่ 2.1(ข) จะทำให้แกนหมุนเคลื่อนที่ครบ 1 รอบ จากรูปที่ 2.1(ข) พบว่า รอบทางไฟฟ้า 1 รอบมีค่าเท่ากับรอบทางกล 1 รอบ แต่ในทางปฏิบัติมอเตอร์แบบ BLDC มีวงจรไฟฟ้ามากกว่า 1 วงจร และมีแกนหมุนแม่เหล็กถาวรหลายขั้ว สำหรับมอเตอร์ที่มีวงจรไฟฟ้า 2 วงจรจะทำให้ 1 รอบทางกลมีค่าเท่ากับรอบทางไฟฟ้า 2 รอบ ซึ่งแต่ละเฟสของ commutation จะครอบคลุม 30 องศาของการเคลื่อนที่ทางกลตำแหน่งของ Commutation สามารถหาด้วยวิธีการที่ง่ายที่สุดคือโดยการใช้ ตัวเซ็นเซอร์วัดตำแหน่ง โดยทั่วไปมอเตอร์แบบ BLDC จะมี Hall effect 3 ตัว เป็นตัวเซ็นเซอร์วัดตำแหน่ง เซ็นเซอร์แต่ละตัวจะมีเอาต์พุตเป็น Digital high level สำหรับ 180 องศาไฟฟ้าต่อรอบทางไฟฟ้าและ Digital low level อีก 180 องศาไฟฟ้าต่อรอบทางไฟฟ้า เซ็นเซอร์ทั้ง 3 ตัวจะวางห่างกันอยู่ 60 องศาไฟฟ้า และแต่ละเซ็นเซอร์จะจัดเรียงตามวงจร

แม่เหล็กไฟฟ้าแต่ละวงจร รูปที่ 2.2 แสดงถึง Timing diagram ซึ่งเป็นความสัมพันธ์ระหว่างสัญญาณเอาต์พุตของเซ็นเซอร์และแรงดันที่ป้อนให้เพื่อขับเคลื่อน ตัวเลขที่แสดงในรูปแสดงถึงเฟสของกระแสดังในรูปที่ 2.1(ข) จากรูปที่ 2.2 พบว่าเอาต์พุตของเซ็นเซอร์ทั้ง 3 ตัวจะซ้อนกันซึ่งทำให้เกิด code แบบ 3 บิตที่มี 6 กรณี โดยแต่ละกรณีจะสัมพันธ์กับแต่ละเฟสขั้ว ตัวเลขที่แสดงรอบ ๆ แผนภาพมอเตอร์แสดงถึง code ของตำแหน่งเซนเซอร์ ทิศเหนือของแกนหมุนชี้ไปยังตำแหน่งของ code คือ เอาต์พุตของตำแหน่งแกนหมุน ตัวเลขที่แสดงเป็น Logic level ของเซ็นเซอร์ โดยบิตที่สำคัญมากที่สุด (Most significant bit) คือ เซ็นเซอร์ C และบิตที่สำคัญน้อยที่สุด (Least significant bit) คือ เซ็นเซอร์ A แต่ละขดลวดขั้วประกอบไปด้วยขั้วมอเตอร์ขั้วแบบ High, Low และ Floating ขดขั้วแต่ละขดจะใช้ขั้วที่ขั้วมอเตอร์แต่ละขั้ว ทั้งนี้จะต้องไม่ขั้วขั้วมอเตอร์แบบ High หรือ Low พร้อมกันสองขั้ว แต่ในปัจจุบันอุปกรณ์คอนเอนโค้ดเดอร์ (Optical encoder) ได้ถูกนำมาใช้สำหรับวัดตำแหน่งในมอเตอร์อย่างแพร่หลาย

2.2 ไมโครคอนโทรลเลอร์ ATMEGA8

ATMEGA8 เป็นไมโครคอนโทรลเลอร์อนุกรม AVR ที่ผลิตโดย Atmel Corporation สหรัฐอเมริกา ที่เลือกใช้ในบอร์ดทดลอง Microcamp เป็นตัวถังแบบ DIP 28 ขา มีการจัดขา ดังรูป 2.3

2.2.1 คุณสมบัติเด่นของ ATMEGA8

1. เป็นไมโครคอนโทรลเลอร์ 8 บิต ในอนุกรม AVR มีสถาปัตยกรรมแบบ Advance RISC มีความเร็วในการทำงานสูง โดยสามารถประมวลผล 1 คำสั่งในเวลา 1 สัญญาณนาฬิกา สามารถทำงานกับความถี่สัญญาณนาฬิกาสูงสุด 16 MHz จึงสามารถประมวลผลคำสั่งได้สูงถึง 16 ล้านคำสั่งต่อวินาที
2. มีหน่วยความจำ 3 แบบ เพื่อรองรับการทำงาน ประกอบด้วย
 - หน่วยความจำโปรแกรมแบบแฟรช ความจุ 8 กิโลไบต์ ที่สามารถรักษาข้อมูลโปรแกรมไว้ได้แม้ไม่มีไฟเลี้ยง และสามารถป้องกันการอ่านได้ สามารถลบ-เขียนข้อมูลใหม่ได้ 10000 รอบ
 - หน่วยความจำข้อมูลอีพรอม ความจุ 512 ไบต์ สามารถรักษาข้อมูลไว้ได้แม้ไม่มีไฟเลี้ยง สามารถลบ-เขียน ข้อมูลใหม่ได้ 10000 รอบ
 - หน่วยความจำข้อมูลแรมความจุ 1 กิโลไบต์ ใช้ประมวลผลหลักร่วมกับซีพียู
3. มีพอร์ตอินพุตเอาต์พุตอิสระที่สามารถโปรแกรมได้ 23 ขา แบ่งเป็น 3 กลุ่มคือ
 - พอร์ต B ใช้งานได้สูงสุด 8 ขา (PB0 ถึง PB7) โดยมี 2 ขาเพื่อต่อกับคริสตอลเพื่อกำเนิดสัญญาณนาฬิกาให้แก่ไมโครคอนโทรลเลอร์
 - พอร์ต C ขา (PC0 ถึง PC6) โดย PC0 ถึง PC5 สามารถใช้เป็นอินพุตอนาลอกได้

- พอร์ต D ขา (PD0 ถึง PD7)

4. มี Timer/Counter ขนาด 8 บิต 2 ชุดที่แยกการทำงานอย่างอิสระ
5. มี Timer/Counter ขนาด 16 บิต 1 ชุด รองรับการทำงานสมบูรณ์แบบตั้งในโหมดตั้งเวลา(timer) คำนับ(counter) เปรียบเทียบสัญญาณ(compare) และตรวจจับสัญญาณ(capture)
6. มีโมดูลกำเนิดสัญญาณ PWM 3 ชุด
7. มีโมดูลแปลงสัญญาณอนาลอกเป็นดิจิตอล(ADC) ความละเอียด 10 บิต จำนวน 6 ช่อง
8. มีอินพุตเปรียบเทียบสัญญาณอนาลอก 2 ช่อง
9. มีโมดูลสื่อสารอนุกรม 2 สาย รองรับการทำงานกับบัส I2C
10. มีโมดูลเชื่อมต่ออุปกรณ์อนุกรมหรือ SPI (Serial Peripheral Interface) ใช้สำหรับการโปรแกรมหน่วยความจำโปรแกรมแบบแฟรชในชิพ โดยปกติแล้วมักสงวนไว้เพื่อการโปรแกรม แต่ก็สามารถนำไปใช้เชื่อมต่ออุปกรณ์บนระบบบัส SPI ได้
11. มีโมดูลสื่อสารข้อมูลอนุกรม UART (Universal Asynchronous Receiver Transmitter)
12. มีวอร์ทซ์ค็อกไทเมอร์เพื่อช่วยในการตรวจสอบการทำงานของระบบ
13. รองรับการอินเตอร์รัพต์ทั้งจากสัญญาณภายนอกและการทำงานของโมดูลต่างๆ ภายในไมโครคอนโทรลเลอร์มีระบบเพาเวอร์-อนรีเซต
14. มีวงจรตรวจจับแรงดันไฟเลี้ยงผิดปกติหรือบราวเอาต์ที่สามารถโปรแกรมได้
15. ไฟเลี้ยง +4.5 ถึง +5.5 V กระแส 3.6 mA แต่ถ้าใช้รุ่น ATMEGA8L สามารถใช้ไฟเลี้ยงในย่านที่กว้างขึ้นจาก +2.7 ถึง +5.5 V แต่สัญญาณนาฬิกาต้องลดลงไม่เกิน 8 MHz
16. กำหนดการทำงานในโหมดหยุดทำงาน (sleep) ได้ 5 โหมด คือ ไอเดิล (Idle) , โหมดลดสัญญาณรบกวนในวงจร ADC , โหมดประหยัดพลังงาน (power-save), โหมดลดพลังงาน (power-down), และ โหมดหยุดรอ(standby)

PDIP

(RESET) PC6	1	28	PC5 (ADC5/SCL)
(RXD) PD0	2	27	PC4 (ADC4/SDA)
(TXD) PD1	3	26	PC3 (ADC3)
(INT0) PD2	4	25	PC2 (ADC2)
(INT1) PD3	5	24	PC1 (ADC1)
(XCK/T0) PD4	6	23	PC0 (ADC0)
VCC	7	22	GND
GND	8	21	AREF
(XTAL1/TOSC1) PB6	9	20	AVCC
(XTAL2/TOSC2) PB7	10	19	PB5 (SCK)
(T1) PD5	11	18	PB4 (MISO)
(AIN0) PD6	12	17	PB3 (MOSI/OC2)
(AIN1) PD7	13	16	PB2 (SS/OC1B)
(ICP1) PB0	14	15	PB1 (OC1A)

รูปที่ 2.3 การจัดขาของ ATMEGA8 [3]

2.2.2 หน้าที่การทำงานของขาสัญญาณทั้งหมดของ ATMEGA8

ตารางที่ 2.1 หน้าที่การทำงานของขาสัญญาณทั้งหมดของ ATMEGA8

ชื่อขา	ตำแหน่งขา	ชนิดของขา	รายละเอียดการทำงาน
- Vcc	7	อินพุต	ขาต่อไฟเลี้ยงบวกตั้งแต่ 4.5 – 5.5 V
GND	8,22	อินพุต	ขาต่อกราวด์
AVcc	20	อินพุต	ขาต่อไฟเลี้ยง +5 V แก่โมดูล ADC ภายใน ATMEGA8
AREF	21	อินพุต	ขาต่อแรงดันอ้างอิงให้แก่โมดูล ADC ภายใน ATMEGA8
ขาพอร์ต B เป็นขาพอร์ต 2 ทิศทาง			
ชื่อขา	ตำแหน่งขา	ชนิดของขา	รายละเอียดการทำงาน
PB0 ICP1	14	อินพุต/เอาต์พุต อินพุต	- ขาพอร์ตดิจิทัล PB0 - อินพุต โมดูลตรวจจับสัญญาณ ชุด 1
PB1 OC1A	15	อินพุต/เอาต์พุต เอาต์พุต	- ขาพอร์ตดิจิทัล PB3 - เอาต์พุตวงจรเปรียบเทียบสัญญาณ PWM ช่อง 1A
PB2 OC1B SS	16	อินพุต/เอาต์พุต เอาต์พุต อินพุต	- ขาพอร์ตดิจิทัล PB3 - เอาต์พุตวงจรเปรียบเทียบสัญญาณ PWM ช่อง 1B - อินพุตเลือกอุปกรณ์สเลฟเมื่อทำงานเป็นอุปกรณ์สเลฟในระบบ SPI ใช้ในกระบวนการ SPI
PB3 OC2 MOSI	17	อินพุต/เอาต์พุต เอาต์พุต อินพุต/เอาต์พุต	- ขาพอร์ตดิจิทัล PB3 - เอาต์พุตวงจรเปรียบเทียบสัญญาณ PWM ช่อง 1B - อินพุตรับข้อมูลเมื่อทำงานเป็นอุปกรณ์สเลฟในระบบบัส SPI
			- เอาต์พุตส่งข้อมูลเมื่อทำงานเป็นอุปกรณ์มาสเตอร์ในระบบบัส SPI , ใช้ในกระบวนการ SPI

ตารางที่ 2.1 (ต่อ)

ขาพอร์ต B เป็นขาพอร์ต 2 ทิศทาง (ต่อ)			
ชื่อขา	ตำแหน่ง	ชนิดของขา	รายละเอียดการทำงาน
	ขา		
PB4 MISO	18	อินพุต/เอาต์พุต อินพุต/เอาต์พุต	- ขาพอร์ตคิจิตอล PB4 - อินพุตรับข้อมูลเมื่อทำงานเป็นอุปกรณ์มาสเตอร์ในระบบบัส SPI , ใช้ในกระบวนการ SPI - เอาต์พุตส่งข้อมูลเมื่อทำงานเป็นอุปกรณ์สเลฟ
PB5 SCK	19	อินพุต/เอาต์พุต อินพุต/เอาต์พุต	- ขาพอร์ตคิจิตอล PB5 - อินพุตรับสัญญาณนาฬิกาเมื่อทำงานเป็นอุปกรณ์สเลฟในระบบบัส SPI - เอาต์พุตส่งสัญญาณนาฬิกาเมื่อทำงานเป็นอุปกรณ์มาสเตอร์ในระบบบัส SPI , ใช้ในกระบวนการ SPI
PB6 XTAL1 TOSC1	9	อินพุต/เอาต์พุต อินพุต อินพุต	- ขาพอร์ตคิจิตอล PB6 เมื่อเลือกทำงานกับสัญญาณนาฬิกาภายใน - อินพุตสัญญาณนาฬิกาภายนอก,ต่อกับคริสตอล - ไม่ใช่ใช้งานเมื่อเลือกทำงานกับสัญญาณนาฬิกาภายใน
PB7 XTAL2 TOSC2	10	อินพุต/เอาต์พุต อินพุต เอาต์พุต	- ขาพอร์ตคิจิตอล PB7 เมื่อเลือกทำงานกับสัญญาณนาฬิกาภายใน - ใช้ต่อกับคริสตอลหรือเซรามิกเรโซเนเตอร์ - เอาต์พุตสัญญาณนาฬิกาเมื่อใช้สัญญาณนาฬิกาภายใน
ขาพอร์ต C เป็นขาพอร์ต 2 ทิศทาง			
ชื่อขา	ตำแหน่ง	ชนิดของขา	รายละเอียดการทำงาน
	ขา		
PC0 ADC0	23 23	อินพุต/เอาต์พุต อินพุต	- ขาพอร์ตคิจิตอล PC0 - อินพุตอนาลอกโมดูลแปลงสัญญาณอนาลอกเป็นคิจิตอล ช่อง0
PC1 ADC1	24	อินพุต/เอาต์พุต อินพุต	- ขาพอร์ตคิจิตอล PC1 - อินพุตอนาลอกโมดูลแปลงสัญญาณอนาลอกเป็นคิจิตอล ช่อง1

ตารางที่ 2.1 (ต่อ)

ขาพอร์ต C เป็นขาพอร์ต 2 ทิศทาง(ต่อ)			
ชื่อขา	ตำแหน่ง	ชนิดของขา	รายละเอียดการทำงาน
	ขา		
PC2 ADC2	25	อินพุต/เอาต์พุต อินพุต	- ขาพอร์ตคิจิตอล PC2 - อินพุตอนาลอกโมดูลแปลงสัญญาณอนาลอกเป็นคิจิตอล ช่อง2
PC3 ADC3	26	อินพุต/เอาต์พุต อินพุต	- ขาพอร์ตคิจิตอล PC3 - อินพุตอนาลอกโมดูลแปลงสัญญาณอนาลอกเป็นคิจิตอล ช่อง3
PC4 ADC0 SDA	27	อินพุต/เอาต์พุต อินพุต เอาต์พุต	- ขาพอร์ตคิจิตอล PC4 - อินพุตอนาลอกโมดูลแปลงสัญญาณอนาลอกเป็นคิจิตอล ช่อง 4 - ขาข้อมูลอนุกรมสำหรับระบบบัส 2 สาย
PC5 ADC5 SCL	28	อินพุต/เอาต์พุต อินพุต เอาต์พุต	- ขาพอร์ตคิจิตอล PC5 - อินพุตอนาลอกโมดูลแปลงสัญญาณอนาลอกเป็นคิจิตอล ช่อง 5 - ขาเอาต์พุตสัญญาณนาฬิกาอนุกรมสำหรับระบบบัส 2 สาย
PC6 RESET	1	อินพุต/เอาต์พุต อินพุต	- ขาพอร์ตคิจิตอล PC6 - ขาอินพุตสัญญาณรีเซต
ขาพอร์ต D เป็นขาพอร์ต 2 ทิศทาง			
ชื่อขา	ตำแหน่ง	ชนิดของขา	รายละเอียดการทำงาน
	ขา		
PD0 RxD	23	อินพุต/เอาต์พุต อินพุต	- ขาพอร์ตคิจิตอล PD0 - อินพุตรับข้อมูลของโมดูลสื่อสารข้อมูลพอร์ตอนุกรมUSART
PD1 TxD	24	อินพุต/เอาต์พุต อินพุต	- ขาพอร์ตคิจิตอล PD1 - เอาต์พุตส่งข้อมูลของโมดูลสื่อสารข้อมูลพอร์ตอนุกรม USART

ตารางที่ 2.1 (ต่อ)

ขาพอร์ต D เป็นขาพอร์ต 2 ทิศทาง (ต่อ)			
ชื่อขา	ตำแหน่ง	ชนิดของขา	รายละเอียดการทำงาน
	ข1		
PD2 INT0	25	อินพุต/เอาต์พุต อินพุต	- ขาพอร์ตคิจิตอล PD2 - อินพุตสัญญาณอินเทอร์รัพต์จากภายนอกช่อง 0
PD3 INT1	26	อินพุต/เอาต์พุต อินพุต	- ขาพอร์ตคิจิตอล PD3 - อินพุตสัญญาณอินเทอร์รัพต์จากภายนอกช่อง 1
PD4 XCK T0	27	อินพุต/เอาต์พุต อินพุต/เอาต์พุต อินพุต	- ขาพอร์ตคิจิตอล PD4 - ขาสัญญาณภายนอกของ โมดูลสื่อสารข้อมูลอนุกรม USART - อินพุตรับสัญญาณจากภายนอกสำหรับ โมดูล ไทเมอร์ 0
PD5 T1	11	อินพุต/เอาต์พุต อินพุต	- ขาพอร์ตคิจิตอล PD5 - อินพุตรับสัญญาณจากภายนอกสำหรับ โมดูล ไทเมอร์ 1
PD6 AIN0	12	อินพุต/เอาต์พุต อินพุต	- ขาพอร์ตคิจิตอล PD6 - อินพุตสำหรับวงจรเปรียบเทียบแรงดันอนาล็อก ช่อง 2
PD7 AIN1	13	อินพุต/เอาต์พุต อินพุต	- ขาพอร์ตคิจิตอล PD7 - อินพุตสำหรับวงจรเปรียบเทียบแรงดันอนาล็อก ช่อง 1

2.2.3 รายละเอียดทั่วไปของ ATMEGA8

การทำงานภายในทั้งหมดของไมโครคอนโทรลเลอร์ ATMEGA8 หัวใจสำคัญคือ AVR ซีพียูที่ทรงประสิทธิภาพผนวกเข้ากับ โมดูลเชื่อมต่ออุปกรณ์พิเศษที่มีอย่างสมบูรณ์อย่างไม่น่าเชื่อสำหรับไมโครคอนโทรลเลอร์ที่มีขนาดใช้งานเพียง 28 ขา ส่วนที่ต้องให้ความสนใจเป็นพิเศษคือขา PB6 และขา PB7 ซึ่งทำงานได้ทั้งขาของพอร์ตอินพุตและพอร์ตเอาต์พุตคิจิตอลและขาติดต่อกับวงจรถูกกำเนิดสัญญาณนาฬิกาจากภายนอก หากเลือกใช้วงจรถูกกำเนิดสัญญาณนาฬิกาภายในด้วยตัวไมโครคอนโทรลเลอร์เองจะสามารถกำหนดให้ขาพอร์ตทั้งสองนั้นเป็นพอร์ตอินพุต พอร์ตเอาต์พุตคิจิตอลได้ จึงมีพอร์ตเพิ่มขึ้นอีก 2 ขา แต่ว่าเพื่อให้การทำงานมีความเที่ยงตรงด้านจังหวะเวลาสูง

โดยเฉพาะอย่างยิ่งหากมีการเชื่อมต่อคอมพิวเตอร์เพื่อแลกเปลี่ยนข้อมูลระหว่างกันจึงมักจะเลือกใช้ขาทั้งสองนี้ในการต่อกับคริสตอลหรือเซรามิก โซเนเตอร์เพื่อรับสัญญาณนาฬิกาจากภายนอกมากกว่า

ATMEGA8 มีหน่วยความจำโปรแกรมแบบแฟรชที่สามารถโปรแกรมได้ด้วยกระบวนการโปรแกรมในวงจรหรือในระบบ (In-System Programming : ISP) ความจุ 8 กิโลไบต์ และสามารถอ่านข้อมูลในขณะที่เขียนได้ด้วย มีหน่วยความจำข้อมูลอีอีพรอมความจุสูงถึง 512 ไบต์ และหน่วยความจำสแตติกแรมมากถึง 1 กิโลไบต์ นับเป็นคุณสมบัติเด่นที่หาได้ยากในไมโครคอนโทรลเลอร์ในระดับเดียวกัน ส่งผลให้ ATMEGA8 สามารถรองรับโปรแกรมที่พัฒนาด้วยภาษา C ได้ดีพอสมควร นอกจากนี้ยังมีรีจิสเตอร์สำหรับใช้งานอีก 32 ตัว มี ไทมเมอร์ / เคาน์เตอร์ที่ทำงานอิสระต่อกันอีก 3 ตัว สามารถกำเนิดสัญญาณ PWM มากถึง 3 ช่อง ส่วนการจัดการอินเตอร์รัพต์ทั้งจากภายในและภายนอก ส่วนติดต่ออุปกรณ์ระบบบัส 2 สาย มีโมดูลแปลงสัญญาณอนาล็อกเป็นดิจิทัลที่มีมากถึง 6 ช่อง ความละเอียดในการแปลงสัญญาณ 10 บิต มีวอตช์ดีอก ไทมเมอร์ในตัว รวมทั้งวงจรตรวจจับไฟเลี้ยงผิดปกติหรือบราวเอาต์ (Brown – out detection) ที่สามารถกำหนดระดับแรงดันได้

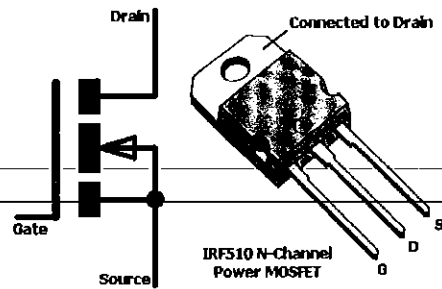
2.3 ทรานซิสเตอร์สนามไฟฟ้า (Field - Effect Transistor ; FET)

ทรานซิสเตอร์สนามไฟฟ้า คือ ทรานซิสเตอร์ที่ใช้สนามไฟฟ้าในการเปลี่ยนแปลงสภาพของสารกึ่งตัวนำเพื่อให้เกิดการนำกระแสเมื่อ ได้รับแรงดันไฟฟ้าที่เหมาะสม โดยทั่วไปมักเรียกชื่อย่อว่า “ เฟท ”

เฟทแบ่งออกเป็น 2 กลุ่มคือ ทรานซิสเตอร์สนามไฟฟ้าแบบรอยต่อ (Junction Field Effect Transister) หรือ เจเฟท (JFET) และทรานซิสเตอร์สนามไฟฟ้าแบบโลหะออกไซด์สารกึ่งนำ (Metal –Oxide–Semiconductor Field Transister) หรือมอสเฟท (MOSFET) ในที่นี้จะกล่าวถึง MOSFET แบบ n-Channel เท่านั้น

2.3.1 มอสเฟท (MOSFET)

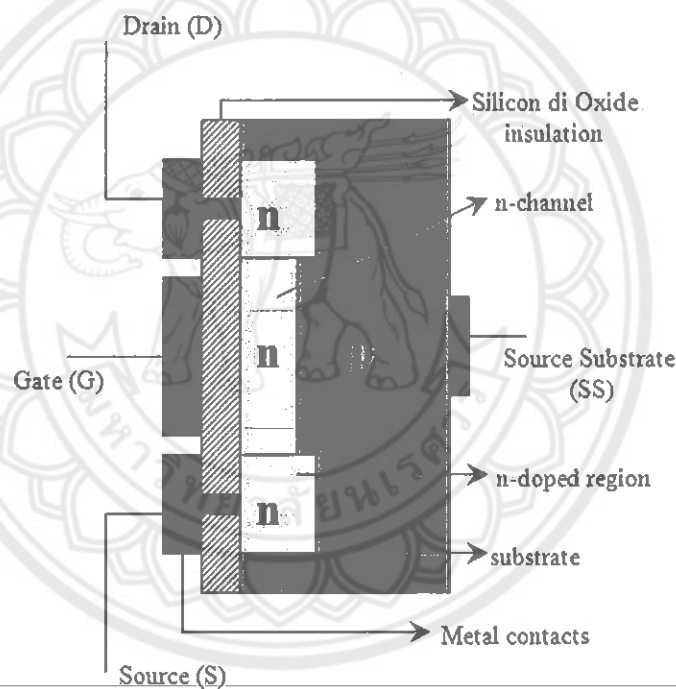
มอสเฟทจะแบ่งออกเป็น 2 ชนิด คือ ดีพลีชันและเอนฮานซ์เมนต์ แต่ละประเภทยังแบ่งออกเป็น 2 แบบ คือ แบบแซนแนล n และแบบแซนแนล p มอสเฟทประเภท ดีพลีชันหรือดีมอสเฟท (D-MOSFET) ทั้ง 2 แบบจะทำงานได้ 2 โหมด คือ โหมดดีพลีชัน (Depletion Mode) และโหมดเอนฮานซ์เมนต์ (Enhancement Mode) กล่าวคือ ถ้าจ่ายแรงดันลบให้กับดีมอสเฟทแซนแนล n จะทำงานในโหมดดีพลีชัน แต่ถ้าจ่ายแรงดันบวกจะทำงานในโหมดเอนฮานซ์เมนต์ ส่วนดีมอสเฟทแซนแนล p ก็ทำงานคล้ายกันเมื่อ ได้รับแรงดันที่มีขั้วตรงข้ามกับแบบแซนแนล n ส่วนมอสเฟทประเภทเอนฮานซ์เมนต์หรืออีมอสเฟท (E-MOSFET) มีโครงสร้างบางอย่างคล้ายกับมอสเฟทแบบดีพลีชัน แต่จะทำงานได้เฉพาะ โหมดเอนฮานซ์เมนต์เท่านั้น



รูปที่ 2.4 มอสเฟตแบบ n แชนแนลและสัญลักษณ์ [4]

2.3.2 คีมอสเฟตแบบแชนแนล n

โครงสร้างของคีมอสเฟตแบบแชนแนล n เป็นคังรูป



รูปที่ 2.5 โครงสร้างคีมอสเฟตแชนแนล n [1]

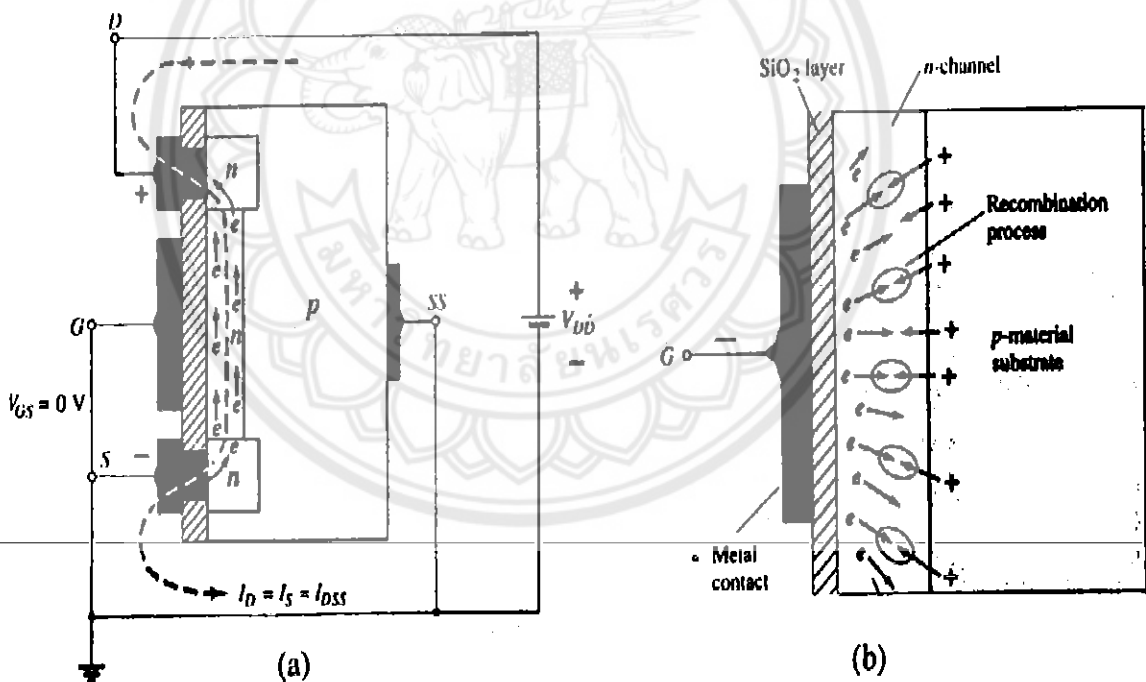
คีมอสเฟตแบบแชนแนล n ประกอบขึ้นจากแผ่นผลึกฐาน p (p-substrate) ที่เป็นสารกึ่งตัวนำทำจากซิลิกอนซั้ว D และซั้ว S ต่อกับบริเวณที่มีการกระตุ้นหรือได้ปให้เป็นบริเวณสารกึ่งตัวนำ n (n-doped region) ทั้งสองส่วนนี้จะเชื่อมกับแชนแนล n สำหรับซั้ว G จะต่อกับวัสดุฉนวนที่เป็นโลหะโดยมีซิลิกอนไดออกไซด์ (SiO_2) กั้นแชนแนล n กับซั้ว G (ซิลิกอนไดออกไซด์เป็นฉนวนประเภทไดอิเล็กทริก) เมื่อมีสนามไฟฟ้าจ่ายเข้ามาที่ชั้นของ SiO_2 ก็จะสร้างสนามไฟฟ้าด้านและสร้างชั้นฉนวนขึ้นภายในตัวเองเพื่อกั้นซั้วเกตกับแชนแนล แสดงว่า ไม่มีการต่อโดยตรงระหว่างซั้ว G กับแชนแนลของมอสเฟต ชั้นที่เป็นฉนวน SiO_2 จะทำให้ Z_1 มีค่าสูงตามความต้องการ

นอกจากนี้บางครั้งจะต่อแผ่นผลึกจากฐานเข้ากับแหล่งจ่ายจึงมีขั้วเพิ่มขึ้นมาเรียกว่า ขั้วผลึกฐาน SS (Substrate : SS) ทำให้มีขั้วเพิ่มเป็น 4 ขั้ว และจากข้างต้น จึงสรุปความหมายของคำว่า MOS ในชื่อ มอสเฟต (ทรานซิสเตอร์สนามไฟฟ้าโลหะออกไซด์สารกึ่งตัวนำ) ได้ดังนี้

- โลหะ (Metal, M) หมายถึง บริเวณสำหรับคาร์ต่อขั้ว D, S และ G กับวัสดุฉนวนออกไซด์ (Oxide, O) หมายถึง ซิลิคอนไดออกไซด์ (SiO_2)
- สารกึ่งตัวนำ (Semiconductor, S) หมายถึง โครงสร้างพื้นฐานบริเวณแพร่กระจายของสารกึ่งตัวนำชนิด p และสารกึ่งตัวนำชนิด n

2.3.3 การทำงานและคุณสมบัติเบื้องต้น (Basic Operation and Characteristics)

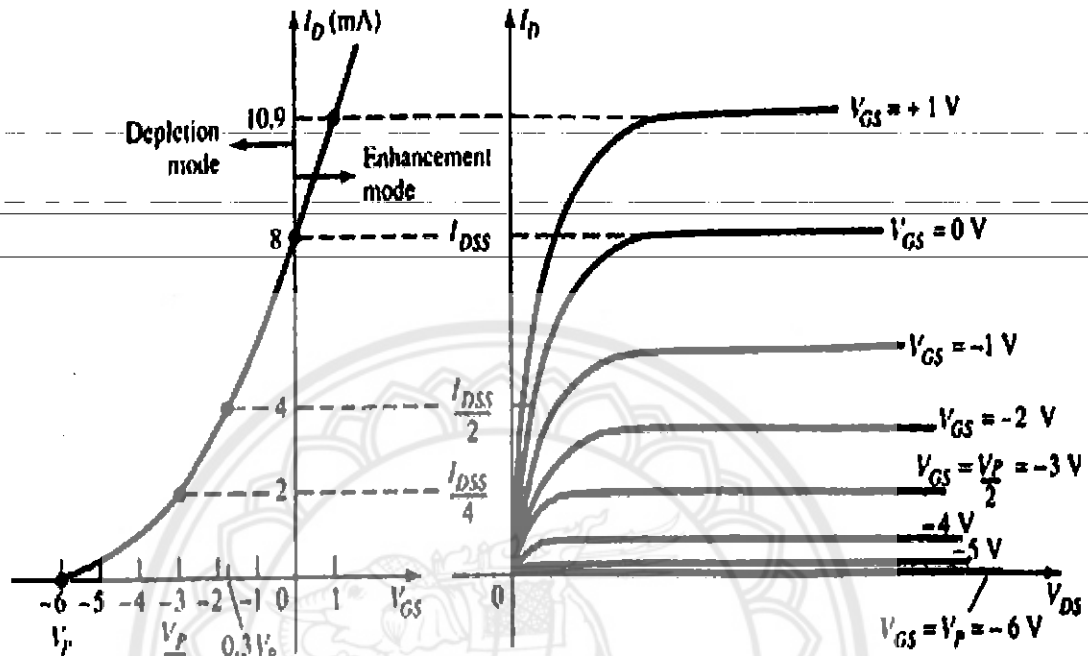
กำหนดให้ V_{GS} ในรูป (a) มีค่าเป็นศูนย์ แล้วจ่าย V_D ที่ขั้ว D และ S ขั้ว D สามารถดึงดูดอิเล็กตรอนอิสระ(e) ผ่านแชนแนล n และทำให้กระแส $I_D = I_S = I_{DSS}$ ไหลผ่านแชนแนล n ได้ (คล้ายกับการไหลของกระแสไฟฟ้าในแชนแนลของเจฟเฟทขณะ $V_{GS} = 0 \text{ V}$)



รูปที่ 2.6 การทำงานของมอสเฟตแบบ n แชนแนล [5]

ถ้าจ่าย V_{GS} ที่มีค่าเป็นลบให้กับขั้วเกต (รูป b) เช่น -1 V ความต่างศักย์ที่ขั้วเกตจะผลักดันให้อิเล็กตรอนอิสระเคลื่อนไปยังแผ่นผลึกฐาน p และดึงดูดโฮลจากแผ่นผลึกฐาน p ทำให้อิเล็กตรอนและโฮลรวมตัวกันใหม่ (Recombination Process) จึงเกิดการลดจำนวนอิเล็กตรอนอิสระในแชนแนล n ที่มีไว้สำหรับการนำกระแส เมื่อมีค่า V_{GS} เป็นลบมากเท่าใดก็จะเกิดการรวมตัวกัน

ใหม่มากขึ้นเท่านั้นและอิเล็กตรอนอิสระที่แกนแนล n ก็จะมีจำนวนลดลง จึงกล่าวได้ว่าถ้า V_{GS} เป็นลบมากขึ้น I_D จะมีค่าน้อยลง เขียนเป็นกราฟคุณลักษณะได้ดังรูปต่อไปนี้ การทำงานขณะ V_{GS} เป็นลบนี้ เราเรียกว่า การทำงานในโหมดคิพลิ้ง



รูปที่ 2.7 กราฟแสดงคุณลักษณะการทำงานของมอสเฟต n แชนแนล [5]

ถ้าจ่าย V_{GS} ที่มีค่าเป็นบวกให้กับขั้วเกตความต่างศักย์ที่ขั้วเกตจะดึงดูดอิเล็กตรอนจากผลิตภัณฑ์ p มายังบริเวณชั้น SiO_2 ทำให้พาหะนำกระแสและสภาพนำกระแสของแกนแนลเพิ่มขึ้น ดังนั้น กระแส I_D จึงเพิ่มขึ้นจนมีค่ามากกว่า I_{DSS} การทำงานขณะ V_{GS} เป็นบวกนี้เราเรียกว่า การทำงานใน โหมดเอนฮานซ์เมนต์

2.4 อินเวอร์เตอร์ 3 เฟส

อินเวอร์เตอร์ คือ วงจรแปลงผันพลังงานไฟฟ้ากระแสตรงเป็นกระแสสลับที่สามารถปรับค่าความถี่เอาต์พุตได้ โดยใช้หลักการปรับค่าแรงดันกระแสตรงด้านอินพุต หรือใช้หลักการควบคุมเทคนิคการสวิตช์ภายในตัวอินเวอร์เตอร์ เช่น ใช้เทคนิคการมอดูเลตตามความกว้างพัลส์ (PWM) เป็นต้น หรือใช้การปรับค่าแรงดันกระแสตรงด้านขาเข้าก็ได้เช่นกัน

2.4.1 วงจรและรูปคลื่นสัญญาณ

อินเวอร์เตอร์ 3 เฟสเหมาะสำหรับกรณีที่ต้องการกำลังไฟฟ้าสูง ซึ่งโครงสร้าง และรูปคลื่นสัญญาณของวงจรอินเวอร์เตอร์ 3 เฟสแสดงอยู่ในรูปที่ 2.8 (ก)

เทคนิค PWM สามารถใช้ได้กับวงจรอินเวอร์เตอร์ 3 เฟส ไม่ว่าจะเป็นเทคนิคไซน์ตัดสามเหลี่ยม หรือเทคนิคการกำจัดฮาร์มอนิกส์ต่ำ หรือการควบคุมรูปคลื่นของกระแส

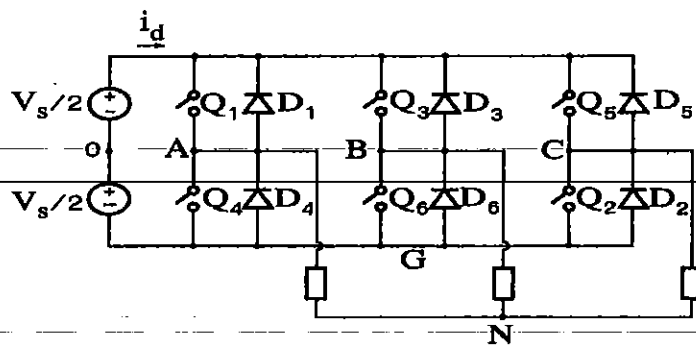
รูปที่ 2.8(ข) แสดงหลักการของเทคนิคไซน์ตัดสามเหลี่ยม เรามีรูปสามเหลี่ยม V_T ซึ่งมีความถี่เท่ากับความถี่การสวิตช์ และมีสัญญาณอ้างอิงรูปไซน์ 3 สัญญาณได้แก่คือ V_{IA} , V_{IB} และ V_{IC} ซึ่งมีมุมเฟสต่างกัน 120° โดยมีจุดตัดระหว่าง V_T กับ V_{IA} (V_{IB} และ V_{IC}) เป็นตัวกำหนดการตัดต่อวงจรของสวิตช์ Q_1 , Q_4 (Q_3 , Q_6 และ Q_5 , Q_2 ตามลำดับ) รูปคลื่นของแรงดัน V_{AG} , V_{BG} (G คือสายลบบของแหล่งแรงดันไฟตรง) และ V_{AB} แสดงอยู่ในรูปที่ 2.8(ค) สังเกตได้ว่าองค์ประกอบหลักมูลของ V_{AG} หรือ V_{AG1} มีแอมพลิจูดเท่ากับค่าที่แสดงในสมการที่(2.3) (V_{AG} ต่างกับ V_{AO} เพียงค่าแรงดันไฟตรง $V_s/2$)

$$V_{AG1} = m_a V_s \quad ; m_a \leq 0.5 \quad (2.3)$$

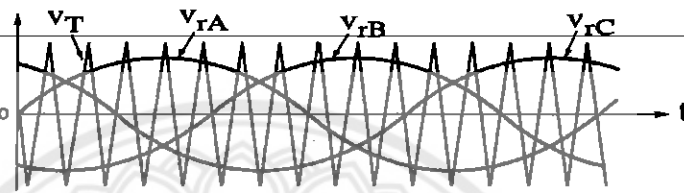
แรงดันสาย V_{AB} มีองค์ประกอบหลักมูลเท่ากับ V_{AB1} แอมพลิจูดของ V_{AB1} เท่ากับ $\sqrt{3}$ ของแอมพลิจูดของ V_{AG1} เนื่องจาก V_{AG} และ V_{BG} มีเฟสต่างกัน 120° และ V_{AB1} มีเฟสล่าหลัง V_{AG1} อยู่ 30° (แสดงในรูปที่ 2.8(ค)) ดังนั้น

$$V_{AB1} = \sqrt{3} m_a V_s \quad ; m_a \leq 0.5 \quad (2.4)$$

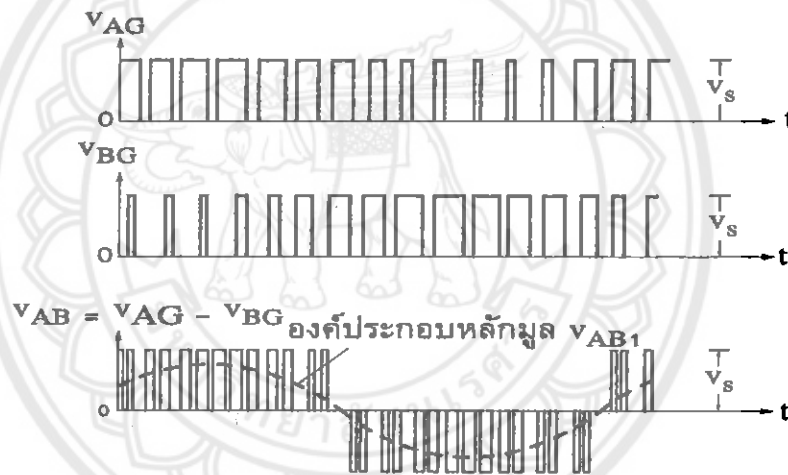
ในการออกแบบระบบ โดยทั่วไปนิยมเลือกอัตราการผลิตความถี่ m_f เป็นเลขคี่และเป็นพหุคูณของ 3 และให้ความลาดชันของ V_T และ V_r ณ จุดที่แรงดันทั้งสองมีค่าเป็นศูนย์มีเครื่องหมายตรงกันข้ามกัน(แสดงในรูปที่ 2.8 (ข)) ในกรณีนี้ถ้า $m_a < 0.5$ สเปกตรัมของ V_{AB} จะเป็นดังแสดงในรูปที่ 2.8 (ง) นอกจากนี้สังเกตได้ว่าฮาร์มอนิกส์ต่ำสุดเกิดขึ้นที่ความถี่ $(m_f \pm 2k)f_s$ ตารางที่ 2.2 แสดง ฮาร์มอนิกส์ของแรงดันสายของอินเวอร์เตอร์ 3-เฟส สำหรับอินเวอร์เตอร์ 3-เฟส เราอาจจะออกแบบให้วงจรทำงานถึงในย่านการมอดูเลตเกิน เพื่อให้ได้แรงดันเอาต์พุตที่มีค่าสูงขึ้น ส่วนข้อเสียที่เนื่องมาแต่ฮาร์มอนิกส์ที่เพิ่มขึ้นมักจะพอยอมรับได้



(ก) วงจร

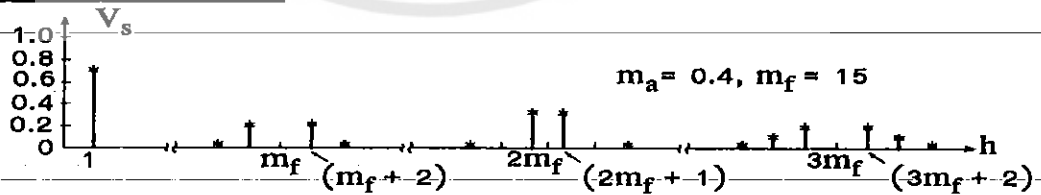


(ข) ไซน์โวลต์สามเหลี่ยม



(ค) รูปคลื่น

ฮาร์มอนิกของแรงดันสาย



(ง) สเปกตรัม

รูปที่ 2.8 วงจรและรูปคลื่นสัญญาณของอินเวอร์เตอร์ 3 เฟสแบบ PWM [6]

เมื่อพิจารณารูปคลื่นสัญญาณในรูปที่ 2.8 จะพบว่าในบางช่วงเวลาดังแสดงในรูปล่าง สวิตช์จะต่อถึงทั้งสามของโหลดเข้ากับขั้วลบของแหล่งจ่าย V_s (เช่น เมื่อ Q_4, Q_6, Q_2 นำกระแส) และบางช่วงเวลาสวิตช์จะต่อถึงทั้งสามของโหลดเข้ากับขั้วบวกของแหล่ง V_s (เช่น เมื่อ Q_1, Q_3, Q_6

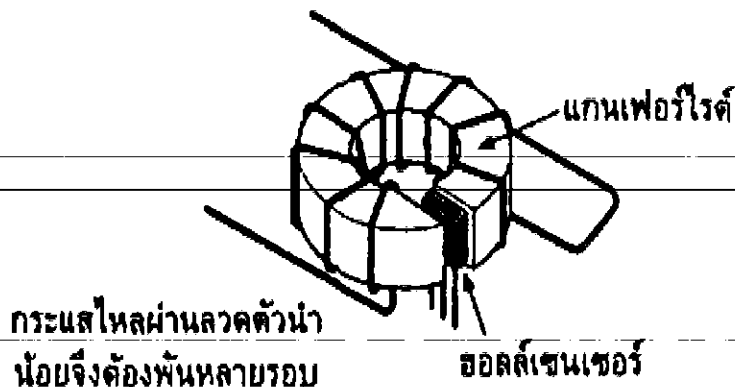
นำกระแส) ซึ่งในช่วงเวลาดังกล่าว กระแสอินพุตของอินเวอร์เตอร์ (i_d ในรูปที่ 2.8(ก)) มีค่าเป็นศูนย์ หมายความว่ากำลังงานไม่ไหลจากแหล่งจ่ายสู่โหลด ซึ่งระยะเวลาของการควบคุมช่วงเวลาดังกล่าวเป็นการควบคุมกำลังงานที่จ่ายให้แก่โหลด หรือควบคุมค่าองค์ประกอบหลักมูลของแรงดันเอาต์พุตนั่นเอง

ตารางที่ 2.2 อัตราส่วนระหว่างค่า RMS ของฮาร์มอนิกกับ V_s ของแรงดันสายของอินเวอร์เตอร์ 3 เฟส

h	m_a	0.1	0.2	0.3	0.4	0.5
1		0.122	0.245	0.367	0.490	0.612
$m_f \pm 2$		0.010	0.037	0.080	0.135	0.195
$m_f \pm 4$					0.005	0.011
$2m_f \pm 1$		0.116	0.200	0.227	0.192	0.111
$2m_f \pm 5$					0.008	0.020
$3m_f \pm 2$		0.027	0.085	0.124	0.108	0.038
$3m_f \pm 4$			0.007	0.029	0.064	0.096
$4m_f \pm 1$		0.100	0.096	0.005	0.064	0.042
$4m_f \pm 5$				0.021	0.051	0.073
$4m_f \pm 7$					0.010	0.030

2.5 เซนเซอร์ตรวจจับกระแส (Hall Sensor)

ปัจจุบันความต้องการใช้เซนเซอร์ตรวจจับกระแสราคาถูก ความเที่ยงตรงสูงและมีขนาดเล็กได้เพิ่มขึ้นอย่างรวดเร็ว ในอุตสาหกรรมยานยนต์ เครื่องใช้ไฟฟ้าและระบบสื่อสารต่างๆ มีเทคนิคหลากหลายวิธีในการตรวจจับสัญญาณเอาต์พุตออกเป็นแรงดันไฟฟ้าที่สัมพันธ์กับกระแสที่ไหลผ่าน หนึ่งในวิธีนั้นคือ การใช้ฮอลล์เซ็นเซอร์ และความก้าวหน้าทางเทคโนโลยีทำให้เราสามารถบูรณาการอิเล็กทรอนิกส์และฮอลล์เซ็นเซอร์ให้รวมไว้ในชิปเพียงตัวเดียว ทำให้ความสามารถในการตรวจจับกระแสมีมากขึ้น โดยยังคงแยกขาดกันทางไฟฟ้าได้อยู่ การใช้ฮอลล์เซ็นเซอร์เป็นเทคนิคตรวจจับกระแสในทางอ้อมโดยการวัดค่าความเข้มของสนามแม่เหล็กที่เกิดขึ้นรอบๆ สายไฟ โดยต้องใช้แกนเฟอร์ไรต์ช่วยในการวัดค่าความเข้มของสนามแม่เหล็ก ดังรูป 2.9



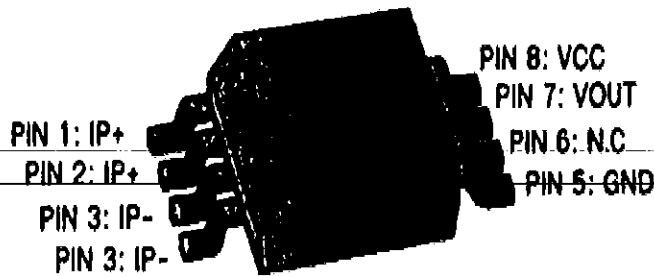
รูปที่ 2.9 การวัดกระแสโดยใช้ฮอลล์เซ็นเซอร์ [7]

ต่อมาเป็น เซ็นเซอร์ตรวจจับกระแสชนิดชีพเดียว โดยรวมเอาฮอลล์เซ็นเซอร์และวงจรอิเล็กทรอนิกส์ไว้ในชิปตัวเดียว โดยแนวคิดของเซ็นเซอร์ชนิดนี้ใช้ขดลวดตัวนำวางภาคไปบนตัวฮอลล์เซ็นเซอร์ แล้วปิดทับด้วยเรซินที่ใช้ในการแพ็คเกจ ไอซี กลายเป็นไอซีตัวหนึ่ง ข้อดีของฮอลล์เซ็นเซอร์แบบชีพเดียวก็คือ กำลังสูญเสียต่ำมากเมื่อเทียบกับการใช้ตัวต้านทานชนิด



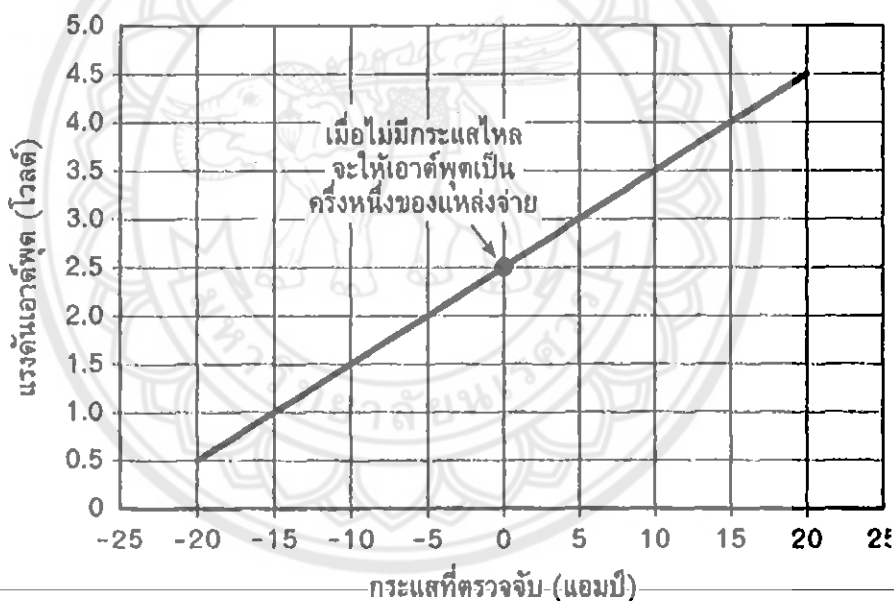
รูปที่ 2.10 ลักษณะภายในของเซ็นเซอร์ตรวจจับกระแส [7]

ถ้าสำหรับขนาดกระแสที่สามารถตรวจจับได้ ขึ้นอยู่กับรุ่นของฮอลล์เซ็นเซอร์ เช่น เบอร์ ACS706TELC-20A สามารถตรวจจับกระแสที่ไหลผ่านตัวมันสูงถึง ± 20 แอมป์ โดยลวดตัวนำที่เป็นทองแดงซึ่งผ่านเข้าไปในไอซีมีความต้านทานประมาณ 1.5 มิลิโอห์มเท่านั้น เพื่อให้มีกำลังสูญเสียน้อยที่สุดซึ่งจะทำให้เกิดความร้อนน้อยตามไปด้วย สำหรับการทำงานของเซ็นเซอร์ตรวจจับกระแสเบอร์ ACS706TELC-20A คือ เมื่อจ่ายไฟเลี้ยงให้กับไอซีแรงดันเอาต์พุตปกติจะอยู่ที่ประมาณครึ่งหนึ่งของแหล่งจ่ายไฟ เช่นเมื่อจ่ายไฟ 5 V แรงดันเอาต์พุตก็จะมีค่า 2.5 V นี่เป็นแรงดันเอาต์พุตขณะที่ยังไม่มีการไหลผ่านจากขา 1,2 ไปยังขา 3,4 และถ้ามีการไหลจากขา 1,2 ไปยังขา 3,4 แรงดันเอาต์พุตก็จะเพิ่มขึ้นจาก 2.5 V แต่ถ้ามีการไหลจากขา 3,4 ไปยังขา 1,2 เซ็นเซอร์ตรวจจับกระแสจะให้แรงดันเอาต์พุตลดลงจาก 2.5 V ส่วนค่าจะเพิ่มขึ้นหรือลดมากน้อยเพียงใดนั้นขึ้นอยู่กับปริมาณกระแสที่ไหลผ่านตัวเซ็นเซอร์นั่นเอง



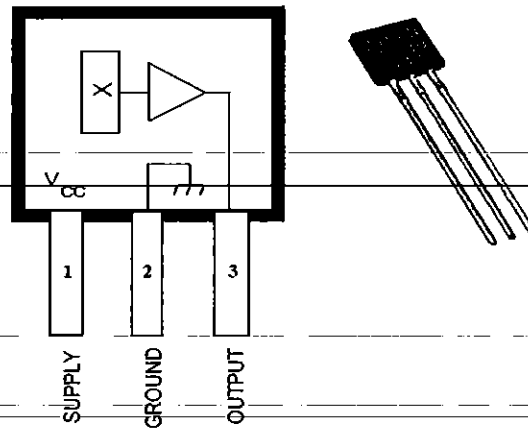
รูปที่ 2.11 เซ็นเซอร์ตรวจจับกระแสเบอร์ ACS706TELC-20A [8]

ในรูปที่ 2.12 แสดงกราฟความสัมพันธ์ระหว่างกระแสที่ไหลผ่านตัวเซ็นเซอร์กับแรงดันเอาต์พุตที่ได้ จากกราฟจะเห็นว่าแรงดันเอาต์พุตจะลดลงเหลือ 0.5 V ที่กระแส - 20 A (กระแสไหลจากขา 3,4 ไปยังขา 1,2) และแรงดันเอาต์พุตจะเพิ่มขึ้นเป็น 4.5 V เมื่อมีกระแส 20 A (กระแสไหลจากขา 1,2 ไปยังขา 3,4)



รูปที่ 2.12 กราฟแสดงแรงดันเอาต์พุตเมื่อตรวจจับกระแสที่ค่าต่างๆ โดยใช้แรงดันไฟเลี้ยง 5V [8]

สำหรับเซ็นเซอร์ที่ใช้ในการตรวจจับสนามแม่เหล็กหรือฮอลล์เซ็นเซอร์ จะมีหลักการทำงานเช่นเดียวกันกับเซ็นเซอร์ตรวจจับกระแสดังที่กล่าวมาข้างต้น เพียงแต่จะตอบสนองต่อสนามแม่เหล็กที่เข้ามาใกล้ตัวไอซี ตัวอย่างเช่น ฮอลล์เซ็นเซอร์เบอร์ UGN 3503 ซึ่งเป็นชนิดเชิงเส้นซึ่งมีการจัดขา ดังรูป 2.13 และตัวอย่างรูปแบบการใช้งานดังรูป 2.14 ตามลำดับ



รูปที่ 2.13 การจัดขาและตัวไอซีฮอลล์เซนเซอร์เบอร์ UGN 3503 [9]



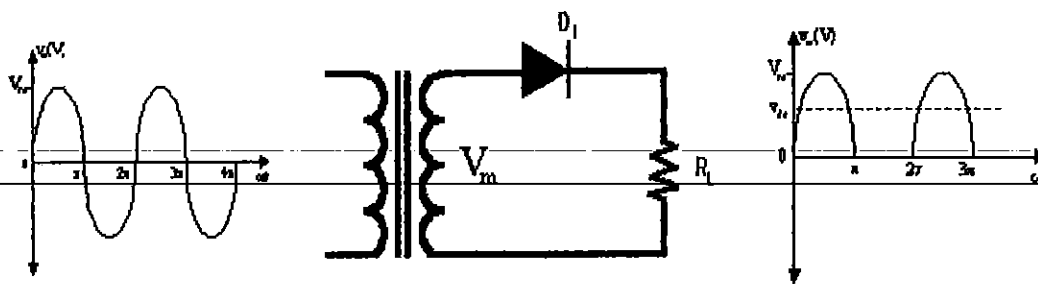
รูปที่ 2.14 รูปแบบการใช้ฮอลล์เซนเซอร์ในการใช้งาน [9]

2.6 วงจรเรียงกระแส (Rectifier)

วงจรเรียงกระแสหรือเรียกอีกอย่างหนึ่งว่า rectifier คือวงจรไฟฟ้าที่มีคุณสมบัติในการแปลงสัญญาณกระแสสลับให้กลายเป็นสัญญาณไฟฟ้ากระแสตรงหรือมีคุณสมบัติยอมให้ไฟฟ้าไหลผ่านไปทิศทางใดทิศทางหนึ่ง อุปกรณ์ที่นิยมใช้ในการแปลงสัญญาณได้แก่ ไดโอด

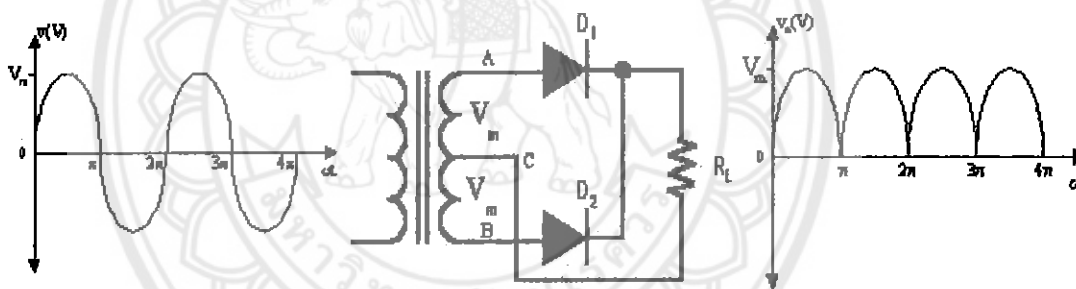
2.6.1 วงจรเรียงกระแสแบบครึ่งคลื่น (Half Wave Rectifier Circuit)

จากรูปที่ 2.15 อธิบายการทำงานของวงจรได้ดังนี้ เมื่อสัญญาณครึ่งบวกเข้ามาไดโอดจะถูกไบอัสตรงทำให้กระแสไหลในวงจรเกิดแรงดันที่ตกคร่อมตัวต้านทาน L R ตามลักษณะของสัญญาณอินพุต แต่เมื่อมีสัญญาณครึ่งลบเข้ามาไดโอดจะถูกไบอัสกลับจึงทำให้มีกระแสไหลในวงจร แรงดันที่ตกคร่อม L R จึงมีค่าเป็นศูนย์คั้งนั้นสัญญาณเอาต์พุตที่ไหลคจึงมีค่าเป็นสัญญาณรูปครึ่งไซเคิล ที่เรียกว่า ฮาล์ฟเวฟ (Half Wave)



รูปที่ 2.15 วงจรเรียงกระแสแบบครึ่งคลื่น [10]

จากข้อเสียของวงจรเรียงกระแสแบบครึ่งคลื่น สามารถแก้ไขปัญหาดังกล่าวได้โดยใช้วงจรที่เรียกว่าวงจรเรียงกระแสแบบเต็มคลื่น วงจรเรียงกระแสแบบนี้จะใช้ไดโอดอย่างน้อย 2 ตัวต่ออยู่ในวงจรเพื่อที่จะทำให้ไดโอดสามารถนำกระแสในแต่ละครึ่งไซเคิลของกระแสสลับ ไดโอดทั้งสองจึงทำหน้าที่เป็นตัวจ่ายกระแสให้กับความต้านทานโหลดตัวละครึ่งไซเคิลแต่ต้องจ่ายให้ทิศทางเดียวกัน ดังนั้นวงจรจึงสามารถจ่ายไฟกระแสตรงได้ตรงกว่าแบบครึ่งคลื่น



รูปที่ 2.16 วงจรเรียงกระแสแบบเต็มคลื่น[10]

2.6.3 วงจรเรียงกระแสเต็มคลื่นแบบเซ็นเตอร์แท็ป (Center Tap Rectifier Circuit)

เมื่อมีแรงดันสลับป้อนเข้ามาทางขดลวดปฐมภูมิของหม้อแปลงจะเกิดแรงดันขึ้นทางขดทุติยภูมิ คือ ขั้ว A และ B เนื่องจากจุด C เป็นจุดกึ่งกลางของขดทุติยภูมิ ดังนั้นแรงดันไฟครึ่งหนึ่งจึงเกิดขึ้นที่ขั้ว AC และอีกครึ่งหนึ่งจะปรากฏที่ขั้ว CB และแรงดันระหว่างขดปฐมภูมิและขดทุติยภูมิจะมีเฟสตรงกันข้าม คือ 180 องศา เมื่อแรงดันไฟในครึ่งไซเคิลแรก คือครึ่งไฟบวกเข้ามาที่อินพุตทางด้านขดปฐมภูมิศักย์ไฟฟ้าที่จุด B จะมีค่าบวกเมื่อเทียบกับ A หรือ C และศักย์ไฟฟ้าที่จุด C จะมีค่าบวกเมื่อเทียบกับจุด A ดังนั้น ไดโอด \$D_1\$ จะไม่นำกระแสไฟฟ้า ส่วน ไดโอด \$D_2\$ จะนำกระแสได้เกิดการไหลของกระแสไฟฟ้าขึ้นในวงจร ตามรูปที่ 2.17

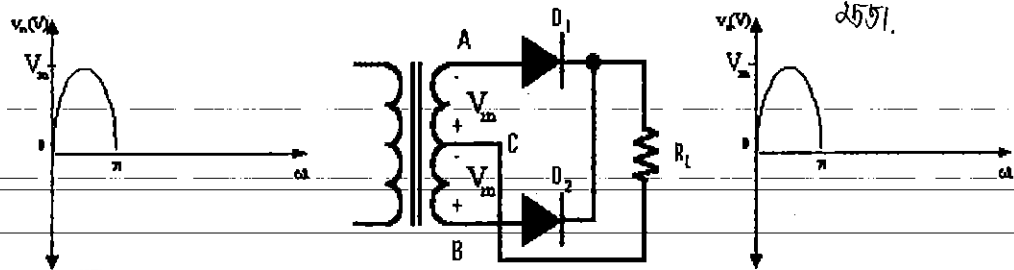
เมื่อแรงดันไฟในครึ่งไซเคิลต่อมา คือครึ่งไฟลบเข้ามาที่อินพุตทางด้านขดปฐมภูมิ ศักย์ไฟฟ้าที่จุด A จะมีค่าบวกเมื่อเทียบกับ B หรือ C และศักย์ไฟฟ้าที่จุด C จะมีค่าบวกเมื่อเทียบกับ

5200061

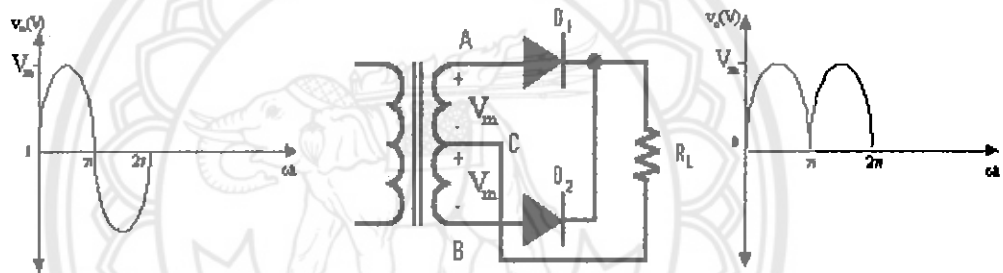
15080952 e.v

จุด B ค้างนั้นไดโอด D2 จะไม่นำกระแสไฟฟ้า ส่วนไดโอด D1 จะนำกระแสได้ เกิดการไหลของกระแสไฟฟ้าขึ้นในวงจร ตามรูปที่ 2.18 ค่าแรงดันเอาต์พุตจะได้อีกครั้ง ไซเคิล กระแสผ่านตัวต้านทาน R_L จะไหลไปในทิศทางเดียวกับตอนแรก

ร.ร.
ย.ร.ร.



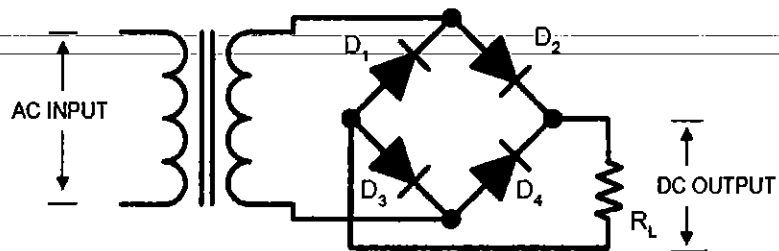
รูปที่ 2.17 แสดงการทำงานของวงจรเรียงกระแสแบบเซ็นเตอร์แท็ปเมื่ออินพุตเป็นซีกบวก [10]



รูปที่ 2.18 แสดงการทำงานของวงจรเรียงกระแสแบบเซ็นเตอร์แท็ปเมื่ออินพุตเป็นซีกลบ [10]

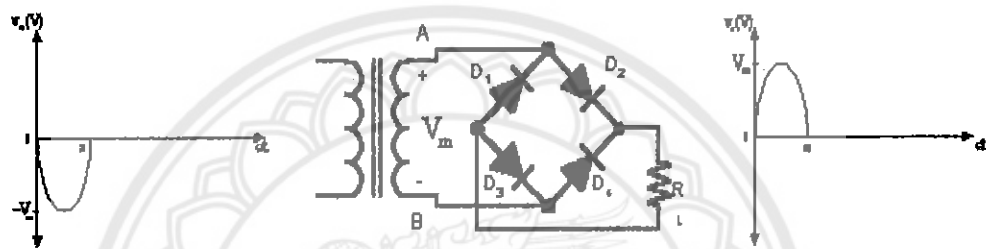
2.6.4 วงจรเรียงกระแสแบบบริดจ์ (Bridge Rectifier Circuit)

วงจรเรียงกระแสแบบเต็มคลื่น (Full Wave) อีกแบบหนึ่งคือวงจรเรียงกระแสแบบบริดจ์ แรงดันไฟสลับจะต่อเข้ากับสองมุมของวงจรบริดจ์และเอาต์พุตจะถูกนำออกที่สองมุมที่เหลือ ดังในรูปที่ 2.19

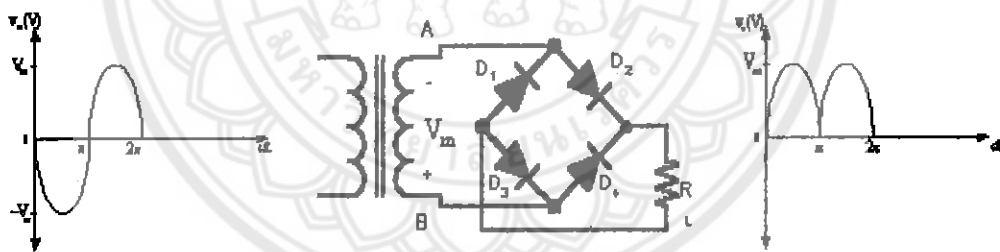


รูปที่ 2.19 วงจรเรียงกระแสแบบบริดจ์ [10]

ในแต่ละครึ่งไซเคิลของวงจรอินพุตสมมติว่าเมื่อขั้ว A ของขดลวดปฐมภูมิมีค่าเป็นบวก และขั้ว B มีค่าเป็นลบ จึงเหมือนกับครึ่งไซเคิลที่ถูกป้อนเข้าทางขดปฐมภูมิของหม้อแปลง ไดโอด D2 และ D3 จะอยู่ในลักษณะไบอัสตรง คังนั้นกระแสจึงไหลครบวงจรจากขั้ว A ผ่านไดโอด D2 ความต้านทานโหลดและไดโอด D3 แล้วกลับเข้าสู่ขั้ว B ของหม้อแปลง ดังรูปที่ 2.20 และเมื่อแรงดันไฟสลับเปลี่ยนข้างมาเป็นขั้วบวก ที่ขั้ว B และที่ขั้ว A เป็นลบ การนำกระแสของไดโอดจะเปลี่ยนไปโดยเริ่มจากจุด B ของขดลวดปฐมภูมิผ่าน D4 ความต้านทานโหลด และ D1 กลับเข้าขั้ว A ของหม้อแปลง ทิศทางแรงดันตกคร่อม โหลดจะมีทิศทางเดียวกับตอนแรกคือ มีขั้วบวกอยู่ทางด้านบน ดังรูปที่ 2.21 คังนั้นการนำกระแสไดโอดจะเกิดสลับกันที่ละสองตัว D2 กับ D3 และ D1 กับ D4



รูปที่ 2.20 แสดงการทำงานของวงจรเรียงกระแสเต็มคลื่นแบบบริดจ์เมื่ออินพุตเป็นซีกลบ [10]



รูปที่ 2.21 แสดงการทำงานของวงจรเรียงกระแสเต็มคลื่นแบบบริดจ์เมื่ออินพุตเป็นซีกบวก [10]

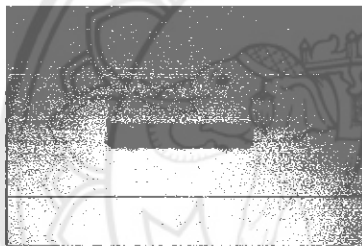
บทที่ 3

การออกแบบการทดลอง

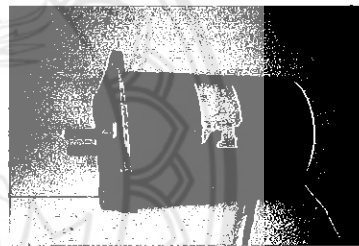
ในส่วนของรายละเอียดและหลักการรวมถึงทฤษฎีต่างๆ ได้กล่าวมาแล้วในบทที่ 2 ในบทนี้ ผู้จัดทำจึงได้เริ่มดำเนินการออกแบบวงจรควบคุมในส่วนของฮาร์ดแวร์ และซอฟต์แวร์บ้างในบางส่วน และเริ่มสร้างวงจรควบคุมการทำงานของมอเตอร์ตามทฤษฎี ซึ่งรายละเอียดการออกแบบมีดังนี้

3.1 อุปกรณ์การทดลอง

อุปกรณ์ที่ใช้ในการทดลองมีทั้งอุปกรณ์อิเล็กทรอนิกส์ต่างๆ สายไฟ และคอมพิวเตอร์ที่ใช้ในการเขียนโปรแกรม รวมทั้งอุปกรณ์ดังรูป



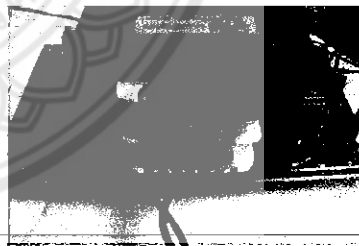
รูปที่ 3.1 ตัวต้านทาน 30วัตต์ 1 โอคั้ม



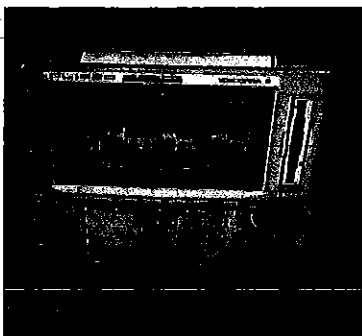
รูปที่ 3.2 Brushless DC Motor 24 V



รูปที่ 3.3 แคลมป์มิเตอร์



รูปที่ 3.4 หม้อแปลง 20 V



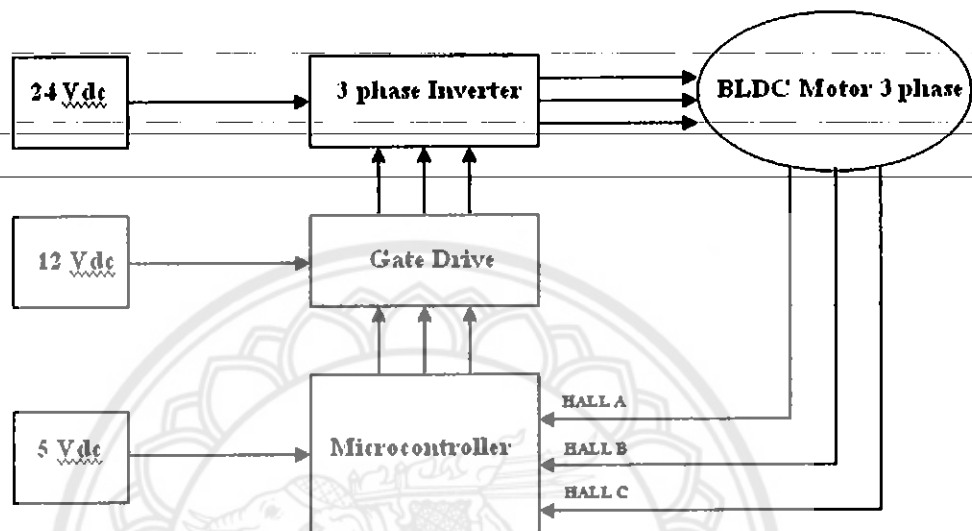
รูปที่ 3.5 Oscilloscope



รูปที่ 3.6 Power & Harmonics Analyzer

3.2 ระบบการทำงานของวงจรควบคุมการทำงานของมอเตอร์

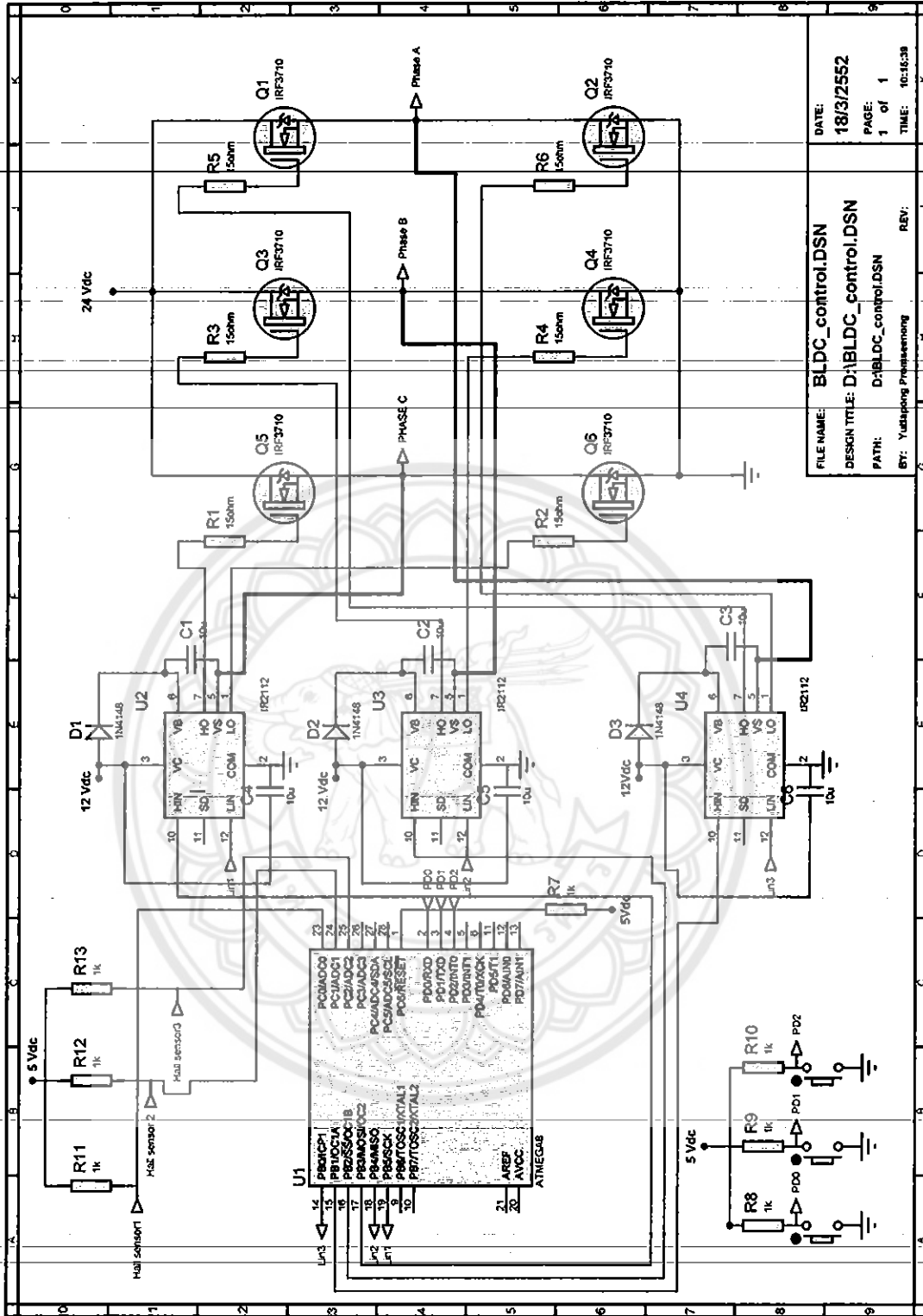
ระบบการทำงานของวงจรควบคุมการทำงานของมอเตอร์ที่ได้ออกแบบและสร้างขึ้นมานั้น ประกอบด้วยส่วนต่างๆ ทั้งส่วนที่เป็นฮาร์ดแวร์และซอฟต์แวร์ โดยการทำงานของระบบควบคุมการทำงานของมอเตอร์นั้นสามารถอธิบายได้ดังนี้



รูปที่ 3.7 Block Diagram

จากบล็อกไดอะแกรมแสดงการควบคุมของมอเตอร์ ชุดขับเคลื่อนมอเตอร์ประกอบด้วย 2 ส่วนหลัก คือ

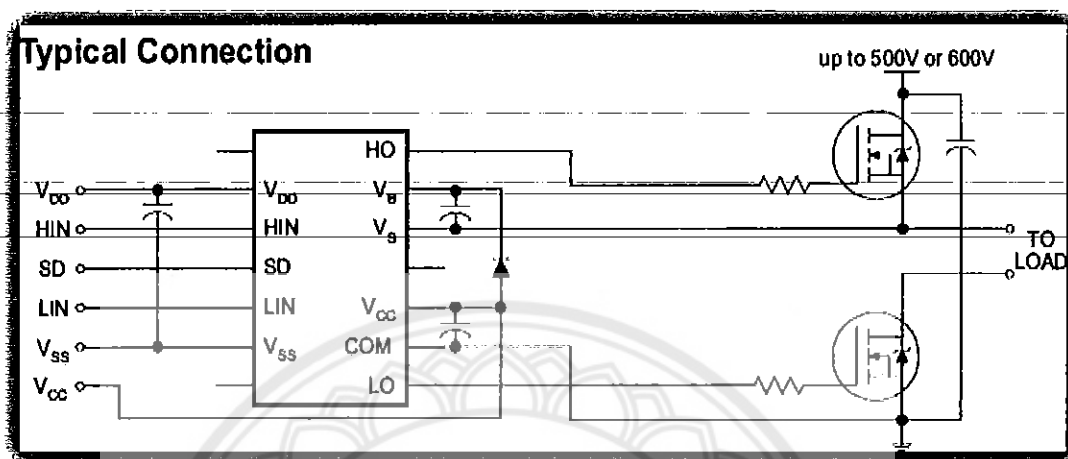
1. วงจรอินเวอร์เตอร์ อินเวอร์เตอร์จะทำหน้าที่สวิตซ์มอสเฟตเพื่อแปลงไฟกระแสตรงเป็นรูปคลื่นสวิตซ์ 3 เฟส โดยแรงดันที่จ่ายให้กับอินเวอร์เตอร์ได้แก่ 24 Vdc
2. ไมโครคอนโทรลเลอร์ จะทำหน้าที่สั่งการควบคุมการเปิด-ปิด ของมอสเฟตในอินเวอร์เตอร์-โดยได้รับสัญญาณจาก-Hall-sensor-ในมอเตอร์เป็นตัวกำหนดการเปิด-ปิดของมอสเฟต โดยจ่ายไฟ 5 Vdc เป็นไฟเลี้ยงไมโครคอนโทรลเลอร์ และ 12 Vdc เป็นไฟเลี้ยงให้กับตัวขับเคลื่อน



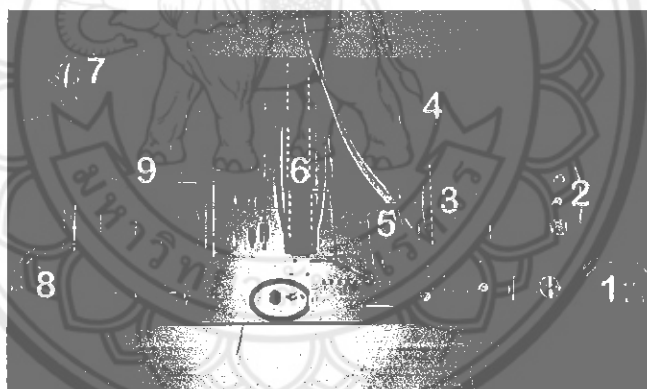
FILE NAME:	BLDC_control.DSN
DESIGN TITLE:	D:\BLDC_control.DSN
DATE:	18/3/2552
PATH:	D:\BLDC_control.DSN
BY:	Yudasing Praeemong
PAGE:	1 of 1
REV:	
TIME:	10:16:39

รูปที่ 3.8 วงจรที่ใช้ขับให้มอเตอร์ทำงาน

จากไออะแกรมของวงจรข้างต้น ไมโครคอนโทรลเลอร์ที่ใช้ในวงจรคือ เบอร์ ATMEGA8 ใช้ขา PB1 PB 2 PB3 เป็นขาสัญญาณ PWM ส่วน Gate Drive ใช้เบอร์ IR 2112 ใช้ขา Ho ขั้วมอเตอร์กึ่งบน / Lo ขั้วมอเตอร์กึ่งล่าง ส่วนมอเตอร์ในวงจรอินเวอร์เตอร์ใช้เบอร์ IRF 3710



รูปที่ 3.9 รูปแบบการต่อขาของ IR 2112



รูปที่ 3.10 จุดขั้วเคลื่อนมอเตอร์

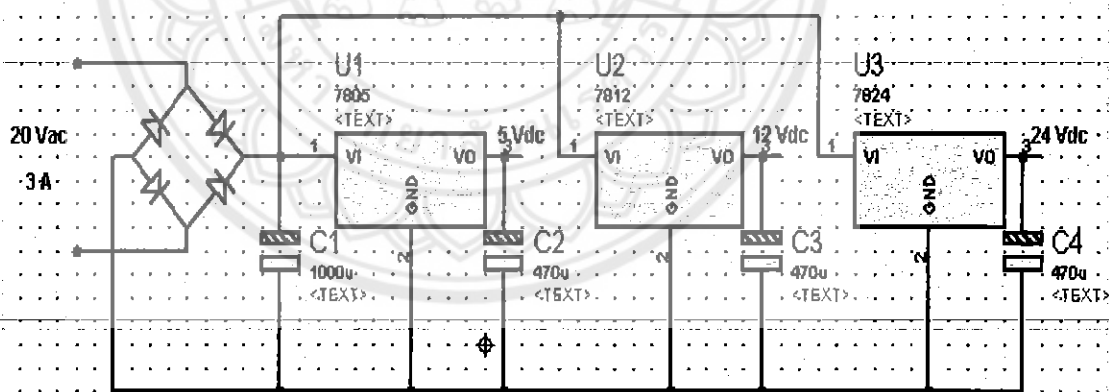
- | | |
|-----------|---|
| หมายเลข 1 | หมายถึง ขั้วต่อไฟดีซี +12 V เพื่อจ่ายให้กับ IR2112 และลดแรงดันด้วย Regulator เบอร์ 7805 ได้ +5 V เพื่อจ่ายให้ MCU |
| หมายเลข 2 | หมายถึง สวิตช์ควบคุมมอเตอร์ |
| หมายเลข 3 | หมายถึง ไมโครคอนโทรลเลอร์เบอร์ ATMEGA8 ของ AVR |
| หมายเลข 4 | หมายถึง คอนเน็คเตอร์ของเครื่องโปรแกรม |
| หมายเลข 5 | หมายถึง Hall Sensor |
| หมายเลข 6 | หมายถึง Gate Drive เบอร์ IR2112 |
| หมายเลข 7 | หมายถึง ขั้วต่อสายไปยังมอเตอร์ |

หมายเลข 8	หมายถึง ขั้วต่อไฟดีซี +24V เพื่อจ่ายให้วงจรมอเตอร์
หมายเลข 9	หมายถึง วงจรมอเตอร์
หมายเลข 10	หมายถึง ไฟแสดงสถานะของวงจร

3.2.1 แหล่งจ่ายไฟ DC ให้กับวงจร

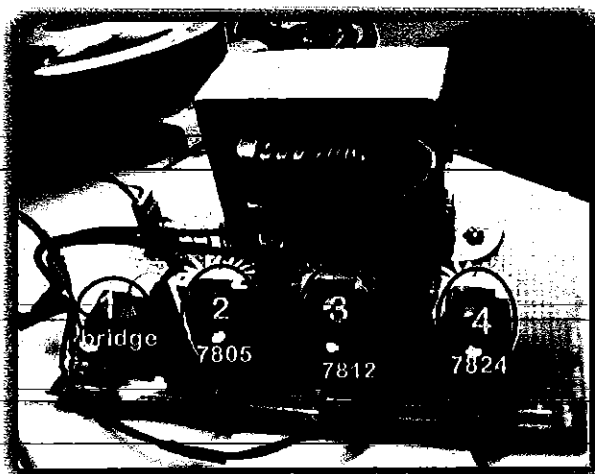
ใช้วงจรเรียงกระแสแบบฟูลบริดจ์แปลงไฟจากกระแสสลับเป็นกระแสตรง อุปกรณ์ที่ใช้ในการสร้างแหล่งจ่ายไฟมีดังนี้

- หม้อแปลงขนาด 20 โวลต์ 3 แอมป์ 1 ลูก เพื่อแปลงไฟจาก 220 โวลต์ เป็น 20 โวลต์
- full bridge rectifier 1 ตัว แปลงไฟกระแสสลับเป็นไฟกระแสตรง
- ตัวเก็บประจุขนาด 1,000 uF 1 ตัว
- ตัวเก็บประจุขนาด 0.47 uF 3 ตัว
- ไอซีเรกูเลเตอร์เบอร์ 7805 เพื่อรักษาระดับแรงดันที่ 5 โวลต์
- ไอซีเรกูเลเตอร์เบอร์ 7812 เพื่อรักษาระดับแรงดันที่ 12 โวลต์
- ไอซีเรกูเลเตอร์เบอร์ 7824 เพื่อรักษาระดับแรงดันที่ 24 โวลต์
- แผ่นปรีนอเนกประสงค์
- ซิงค์ระบายความร้อนเพื่อระบายความร้อนให้กับไอซีเรกูเลเตอร์



รูปที่ 3.11 วงจรเรียงกระแส

จากวงจรนี้จะได้แรงดันที่ต้องการคือ 5 V , 12 V , 24 V เพื่อนำไปจ่ายให้กับชุดใคร่ของมอเตอร์

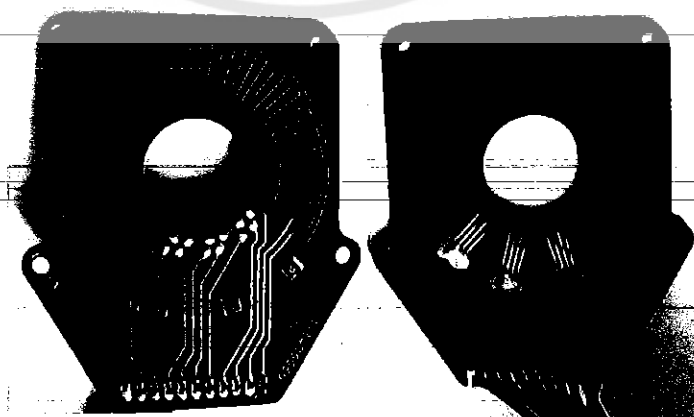


รูปที่ 3.12 แหล่งจ่ายไฟกระแสตรง

- | | |
|-----------|--|
| หมายเลข 1 | หมายถึง บริดจ์เรียงกระแส |
| หมายเลข 2 | หมายถึง Regulator เบอร์ 7805 ได้ไฟ 5 Vdc 1A |
| หมายเลข 3 | หมายถึง Regulator เบอร์ 7812 ได้ไฟ 12 Vdc 1A |
| หมายเลข 4 | หมายถึง Regulator เบอร์ 7824 ได้ไฟ 24 Vdc 1A |

3.2.2 Hall Sensor

ภายในมอเตอร์ไฟฟ้ากระแสตรงแบบไร้แปรงถ่านแบบ 3 เฟส ภายในจะมีวงจรของ Hall Sensor ซึ่งวงจรที่มาพร้อมกับมอเตอร์จะมีสายของ Hall Sensor อยู่ 10 สาย แต่เพื่อง่ายต่อการเขียนโปรแกรมควบคุมการทำงานของมอเตอร์ จึงมีการติดตั้ง Hall Sensor ใหม่โดยใช้ Hall Sensor แบบเชิงเส้น รุ่น Allegro เบอร์ UGN 3503 ติดตรงบริเวณร่องสลักของมอเตอร์ ห่างกันตัวละ 120 องศา



รูปที่ 3.13 วงจรของ Hall Sensor

Hall Sensor จะตอบสนองต่อความเข้มของสนามแม่เหล็กที่เข้ามาใกล้ตัวไอซี แรงดัน output ปกติของ hall sensor จะเป็นครึ่งหนึ่งของแหล่งจ่าย input แรงดัน output จะมากหรือน้อย ขึ้นอยู่กับความเข้มของสนามแม่เหล็กที่เข้ามาใกล้ตัว Hall Sensor ซึ่งมีค่าประมาณ 1.3 mV/G

3.3 การออกแบบการเขียนโปรแกรม

การ ON/OFF ของมอเตอร์ในวงจรอินเวอร์เตอร์ จะถูกกำหนดโดยสัญญาณจาก Hall Sensor ซึ่งจะมี 6 สเต็ปวนซ้ำกันไปเรื่อยๆ ดังตาราง

ตารางที่ 3.1 แสดงการ ON/OFF ของมอเตอร์ในวงจรอินเวอร์เตอร์

Pin	RE2	RE1	RE0	RC5	RC4	RC3	RC2	RC1	RC0
Phase	Sensor C	Sensor B	Sensor A	C High Drive	C Low Drive	B High Drive	B Low Drive	A High Drive	A Low Drive
1	1	0	1	0	0	0	1	1	0
2	1	0	0	1	0	0	1	0	0
3	1	1	0	1	0	0	0	0	1
4	0	1	0	0	0	1	0	0	1
5	0	1	1	0	1	1	0	0	0
6	0	0	1	0	1	0	0	1	0

Hall 1 0 1	มอเตอร์เฟสตัวบนของเฟส A และมอเตอร์เฟสตัวล่างของเฟส B ทำงาน ได้ Vab เขียนเป็น โปรแกรมได้ #define t1 0x06
Hall 1 0 0	มอเตอร์เฟสตัวบนของเฟส C และมอเตอร์เฟสตัวล่างของเฟส B ทำงาน ได้ Vcb เขียนเป็น โปรแกรมได้ #define t2 0x24
Hall 1 1 0	มอเตอร์เฟสตัวบนของเฟส C และมอเตอร์เฟสตัวล่างของเฟส A ทำงาน ได้ Vca เขียนเป็น โปรแกรมได้ #define t3 0x21
Hall 0 1 0	มอเตอร์เฟสตัวบนของเฟส B และมอเตอร์เฟสตัวล่างของเฟส A ทำงาน ได้ Vba เขียนเป็น โปรแกรมได้ #define t4 0x09
Hall 0 1 1	มอเตอร์เฟสตัวบนของเฟส B และมอเตอร์เฟสตัวล่างของเฟส C ทำงาน ได้ Vbc เขียนเป็น โปรแกรมได้ #define t4 0x18
Hall 0 0 1	มอเตอร์เฟสตัวบนของเฟส A และมอเตอร์เฟสตัวล่างของเฟส C ทำงาน ได้ Vba เขียนเป็น โปรแกรมได้ #define t4 0x12

3.3.1 ตัวอย่างการเขียนการควบคุมการจ่ายไฟให้มอเตอร์

```

if((Hc==1)&&(Hb==0)&&(Ha==1))
{
    PORTB=t1;    //0x06 0000 0110 มอสเฟต 1 และ 4
    OCR1A=0x00; // ขา OCR1A ขั้วขา Hin ของ IR2112 ตัวที่สาม (เฟส C)
    OCR1B=0x00; // ขา OCR1B ขั้วขา Hin ของ IR2112 ตัวที่สอง (เฟส B)
    OCR2=PWM;   // Vab // ขา OCR2 ขั้วขา Hin ของ IR2112 ตัวแรก (เฟส A)
}
if((Hc==1)&&(Hb==0)&&(Ha==0))
{
    PORTB=t2;    //0x24 0010 0100 มอสเฟต 5 และ 4
    OCR1B=0x00;
    OCR1A=PWM; // Vcb
    OCR2=0x00;
}
if((Hc==1)&&(Hb==1)&&(Ha==0))
{
    PORTB=t3;    //0x21 0010 0001 มอสเฟต 5 และ 2
    OCR1A=PWM; // Vca // ขา OCR1A ขั้วขา Hin ของ IR2112 ตัวที่สาม (เฟส C)
    OCR1B=0x00; // ขา OCR1B ขั้วขา Hin ของ IR2112 ตัวที่สอง (เฟส B)
    OCR2=0x00; // ขา OCR2 ขั้วขา Hin ของ IR2112 ตัวแรก (เฟส A)
}
if((Hc==0)&&(Hb==1)&&(Ha==0))
{
    PORTB=t4;    //0x09 0000 1001 มอสเฟต 3 และ 2
    OCR1B=PWM; // Vba
    OCR1A=0x00;
    OCR2=0x00;
}

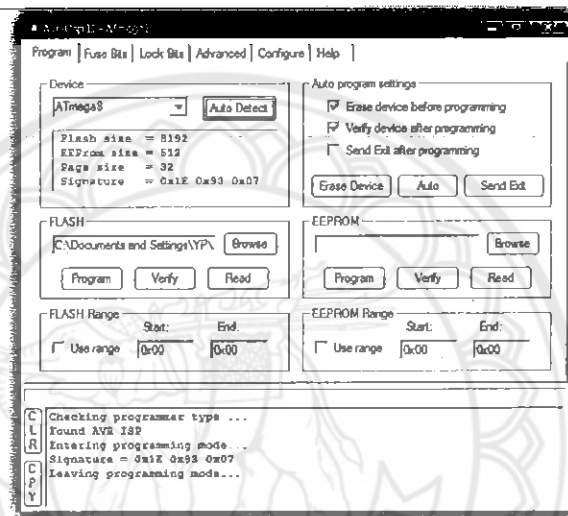
```

โดยเขียนให้ครบทั้ง 6 สเต็ป ดังในตารางที่ 3.1

3.4 การโปรแกรมไฟล์ .hex ลงในไมโครคอนโทรลเลอร์

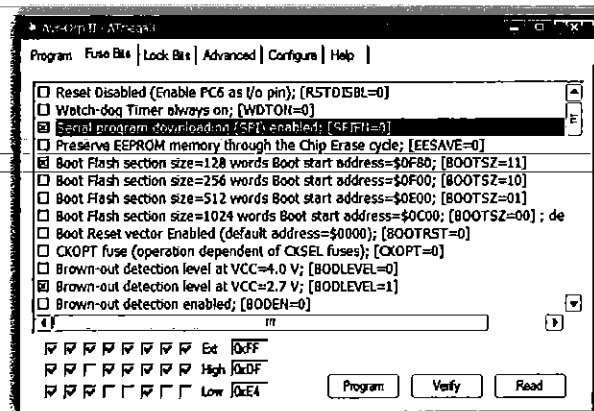
เมื่อทำการคอมไพล์ source code เสร็จแล้วก็จะได้ไฟล์นามสกุล .hex ซึ่งเป็นไฟล์ที่จะต้องดาวน์โหลดลงสู่ไมโครคอนโทรลเลอร์ เครื่องโปรแกรมที่ใช้ได้แก่เครื่อง PX-400 ซึ่งเป็นเครื่องโปรแกรมที่ใช้กับ MCU ตระกูล AVR โปรแกรมที่ใช้ในการดาวน์โหลดได้แก่ AvrOsp II ซึ่งมีขั้นตอนการโปรแกรมดังนี้

1. เปิดโปรแกรมเลือกเบอร์ของ MCU แล้วคลิกที่ auto detect โปรแกรมจะทำการตรวจสอบว่ามี MCU เชื่อมต่ออยู่หรือไม่



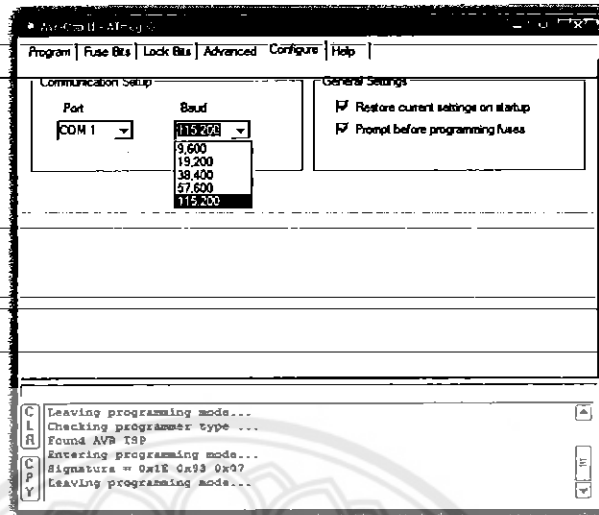
รูปที่ 3.14 ตรวจสอบว่ามี MCU เชื่อมต่ออยู่หรือไม่

2. คลิกที่แถบ fuse bit ใส่เครื่องหมายถูกตรงที่ serial program downloading (SPI) enable และ Int. RC osc.8 MHz; start-up time: 6 ck+64ms



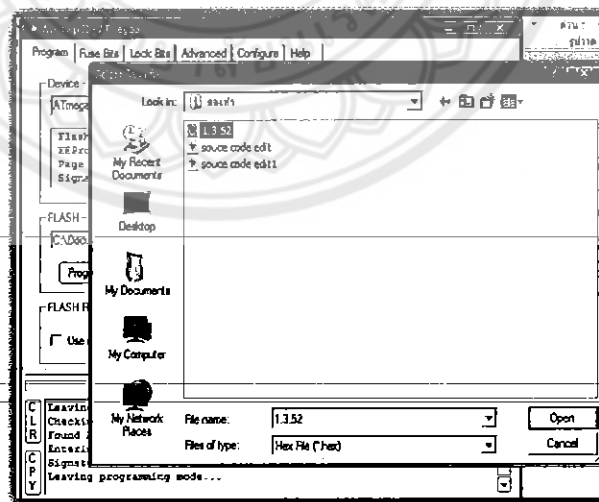
รูปที่ 3.15 เลือกโปรแกรม

3. คลิกที่แถบ **configure** และตรงช่อง Baud เลือก 115200 เท่านั้น



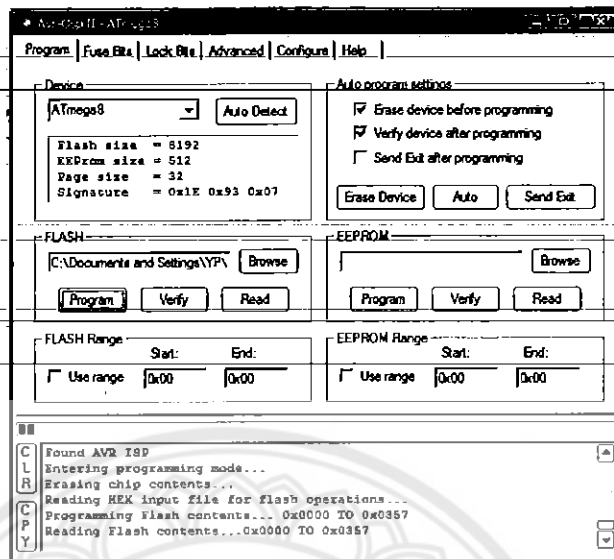
รูปที่ 3.16 การตั้งค่า Configure

4. กลับมาที่แถบ โปรแกรม คลิกตรง **Browse** เพื่อเลือกไดเรกทอรีที่เก็บไฟล์ .hex ไว้แล้ว
คลิก **OPEN**



รูปที่ 3.17 เลือกไดเรกทอรีที่เก็บไฟล์ .hex

5. คลิกที่กรอมโปรแกรม จากนั้นเครื่องก็จะทำการดาวน์โหลดไฟล์ .hex ลงสู่ MCU



รูปที่ 3.18 ดาวน์โหลดไฟล์ .hex ลงสู่ MCU



รูปที่ 3.19 ชุดขับมอเตอร์

บทที่ 4

ผลการทดลอง

4.1 จุดประสงค์การทดลอง

4.1.1 เพื่อต้องการทดสอบดูกราฟแรงดันและกระแสของมอเตอร์

4.1.2 เพื่อหาจุดบกพร่อง สาเหตุ และวิธีในการแก้ไขและปรับปรุง

4.2 ขั้นตอนการทดลอง

4.2.1 การทดสอบเพื่อดูกราฟกระแสของมอเตอร์เมื่อมอเตอร์ทำงาน

- ทำการต่อแหล่งจ่ายไฟกระแสตรงกับชุดขับเคลื่อนมอเตอร์ และมอเตอร์เข้าด้วยกัน
- เมื่อจะทำการวัดค่าเพื่อดูกราฟกระแสของมอเตอร์ขณะหมุนนั้น จะต้องต่อตัว R ตรงสายไฟ 3 เฟส 24 V ก่อนที่จะต่อเข้ามอเตอร์ และทำการวัดเพื่อดูกราฟกระแส โดยใช้สายโพรบของเครื่อง Oscilloscope มาต่อคร่อมตัว R เพื่อดูกราฟกระแสในช่วง Δ เวลาหนึ่ง



รูปที่ 4.1 การต่อตัว R เพื่อวัดกราฟกระแส

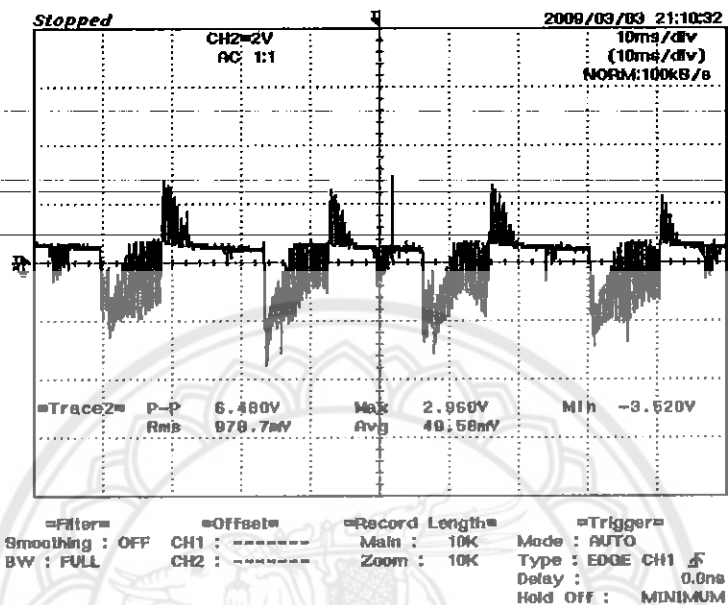
4.2.2 การทดสอบเพื่อดูกราฟแรงดันของมอเตอร์เมื่อมอเตอร์ทำงาน

- ทำการต่อแหล่งจ่ายไฟกระแสตรงกับชุดขับเคลื่อนมอเตอร์และมอเตอร์เข้าด้วยกัน
- ใช้ Oscilloscope วัดแรงดันระหว่างเฟส

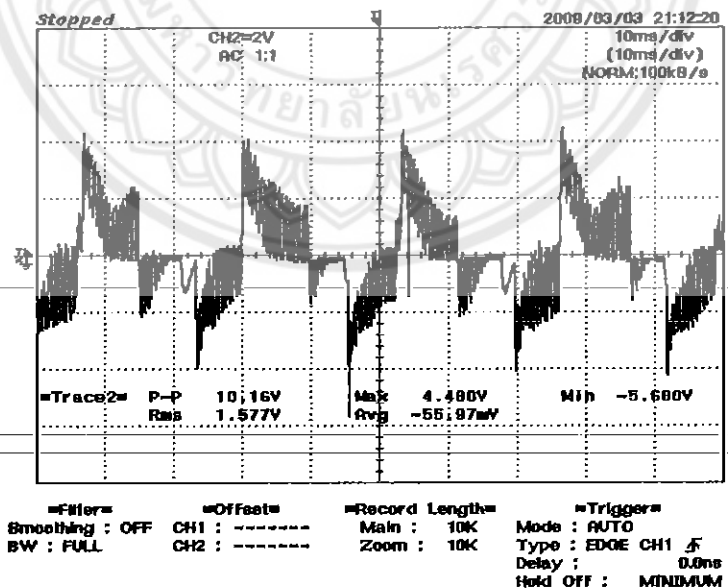
4.3 ผลการทดลอง

จากการทดลองจะได้ผลการทดลองเป็นกราฟแรงดันและกระแสดังนี้

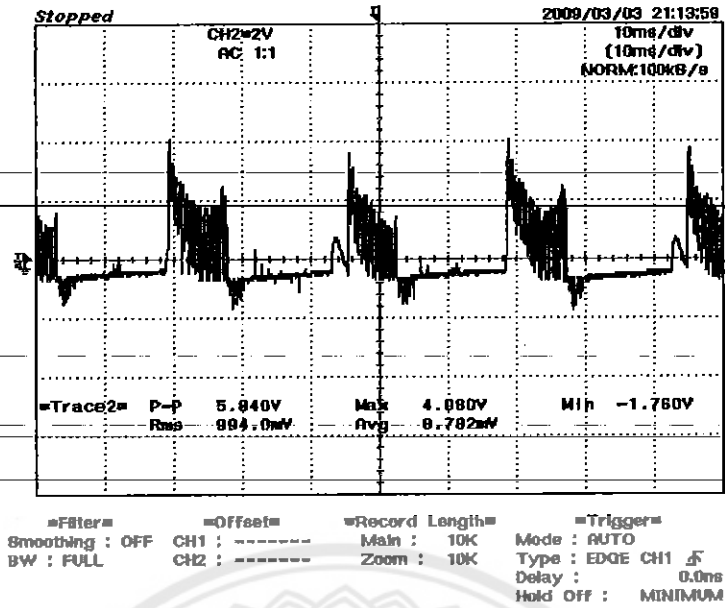
4.3.1 กราฟกระแสที่วัดจากเครื่อง Oscilloscope



รูปที่ 4.2 กราฟกระแสจากเฟส A

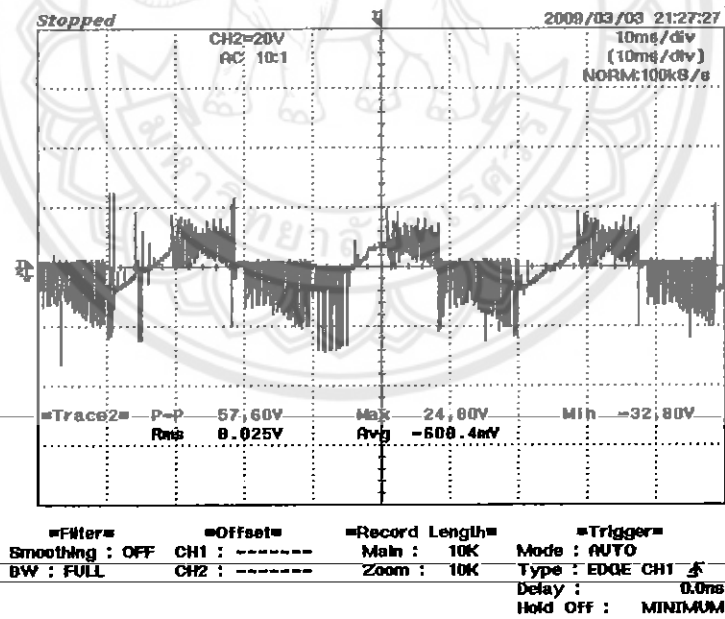


รูปที่ 4.3 กราฟกระแสจากเฟส B

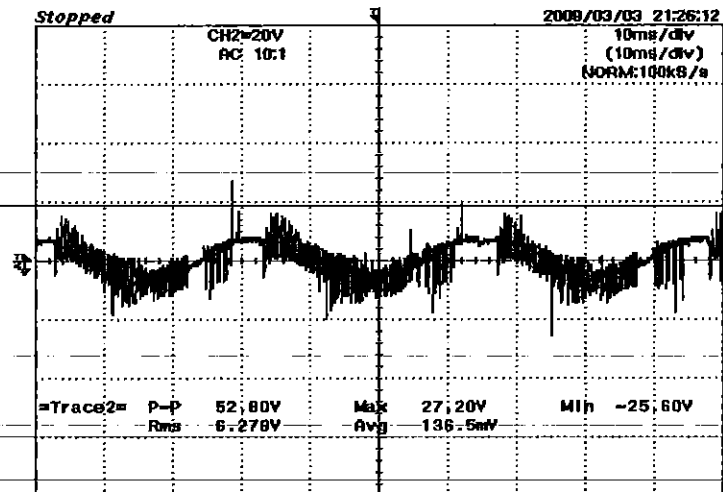


รูปที่ 4.4 กราฟกระแสจากเฟส C

4.3.2 กราฟแรงดันที่วัดจากเครื่อง Oscilloscope

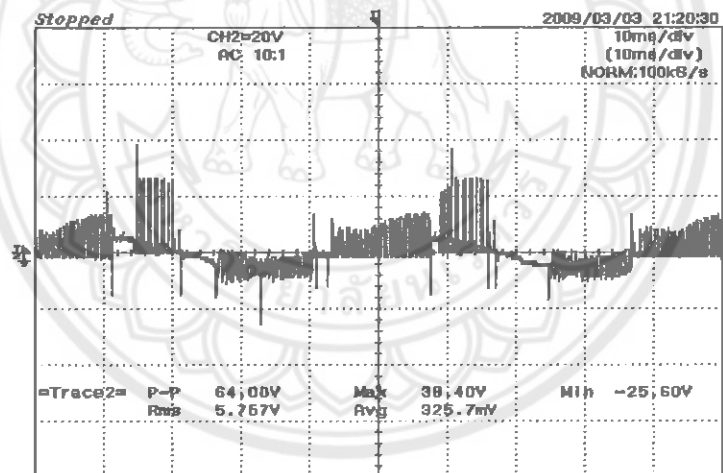


รูปที่ 4.5 กราฟแรงดันระหว่างเฟส A กับ B



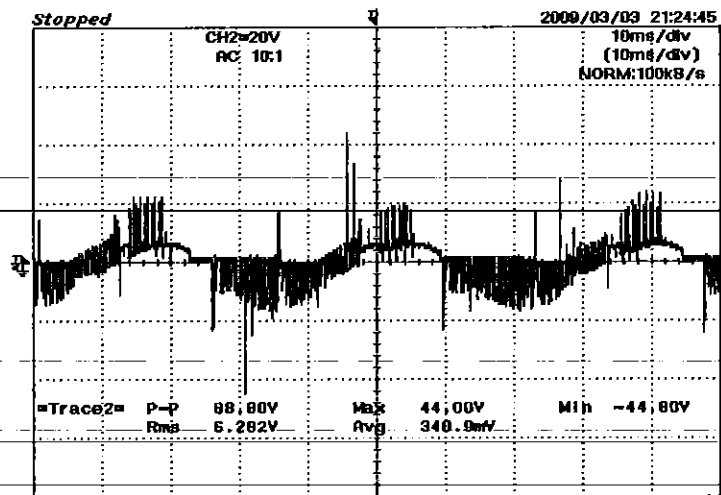
=Filter= =Offset= =Record Length= =Trigger=
Smoothing : OFF CH1 : ----- Main : 10K Mode : AUTO
BW : FULL CH2 : ----- Zoom : 10K Type : EDGE CH1 \uparrow
Delay : 0.0ns
Hold Off : MINIMUM

รูปที่ 4.6 กราฟแรงดันระหว่างเฟส A กับ C



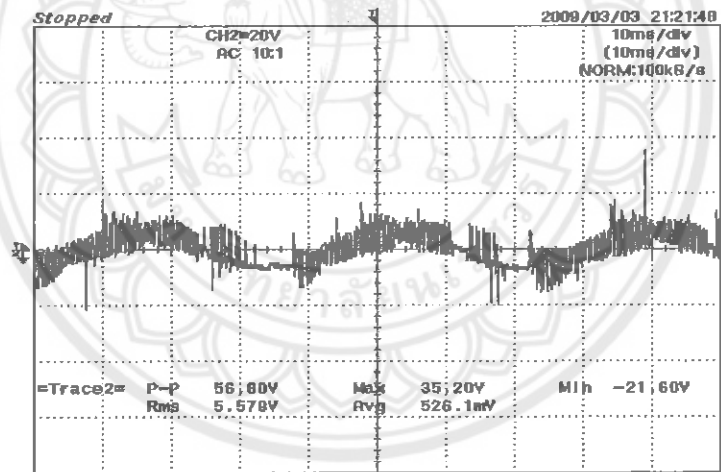
=Filter= =Offset= =Record Length= =Trigger=
Smoothing : OFF CH1 : ----- Main : 10K Mode : AUTO
BW : FULL CH2 : ----- Zoom : 10K Type : EDGE CH1 \uparrow
Delay : 0.0ns
Hold Off : MINIMUM

รูปที่ 4.7 กราฟแรงดันระหว่างเฟส B กับ A



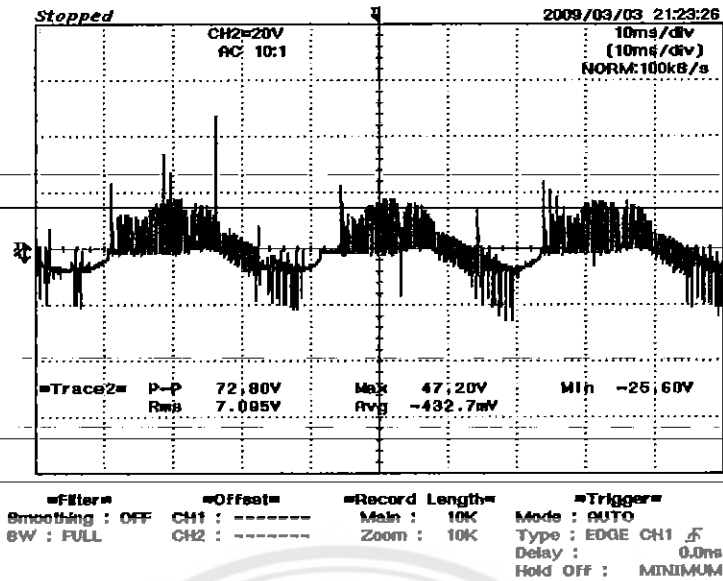
=Filter= =Offset= =Record Length= =Trigger=
 Smoothing : OFF CH1 : ----- Main : 10K Mode : AUTO
 BW : FULL CH2 : ----- Zoom : 10K Type : EDGE CH1 \uparrow
 Delay : 0.0ns
 Hold Off : MINIMUM

รูปที่ 4.8 กราฟแรงดันระหว่าง B กับ C



=Filter= =Offset= =Record Length= =Trigger=
 Smoothing : OFF CH1 : ----- Main : 10K Mode : AUTO
 BW : FULL CH2 : ----- Zoom : 10K Type : EDGE CH1 \uparrow
 Delay : 0.0ns
 Hold Off : MINIMUM

รูปที่ 4.9 กราฟแรงดันระหว่าง C กับ A



รูปที่ 4.10 กราฟแรงดันระหว่าง C กับ B

4.4 วิเคราะห์ผลการทดลอง

จากการทดลองจะเห็นได้ว่ากราฟแรงดันและกราฟกระแสที่ได้จากการวัดค่าจากเครื่อง Oscilloscope เป็นกราฟที่แสดงให้เห็นว่าขณะที่มอเตอร์ทำงานกระแสและแรงดันยังไม่คงที่ สังเกตได้จากรูปลูกคลื่นซึ่งมีลักษณะไม่สม่ำเสมอและนอกจากนี้จะเห็นได้ว่ายังมีสัญญาณฮามอนิกส์มาก

บทที่ 5

บทสรุป

ในบทนี้จะเป็นการสรุปผลที่ได้จากการทดลองในโครงการนี้ พร้อมเสนอแนะแนวทางในการนำโครงการนี้ไปพัฒนาให้มีประสิทธิภาพมากยิ่งขึ้นต่อไปในอนาคต

5.1 สรุปผลการทดลอง

ในโครงการนี้ได้ออกแบบวงจรควบคุมมอเตอร์ไฟฟ้ากระแสตรงแบบไร้แปรงถ่าน ซึ่งมีอินเวอร์เตอร์ทำหน้าที่ on/off มอเตอร์เพื่อแปลงไฟจากกระแสตรงเป็นกระแสสลับ และมีไมโครคอนโทรลเลอร์ควบคุมการทำงานของมอเตอร์ โดยตัวไมโครคอนโทรลเลอร์จะรับสัญญาณจาก Hall Sensor ในมอเตอร์

5.2 ปัญหาในการทำงานและแนวทางแก้ไข

1. ขณะสตาร์ทมอเตอร์ในบางครั้งจะต้องใช้มือช่วยหมุน โรเตอร์มอเตอร์จึงจะหมุน และ Regulator เบอร์ 7824 จะร้อนเร็วมากเมื่อมอเตอร์เริ่มทำงาน

สาเหตุ Regulator เบอร์ 7824 นี้จ่ายกระแสได้เพียง 1 A เท่านั้น จึงทำให้กระแสสตาร์ทไม่เพียงพอ

ผล เมื่อกดสวิทช์มอเตอร์จะทำงานเองในบางครั้งหรือไม่ก็จะต้องใช้มือช่วยหมุน โรเตอร์มอเตอร์จึงจะทำงาน

วิธีแก้ไข หา Regulator ที่สามารถจ่ายกระแสได้มากกว่า 1 A มาแทน Regulator เบอร์ 7824

2. ไม่ทราบวงจรการทำงานของตัว Hall Sensor ที่ติดมากับมอเตอร์รุ่นนี้

สาเหตุ เนื่องจาก Brushless DC Motor รุ่นที่นำมาทดลองเป็นมอเตอร์รุ่นเก่าจึงไม่มี Data Sheet ของวงจรการทำงานของตัว Hall Sensor

ผล ไม่สามารถทำการทดลองให้มอเตอร์ทำงานหรือหมุนได้

วิธีแก้ไข ทำการติดตัว Hall Sensor ใหม่ โดยใช้ตัว Hall Sensor แบบเชิงเส้น ของ Allegro เบอร์ UGN 3503 มาติดตั้งบริเวณช่องสลีตของมอเตอร์ห่างกันตัวละ 120 องศา

5.3 ประเมินผล

จากการดำเนินงานโครงการเมื่อเทียบกับวัตถุประสงค์ในการทำโครงการ ได้ผลดังนี้

5.3.1 ได้รับความรู้เพิ่มเติมเกี่ยวกับหลักการการทำงานของอุปกรณ์อิเล็กทรอนิกส์

5.3.2 สามารถสร้างวงจรควบคุมการทำงานของมอเตอร์ไฟฟ้ากระแสตรงแบบไว้เบรกรถยนต์
และสามารถสั่งการให้มอเตอร์หมุนได้

5.3.3 สามารถใช้ไมโครคอนโทรลเลอร์เบอร์ ATMEGA8 ควบคุมการทำงานของมอเตอร์ได้

5.4 แนวทางการพัฒนา

5.4.1 ออกแบบให้มอเตอร์สามารถหมุนไปและหมุนกลับได้

5.4.2 ออกแบบให้กระแสและแรงดันของมอเตอร์ขณะทำงานมีฮามอนิกส์น้อยลง

5.4.3 ออกแบบสวิตช์ควบคุมการทำงานของมอเตอร์ให้มีปุ่มกด Forward , Reverse , Stop



เอกสารอ้างอิง

- [1] พรจิต ประทุมสุวรรณ. **พื้นฐานการขับเคลื่อนมอเตอร์ไฟฟ้าด้วยอิเล็กทรอนิกส์กำลัง.**
พิมพ์ครั้งที่ 1. กรุงเทพมหานคร: เรือนแก้วการพิมพ์. 2547.
- [2] http://oretano.iele-ab.uclm.es/~mhidalgo/temas/Brushless/00857a_Control_PIC_esquema.pdf
สืบค้นเมื่อ 15 พฤศจิกายน พ.ศ.2551
- [3] นคร ภัคศิชาติ . กฤษฎา ใจเย็น . ชัยวัฒน์ ลิ้มพรจิตรวิไล . **เรียน – เล่นไมโครคอนโทรลเลอร์ AVR กับ ATMEGA8** . พิมพ์ครั้งที่ 1. กรุงเทพมหานคร : อินโนเวตีฟ อีเล็คโทรนิคส์ . 2550.
- [4] http://nara2.homeip.net/~salamai/e_learning/transistor/mosfet.gif สืบค้นเมื่อ 10 พฤศจิกายน พ.ศ.2551
- [5] http://www.cpe.ku.ac.th/~yuen/204471/device/diode_transistor/D_MOSFET.htm สืบค้นเมื่อ 15 พฤศจิกายน พ.ศ.2551
- [6] Anish Varsani, “**Low cost Brushless DC Motor Controller**” . Bachelor's degree of Engineering in the division of Electircal Engineering. 2003.
- [7] http://images.google.co.th/images?ndsp=18&hl=th&lr=lang_th&um=1&q=Hall+Sensor&start=90&sa=N สืบค้นเมื่อ 10 พฤศจิกายน พ.ศ.2551
- [8] <http://www.es.co.th/detail.asp?Prod=ACS706TELC-20A> สืบค้นเมื่อ 15 พฤศจิกายน พ.ศ.2551
- [9] <http://www.datasheetcatalog.org/datasheet/allegromicrosystems/3503.pdf> สืบค้นเมื่อ 15 พฤศจิกายน พ.ศ.2551
- [10] http://elecprnt.rmutl.ac.th/c/index2.php?option=com_docman&task=doc_view&gid=5&Itemid=27 สืบค้นเมื่อ 15 พฤศจิกายน พ.ศ.2551

๙

๑๖



๑

๒

ภาคผนวก (ก)

Source code ของโปรแกรม

```
#include<mega8.h>
#include<math.h>
#include<delay.h>

#define H1 PINC.0
#define H2 PINC.1
#define H3 PINC.2

#define S1 PIND.0
#define S2 PIND.1
#define S3 PIND.2

// CLOCK WISE
#define t1 0x22 // 0010 0010
#define t2 0x24 // 0010 0100
#define t3 0x05 // 0000 0101
#define t4 0x09 // 0000 1001
#define t5 0x18 // 0001 1000
#define t6 0x12 // 0001 0010

int PWM=150;
int temperature=0;
int current=0;
//char Imax=0;
volatile long timerTicks = 0;
int startedTiming = 0;
int throughEnd = 1;
```

```

//long motorSpeed;
//int loops = 0;
//int MOTOR_RPM=900;

#define CHECK_LOOPS 5
#define CHANGE_DIVISOR 50
#define FIRST_ADC_INPUT 3
#define LAST_ADC_INPUT 5

unsigned char adc_data[LAST_ADC_INPUT-FIRST_ADC_INPUT+1];
#define ADC_VREF_TYPE 0x20
// ADC interrupt service routine
// with auto input scanning

#define MIN_PWM 100
#define MAX_PWM 200

#pragma savereg-
interrupt[ADC_INT] void adc_isr(void)
{
#asm
push r26
push r27
push r30
push r31
in r30,sreg
push r30
#endasm

register static unsigned char input_index=0;
// Read the 8 most significant bits
// of the AD conversion result
adc_data[input_index]=ADCH;
// Select next ADC input

```

```
if(++input_index > (LAST_ADC_INPUT-FIRST_ADC_INPUT))
input_index=0;
ADMUX=(FIRST_ADC_INPUT|ADC_VREF_TYPE)+input_index;

// Start the AD conversion
ADCSRA|=0x40;
#asm
pop r30
out sreg,r30
pop r31
pop r30
pop r27
pop r26
#endasm
temperature=(char) ((int) adc_data[0]*100)/256;
current=adc_data[1];
}
#pragma savereg+
// Timer 0 overflow interrupt service routine
interrupt [TIM0_OVF] void timer0_ovf_isr(void)
{
timerTicks += 0xFF;
}

void main(void)
{
PORTB=0x00;
DDRB=0x3F;
PORTC=0x00;
DDRC=0x00;
PORTD=0xF0;
DDRD=0x08;

// Timer/Counter 0 initialization
```

```
// Clock source: System Clock
// Clock value: 7.813 kHz
TCCR0=0x05; //Pre 1024
TCNT0=0x00; // Register Counter
// Timer/Counter 1 initialization
// Clock source: System Clock
// Clock value: 8000.000 kHz
// Mode: Ph. correct PWM top=00FFh
// OC1A output: Non-Inv.
// OC1B output: Non-Inv.
// Noise Canceler: Off
// Input Capture on Falling Edge
TCCR1A=0xA1; // 1010 0001
TCCR1B=0x01;
TCNT1H=0x00;
TCNT1L=0x00;
OCR1AH=0x00;
OCR1AL=0x00;
OCR1BH=0x00;
OCR1BL=0x00;
TCCR2=0x61;
ASSR=0x00;
TCNT2=0x00;
OCR2=0x00;
TCNT0=0xff;
// Timer/Counter 2 initialization
// Clock source: System Clock
// Clock value: 8000.000 kHz
// Mode: Phase correct PWM top=FFh
// OC2 output: Non-Inverted PWM
ASSR=0x00;
TCCR2=0x61;
```

```

TCNT2=0x00;
OCR2=0x00;
// External Interrupt(s) initialization
//INT0: Off
//INT1: Off
GICR|=0x00;
MCUCR=0x00;
//Timer(s)/Counter(s) Interrupt(s) initialization
TIMSK=0x00 | 0x01;

// Analog Comparator initialization
// Analog Comparator: Off
// Analog Comparator Input Capture by Timer/Counter 1: Off
// Analog Comparator Output: Off
//ACSR=0x80;
//SFIOR=0x00;
// ADC initialization
// ADC Clock frequency: 62.500 kHz
// ADC Voltage Reference: AREF pin
// ADC High Speed Mode: Off
// ADC Auto Trigger Source: None
// Only the 8 most significant bits of
//the AD conversion result are used
//ADMUX=FIRST_ADC_INPUT|ADC_VREF_TYPE;
//ADCSRA=0xCF;
//SFIOR&=0xEF;

#asm("sei")

//Commutation code. Motor running with PWM
//*****
***
// MOTOR FORWARD

```

```

/*****
***
while (1)
{
if(S1 == 0)
{
if((H1==1)&&(H2==0)&&(H3==0))
{
PORTB=t1; //0x22 0010 0010
OCR1A=PWM;
OCR1B=0x00;
OCR2=0x00;
}
if((H1==1)&&(H2==1)&&(H3==0))
{
PORTB=t2; //0x24 0010 0100
OCR1B=PWM;
OCR1A=0x00;
OCR2=0x00;
}
if((H1==0)&&(H2==1)&&(H3==0))
{
PORTB=t3; //0x05 0000 0101
OCR1B=PWM;
OCR1A=0x00;
OCR2=0x00;
}
if((H1==0)&&(H2==1)&&(H3==1))
{
PORTB=t4; //0x09 0000 1001
OCR2=PWM;
OCR1A=0x00;
}
}
}
}

```

```
OCR1B=0x00;  
}
```

```
if((H1==0)&&(H2==0)&&(H3==1))
```

```
{  
PORTB=t5; //0x18 0001 1000
```

```
OCR2=PWM;
```

```
OCR1A=0x00;
```

```
OCR1B=0x00;
```

```
}
```

```
if((H1==1)&&(H2==0)&&(H3==1))
```

```
{
```

```
PORTB=t6; //0x12 0001 0010
```

```
OCR1A=PWM;
```

```
OCR2=0x00;
```

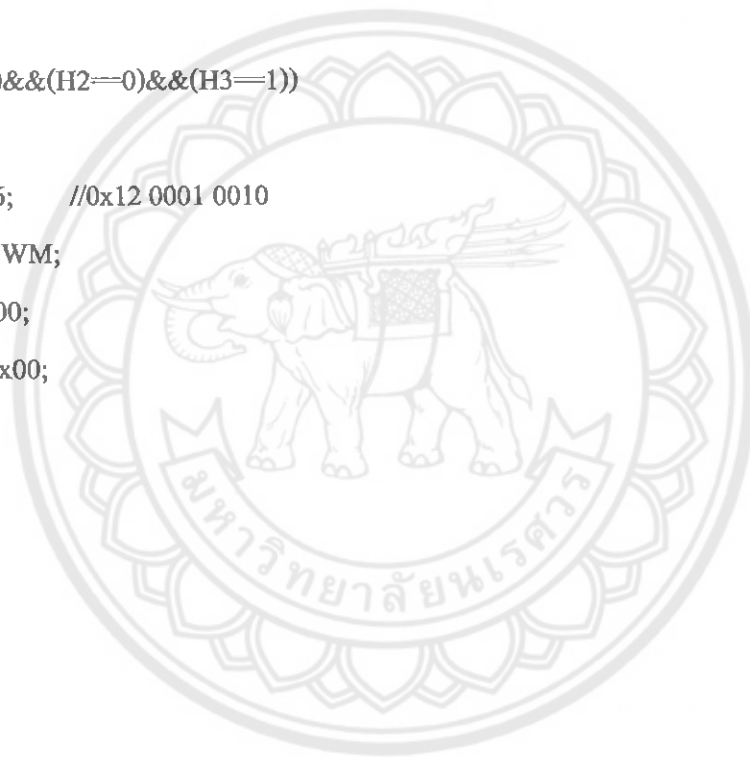
```
OCR1B=0x00;
```

```
}
```

```
}
```

```
}
```

```
}
```



ภาคผนวก (ข)

เครื่องโปรแกรมไมโครคอนโทรลเลอร์ AVR แบบ ISP

PX-400 AVR In-System programmer

คุณสมบัติทางเทคนิค

เชื่อมต่อกับพอร์ตอนุกรมของคอมพิวเตอร์ในการทำงาน หากคอมพิวเตอร์มีเฉพาะพอร์ต USB สามารถใช้ตัวแปลงสัญญาณ USB เป็นพอร์ตอนุกรม RS-232 มาต่อรวมด้วยได้ (รุ่นที่แนะนำ คือ UCON -232S ของ INEX)

- ทำการโปรแกรมไมโครคอนโทรลเลอร์ AVR ผ่านสาย ISP โดยสามารถอ่าน เขียน ลบ และป้องกันการอ่านได้
- ใช้ไฟเลี้ยง +5 V จากบอร์ดไมโครคอนโทรลเลอร์ AVR ที่ต้องการโปรแกรม
- ทำงานผ่านโปรแกรม AVR Prog ที่บรรจุภายในชุดซอฟต์แวร์ AVR Studio หรือใช้ซอฟต์แวร์ AvrOspII ในกรณีที่ไม่ต้องการทำงานผ่าน AVR Studio

การพัฒนาโปรแกรมไมโครคอนโทรลเลอร์ AVR สามารถใช้ได้ทั้งโปรแกรมภาษาแอสเซมบลีและภาษา C สำหรับในปัจจุบันนิยมใช้โปรแกรมภาษา C เป็นหลัก เนื่องจากมี C คอมไพเลอร์ ที่แจกฟรีและมีความสามารถในการทำงานสูง รวมถึง Atmel อันเป็นผู้ผลิตไมโครคอนโทรลเลอร์ AVR เองก็ได้พัฒนาชุดซอฟต์แวร์สำหรับการพัฒนาโปรแกรมทั้งภาษาแอสเซมบลีและ C ออกมาแจกฟรีด้วย ดังนั้นก่อนการใช้งานจึงต้องติดตั้งซอฟต์แวร์อันเป็นเครื่องในการพัฒนาโปรแกรมบนคอมพิวเตอร์ 3 รายการหลักๆดังนี้

1. AVR Studio เป็นซอฟต์แวร์สำหรับสร้างโปรแกรมควบคุม โดยภายใน AVR Studio ได้บรรจุเครื่องมือที่ช่วยอำนวยความสะดวกต่อการพัฒนาโปรแกรมภาษาแอสเซมบลีและ C ทั้งยังสามารถเชื่อมโยงการทำงานในส่วนต่างๆกับคอมไพเลอร์ที่กำหนดได้ด้วย โดย C คอมไพเลอร์ที่นำมาใช้ร่วมกันคือ WinAvr ทั้งยังสามารถติดต่อกับซอฟต์แวร์สำหรับโปรแกรมข้อมูลลงในไมโครคอนโทรลเลอร์ AVR ได้ด้วย ดังนั้นซอฟต์แวร์ AVR Studio จึงเหมาะสมอย่างยิ่ง ในการนำใช้พัฒนาโปรแกรม สำหรับไมโครคอนโทรลเลอร์ AVR โดย AVR Studio พัฒนาและเผยแพร่โดยไม่คิดมูลค่าโดย Atmel

2. Win AVR เป็นซอฟต์แวร์ C คอมไพเลอร์หรือตัวแปลโปรแกรมภาษา C สำหรับไมโครคอนโทรลเลอร์ AVR โดย WinAVR เป็นซอฟต์แวร์แบบ (OPEN SOURCE) พัฒนาต่อจาก GNU GCC คอมไพเลอร์ สามารถดูข้อมูลเพิ่มเติมรวมทั้งดาวน์โหลดซอฟต์แวร์ใหม่ๆจาก <http://sourceforge.net/projects/winavr/> เมื่อทำการติดตั้ง WinAVR แล้วจะสามารถเชื่อมโยงการ

ทำงานเข้ากับ AVR Studio ได้ ดังนั้นจึงสามารถทำการเขียนโปรแกรมภาษาซีบน AVR Studio แล้วทำการคอมไพล์โปรแกรมด้วย WinAVR ได้อย่างต่อเนื่อง โดยผลลัพธ์ของการคอมไพล์จะได้เป็นไฟล์นามสกุล .hex อันเป็นไฟล์รหัสภาษาเครื่องหรือที่เรียกว่า แมชชีน โค้ด โดยเป็นไฟล์ผลลัพธ์ที่ได้จากการพัฒนาสามารถนำไปดาวน์โหลดลงไมโครคอนโทรลเลอร์ต่อไปได้ทันที

3.ซอฟต์แวร์สำหรับดาวน์โหลดโปรแกรมหลังจากที่ทำการคอมไพล์โปรแกรมภาษา C แล้วไฟล์ผลลัพธ์ที่ได้จะเป็นไฟล์นามสกุล .hex ซึ่งจะต้องนำไปโปรแกรมหรือเขียนลงในหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ AVR สำหรับในชุด PX-400 สามารถใช้ได้ 2 ตัวคือ

1. AVR Prog เป็นผลงานของ Atmel โดยได้รับการติดตั้งไปพร้อมกับการติดตั้งซอฟต์แวร์ AVR Studio

2. AVR-Osp สามารถดาวน์โหลดเวอร์ชันใหม่ได้ที่ <http://esnips.com/web/AtmelAVR>

ขั้นตอนการติดตั้งโปรแกรม

1. การติดตั้งโปรแกรม AVR Studio

คอมพิวเตอร์ที่ใช้ควรติดตั้งระบบปฏิบัติการวินโดวส์ ME ขึ้นไป แต่ขอแนะนำให้ใช้กับวินโดวส์ XP จะสามารถใช้งานได้ดีกว่า ส่วนความต้องการทางฮาร์ดแวร์คือควรมีพอร์ตอนุกรมว่างอย่างน้อย 1 ช่อง อย่างไรก็ตาม หากมีเฉพาะพอร์ต USB แนะนำให้ใช้ตัวแปลงสัญญาณพอร์ต USB เป็นพอร์ตอนุกรม RS-232 มาใช้งานร่วมด้วย

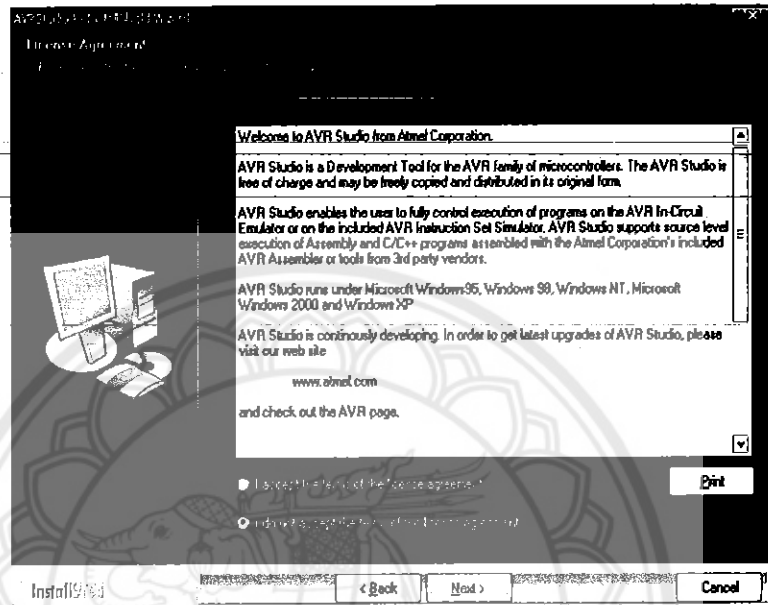
การติดตั้งโปรแกรมมีขั้นตอนโดยสรุปดังนี้

1.1 ดับเบิลคลิกที่ไฟล์ติดตั้งของโปรแกรม AVR Studio ดังรูป

1.2 หลังจากนั้นเข้าสู่หน้าต่างแรกของการติดตั้ง AVR Studio ดังรูป จากนั้นคลิกที่ปุ่ม Next



1.3 เข้าสู่หน้าต่างข้อตกลงเงื่อนไขลิขสิทธิ์และการใช้งานซึ่งจะมีรายละเอียดของคุณสมบัติของเครื่องคอมพิวเตอร์และระบบปฏิบัติการที่สามารถรองรับการทำงานของ AVR Studio รวมไปถึงชื่อบริษัทผู้ผลิตและอื่นๆ โดยให้เลือกช่อง I accept the terms of the license agreement จากนั้นคลิกปุ่ม Next

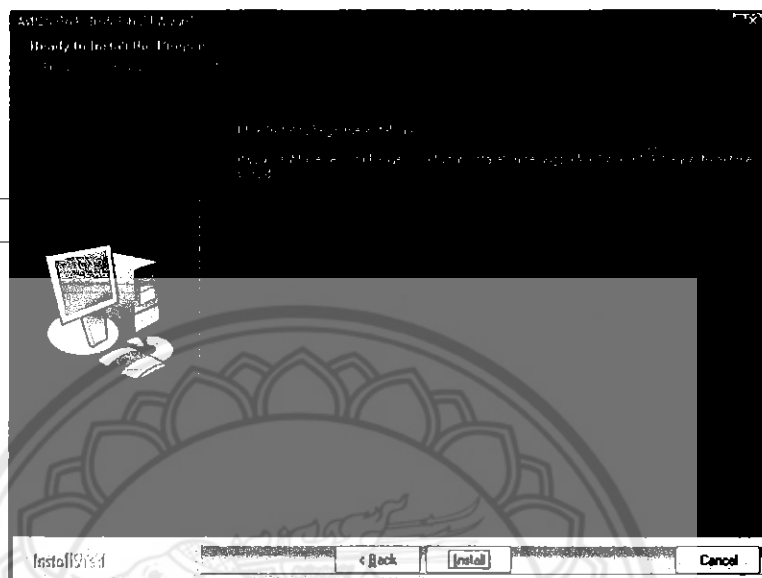


1.4 เข้าสู่หน้าต่างการติดตั้งในลำดับถัดไป ที่แสดงชื่อพาธของคอมพิวเตอร์ Studio ซึ่งผู้ติดตั้งสามารถเปลี่ยนพาธได้โดยคลิกปุ่ม Change แล้วกำหนดพาธใหม่ที่ต้องการจากนั้นคลิกปุ่ม Next



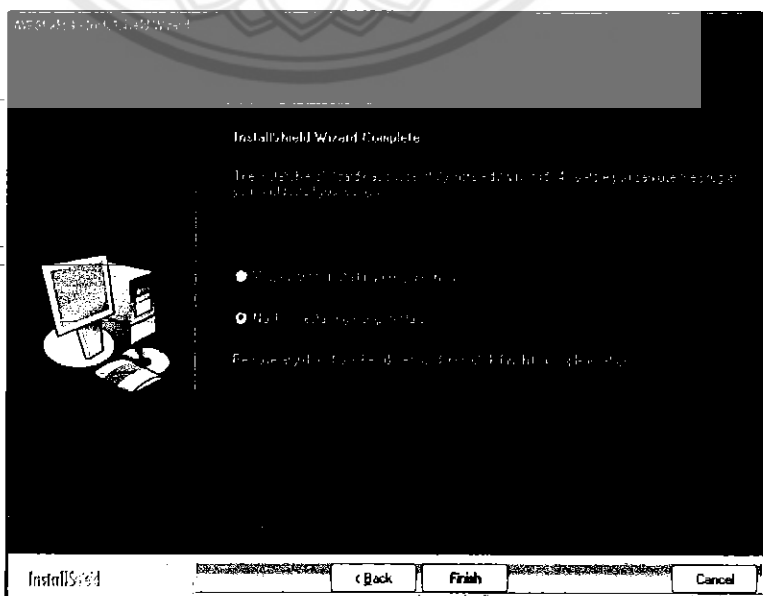
1.5 หลังจากนั้นเข้าสู่หน้าต่างการอัปเดตเฟิร์มแวร์ USB ให้คลิกที่ปุ่ม Next ผ่านไป เนื่องจากเฟิร์มแวร์ดังกล่าว ไม่มีการใช้งานใดๆกับเครื่องโปรแกรม PX-400

1.6 เข้าสู่หน้าต่างเริ่มต้นการติดตั้งโปรแกรมให้คลิกที่ปุ่ม Install



1.7 หลังจากนั้นจะเริ่มกระบวนการติดตั้งโปรแกรม AVR Studio รอจนกระทั่งการติดตั้งเสร็จสมบูรณ์

1.8 เมื่อการติดตั้งเสร็จสมบูรณ์ ให้คลิกที่ปุ่ม Finish เป็นอันเสร็จสิ้นการติดตั้งโปรแกรม AVR Studio



2. การติดตั้งโปรแกรม Win AVR

2.1 ดับเบิลคลิกที่ไฟล์ WinAVR-20050214-install.exe เพื่อติดตั้งโปรแกรม WinAVR

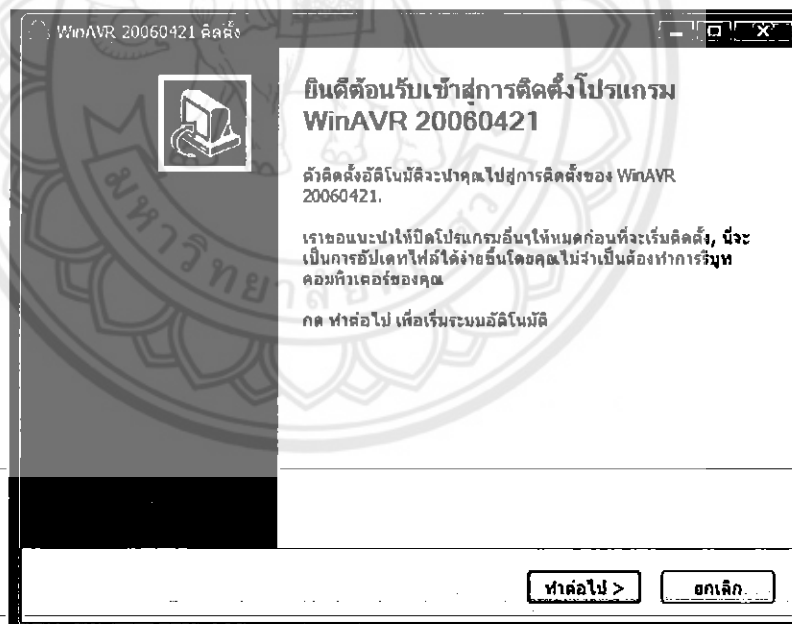
2.2 เข้าสู่หน้าต่างแรกของการติดตั้งโปรแกรม WinAVR เป็นการเลือกภาษาในการติดตั้ง

คลิกที่ปุ่ม OK

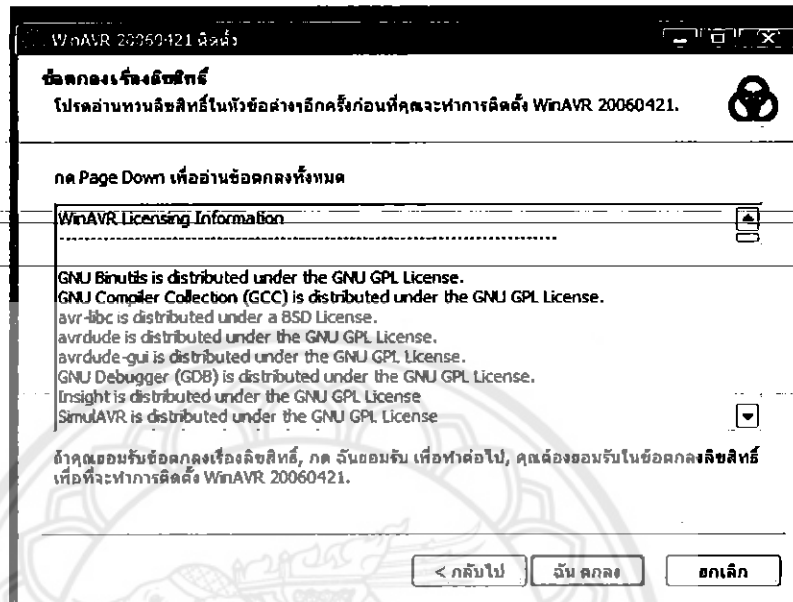


2.3 หลังจากนั้นเข้าสู่หน้าต่างแสดงรายละเอียดการดำเนินการติดตั้งโปรแกรม Win AVR

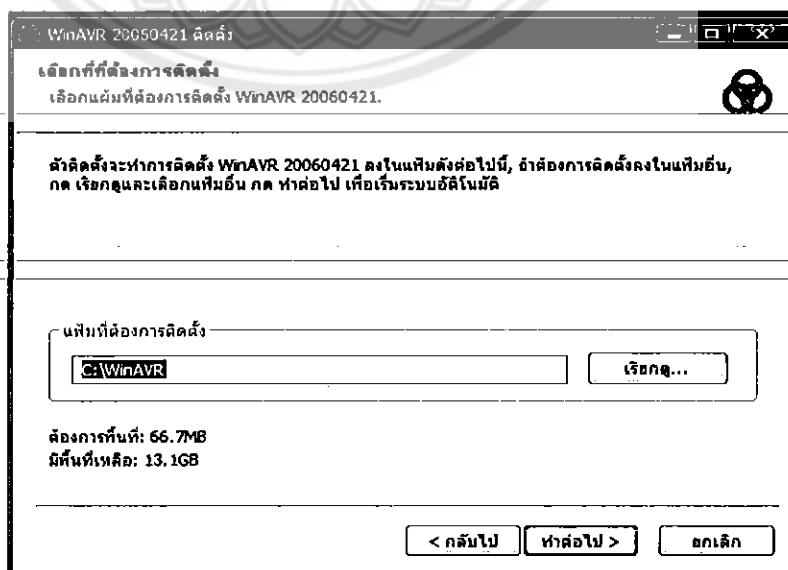
คลิกที่ปุ่มต่อไป



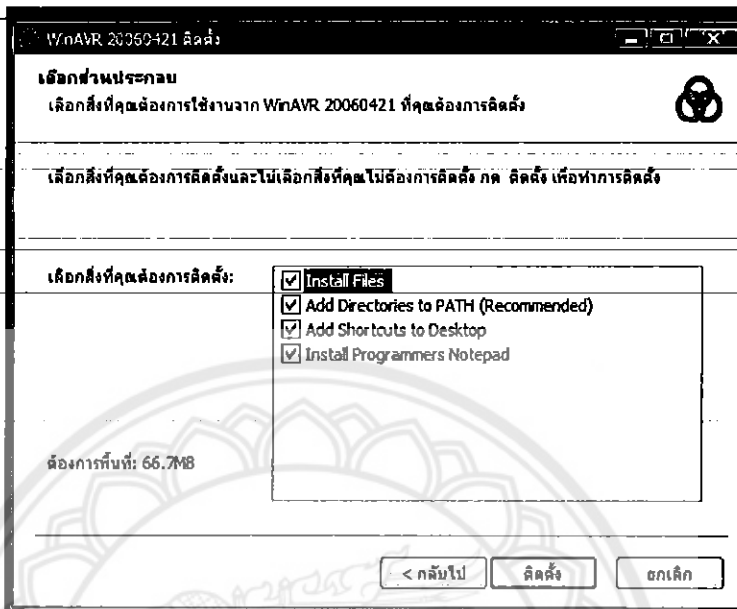
2.4 เข้าสู่หน้าต่างแสดงรายละเอียดและเงื่อนไขสำหรับการใช้งาน โปรแกรม WinAVR
หลังจากนั้นคลิกที่ปุ่มยอมรับ



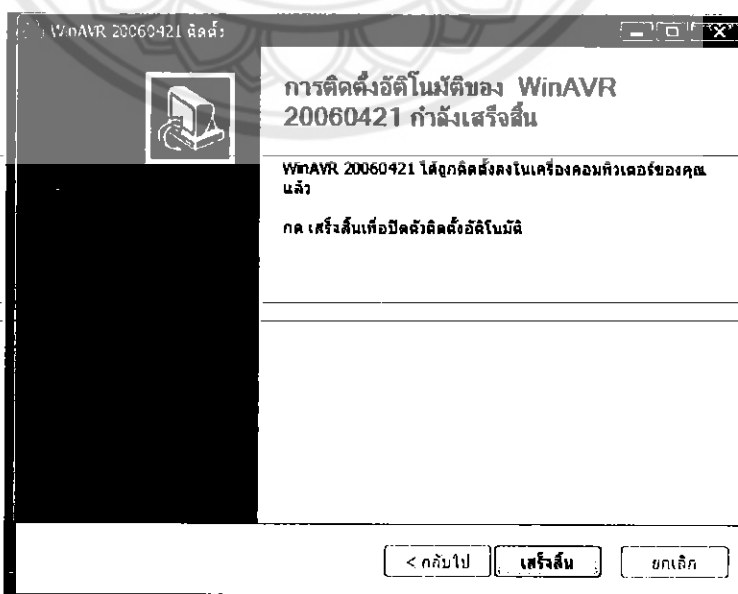
2.5 จากนั้นเข้าสู่หน้าต่างแสดงชื่อพาธของคอมพิวเตอร์ที่ต้องการติดตั้งโปรแกรม WinAVR และพื้นที่ที่ต้องการใช้สำหรับการติดตั้งโปรแกรม โดยผู้ติดตั้งสามารถเปลี่ยนพาธได้โดยการคลิกที่ปุ่ม เรียกว่า แล้วกำหนดพาธใหม่ที่ต้องการ จากนั้นคลิกที่ปุ่ม ต่อไป



2.6 ต่อไปเข้าสู่หน้าต่างการกำหนดคุณสมบัติพิเศษ ซึ่งผู้ติดตั้งสามารถเลือกรายการตามที่ต้องการแล้วคลิกที่ปุ่มติดตั้ง



2.7 หลังจากนั้นจะเริ่มกระบวนการติดตั้งโปรแกรม Win AVR และแสดงความทึ่งหน้าตามรูป หลังจากนั้นให้ผู้ติดตั้งรอนกระทั่งติดตั้งเสร็จสมบูรณ์ หลังจากนั้นให้คลิกที่ปุ่มเสร็จสิ้นเป็นอันว่าติดตั้งโปรแกรม Win AVR เสร็จสมบูรณ์



3. ติดตั้งซอฟต์แวร์สำหรับการโปรแกรม

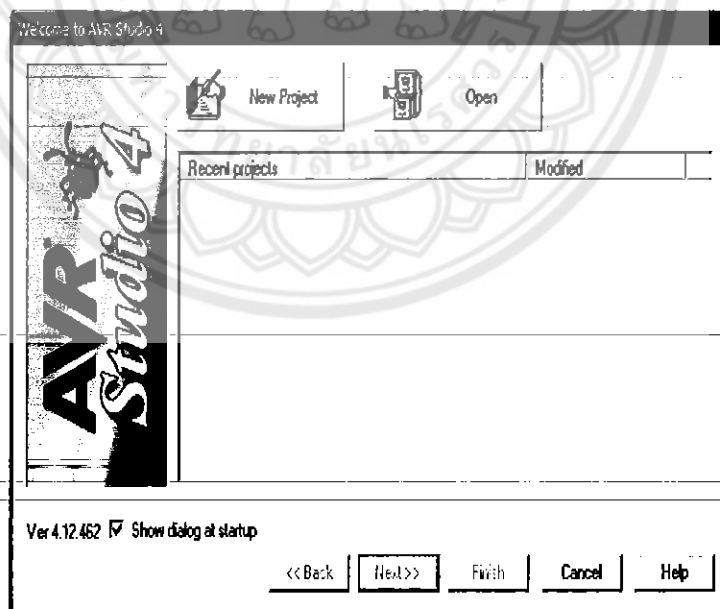
3.1 AVR Prog สำหรับ AVR Prog จะได้รับการติดตั้งไว้เรียบร้อยแล้วหลังจากที่ติดตั้ง AVR Studio เสร็จสมบูรณ์

3.2 AVR-Osp-II-4.00 เพียงกดคลิกไฟล์ AVR-Osp-II_400.zip ลงในฮาร์ดดิสก์ จากนั้นทำการขยายไฟล์ .zip นี้ จะได้ไฟล์เตอร์ AVR-Osp-II_400 ภายในจะมีไฟล์หลักๆ 3 ไฟล์คือ AVR-OspII.exe อันเป็นไฟล์ที่เรียกใช้งาน AVR-OspII.ini เป็นไฟล์เก็บค่าการทำงาน และ DeviceList.dat เป็นไฟล์ข้อมูลของไมโครคอนโทรลเลอร์ AVR แต่ละตัวที่ใช้งานร่วมด้วยได้

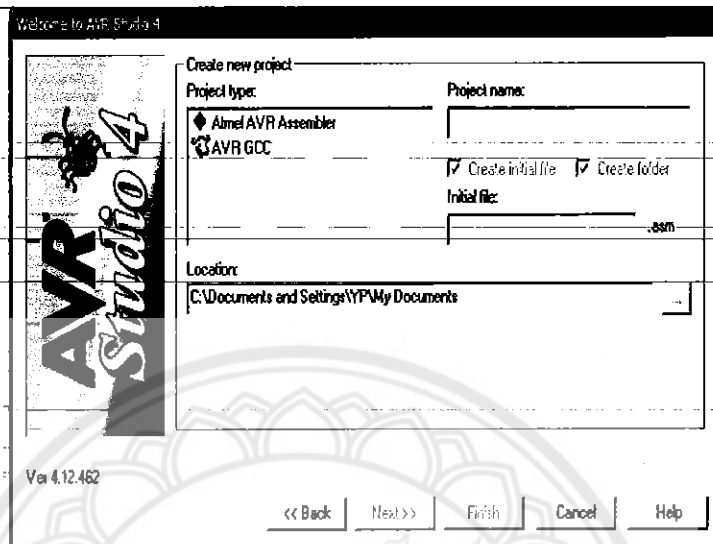
4. การใช้งาน AVR Prog ผ่านทางซอฟต์แวร์ AVR Studio ในโปรแกรมไมโครคอนโทรลเลอร์

ในการทำงานปกติจะไม่สามารถเปิดโปรแกรม AVR Prog ได้โดยตรง ดังนั้นในการทำงานต้องเปิดโปรแกรม AVR Studio ขึ้นมาก่อน โดย AVR Studio ที่ใช้งานกับ PX-400 ได้โดยจะต้องเป็นเวอร์ชันตั้งแต่ 4.12 Build 462 ขึ้นไป

4.1 ขั้นตอนแรกในการใช้ AVR Studio คือ การสร้างผังงานหรือ Project โดยปกติเมื่อเปิดโปรแกรมครั้งแรก โปรแกรมจะแสดงเมนูสำหรับการเรียกใช้งาน Project เดิมที่เคยใช้งานก่อนหน้านี้ แต่สำหรับการ สร้าง Project ใหม่สามารถคลิกที่ปุ่ม new project

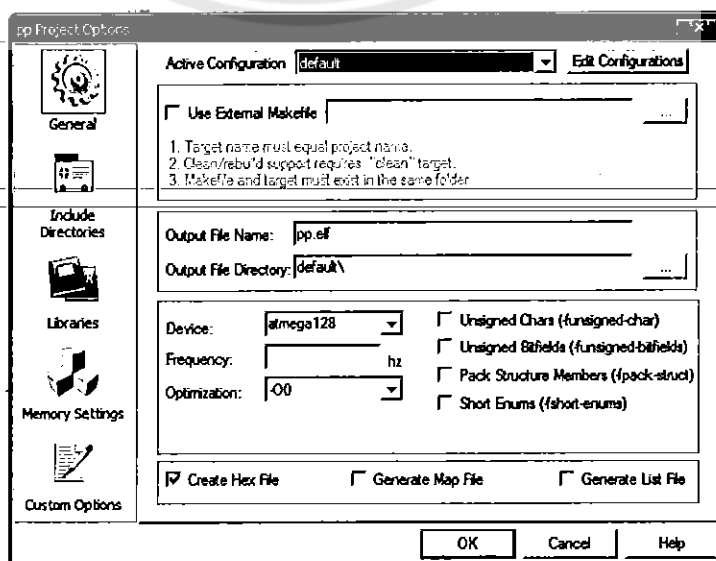


4.2 สำหรับกรณีที่ซอฟต์แวร์ไม่ได้แสดงหน้าต่างขึ้นมาสามารถเรียกคำสั่งการสร้าง Project ได้จากเมนู Project >New Project จะปรากฏหน้าต่างสำหรับการสร้าง Project ขึ้นมา เช่นเดียวกัน



4.3 สำหรับในการพัฒนานี้จะใช้งานภาษา C ดังนั้นการสร้าง Project จะต้องเลือกชนิด Project Type เป็น AVR GCC จากนั้นจะต้องตั้งชื่อ Project แล้วกาเครื่องหมายถูกที่หน้า หัวข้อ create initial file เพื่อให้สร้างไฟล์source codeขึ้นมาโดยชื่อของsource codeจะเป็นชื่อเดียวกับ Project โดยอัตโนมัติ แต่สามารถเปลี่ยนชื่อของsource codeเป็นชื่ออื่นได้หลังจากนี้สามารถคลิกที่ปุ่ม Finish เพื่อจบขั้นตอนการสร้าง Project

4.4 ทำการกำหนดตำแหน่งไฟล์ผนวก (Include file) ต่างๆ ที่ต้องใช้งานในไฟล์ Project โดยไปที่เมนู Project เลือก Configuration Option จะปรากฏหน้าต่างดังรูป



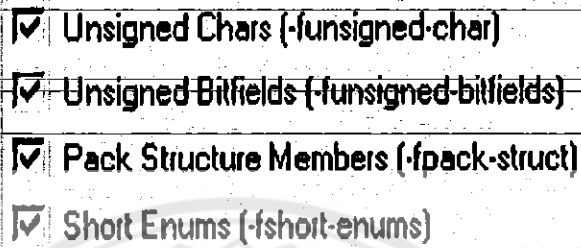
4.5 ในหัวข้อ General จะต้องกำหนดรายละเอียดต่างๆดังนี้

Device : กำหนดเป็น ATMEGA16 (หรือเบอร์อื่นๆ ตามต้องการ)

Frequency : กำหนดความถี่เป็น 16000000 Hz

กาเครื่องหมายถูกหน้าหัวข้อต่างๆดังนี้

คลิกถูกที่หน้าช่อง create hex file ด้วย

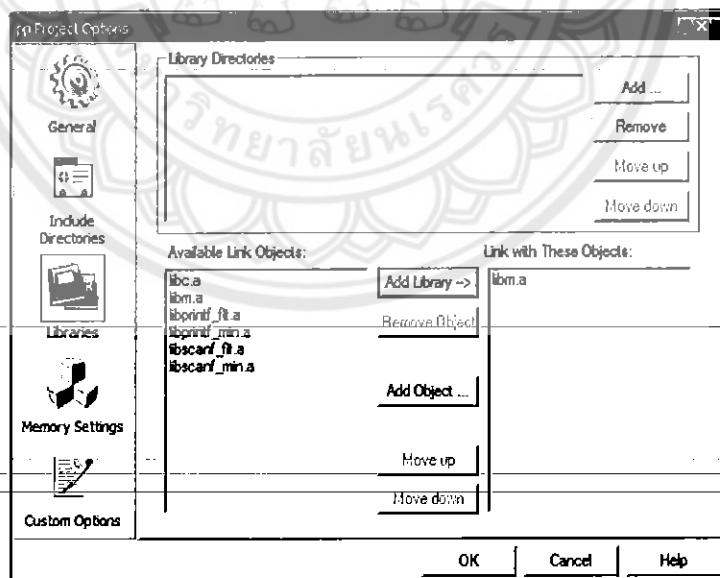


4.6 ที่หัวข้อ Include Directories กดปุ่ม Add.. เพื่อเพิ่มโคเร็คทอรีเพื่อเก็บไฟล์ไลบรารี

เอาไว้

4.7 ที่หัวข้อ libraries เลือก libm.a แล้วคลิกที่ Add library เพื่อเพิ่มไลบรารีตัวกระทำทาง

คณิตศาสตร์



4.8 จากนั้นคลิกที่ปุ่ม OK เพื่อสิ้นสุดการตั้งค่า

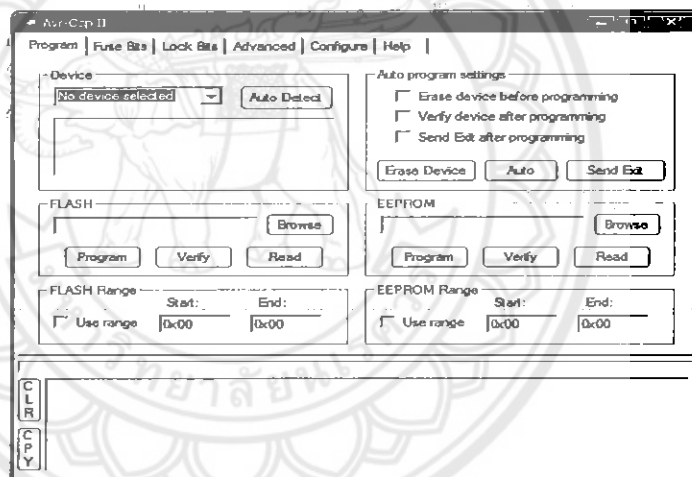
5. การใช้งาน AVR-Osp II ในการโปรแกรมไมโครคอนโทรลเลอร์ AVR

Avr-Osp II เป็นซอฟต์แวร์ที่ใช้ในการโปรแกรมไมโครคอนโทรลเลอร์ AVR แจกฟรีที่ใช้งานได้ดีโดยเฉพาะอย่างยิ่งกับการรองรับเบอร์ของไมโครคอนโทรลเลอร์ AVR ที่มีมากกว่า AVR Prog หากแต่อาจไม่สะดวกเท่ากับ AVR Prog ในกรณีที่พัฒนาโปรแกรมในลักษณะไฟล์ Project เพราะผู้ใช้งานสามารถเรียกใช้งาน AVR Prog ได้จากเมนูคำสั่งของ AVR Studio โดยตรง

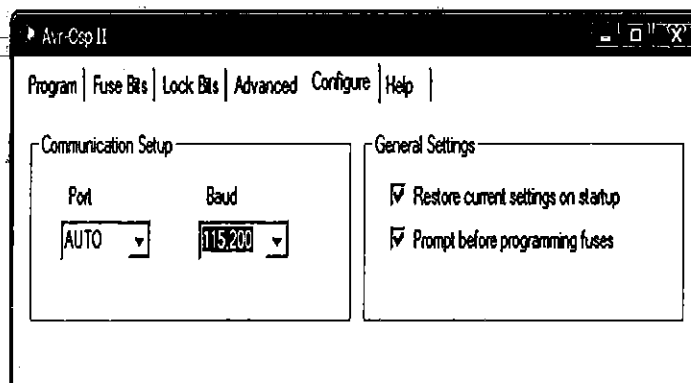
แต่ Avr-Osp II ก็มีข้อดีตรงที่ไฟล์มีขนาดเล็กสามารถเรียกใช้งานได้อย่างอิสระ โดยไม่ต้องเปิด AVR Studio ก่อน อย่างไรก็ตามคอมพิวเตอร์ที่ทำงานร่วมกับ Avr-Osp II ควรติดตั้งระบบปฏิบัติการวินโดวส์ XP หากใช้งานกับวินโดวส์ ME หรือ 98SE จะทำให้การเรียกปิดไฟล์ทำได้ยาก จึงไม่แนะนำให้ใช้งานกับวินโดวส์ ME หรือ 98SE

5.1 เชื่อมต่อ PX-400 กับคอมพิวเตอร์และบอร์ดไมโครคอนโทรลเลอร์ AVR จากนั้นจ่ายไฟให้เรียบร้อย

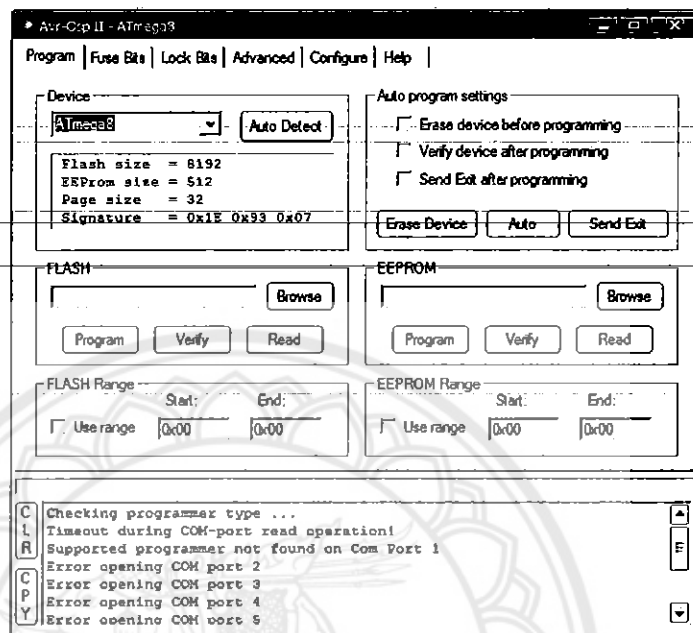
5.2 เปิดโปรแกรม Avr-Osp II จะปรากฏหน้าต่างหลักของโปรแกรมดังนี้



5.3 เลือกไปที่แท็บ Configure เพื่อเลือกพอร์ตและอัตราบอดสำหรับการติดต่อ โดยต้องเลือกอัตราบอดเป็น 115,200 เท่านั้น ส่วนตำแหน่งพอร์ตให้กำหนดเป็น Auto



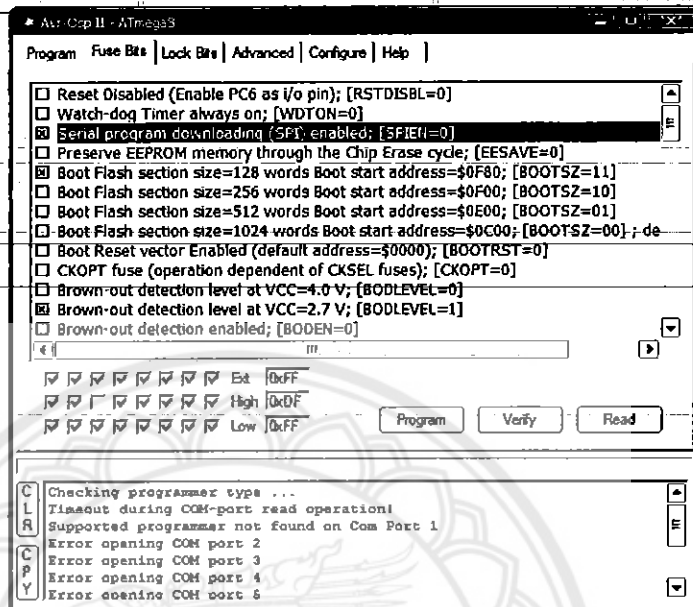
5.4 จากนั้นกลับมาที่หน้าต่างหลัก โดยเลือกที่แท็บ Program จากนั้นคลิกที่ปุ่ม Auto Detect ในกรอบ Device รอสักครู่สังเกตการทำงาน ได้ที่ช่องแสดงสถานะ



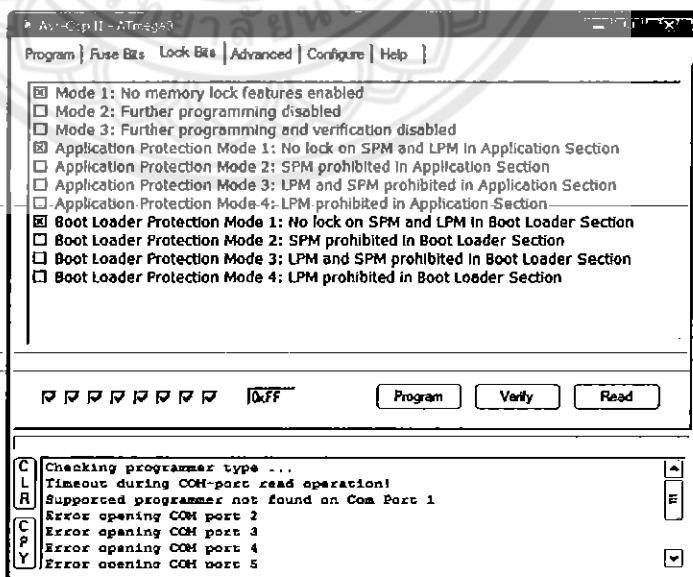
ที่กรอบ Device จะปรากฏเบอร์ของไมโครคอนโทรลเลอร์ AVR พร้อมกับข้อมูลจำเพาะ ในขณะที่ช่องแสดงสถานะจะแจ้งการทำงาน โดยเริ่มจาก

- ตรวจสอบชนิดของเครื่องโปรแกรม (Checking programmer type...)
- ตรวจพบว่าเป็นเครื่องโปรแกรมแบบในวงจร (Found AVR ISP)
- เข้าสู่โหมดโปรแกรม (Entering programming mode)
- ตรวจสอบข้อมูลจำเพาะของไมโครคอนโทรลเลอร์ AVR
(Signature = xxxx-xxxx-xxxx)
- ออกจากโหมดโปรแกรม (Leaving programming mode)

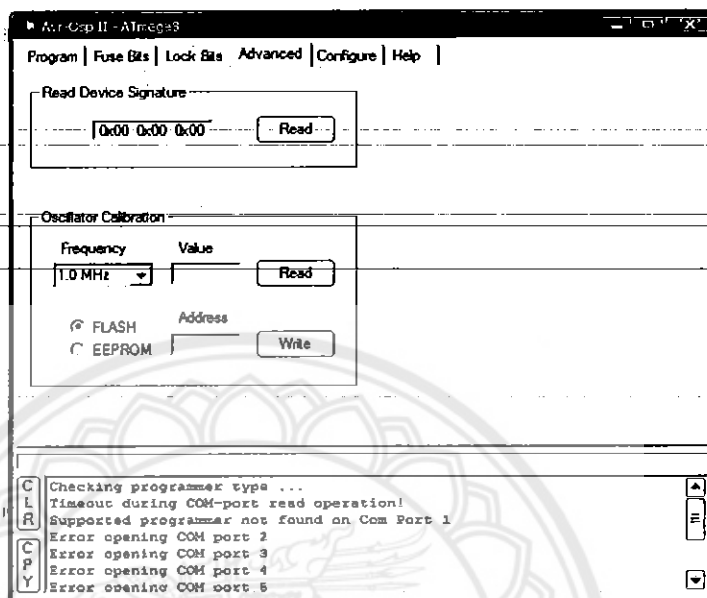
5.5 เลือกที่แท้ Fuse Bits เพื่ออ่านค่า Configuration หรือ Fuse ของ ไมโครคอนโทรลเลอร์ AVR



5.6 เลือกที่แท้ Lock Bits เพื่อกำหนดการป้องกันการอ่านข้อมูลออกจากตัวชิพหลังจาก โปรแกรมแล้ว



5.7 เลือกที่แท็บ Advance เพื่อตรวจสอบค่าจำเพาะของไมโครคอนโทรลเลอร์ โดยกลุ่ม Read ในช่อง Read Device Signature เพื่อเป็นการยืนยันว่าติดต่อกับชิพที่ต้องการ โปรแกรมได้



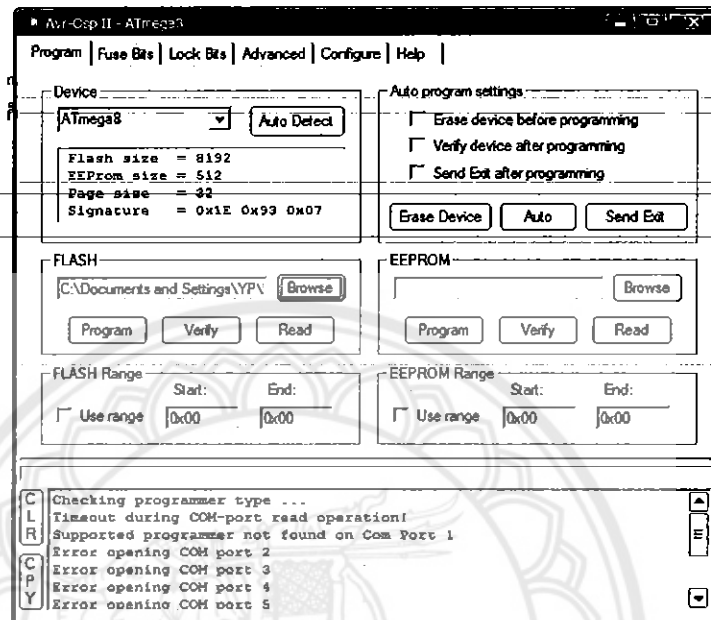
5.8 กลับไปที่หน้าต่างหลักทำการเลือกไฟล์นามสกุล .hex ที่ต้องการ โปรแกรมในกรอบ flash

5.9 ที่กรอบ Auto Program Setting สามารถกำหนดกระบวนการ โปรแกรมแบบอัตโนมัติได้ โดยมีให้เลือก 3 กระบวนการคือ

- (1) Erase device before programming : ลบข้อมูลก่อน โปรแกรมตัวชิพ
- (2) Verify device after programming : ตรวจสอบข้อมูลหลังการ โปรแกรมตัวชิพ
- (3) SendExit after programming : ออกจากโปรแกรม Avr-Osp II หลังจาก

โปรแกรมตัวชิพแนะนำให้เลือก Erase device before programming อย่างน้อย 1 ข้อ

5.10 การโปรแกรมทำได้ 2 ทางคือคลิกปุ่ม Program ในกรอบ Flash และคลิกปุ่ม Auto ในกรอบ Auto program setting ปรากฏแถบแสดงสถานะทำงานขึ้นเหนือช่องแสดงสถานะ พร้อมกับนั้นที่ช่องแสดงสถานะก็จะปรากฏข้อความเพื่อแจ้งสถานะการทำงาน ไปพร้อมกันด้วย



5.11 ข้อจำกัดประการหนึ่งของโปรแกรมนี้คือไม่สามารถบันทึกไฟล์ .hex ได้และไม่มีการแสดงข้อมูลรายละเอียดของไฟล์ .hex ที่นำมาโปรแกรม

ภาคผนวก (ค)
รายการอุปกรณ์อิเล็กทรอนิกส์

ลำดับที่	อุปกรณ์	จำนวน
1	ไมโครคอนโทรลเลอร์ เบอร์ ATMEGA8	1
2	Gate Drive เบอร์ IR 2112	3
3	มอสเฟต เบอร์ IRF 3710	6
4	ไดโอด 1N4148	3
5	ตัวต้านทาน 15 โอห์ม	10
6	ตัวต้านทาน 1 กิโลโอห์ม	3
7	คาปาซิเตอร์ 0.47 uF	3
8	คาปาซิเตอร์ 1000 uF	2
9	คาปาซิเตอร์ 0.1 uF	3
10	สวิตช์ปุ่มกด	3
11	Regulator เบอร์ 7805	1
12	Regulator เบอร์ 7812	1
13	Regulator เบอร์ 7824	1
14	Bridge Rectifier	1
15	ฟิวส์ขนาด 5 แอมป์	1
16	Connector แบบต่างๆ	

ประวัติผู้เขียนโครงการ



ชื่อ นาย ชaturวิทย์ ปารี
ภูมิลำเนา 283/2 ม.8 ต.ทุ่งเสลี่ยม อ.ทุ่งเสลี่ยม จ.สุโขทัย
64150

ประวัติการศึกษา

- จบมัธยมศึกษาจาก โรงเรียนทุ่งเสลี่ยมชนูปถัมภ์
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนครพนม

E-mail : geer_engineering@hotmail.com



ชื่อ นาย ยุทธพงษ์ พรหมสินอง
ภูมิลำเนา 10 ม.1 ต.น้ำกุ่ม อ.นครไทย จ.พิษณุโลก 65120

ประวัติการศึกษา

- จบมัธยมศึกษาจาก โรงเรียนนครชุมพิทยา รัชมังคลาภิเษก
- ปัจจุบันกำลังศึกษาในระดับปริญญาตรีชั้นปีที่ 4 สาขาวิชาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนครพนม

E-mail : p_overclock@hotmail.com