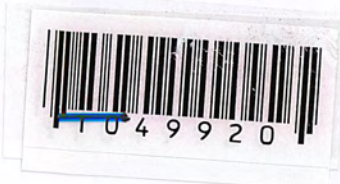


เตาหุงต้มเหนียวน่า
INDUCTION COOKING



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
ภาควิชาวิศวกรรมไฟฟ้า
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2545

เลขหมู่.....
เลขทะเบียน.....49920
วันเดือนปี..... 2 เม.ย. 2545

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

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ประกอบการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปีการศึกษา 2545

ตำราหุงต้มเหนี่ยวนำ

INDUCTION COOKING



อาจารย์ที่ปรึกษา

ผศ.ดร.วิจิตร กิณเรศ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2545


ภาควิชาวิศวกรรมไฟฟ้า

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง เตาหุงต้มเหนี่ยวนำ
INDUCTION COOKING

ผู้จัดทำ

- | | | | |
|-----------------|------------|--------------|----------|
| 1. นายกิตติพงษ์ | สุทธิมุต | รหัสประจำตัว | 43015107 |
| 2. นายณรงค์ชัย | ทศพร | รหัสประจำตัว | 43015114 |
| 3. นายบุญมี | นิมมวล | รหัสประจำตัว | 43015121 |
| 4. นายพินิจ | หยกส่องแสง | รหัสประจำตัว | 43015129 |


อาจารย์ที่ปรึกษา
(ผศ.ดร.วิจิตร กิณเรศ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เตาหุงต้มเหนียวน้ำ

นายกิตติพงษ์	สุทธิมูล	43015107
นายณรงค์ชัย	ทศพร	43015114
นายบุญมี	นันทนวล	43015121
นายพินิจ	หยกส่องแสง	43015129
ผศ.ดร. วิจิตร	กิมเรศ	อาจารย์ที่ปรึกษา

ปีการศึกษา 2545

บทคัดย่อ

โครงการนี้เป็นนำเสนอการออกแบบและการสร้างหม้อต้มน้ำ อาศัยการเหนียวน้ำความถี่สูง เพื่อนำไปใช้งานด้านการหุงต้มอาหารในร้านอาหารซึ่งเป็นหม้อต้มน้ำที่มีขนาดเล็กกระทัดรัดให้ประสิทธิภาพสูงอีกทั้งประหยัดพลังงาน โดยมีกำลังไฟฟ้าขาเข้าประมาณ 1.2 กิโลวัตต์ โครงการนี้จะเป็นการประยุกต์ใช้งาน เรื่องการถ่ายเทความร้อนโดยใช้หลักการรีโซแนนซ์ซึ่งจะใช้ไมโครคอนโทรลเลอร์ PIC 18F458 สร้างสัญญาณความถี่ 29 - 31 กิโลเฮิรตซ์ ขับวงจรฮาร์ฟบริดจ์ อินเวอร์เตอร์ โดยมีไอจีบีที เป็นอุปกรณ์สวิตช์และสามารถควบคุมระดับความร้อนได้โดยใช้หลักการควบคุมกระแส



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

INDUCTION COOKING

Kittiphong Suttimul

Narongchai Thodsaphorn

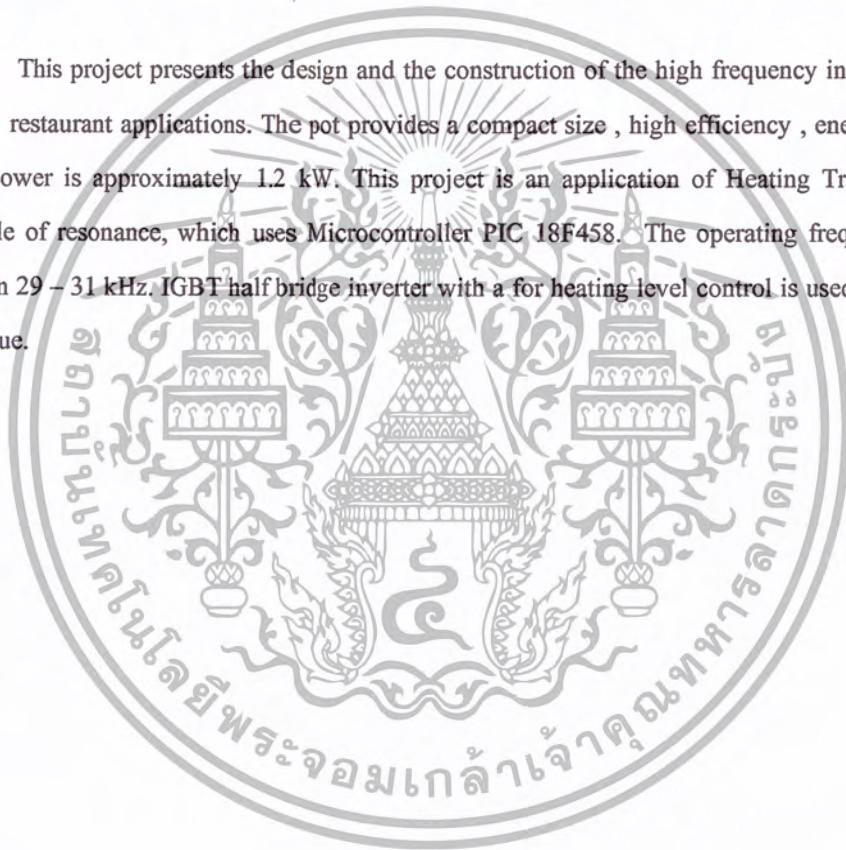
Boonmee Nimnual

Pinit Yoksongsang

Assist.Prof.Dr. Vijit Kinares Advisor

ABSTRACT

This project presents the design and the construction of the high frequency induction cooking pot for restaurant applications. The pot provides a compact size , high efficiency , energy saving. The input power is approximately 1.2 kW. This project is an application of Heating Transfer based on principle of resonance, which uses Microcontroller PIC 18F458. The operating frequency is varied between 29 – 31 kHz. IGBT half bridge inverter with a for heating level control is used current control technique.



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญาานิพนธ์ฉบับนี้สำเร็จลุล่วงได้อย่างดี ด้วยคำแนะนำและคำปรึกษาเกี่ยวกับการดำเนินการศึกษาและวิจัยจาก ผศ.ดร. วิจิตร กิณเรศ ซึ่งเป็นอาจารย์ผู้ควบคุมปริญญาานิพนธ์ ผู้วิจัยรู้สึกซาบซึ้งในความอนุเคราะห์จากท่านและขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบคุณพี่ทศ ที่ให้ความช่วยเหลือให้คำแนะนำทางการเขียน โปรแกรมและปรับปรุงแก้ไขอุปกรณ์ ขอบขอบคุณทีดี ที่ให้แนวทางการทำงาน ที่คอยดูแลเอาใจใส่ ให้ความสนทนสนม พี่แก้วที่ให้ยืมอุปกรณ์เครื่องใช้เครื่องมือ พี่โก้ พี่กร พี่ดี และพี่ๆ ป.โททุกๆ ท่านที่ให้คำแนะนำตลอดจนอุปกรณ์เครื่องมือเครื่องใช้อื่น ตลอดจนเพื่อน ๆ พี่ ๆ ทุกคนที่ได้ช่วยเหลือทั้งร่างกายและแรงใจจนโครงการนี้สำเร็จไปได้ด้วยดี

สุดท้ายนี้ขอขอบพระคุณ บิดา มารดา ที่เป็นผู้สนับสนุนและให้กำลังใจตั้งแต่เริ่มต้นจนสำเร็จสมความตั้งใจ

คุณค่าและประโยชน์อันพึงมีจากปริญญาานิพนธ์ฉบับนี้ ผู้จัดทำขอขอบแต่ผู้มีพระคุณทุกท่าน



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

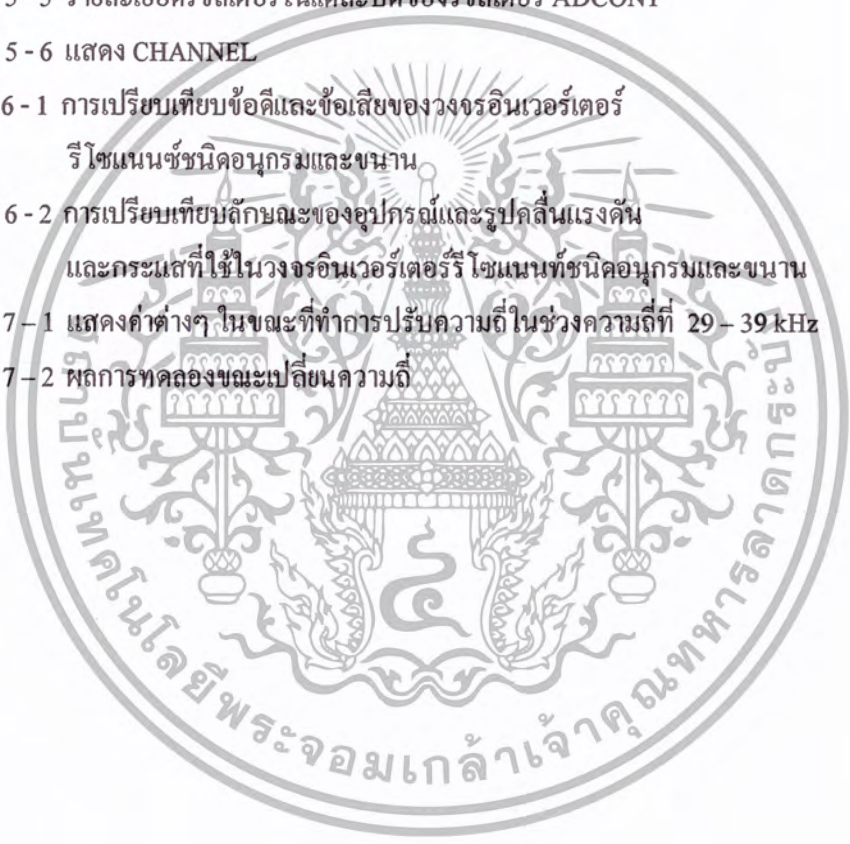
	หน้า
บทคัดย่อ	I
ABSTRACT	II
กิตติกรรมประกาศ	III
สารบัญ	IV
สารบัญตาราง	VII
สารบัญภาพ	VIII
บทที่ 1 บทนำ	1
1.1 ความสำคัญและที่มา	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตของงาน	2
1.4 วิธีการดำเนินงาน	2
บทที่ 2 ทฤษฎีหลักการให้ความร้อนเหนี่ยวนำ	3
2.1 บทนำ	3
2.2 พื้นฐานของการเกิดความร้อน โดยการเหนี่ยวนำความถี่สูง	3
บทที่ 3 หลักการรีโซแนนซ์	10
3.1 บทนำ	10
3.2 รีโซแนนซ์อนุกรม	10
3.2.1 แรงดันที่รีโซแนนซ์	12
3.2.2 พลังงานที่ส่งผ่านระหว่าง L และ C	13
3.2.3 Q แฟกเตอร์ของวงจรอนุกรม RLC	13
3.3 รีโซแนนซ์ขนาน	14
3.3.1 Q แฟกเตอร์ของวงจรขนาน RLC	17
บทที่ 4 เทคโนโลยีไอจีบีที	18
4.1 ความรู้พื้นฐานเกี่ยวกับ ไอจีบี	18
4.2 สัญลักษณ์และ โครงสร้างของ ไอจีบีที	18
4.3 สภาวะนำกระแส	20
4.4 สภาวะหยุดนำกระแส	21
4.5 การแลตช์ใน ไอจีบีที	22
4.6 การป้องกันการแลตช์	23
4.7 วงจรสมมูลของ ไอจีบีที	23
4.8 ลักษณะการสวิตช์	25
4.9 พื้นที่การทำงานที่ปลอดภัย	26

	หน้า
บทที่ 5 ไมโครคอนโทรลเลอร์ และการเขียนโปรแกรม	27
5.1 บทนำ	27
5.2 แนวทางการพัฒนาโปรแกรมในงานไมโครคอนโทรลเลอร์	27
5.2.1 พัฒนาในรูปแบบ Single Board	27
5.2.2 พัฒนาในรูปแบบ Embedded Board ร่วมกับชิพแบบ Softchip	28
5.2.3 พัฒนาด้วยเครื่องมือ Eprom Emulator	28
5.2.4 พัฒนาด้วยเครื่องโปรแกรมตัว Single Chip	29
5.2.5 พัฒนาด้วยเครื่องโปรแกรมผ่าน SPI-Port	29
5.3 PIC	30
5.3.1 ความเร็วของ PIC	30
5.3.2 หน่วยความจำของ PIC	30
5.3.3 สถาปัตยกรรมของ PIC	30
5.4 ชนิดของ PIC	30
5.4.1 OTP	30
5.4.2 EPROM	31
5.4.3 EEPROM / Flash	31
5.5 PICเบอร์ต่างๆ	32
5.6 สรุป concept สถาปัตยกรรมของ PIC	36
5.7 โปรแกรม MPLAB C18	37
5.8 การเขียนโปรแกรมด้วยภาษาซี	37
5.9 โครงสร้างของภาษาซี	39
5.10 ตัวแปรค่าคงที่	40
5.11 ตัวดำเนินการในภาษาซี	41
5.11.1 ตัวดำเนินการที่กระทำกับตัวถูกกระทำตัวเดียว	43
5.11.2 ตัวดำเนินการที่กระทำกับตัวถูกกระทำสองตัว	43
5.12 ประโยคควบคุมในภาษาซี	44
5.12.1 ประโยค IF / ELSE	44
5.12.2 ประโยค switch	46
5.12.3 ประโยค for	46
5.12.4 ประโยค while	47
5.12.5 ประโยค do - while	48
5.13 อาร์เรย์ พอยน์เตอร์ และ สตริงเจอร์	49
5.14 โปรแกรมควบคุมการทำงาน INDUCTION COOKING	52

	หน้า
5.14.1 โปรแกรมควบคุมสวิทช์กด(KEYPACK)	52
5.14.2 โปรแกรมกำเนิดเสียงจากลำโพงขนาดเล็ก(PIZZO)	55
5.14.3 โปรแกรมควบคุมการปิดเปิดพัดลมระบายความร้อน	56
5.14.4 โปรแกรมสร้างคลื่นรูปสี่เหลี่ยมด้วยโหมด CCP	56
5.14.5 โปรแกรมแปลงสัญญาณอะนาลอกเป็นดิจิทัล	62
5.15 โปรแกรมที่ใช้ควบคุม INDUCION COOKING	69
บทที่ 6 การออกแบบและสร้างเตาหุงต้มเหนี่ยวนำ	79
6.1 วงจรกำลัง	79
6.1.1 วงจรแปลงสัญญาณไฟฟ้ากระแสสลับให้เป็นสัญญาณไฟฟ้ากระแสตรง	79
6.1.2 วงจรกรองแรงดันให้เรียบ (C-filter)	80
6.1.3 วงจรอินเวอร์เตอร์	80
6.1.4 โทลครีโซแนนซ์	81
6.1.5 EMI FILTER	84
6.2 ชุดควบคุมอินเวอร์เตอร์	84
6.2.1 วงจรควบคุม (control circuit)	84
6.2.2 วงจรปรับแต่งเวลาเดดไทม์(Dead Time)	85
6.2.3 วงจรขยายสัญญาณขับ ไอจีบีที (Gate Drive Circuit)	86
6.2.4 วงจรป้องกันกระแสเกิน	87
บทที่ 7 ผลการทดลอง	88
7.1 ผลการทดลองทางด้านอินพุทขณะเปลี่ยนความถี่ต่างๆ	88
7.2 ผลการทดลองทางด้านเอาต์พุทขณะปรับความถี่	100
บทที่ 8 บทวิจารณ์และสรุป	102
8.1 จุดเด่นของโครงการ	102
8.2 ปัญหาที่พบและข้อเสนอแนะเพื่อพัฒนา	102
8.3 ประโยชน์ที่ได้รับ	103
ภาพผนวก	
เอกสารอ้างอิง	

สารบัญตาราง

	หน้า
ตารางที่ 2-1 คุณสมบัติทางไฟฟ้าของวัสดุต่าง ๆ ที่ใช้ในภาชนะหุงต้มที่ความถี่ 30 kHz	8
ตารางที่ 5-1 ประเภทของข้อมูลในการประกาศ	40
ตารางที่ 5-2 แสดงความสัมพันธ์ของค่าปริสกลเลอ์ในไทเมอร์ 2 ที่มีต่อการกำหนด ค่าความละเอียดต่ำสุดในหน่วยของเวลาของสัญญาณ PWM	60
ตารางที่ 5-3 แสดงความสัมพันธ์ของปริสกลเลอ์และค่ารีจิสเตอร์ในไทเมอร์ 2 ที่มีต่อ การกำหนดรายละเอียดสูงสุดของสัญญาณ PWM ที่ความถี่ต่างๆ	60
ตารางที่ 5-4 อธิบายค่ารีจิสเตอร์แต่ละตัว	62
ตารางที่ 5-5 รายละเอียดรีจิสเตอร์ในแต่ละบิตของรีจิสเตอร์ ADCON1	63
ตารางที่ 5-6 แสดง CHANNEL	65
ตารางที่ 6-1 การเปรียบเทียบข้อดีและข้อเสียของวงจรอินเวอร์เตอร์ รีโซแนนซ์ชนิดอนุกรมและขนาน	82
ตารางที่ 6-2 การเปรียบเทียบลักษณะของอุปกรณ์และรูปคลื่นแรงดัน และกระแสที่ใช้ในวงจรอินเวอร์เตอร์รีโซแนนซ์ชนิดอนุกรมและขนาน	83
ตารางที่ 7-1 แสดงค่าต่างๆ ในขณะที่ทำการปรับความถี่ในช่วงความถี่ที่ 29 – 39 kHz	99
ตารางที่ 7-2 ผลการทดลองขณะเปลี่ยนความถี่	101



สารบัญภาพ

	หน้า
รูปที่ 2-1 หลักการให้ความร้อนเหนี่ยวนำ	3
รูปที่ 2-2 วงจรเทียบเท่าของเตาหุงต้มเหนี่ยวนำกับภาชนะหุงต้ม	4
รูปที่ 2-3 อิมพีแดนซ์เทียบเท่าของขดลวดเหนี่ยวนำกับภาชนะหุงต้ม	4
รูปที่ 2-4 ความสัมพันธ์ระหว่างความหนาแน่นกระแสกับระยะทาง จากผิวดของภาชนะหุงต้ม	7
รูปที่ 2-5 เส้นกราฟแสดงการหาค่าคุณสมบัติทางไฟฟ้า ของภาชนะหุงต้มจากตารางที่ 2-1	8
รูปที่ 3-1 การพล็อตค่า $+jX_L$ และ $-jX_C$ กับความถี่สำหรับวงจรอนุกรม RLC รีโซแนนซ์	10
รูปที่ 3-2 กราฟอิมพีแดนซ์กระแสและมุมเฟสกับความถี่และกระแส ที่ไหลเมื่อรีโซแนนซ์ ($X_L = X_C$)	11
รูปที่ 3-3 กราฟความสัมพันธ์ระหว่างค่า V_R , V_C และ V_L กับความถี่	12
รูปที่ 3-4 รูปเฟสเซอร์โคแอสแกรมของแรงดันในวงจร RLC รีโซแนนซ์	13
รูปที่ 3-5 เฟสเซอร์โคแอสแกรมรูปคลื่นของกระแส ที่จุกรีโซแนนซ์ของวงจร RLC ขนาน	14
รูปที่ 3-6 วงจรปกติของการต่อแบบ RLC ขนาน	15
รูปที่ 3-7 กราฟแสดงค่าอิมพีแดนซ์และกระแสต่อค่าความถี่	16
รูปที่ 4-1 สัญลักษณ์ไอจีบีทีในลักษณะของไบโพลาร์ทรานซิสเตอร์ และมอสเฟตตามลำดับ	18
รูปที่ 4-2 โครงสร้างของไอจีบีทีชนิดอื่นบนเนลเป็นภาพตัดขวาง	19
รูปที่ 4-3 กราฟลักษณะคุณสมบัติการถ่ายโอนของไอจีบีที	19
รูปที่ 4-4 กราฟลักษณะคุณสมบัติระหว่างกระแสและแรงดันของไอจีบีที	20
รูปที่ 4-5 ทิศทางการไหลของอิเล็กตรอนและโฮลในขณะนำกระแส	21
รูปที่ 4-6 โครงสร้างที่ปรับปรุงเพื่อป้องกันการแลตช์อินของ ไอจีบีที	23
รูปที่ 4-7 โครงสร้างที่มีไบโพลาร์ทรานซิสเตอร์และมอสเฟตแฝงอยู่ภายใน	24
รูปที่ 4-8 ลักษณะของกระแสและแรงดันขณะนำกระแส	25
รูปที่ 4-9 ลักษณะของกระแสและแรงดันขณะหยุดนำกระแส	26
รูปที่ 5-1 ชิพ 40 ขาแบบ OTP	31
รูปที่ 5-2 ชิพ 40 ขาแบบ EPROM	31
รูปที่ 5-3 ชิพ 40 ขาแบบ EEPROM / Flash	31
รูปที่ 5-4 ชิพ 18ขาตระกูล PIC16CXXX, PIC16FXXX	32
รูปที่ 5-5 ชิพ 20 ขา ตระกูล PIC16CXXX, PIC18FXXX	33

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

	หน้า
รูปที่ 5-6 ชิป 8 ขาตระกูลPIC12CXXX, PIC12FXXX	34
รูปที่ 5-7 ชิป 28 ขาตระกูลPIC16CXX, PIC16FXXX, PIC18CXXX, PIC18FXXX	35
รูปที่ 5-8 ชิป 40 ขาตระกูลPIC16CXXX, PIC16FXXX, PIC18CXXX, PIC18FXXX	36
รูปที่ 5-9 ไคอะแกรมการทำงานของโมดูล CCP เพื่อสร้างสัญญาณ PWM	57
รูปที่ 5-10 แสดง Set Lock Diagram ของ A/D	65
รูปที่ 6-1 วงจรกำลังของ เต่าหุงคัม	79
รูปที่ 6-2 รูปวงจรบริดจ์เร็คติไฟเออร์	79
รูปที่ 6-3 ก) วงจรอินเวอร์เตอร์ชนิดฟูลบริดจ์ ข) วงจรอินเวอร์เตอร์ชนิดฮาร์ฟบริดจ์	80
รูปที่ 6-4 แสดงวงจร โทลครีโศแนนซ์ของเต่าหุงคัม	81
รูปที่ 6-5 แสดงวงจร EMI FINDER	84
รูปที่ 6-6 ไคอะแกรมการทำงานของโมดูล CCP ที่สร้างสัญญาณ PWM	85
รูปที่ 6-7 วงจรปรับแต่งเวลา (Dead Time)	86
รูปที่ 6-8 วงจรขยายสัญญาณขับไอจีบีที	87
รูปที่ 6-9 แสดงวงจรป้องกัน	87
รูปที่ 7-1 ลักษณะรูปคลื่นแรงดัน (V_s) และกระแส (I_s) ของแหล่งจ่ายจากการไฟฟ้า	88
รูปที่ 7-2 สเปกตัมของแรงดัน V_s และกระแส I_s ของแหล่งจ่ายการไฟฟ้า	88
รูปที่ 7-3 แรงดันด้านแหล่งจ่ายการไฟฟ้า (V) ขณะปรับความถี่	89
รูปที่ 7-4 กระแสด้านแหล่งจ่ายการไฟฟ้า (I) ขณะปรับความถี่	90
รูปที่ 7-5 ค่าความผิดเพี้ยนของแรงดันด้านแหล่งจ่ายการไฟฟ้า (% THD V) ขณะปรับความถี่	91
รูปที่ 7-6 ค่าความผิดเพี้ยนของกระแสด้านแหล่งจ่ายการไฟฟ้า (% THD i) ขณะปรับความถี่	92
รูปที่ 7-7 เพาเวอร์แฟกเตอร์ด้านแหล่งจ่ายการไฟฟ้า (PF) ขณะปรับความถี่	93
รูปที่ 7-8 กำลังไฟฟ้าอินพุตด้านแหล่งจ่ายการไฟฟ้า (P_{in}) ขณะปรับความถี่	94
รูปที่ 7-9 กำลังไฟฟารีแอกแตนด์ด้านแหล่งจ่ายการไฟฟ้า (Q) ขณะปรับความถี่	95
รูปที่ 7-10 กำลังไฟฟ้าอินพุตปรากฏด้านแหล่งจ่ายการไฟฟ้า (S) ขณะปรับความถี่	96
รูปที่ 7-11 อุณหภูมิของน้ำปริมาณ 1 ลิตร ขณะปรับความถี่	97
รูปที่ 7-12 อุณหภูมิของน้ำปริมาณ 1 ลิตร เมื่อปรับความถี่	98
รูปที่ 7-13 รูปคลื่นกระแสและแรงดันที่ความถี่ 29 kHz	100
รูปที่ 7-14 รูปคลื่นกระแสและแรงดันที่ความถี่ 30 kHz	100
รูปที่ 7-15 รูปคลื่นกระแสและแรงดันที่ความถี่ 31 kHz	101

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มา

เตาไฟฟ้าในท้องตลาดส่วนใหญ่ในปัจจุบัน มีการทำให้เกิดความร้อนโดยอาศัยกระแสไฟฟ้าผ่านขดลวดความร้อน และนำความร้อนที่ได้จากขดลวดส่งผ่านไปยังภาชนะอีกทีหนึ่ง ซึ่งวิธีดังกล่าวนี้ความร้อนไม่ได้ถูกถ่ายเทไปยังภาชนะโดยตรง ซึ่งจะมีการสูญเสียกำลังไฟฟ้าไปข้างนอกด้วย ดังนั้นเพื่อลดการสูญเสียดังกล่าวจึงใช้หลักการให้ความร้อนแบบเหนี่ยวนำ ซึ่งหลักการดังกล่าวเป็นการส่งพลังงานออกไปยังภาชนะในรูปของสนามแม่เหล็ก ซึ่งสนามแม่เหล็กดังกล่าวจะไปเกี่ยวข้องกับที่ภาชนะโดยตรง และภาชนะมีลักษณะเป็นวงจรถัด ทำให้เกิดกระแสไหลวนซึ่งจะทำให้เกิดเป็นความร้อนในที่สุด และความร้อนก็จะถ่ายเทไปยังอาหารที่เรานำมาปรุง ซึ่งหลักการการให้ความร้อนเหนี่ยวนำนี้มีประโยชน์มากมาย คือ ประหยัดพลังงาน มีความปลอดภัยสูง ไม่มีมลภาวะเป็นพิษ และให้ความร้อนที่เร็วกว่า

สำหรับแหล่งจ่ายพลังงานที่ใช้ในการเหนี่ยวนำความร้อน เราจะใช้ความถี่สูง (20 KHz) โดยการนำความถี่สูงนี้สามารถลดขนาดของอุปกรณ์ลงได้ อีกทั้งยังสามารถให้กำลังไฟฟ้าสูงๆอีกด้วย ปัจจุบันการวิจัยทางด้านเพาเวอร์อิเล็กทรอนิกส์ได้มีการพัฒนาไปอย่างมาก ซึ่งรวมถึงการพัฒนาอินเวอร์เตอร์ที่มีความถี่สูง โดยการใช้อุปกรณ์ประเภท MOSFET และ ไอจีบีที ซึ่งมีความเร็วในการสวิตช์สูงและเนื่องจากโหลดแบบขดลวดความร้อนเหนี่ยวนำจะมีค่าเพาเวอร์แฟคเตอร์ต่ำ ดังนั้นจึงใช้อินเวอร์เตอร์แบบรีโซแนนซ์อนุกรม ซึ่งใช้หลักการรีโซแนนซ์เป็นแหล่งจ่ายพลังงานส่งต่อไปยังโหลดรีโซแนนซ์อนุกรม ทำให้ได้ประสิทธิภาพที่สูงมากขึ้น

เตาหุงต้มเหนี่ยวนำนี้เป็นการให้ให้ความร้อนแก่ภาชนะ โดยการอาศัยกระแสไหลวนที่กั้นภาชนะและกระแสไหลวนดังกล่าวทำให้เกิดความร้อนขึ้นมาโดยจะมีพลังงานที่เข้ามาเกี่ยวข้องกับเตาหุงต้มเหนี่ยวนำคือ พลังงานสนามไฟฟ้า พลังงานสนามแม่เหล็ก และพลังงานความร้อน

ในเบื้องต้นนั้น เตาหุงต้มแบบเหนี่ยวนำจะรับพลังงานไฟฟ้า 220 โวลต์ 50 Hz หลังจากนั้นจะผ่านชุดเรกติไฟเออร์ (rectifier) ผ่านวงจรถัด (filter) เพื่อกรองสัญญาณแรงดันไฟฟ้ากระแสตรงให้เรียบ เพื่อควบคุมกระแสให้คงที่และสัญญาณจะผ่านชุด ฮาร์ฟบริดจ์ อินเวอร์เตอร์ ซึ่งที่วงจรถัดบริดจ์ อินเวอร์เตอร์จะใช้ไอจีบีทีเป็นอุปกรณ์สวิตช์ ซึ่งควบคุมสัญญาณการขับสวิตช์ด้วยไมโครคอนโทรลเลอร์ PIC 18F458 หลังจากนั้นสัญญาณจะถูกส่งไปยังโหลดรีโซแนนซ์อนุกรม ซึ่งมีขดลวดเหนี่ยวนำต่ออนุกรมกับตัวเก็บประจุหลังจากนั้นปรับความถี่ที่ 22 KHz ซึ่งเป็นความถี่รีโซแนนซ์ที่โครงการนี้ได้ออกแบบไว้ ณ. ความถี่รีโซแนนซ์จะเป็นความถี่ที่ทำให้เกิดกระแสเหนี่ยวนำสูงสุด ทำให้มีสนามแม่เหล็กเกี่ยวข้องกับที่กั้นภาชนะมาก และทำให้เกิดกระแสไหลวนที่กั้นภาชนะมากทำให้เกิดความร้อนในที่สุด

1.2 วัตถุประสงค์

- 1.2.1 เพื่อศึกษาหลักการให้ความร้อนโดยการเหนี่ยวนำความถี่สูง
- 1.2.2 เพื่อศึกษาและเปรียบเทียบประสิทธิภาพกับเตาหุงต้มทั่วไป
- 1.2.3 วิเคราะห์การเปลี่ยนแปลงของตัวแปรต่างๆในวงจรไฟฟ้ารีโซแนนซ์อนุกรม เช่น อิมพีแดนซ์ ควอลิตี แฟคเตอร์ กำลังไฟฟ้าและความถี่ในย่านต่างๆ
- 1.2.4 เพื่อเพิ่มตัวเลือกในการหาพลังงานทดแทนมาใช้ให้เกิดประโยชน์สูงสุด

1.3 ขอบเขตของงาน

ศึกษาหลักการให้ความร้อนเหนี่ยวนำ

1.4 วิธีการดำเนินงาน

- 1.4.1 ทำการวางแผนการทำงาน
- 1.4.2 ศึกษาข้อมูลที่เกี่ยวข้องกับเตาหุงต้มเหนี่ยวนำความถี่สูง
- 1.4.3 ศึกษาข้อมูลเกี่ยวกับวงจรอินเวอร์เตอร์รีโซแนนซ์แบบต่างๆที่สามารถใช้กับเตาหุงต้มเหนี่ยวนำ
- 1.4.4 ออกแบบวงจร
- 1.4.5 ทดสอบพร้อมทั้งเก็บผลการทดลอง
- 1.4.6 เขียนและพิมพ์ปริิญาานิพนธ์



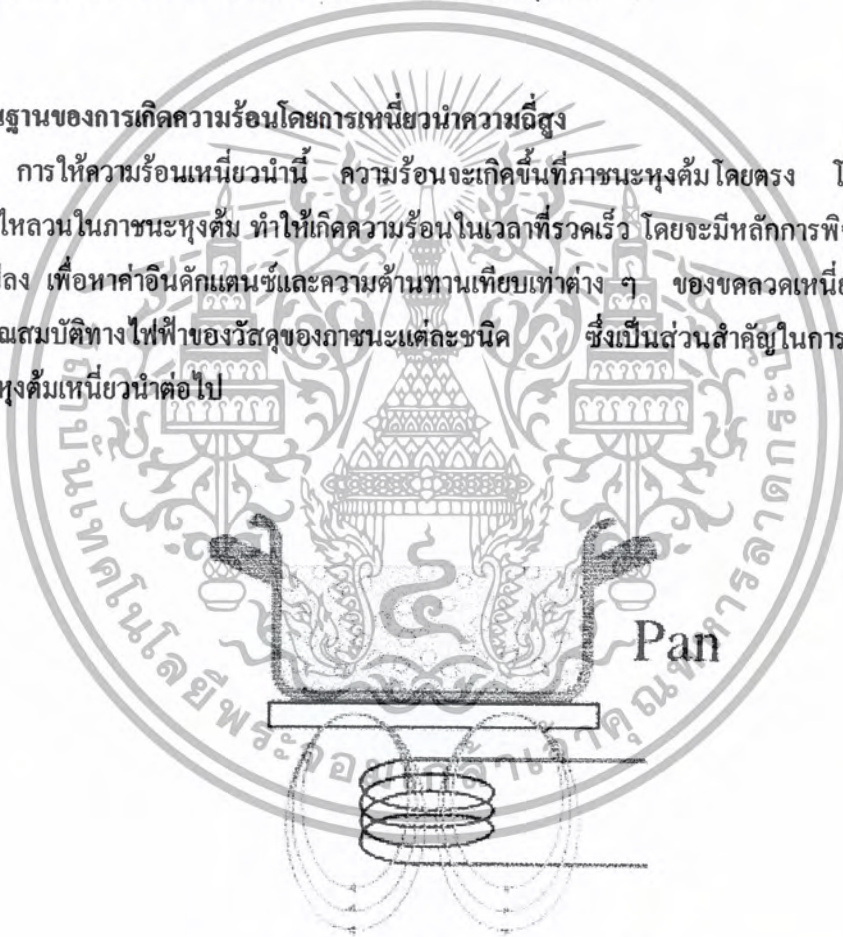
บทที่ 2 ทฤษฎีหลักการให้ความร้อนเหนี่ยวนำ

2.1 บทนำ

ในบทนี้จะกล่าวถึง หลักการให้ความร้อนโดยอาศัยการเหนี่ยวนำความถี่สูง การพิจารณาวงจรเทียบเท่าของเตาหุงต้มเหนี่ยวนำในลักษณะหม้อแปลง เพื่อให้สามารถวิเคราะห์ชุดโหลดของภาชนะหุงต้มได้ง่ายขึ้น สำหรับโหลดภาชนะหุงต้มที่ใช้วัสดุต่างชนิดกันจะมีค่าพารามิเตอร์ทางไฟฟ้าที่แตกต่างกัน ดังนั้นวัสดุภาชนะหุงต้มแต่ละชนิดจึงเป็นองค์ประกอบสำคัญในการพิจารณาในการออกแบบลักษณะการพันขลวดเหนี่ยวนำเพื่อให้เหมาะสมกับโหลดภาชนะหุงต้มนั้นๆ

2.2 พื้นฐานของการเกิดความร้อนโดยการเหนี่ยวนำความถี่สูง

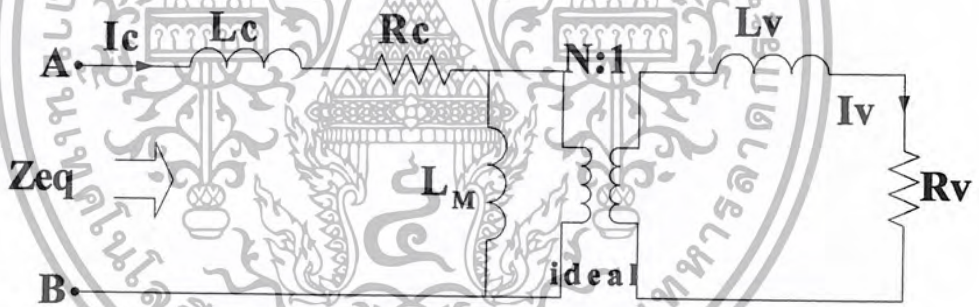
การให้ความร้อนเหนี่ยวนำนี้ ความร้อนจะเกิดขึ้นที่ภาชนะหุงต้มโดยตรง โดยอาศัยหลักการกระแสไหลวนในภาชนะหุงต้ม ทำให้เกิดความร้อนในเวลาทีรวดเร็ว โดยจะมีหลักการพิจารณาเหมือนกับหม้อแปลง เพื่อหาค่าอินดักแตนซ์และความต้านทานเทียบเท่าต่าง ๆ ของขลวดเหนี่ยวนำ ซึ่งจะขึ้นอยู่กับคุณสมบัติทางไฟฟ้าของวัสดุของภาชนะแต่ละชนิด ซึ่งเป็นส่วนสำคัญในการพิจารณาเลือกใช้ภาชนะหุงต้มเหนี่ยวนำต่อไป



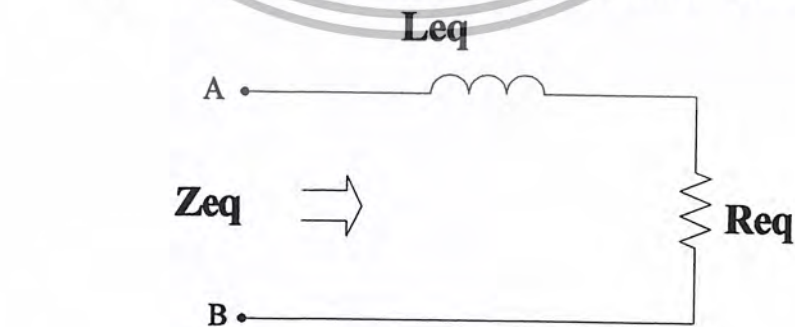
Magnetic Friction

รูปที่ 2-1 หลักการให้ความร้อนเหนี่ยวนำ

หลักการให้ความร้อนแบบเหนี่ยวนำ จากรูปที่ 2-1 เมื่อป้อนกระแสสลับความถี่สูงให้กับขดลวดเหนี่ยวนำที่อยู่ใต้ภาชนะหุงต้ม ขดลวดเหนี่ยวนำจะสร้างสนามแม่เหล็กที่เคลื่อนที่ (Magnetic Flux) กับภาชนะหุงต้ม ถ้าภาชนะหุงต้มเป็นโลหะที่มีคุณสมบัติเป็นสาร Ferro-Magnetic สนามแม่เหล็กที่เกี่ยวข้องกับภาชนะจะทำให้เกิดกระแสไหลวน (Eddy Current) ในภาชนะหุงต้มเป็นเส้นทางปิด ทำให้เกิดการสูญเสียเนื่องจากกระแสไหลวน เป็นผลให้เกิดความร้อนขึ้นที่ก้นของภาชนะหุงต้ม และจะถ่ายเทความร้อนไปยังน้ำที่อยู่ภายในภาชนะ ทำให้น้ำเดือดได้ในที่สุด ความร้อนที่เกิดขึ้นนี้จะไม่มีการสัมผัสกันทางไฟฟ้าระหว่างขดลวดเหนี่ยวนำกับภาชนะหุงต้ม ทำให้ปลอดภัยต่อการใช้งาน เตาหุงต้มเหนี่ยวนำนี้จะมีประสิทธิภาพในการหุงต้มสูง เนื่องจากความร้อนจะเกิดขึ้นที่ตัวภาชนะโดยตรง จะเห็นได้ว่าโครงสร้างของขดลวดเหนี่ยวนำและ โหลดภาชนะหุงต้มจะมีเส้นแรงแม่เหล็กเคลื่อนที่ผ่านซึ่งกันและกัน ซึ่งมีลักษณะคล้ายกับหม้อแปลง โดยสามารถนำมาเขียนเป็นวงจรเทียบเท่าของหม้อแปลงดังรูปที่ 2-2 โดยภาชนะหุงต้มทำหน้าที่เป็นขดลวดทุติยภูมิมีจำนวนขดลวดเพียง 1 รอบ โดยมี Leakage Inductance ของภาชนะหุงต้ม L_v แลความต้านทานของภาชนะ R_v ส่วนขดลวดทางด้านปฐมภูมิสามารถแทนด้วยขดลวดเหนี่ยวนำ R_c และ Magnetizing Inductance L_M ซึ่งมีค่าน้อย และเมื่อรวมกับ R_v และ L_v ทางด้านทุติยภูมิที่ย้ายมารวมกับด้านปฐมภูมิแล้วจะได้อิมพีแดนซ์รวม Z_{eq} ประกอบด้วย R_{eq} และ L_{eq} ดังรูปที่ 2-3 ดังนั้นขดลวดเหนี่ยวนำและ โหลดภาชนะหุงต้มนี้สามารถเขียนแทนด้วยวงจรเทียบเท่า ซึ่งประกอบด้วย อินดักแตนซ์เทียบเท่า L_{eq} และความต้านทานเทียบเท่า R_{eq}



รูปที่ 2-2 วงจรเทียบเท่าของเตาหุงต้มเหนี่ยวนำกับภาชนะหุงต้ม



รูปที่ 2-3 อิมพีแดนซ์เทียบเท่าของขดลวดเหนี่ยวนำกับภาชนะหุงต้ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขดลวดเหนี่ยวนำ และโหลดภาชนะหุงต้มสามารถเขียนแทนด้วยวงจรถอมนุกรมเทียบเท่า ซึ่งประกอบด้วย อินดักแตนซ์เทียบเท่า และความต้านทานเทียบเท่าสามารถหาค่าได้ดังสมการ

$$R_{eq} = R_c + N^2 R_v \quad (2.1)$$

เมื่อ $R_{vp} = N^2 R_c$ จะได้

$$R_{eq} = R_c + R_{vp} \quad (2.2)$$

$$L_{eq} = L_c + N^2 L_v \quad (2.3)$$

โดยที่

N คือ จำนวนรอบของขดลวดเหนี่ยวนำ

R_c คือ ความต้านทานของขดลวดปฐมภูมิหรือขดลวดเหนี่ยวนำ

R_v คือ ความต้านทานของขดลวดทุติยภูมิหรือโหลดภาชนะหุงต้ม

L_c คือ Leakage Inductance ทางด้านปฐมภูมิ

L_v คือ Leakage Inductance ทางด้านทุติยภูมิ

R_{vp} คือ ความต้านทานทางด้านปฐมภูมิของโหลดภาชนะหุงต้ม

กำลังไฟฟ้าสูญเสียในขดลวดเหนี่ยวนำมีค่าเท่ากับ

$$P_c = I_c^2 R_c \quad (2.4)$$

กำลังไฟฟ้าสูญเสียในโหลดภาชนะหุงต้มมีค่าเท่ากับ

$$P_v = I_v^2 R_v \quad (2.5)$$

เมื่อพิจารณาทางด้านปฐมภูมิจะได้

$$R_v = I_c^2 R_{vp} \quad (2.6)$$

เนื่องจาก

$$I_v = N I_c \quad (2.7)$$

โดยที่

I_c คือ กระแสในโหลดภาชนะหุงต้ม

I_v คือ กระแสในขดลวดเหนี่ยวนำ

ดังนั้นประสิทธิภาพของโหลดภาระหุงต้มจะได้ดังนี้

$$\eta = \frac{P_v}{P_v + P_c} \quad (2.8)$$

$$= \frac{I_v^2 R_v}{I_v^2 R_v + I_c^2 R_c} \quad (2.9)$$

$$= \frac{I_v^2 R_v}{\left[\frac{I_v^2}{N^2} \right] [R_v N^2] + I_c^2 R_c} \quad (2.10)$$

$$= \frac{I_c^2 R_{vp}}{\left[I_c^2 \right] [R_{vp}] + I_c^2 R_c} \quad (2.11)$$

$$= \frac{R_{vp}}{R_{vp} + R_c} \quad (2.12)$$

$$= \frac{1}{1 + \frac{R_c}{R_{vp}}} \quad (2.13)$$

ความร้อนส่วนมากจะเกิดขึ้นโดยกระแสไหลวนใน โหลดภาระหุงต้มที่บริเวณชั้นผิวนอกของ ก้นภาระหุงต้มที่มีความลึกเท่ากับ ซึ่งมีความสัมพันธ์กับพารามิเตอร์ของภาระหุงต้มดังสมการ

$$\delta = \sqrt{\frac{1}{4\pi^2 \times 10^{-7}}} \cdot \sqrt{\frac{\rho}{\mu_r f}} \quad (2.14)$$

- โดยที่ ρ คือ สภาพความต้านทานทางไฟฟ้าของภาระ
 f คือ ความถี่ไฟฟ้ากระแสสลับที่ป้อน
 μ_r คือ ความซึมซาบแม่เหล็กสัมพัทธ์

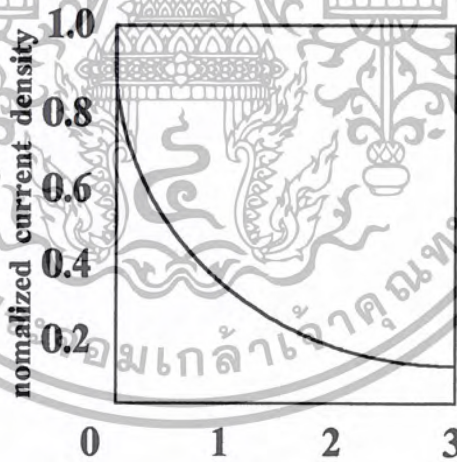
ดังนั้นความต้านทานที่ผิวของโพลีเอทิลีนหุ้มในย่านความถี่สูง มีค่าดังนี้

$$R_v = \frac{\rho}{\delta} = \sqrt{4\pi^2 \times 10^{-7}} \cdot \sqrt{\rho\mu_r \cdot f} \quad (2.15)$$

แทนค่าสมการ ใน จะได้กำลังไฟฟ้าเอาต์พุตของโพลีเอทิลีนหุ้ม ดังนี้

$$P_v = I_v^2 R_v = \sqrt{4\pi^2 \times 10^{-7}} \cdot (NI_c)^2 \cdot \sqrt{\rho\mu_r \cdot f} \quad (2.16)$$

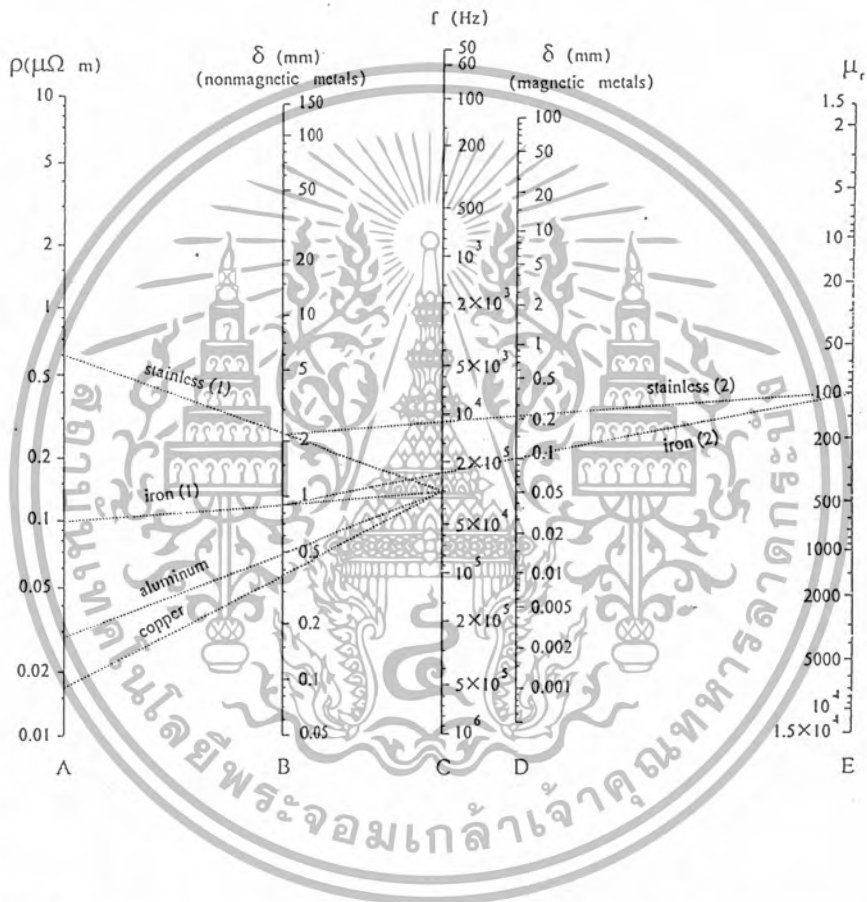
จากสมการ (2.15) จะเห็นได้ว่าความต้านทานของภาชนะหุ้ม จะแปรผันโดยตรงกับสภาพความต้านทานทางไฟฟ้า ของภาชนะหุ้ม ค่าความซึมซาบแม่เหล็ก และความถี่ไฟฟ้ากระแสสลับ ที่ป้อนจากอินเวอร์เตอร์ จากสมการ (2.16) จะได้ว่า การเพิ่มของเพาเวอร์เอาต์พุต ของภาชนะหุ้ม สามารถทำได้โดยการเพิ่มความถี่ ของกระแสในขดลวดเหนี่ยวนำและแอมแปร์-เทอรัน ของขดลวดทางด้านปฐมภูมิ รูปที่ 2-4 แสดงความสัมพันธ์ระหว่างความหนาแน่นกระแสกับระยะความถี่จากผิวด้านนอกของกันภาชนะและจะลดลงเมื่อระยะความถี่จากผิวมากขึ้น



รูปที่ 2-4 ความสัมพันธ์ระหว่างความหนาแน่นกระแสกับระยะทางจากผิวด้านนอกของภาชนะหุ้ม

ตารางที่ 2-1 คุณสมบัติทางไฟฟ้าของวัสดุต่างๆ ที่ใช้ในภาชนะหุงต้มที่ความถี่ 30 kHz

material	Iron	stainless	Aluminum	Copper
Resistivity:	9.8×10^{-8}	60×10^{-8}	2.8×10^{-8}	1.7×10^{-8}
Relative Permeability:	100	100	1	1
Skin depth(mm):	0.909	0.225	0.486	0.378
Surface resistivity:	11×10^{-4}	27×10^{-4}	0.575×10^{-4}	0.448×10^{-4}



กรณีใช้วัสดุสาร Non-Ferro-Magnetic

ลากเส้นจากตำแหน่งบนสเกล ไปยังตำแหน่งบนสเกล $f : C$ และอ่านค่าที่ได้บนสเกล

กรณีใช้วัสดุสาร Ferro-Magnetic

ลากเส้นระหว่างตำแหน่งบนสเกล และสเกล $f : C$ จะได้จุดตัดกับตำแหน่งบนสเกล จากนั้นลากเส้นระหว่างจุดตัดนี้กับตำแหน่งบนสเกล จะได้จุดตัดใหม่ที่ตำแหน่งบนสเกลอ่านค่า ที่ได้ตำแหน่งจุดตัดนี้

รูปที่ 2-5 เส้นกราฟแสดงการหาค่าคุณสมบัติทางไฟฟ้าของภาชนะหุงต้มจากตารางที่ 2-1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 2-1 แสดงคุณสมบัติทางไฟฟ้าของภาชนะหุงต้มที่วัสดุต่าง ๆ กัน ที่ความถี่ 20 kHz จะเห็นได้ว่า วัสดุที่ทำด้วยสาร non ferro-magnetic เช่น อลูมิเนียมและทองแดง มีค่าสภาพความต้านทาน และความซึมซาบแม่เหล็ก ต่ำมากเมื่อเทียบกับสแตนเลสและเหล็กซึ่งทำด้วยสาร ferro-magnetic ดังนั้นการเพิ่มกำลังเอาต์พุทเพาเวอร์สำหรับโหลดที่ภาชนะทำด้วยสาร nonferro-magnetic สามารถกำหนดได้ตามสมการที่ (2.16) คือ เพิ่มความถี่ในขดลวดเหนี่ยวนำและเพิ่มจำนวนรอบของขดลวดเหนี่ยวนำ



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3 หลักการรีโซแนนซ์

3.1. บทนำ

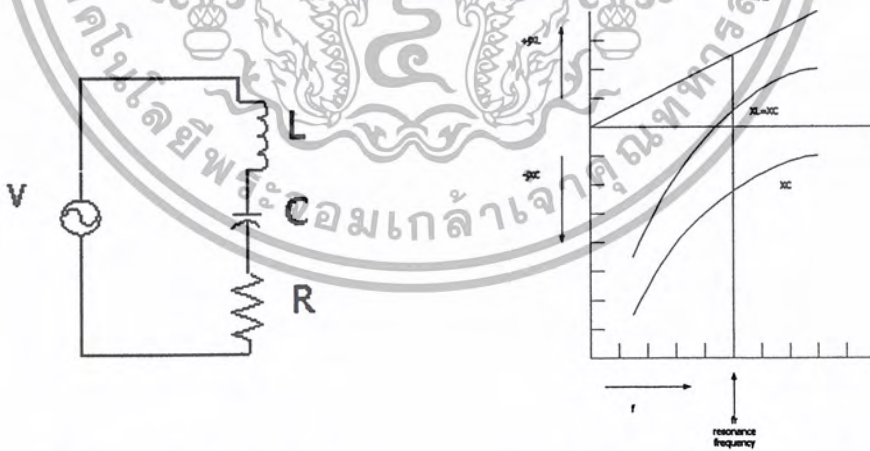
เมื่อเราปรับความถี่ที่ป้อนให้กับวงจรอนุกรม RLC จะพบว่าที่ความถี่ค่าหนึ่ง ค่า X_L จะเท่ากับ X_C เป็นผลให้ค่าอิมพีแดนซ์ (Impedance) ของวงจร RLC มีค่าต่ำที่สุด นั่นคือ $Z=R$ ในสภาวะเช่นนี้ เราเรียกว่าอยู่ในสภาวะรีโซแนนซ์ค่าความถี่ที่เกิดขึ้นนี้เรียกว่าความถี่รีโซแนนซ์ (Resonance Frequency) เนื่องจากค่าอิมพีแดนซ์มีค่าต่ำสุด กระแสที่ไหลในวงจรจึงมีค่ามากที่สุดและเป็นผลให้แรงดันที่ตกคร่อม L และ C มีค่าเพิ่มขึ้น

วงจรขนาน LC ก็สามารถที่ทำให้อยู่ในสภาวะรีโซแนนซ์ได้เช่นกันและ X_L จะเท่ากับ X_C ในขณะเช่นนี้ค่าอิมพีแดนซ์ของวงจรจะมีความมากที่สุด และส่งผลให้กระแสมีค่าน้อยที่สุดแต่กระแสที่ไหลวนในลูป (LOOP) ของ L และ C จะมีค่ามากขึ้น วงจรสามารถปรับให้รีโซแนนซ์ได้โดยการปรับค่า L และ C

3.2 รีโซแนนซ์อนุกรม

ผลของความถี่ที่มีต่ออิมพีแดนซ์ของวงจรอนุกรม RLC จะแสดงค่าดังนี้

$$Z = R + j(X_L - X_C) \tag{3.1}$$



ก) วงจรอนุกรม RLC

ข) X_L, X_C เมื่อพล็อตกับความถี่

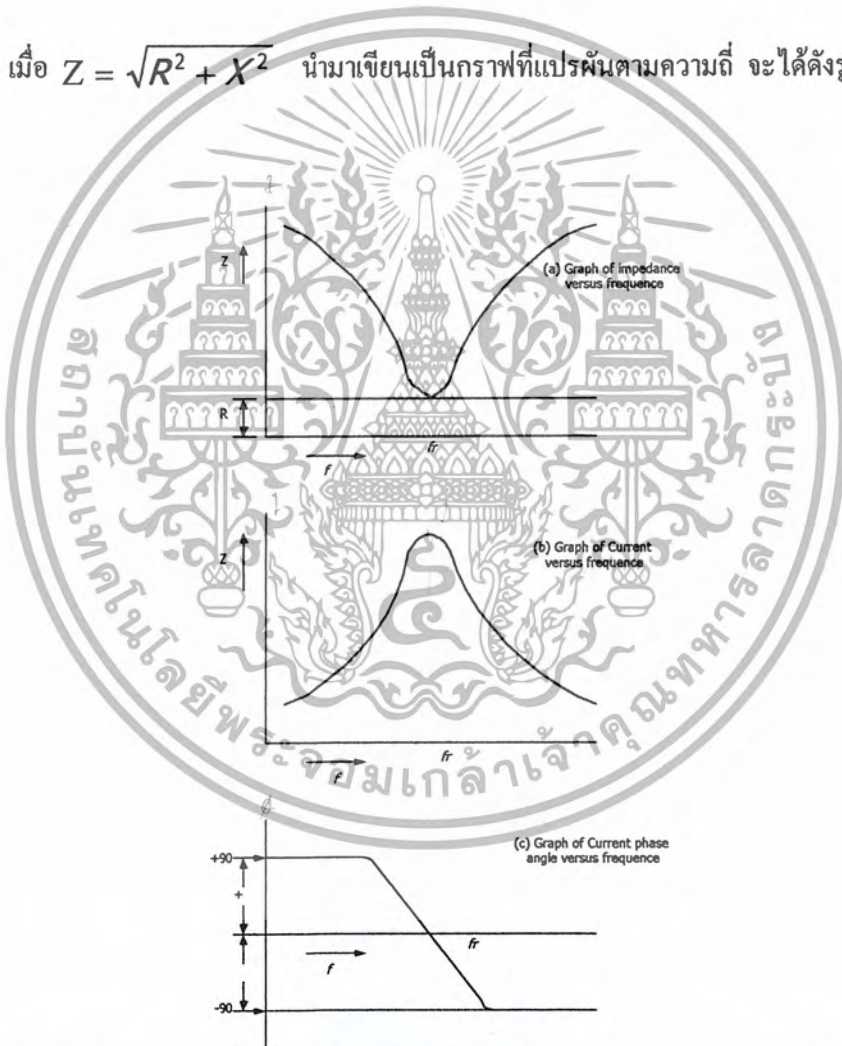
รูปที่ 3-1 การพล็อตค่า $+jX_L$ และ $-jX_C$ กับความถี่สำหรับวงจรอนุกรม RLC รีโซแนนซ์

เนื่องจากค่า $X_L = 2\pi fL$ และ $X_C = 1/(2\pi fc)$ ดังนั้นค่าของอิมพีแดนซ์จึงขึ้นอยู่กับค่าความถี่ของแหล่งจ่ายไฟ ดังรูปที่ 3-1 ข.) จะพบว่าเมื่อความถี่เพิ่มขึ้นจะทำให้ค่า X_L เพิ่มขึ้นอย่างเป็นเชิงเส้น (Linear) ในทางกลับกัน X_C จะลดลงเมื่อความถี่เพิ่มขึ้น แต่ค่า X_C จะไม่มีทางเท่ากับศูนย์ ผลรวมของรีแอกแตนซ์คือ $X_L - X_C$ ดังแสดงไว้เป็นเส้นประในกราฟของรูปที่ 3-1 ข.) ที่ความถี่ต่ำ ค่า X_L จะมีค่าน้อยกว่าค่า X_C ดังนั้นผลรวมของรีแอกแตนซ์จะเป็นค่าของ X_C ที่ความถี่สูง $X_L = X_C$ คือค่าที่ความถี่รีโซแนนซ์

$$Z = R + j(0) \tag{3.2}$$

$$Z = R$$

เมื่อ $Z = \sqrt{R^2 + X^2}$ นำมาเขียนเป็นกราฟที่แปรผันตามความถี่ จะได้ดังรูปที่ 3-2



รูปที่ 3-2 กราฟอิมพีแดนซ์กระแสและมุมเฟสกับความถี่และกระแสที่ไหลเมื่อรีโซแนนซ์ ($X_L = X_C$)

$$I = \frac{V}{R} \tag{3.3}$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3-2 จะพบว่ากราฟจะมีค่าต่ำเมื่อความถี่ในวงจร ไม่อยู่ที่จุดรีโซแนนซ์ที่ความถี่รีโซแนนซ์

$$X_C = \frac{1}{2\pi fC} \quad (3.4)$$

$$X_L = 2\pi fL \quad (3.5)$$

$$2\pi fL = \frac{1}{2\pi fC} \quad (3.6)$$

ดังนั้น เมื่อ L มีหน่วยเป็นเฮนรี่ (henrys) , C มีหน่วยเป็นฟาร์ด (farads)
F มีหน่วยเป็นเฮิรตซ์ (Hertz)

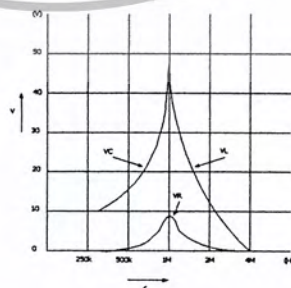
3.2.1 แรงดันที่จุดรีโซแนนซ์

แรงดันที่ตกคร่อมตัวต้านทานในวงจร RLC (รูปที่ 3-1 ก.) จะมีค่าเท่ากับ $I_X R$ และที่จุดรีโซแนนซ์ แรงดันที่ตกคร่อมตัวต้านทานจะมีค่าเท่ากับแรงดันที่แหล่งจ่าย ส่วนแรงดันที่ตกคร่อมตัวเก็บประจุและตัวเหนี่ยวนำจะมีค่าดังนี้

$$V_C = I X_C \quad (3.7)$$

$$V_L = I X_L \quad (3.8)$$

เมื่อนำค่า V_R , V_C และ V_L มาพล็อตกราฟกับค่าความถี่ จะพบว่ามัลักษณะคล้ายกับกราฟของกระแสกับความถี่ ดังแสดงในรูปที่ 3-3



รูปที่ 3-3 กราฟความสัมพันธ์ระหว่างค่า V_R V_C และ V_L กับความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.2 พลังงานที่ส่งผ่านระหว่าง L และ C



ก) รูปเฟสเซอร์ไดอะแกรมของวงจร RLC รีโซแนนซ์

ข) รูปคลื่นแรงดัน V_R , V_C และ V_L ที่จุกรีโซแนนซ์

รูปที่ 3-4 รูปเฟสเซอร์ไดอะแกรมของแรงดันในวงจร RLC รีโซแนนซ์

เมื่อเรานำรูปคลื่นของ V_R , V_C และ V_L ที่จุกรีโซแนนซ์มาพล็อต ดังรูปที่ 3-4 ก.) จะพบว่า V_L จะกลับเฟสกัน 180 องศา ส่วน V_R จะมีค่ามากที่สุดเมื่อ V_C และ V_L มีค่าเท่ากับศูนย์ จากที่กล่าวมานี้ แสดงให้เห็นว่าจะมีการส่งผ่านพลังงานระหว่าง L กับ C กลับไปมา เมื่ออยู่ในสภาวะรีโซแนนซ์ อย่างไรก็ตาม ที่จุกรีโซแนนซ์จะไม่มีกระแสรวมพลังงาน

3.2.3 Q แฟคเตอร์ของวงจรอนุกรม RLC

พิจารณาสมการ I , V_C , V_L ที่จุกรีโซแนนซ์

$$I = \frac{V}{R} \tag{3.9}$$

$$V_L = IX_L \tag{3.10}$$

$$V_L = \frac{V \times X_L}{R} \tag{3.11}$$

ดังนั้น

$$\frac{V_L}{V} = \frac{X_L}{R} \tag{3.12}$$

อัตราส่วน $\frac{V_L}{V}$ และ $\frac{V_C}{V}$ คือค่าควอลิตี้ (Quality) ของวงจรรีโซแนนซ์หรือค่า Q แฟคเตอร์นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

$$Q = \frac{X_L}{R} = \frac{X_C}{R} = \frac{1}{\omega CR} = \frac{1}{R} \sqrt{\frac{L}{C}} \quad (3.13)$$

3.3 รีโซแนนซ์ขนาน

วงจรรีโซแนนซ์ขนานแสดงไว้ดังรูป ค่าแอดมิตแตนซ์ (admittance) ของวงจรมีค่า

$$Y = \frac{1}{R} - j\frac{1}{X_L} + j\frac{1}{X_C} \quad (3.14)$$



Phase diagram for parallel

Waveform of I_C , I_R and I_L

รูปที่ 3-5 เฟสเซอร์ไดอะแกรมรูปคลื่นของกระแสที่จุดรีโซแนนซ์ของวงจร RLC ขนาน

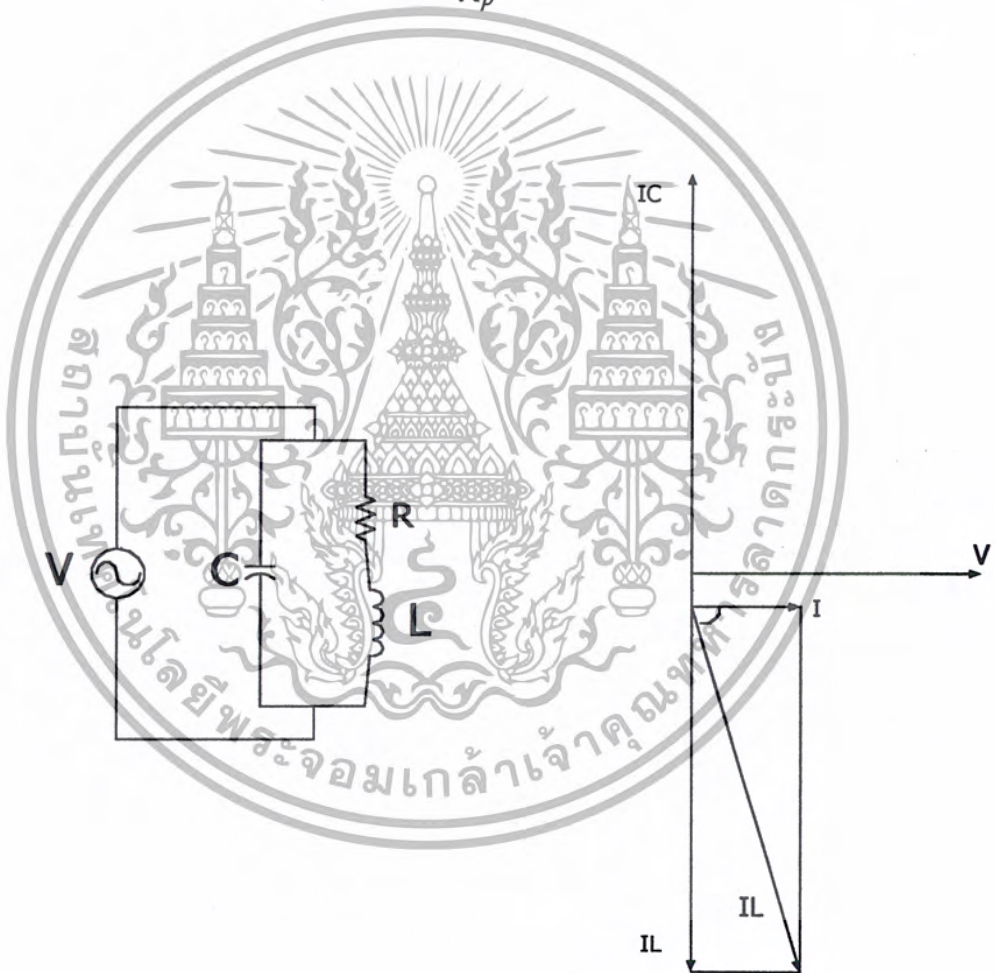
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ถ้า
ถ้าปรับความถี่ของแหล่งจ่ายไปจนกระทั่ง X_L และ X_C มีค่าเท่ากัน ค่าแอดมิตแดนซ์ จะมี

$$Y = \frac{1}{R_p} \quad (3.15)$$

และค่าอิมพีแดนซ์มีค่า $Z = R_p$ กระแสจากแหล่งจ่ายมีค่า

$$I = \frac{V}{R_p} \quad (3.16)$$



รูปที่ 3-6 วงจรปกติของการต่อแบบ RLC ขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กระแสที่ไหลผ่าน R จะมีมุมอินเฟสกับแรงดันของแหล่งจ่าย กระแสที่ไหลผ่าน L จะล่าหลังแรงดันไป 90 องศา ส่วนกระแสที่ไหลผ่าน C จะนำหน้าแรงดันอยู่ 90 องศา ดังแสดงไว้ในรูป คือมีการส่งผ่านพลังงานกลับไปมาระหว่างตัวเหนี่ยวนำและตัวเก็บประจุ

เนื่องจากในความเป็นจริงในตัวเหนี่ยวนำ จะมีค่าความต้านทานรวมอยู่ด้วย โดยทั่วไปที่ตัวเก็บประจุมีความต้านทานอยู่น้อยมาก จึงเขียนเป็นวงจรสมมูลได้ดังรูปที่ 3-6

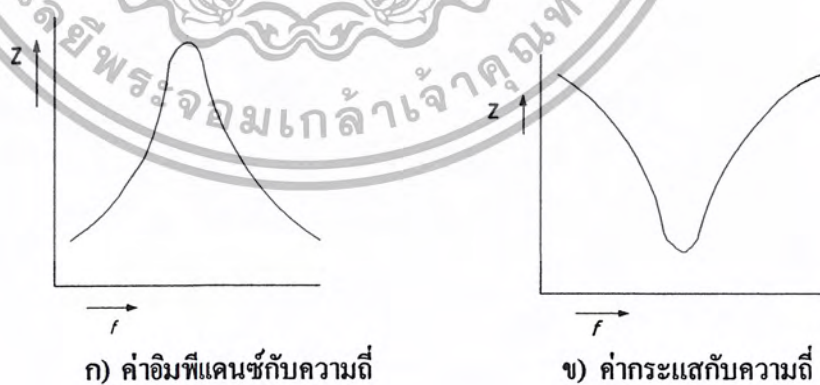
ค่าแอดมิตแตนซ์ของวงจรสมมูลมีค่า

$$Y = \frac{1}{R_L + jX_L} + \frac{1}{jX_C} \quad (3.17)$$

ที่จุดรีโซแนนซ์

$$X_C = \frac{R_L^2 + X_L^2}{X_L} \quad (3.18)$$

$$Z = \frac{L}{CR_L} \quad (3.19)$$



รูปที่ 3-7 กราฟแสดงค่าอิมพีแดนซ์และกระแสต่อค่าความถี่

จากกราฟรูปที่ 3-7 จะพบว่าวงจรขนาน LC จะมีค่าอิมพีแดนซ์สูงสุดที่ความถี่รีโซแนนซ์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.1 Q แฟกเตอร์ของวงจรขนาน RLC

จากรูปที่ 3-5 จะพบว่าผลรวมของกระแสที่ไหลในวงจร (I) จากแหล่งจ่ายที่ความถี่ รัโซแนนซ์นั้น จะอินเฟสกับแรงดันของแหล่งจ่าย (V) I จะมีค่าน้อยกว่า IC และ IL หลายเท่า ดังนั้นในวงจรขนานรีโซแนนซ์ LC จะมีค่าการขยายกระแส ค่า Q แฟกเตอร์คือค่าอัตราส่วนของ IL หรือ IC ต่อค่ากระแสจากแหล่งจ่าย (I)

ความถี่ที่จตุรีโซแนนซ์ จากค่า

$$X_C = \frac{R_L^2 + X_L^2}{X_L} \tag{3.20}$$

ดังนั้น



$$X_C \approx X_L \tag{3.21}$$

ซึ่งค่าความถี่ที่จตุรีโซแนนซ์ สำหรับวงจรขนาน LC เมื่อ $Q > 10$ คือ

$$f_r = \frac{1}{2\pi\sqrt{LC}} \tag{3.22}$$

ในกรณีที่ $Q < 10$ ความถี่ที่จตุรีโซแนนซ์จะมีค่า

$$f_r = \frac{1}{2\pi\sqrt{LC}} \sqrt{1 - \frac{CR_L^2}{L}} \tag{3.23}$$

จากรูปที่ 3-7 จะพบว่าวงจรรีโซแนนซ์แบบขนานทางปฏิบัติ นั้น จะมีค่าอิมพีแดนซ์สูงสุด และกระแสมีค่าสูงสุดที่ความถี่รีโซแนนซ์

จากที่กล่าวมาจะเห็นว่า การทำงานที่จตุรีโซแนนซ์จะให้ค่ากำลังที่สูงที่สุด (Pmax) เนื่องจากจะทำให้ค่ากระแสที่ไหลในขดลวดเหนี่ยวนำได้สูงสุด ซึ่งจำเป็นอย่างยิ่งต่อการเหนี่ยวนำให้เกิดกระแสไหลวนในทีบริเวณกันหม้อ เพื่อทำให้เกิดความร้อนให้ได้มากที่สุด จนสามารถต้มน้ำโดยใช้เวลาในการต้มให้น้อยที่สุด ซึ่งสามารถทำได้โดยการทำให้ความถี่ของวงจรอินเวอร์เตอร์ และความถี่ของวงจรตรงส่วนโหลดมีค่าตรงกัน โดยที่ค่าความต้านทานและค่าความเหนี่ยวนำในวงจร จะมาจากผลของขดลวดสร้างสนามแม่เหล็ก และกันหม้อ ส่วนตัวเก็บประจุที่ใส่เพิ่มเข้าไปก็เพื่อปรับปรุค่าตัวประกอบกำลังของวงจร

เอกสารนี้เป็นให้ตีพิมพ์ที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

เทคโนโลยีไอจีบีที

(Insulate Gate Bipolar Trasistor : IGBT)

4.1 ความรู้พื้นฐานเกี่ยวกับไอจีบี

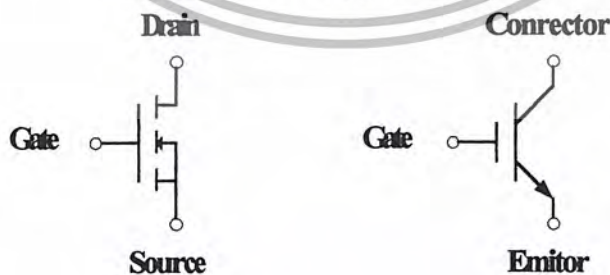
อุปกรณ์เพาเวอร์อิเล็กทรอนิกส์คอนโทรล ได้แก่ เอสซีอาร์ (SCR), ไตรแอก (TRIAC), ไบโพลาร์ทรานซิสเตอร์(BJT) และมอสเฟต(MOSFET) โดยเฉพาะไบโพลาร์ทรานซิสเตอร์และมอสเฟต จะเป็นจุดพัฒนาการของอุปกรณ์ทั้งสองชนิดก็มีคุณสมบัติที่แตกต่างกันออกไป กล่าวคือ

ไบโพลาร์ทรานซิสเตอร์ขณะอยู่ในสภาวะนำกระแสจะมีอัตราการสูญเสียกำลังงานต่ำ มีอัตราทนแรงดันและขยายกระแสได้สูง แต่ความเร็วในการสวิตซ์ทำงานยังต่ำอยู่ โดยเฉพาะช่วงหยุดนำกระแสจะมีช่วงเวลาที่ยาวกว่าซึ่งจะเป็นคุณสมบัติที่ตรงกันข้ามกับมอสเฟต ที่มีความเร็วในการสวิตซ์ทำงานนำกระแสและหยุดนำกระแสได้เร็วมากกว่ามาก แต่ก็มีอัตราการสูญเสียกำลังงานสูงมากเช่นกัน

จากเหตุผลที่กล่าวมาของไบโพลาร์ทรานซิสเตอร์และมอสเฟตจึงได้มีการพัฒนาอุปกรณ์ประเภทนี้จนสามารถได้อุปกรณ์อิเล็กทรอนิกส์เพาเวอร์คอนโทรลชนิดใหม่ขึ้นมา โดยคุณสมบัติต่าง ๆ จะรวมเอาข้อได้เปรียบของไบโพลาร์ทรานซิสเตอร์ และมอสเฟตเข้ามารวมไว้ในอุปกรณ์ชนิดใหม่นี้โดยมีการตั้งชื่อว่า ไอจีบีที (Insulate Gate Bioplar Transistor : IGBT)

4.2 สัญลักษณ์และโครงสร้างของไอจีบีที

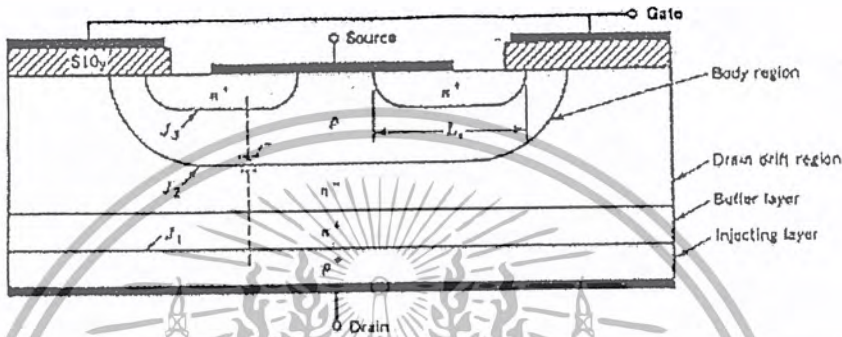
ไอจีบีทีก่อนข้างจะเป็นอุปกรณ์ชนิดใหม่ สัญลักษณ์แทนชนิดและตัวไอจีบีที มี 2 ชนิด คือ แบบเอ็นแชนเนล (N-Channel IGBT Type) และแบบพีแชนเนล (P-Channel IGBT Type) นอกจากนี้สามารถเขียนอยู่ในลักษณะของไบโพลาร์ทรานซิสเตอร์และมอสเฟตได้ ดังแสดงในรูปที่ 4-1 เป็นลักษณะไอจีบีทีชนิดเอ็นแชนเนลที่นิยมใช้กันมาก



รูปที่ 4-1 สัญลักษณ์ไอจีบีทีในลักษณะของไบโพลาร์ทรานซิสเตอร์และมอสเฟตตามลำดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

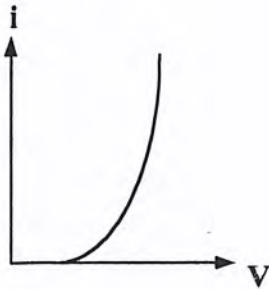
จากรูปที่ 4-1 จะเห็นว่ามีลักษณะคล้ายกับสัญลักษณ์ของมอสเฟตมาก เพียงแต่ว่าสัญลักษณ์ของไอจีบีทีนั้นจะมีลูกศรเพิ่มขึ้นมาก เพียงแต่ว่าสัญลักษณ์ของไอจีบีทีนั้นจะมีลูกศรเพิ่มมาตรงขาแครง ลักษณะของลูกศรจะชี้เข้าหาตัวหรือชี้เข้าหาชั้นของซิลิคอนภายในตัว ไอจีบีทีใช้สัญลักษณ์รูป 4-1 (ก) นี้ ส่วนสัญลักษณ์ในรูป 4-1(ข) จะเหมือนกับสัญลักษณ์ของไบโพลาร์ทรานซิสเตอร์ แต่ตรงขาแครง (หรือเบสของไบโพลาร์ทรานซิสเตอร์) จะเพิ่มขีดขึ้นมาอีกหนึ่งขีด เพื่อชี้ให้เห็นว่าขีดที่เพิ่มมานี้ไม่ได้ ต่อถึงกันโดยตรงกับขาที่ต่อออกมาภายนอก



รูปที่ 4-2 โครงสร้างของไอจีบีทีชนิดเอ็นแชนเนลเป็นภาพตัดขวาง

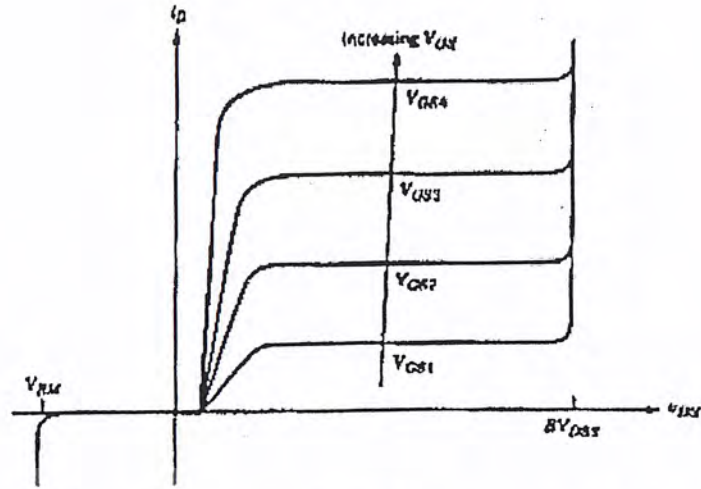
โครงสร้างของไอจีบีทีชนิดเอ็นแชนเนลแสดงเป็นภาพตัดขวางได้ดังรูปที่ โครงสร้างโดยรวมส่วนใหญ่มีลักษณะคล้ายกันกับโครงสร้างของมอสเฟตมาก จะแตกต่างกันตรงไอจีบีทีจะมีชั้นหรือชั้นอินเจกติ้ง (Injecting) ต่ออยู่ระหว่างขาแครง ซึ่งในมอสเฟตนั้น ไม่มีจากการที่ขาแครงถูกกันด้วยชั้นของซิลิคอนออกไซด์ (SiO_2) เป็นผลทำให้ความต้านทานอินพุตที่ขาแครงมีค่าสูงมากเหมือนกับเพาเวอร์มอสเฟตโดยทั่วไปจะมีค่าอยู่ช่วง 109 โอห์ม

จากผลดังกล่าวทำให้ลักษณะของกราฟแสดงคุณสมบัติของกระแสและแรงดันของไอจีบีทีมีลักษณะคล้ายกับกราฟของไบโพลาร์ทรานซิสเตอร์ แต่การควบคุมกระแสแครงจะอาศัยการควบคุมแรงดันระหว่างขาแครงกับขาซอร์สมากกว่า การควบคุมกระแสแครงจะอาศัยการควบคุมแรงดันระหว่างขาแครงกับขาซอร์สมากกว่า การควบคุมกระแสที่ขานี้เหมือนกับไบโพลาร์ทรานซิสเตอร์ ซึ่งกราฟแสดงคุณสมบัติของกระแสและแรงดันของไอจีบีที



รูปที่ 4-3 กราฟลักษณะคุณสมบัติการถ่ายโอนของไอจีบีที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-4 กราฟลักษณะคุณสมบัติระหว่างกระแสและแรงดันของไอจีบีที

รูปกราฟแสดงให้เห็นว่าส่วนใหญ่ของเส้นกราฟมีลักษณะเป็นเส้นตรงแต่จะเริ่มโค้งที่กระแสเดรนมีค่าต่ำ ๆ นั่นก็คือจุดที่แรงดันระหว่างขาเกตและขาซอร์สต่ำลงใกล้แรงดันจุดเริ่มเปลี่ยนสถานะการทำงาน (จุด Threshold voltage: $V_{GS(th)}$) โดยถ้าแรงดันระหว่างเกตและซอร์สมีต่ำกว่าแรงดันที่จุด $V_{GS(th)}$ แล้วไอจีบีทีจะอยู่ในสถานะหยุดนำกระแสหรือคัตออฟ ในกรณีของ ไอจีบีทีชนิดพีแชนแนล นั้น คุณสมบัติจะคล้ายกับเอ็นแชนแนล เช่น ชนิดของสารที่ได้อิเล็กตรอนจากเอ็นแชนแนลก็จะเปลี่ยนเป็นตรงกันข้าม, สัญลักษณ์ลูกศรก็จะกลับเอาหัวลูกศรกลับไปในทางตรงกันข้าม

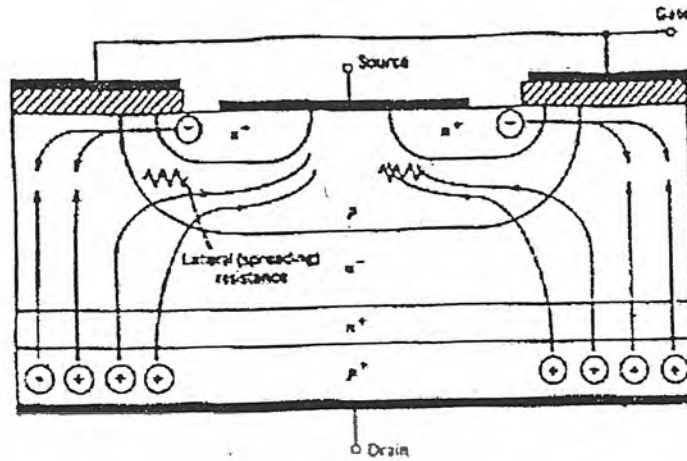
4.3 สถานะนำกระแส

เมื่อขาเดรนได้รับแรงดันไบแอสตรง คือ เป็นบวกเมื่อเทียบกับซอร์ส และแรงดันระหว่างเกตกับซอร์สมีค่าเกิน $V_{GS(th)}$ ประจุไฟฟ้าบวกที่เกิดจากแรงดันที่ขาเกตจะดึงเอาอิเล็กตรอนให้มารวมกันอยู่ในบริเวณภายใต้เกต ทำให้ชั้นบอดี้ (body layer) ตรงส่วนใต้เกตแปรสภาพเป็น n ทำให้เกิดการต่อกันของบริเวณ n^- (drift region) เข้ากับบริเวณซอร์ส n^+ (source region) ซึ่งลักษณะเช่นนี้เหมือนกับการทำงานของมอสเฟต

กระแสอิเล็กตรอนที่ไหลจากซอร์สผ่านบริเวณใต้เกตมายังบริเวณลอยเลื่อน n จะรวมกับโฮลที่เป็นพาหะข้างน้อยที่ถูกฉีดมาจากชั้นอินเจกต์ p^+ (ดูรูปโครงสร้างในรูปที่ 2) เพราะรอยต่อ J_1 ได้รับแรงดันไบแอสตรง ทำให้ IGBT อยู่ในสถานะนำกระแส เกิดการไหลของกระแสไฟฟ้าจากเดรนไปซอร์สได้ การรวมกันของโฮลและอิเล็กตรอนภายในบริเวณ n^- เรียกว่า การมอดูเลตสภาพนำ (conductivity modulation)

ผลของการมอดูเลตนี้จะทำให้ความต้านทานของบริเวณ n^- มีค่าต่ำลง เป็นการเพิ่มความสามารถในการขับผ่านกระแสได้สูงขึ้น ซึ่งจะมีลักษณะเหมือนกับทรานซิสเตอร์กำลัง ผลของความต้านทานที่ลดลง ทำให้แรงดันตกคร่อมที่สถานะนำกระแสลดต่ำลง การสูญเสียกำลังงานขณะนำกระแสจึงลดลงด้วย ทิศทางการไหลของอิเล็กตรอนและโฮลแสดงไว้ในรูปที่ 4-5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4-5 ทิศทางการไหลของอิเล็กตรอนและโฮลในขณะนำกระแส

4.4 สถานะหยุดนำกระแส

เมื่อแรงดันระหว่างเกตและซอร์สลดลงต่ำกว่าแรงดัน $V_{GS(th)}$ จะทำให้มีแรงดันไม่เพียงพอสำหรับการแปรสภาพชั้นบอดี้ p เป็น n ได้ ทำให้บริเวณ n^- ไม่ต่อกับบริเวณซอร์ส n^+ ไอจีบีทีในสถานะนี้รอยต่อ J_2 ที่ได้รับแรงดันไบแอสกลับจะทำให้เกิดกระแสรั่วไหลเพียงเล็กน้อยเท่านั้น นอกจากนี้ยังทำให้เกิดบริเวณปลอดพาหะ (depletion region) ขึ้นที่รอยต่อ J_2 ด้วย บริเวณปลอดพาหะนี้จะขยายบริเวณกว้างขึ้นจนเกินเข้ามายังบริเวณ n^- มากกว่าที่จะขยายไปยังบริเวณชั้นบอดี้ p ทั้งนี้เพราะชั้นบอดี้ p มีความหนาแน่นในการได้ปัสสารมากกว่า ถ้าความหนาแน่นของสารที่ได้ปัสสารในบริเวณลอยเลื่อน n^- มากเพียงพอ ก็จะทำให้การขยายของบริเวณปลอดพาหะไม่สามารถแตะกับชั้นอินเจ็กต์ p^+ ได้ ชั้นบัฟเฟอร์ n^+ (buffer layer) (ดังรูป 4-2) ก็ไม่จำเป็นต้องทำให้เกิดขึ้นหรือไม่จำเป็นต้องได้ปัสสาร

ทั้งนี้เพราะการแตะกันของบริเวณทั้งสองจะทำให้เกิดการพังทลายทางด้านไบแอสตรง สำหรับ ไอจีบีทีที่ไม่มีการได้ปัสสารในชั้นบัฟเฟอร์ n^+ นี้ จะเรียกว่าไอจีบีทีแบบสมมาตร ซึ่งจะมีอัตราทนแรงดันย้อนกลับ (V_{RM} หรือ BV_{SDS}) สูงพอ ๆ กับค่าอัตราทนแรงดันไหลตรง (BV_{SDS}) เหมาะสำหรับการนำไปประยุกต์ใช้ในวงจรไฟฟ้ากระแสสลับ

การลดความหนาแน่นบริเวณ n^- ลงแต่ยังคงความสามารถของอัตราทนแรงดันไหลตรงไว้ สามารถทำได้โดยเพิ่มชั้นบัฟเฟอร์ n^+ เข้าไป เพื่อป้องกันการแตะกันของบริเวณปลอดพาหะกับบริเวณอินเจ็กต์ p^+ ซึ่งจะเรียกชนิดนี้ว่าไอจีบีทีแบบไม่สมมาตร และจากการลดความหนาแน่นของบริเวณลอยเลื่อน n^- ลงจะช่วยส่งผลให้เกิดข้อดีสองประการ คือ

- ทำให้แรงดันตกคร่อมขณะนำกระแสต่ำลง เป็นผลให้การสูญเสียกำลังงานลดน้อยลงด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ช่วยลดช่วงเวลาหยุดนำกระแสให้สั้นลงได้

แต่ข้อเสียของการเพิ่มชั้นบัฟเฟอร์ n^+ ก็มี คือจุดความสามารถของอัตราทนแรงดันย้อนกลับให้น้อยลงเหลือเพียงไม่กี่สิบโวลต์ ทั้งนี้เพราะเมื่อ ไอจีบีที ได้รับแรงดัน ไบแอส กลับที่ขาแคทรน รอยต่อ J_1 ซึ่งทั้งสองข้างมีความหนาแน่นในการโด๊ปของสารมากร จะไม่สามารถทนแรงดันย้อนกลับได้สูงดังนั้น ไอจีบีทีชนิดนี้จึงไม่เหมาะที่จะนำไปใช้ในวงจรไฟฟ้ากระแสกลับ

4.5 การแลตจในไอจีบีที

นอกจากโหนดส่วนใหญ่ที่รวมกับอิลีคตรอนภายใต้บริเวณ n^- แล้ว ยังมีกระแสโหนดบางส่วนที่ไหลข้ามบริเวณ n^- เข้าสู่บริเวณชั้นบอดี p โดยตรง ผลของกระแสโหนดนี้ทำให้เกิดแรงดันตกคร่อมความต้านทานข้างเคียง (Lateral resistance) ดังในรูป 4-2 ถ้าแรงดันนี้มีค่ามากพอคือประมาณ 0.7 โวลต์ จะทำให้รอยต่อ J_2 ได้รับไบแอสตรง เป็นผลให้อิลีคตรอนจากบริเวณซอร์ส n^+ ถูกฉีกเข้าไปในชั้นบอดี p

ถ้าดูจากวงจรสมมูลในรูป 4-5 จะหมายถึงขาเบสและอิมิตเตอร์ของทรานซิสเตอร์เอ็นพีเอ็นได้รับแรงดันไบแอสตรงส่งผลให้ ไทริสเตอร์ซึ่งแฝงอยู่ใน โครงสร้างของ ไอจีบีทีอยู่ในภาวะแลตจการนำกระแส ทำให้ที่ขาแคทรนไม่สามารถควบคุมปริมาณของกระแสแคทรนได้อีกต่อไป แต่การควบคุมกระแสแคทรนนี้จะขึ้นอยู่กับตัวต้านทานที่นำมาต่อในวงจรภายนอก ถ้าหากมีการแลตจเกิดขึ้นเป็นเวลานาน อาจทำให้ ไอจีบีทีเสียหายได้ เพราะมีการสูญเสียกำลังงานเกินค่าที่กักที่ทนได้

ส่วนใหญ่หรือเป็นมาตรฐานคู่มือของผู้ผลิต มักจะมีการบอกค่ากระแสแคทรนสูงสุด ที่สามารถไหลผ่าน ไอจีบีทีได้โดยยัง ไม่มีการแลตจขึ้น (I_{DB}) แต่เนื่องจากกระแสแคทรนถูกกำหนดหรือควบคุมโดยแรงดันระหว่างขาแคทรนกับซอร์ส บางครั้งคู่มือจึงบอกค่าแรงดันระหว่างแคทรนและซอร์สสูงสุดที่จะไม่ทำให้เกิดการแลต แทนการบอกค่ากระแสแคทรนสูงสุด (I_{DB})

การแลตที่กล่าวถึงข้างต้นเรียกว่าการแลตในโหมคสแตติค เพราะเกิดขึ้นเมื่อกระแสที่ไหลในสถานะนำกระแสมีค่าเกิน I_{DB} แต่ลักษณะการแลตนี้ก็สามารถเกิดขึ้นได้ เรียกว่าในโหมคไดนามิก ซึ่งจะเกิดขึ้นเมื่อมีการเปลี่ยนการทำงานจากสถานะนำกระแสเข้าสู่สถานะหยุดนำกระแสได้ค้ำย บางครั้งการแลตนี้อาจเกิดขึ้นได้แม้ว่ากระแสแคทรนขณะนำกระแสยังมีค่าต่ำกว่า I_{DB} ก็ตาม ทั้งนี้เพราะเมื่อ ไอจีบีทีเริ่มหยุดนำกระแส กระแสแคทรนจะตกลงอย่างรวดเร็วรอยต่อ J_2 จะต้องลบแรงดันย้อนกลับที่เพิ่มขึ้นอย่างรวดเร็วเช่นกัน

ผลที่เกิดขึ้นจะทำให้บริเวณปลอดพาหะขยายบริเวณชั้นบอดี p โดยเฉพาะขยายเข้าสู่บริเวณ n^- มากกว่า เพราะมีความหนาแน่นของการโด๊ปต่ำกว่า การขยายบริเวณปลอดพาหะอย่างรวดเร็ว จะทำให้ โหนดที่ค้างอยู่บริเวณ n^- ขณะนำกระแสและยังไม่ได้รวมกับอิลีคตรอนหลุดรอดจากการขัดขวางของบริเวณ

ปลอดพาหะเข้าไปสะสมอยู่ในบริเวณรอยต่อ J_2 เป็นการเพิ่มกระแสที่ไหลผ่านตัวต้านทานข้างเคียงให้

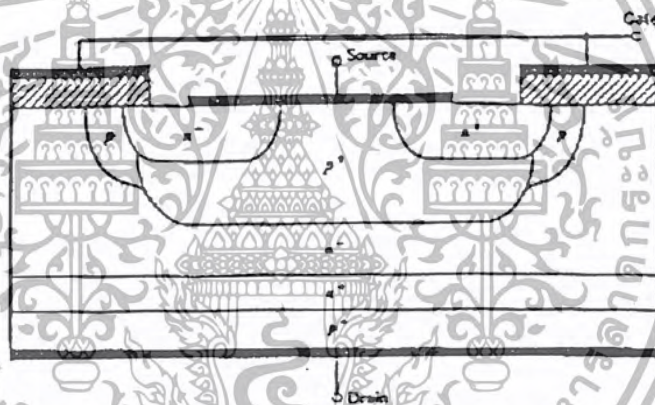
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สูงขึ้น ทำให้ทรานซิสเตอร์ภายในไอจีบีทีเกิดการแลตช์ขึ้นได้ นอกจากนี้การแลตช์ยังสามารถเกิดขึ้นได้อีก เนื่องจากการเพิ่มขึ้นของอุณหภูมิที่รอยต่อ ในขณะที่กระแสเดรนยังมีค่าต่ำกว่า I_{DM} อยู่เช่นกัน

4.6 การป้องกันการแลตช์

การหลีกเลี่ยงการแลตช์ของ IGBT สามารถทำได้ทั้งผู้ผลิตและผู้ใช้งานเอง ผู้ผลิตอาจจะออกแบบโครงสร้างในส่วนบริเวณบอดี p ให้มีความต้านทานข้างเคียงต่ำสุด เพื่อจะได้เพิ่มค่าของ I_{DM} ให้มากที่สุด ซึ่งเป็นการลดโอกาสที่จะเกิดการแลตช์ลงได้

วิธีแรกอาจทำได้โดยลดความกว้างของบริเวณซอร์ส n^+ ลง นั่นคือลดค่า L_S ที่แสดงในรูปที่ 4.2 ลงนั่นเอง วิธีที่สองเป็นการแบ่งระดับความหนาแน่นในการได้ปัสสารของบริเวณบอดี p ดังในรูปที่ 4.6 จะเห็นว่าบอดี p ภายใต้ออกจะไต่ไปด้วยความหนาแน่นในระดับปกติ 10^{16}cm^{-3} และมีความหนาน้อยกว่าของบริเวณซอร์ส n^+ แต่ส่วนอื่นที่เหลือของบริเวณบอดี p จะไต่ไปด้วยความหนาแน่นที่มากกว่า 10^{19}cm^{-3} รวมถึงความหนาแน่นก็จะมากกว่าด้วย



รูปที่ 4-6 โครงสร้างที่ปรับปรุงเพื่อป้องกันการแลตช์ในของไอจีบีที

สำหรับผู้ใช้งานก็สามารถป้องกันการแลตช์ใน โหมคคตแคตติค ได้ โดยออกแบบไม่ให้กระแสที่ไหลในโหมคต เกินค่ากระแส I_{DM} และห้องกันการแลตช์ใน โหมคตโคนามิกได้ โดยหน่วงเวลาขณะหยุดนำกระแสให้ยาวนานขึ้น เพื่อให้โหมคตที่ยังค้างอยู่ในบริเวณ n^- มีเวลาพอที่จะรวมกับอิเล็กตรอนเป็นการลดกระแสที่ไหลผ่านความต้านทานข้างเคียงให้น้อยลงได้ การหน่วงเวลาขณะหยุดนำกระแสให้ยาวนานขึ้นทำได้โดยเพิ่มความต้านทานภายนอกอนุกรมเข้ากับขาเกตของ ไอจีบีที

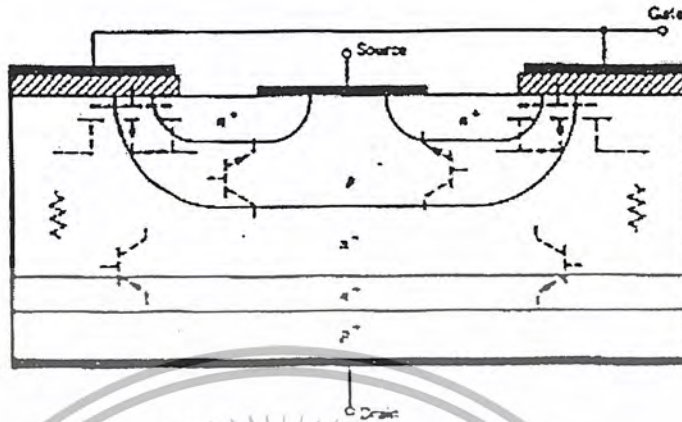
4.7 วงจรสมมูลของไอจีบีที

วงจรสมมูลของไอจีบีทีแสดงไว้ในรูปที่ 4-6 ซึ่งในรูปที่ 4-6 (ก) นั้นจะเห็นว่าในบริเวณบอดี p

ชั้นบริเวณ n^- และชั้นอินเจ็คติง p^+ จะคล้ายกับทรานซิสเตอร์ชนิดพีเอ็นพี โดยแทนได้ด้วยขาคอลเล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เตอร์, เบส และอิมิตเตอร์ ตามลำดับ และบริเวณ n^- เชื่อมขาเบสของทรานซิสเตอร์พีเอ็นพีเข้ากับขาแคโรนของมอสเฟต ซึ่งเมื่อเขียนวงจรสมมูลออกมาจะได้วงจรดังรูปที่ 4-7



รูปที่ 4-7 โครงสร้างที่มีไบโพลาร์ทรานซิสเตอร์และมอสเฟตแฝงอยู่ภายใน

จากรูปที่ 4-7 จะเห็นว่าเป็นวงจรคาร์ลิงตัน โดยมีมอสเฟตเป็นตัวขับทรานซิสเตอร์พีเอ็นพี แต่มีจุดพิเศษที่แตกต่างจากวงจรทั่วไป คือ กระแสแคโรนส่วนใหญ่จะไหลจากอิมิตเตอร์มายังเบส ผ่านความต้านทานบริเวณลอยเลื่อน และผ่านขาแคโรนของมอสเฟตจบลงที่ขาซอร์ส จะมีกระแสส่วนน้อยเท่านั้นที่ไหลจากอิมิตเตอร์มายังคอลเล็กเตอร์และขาซอร์ส สำหรับวงจรสมมูลในรูปที่ 4-7 แสดงให้เห็นว่าภายในไอจีบีทีมีทรานซิสเตอร์แฝงอยู่ด้วย โดยดูได้จากที่ทรานซิสเตอร์ เอ็นพีเอ็นและพีเอ็นพี ต่อเข้าด้วยกันในลักษณะที่มีการป้อนกลับ ทำให้เห็นได้ชัดถึงเหตุที่ทำให้เกิดการแลตซ์ของ ไอจีบีที

โดยถ้ากระแสส่วนน้อยที่ไหลผ่านจากอิมิตเตอร์มายังคอลเล็กเตอร์ของทรานซิสเตอร์พีเอ็นพีผ่านความต้านทานข้างเคียงแล้ว ทำให้เกิดแรงดันตกคร่อมความต้านทานสูงกว่า 0.7 โวลต์ ทรานซิสเตอร์เอ็นพีเอ็นจะนำกระแส ส่งผลให้เกิดการแลตซ์ขึ้นในไอจีบีทีสำหรับแรงดันตกคร่อมขาแคโรน และซอร์สของไอจีบีทีขณะนำกระแส ($V_{DS(ON)}$) สามารถเขียนเป็นสมการได้ดังนี้

$$V_{DS(ON)} = V_{J1} + V_{drift} + I_D R_{channel}$$

เมื่อ V_{J1} เป็นแรงดันไบแอสตรงที่ตกคร่อมรอยต่อพีเอ็น จึงมีค่าค่อนข้างคงที่ จะมีกระแสเปลี่ยนแปลงบ้างก็เพียงเล็กน้อย เพราะมีความสัมพันธ์โดยตรงในลักษณะเอ็กซ์โปเนนเชียลกับกระแสทำให้ค่าอยู่ระหว่าง 0.7 - 1 โวลต์

V_{drift} เป็นแรงดันที่ตกคร่อมความต้านทานบริเวณลอยเลื่อน ซึ่งความต้านทานนี้มีค่าค่อนข้างคงที่ แต่เมื่อเทียบกับค่าแรงดันในมอสเฟตแล้วจะมีค่าน้อยกว่า เพราะผลของการออคูเลตสภาพน้ำที่เกิดขึ้นในไอจีบีที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

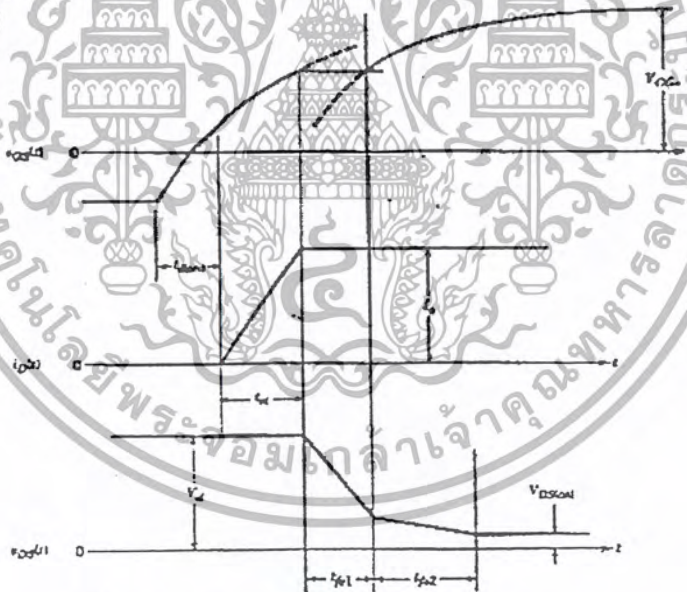
R_{channel} เป็นค่าความต้านทานในย่าน 1-1000 โอห์ม มีค่าค่อนข้างจะคงที่

$(I_D R_{\text{channel}})$ เป็นแรงดันตกคร่อมมอสเฟต

ดังนั้นจึงพอจะสรุปได้ว่า $V_{\text{DS(ON)}}$ จะมีค่าสูงมากขึ้นตามค่ากระแสเดรนที่สูงขึ้น โดยทั่วไปจะสามารถทำงานได้ในอุณหภูมิรอยต่อสูงสุดถึง 150 องศาเซลเซียส และผลของการเปลี่ยนอุณหภูมิจากค่าอุณหภูมิห้องไปจนถึงค่าสูงสุดนี้ จะส่งผลให้เกิดการเปลี่ยนแปลงค่า $V_{\text{DS(ON)}}$ เป็นผลรวมระหว่างแรงดันตกคร่อมมอสเฟตที่มีสัมประสิทธิ์ทางอุณหภูมิเป็นบวก (หมายถึงอุณหภูมิสูงขึ้นแรงดันตกคร่อมก็จะสูงขึ้นตาม) กับแรงดันตกคร่อมความต้านทานบริเวณรอยเลื่อนที่มีสัมประสิทธิ์อุณหภูมิลบ

4.8 ลักษณะการสวิทช์

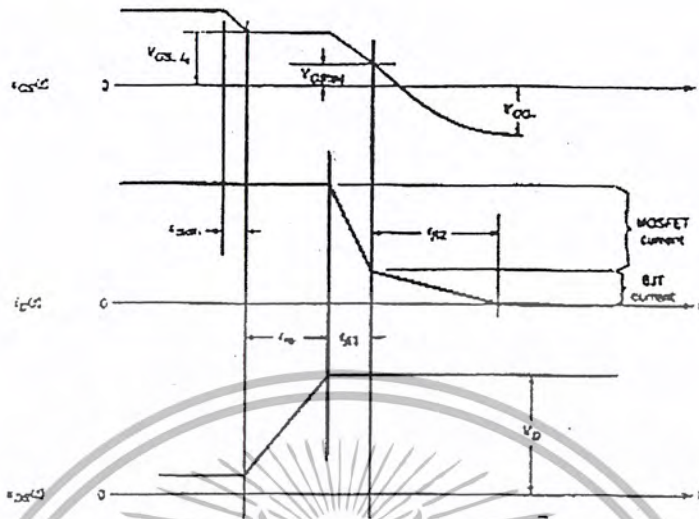
ลักษณะของสัญญาณกระแสและแรงดันในช่วงเวลาที่เกิดการนำกระแสและหยุดกระแสแสดงไว้ในรูปที่ 4.8 ช่วงเวลาในการนำกระแสของ ไอจีบีที แสดงไว้ในรูปที่ 4.8 ซึ่งจะมีลักษณะคล้ายกับการนำกระแสของมอสเฟต คือ จะมีเวลาดำเนินการนำกระแส (t_{on}) นับตั้งแต่เวลาที่แรงดันระหว่างเกตกับซอร์สอย่างในช่วง $V_{\text{GG-}}$ จนถึง $V_{\text{GS(th)}}$ ความจริงแล้วการป้อนแรงดันนี้จะมีลักษณะการเปลี่ยนแปลงทันทีทันใดจากค่า $V_{\text{GG-}}$ เป็น $V_{\text{GG+}}$ แต่กลับมีลักษณะเป็นเอ็กซ์โปเนนเชียล



รูปที่ 4 – 8 ลักษณะของกระแสและแรงดันขณะนำกระแส

เหตุที่เป็นเช่นนั้นเนื่องจากผลการชาร์จประจุของตัวเก็บประจุระหว่างเกตกับซอร์สและเกตกับเดรนภายในไอจีบีทีแรงดันที่ขาเดรนจะอยู่คงที่ในช่วงเวลาขาขึ้น (t_{r}) หรือในช่วงเวลาที่กระแสเดรนยังไม่ถึงค่ากระแสทำงาน (I_{D}) หลังจากนั้นกระแสเดรนก็จะคงที่ แต่แรงดันจะตกลงสู่ค่า $V_{\text{DS(on)}}$ โดยแบ่ง

ช่วงเวลาลงเป็นสองช่วง คือ ช่วง t_{fv1} เป็นช่วงที่ทำงานอยู่ในย่านความต้านทานสูง ($R_{channel}$) ส่วน t_{fv2} ช่วงที่ทำงานอยู่ในย่านความต้านทานต่ำ ($R_{channel}$)



รูปที่ 4-9 ลักษณะของกระแสและแรงดันขณะหยุดนำกระแส

จากรูปที่ 4-9 จะเป็นรูปแสดงลักษณะของกระแสและแรงดัน ในช่วงเวลาที่ไอจีบีทีหยุดนำกระแส จะเห็นว่ากระแสเดรนจะยังคงที่อยู่ตลอดเวลาที่แรงดันขาเดรนเพิ่มขึ้น และช่วงเวลาลงที่กระแสเดรนที่แตกต่างชัดเจนสองช่วง โดยช่วงแรก t_{s1} จะเป็นช่วงหยุดนำกระแสของมอสเฟตภายในไอจีบีที และช่วง t_{s2} จะเป็นช่วงหยุดนำกระแสของทรานซิสเตอร์พีเอ็นพี ซึ่งจะช้ากว่ามอสเฟต ทำในช่วงเวลานี้ นานกว่าช่วงแรกและมีการสูญเสียกำลังงานมากกว่าในช่วงนี้

4.9 พื้นที่การทำงานที่ปลอดภัย

ไอจีบีที มีพื้นที่การทำงานที่ปลอดภัยทั้งในระหว่างนำกระแสและหยุดนำกระแส โดยมีพื้นที่การทำงานปลอดภัยในขณะไปแอสตรง (Forward Bias Safe Operation Area : FBSOA) ที่กว้างมากเปรียบได้กับเกือบเป็นสี่เหลี่ยม สำหรับเวลาในการสวิทซ์ที่สั้น ๆ แต่จะแคบลงเมื่อเวลาในการสวิทซ์ยาวนานขึ้น ซึ่งถ้าเทียบกับเพาเวอร์มอสเฟตแล้ว ไอจีบีทีจะทำงานได้ในช่วงพื้นที่ที่กว้างกว่าเมื่อเวลาในการสวิทซ์เท่ากัน

บทที่ 5

ไมโครคอนโทรลเลอร์ และการเขียนโปรแกรม

5.1 บทนำ

ไมโครคอนโทรลเลอร์คือคอมพิวเตอร์รูปแบบหนึ่ง ซึ่งมีขนาดเล็กและเน้นการใช้งานในแบบฝังตัว (Embedded) คืออยู่ในรูปของบอร์ดที่อยู่ภายในสินค้าต่าง ๆ อีกที เรามักจะเรียกกันว่า "แผงวงจร" ความจริงบอร์ดเหล่านี้ อาจจะเป็นเพียงวงจรอิเล็กทรอนิกส์ธรรมดา หรืออาจเป็นวงจรในรูปแบบดิจิทัล (Digital) ก็ได้ แต่ในปัจจุบันจะพบบอร์ดในรูปแบบไมโครคอนโทรลเลอร์มากขึ้น เนื่องด้วยการทำงานที่มีความยืดหยุ่นสูง ง่ายต่อการปรับปรุงแก้ไข ใช้ได้ดีกับงานที่มีความซับซ้อนมาก และที่สำคัญก็คือ บอร์ดไมโครคอนโทรลเลอร์จะเปลี่ยนแนวทางการพัฒนาจากที่เน้นด้าน Hardware ไปเป็นเน้นทางด้าน Software แทน ตัวอย่างงานเหล่านี้ได้แก่ บอร์ดเก็บข้อมูลเครื่องดนตรี บอร์ดควบคุมแผงแสดงตัวอักษร บอร์ดเครื่องนับจำนวน บอร์ดตั้งเวลาไมโครคอนโทรลเลอร์ยังประยุกต์ใช้งานอื่น ๆ ได้อีกมากมาย จนแทบจะเรียกได้ว่าสามารถพัฒนาและสร้างสรรค์ได้ตามแต่จินตนาการ

5.2 แนวทางการพัฒนาโปรแกรมในงานไมโครคอนโทรลเลอร์

ไมโครคอนโทรลเลอร์ หรือ แผงวงจรในรูปแบบ Embedded Board จำเป็นจะต้องมีขั้นตอนในการพัฒนาโปรแกรม เพราะถ้าพึ่งเพียงบอร์ดที่เป็น Hardware นั้น ยังไม่สามารถนำไปใช้งานอะไรได้ วิธีการพัฒนามีหลากหลายรูปแบบ แล้วแต่ลักษณะของบอร์ด หรือแล้วแต่ความเหมาะสมของงานและราคา ซึ่งจะสรุปได้ดังนี้

5.2.1 พัฒนาในรูปแบบ Single Board

วิธีนี้จะใช้บอร์ดประเภทที่มีส่วนแสดงผล (ส่วนใหญ่จะเป็นตัวเลข 7-Segment LED) และมีส่วนคีย์บอร์ดที่สามารถดูและป้อนข้อมูลเข้าไปได้ในตัว บอร์ดประเภทนี้เป็นที่คุ้นเคยในการเรียนตามสถาบันการศึกษาต่าง ๆ ในบ้านเรามาก การป้อนข้อมูลมักจะกระทำในระดับเลขฐานสิบหก ซึ่งสอดคล้องกับภาษา Assembly ที่ผู้เรียนจะต้องทำการแปลงจากตัวภาษา Assembly มาเป็น Op-code อีกที ซึ่งก็คือว่าไม่ เป็นปัญหาสำหรับโปรแกรมเพื่อการเรียนรู้ เนื่องจากมีความยาวไม่มาก บอร์ดประเภทนี้ยังมีความสะดวกในเรื่องของ System Call คือผู้ใช้สามารถเรียกโปรแกรมย่อยต่าง ๆ จากตัวโปรแกรม Monitor ของบอร์ดได้ เช่น โปรแกรมส่วนรับคีย์บอร์ด ซึ่งทำให้การเขียนโปรแกรมสะดวกยิ่งขึ้น และเนื่องจากมีส่วนแสดงผลและคีย์บอร์ดในตัว จึงทำให้การเขียนโปรแกรมสามารถเขียนเพื่อการทดลองได้สะดวกเช่น สามารถแสดงค่าต่าง ๆ บน Display ได้ทันที อย่างไรก็ตาม บอร์ดประเภทนี้จะต้องยอมเสียเวลาเรียนรู้วิธีการใช้งาน และอาจจะไม่เหมาะกับการพัฒนาโปรแกรมยาวๆ (โปรแกรมที่ใช้งานจริง) ซึ่งโปรแกรมที่มีความยาวมากๆ ควรจะมีการเขียนโปรแกรมบนเครื่องคอมพิวเตอร์ PC มากกว่า และทำการแปลเป็น Op-code บนเครื่อง PC เลย (เรียกว่าการทำ Assembler) จากนั้นจึงนำ Op-code มาโหลดเข้าบอร์ดไมโครอีกที บอร์ดประเภท Single Board ส่วนใหญ่ก็จะสามารถโหลดโปรแกรมจากเครื่อง PC ได้เช่นกัน แต่ราคาคงจะสูงเกินไป ถ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ใช้ Single Board แต่ใช้คุณสมบัติเพียงแค่นี้ Single Board ยังมีข้อจำกัดในเชิง Hardware อยู่บ้าง เนื่องจากมีโปรแกรม Monitor อยู่ในตัว เพราะฉะนั้นการทดลองในเรื่องเล็ก ๆ บางอย่างอาจจะทำไม่ได้ หรือทำได้ไม่สมบูรณ์ เช่น การทดลองเรื่องระบบ Interrupt นอกจากนี้ ถ้ามองในมุมของความเป็นชิปเดี่ยว (Single chip) แล้ว บอร์ดประเภทนี้จะมีข้อด้อยไปด้วย เนื่องจากเราไม่สามารถใช้ความเป็นชิปเดี่ยวได้เต็มที่ เนื่องจาก I/O ของตัวชิปจะถูกนำไปใช้เพื่อการจัดหน่วยความจำ โดยสรุปบอร์ดประเภทนี้เหมาะสำหรับการเรียนรู้ ตั้งแต่ขั้นเริ่มต้น และใช้เป็นบอร์ดทดลองต่าง ๆ ที่ต้องการความคล่องตัว สะดวก และรวดเร็ว

5.2.2 พัฒนารูปแบบ Embedded Board ร่วมกับชิปแบบ Softchip

วิธีการนี้จะคล้าย ๆ กับแบบ Single Board แต่แทนที่จะใช้ดีวีบอร์ด และส่วนแสดงผลบนบอร์ดก็ใช้จากเครื่องคอมพิวเตอร์ PC แทน ทั้งนี้โดยอาศัยการสื่อสารผ่าน RS232 ร่วมกับโปรแกรมบน PC ที่เป็น Terminal ธรรมดา เช่น Xtalk, Procom หรือ Hyper Terminal บน Windows เมื่อเรากดคีย์ใด ๆ บน PC ข้อมูลจะถูกส่งมายัง Embedded Board ทันที และโปรแกรม Monitor ที่อยู่ใน Softchip ก็จะได้ความหมายและตอบสนองส่งกลับไปยังจอภาพบนเครื่อง PC วิธีการนี้ทำให้บอร์ดมีราคาถูกลง เนื่องจากไม่ต้องมีส่วนดีวี และส่วนแสดงผล ขณะเดียวกันยังสามารถเลือกบอร์ดได้หลายรุ่น ตามคุณสมบัติต่าง ๆ ที่ต้องการ และศักยภาพในการทำงานทั่วไปจะสูงกว่า เพราะว่าดีวีบอร์ด และจอภาพของเครื่องคอมพิวเตอร์ PC จะใช้สะดวกและดูข้อมูลได้มากกว่าแผงตัวเลขบน Single Board ตัว Softchip ก็มีหลายแบบ คือเป็นตัว Eprom หรืออาจจะเป็นตัวชิปไมโครคอนโทรลเลอร์ที่มีโปรแกรมอยู่ภายในเลย อย่างไรก็ตาม วิธีการนี้ยังคงมีเนื้อหาการพัฒนาในตัวเองเดียวกับ Single Board โดยบนบอร์ดจะต้องมี Memory เพื่อให้ผู้ใช้ทดลองโปรแกรมได้ การเขียนโปรแกรมก็จะต้องกำหนด Address ตามตำแหน่งของบอร์ดนั้น ๆ และที่สำคัญจะต้องมีเครื่องคอมพิวเตอร์ PC เสมอ

5.2.3 พัฒนาคู่มือเครื่องมือ Eprom Emulator

วิธีนี้นับว่าเป็นวิธีที่ดีพอสมควร เพราะใช้กันมานานแล้ว และยังคงใช้งานกันอยู่ในปัจจุบันโดยหลักการก็คือใช้เครื่องมือที่ทำหน้าที่เสมือนเป็นตัว Eprom ตัวหนึ่ง แต่ไม่ต้องทำการโปรแกรมเข้าตัว Eprom จริง ๆ แต่ใช้การโหลดข้อมูลจากเครื่องคอมพิวเตอร์ PC แทน เครื่องมือดังกล่าวจะเรียกว่า Eprom Emulator ซึ่งจะมีส่วนที่เป็นเหมือนขาตัว Eprom ที่สามารถนำไปเสียบกับบอร์ดเป้าหมายได้ทันที วิธีนี้ช่วยให้การพัฒนาทำได้ง่าย เพราะไม่ต้องเสียเวลากับการโปรแกรมและล้างตัว Eprom จริง ๆ นอกจากนี้ยังเป็นการพัฒนาขั้นพื้นฐานที่สุด นั่นคือสามารถใช้กับบอร์ดใด ๆ ก็ได้ ไม่ขึ้นอยู่กับเบอร์ของไมโคร และเป็นบอร์ดจากผู้ผลิตรายใดก็ได้ รวมทั้งบอร์ดที่พัฒนาขึ้นเอง หรือแม้กระทั่งการทดลองบน Photo Board การพัฒนาวิธีนี้ยังเน้นกับไมโครที่ต้องการ Memory ภายนอก ซึ่งอาจจะไม่สะดวกกับไมโครรุ่นหลัง ๆ ในแบบ Single Chip ที่มี Memory อยู่ในตัวชิปเลย อีกประการหนึ่ง การพัฒนาแบบนี้จะเริ่มต้นโปรแกรมที่ Origin 0000h (โดยทั่วไป) ซึ่งมองเป็นข้อดีหรือข้อเสียก็ได้ ข้อดีคือเป็นการพัฒนาโปรแกรมของตัวเองล้วน ๆ ไม่ต้องเสียเวลากับการทำความเข้าใจโปรแกรม Monitor เลย หรือข้อเสียก็คือ ไม่สามารถทำการ Debug ได้นั่นเอง ซึ่งจุดนี้คงต้องแล้วแต่ผู้พัฒนาจะพิจารณาเอง โดยภาพรวม วิธีนี้จะเหมาะกับผู้ที่พัฒนา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในระดับเริ่มต้น คือสร้างบอร์ดขึ้นมาใช้เอง รวมทั้งผู้ที่ใช้งานชิป ไมโครหลาย ๆ เบอร์ ก็สามารถใช้ Eprom Emulator เพียงบอร์ดเดียวก็พัฒนางานได้กับบอร์ดทุก ๆ บอร์ด

5.2.4 พัฒนาด้วยเครื่องโปรแกรมตัว Single Chip

วิธีนี้นับว่ามาแรงในปัจจุบัน ด้วยเทคโนโลยีแบบ Flash Memory ในปัจจุบัน ทำให้มีการผลิตไมโครคอนโทรลเลอร์ที่มี Flash Memory อยู่ภายในตัวชิปเลย (Single Chip) และด้วย Flash Memory นี้เอง เราสามารถโปรแกรมและล้างได้ถึง 1000 ครั้ง การล้างก็ทำได้ง่ายเพียงแค่การกระทำทางไฟฟ้าเท่านั้น ไม่ต้องให้แสง UV เหมือนตัว Eprom อีก การพัฒนาจะวนเวียนกับการ โปรแกรม-ทดลอง-ล้าง และจะต้องนำตัวชิปเสียบเข้าออกระหว่างเครื่อง โปรแกรม กับบอร์ดเป้าหมายอยู่ตลอด คล้าย ๆ กับการพัฒนาด้วยตัว Eprom สมัยก่อนแต่อย่างไรก็ตาม ขั้นตอนเหล่านี้ก็ทำได้เร็วกว่าตัว Eprom มาก และเนื่องจากทำได้ถึง 1000 ครั้ง จึงแทบจะไม่มี การ สูญเปล่าของตัวชิปที่ทดลองเลย วิธีนี้เป็นการพัฒนาที่ Origin 0000h ด้วย และใช้กับบอร์ดที่มีขายกันอยู่ หรือจะพัฒนาบอร์ดขึ้นมาใหม่ก็ได้ แต่จะต้องเป็นตระกูลของชิปที่เครื่อง โปรแกรมนั้น ๆ กระทำได้ จุดเด่นอีกประการหนึ่งก็คือ การพัฒนาแบบนี้ให้ผลต่อเนื่องของงานทุกขั้นตอน คือเมื่อทำคั่นแบบเสร็จแล้ว ก็สามารถใช้เครื่อง โปรแกรมที่มีอยู่ สร้างชิปเพิ่มเติมสำหรับงานผลิตที่มีจำนวนมาก ๆ ได้ด้วย ซึ่งถ้าเป็นวิธีอื่น ๆ ที่กล่าวมา อาจจะต้องมีการหาเครื่องมือเพิ่มเติมใหม่อีก การสั่งซื้อตัวชิปก็ทำเพียงรายการเดียว ใช้ได้ทั้งการพัฒนาจนถึงการผลิตขาย นอกจากนี้ เครื่องโปรแกรม Single Chip ก็ยังสามารถใช้ได้กับชิปหลาย ๆ เบอร์ หลาย ๆ ขนาด ทำให้เครื่องมือเพียงตัวเดียว สามารถใช้กับงานทุกระดับนับว่าเป็นการคุ้มค่าอย่างมาก

5.2.5 พัฒนาด้วยเครื่องโปรแกรมผ่าน SPI-Port

วิธีการนี้จะมีหลักการคล้าย ๆ กับวิธีที่ 4 แต่แทนที่จะโปรแกรมตัวชิป และเสียบเข้า, ออก ระหว่างเครื่อง โปรแกรมกับบอร์ดเป้าหมาย ก็จะใช้การ โปรแกรมลงบนบอร์ดเป้าหมายเข้าในตัวชิปได้โดยตรง ทั้งนี้เป็นคุณสมบัติของชิปที่มี SPI-Port อยู่ในตัว วิธีการนี้นับว่าทันสมัยแต่ตัวชิปราคาค่อนข้างสูง และบอร์ดเป้าหมาย ก็จะต้องออกแบบไว้ให้มีจุดเพื่อการ โปรแกรมได้ด้วยรวมทั้งการ โปรแกรมจะทำได้ช้ากว่าการใช้เครื่อง โปรแกรม Single Chip ประมาณ 1 เท่า อย่างไรก็ตาม วิธีนี้นับว่าคุ้มค่ามากในภาพรวม เพราะเพียงบอร์ดเป้าหมายที่จะพัฒนากับบอร์ดที่ช่วยโหลดโปรแกรม เท่านั้นที่เพียงพอต่อการพัฒนางานได้แล้ว โดยจะต้องใช้ร่วมกับเครื่องคอมพิวเตอร์ PC ด้วย

วิธีการทั้งหมดที่กล่าวมานี้ ไม่ใช่วิธีการทั้งหมดที่ใช้กันอยู่ในปัจจุบันเพราะยังมีวิธีอื่น ๆ อีก เช่น การใช้เครื่องมือแบบ Emulator ที่มีกจะมีราคาแพงมาก ๆ หรือวิธีการในรูปแบบ Eprom, OTP คือการพัฒนาด้วยตัวชิปแบบ Eprom แล้วจึงนำไปผลิตด้วยชิปแบบ OTP (One Time Programming) ซึ่งมีราคาถูกกว่า อย่างไรก็ตามวิธีการที่ได้รวบรวมมานี้ จะเหมาะสำหรับการพัฒนางานในระดับ Local (ภายในประเทศ) ซึ่งจะมีจำนวนการผลิตในระดับ 10,100,1000 ชิ้น ซึ่งภาพต่าง ๆ ของวิธีการพัฒนานี้จะทำให้พอเห็นเป็นแนวทางที่ชัดเจนขึ้นและสามารถเลือกให้เหมาะสมกับผู้สนใจได้ตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 PIC

PIC คือ Microcontroller อีกตระกูลหนึ่ง ย่อมาจากคำว่า Peripheral Interface Controller ซึ่ง concept ของผู้ผลิต Microcontroller ตระกูลนี้ก็คือ พยายามรวมเอาทุกอย่างเอาไว้ในตัวชิปไม่ว่าจะเป็น PROGRAM MEMORY, RAM, EEPROM, SERIAL, I2C, PWM, A/D ฯลฯ โดยไม่จำเป็นต้องต่ออุปกรณ์เสริมจากภายนอก ในตัวของ PIC จะมีฟังก์ชันที่ใช้ในการประมวลผล รวมทั้งหน่วยความจำ ซึ่งทำให้คล้ายกันเป็น CPU ตัวหนึ่ง

5.3.1 ความเร็วของ PIC

ภาคของความถี่สัญญาณนาฬิกา ปัจจุบันสามารถทำสัญญาณนาฬิกาได้ที่ 20 MHz ซึ่งทำให้หนึ่งคำสั่งของ PIC ใช้เวลาเพียง 0.25 uSec แต่อย่างไรก็ตามได้มีบริษัทอื่นได้ซื้อลิขสิทธิ์ PIC จาก microchip และได้สร้าง chip ที่มีความเร็วได้มากกว่าเดิมขึ้นไปอีก

5.3.2 หน่วยความจำของ PIC

ในอดีตหน่วยความจำของ PIC จะค่อนข้างน้อย คืออยู่ระหว่าง 512 words ถึง 4K words แต่ในปัจจุบัน บริษัท microchip ซึ่งเป็นเจ้าของ PIC ได้พัฒนาจนทำให้ memory ของ PIC มีขนาดเป็นหลายสิบกิโลไบต์ และมีที่ท่าว่าจะขยายได้ใหญ่ขึ้นเรื่อยๆ ในเรื่องของการนับขนาดของหน่วยความจำของ PIC จะนับไม่เหมือนปกติ โดยที่ หนึ่งคำสั่งของ PIC จะมีขนาด 14 bits ดังนั้นจะเรียกว่า 1 word ของ PIC จะมีขนาด 14 bits เช่น PIC16F84A ระบุว่ามีความจำ 1 K (ซึ่งหมายถึง 1 Kword) ถ้าคำนวณให้เป็นแบบ 1 byte = 8 bit จะได้ว่า $1 \times 1,024 \times 14 = 14,336 \text{ bits}$ ดังนั้นก็คือ $14,336 / (8 \times 1,024) = 75K \text{ bytes}$

5.3.3 สถาปัตยกรรมของ PIC

ปัจจุบันนี้มี 3 สายหลักๆ จากอดีตมีเพียงสองสาย คือขึ้นต้นด้วย 16xxx, 17xxx และใหม่ล่าสุดคือ 18xxx ถ้าเปรียบเทียบ คุณสมบัติที่เหนือกว่าเรียงจากน้อยสุด ไปมากที่สุดดังนี้ $16 > 17 > 18$ คำสั่ง assembly ของ 17 และมี 18 จะมีมากกว่า 16 ทำให้เขียน โปรแกรมได้ง่ายกว่า ราคาจะสูงกว่าด้วย แต่ที่เป็นที่นิยมคือตระกูล 16xxx

5.4 ชนิดของ PIC

MCU ในตระกูล PIC แบ่งออกตามชนิดของ PROGRAM MEMORY ได้เป็นบทย่อยคือ

1. OTP (one time programmable)
2. EPROM (erasable programmable ROM)
3. EEPROM / Flash (electronically erasable programmable ROM)

5.4.1 OTP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นชิพที่มีราคาถูกที่สุด ในสามประเภท สาเหตุก็มาจากว่า chip แบบ OTP จะสามารถทำการโปรแกรมได้ เพียงครั้งเดียวเท่านั้น หากจาก chip ใดถูกโปรแกรมไปแล้วจะไม่สามารถโปรแกรมเข้าไปใหม่ได้อีก ดังนั้น chip ประเภทนี้จะนิยมใช้หลังจากได้พัฒนาโปรแกรมจนกระทั่งจุดบกพร่องต่างๆ ในโปรแกรม ไม่มีอีกแล้ว เพราะจะมีต้นทุนต่ำเมื่อเทียบตัว memory ประเภทอื่น จะมีตัวอักษร C แสดงบนตัวชิพ



40-LEAD PDIP
"P" OR "PL"

รูปที่ 5-1 ชิพ 40 ขาแบบ OTP

5.4.2 EPROM

เป็น chip ที่มี program memory ที่เมื่อเขียนโปรแกรมเข้าไปแล้วสามารถโปรแกรมใหม่ด้วยการลบโปรแกรมเดิม โดยให้แสง uV (ultra Violet) ส่องผ่านเข้าไปยัง chip ประมาณ 5-10 นาที ดังนั้นที่ด้านบนของ chip จะมีกรอบกระจกเพื่อให้แสง uV สามารถส่องผ่านเข้าไปในตัว chip ได้ แต่ก็มีจำนวนครั้งในการลบโปรแกรม เมื่อลบโปรแกรมด้วยแสง uV มากๆ ก็จะทำให้เกิดอาการค้าง คือโปรแกรมไม่เข้า จะมีตัวอักษร JW



40-LEAD CERDIP
"JW"

รูปที่ 5-2 ชิพ 40 ขาแบบ EPROM

5.4.3 EEPROM / Flash

เป็น chip ที่ออกมาไม่กี่ปีเอง ส่วนของ program memory สามารถอ่านหรือเขียนด้วยสัญญาณทางไฟฟ้า ใช้เวลาในการ ลบข้อมูลไม่กี่วินาที และสามารถลบ และเขียนใหม่ได้หลายพันครั้ง ทำให้เป็นที่นิยมที่สุดใน 3 ประเภท มีตัวอักษร F เป็นตัวบอก เช่น 16F84, 16F877, 17FXX, 18F452, 18F458



40-LEAD PDIP
"P" OR "PL"

รูปที่ 5-3 ชิพ 40 ขาแบบ EEPROM / Flash

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.5 PICเบอร์ต่างๆ

ปัจจุบัน MCU ของ PIC มีหลากหลายเบอร์ สามารถแบ่งออกเป็น 6 กลุ่ม ได้แก่

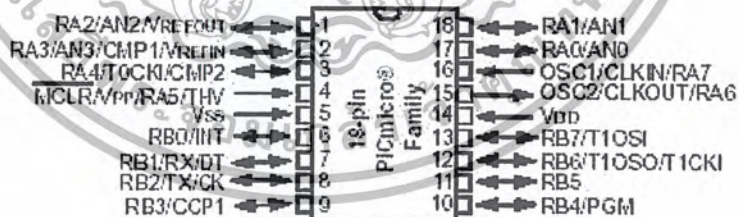
1. PIC12CXXX, PIC12FXXX (FLASH MCUs)
2. PIC16C5X
3. PIC16CXXX
4. PIC17CXXX
5. PIC16FXXX (FLASH MCUs)
6. PIC18CXXX, PIC18FXXX (FLASH MCUs)

จุดเด่นของแต่ละกลุ่ม

PIC165X

1. มีคำสั่งในภาษา assembly 33 คำสั่ง
2. มี I/O, Timer, Watch dog
3. มี feature ค่อนข้างน้อย จะ ไม่มี I2C หรือ SERIAL เพราะเป็น chip ที่ออกมาในยุคแรกๆ ของ Microchip เนื่องจากเป็นเบอร์ที่ออกมานานมากแล้ว จึงมีแต่แบบ OTP และ EPROM feature ก็มีน้อย I2C หรือ USART นี้ต้องเขียน โปรแกรมใช้เอง Microchip

18-pin PICmicro[®] MCU Family

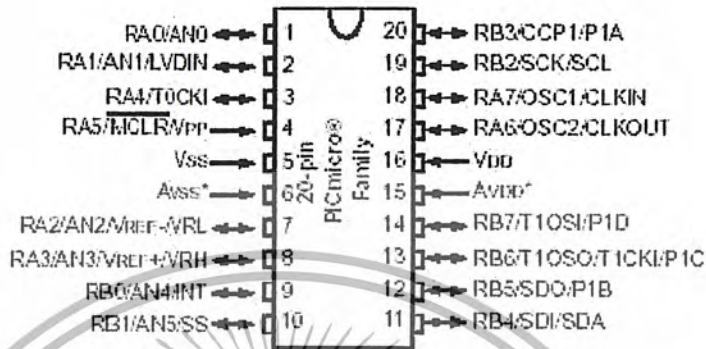


PIC16C620A	PIC16CE625	PIC16C710	PIC16F84A
PIC16CR620A	PIC16F627	PIC16C711	PIC16F818
PIC16C621A	PIC16F628	PIC16C712	PIC16F819
PIC16C622A	PIC16F627A	PIC16C715	PIC16F87
PIC16CE623	PIC16F628A	PIC16C716	PIC16F88
PIC16CE624			

รูปที่ 5-4 ชิป 18 ขาตระกูล PIC16CXXX, PIC16FXXX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

20-pin PICmicro[®] MCU Family



PIC16C747*

PIC16C771

PIC18F1220*

PIC16C770

PIC16C781

PIC18F1320*

PIC16C782

*18 pin device; Avss and Avdd are not valid pins for this device

รูปที่ 5-5 ชิป 20 ขาตระกูล PIC16CXXX, PIC18FXXXX

PIC16CXXX

1. มีคำสั่งในภาษา assembly 35 คำสั่ง
2. มี I/O, มี Timer มากกว่า 1 ตัว, Watch dog, I2C, USART, SPI, PWM
3. เริ่มมี feature ใส่เข้าไปมากขึ้น ทำให้การ interface กับอุปกรณ์ภายนอกทำได้ง่ายขึ้น เห็นว่า feature มีเพียง ทั้ง I2C, USART, PWM ช่วยในการเขียนโปรแกรมได้มาก หลังจากนั้นก็ได้ผลิตตระกูล PIC12CXXX ออกมา ซึ่งเป็น pic ที่มีเพียง 8 ขา เหมาะกับงานเล็กๆ จุดเด่นคือมี OSC ขนาด 4 MHz ภายใน และมี DATA memory ส่วนหนึ่งเป็น EEPROM อยู่ภายในด้วย แต่ในส่วนของ PROGRAM memory ยังเป็น OTP และ EPROM หลังจากนั้นก็ได้ผลิต PIC12FXXX

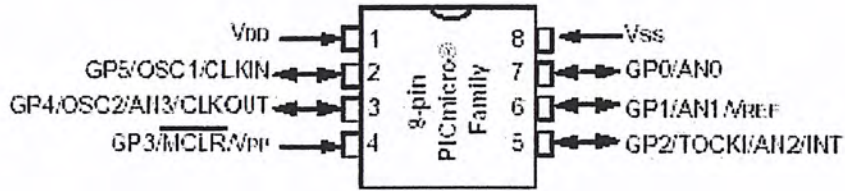
PIC12CXXX และ PIC12FXXX

1. มี 8 ขา
2. มีคำสั่งในภาษา assembly ทั้งหมด 33 หรือ 35 คำสั่ง
3. มี I/O, Timer, Watch dog
4. บางเบอร์มี internal EEPROM

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. 4MHz oscillator อยู่ภายในตัว chip ไม่ต้องมี oscillator ภายนอก (แต่หากต้องการความแม่นยำของเวลาก็จำเป็นต้องใช้ Crystal oscillator ภายนอก)

8-pin PICmicro[®] MCU Family



PIC12C509A	PIC12CE519	PIC12CE673
PIC12C509A	PIC12C671	PIC12CE674
PIC12CR509A	PIC12C672	PIC12F629
PIC12CE518		PIC12F675

รูปที่ 5-6 ชิป 8 ขาตระกูล PIC12CXXX, PIC12FXXX

PIC16CXXX และ PIC17CXXX จะออกมาพร้อมๆ กัน แต่ต่างกันที่ PIC17CXXX ถือว่าเป็นเบอร์ที่ใหญ่กว่ามีคำสั่งใน assembly มากกว่า รวมทั้งขนาดของ PROGRAM memory จะมีขนาดมากกว่าในตระกูล PIC16CXXX และสามารถต่อ memory ภายนอกได้

PIC17CXXX

1. มีคำสั่งในภาษา assembly 58 คำสั่ง มีคำสั่งการ คูณ หหาร
2. สามารถต่อกับ program memory ภายนอก โดยอ้างได้ถึง 64K x 16 Program memory
3. มี I/O, มี Timer มากกว่า 1 ตัว, Watch dog, I2C, USART, SPI, PWM

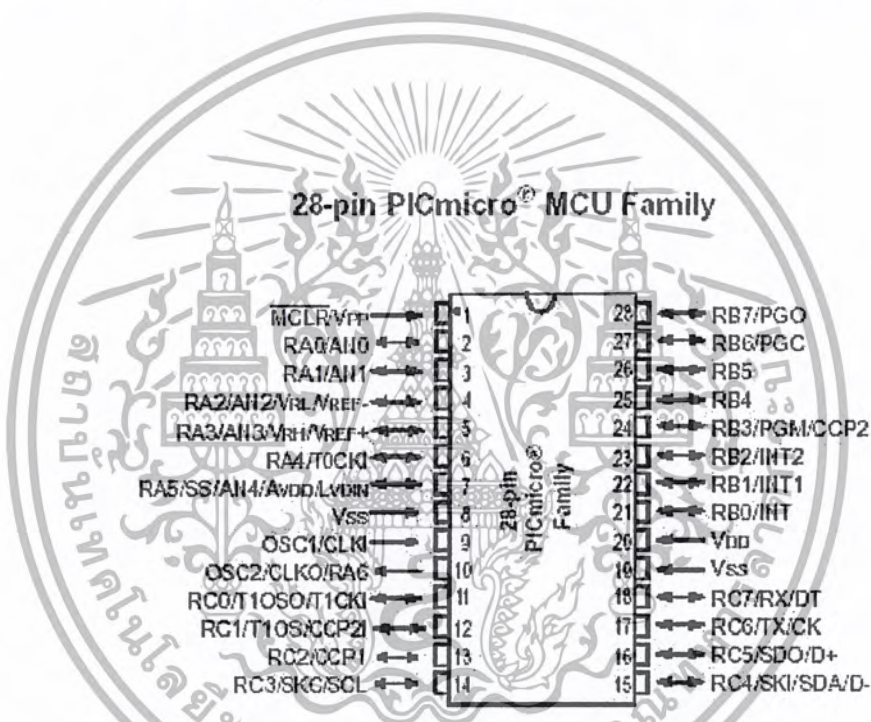
PIC16F8XXX เป็น PIC ที่ได้รับความนิยมมาก เพราะเป็นชิพยุคแรกๆ ที่มี PROGRAM MEMORY เป็นแบบ FLASH และมี DATA MEMORY ที่เป็น EEPROM เพิ่มเติมเข้ามารวมทั้งตัว PROGRAM CHIP สามารถสร้างได้ง่าย ทำให้กลุ่ม HOBBY และกลุ่มนักศึกษา นิยมใช้เพราะรวมค่าใช้จ่ายแล้วไม่แพงชิพที่ผลิตออกมาก่อนได้แก่ PIC16F83, PIC16F84 หลังจากนั้นได้ออกตระกูล PIC16F87X ซึ่งถือว่าเปลี่ยนแปลงครั้งใหญ่ เพราะมี In Circuit Debugger อยู่ภายในทำให้ ไม่จำเป็นต้องพึ่ง Emulator ที่มีราคาแพงอีกต่อไป ซึ่งเป็นผลให้ CHIP ในตระกูลนี้ ได้รับความนิยมเป็นอย่างมาก

PIC16F8XXX (FLASH MCUs)

1. มีคำสั่งในภาษา assembly 35 คำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. มี I/O, มี Timer มากกว่า 1 ตัว, Watch dog, I2C, USART, SPI, PWM
3. มี A/D ขนาด 10 bits
4. มี Program memroy เป็นแบบ Flash ทำให้สามารถโปรแกรมใหม่ได้หลายครั้ง
5. มี EEPROM ภายใน
6. ในตระกูล 16F87X สนับสนุน In Circuit Debugger (ICD) เป็นผลทำให้ไม่จำเป็นต้องซื้อ Emulatราคาแพงจุดด้อยของ PIC ก็คือ program memory มีลักษณะเป็น page ทาง Microchip ก็เห็นจุดด้อยตรงนี้จึงได้พัฒนาออกมาเป็น PIC18CXXX ซึ่ง PROGRAM MEMORY ไม่ถูกแบ่งเป็น page อีกต่อไป และเพิ่มคำสั่ง ASSEMBLY เป็น 77 คำสั่ง รวมทั้งออกแบบให้ PROGRAM MEMORY มีขนาดใหญ่ขึ้นเพื่อรองรับ การเขียนโปรแกรมภาษา C หลังจากนั้นจึงได้ออก PIC18FXXX ตามมา พร้อมกับ ICD2



PIC16C62B	PIC16F76	PIC18C242
PIC16CR63	PIC16C642	PIC18C252
PIC16C63A	PIC16C745	PIC18F2220
PIC16C66	PIC16C773	PIC18F2320
PIC16CR72	PIC16F870	PIC18F2331
PIC16C72A	PIC16F872	PIC18F2431
PIC16F72	PIC16F873	PIC18F242
PIC16C73B	PIC16F873A	PIC18F252
PIC16C76	PIC16F876	PIC18F248
PIC16F73	PIC16F876A	PIC18F258

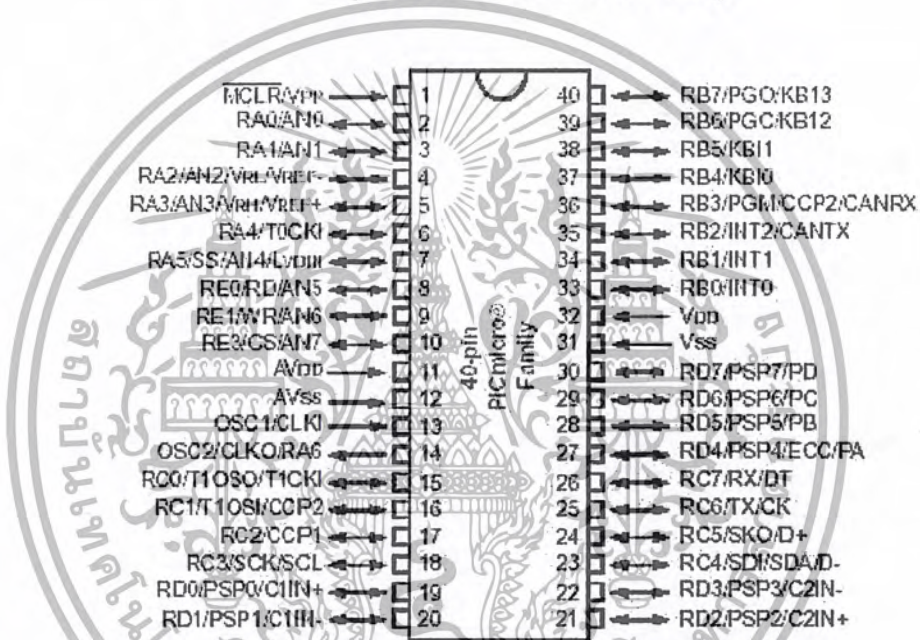
รูปที่ 5-7 ชิป 28 ขาตระกูล PIC16CXX, PIC16FXXX, PIC18CXXX, PIC18FXXX

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIC18CXXX, PIC18FXXX (FLASH MCUs)

1. มีคำสั่งในภาษา assembly 77 คำสั่ง
2. มี I/O, มี Timer มากกว่า 1 ตัว, Watch dog, I2C, USART, SPI, PWM, CAN
3. มี A/D ขนาด 10 bits
4. สามารถต่อกับ Program memory ภายนอก โดยอ้างได้ถึง 64K x 16 Program memory
5. มี Program memroy เป็นแบบ Flash ทำให้สามารถโปรแกรมใหม่ได้หลายครั้ง
6. ในตระกูล 18FXXX มี EEPROM ภายใน
7. ในตระกูล 18FXXX สนับสนุน In Circuit Debugging (ICD)

40-pin PICmicro[®] MCU Family



PIC16C65	PIC16C765	PIC18F4220
PIC16C65B	PIC16C774	PIC18F4320
PIC16C67	PIC16F871	PIC18F4331
PIC16C662	PIC16F874	PIC18F4431
PIC16C74B	PIC16F874A	PIC18F442
PIC16C77	PIC16F877	PIC18F452
PIC16F74	PIC16F877A	PIC18F448
PIC16F77	PIC18C442	PIC18F458
	PIC18C452	

รูปที่ 5-8 ชิป 40 ขา ตระกูล PIC16CXXX, PIC16FXXX, PIC18CXXX, PIC18FXXX

5.6 สรุป concept สถาปัตยกรรมของ PIC

PIC จะยึดถือการออกแบบโดยรวมทุกอย่างไว้ในชิปตัวเดียวซึ่งไม่ต้องต่ออุปกรณ์ใดๆ เพิ่มเติม

สิ่งที่ได้คือแผ่นวงจรจะมีขนาดเล็ก และอุปกรณ์ที่ใช้จะไม่มาก บางงานอาจจะใช้ PIC เพียงตัวเดียวโดยไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ต้องใช้ chip อื่นมาเพิ่มเติม นี่คือคุณสมบัติพิเศษของ PIC ซึ่งปัจจุบันหลายบริษัทที่ผลิต microcontroller ก็เริ่มจะหัดมาเขียนแบบแนวทางนี้ แต่ทุกอย่างย่อมมีข้อเสีย เนื่องจาก จุดประสงค์ที่จะรวมทุกอย่างไว้ในชิพเดียว ทำให้ program memory และ data memory ไม่สามารถขยายโดยใช้กับ memory ภายนอกได้ (ในทางทฤษฎี ซึ่งงานจริงสามารถทำได้ แต่ไม่เป็นที่นิยม) PIC จึงเหมาะสำหรับงานเล็กๆ ไม่ใช่งานใหญ่ๆ ที่ต้องใช้การคำนวณ และ memory มากๆ แนวโน้มในอนาคต MICROCONTROLLER เริ่มจะมี feature ต่างๆ รวมเข้าด้วยกันมากขึ้น สำหรับ In Circuit Debugger ต่อไปจะเป็นมาตรฐานของ MCU ทั่วๆ ไป ต้องมีการรวม FPGA หรือ DSP เพื่อให้สมบูณ์มากขึ้น

5.7 โปรแกรม MPLAB C18

โปรแกรม MPLAB C18 เป็นซีคอมไพเลอร์ที่ออกแบบมาสำหรับการเขียนภาษาซี กับไมโครคอนโทรลเลอร์ ตระกูล PIC18XXXX ของบริษัท Microchip ซึ่งสามารถดาวน์โหลดมาทดสอบใช้ได้ที่ WWW.Microchip.com โปรแกรมรุ่นนี้สามารถทำงานบน DOS และทำงานบน WINDOW เมื่อใช้ร่วมกับโปรแกรม MPLAB IDE โปรแกรมนี้มีข้อดีหลายประการ เช่น ใ้รหัสภาษาเครื่องของชิพตระกูล PIC ขนาดเล็ก สามารถใช้กับชิพตระกูล PIC18XXXX ทุกเบอร์ มีฟังก์ชันเกี่ยวกับการอินเตอร์รัปต์ ฟังก์ชันเกี่ยวกับสตริง รวมทั้งฟังก์ชัน ทางคณิตศาสตร์ต่างๆ ให้ใช้ได้ โปรแกรม MPLAB C18 นี้มีรายละเอียดต่างๆ มากมาย ผู้สนใจสามารถศึกษาได้จากคู่มือโดยตรง สำหรับปริญญาโทฉบับนี้ จะเน้นการเขียนโปรแกรมควบคุมและการออกแบบวงจรเป็นหลัก

5.8 การเขียนโปรแกรมด้วยภาษาซี

การเรียนรู้ทางด้านไมโครคอนโทรลเลอร์กับภาษาแอสเซมบลี(Assembly) ซึ่งเป็นระดับพื้นฐานที่สุดของงานด้านนี้ และเนื่องจากความคุ้นเคยที่มีมานานประกอบกับตัวแปลภาษาแอสเซมบลีสามารถหาได้ง่ายและมีราคาถูก แต่อย่างไรก็ตาม ปัจจุบันเราพบว่าผู้ใช้ภาษาซี ในวงการนี้มากขึ้น ถึงแม้ว่าราคา C Compiler อาจจะสูงสักหน่อย แต่ก็นับว่าคุ้มค่าเมื่อเทียบกับผลงานที่ได้ ภาษาซีมีความเหมาะสมกับงานทางด้านไมโครมาก เป็นภาษาที่ใกล้เคียงกับ ภาษาแอสเซมบลีในด้านความเร็ว การพัฒนาจะต้องใช้เครื่องคอมพิวเตอร์ PC เป็นเครื่องมือ จากนั้นจึงใช้ Hex File ที่แปลได้นำมาลงบนบอร์ดไมโครเพื่อทดสอบโปรแกรมอีกครั้ง เหตุผลหลักๆ ที่นักพัฒนาโปรแกรมเริ่มหันมาใช้ภาษาซีในการทำงานมากขึ้น

1.เป็นภาษาที่มีความเร็วในการทำงานสูง หมายถึง ความเร็วของโปรแกรมที่ทำงาน ซึ่งถือว่าใกล้เคียงกับภาษาแอสเซมบลีมากที่สุด งานทางด้านไมโครจะถือว่าความเร็วในการทำงานมีความสำคัญมาก เพราะงานหลายอย่างจะต้องดูแลอุปกรณ์ Input / Output หลาย ๆ ตัวพร้อมกัน ถ้าโปรแกรมทำงานช้าก็อาจจะเกิดความผิดพลาดได้ ภาษาซีจะทำงานได้เร็วมากในบางคำสั่งของภาษาซีอาจจะแปลออกมาเทียบได้กับภาษาแอสเซมบลีเพียงไม่กี่คำสั่งเท่านั้นเอง และปัจจุบันชิพไมโครคอนโทรลเลอร์รุ่นหลัง ๆ มักจะถูกออกแบบมาให้ใช้กับ Xtalk ที่มีความถี่สูงขึ้น อาจจะเป็น 20-30 MHz ซึ่งถือว่าพอเพียงต่อทำงาน ของภาษาซีได้ และในบางกรณีอาจจะมีการปรับโครงสร้างภายในของตัวชิพ เพื่อให้ทำงานได้เร็วขึ้นอีก เพราะฉะนั้น ถ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ความเร็วอาจจะยังไม่พอ เราก็สามารถเลือกที่จะปรับ Hardware แทนได้โดยไม่ต้องกลับไปใช้ภาษาแอสเซมบลีอีก

2. เป็นภาษาที่ยังคงเชื่อมต่อกับแอสเซมบลีได้อย่างกลมกลืน ปกติตัว C Compiler จะแปลภาษาซีให้เป็นภาษาแอสเซมบลีอีกที เพราะฉะนั้นเรายังคงสามารถปรับปรุง ในส่วนของภาษาแอสเซมบลีได้ นอกจากนี้ในบางกรณี จะสามารถเขียนภาษาแอสเซมบลีผสมผสานไปกับภาษาซีได้ทันทีด้วย (เรียกคุณสมบัตินี้ว่า In-line Assembly) ซึ่งนับว่ามีความยืดหยุ่นสูง ในงานที่จำเป็นต้องมีความเร็วมากๆ หรืองานที่เกี่ยวข้องกับการสร้างสัญญาณ Clock ต่าง ๆ เรายังคงเลือกที่จะใช้แอสเซมบลีในการพัฒนาได้

3. เป็นภาษาโครงสร้างภาษาซีเป็นภาษาที่มีลักษณะ โครงสร้างอยู่แล้ว การเขียน โปรแกรมจะต้องมีส่วนฟังก์ชันที่ชัดเจนและมีลำดับของการทำงานอย่างเป็นระเบียบ จะเขียนให้กระโดดไปกระโดดมาตามใจชอบไม่ได้ ซึ่งเป็นข้อดีที่บังคับให้ผู้เขียน โปรแกรม ต้องออกแบบอย่างเป็นระเบียบทำให้วิเคราะห์โปรแกรมได้ง่าย รวมไปถึงการปรับปรุงแก้ไขก็ทำได้ง่ายด้วยจากประสบการณ์เราพบว่า การเขียนโปรแกรมด้วยภาษา ซีจะมีลักษณะเป็นความเร็วตอนปลาย (ซึ่งตรงข้ามกับภาษาแอสเซมบลี) กล่าวคือ เมื่อเริ่มต้นเขียนจะทำให้ช้ามากคือต้องค่อย ๆ สร้างฟังก์ชันต่าง ๆ ทีละอันจนครบตามเนื้อหางานแต่เมื่อถึงตอนท้ายจะทำให้เร็วมาก และที่สำคัญก็คือเมื่อระยะเวลาผ่านไป แล้ว และต้องนำโปรแกรมกลับมาปรับปรุงใหม่ ก็สามารถทำได้ง่ายมาก ซึ่งหาความแน่นอนในภาษาแอสเซมบลีไม่ได้

4. เป็นภาษาที่มี Source Code สั้นลงมากถ้างานใดที่พัฒนาเป็นภาษาแอสเซมบลีมีความยาวของโปรแกรมประมาณ 10 หน้ากระดาษการพัฒนาเป็นภาษาซีอาจจะทำให้เหลือความยาวเพียง 3-4 หน้ากระดาษเท่านั้น ซึ่งเป็นอีกเหตุผลหนึ่งที่ทำให้ปรับปรุงแก้ไข ได้ง่ายขึ้นจากนี้ ถ้ามีฟังก์ชันที่ซ้ำบ่อย ๆ ทำเป็น Library ไว้ก็จะยิ่งทำให้ Source Code เหลือน้อยลงอีกยิ่ง Source code สั้นเท่าไรก็จะยิ่งทำให้ดูแลได้ง่ายขึ้นเท่านั้น

5. เป็นภาษาที่มีความเป็นมาตรฐานสูงภาษาซีมีความเป็นมาตรฐานสูงมากถึงแม้ว่ารายละเอียดต่างๆ จะแตกต่างกันบ้าง ตามแต่ Compiler ที่มีใช้อยู่แต่เนื้อหาทั่ว ๆ ไปจะเหมือนกันหมดผู้ที่เคยเขียนภาษาซีบนเครื่องคอมพิวเตอร์ก็สามารถนำความรู้มาใช้ได้ทันที และตำราที่เกี่ยวกับภาษาซีก็มีให้เลือกมากมาย กล่าวได้ว่า ความรู้ในภาษาซีจะไม่หยุดอยู่ที่ใดที่หนึ่งแต่จะประยุกต์ ไปใช้ได้ ในหลายวงการ อย่างเช่น โปรแกรม Linux ที่เป็น Operating System ที่กำลังมาแรงมากก็ถูกเขียนด้วย ภาษาซี เช่นเดียวกัน

6. เป็นภาษาที่เหมาะสมกับทรัพยากรในระดับคอนโทรลเลอร์ ภาษาซีเป็นภาษาที่มีการจัดทรัพยากรต่าง ๆ อย่างคุ้มค่า เช่นการใช้ตัวแปรต่าง ๆ ใน โปรแกรม หรือการจัดพื้นที่ เพื่อทำเป็น Stack ก็ทำได้เหมาะสมพอดี จึงเป็นเรื่องปกติ ที่ใช้หน่วยความจำ RAM เพียง 128 Byte กับส่วนของโปรแกรมอีก 8 Kbyte ก็สามารถใช้งาน ได้มากมาย ภาษาซีจะสร้าง Code ตามเนื้อหาของงานที่ทำ ถ้างานเล็ก Code ก็เล็กไปด้วย ซึ่งอาจจะใช้กับชิพขนาดเล็กได้ แต่ถ้างานใหญ่ Code ก็จะไปใหญ่ตามไปด้วย จุดนี้จะแตกต่างกับภาษา Basic ในงานไมโครคอนโทรลเลอร์ ซึ่งไม่ว่างานจะเล็กหรือใหญ่ ก็จำเป็นต้องมีทรัพยากรพื้นฐานในระดับหนึ่งเสมอ ทำให้ไม่สามารถสร้างงานขนาดเล็ก ๆ ได้

7. เป็นภาษาที่เป็นอิสระจากตระกูลชิพ ภาษาซีเป็นแนวโน้มที่ชัดเจนของการพัฒนางานทางด้านไมโครคอนโทรลเลอร์ จะเห็นได้ว่าชิพจากผู้ผลิตทุกราย สามารถเลือกใช้ภาษาซีในการพัฒนาได้ นอกจากนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นี้ ตัวชิพเองยังถูกออกแบบให้มีโครงสร้างสอดคล้องกับลักษณะของภาษาซีมากขึ้นด้วยและนั่นก็หมายถึงว่าถ้าอนาคตเราจะต้องใช้ชิพที่มีความเร็วสูงขึ้นหรือจำเป็นต้องเปลี่ยน ตระกูลเนื่องด้วยเหตุผลต่าง ๆ เช่น MCS51, PIC, DSP ความรู้ในด้านภาษาซีจะยังคงนำไปใช้ได้อย่างต่อเนื่อง หรือแม้กระทั่ง Source Code ที่มีอยู่ก็สามารถนำไปปรับใช้ได้อย่างไม่ยากนัก

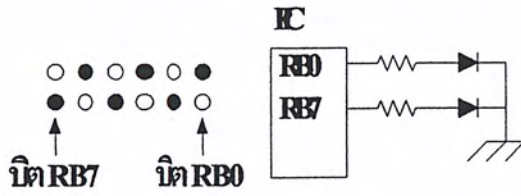
5.9 โครงสร้างของภาษาซี

ด้วยภาษาซีเป็นภาษาที่สามารถเขียน โปรแกรมเป็นแบบโครงสร้างได้ โดยโปรแกรมจะแบ่งการทำงานต่าง ๆ ออกเป็นกลุ่ม ๆ หรือฟังก์ชัน โดยฟังก์ชันเหล่านั้นสามารถเรียกขึ้นมาใช้ใหม่ได้ ในการเขียนโปรแกรมจะต้องระบุไว้ว่าในโปรแกรมนั้นมีฟังก์ชันใดให้ใช้บ้าง แต่ทุกโปรแกรมจะต้องมีฟังก์ชันหลักที่ชื่อว่า main () เสมอ พิจารณาตัวอย่าง โครงสร้างของ โปรแกรมต่อไปนี้

```
#include < P18f458.h>          /* Preprocessor */
void func 1 ( void );          /* Pro to type */
int func 2 ( int x );
void mein ( )                  /* ฟังก์ชันหลัก */
{
    int a;                      /* ประกาศตัวแปร */
    TRISB = 0;                  /* กำหนดให้พอร์ต B เป็นอินพุต */
    PORTB = 0XFF;
    func 1 ( )                  /* เรียกใช้ฟังก์ชัน */
    a = func 2 ( 4 );           /* เรียกใช้ฟังก์ชันที่มีการส่งค่า */
    PORTB = a;
}
void func 1 ( void )           /* ฟังก์ชันที่ไม่มีการคืนค่า */
{
    .....
    .....
}
int func 2 ( int x )           /* ฟังก์ชันที่มีคืนค่า */
{
    return ( x * 2 );
}
```

จากโปรแกรมจะพบว่า ส่วนประกาศโปรโตไทป์จะบอกว่าโปรแกรมนั้นมีฟังก์ชันชื่อ func 1 ให้ใช้งานและฟังก์ชันนี้จะทำงานเป็น โปรแกรมย่อยเพราะมี void นำหน้า และมีฟังก์ชันชื่อ func 2 (int x) ซึ่งจะรับค่าเข้าไปผ่านทางตัวแปร x และคืนค่าออกมาเป็นจำนวนเต็ม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



วิธีทำ การทำให้หลอด LED สว่างควมเว้นควมโดยส่งค่า 55H ออกไปทางพอร์ต์ B จากนั้นหน่วงเวลา และส่งค่า 0AAH ออกไปทางพอร์ต์ B และหน่วงเวลาจากนั้นทำโปรแกรมซ้ำจะเขียนโปรแกรมได้ดังนี้

```
#include <P18f458.h>
void delay (int count);
void main()
{
    TRISB = 0;
    while (1)
    {
        PORTB = 0x55;
        delay (100);
        PORTB = 0x0AA;
        delay (100);
    }
}
void delay (int count)
{
    int i,j;
    for (i = 0; i < count; i++)
        for (j = 0; j < 500; j++);
}
```

จากโปรแกรมจะเห็นว่ามีการสร้างฟังก์ชันชื่อ delay ขึ้นมาใช้งาน ทำให้โปรแกรมหลักสามารถเรียกขึ้นมาทำงานได้โดยละเอียดของโปรแกรมเป็นดังนี้

หมายเลข 1 จะเป็นการประกาศโปรโตไทป์ว่ามีฟังก์ชันชื่อ delay อยู่ เพื่อให้โปรแกรมหลัก main สามารถเรียกฟังก์ชันขึ้นมาใช้ได้ ถ้าหากไม่ประกาศเมื่อมีการคอมไพล์โปรแกรมจะมี warning

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ error เกิดขึ้นซึ่งจะบอกว่าโปรแกรมหลักจะไม่รู้จักฟังก์ชันชื่อ delay ในส่วนของโปรแกรมหลัก หมายเลข 2 จะเป็นฟังก์ชัน while ของภาษาซี ซึ่งจะทำให้โปรแกรมทำฟังก์ชันที่อยู่ในวงเล็บปีกกาซ้ำ ซึ่งจะทำให้หลอด LED ที่อยู่ที่พอร์ต B เกิดการติดดับสลับกัน สำหรับโปรแกรมน้อยหมายเลข 3 จะมีการส่งผ่านค่าเข้าไปทางตัวแปร count ตัวแปรนี้บางครั้งเรียกว่าฟอร์มอลพารามิเตอร์ (formal parameters) ถ้าหากค่านี้มีค่าน้อยโปรแกรมจะหน่วงเวลาน้อย ถ้าหากมีค่ามากโปรแกรมจะหน่วงเวลา มาก ในฟังก์ชัน delay จะมีการประกาศตัวแปรขึ้นมาสองตัวคือ i และ j (หมายเลข 4) ตัวแปรนี้เรียกว่า ตัวแปรดลเคลิ เพราะว่าจะประกาศขึ้นมาเพื่อใช้ภายในฟังก์ชันนี้เท่านั้น

ในการเขียน โปรแกรมภาษาซีในลักษณะมอดูล ถ้าหากนำฟังก์ชันต่างๆ ขึ้นมาเขียนก่อน ฟังก์ชัน (main) สามารถมองเห็นฟังก์ชันที่อยู่ก่อนหน้านั้นได้

```
#include <P18f458.h>
void delay (int count);
{
    int i,j;
    for (i = 0; i < count; i++)
    for (j = 0; j < 500; j++);
}
void main ()
{
    TIRSB = 0;
    while (1)
    {
        PORTB = 0x55;
        delay ( 100 );
        PORTB = 0x0AA;
        delay ( 100 );
    }
}
```

5.11 ตัวดำเนินการในภาษาซี

ตัวดำเนินการจะเป็นตัวที่ใช้กระทำกับตัวแปร ค่าคงที่ต่างๆ ให้รวมเป็นค่าเดียวกันโดยอาจกระทำทางคณิตศาสตร์หรือกระทำทางลอจิกได้ ในการเขียนโปรแกรมด้วยภาษาซีนั้น ตัวดำเนินการจะแบ่งออกเป็นสองกลุ่มใหญ่ๆ คือตัวดำเนินการที่กระทำกับตัวกระทำตัวเดียว (single operand operators) และ ตัวดำเนินการที่กระทำกับตัวถูกกระทำสองตัว (two operands operators)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.11.1 ตัวดำเนินการที่กระทำกับตัวถูกกระทำตัวเดียว

ดังนี้

ตัวดำเนินการประเภทนี้จะกระทำกับตัวถูกกระทำเพียงตัวเดียว ประกอบด้วยตัวดำเนินการต่างๆ

- ลบ (negate)
- ~ กลับค่าลอจิกของบิตข้อมูล (bit wise complement)
- ! กลับค่าทางลอจิก (logical complement)
- ++ เพิ่มค่าขึ้นหนึ่งค่า (increment)
- ลดค่าลงหนึ่งค่า (decrement)
- * ตัวดำเนินการทางพอยท์เตอร์
- & ตำแหน่งหน่วยความจำของตัวแปร

5.11.2 ตัวดำเนินการที่กระทำกับตัวถูกกระทำสองตัว

ตัวดำเนินการประเภทนี้จะกระทำกับตัวถูกกระทำสองตัว ถ้าหากมีตัวถูกกระทำหลายๆ ตัวสามารถนำมาเขียนรวมกันเป็นประโยคได้ซึ่งประกอบด้วยตัวดำเนินการที่ใช้กำหนดค่า ตัวดำเนินการทดสอบค่าซึ่งจะให้ผลลัพธ์เป็นค่าลอจิก (จริง,เท็จ) ตัวดำเนินการทางคณิตศาสตร์ และตัวดำเนินการทางลอจิกดังต่อไปนี้

- = กำหนดค่าในประโยค (assignment)
- + บวก
- ลบ
- * คูณ
- / หาร (division)
- % หารแบบบวก (module)
- && การแอนด์ (logical AND)
- !! การออร์ (logical OR)
- & การแอนด์แบบบิตต่อบิต (bit wise AND)
- ! การออบแบบบิตต่อบิต (bit wise OR)
- ^ การเอ็กคลูซีฟออร์แบบบิตต่อบิต (bit wise exclusive OR)
- << เลื่อนบิตไปทางซ้าย
- >> เลื่อนบิตไปทางขวา
- == ทดสอบว่าเท่ากันหรือไม่
- != ทดสอบว่าไม่เท่ากันหรือไม่
- > ทดสอบว่ามากกว่าหรือไม่
- < ทดสอบว่าน้อยกว่าหรือไม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

> = ทดสอบว่ามากกว่าหรือเท่ากันหรือไม่

< = ทดสอบว่าน้อยกว่าหรือเท่ากันหรือไม่

นอกจากนี้ตัวดำเนินการบางประเภทสามารถนำมารวมกันเป็น compound operators ได้ ตัวอย่างเช่นถ้ามีประโยค $y = y * 2$; อาจเขียนตัวดำเนินการรวมได้เป็น $y * = 2$; รูปแบบของตัวดำเนินการที่รวมกันได้เป็นดังนี้

+ = บวกและให้เท่ากับ

- = ลบและให้เท่ากับ

* = คูณและให้เท่ากับ

/ = หารและให้เท่ากับ

% = หารแบบบวกและให้เท่ากับ

& = ทำการแอนด์และให้เท่ากับ

!= ทำการออร์และให้เท่ากับ

^ = ทำการเอ็กสพอร์และให้เท่ากับ

<< = เลื่อนบิตไปทางซ้ายและให้เท่ากับ

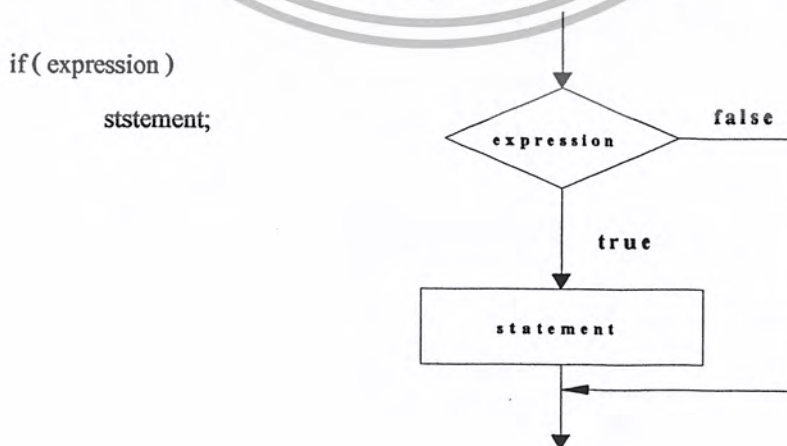
>> = เลื่อนบิตไปทางขวาและให้เท่ากับ

5.12 ประโยคควบคุมในภาษาซี

การทำงานของโปรแกรมนั้นจะทำคำสั่งแต่ละคำสั่งเรียงกันไป และเราสามารถให้โปรแกรมตัดสินใจในการเลือกทำได้ หรือให้ทำงานใดงานหนึ่งซ้ำ ๆ ตามเงื่อนไขที่กำหนดได้โดยใช้คำสั่งควบคุมในภาษานั้นจะมีประโยคควบคุมคำสั่งที่ใช้ในการเลือกทำและทำซ้ำๆ ดังนี้

5.12.1 ประโยค IF / ELSE

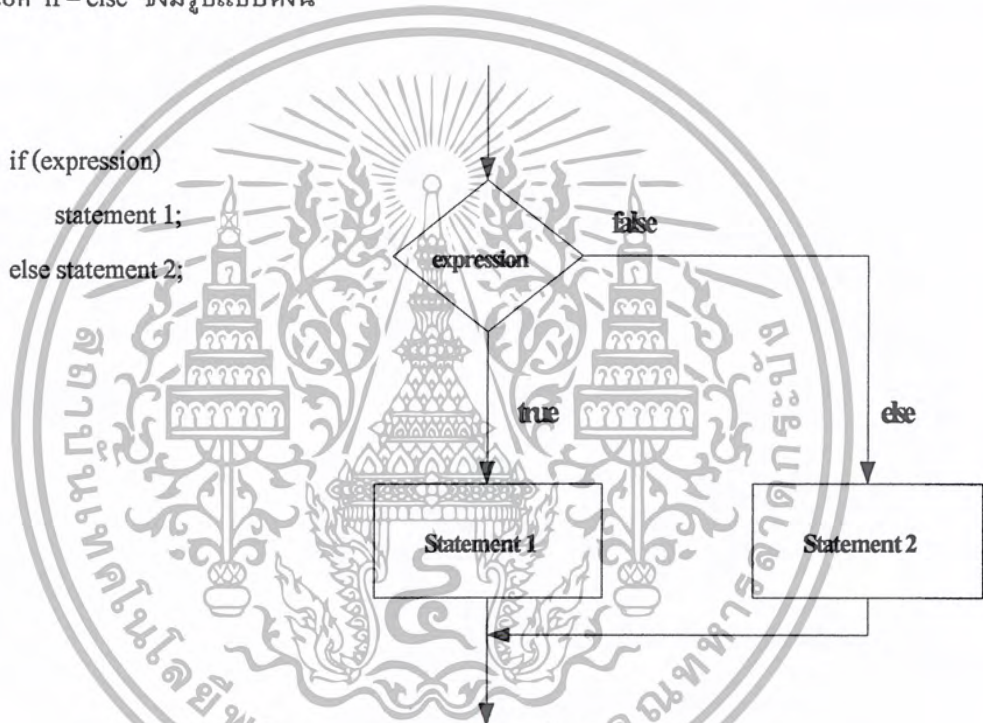
ประโยคคำสั่งนี้ใช้ควบคุมทิศทางการทำงานของโปรแกรมโดยจะถูกแปรออกมาเป็นคำสั่งในภาษาแอสเซมบลีดังนี้ GOTO, INCFSZ, RETLW, RETFIE โดยมีรูปแบบประโยคคำสั่งดังนี้



ประโยคนี้ใช้ในการทดสอบว่าจะทำสแตตเมนต์ที่ตามมาหรือไม่ ถ้าค่าใน expression เป็นจริงหรือมีค่าไม่เป็นศูนย์จะทำสแตตเมนต์ที่ตามมา ถ้าเป็นเท็จหรือมีค่าเท่ากับศูนย์จะไม่ทำ และสแตตเมนต์ที่จะทำงานนั้น อาจจะเป็นประโยคคำสั่งประโยคเดียว หรือเป็นสแตตเมนต์ซ้อนก็ได้ (ต้องมีปีกกาคลุม) ตัวอย่างเช่น ถ้าค่าของพอร์ต PORTB มีค่าไม่เป็น 0 ให้ตัวแปร c เป็นศูนย์ จะเขียนได้ดังนี้

```
if(PORTB != 0)
    c = 0;
```

ถ้าหากเป็นการทำงานเลือกทำแบบมีสองทางเลือก และต้องการทำงานเพียงอย่างใดอย่างหนึ่งจะใช้ประโยค if-else ซึ่งมีรูปแบบดังนี้



ตัวอย่างเช่น ถ้าจะทดสอบว่าค่าของพอร์ต PORTB ถ้ามีค่าไม่เป็นศูนย์ให้ตัวแปร c มีค่าเป็น 20 ถ้ามีค่าเป็นศูนย์ให้ตัวแปร c มีค่าเป็นศูนย์ จะเขียนได้ดังนี้

```
if(PORTB != 0)    c = 20;
else              c = 0;
```

5.12.2 ประโยค switch

การเลือกที่มีทางเลือกหลายๆ ทางเลือกนั้นเราสามารถนำประโยค if-else มาซ้อนกันก็ได้ แต่จะทำให้มองดูเข้าใจยาก ในภาษาซีจึงมีประโยค switch ที่ใช้ในการเลือกทำอย่างใดอย่างหนึ่งจากหลายๆ ทางเลือก โดยมีรูปแบบของประโยคดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

switch (k)
{
    case 1 :      statement 1;
                break;
    case 2 :      statement 2;
                break;
    case 2 :      statement 3;
                break;
                :      :
                :      :
    default :     statement n;
}

```

การทำงานของโปรแกรมจะนำค่าในตัวแปร k ที่อยู่ในวงเล็บหลัง switch มาเปรียบเทียบกับค่าคงที่หลังคำสั่ง case ตัวใด และจะทำงานสแตคเมนต์ที่ตามหลัง case นั้น และจะออกนอกปีกกาของ switch เมื่อพบคำสั่ง break โดยสแตคเมนต์ที่ทำงานนั้นจะเป็นสแตคเมนต์ซ้อนกันก็ได้ แต่ถ้าค่าในตัวแปร k ไม่เท่ากับค่าคงที่ค่าใดเลย โปรแกรมจะทำสแตคเมนต์ที่ตามหลัง default

คำสั่งให้ โปรแกรมทำงานซ้ำถือว่าเป็นประโยคคำสั่งควบคุมอย่างหนึ่ง การทำซ้ำหรือที่เรียกว่า การทำลูปนั้นจะมีประโยคคำสั่งอยู่สามประเภทคือ for, while, do – while ซึ่งแต่ละแบบจะต่างกันตรงเงื่อนไขของการทำซ้ำ

5.12.3 ประโยค for

ประโยคคำสั่งนี้จะใช้ในกรณีที่มีจำนวนรอบของการทำซ้ำที่แน่นอน โดยมีรูปแบบดังนี้

```
for (initialization ; condition ; increment)
```

```
statement ;
```

โดยที่ initialization เป็นค่ากำหนดเริ่มต้นให้กับตัวแปรของการทำลูป condition เป็นเงื่อนไขที่ใช้ทดสอบการทำซ้ำครั้งต่อไป ซึ่งจะเป็นการกระทำลูปอีก increment เป็นการเพิ่มค่าให้ตัวแปรในการทำซ้ำในแต่ละครั้ง สำหรับ statement จะเป็นสแตคเมนต์ของคำสั่งที่จะทำซ้ำ ซึ่งอาจเป็นสแตคเมนต์รวมก็ได้ ตัวอย่างเช่น

```
unsigned char x;
```

```
TIRSB = 0;
```

```
For (x = 0; x <= 255; x++)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
PORTB = x;
```

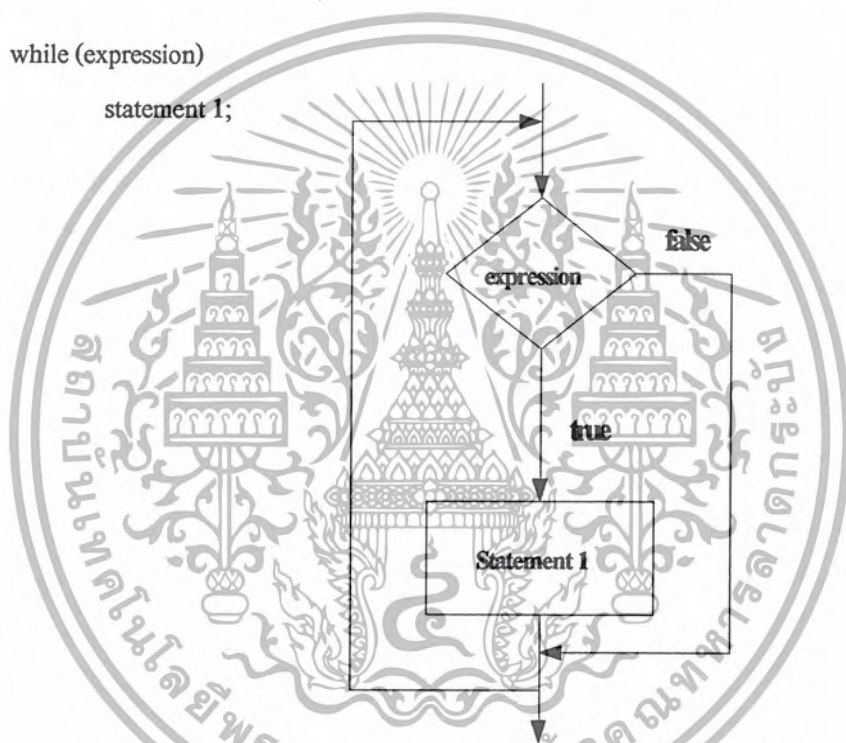
การทำงานตามชุดคำสั่งข้างบนจะเป็นการส่งค่า 0 ถึง 255 ออกมาทางพอร์ต B โดยเริ่มต้นจะประกาศตัวแปร x สำหรับนับการทำซ้ำ กำหนดพอร์ต B เป็นเอาต์พุตให้ x เท่ากับ 0 ต่อมาส่ง x ออกทาง B และตรวจสอบว่า x น้อยกว่าหรือเท่ากับ 255 จริงหรือไม่ ถ้าจริงให้เพิ่มค่า x ขึ้นอีกหนึ่ง (x++) และทำซ้ำไปเรื่อยๆ

5.12.4 ประโยค while

การทำซ้ำแบบนี้จะตรวจสอบเงื่อนไขก่อนการทำซ้ำ ถ้าเงื่อนไขเป็นจริงจะทำสแตทเมนต์ที่กำหนดและทดสอบเงื่อนไขใหม่ ถ้าเงื่อนไขเป็นเท็จจะออกจากการทำซ้ำทันที โดยที่รูปแบบดังนี้

```
while (expression)
```

```
statement 1;
```



ในส่วนของ expression นั้นสามารถตรวจสอบค่าคงที่ได้ด้วย ถ้าค่าไม่เท่ากับศูนย์จะซ้ำ ถ้าค่าไม่เท่ากับศูนย์จะไม่ซ้ำ ตัวอย่างเช่น ถ้าหากต้องการให้ค่าลอจิกทางพอร์ต B ของ PIC ทุกบิตมีค่าลอจิกกลับไปมาจะเขียนคำสั่งได้ดังนี้

```
TIRSB = 0;
```

```
While (1)
```

```
{
```

```
    PORTB = 0 x 55;
```

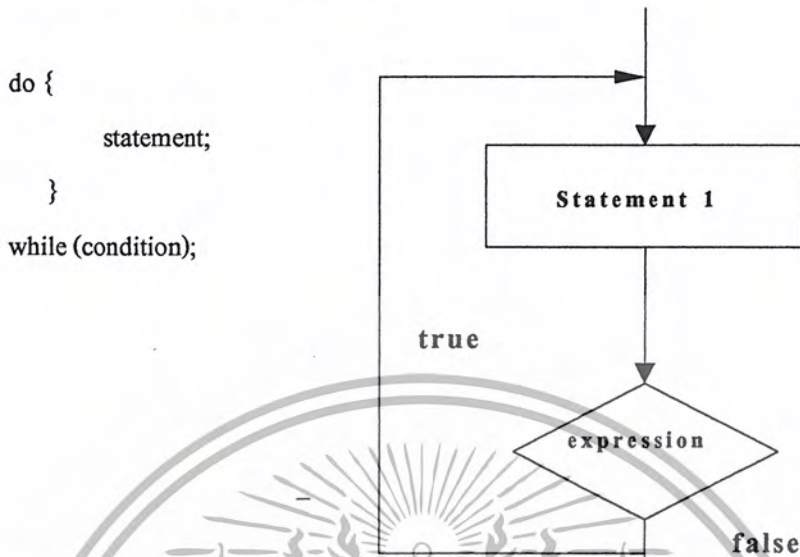
```
    PORTB = 0 x 0AA;
```

```
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.12.5 ประโยค do - while

การทำซ้ำประเภทนี้จะตรวจสอบเงื่อนไขภายหลังจากการทดสอบแต่ละครั้ง ถ้าหากเงื่อนไขเป็นเท็จจะออกจากการทำซ้ำทันทีโดยมีรูปแบบดังนี้



ตัวอย่าง จงเขียนฟังก์ชันหน่วงเวลาโดยใช้คำสั่งการทำซ้ำให้กับ PIC18F458

วิธีทำ การหน่วงเวลาจะทำได้โดยให้ไมโครคอนโทรลเลอร์เสียเวลาไปเปล่า ๆ แต่ต้องไม่ทำให้ระบบทำงานผิดพลาด ซึ่งอาจทำได้โดยเขียนโปรแกรมวนลูปซ้ำ ๆ ดังนี้

```
void delay (unsigned int x)
{
    unsigned char j;
    while (x-- )
    {
        for (j = 0; j < 125; j ++);
    }
}
```

จากจุดคำสั่งข้างบนจะสร้างฟังก์ชันที่ชื่อ delay ขึ้นมาและมีการประกาศตัวแปร j สำหรับใช้ถ่ายในฟังก์ชันและจะผ่านค่าเข้าไปในฟังก์ชันผ่านทางตัวแปร x ยังไม่เป็นที่แน่นอนว่าจะทดสอบเงื่อนไขที่ตามมาโดยวนลูป for จำนวน 125 ครั้ง และกลับไปลดค่าในประโยค while ใหม่จนกว่าค่า x เป็นศูนย์

5.13 อาร์เรย์ พอยน์เตอร์ และ สตริงเจอร์

ในการเขียนโปรแกรมด้วยภาษาซีถ้าหากต้องการใช้งานตัวแปรหลายตัวเราสามารถประกาศชื่อตัวแปรออกมาหลายตัวได้ เช่น $x_1, x_2, x_3, \dots, x_n$ แต่ถ้าหากตัวแปรทุกตัวใช้เก็บข้อมูลประเภทเดียวกัน เราสามารถประกาศเป็นตัวแปรอาร์เรย์ (array) ได้ การประกาศตัวแปรแบบอาร์เรย์สามารถทำได้ดังรูปแบบต่อไปนี้

ประเภทของข้อมูล ชื่ออาร์เรย์[ขนาดอาร์เรย์]

ถ้าเป็นอาร์เรย์แบบสองมิติสามารถประกาศได้ดังนี้

ประเภทของข้อมูล ชื่ออาร์เรย์[ขนาดอาร์เรย์][ขนาดอาร์เรย์]

ตัวอย่างเช่น `char x[8];`

เป็นการประกาศตัวแปรชื่อ `x` จำนวน 8 เซลล์ แต่ละเซลล์จะเก็บข้อมูลประเภทตัวอักษร การอ้างถึงอาร์เรย์ `x` แต่ละเซลล์จะใช้อินเด็กซ์เป็นตัวอ้าง เช่น `x[0]` เป็นการอ้างถึงเซลล์แรก นอกจากนี้การประกาศตัวแปรอาร์เรย์สามารถกำหนดค่าข้อมูลเข้าไปในตัวแปรอาร์เรย์ได้เลย ตัวอย่างเช่น

```
unsigned char ab[] = {0xa, 0x9, 0x5, 0x6};
```

จะเห็นว่า การประกาศตัวแปรอาร์เรย์ `ab` จะไม่ระบุขนาดของอาร์เรย์ ระบบจะจองหน่วยความจำเท่ากับค่าที่กำหนด การประกาศแบบนี้เซลล์แรก `ab[0]` จะเก็บค่า `0A` ตามสิบหกขนาดหนึ่งไปด้

ในระบบไมโครคอนโทรลเลอร์ เมื่อมีการประกาศตัวแปรอาร์เรย์ ระบบจะจองหน่วยความจำที่สำหรับเก็บตัวแปรอาร์เรย์นั้นตัวแปรประเภทนี้สามารถนำไปประยุกต์ใช้ในการเขียนโปรแกรมแบบเปิดตารางได้

ในการอ้างถึงข้อมูลแต่ละเซลล์ในอาร์เรย์สามารถใช้ตัวแปรพอยน์เตอร์ชี้ไปที่ตำแหน่งของอาร์เรย์ได้โดยตรง โดยพอยน์เตอร์จะเป็นตัวแปรที่ใช้เก็บแอดเดรสหรือตำแหน่งหน่วยความจำ ตัวดำเนินการที่ใช้กับพอยน์เตอร์คือ `&` และ `*` ตัวอย่างเช่น ถ้ามีการประกาศตัวแปรเป็น

```
char j;
```

จะเป็นการประกาศหน่วยความจำชื่อ `j` สำหรับเก็บตัวอักษร เราสามารถอ้างแอดเดรสของตัวแปร `j` ได้ดังนี้

```
&j;
```

ถ้าหากมีการประกาศตัวแปรพอยน์เตอร์สำหรับเก็บแอดเดรสของตัวแปร `j` จะทำได้ดังนี้

```
char * dptr;
```

```
dptr = &j;
```

เป็นการประกาศตัวแปรพอยน์เตอร์ที่ชื่อ dptr และให้ชี้ไปที่แอดเดรสของตัวแปร j ถ้าหากมีการประกาศตัวแปรเป็น

```
int ax[20];
```

หมายความว่าอาร์เรย์ที่ชื่อ ax มีจำนวน 20 เซล แต่ละเซลล์จะเก็บจำนวนเต็ม การอ้างถึงข้อมูลแต่ละเซลล์จะเขียนเป็น ax[i] โดยที่ i เป็นค่าอินเด็กซ์ และถ้าประกาศตัวแปร ip ให้เป็นตัวแปรพอยน์เตอร์ชี้ไปที่อาร์เรย์ของเลขจำนวนเต็มสามารถทำได้ดังตัวอย่างต่อไปนี้

```
int * ip;
```

```
ip = &ax[0];
```

จะทำให้ตัวแปร ip ชี้ไปที่ตำแหน่งของเซลล์ ax[0] ถ้าหากต้องการนำข้อมูลที่อยู่ในตัวแปร ax[0] มาใส่ในตัวแปร x สามารถทำได้ดังนี้

```
x = * ip;
```

และถ้า ip ชี้ไปที่ ax[0] ถ้าหากมีการอ้างเป็น ip + 1 จะเป็นการชี้ไปที่ ax[1] ดังนั้นถ้าหากมีการอ้างเป็น ip + 1 จะเป็นการอ้าง ไปที่อาร์เรย์ ax[1] ได้

ในระบบไมโครคอนโทรลเลอร์บ้างครั้งจะต้องมีการออกแบบหน่วยความจำ ROM และ RAM ต่ออยู่ภายนอกตัวไมโครคอนโทรลเลอร์ และอุปกรณ์ต่างๆ ที่ต่อกับไมโครคอนโทรลเลอร์จะมีตำแหน่งหรือแอดเดรสที่แน่นอน เราสามารถใช้พอยน์เตอร์ชี้ไปที่ตำแหน่งความจำภายนอกได้ ตัวอย่างเช่น

```
char * abs_ptr = 0x8000;
```

เป็นการประกาศตัวแปรพอยน์เตอร์ชื่อ abs_ptr ให้ชี้ไปที่ตำแหน่ง 8000H

จากที่ผ่านมาจะพบว่าตัวแปรประเภทอาร์เรย์เราอาจมองว่าเป็นกลุ่มข้อมูลได้ โดยข้อมูลในกลุ่มนั้นจะเป็นข้อมูลประเภทเดียวกัน ถ้าหากต้องการประกาศตัวแปรเป็นกลุ่มของข้อมูลที่ข้อมูลในกลุ่มนั้นเป็นชนิดต่างกัน จะต้องประกาศตัวแปรเป็นแบบโครงสร้าง หรือสตรักเจอร์ (structure) มีรูปแบบดังนี้

```
struct {
    ประเภทข้อมูล   ชื่อตัวแปร
    .....
    .....
} ชื่อ โครงสร้าง;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยกลุ่มของข้อมูลที่ประกาศขึ้นนั้นจะอยู่ในเครื่องหมายปีกกา และเราสามารถอ้างไปที่ข้อมูล
ตัวใดๆ ก็ได้ โดยใช้เครื่องหมายจุด (.) ตัวอย่างเช่น

```
struct {
    unsigned long s;
    unsigned int t;
    unsigned char done;
} state;
```

จะเป็นการประกาศตัวแปรโครงสร้างชื่อ state ซึ่งจะใช้หน่วยความจำทั้งหมด 7 ไบต์ (long 4
+ int 2 + char 1) ถ้าหากต้องการใส่ข้อมูลค่า 321 ให้กับตัวแปร t ในโครงสร้างจะทำได้ดังนี้

```
state.t = 321;
```

ถ้าหากมีการประกาศตัวแปรดังต่อไปนี้จะทำให้ผลลัพธ์มีค่าเหมือนกับตัวแปรแบบ โครงสร้างใน
ตัวอย่างที่ผ่านมา

```
#define uchar unsigned char
#define uint unsigned
struct statefrom{
    unsigned long s;
    uint t;
    uchar done;
};
```

ประกาศตัวแปรใหม่
เรียกใช้ในโครงสร้าง

จะเห็นได้ว่าจะใช้ #define ประกาศตัวแปรประเภทของข้อมูลใหม่ขึ้นมา และถูกเรียกใช้ในตัว
แปรโครงสร้าง ถ้าหากต้องการให้ตัวแปรโครงสร้างนี้มีชื่อว่า state จทำได้ดังนี้

```
struct statefrom    state;

↑                  ↑
ข้อมูลแบบโครงสร้าง  ชื่อตัวแปร
```

ถ้าหากต้องการประกาศตัวแปรแบบ โครงสร้างหลายตัวสามารถทำได้ในรูปของอาร์เรย์ของโครง
สร้าง (array of structures) ดังตัวอย่างต่อไปนี้

```
#define uchar unsigned char
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define uint unsigned

struct stateform {

    unsigned long s;

    uint t;

    uchar done;

};

struct stateform state[20];
```

จะเป็นการประกาศตัวแปรอาร์เรย์ชื่อ state ที่มีทั้งหมด 20 เซลล์ แต่ละเซลล์จะเป็นตัวแปรแบบโครงสร้างขนาด 7 ไบต์ ทำให้ตัวแปรนี้ใช้หน่วยความจำทั้งหมด 140 ไบต์ (20 x 7)

5.14 โปรแกรมควบคุมการทำงาน INDUCTION COOKING

โดยแบ่งออกเป็นส่วยย่อยๆดังนี้

5.14.1 โปรแกรมควบคุมสวิตช์กด(KEYPACK)

สวิตช์กดจะประกอบไปด้วย สวิตช์ POWER, UP และ DOWN เพื่อเลือกระดับความร้อน โดยมีหลอด led แสดงผล POWER ON, ร้อนมาก(RED), ร้อน(YELLOW), อุณหภูมิ(GREEN)

```
#include<p18f458.h>

#define SW_START PORTCbits.RC0 /* ใช้ขา 15 พอร์ต C บิต RC0 เป็นสวิตช์ POWER */
#define SW_UP PORTCbits.RC1 /* ใช้ขา 16 พอร์ต C บิต RC1 เป็นสวิตช์ UP */
#define SW_DOWN PORTCbits.RC3 /* ใช้ขา 18 พอร์ต C บิต RC3 เป็นสวิตช์ DOWN */
#define POWER PORTCbits.RC7 /* ใช้ขา 26 พอร์ต C บิต RC7 แสดงผล POWER */
#define RAD PORTCbits.RC6 /* ใช้ขา 25 พอร์ต C บิต RC6 แสดงผลร้อนมาก(RED) */
#define YELLOW PORTCbits.RC5 /* ใช้ขา 24 พอร์ต C บิต RC5 แสดงผลร้อน(YELLOW) */
#define GREEN PORTCbits.RC4 /* ใช้ขา 23 พอร์ต C บิต RC4 แสดงผลอุณหภูมิ(GREEN) */

void delay(unsigned int count) /* ฟังก์ชันหน่วงเวลา ชื่อ delay */
{
    int i,j;
    for(i=0; i< count; i++)
        for(j=0; j< 500; j++);
}

void power_on(void) /*ฟังก์ชัน power on เริ่มทำงานของINDUCION COOKING*/
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

POWER = 0;          /* "0" หลอด LED สว่าง , "1" หลอด LED ดับ */
RAD = 0;
YELLOW = 1;
GREEN = 1;
}

void power_off(void) /* ฟังก์ชัน power_off */
{
    POWER = 1;
    RAD = 1;
    YELLOW = 1;
    GREEN = 1;
}

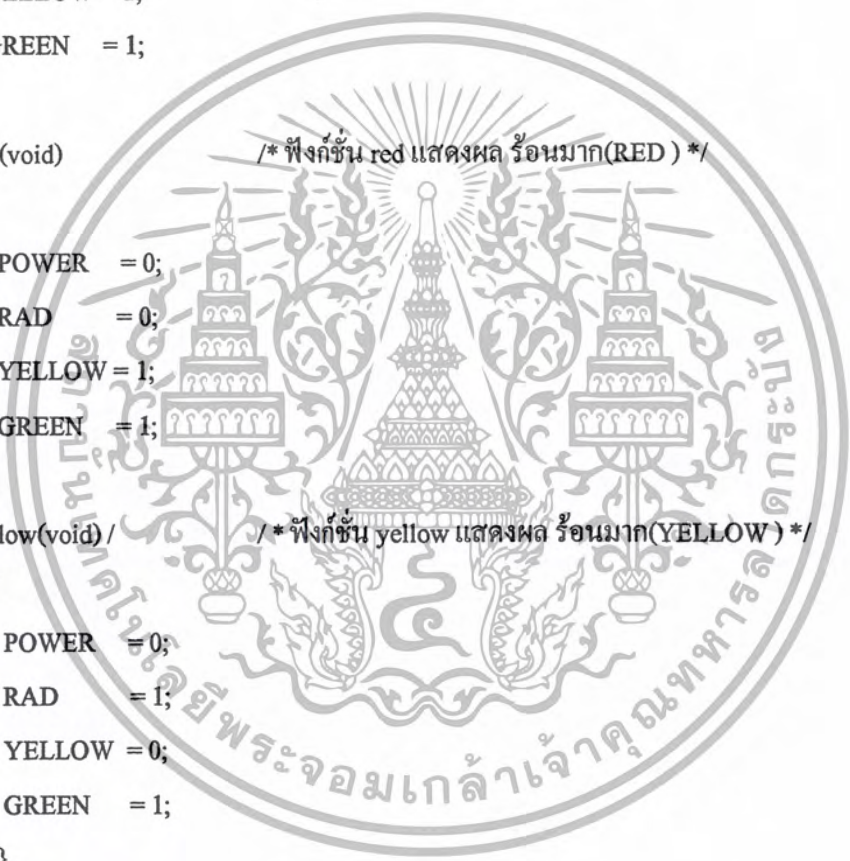
void red(void) /* ฟังก์ชัน red แสดงผล ร้อนมาก(RED) */
{
    POWER = 0;
    RAD = 0;
    YELLOW = 1;
    GREEN = 1;
}

void yellow(void) /* ฟังก์ชัน yellow แสดงผล ร้อนมาก(YELLOW) */
{
    POWER = 0;
    RAD = 1;
    YELLOW = 0;
    GREEN = 1;
}

void green(void) /* ฟังก์ชัน green แสดงผล ร้อน(GREEN) */
{
    POWER = 0;
    RAD = 1;
    YELLOW = 1;
    GREEN = 0;
}

void main (void) /* โปรแกรมหลัก */

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
TRISC =0b00001011;          /* กำหนด input/output พอร์ตC “1” เป็นinput, “0” เป็น output */
PORTCbits.RC4 = 1;
PORTCbits.RC5 = 1;
PORTCbits.RC6 = 1;
PORTCbits.RC7 = 1;
LAMP      = 3;
while(1)                    /* วนลูปซ้ำตรวจสอบการกดสวิตช์ทั้งหมด*/
{
    if(SW_START == 0)      /*ตรวจสอบการกดสวิตช์ POWER */
    {
        power_on();
        delay(200);
        while(SW_START == 0);
        do
        {
            if((SW_UP == 0) && (LAMP < 3)) /*ตรวจสอบการกดสวิตช์ UP*/
            {
                delay(100);
                if((SW_UP == 0) && (LAMP < 3))
                {
                    LAMP++;
                    if(LAMP == 2) yellow(); else red();
                    delay(200);
                    while(SW_UP == 0);
                }
            }
            if((SW_DOWN == 0) && (LAMP > 1)) /*ตรวจสอบการกดสวิตช์ DOWN*/
            {
                delay(200);
                if((SW_DOWN == 0) && (LAMP > 1))
                {
                    LAMP--;
                    if(LAMP == 2) yellow(); else green();
                    delay(200);
                    while(SW_DOWN == 0);
                }
            }
        }
    }
}

```



```

    }
    while(SW_START == 1);          /* ตรวจสอบการกดสวิทซ์ POWER OFF */
    power_off();
    delay(500);
    while(SW_START == 0);
    LAMP = 3;
  }
}
}
}

```

5.14.2 โปรแกรมกำเนิดเสียงจากลำโพงขนาดเล็ก(PIZZO)

ให้หลักการควบคุมกระแสไหลผ่านลำโพงกลับไปกลับมาจะทำให้ลำโพงสั่นออกมาเป็นเสียง

```

#include<p18f458.h>
#define SOUND PORTDbits.RD4 /*ใช้ขา 19 พอร์ต D บิต RD4 เป็นขาเอาต์พุตต่อกับลำโพง
pizzo*/
void sound(unsigned char freq,int time) /* ฟังก์ชันกำเนิดเสียง */
{
  unsigned char i;
  while (time > 0)
  {
    SOUND = 0;
    for(i=1; i<freq; i++)
    time--;
    SOUND = 1;
    for(i=1; i<freq; i++)
    time--;
  }
}
void main(void)
{
  TIRSDbits.TIRSD4 = 0; /* กำหนดบิต RD4 เป็นเอาต์พุต */
  sound(90, 3000); /* เรียกโปรแกรมกำเนิดเสียง */
}

```

5.14.3 โปรแกรมควบคุมการปิดเปิดพัลลวมระบายนรอื่น

```
#include<p18f458.h>
#define FAN    PORTEbits.RE1 //ให้ขา 9 พอร์ต E บิต RE1 เป็นเอาต์พุตพัลลวม
void main (void)
{
    TIRSEbits,TIRSTE1 = 0;
    FAN = 0;           //ให้ "0" พัลลวมทำงาน, "1" หยุดทำงาน
}
```

5.14.4 โปรแกรมสร้างคลื่นรูปสี่เหลี่ยมด้วยโหมด CCP

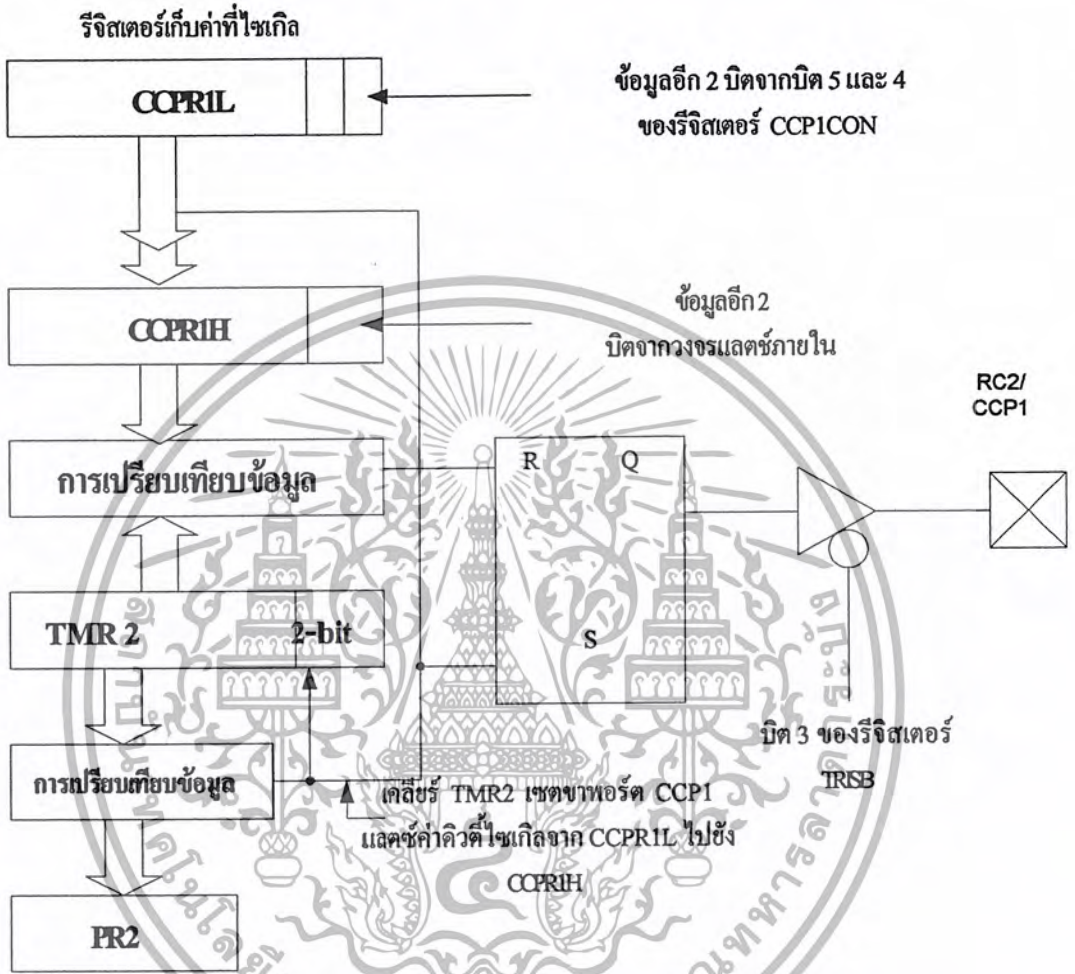
รายละเอียดของโมดูล CCP (Capture /Compare/PWM) ใน PIC 18F458 จะมีโมดูลตรวจจับสัญญาณเปรียบเทียบข้อมูลและสร้าง PWM หรือเรียกย่อว่า CCP (Capture /Compare/PWM) ในโมดูล CCP นี้บรรจุรีจิสเตอร์ 16 บิตที่สามารถทำงาน 3 หน้าที่ คือ เป็น รีจิสเตอร์เก็บค่าของสัญญาณที่ตรวจจับได้ (Capture register) , เป็นรีจิสเตอร์สำหรับเก็บข้อมูลสำหรับเปรียบเทียบ (Compare register) , เป็นรีจิสเตอร์เก็บค่าคิ่วถี่ที่เกิดเพื่อสร้างสัญญาณมอดดูเลชั่นทางความกว้างของพัลล์หรือ PWM

การทำงานในโหมด PWM

ในโหมดนี้โมดูล CCP จะกำเนิดสัญญาณมอดดูเลชั่นทางความกว้างพัลล์ หรือ PWM ความละเอียด 10 บิต ลักษณะของสัญญาณ PWM รวมถึงค่าพารามิเตอร์ที่ควรทราบแสดงในรูปที่ 5-9 โดยมีไดอะแกรมการทำงานแสดงในรูปที่ 5-10 สัญญาณ PWM ที่สร้างขึ้นนี้จะส่งออกทางขา RC2/CCP หลักการสร้างสัญญาณ PWM ของโมดูล CCP มีดังนี้

1. กำหนดค่าให้แกรีจิสเตอร์ RC2 เพื่อกำหนดค่าเวลาของสัญญาณ PWM ที่ต้องการ
2. กำหนดค่าคิ่วถี่ที่เกิด โดยเขียนข้อมูลลงในรีจิสเตอร์ CCPR1L ร่วมกับบิต 5 และ 4 ในรีจิสเตอร์ CCP1CON
3. กำหนดให้ขาพอร์ต RC2/CCP เป็นเอาต์พุต เพื่อเป็นทางออกของสัญญาณ PWM โดยการเคลียร์บิต 3 ของรีจิสเตอร์ TRISB
4. กำหนดค่าปริสเกลเลอร์ของ TMR2 และเอ็นเอเบิลการทำงานของไทมเมอร์ 2
5. กำหนดให้โมดูล CCP ทำงานในโหมด PWM

ทันทีที่โมดูล CCP เริ่มทำงาน ค่าของ TMR2 จะเพิ่มขึ้นจนเท่ากับ PR2 ที่ขาพอร์ต RC2/CCP จะเกิดลอจิก “1” และคงสถานะอยู่เช่นนั้น หลังจากค่าของ TMR2 จะเคลียร์ พร้อมกันนั้นค่าดีวีดีไซ์ที่เกิดขึ้นกำหนดไว้ในรีจิสเตอร์ CCP1RL และ 2 บิตใน CCP1CON จะถูกถ่ายทอดไปยัง CCP1RH และ 2 บิตในหน่วยความจำพิเศษ เพื่อทำการเปรียบเทียบกับค่าใน TMR2 และบิตปรีสเกลเลอร์ 2 บิต



รูปที่ 5-9 ไลอะแกรมการทำงานของโมดูล CCP เพื่อสร้างสัญญาณ PWM

ซึ่งเริ่มต้นเพิ่มค่าใหม่ จนกระทั่งเมื่อข้อมูลทั้ง 2 กลุ่มเท่ากัน จะส่งสัญญาณไปทำให้ขาพอร์ต RC2/CCP กลับมาเป็นลอจิก “0” และคงสถานะอยู่เช่นนั้นจนกระทั่งค่าของ TMR2 เท่ากับ PR2 อีกครั้ง ก็เกิดสัญญาณลอจิก “1” เป็นการเริ่มต้นรอบใหม่ของสัญญาณ และจะทำงานวนเช่นนี้จนกระทั่งมีการคิส์เอเบิล

ค่าดีวีดีไซ์ของสัญญาณ PWM ที่สร้างขึ้นนี้สามารถเปลี่ยนแปลงได้ตลอดเวลา แต่ค่าของคาบเวลาหรือความถี่จะไม่เปลี่ยนแปลง จนกว่าจะหยุดการทำงาน แล้วกำหนดค่าของเวลาลงในรีจิสเตอร์ PR2 ใหม่ ดังนั้นสัญญาณ PWM ที่สร้างขึ้นจึงอาจมีทั้งช่วงดีวีดีไซ์สูงหรือต่ำสลับไปมาได้ตามความต้องการของผู้ใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ประโยชน์ของสัญญาณ PWM ที่สร้างขึ้นนี้คือ นำไปใช้ในการควบคุมความเร็วมอเตอร์ไฟตรง ควบคุมความสว่างของหลอดไฟหรือใช้ในวงจรหรีไฟในเอง ใช้ในการปรับแรงดัน โดยต้องมีการต่อวงจรกรองความถี่ต่ำผ่านทางที่เอาต์พุทของขาพอร์ต RC2/CCP เป็นต้น

การกำหนดคาบเวลาของสัญญาณ PWM

สามารถกำหนดโดยการเขียนข้อมูลลงในรีจิสเตอร์ PR2 (รีจิสเตอร์คาบเวลาของไทเมอร์ 2) แล้วนำค่าของ PR2 มาคำนวณหาค่าคาบเวลาของสัญญาณ PWM ได้จาก

คาบเวลาของสัญญาณ PWM = (ค่าในรีจิสเตอร์ PR2+1) x 4 x Tosc x ค่าปีสเกลเลอร์ของTMR2

โดยที่

Tosc คือ ค่าคาบเวลาของสัญญาณนาฬิกาหลักมีหน่วยเป็นวินาทีและค่าในรีจิสเตอร์ทั้งหมด

คำนวณในเลขฐานสิบ

ส่วนความถี่ของสัญญาณ PWM สามารถคำนวณได้จาก

$$f = \frac{1}{t_{pwm}}$$

โดยที่ $t_{pwm} =$ คาบเวลาของสัญญาณ PWM

เมื่อค่าของ TMR2 เท่ากับ PR2 จะเกิดเหตุการณ์ขึ้น 3 เหตุการณ์ในไซเคิลการทำงานถัดไป ดังนี้

1. เกิดการเคลียร์ค่าของ TMR2
2. เกิดลอคจิก “1” ขึ้นที่ขา RC2/CCP ยกเว้นในกรณีที่กำหนดค่าคิวตี้ไซเคิลของสัญญาณ PWM เท่ากับ 0% ที่ขา RC2/CCP นี้จะไม่เกิดการเซตเป็นลอคจิก “1”
3. ค่าคิวตี้ไซเคิลของสัญญาณ PWM จะถูกแลตซ์เก็บไว้ โดยถ่ายทอดข้อมูลจาก CCP1L ไปยัง CCP1H

การกำหนดค่าความถี่ของสัญญาณ PWM จะไม่ใช่โพสต์สเกลเลอร์ภายในไทเมอร์ 2

การกำหนดค่าคิวตี้ไซเคิลของสัญญาณ PWM

สามารถกำหนดโดยการเขียนข้อมูลไปยังรีจิสเตอร์ CCP1L ร่วมกับบิต 5 และ 4 ของรีจิสเตอร์ CCP1CON ทำให้สามารถกำหนดความละเอียดของสัญญาณ PWM ได้สูงถึง 10 บิต โดย 8 บิตบนจะใช้ข้อ

มูลในรีจิสเตอร์ CCPR1L ส่วน 2 บิตล่างใช้ข้อมูลในบิต 5 และ 4 ของรีจิสเตอร์ CCP1CON สามารถคำนวณหาค่าควิตซ์ไชเกิดได้จากสูตร

ควิตซ์ไชเกิดของสัญญาณ PWM = ค่าในรีจิสเตอร์ CCPR1L (8 บิต) ร่วมกับข้อมูลบิต 5 และ 4

ของรีจิสเตอร์ CCP1CON (2บิต) x T_{Osc}

โดยที่ ค่าของ T_{Osc} คือ คาบเวลาของสัญญาณนาฬิกาหลัก มีหน่วยเป็นวินาที และ

ค่าในรีจิสเตอร์ทั้งหมดคำนวณในรูปของเลขฐานสิบ

สำหรับการกำหนดค่าของรีจิสเตอร์ CCPR1L และบิต 5 และบิต 4 ของรีจิสเตอร์ CCP1CON ซึ่งเป็นค่าควิตซ์ไชเกิดของสัญญาณ PWM สามารถกระทำได้ตลอดเวลา แต่ค่าควิตซ์ไชเกิดนี้จะไม่ถูกแลตช์หรือถ่ายทอดไปยังรีจิสเตอร์ CCPR1H จนกว่าค่าของรีจิสเตอร์ PR2 จะเท่ากับ TMR2 ซึ่งมีค่าเท่ากันแล้ว ค่า CCPR1H จะได้รับการเขียนโดยอัตโนมัติแต่ผู้ใช้งานไม่สามารถทำการเขียนได้โดยตรงแต่สามารถอ่านค่าได้เพียงอย่างเดียว

รีจิสเตอร์ CCPR1H และหน่วยความจำพิเศษภายในอีก 2 บิต (รวมเป็น 10 บิต) ถูกใช้เป็นปีพเพอร์สำหรับเก็บค่าควิตซ์ไชเกิดของสัญญาณของ PWM เพื่อแก้ไขความผิดพลาดของสัญญาณ PWM ที่สร้างขึ้นเมื่อค่ารีจิสเตอร์ CCPR1H และหน่วยความจำพิเศษภายในอีก 2 บิตเท่ากับค่าของ TMR2 ที่เชื่อมกับปริสเกลเลอร์ 2 บิตของ ไทเมอร์ 2 (บิต T2CKS1 และ T2CKS0 ในรีจิสเตอร์ T2CON) สถานะลอจิกที่ขา RC2/CCP1 จะกลายเป็น "0"

ความละเอียดของสัญญาณ PWM

ความละเอียดค่าสุดของสัญญาณ PWM จะขึ้นอยู่กับปริสเกลเลอร์ของ ไทเมอร์ 2 สามารถสรุปได้ดัง ตารางที่ 5-2 ในขณะที่ความละเอียดสูงสุดของสัญญาณ PWM จะสัมพันธ์กับความถี่ที่กำหนด, ความถี่ของสัญญาณของนาฬิกา และค่าของปริสเกลเลอร์ ซึ่งสามารถคำนวณได้จาก

$$\text{ความละเอียด (บิต)} = \log [f_{osc} / (f_{PWM} \times N)] / \log 2$$

โดยที่ f_{osc} คือ ความถี่ของสัญญาณนาฬิกาหลัก ในหน่วย Hz

f_{PWM} คือ ความถี่ของสัญญาณ PWM ในหน่วย Hz

N คือ ค่าของปริสเกลเลอร์ใน ไทเมอร์ 2

ค่าของรีจิสเตอร์	บิตกำหนดอปริสเกลเลอร์		ความละเอียดต่ำสุดของสัญญาณ PWM (ในหน่วยของเวลา)
	T2CKPS1	T2CKPS0	
1	0	0	TOSC
4	0	1	TCY = 4 TOSC
16	1	X	4TCY = 16 TOSC

ตารางที่ 5-2 แสดงความสัมพันธ์ของค่าปริสเกลเลอร์ในไทมเมอร์ 2 ที่มีต่อการกำหนดค่าความละเอียดต่ำสุดในหน่วยของเวลาของสัญญาณ PWM

ความถี่สัญญาณ PWM	1.22kHz	4.88kHz	19.53kHz	78.12kHz	156.3kHz	208.3kHz
ค่าของปริสเกลเลอร์	16	4	1	1	1	1
ค่าของรีจิสเตอร์ PR2	0xFF	0xFF	0xFF	0x3F	0x1F	0x17
ความละเอียด (บิต)	10	10	10	8	7	~7

ตารางที่ 5-3 แสดงความสัมพันธ์ของปริสเกลเลอร์และค่ารีจิสเตอร์ในไทมเมอร์ 2 ที่มีต่อการกำหนดรายละเอียดสูงสุดของสัญญาณ PWM ที่ความถี่ต่างๆ

โดยสัญญาณนาฬิกาหลักมีความถี่ 20 MHz ในตารางที่ 5-3 แสดงความสัมพันธ์ระหว่างความละเอียดของสัญญาณ PWM ค่าปริสเกลเลอร์และค่าของ PR2 ซึ่งใช้กำหนดคาบเวลาของสัญญาณ PWM ที่สัญญาณ PWM ความถี่ต่างๆกัน

ตัวอย่าง การสร้างสัญญาณรูปสี่เหลี่ยมที่มีความถี่ 5 KHz ปรับ Duty cycle ตอนเริ่มสตาร์ทจากศูนย์ถึงห้าสิบเปอร์เซ็นต์

```
#include<p18f458.h>
```

```
void delay(unsigned char count)
```

```
{
```

```
int ij;
```

```
for(i=0; i< count; i++)
```

```
for(j=0; j< 150; j++);
```

```
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void main (void)
{
    int i;
    CCP1CON = 0b00001100;    //ccpxm3:ccpxm0 11xx=pwm mode
    T2CONbits.TMR2ON = 0;    // STOP TIMER2 registers to POR state
    PR2 = 19;                // Set period
    CCPR1L = 0;              // Set Duty cycle
    TRISCbits.TRISC2=0;      //configure pin portc,2 an output
    T2CONbits.TMR2ON = 1;    // Turn on PWM1
    for(i=1; i<=10;i++)
    {
        CCPR1L = i;
        delay(20);
    }
    while(1);
}

```

การสร้างสัญญาณสี่เหลี่ยม 50 KHz โดยใช้ตัวคูณลด(ค่าปรีสเกลเลอร์)เข้าช่วยเพื่อให้ได้ความถี่สูง

ขึ้น

```
#include <P18F458.h>
```

```
void main(void)
```

```

{
    TRISCbits.RC2 = 0;        //ใช้ขา 17 พอร์ต C บิต RC2 เป็น เอาต์พุต CPP(สัญญาณ PWM)
    T2CONbits.T2CKPS1 = 0;    //กำหนดตัวคูณลด 1/4
    T2CONbits.T2CKPS0 = 1;    //กำหนดตัวคูณลด 1/4
    pr2 = 50;                 //ใส่ค่าความถี่ที่ต้องการ(KHz)
    T2CONbits.TMR2ON = 0;    // STOP TIMER2 registers to POR state
    PR2 = (2500 / pr2) - 1;   // Set period
    Duty = 5000 / pr2;
    CCPR1L = (Duty & 0b1111100)>>2;    // Set Duty cycle,ccpxm3:ccpxm0 11xx=pwm mode
}
(1/4)
CCP1CON = ((Duty & 0b00000011)<<4)|0b00001100;    //configure pin portc,2 an output
T2CONbits.TMR2ON = 1;    // Turn on PWM1
While(1);                //วนลูปอยู่กับที่
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.14.5 โปรแกรมแปลงสัญญาณอะนาลอกเป็นดิจิทัล

รายละเอียดของรีจิสเตอร์ ANALOG-TO-DIGITAL CONVERTER

สำหรับ A/D ในชิพ แบบ 28 pins จะมี 5 inputs แบบ 40 pins จะมี 8 inputs register ที่เกี่ยวข้องจะมีอยู่ 4 ตัวด้วยกันคือ

ADRESH	คือ	ค่า register ผลลัพธ์ byte สูง ของการแปลงสัญญาณ
ADRESL	คือ	ค่า register ผลลัพธ์ byte ต่ำ ของการแปลงสัญญาณ
ADCON0	คือ	register ควบคุมเกี่ยวกับ A/D byte ที่ 1
ADCON1	คือ	register ควบคุมเกี่ยวกับ A/D byte ที่ 2

ADCON0

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADCS	ADCS0	CHS2	CHS1	CHS0	GO/(DONE)	-	ADON
1							

ตารางที่ 5-4 อธิบายค่ารีจิสเตอร์แต่ละตัว

จากตารางเราสามารถอธิบายรายละเอียดได้ดังต่อไปนี้

Bit7-6: ADCS1:ADCS0: เป็น bit ที่ใช้เลือกสัญญาณนาฬิกาในการแปลง A/D

00 = $F_{osc}/2$ F_{osc} = ความถี่ของ crystal ที่ใช้

01 = $F_{osc}/8$

00 = $F_{osc}/32$

11 = FRC (เลือกใช้ความถี่จากวงจร RC ที่อยู่ภายใน)

Bit5-3: CHS2:CHS0 เป็น bit ที่ใช้เลือก channel ของสัญญาณ A/D

000 = channel 0, (RA0/AN0)

001 = channel 1, (RA0/AN1)

000 = channel 2, (RA0/AN2)

000 = channel 3, (RA0/AN3)

000 = channel 4, (RA0/AN4)

000 = channel 5, (RA0/AN5) (ไม่มีใน mcu แบบ 28 pins)

000 = channel 6, (RA0/AN6)

000 = channel 7, (RA0/AN7)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Bit 2: GO/(DONE): เป็นบิตที่ใช้ในการแสดงสถานะของการแปลง A/D

ถ้า ADON bit ถูก set เป็น 1 แล้ว เมื่อบิตนี้เป็น

1 หมายถึง A/D กำลังอยู่ในช่วงการแปลงค่า (ให้ set บิต นี้ในการเริ่มต้นการแปลงสัญญาณ)

0 หมายถึง A/D ไม่ได้อยู่ในช่วงการแปลงค่า (บิตนี้จะ mcu จะ clear เป็น 0 โดยอัตโนมัติ เมื่อทำการแปลงสัญญาณเสร็จเรียบร้อยแล้ว)

Bit 1: ยังไม่ถูกใช้งาน

Bit 0: ADON: A/D On bit (บอกสถานะของ A/D ในขณะนั้น)

1 = A/D convertor กำลังถูกใช้งาน

0 = A/D convertor ไม่ได้ถูกใช้งาน

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
ADFM	-	-	-	PCFG3	PCFG2	PCFG1	PCFG0

ตารางที่ 5-5 รายละเอียดรีจิสเตอร์ในแต่ละบิตของรีจิสเตอร์ ADCON1

1 = หลังจากแปลงสัญญาณให้ด้านซ้ายของ ADRESH เป็น 0

ADRES	ADRES
H	L
000000x	xxxxxxx
x	x

0 = หลังจากแปลงสัญญาณให้ด้านขวาของ ADRESL เป็น 0

ADRES	ADRES
H	L
Xxxxxxx	xx00000
x	0

Bit 6-4: ไม่ได้ถูกใช้

Bit 3-0: PCFG3:PCFG0 เป็นตัว set คุณสมบัติต่างๆ ในการทำงาน A/D ให้กับ PIC โดยเราสามารถเลือกว่าจะใช้ VREF แยกต่างหากหรือจะใช้จาก VDD,VSS ได้ก็ได้ ถ้าต้องการใช้ CHANNEL ไหน port ไหนสามารถดูได้ตารางที่ 5-6

PCFG3: PCFG0	AN7 RE2	AN6 RE1	AN5 RE0	AN4 RA5	AN3 RA3	AN2 RA2	AN1 RA1	AN0 RA0	VREF+	VREF -	CHAN/ REFS
0000	A	A	A	A	A	A	A	A	VDD	VSS	8/0
0001	A	A	A	A	VREF +	A	A	A	RA3	VSS	7/1
0010	D	D	D	A	A	A	A	A	VDD	VSS	5/0
0011	D	D	D	A	VREF +	A	A	A	RA3	VSS	4/1
0100	D	D	D	D	A	D	A	A	VDD	VSS	3/0
0101	D	D	D	D	VREF +	D	A	A	RA3	VSS	2/1
011x	D	D	D	D	D	D	D	D	VDD	VSS	0/0
1000	A	A	A	A	VREF +	VREF -	A	A	RA3	RA2	6/2
1001	D	D	A	A	A	A	A	A	VDD	VSS	6/0
1010	D	D	A	A	VREF +	A	A	A	RA3	VSS	5/1
1011	D	D	A	A	VREF +	VREF -	A	A	RA3	RA2	4/2
1100	D	D	D	A	VREF +	VREF -	A	A	RA3	RA2	3/2
1101	D	D	D	D	VREF +	VREF -	A	A	RA3	RA2	2/2
1110	D	D	D	D	D	D	D	A	VDD	VSS	1/0
1111	D	D	D	D	VREF +	VREF -	D	A	RA3	RA2	1/2

ตารางที่ 5-6 แสดง CHANNEL

โดยที่

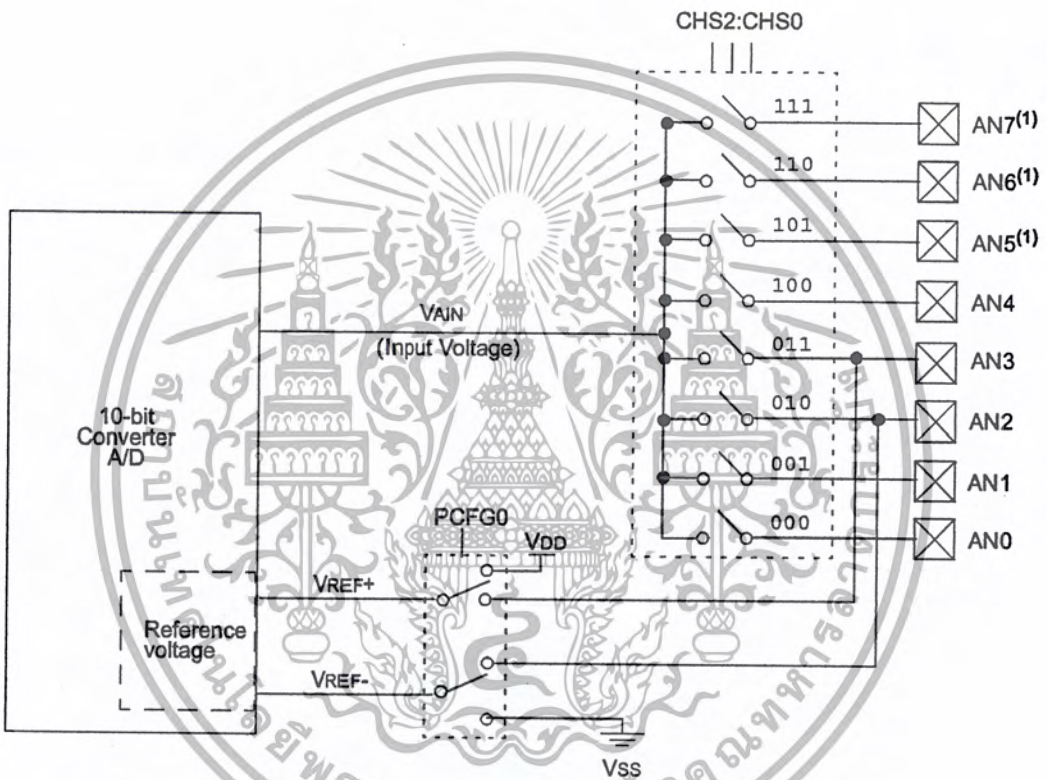
A หมายถึง Analog input

D หมายถึง Digital I/O

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วน An7-AN5 จะไม่มีในตระกูลที่เป็น 28 ขา

เมื่อการแปลง A/D เสร็จสิ้น ผลลัพธ์ของการแปลง A/D จะมีขนาด 10 bit ซึ่งจะเก็บอยู่ใน Register 2 ตัวต่อกันคือ ADRESH:ADRESL ส่วน register bit GO/DONE (ADCON0<2>) จะถูก cleared และ ADIF จะถูก set (A/D interruptflag) lock diagram ของ A/D จะเป็นดังรูปที่ 5-10



Note 1: Channels AN5 through AN7 are not available on PIC18F2X8 devices.

Note 2: All I/O pins have diode protection to VDD and VSS.

รูปที่ 5-10 แสดง Set Lock Diagram ของ A/D

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขั้นตอนการใช้งาน A/D module เลือก set config ของ A/D

1. เลือก analog pins/ voltage reference ด้วย ADCON1
2. เลือก A/D Input channel ด้วย ADCON0
3. เลือก A/D conversion clock (ความถี่ของสัญญาณนาฬิกาที่ใช้ใน A/D) จาก ADCON0
4. ตั้งให้ A/D module ทำงาน ด้วย ADCON0
5. ถ้าต้องการใช้ A/D interrupt ต้อง set flag ต่างๆ ดังนี้ คือ clear ADIF bit, set ADIE bit, set ADIF bit รอเวลาเพื่อให้ A/D module พร้อม (acquisition time) หาได้จากการคำนวณ
6. เริ่มทำการ A/D ด้วยการ set GO/DONE bit
7. รอจนกว่าการแปลง A/D จะเสร็จสมบูรณ์ ซึ่งได้ 2 วิธี คือ 1) เขียน โปรแกรมวน loop รอ ดู GO/DONE bit จนกว่าจะ cleared และ 2) รอ A/D interrupt
8. อ่านค่าผลลัพธ์ของ A/D จาก ADRESH:ADRESL โดย clear ADIF bit หลังจาก การอ่าน ด้วยถ้าใช้ A/D interrupt
9. หลังจากอ่านค่าเรียบร้อยแล้ว ต้องรอเป็นระยะเวลา 2TAD (มีอยู่ใน data sheet) ก่อนที่จะทำการแปลง A/D อีกครั้งหนึ่ง

การเลือก A/D Conversion Clock

เวลาที่ใช้ในการ แปลง A/D หนึ่งครั้งเรากำหนดให้เป็น TAD ในหนึ่งครั้งของการแปลง A/D PIC ต้องการอย่างน้อย 12TAD ต่อการแปลงแบบ 10 bits ในการเลือกสัญญาณความถี่ของ A/D สามารถเลือกได้ 4 แบบคือ

1. 2Tosc
2. 8Tosc
3. 32Tosc
4. Internal RC oscillator

ซึ่งค่าเหล่านี้เราเลือกได้จากการ set ค่า register (ADCS1:ADCS0)

ตัวอย่าง โปรแกรมแปลงสัญญาณจาก ANALOG TO DIGITAL ด้วยภาษาซี

```
#include<p18f448.h> // หัวโปรแกรม
static unsigned char AtoD_HI,AtoD_LO; // ประกาศตัวแปรเก็บค่าการประมวลผล
void InterruptHandlerHigh (void); //
void main(void) // ทางเข้าเมนโปรแกรม
{RCONbits.IPEN = 1; // ให้มีการจัดลำดับอินเตอร์รัพ
INTCONbits.GIE=1; // ให้ความสำคัญอินเตอร์รัพสูงสุด
PIR1bits.ADIF=0; // ทำการเคลียร์บิตอินเตอร์รัพโอเวอร์โฟลว์
PIE1bits.ADIE = 1; // ให้ A/D อินเตอร์รัพได้
IPR1bits.ADIP = 1; // ให้ A/D สำคัญสุด
TRISC=0; // ให้พอร์ต C เป็น เอาท์พุท
TRISA=0xFF; // ให้พอร์ต A เป็นอินพุท
TRISB=0; // ให้พอร์ต B เป็นเอาท์พุท
ADCON0 = 0b0000001; // Fosc/2, An0 ANALOG INPUT,สถานะการใช้งาน
ADCON1 = 0b00001110; // Left justified, เลือก AN0 input
ADCON0bits.ADON = 1; // บอกสถานะการใช้งาน A/D
ADCON0bits.GO = 1; // เริ่มแปลงสัญญาณ A/D
while(1); // วนลูปรอนกว่าแปลงสัญญาณเสร็จ
}
#pragma code InterruptVectorHigh = 0x08 // ประกาศเวกเตอร์อินเตอร์รัพสูงสุด
void
InterruptVectorHigh (void)
{
_asm
goto InterruptHandlerHigh //jump to interrupt routine
_endasm
}
//High priority interrupt routine
#pragma code
#pragma interrupt InterruptHandlerHigh
void
InterruptHandlerHigh () // ฟังก์ชันบริการอินเตอร์รัพ
{ ADCON0bits.ADON = 0; // หยุดการใช้งาน
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

PIR1bits.ADIF=0;           // เคลียร์ อินเตอร์รัฟ โอเวอร์โฟล
PORTB = ADRESH ;          // ส่งค่าการแปลงA/Dบิตสูงออกพอร์ท B
PORTC = ADRESL ;          // ส่งค่าการแปลงA/Dบิตต่ำออกพอร์ท C

ADCON0bits.ADON = 1;      // ใช้งาน A/D
ADCON0bits.GO = 1;        // เริ่มแปลงสัญญาณอีกครั้ง
}

```

ตัวอย่างใช้การวนลูปเมื่อต่ออินพุตอะนาลอกที่ขา 2 (AN0) ส่งค่าหลังจากการแปลงสัญญาณออกทางพอร์ท B(ADRESH) และD(ADRESH)

```

#include<p18f458.h>
void complate (void);
void main(void)
{ unsigned char H,L;
  TIRSB = 0;
  TIRSD = 0;
  ADCON0 = 0b00000001;
  ADCON1 = 0b00001110;
  TIRSAbits.TIRSTA0 = 1;
  ADCON0bits.ADON = 1;
  ADCON0bits.GO = 1;
  while(1)
  {
    if(ADCON0bits.GO == 0) complate ();
  }
  void complate (void)
  {
    PORTB = ADRESH ;
    PORTD = ADRESL ;
    ADCON0bits.GO = 1;
  }
}

```

การใช้การอินเตอร์รัปต์ ส่งค่าหลังจากการแปลงสัญญาณออกทางพอร์ท B(ADRESH) และD(ADRESH)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

#include<p18f458.h>

void InterruptHandlerHigh (void);

void main(void)
{
    TIRSB =0;
    TIRSD = 0;
    PIR1bits.ADIF =0;
    PIE1bits.ADIE = 1;
    INTCONbits.GIE =1;
    TRISAbits.TRISA0 = 1;
    ADCON0 = 0b00000001;
    ADCON1 = 0b00000100;
    ADCON0bits.ADON = 1;
    ADCON0bits.GO = 1;
while(1);
}
#pragma code InterruptVectorHigh = 0x08
void InterruptVectorHigh (void)
{
    _asm
    goto InterruptHandlerHigh //jump to interrupt routine
    _endasm
}
#pragma code
#pragma interrupt InterruptHandlerHigh
void InterruptHandlerHigh ()
{
    PIR1bits.ADIF=0;
    PORTB = ADRESH ;
    PORTD = ADRESL ;
    ADCON0bits.GO = 1;
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.15 โปรแกรมที่ใช้ควบคุม INDUCION COOKING

```

#include<p18f448.h>

#define SW_START PORTCbits.RC0 //pin<15> input sw_power
#define SW_UP PORTCbits.RC1 //pin<16> input sw_up
#define SW_DOWN PORTCbits.RC3 //pin<18> input sw_down
#define SOUND PORTDbits.RD4 //pin<19> pizzo
#define POWER PORTCbits.RC7 //pin<26>
#define RAD PORTCbits.RC6 //pin<25>
#define YELLOW PORTCbits.RC5 //pin<24>
#define GREEN PORTCbits.RC4 //pin<23>
#define CON_PWM PORTEbits.RE2 //pin<10>
#define FAN PORTEbits.RE1 //pin<9>
#define r 29
#define y 30
#define g 31
#define start 60
#define speed 150
#define protec 80
void main(void);
void over(void);
static unsigned char LAMP,TEMP;
static unsigned char temp_pr2;
void PWM(unsigned char pr2);
void PWM1(unsigned char pr2);
void PWM2(unsigned char pr2);
void power_on(void);
void power_off(void);
void power_off1(void);
void red(void);
void yellow(void);
void green(void);
void delay(unsigned int count)

```

{
 เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

int i,j;
for(i=0; i< count; i++)
{
if(ADCON0bits.GO ==0) over();
for(j=0; j< 500; j++);
}
}

void delay1(unsigned int count)
{
int i,j;
for(i=0; i< count; i++)
for(j=0; j< 500; j++);
}

void sound(unsigned char freq,int time)
{
unsigned char i;
while (time > 0)
{
SOUND = 0;
for(i=1; i<freq; i++)
time--;
SOUND = 1;
for(i=1; i<freq; i++)
time--;
}
return;
}

void main (void)
{
TRISEbits.TRISE2 = 0; //CONTROL PWM OUTPUT
CON_PWM = 0;
TRISEbits.TRISE1 =0;
TRISC =0b00001011; //<bit 0,1,3 input key ,bit 2 pwm,bit 4:7 led show
TRISDbits.TRISD4 = 0; //SOUND

```

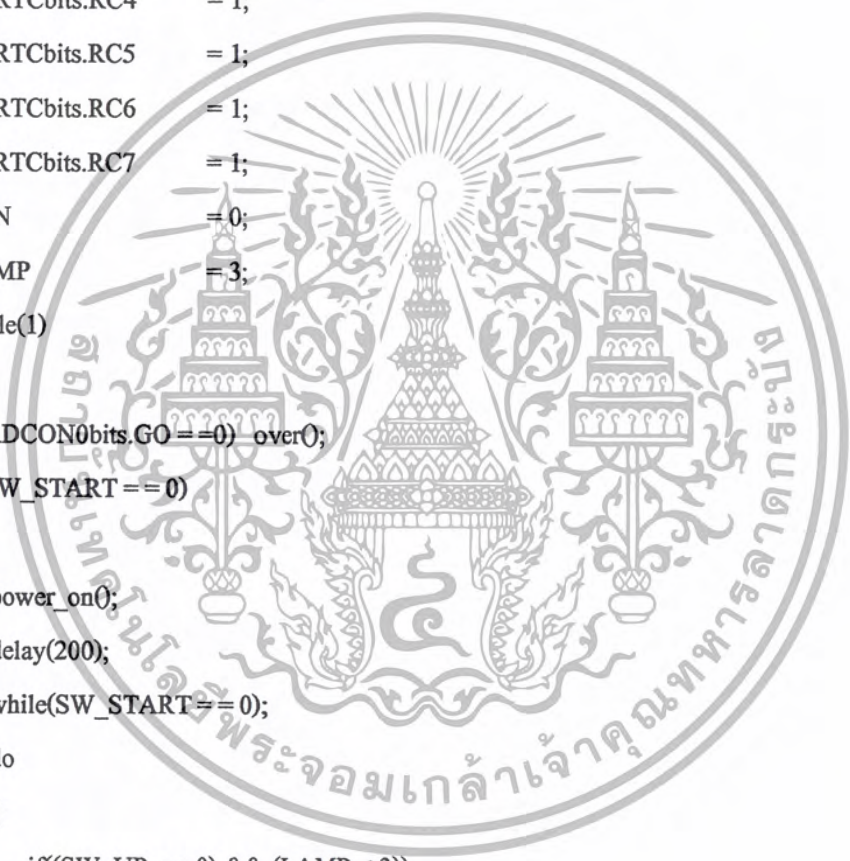
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

T2CONbits.T2CKPS1 =0;
T2CONbits.T2CKPS0 =1;

ADCON0 = 0b00000001;    //ADC
ADCON1 = 0b00001110;    //ADC
TRISAbits.TRISA0 = 1;    //ADC 220AC
TRISAbits.TRISA1 = 1;    //ADC 310DC
TRISAbits.TRISA2 = 0;    //TEMP
ADCON0bits.ADON = 1;
ADCON0bits.GO = 1;
PORTCbits.RC4 = 1;
PORTCbits.RC5 = 1;
PORTCbits.RC6 = 1;
PORTCbits.RC7 = 1;
FAN = 0;
LAMP = 3;
while(1)
{
if(ADCON0bits.GO ==0) over0;
if(SW_START == 0)
{
power_on0;
delay(200);
while(SW_START == 0);
do
{
if((SW_UP == 0) && (LAMP < 3))
delay(100);
if((SW_UP == 0) && (LAMP < 3))
{
LAMP ++;
if(LAMP == 2) yellow0; else red0;
delay(200);
while(SW_UP == 0);
}
}
}
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    if(ADCON0bits.GO == 0) over();
    if((SW_DOWN == 0) && (LAMP > 1))
    delay(100);
    if((SW_DOWN == 0) && (LAMP > 1))
    {
        LAMP--;
        if(LAMP == 2) yellow(); else green();
        delay(200);
        while(SW_DOWN == 0);
    }
}

while(SW_START == 1);
power_off();
delay(500);
while(SW_START == 0);
LAMP = 3;
}
}
}

/*****MONITOR*****/
void power_on(void)
{
    T2CONbits.TMR2ON = 0;
    sound(100,3500); //led power_red on
    FAN = 0;
    POWER = 0;
    RAD = 0;
    YELLOW = 1;
    GREEN = 1;
    PWM(r);
    return;
}

void power_off(void)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{
    CON_PWM = 0;
    sound(100,3500);          //led power_red_yellow_green off
    FAN      = 1;
    T2CONbits.TMR2ON = 0;    // Turn off PWM1
    POWER    = 1;
    RAD      = 1;
    YELLOW   = 1;
    GREEN    = 1;
    FAN      = 1;
    return;
}

void power_off1(void)
{
    CON_PWM = 0;
    sound(100,3500);          //led power_red_yellow_green off
    FAN      = 1;
    T2CONbits.TMR2ON = 0;    // Turn off PWM1
    POWER    = 1;
    RAD      = 1;
    YELLOW   = 1;
    GREEN    = 1;
    FAN      = 1;
    while(SW_START == 0);
    delay(200);
    while(SW_START == 1);
    _asm
    goto main
    _endasm
}

void red(void) //led red on _yellow off _green off
{
    sound(100,3500); POWER = 0;
    RAD = 0;

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

YELLOW = 1;
GREEN = 1;
if(temp_pr2 !=r)
PWM2(r);
return;
}

void yellow(void) //led red off_yellow on_green off
{
sound(100,3500);
POWER = 0;
RAD = 1;
YELLOW = 0;
GREEN = 1;
PWM2(y);
return;
}

void green(void) //led red off_yellow off_green on
{
sound(100,3500);
POWER = 0;
RAD = 1;
YELLOW = 1;
GREEN = 0;
PWM2(g);
return;
}

void PWM(unsigned char pr2)
{
unsigned int Duty,i;
T2CONbits.TMR2ON = 0;
{
PR2 = (2500 / start) - 1; // Set period
Duty = 5000 /start ;
CCPR1L = (Duty & 0b11111100)>>2; // Set Duty cycle

```



```

CCP1CON = ((Duty & 0b00000011)<<4)|0b00001100;
T2CONbits.TMR2ON = 1;
CON_PWM = 1;
PWM1(pr2);
}
return;
}

```

```
void PWM1(unsigned char pr2)
```

```

{
    unsigned int Duty,i;
    for(i=start; i>=pr2; i--)
    {
        if(ADCON0bits.GO ==0) over();
        PR2 = (2500 / i) - 1; // Set period
        Duty = 5000 / i;
        CCP1L = (Duty & 0b11111100)>>2; // Set Duty cycle
        CCP1CON = ((Duty & 0b00000011)<<4)|0b00001100;
        delay(speed);
    }
    temp_pr2 = pr2;
    return;
}

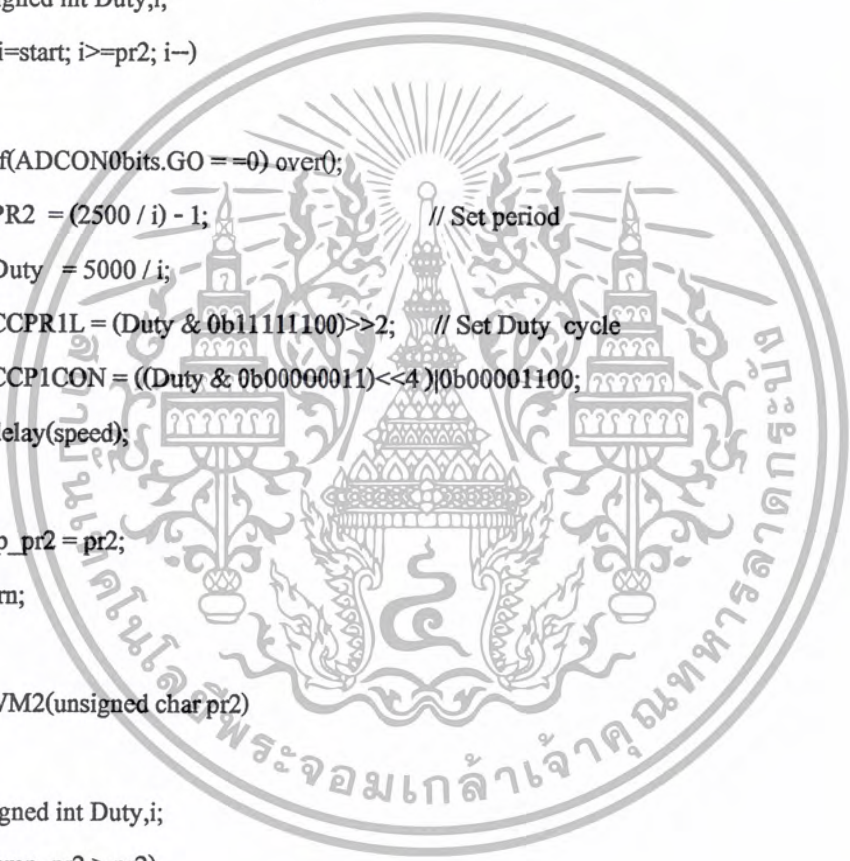
```

```
void PWM2(unsigned char pr2)
```

```

{
    unsigned int Duty,i;
    if(temp_pr2 > pr2)
    for(i= temp_pr2; i>=pr2; i--)
    {
        if(ADCON0bits.GO ==0) over();
        PR2 = (2500 / i) - 1; // Set period
        Duty = 5000 / i;
        CCP1L = (Duty & 0b11111100)>>2; // Set Duty cycle
        CCP1CON = ((Duty & 0b00000011)<<4)|0b00001100;
        delay(speed);
    }
}

```



```

}
else if(temp_pr2 < pr2)
for(i=temp_pr2; i <= pr2; i++)
{
if(ADCON0bits.GO == 0) over();
PR2      = (2500 / i) - 1;    // Set period
Duty     = 5000 / i;
CCPR1L  = (Duty & 0b11111100)>>2; // Set Duty cycle
CCP1CON = ((Duty & 0b00000011)<<4)|0b00001100;
delay(speed);
}
temp_pr2 = pr2;
return;
}
void over(void)
{
ADCON0bits.ADON = 0;
if(ADRESH > protec)
{
CON_PWM      = 0;
T2CONbits.TMR2ON = 0;
FAN          = 1;
POWER       = 1;
RAD         = 1;
YELLOW      = 1;
GREEN       = 1;
do
{
POWER      = 1;
SOUND     = 1;
delay1(300);
POWER     = 0;
SOUND    = 0;
delay1(300);

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}  
while(SW_START == 1);  
    delay1(200);  
    while(SW_START == 0);  
    _asm  
    goto main  
    _endasm  
}  
ADCON0bits.ADON = 1;  
ADCON0bits.GO = 1;  
return;  
}
```

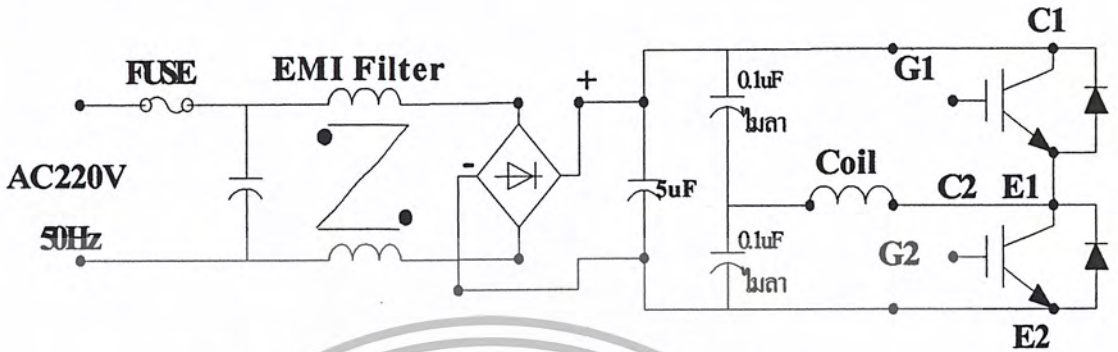


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 6

การออกแบบและสร้างเตาหุงต้มเหนียว

6.1 วงจรกำลัง

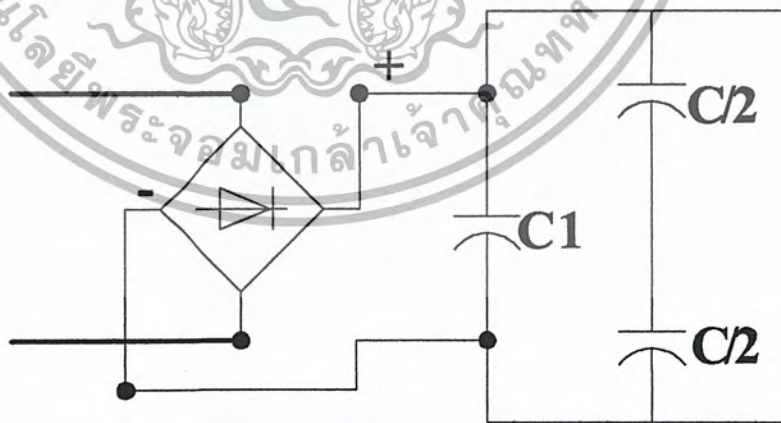


รูปที่ 6-1 วงจรกำลังของ เตาหุงต้ม

ในส่วนของวงจรกำลังตามรูปที่ 6-1 ประกอบด้วยส่วนสำคัญ มีหน้าที่และหลักการทำงานดังนี้

6.1.1 วงจรแปลงสัญญาณไฟฟ้ากระแสสลับให้เป็นสัญญาณไฟฟ้ากระแสตรง (Rectifier)

วงจรส่วนนี้ประกอบด้วยไดโอด 4 ตัว ต่อแบบ พูลบริดจ์ (Full Bridge) ทำหน้าที่รับไฟฟ้ากระแสสลับอินพุต 1 เฟส แล้วเปลี่ยนเป็นไฟฟ้ากระแสตรงที่มีการกระเพื่อมของแรงดันก่อนเข้าวงจรกรองแรงดันด้วยคาปาซิเตอร์ฟิลเตอร์เพื่อกรองแรงดันให้เรียบต่อไป จึงเลือกใช้บริดจ์ไดโอดที่เป็นแบบโมดูลขนาด 600 โวลต์ 25 แอมป์ โดยได้ออกแบบเพื่อให้สามารถทำงานที่แรงดันขาเข้าเกิดสถานะแรงดันเกิน (over voltage) และกระแสกระชากได้อีกด้วย



รูปที่ 6-2 รูปวงจรรีดิฟายเออร์

6.1.2 วงจรกรองแรงดันให้เรียบ (C-filter)

ในส่วนของตัวเก็บประจุที่ต่อหลังจากวงจรเรกติไฟเออร์นี้จะใช้สำหรับกรองแรงดันเพื่อลดการกระเพื่อมของแรงดัน โดยเราจะใช้อินเวอร์เตอร์เป็นตัวจ่ายแรงดัน จึงจำเป็นต้องมีตัวเก็บประจุที่มีขนาดใหญ่เพียงพอและทนแรงดันได้สูง จึงใช้ตัวเก็บประจุเป็นแบบอิเล็กโทรไลติก ขนาด 5000 ไมโครฟารัด 450 โวลต์ ซึ่งเป็นค่าที่เพียงพอที่จะไม่ทำให้เกิดการกระเพื่อมของแรงดัน และในการคำนวณหาขนาดของตัวเก็บประจุสามารถหาได้จาก

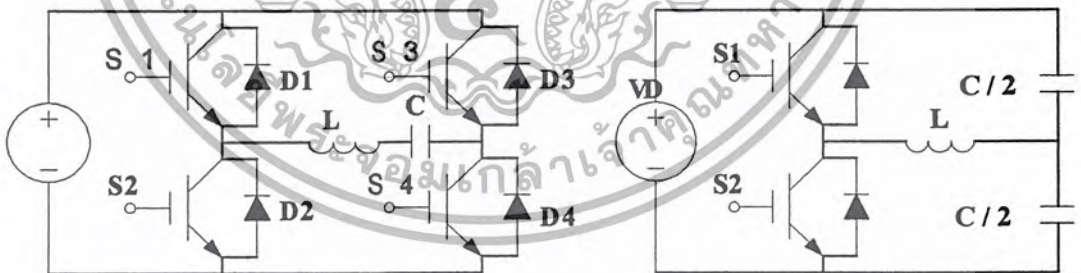
$$C_{\text{filter}} = \frac{I_O \cdot t_{\text{on(max)}}}{V_r} \quad (5-1)$$

โดย I_O คือ กระแสประสิทธิผล (root mean square) ของวงจรเรียงกระแส A
 $t_{\text{on(max)}}$ คือ ช่วงเวลาที่ C คายประจุ (Sec)
 V_r คือ ค่ายอดถึงยอดของแรงดันกระเพื่อมที่เอาที่พุทของวงจรเรียงกระแส (V)

6.1.3 วงจรอินเวอร์เตอร์

วงจรอินเวอร์เตอร์เป็นวงจรที่ทำหน้าที่เปลี่ยนไฟฟ้ากระแสตรงให้เป็นไฟฟ้ากระแสสลับและสามารถปรับเปลี่ยนแรงดันและความถี่เอาที่พุทได้ วงจรอินเวอร์เตอร์ที่ออกแบบสร้างขึ้นเป็นชนิดจ่ายแรงดัน เพราะสามารถควบคุมและออกแบบได้ง่าย

วงจรอินเวอร์เตอร์ที่ใช้โดยทั่วไปนั้น จะมีวงจรอินเวอร์เตอร์ชนิดฟูลบริดจ์และชนิดฮาร์ฟบริดจ์ ซึ่งสามารถแสดงได้ดังรูปที่ 5-2



รูปที่ 6-3 ก) วงจรอินเวอร์เตอร์ชนิดฟูลบริดจ์

ข) วงจรอินเวอร์เตอร์ชนิดฮาร์ฟบริดจ์

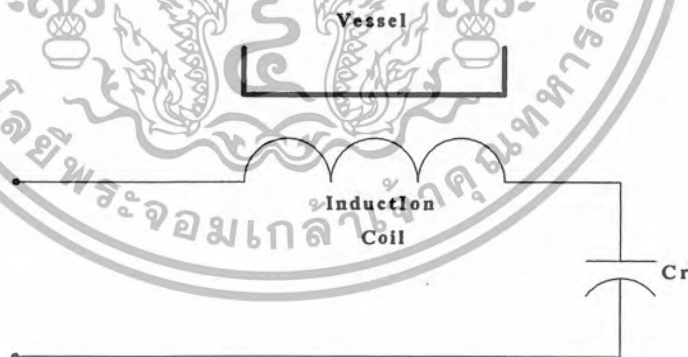
เราจะใช้อินเวอร์เตอร์ชนิดฮาร์ฟบริดจ์ โดยเราจะใช้ IGBT เป็นอุปกรณ์สวิตช์ ซึ่งมีพิคัด 50 แอมป์ 600 โวลต์ รุ่น CM 100PY-24H ซึ่ง IGBT สามารถทนพิคัดกำลังสูงได้ดี ในส่วนต่าง ๆ ของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วงจรอินเวอร์เตอร์นี้ เราจำเป็นต้องมีฟรีวิลลิงไดโอด (free-wheeling diode) ต่อขนานกับ IGBT เพื่อป้องกันกระแสไหลย้อนกลับจาก โหลดที่เป็นขดลวดเหนี่ยวนำ

6. 1.4 โหลดรีโซแนนซ์

สำหรับโหลดรีโซแนนซ์ที่ใช้กับวงจรอินเวอร์เตอร์ความถี่สูงโดยทั่วไป เช่น เครื่องชุบแข็งผิวดโลหะ เครื่องหลอมโลหะ เตาหุงต้มเหนี่ยวนำ บัลลาสต์อิเล็กทรอนิกส์ และเครื่องเชื่อมความถี่สูง จะมีข้อได้เปรียบที่สำคัญ คือ ลดการสูญเสียในการสวิตช์ เมื่อสวิตช์ตัดวงจรที่กระแสศูนย์ (Zero Current Switch :ZCS) หรือสวิตช์ตัดวงจรที่แรงดันศูนย์ (Zero Voltage Switch :ZVS) และข้อเสียเปรียบที่สำคัญของวงจรรีโซแนนซ์คือ ที่ตำแหน่งใกล้จุดความถี่รีโซแนนซ์กระแสในวงจรมีค่ามาก ส่งผลให้แรงดันตกคร่อมวงจรมีค่ามากด้วย สำหรับวงจรรีโซแนนซ์ที่มีค่า Q สูง ดังนั้นสวิตช์ที่เลือกใช้จะต้องสามารถทนแรงดันตกคร่อมขณะ OFF ได้สูง และ LC ในวงจรรีโซแนนซ์ที่เลือกใช้ก็เช่นกัน จะต้องมีความถี่ใหญ่เพื่อให้สามารถเก็บพลังงานได้มาก อินเวอร์เตอร์รีโซแนนซ์โดยทั่วไปจะมีด้วยกัน 2 ชนิด คือ อินเวอร์เตอร์รีโซแนนซ์ชนิดอนุกรมและชนิดขนาน ทั้งสองชนิดนี้มีข้อดีและข้อเสียแตกต่างกันดังแสดงในตารางที่ 5-2 และมีโครงสร้างของวงจรดังแสดงในตารางที่ 5-3 จะเห็นได้ว่าอินเวอร์เตอร์โหลดรีโซแนนซ์ชนิดอนุกรมจะใช้แหล่งจ่ายชนิดแรงดันคงที่ (Constant Voltage Source) เนื่องจากต้องใช้คาปาซิเตอร์ขนานใหญ่เพื่อกรองแรงดันให้เรียบร้อยแล้วจ่ายให้กับอินเวอร์เตอร์ สำหรับลักษณะคลื่นแรงดันจะเป็นรูปคลื่นสแควร์ และกระแสจะเป็นรูปคลื่นไซน์ ส่วนอินเวอร์เตอร์โหลดรีโซแนนซ์ขนานจะมีลักษณะที่ตรงข้ามกันคือ จะเป็นชนิดแหล่งจ่าย

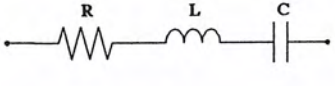
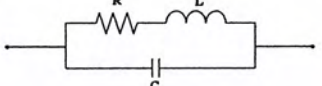
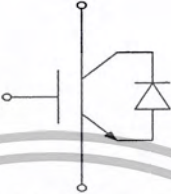
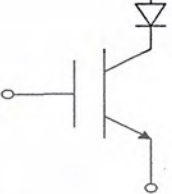

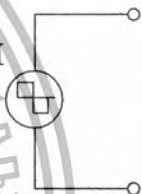
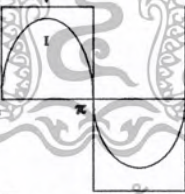
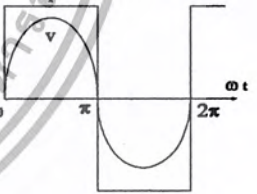


รูปที่ 6 - 4 แสดงวงจรโหลดรีโซแนนซ์ของเตาหุงต้ม

วงจรอินเวอร์เตอร์ชนิด	ข้อดี	ข้อเสีย
รีโซแนนซ์อนุกรม	<ul style="list-style-type: none"> - โครงสร้างง่ายและราคาถูก - สามารถจ่ายกำลังไฟฟ้าได้โดยตรงจากวงจรไดโอดเรกติฟายเออร์โดยไม่ต้องมีการปรับแรงดัน - กำลังไฟฟ้าที่จ่ายสามารถปรับควบคุมได้ง่ายโดยการปรับความถี่ที่วงจรอินเวอร์เตอร์ 	<ul style="list-style-type: none"> - ขณะป้อนสัญญาณเพื่อขับวงจรอินเวอร์เตอร์ วงจรโหลดจะเปิดวงจรออกไม่ได้ - ไม่สามารถทนการลัดวงจรที่โหลดได้ - ควบคุมกำลังไฟฟ้าโดยการปรับความถี่สวิตซ์ให้เลื่อนห่างออกจากความถี่เรโซแนนซ์ ทำให้กระแสไม่ค้อยเป็นไซน์ ซึ่งเป็นผลให้มีฮาร์โมนิกส์เกิดขึ้นตามมา
	<ul style="list-style-type: none"> - ขณะป้อนสัญญาณเพื่อขับวงจรอินเวอร์เตอร์ วงจรโหลดสามารถเปิดวงจรออกได้ - สามารถทนการลัดวงจรที่โหลดได้ - อุปกรณ์ L และ C ที่ใช้ในวงจรเรโซแนนซ์ไม่จำเป็นต้องมีพิกัดแรงดันสูง เนื่องจากกระแสเป็นรูปคลื่นออสซิลเลชัน จึงมีค่า Peak คงที่ ตำแหน่งรีโซแนนซ์ 	<ul style="list-style-type: none"> - ไม่สามารถจ่ายกำลังไฟฟ้าได้โดยตรงจากวงจรไดโอดเรกติฟายเออร์ แต่ต้องมีการปรับแรงดันจากการใช้สวิตซ์ควบคุม - จำเป็นต้องมี Choke เนื่องจากเป็น Contant Current - มีขนาดใหญ่เนื่องจากมี Choke และวงจรควบคุมแรงดันดิซอินพุทให้กับอินเวอร์เตอร์

ตารางที่ 6-2 การเปรียบเทียบข้อดีและข้อเสียของวงจรอินเวอร์เตอร์รีโซแนนซ์ชนิดอนุกรมและขนาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Inverter	Series Resonant	Parallel Resonant
Load		
Switch		
Source		
Output Waveform		

ตารางที่ 6-3 การเปรียบเทียบลักษณะของอุปกรณ์และรูปคลื่นแรงดันและกระแสที่ใช้ในวงจรอินเวอร์เตอร์
รีโซแนนซ์ชนิดอนุกรมและขนาน

กระแสคงที่ (Constant Current Source) เนื่องจาก ต้องใช้อินดักเตอร์ขนานใหญ่เพื่อกรองกระแสให้เรียบแล้วง่ายให้กับอินเวอร์เตอร์ ส่วนลักษณะรูปคลื่นแรงดันจะเป็นรูปคลื่นไซน์ในขณะรูปคลื่นกระแสจะเป็นรูปสแควร์ ในกรณีเตาหุงต้มเหนี่ยวนำนั้นเหมาะที่จะใช้กับวงจรอินเวอร์เตอร์เรโซ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แนบที่ชนิดอนุกรม เนื่องจากต้องการกำลังไฟฟ้าไม่มากนัก วงจรควบคุมไม่ซับซ้อนและมีขนาดเล็ก เหมาะที่จะใช้เป็นเครื่องใช้ไฟฟ้าในครัวเรือน

ในปริยญาณินพจน์นี้เราเลือกใช้โซ่แนบที่แบบอนุกรมและโหลด LC โดยค่า L นั้นได้มาจากการชดลวดทองแดงในลักษณะก้นหอย โดยใช้ลวดทองแดงเบอร์ 30 จำนวน 16 เส้น มาตีเป็นเกลียว และขดเป็นลักษณะก้นหอยทั้งหมด 43 รอบ และจะใช้ตัวเก็บประจุขนาด 0.12 ไมโครฟารัด

6.1.5 EMI FILTER

เตาหุงต้มเหนี่ยวนำนี้ ภายในวงจรจะมีอุปกรณ์สารกึ่งตัวนำ อุปกรณ์สารกึ่งตัวนำนี้เองจะเป็นตัวทำให้เกิดการรบกวน (EMI) ซึ่งสัญญาณรบกวนดังกล่าวจะทำให้เกิดการรบกวนกับอุปกรณ์อื่น ๆ เช่น ปรกวดสัญญาณที่ตัวไมโครคอนโทรลเลอร์ ซึ่งสัญญาณรบกวน (EMI) ที่เกิดจากอุปกรณ์สารกึ่งตัวนำสามารถแบ่งออกได้เป็น 2 ชนิด

- 1. Conducted EMI
- 2. Radiated EMI



6.2 ชุดควบคุมอินเวอร์เตอร์

6.2.1 วงจรควบคุม (control circuit)

วงจรควบคุมจะทำหน้าที่สร้างสัญญาณพัลส์รูปสี่เหลี่ยม (Square Wave) ซึ่งสัญญาณสี่เหลี่ยมดังกล่าวจะนำไปใช้ขับสัญญาณที่ขาเกตของ ไอจีบีที โดยโครงงานนี้เราจะใช้ไมโครคอนโทรลเลอร์ PIC18F458 เป็นตัวสร้างสัญญาณพัลส์รูปสี่เหลี่ยมเพื่อใช้ในการขับเกตของ ไอจีบีทีและใช้ไมโครคอนโทรลเลอร์

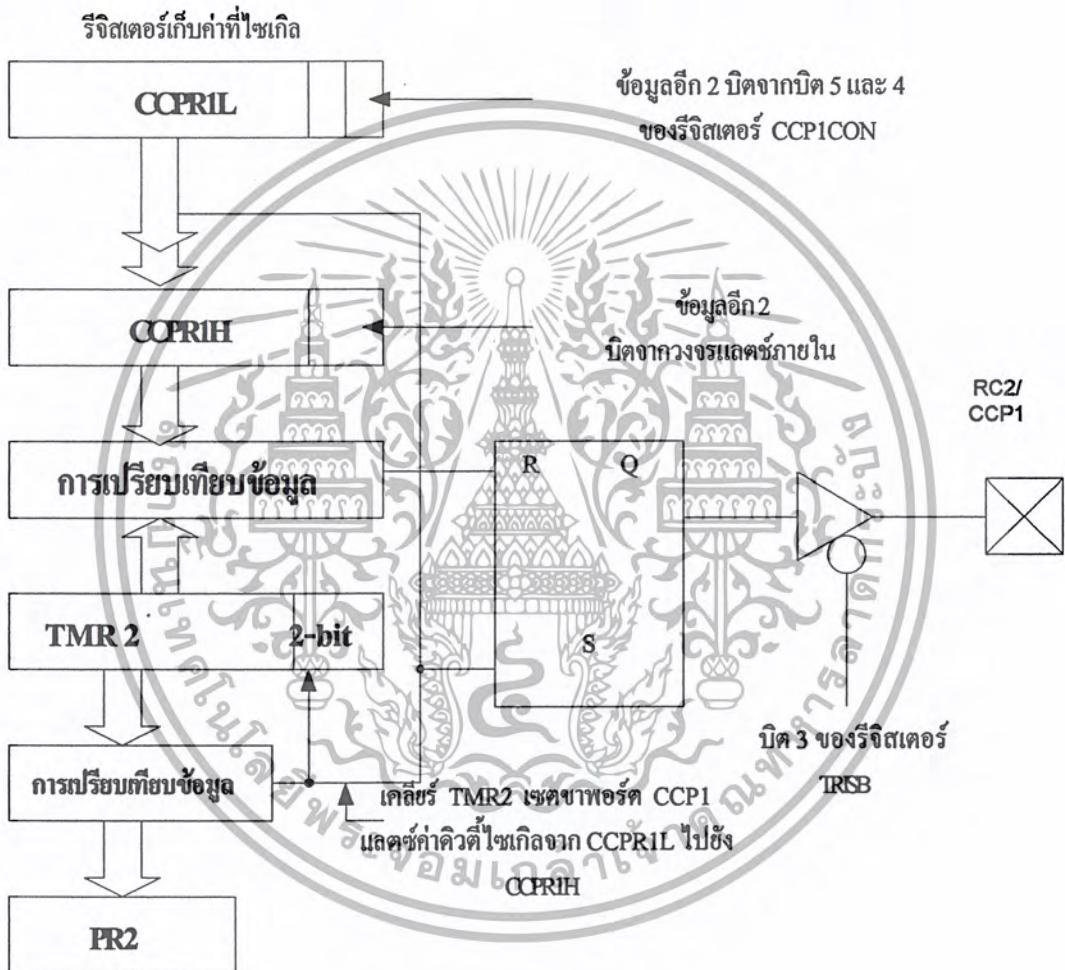
PIC18F 458 เป็นตัวประมวลผลสัญญาณที่มาจากวงจรควบคุมกระแสทางด้านอินพุทและ ทางด้านเอาต์พุท

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยส่วนที่ใช้ในการสร้างสัญญาณพัลส์รูปสี่เหลี่ยมนี้ ในไมโครคอนโทรลเลอร์ PIC 18F 458 จะมีโมดูลตรวจจับสัญญาณเปรียบเทียบข้อมูลและสร้าง PWM ซึ่งเรียกย่อ ๆ ว่า CCP (Capture/Compare/PWM) โดยมีโมดูล CCP นี้บรรจุรีจิสเตอร์ 16 บิต โดยมีการทำงาน 3 หน้าที่ คือ

1. เป็นรีจิสเตอร์เก็บค่าสัญญาณที่ตรวจจับได้ (Capture Register)
2. เป็นรีจิสเตอร์สำหรับเก็บข้อมูลสำหรับเปรียบเทียบ(Compare Register)
3. เป็นรีจิสเตอร์ที่ใช้เก็บค่าควิ์ไซเคลเพื่อสร้างสัญญาณมอดูลเลขขึ้นทางความกว้างของพัลส์หรือ

PWM



รูปที่ 6-6 ไดอะแกรมการทำงานของโมดูล CCP ที่สร้างสัญญาณ PWM

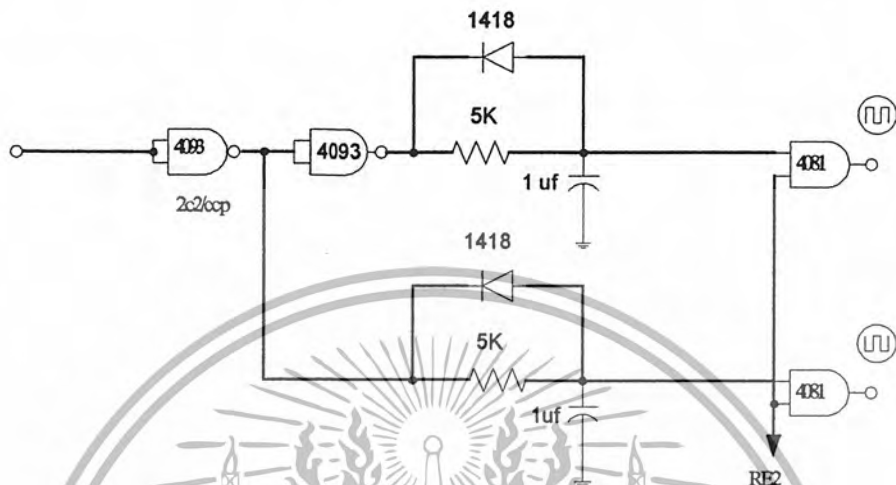
6.2.2 วงจรปรับแต่งเวลาเดทไทม์ (Dead Time)

เนื่องจากการทำงานของวงจรจะอาศัยหลักการสวิตซ์ซึ่งของไอจีบีที โดยจะสลับการทำงาน กล่าวคือ เมื่อไอจีบีทีตัวหนึ่งนำกระแส ไอจีบีทีอีกตัวหนึ่งต้องหยุดนำกระแส ดังนั้นข้อควรระมัดระวังในการควบคุมการทำงานในส่วนนี้ คือจะต้องไม่ให้ไอจีบีทีสองตัวนำกระแสพร้อมกัน เพราะจะทำให้แหล่งจ่าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ไฟฟ้าที่ต่ออยู่กับวงจรกำลังจะถูกตัดวงจรลงดิน มีผลทำให้กระแสจำนวนมากไหลผ่านไอจีบีทีสองตัวทำให้เกิดความเสียหายได้

ในการป้องกันไม่ให้ไอจีบีทีสองตัวนำกระแสพร้อมกัน ซึ่งเรียกเวลาช่วงนี้ว่า เดดไทม์ (Dead Time) ซึ่งในโครงการนี้เราจะใช้ไอซีเบอร์ 4093 และ 4081 ซึ่งสามารถแสดงได้ดังรูปที่ 3-9

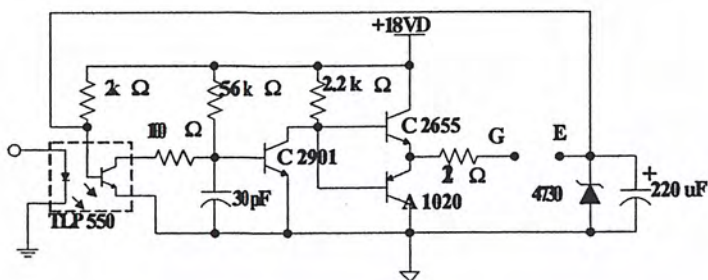


รูปที่ 6-7 วงจรปรับแต่งเวลา (Dead Time)

6.2.3 วงจรขยายสัญญาณขับไอจีบีที (Gate Drive Circuit)

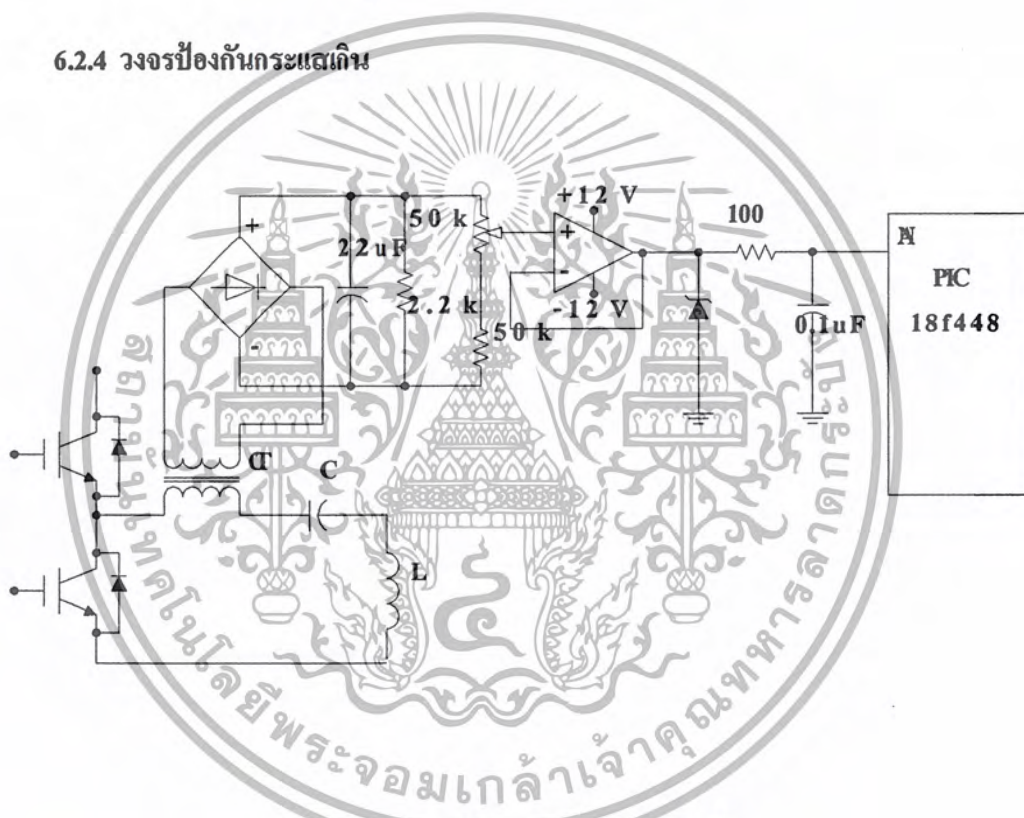
ประกอบด้วยส่วนสำคัญ 2 ส่วน คือ ออปโตคัปเปิลเลอร์ (Opto Coupler) ซึ่งใช้ในการแยกวงจรควบคุมออกจากวงจรถูกควบคุมเพื่อป้องกันความเสียหายที่อาจเกิดขึ้นกับวงจรคอนโทรลเลอร์ และส่วนวงจขยายที่ใช้คลาสเอบี (Class AB) มีหลักการทำงานดังนี้ เริ่มต้นสัญญาณ PWM ผ่านเข้ามาในออปโตคัปเปิลเลอร์ เบอร์ TLP550 ซึ่งทำให้ LED (Light Emitting Diode) เปล่งแสงขึ้นและส่งสัญญาณแสงไปทรานซิสเตอร์ที่อยู่ภายในให้ทำงานเป็นสวิตช์ตามลักษณะของสัญญาณที่เข้ามา 2 และ ขา 3 จากนั้นสัญญาณเอาต์พุตจะส่งต่อมายังส่วนวงจขยาย

วงจขยายเริ่มจาก ใช้ทรานซิสเตอร์เบอร์ C2901 เป็นตัวขยายสัญญาณ และส่งสัญญาณมาที่ทรานซิสเตอร์ เบอร์ C2655 และเบอร์ A1020 ที่ต่อกันเป็นวงจขยายคลาสเอบีดังรูปที่ ซึ่งไบแอส (Bias) ทรานซิสเตอร์ เบอร์ C2655 ให้ทำหน้าที่ขยายในโซลิตบวกลบและทรานซิสเตอร์ เบอร์ A1020 ทำหน้าที่ขยายในโซลิตลบ จากนั้นนำเอาท์พุตทั้งสองมารวมกันจะได้สัญญาณเอาต์พุตที่สูงขึ้น ซึ่งค่าแอมพลิจูดทั้งโซลิตบวกลบและลบขึ้นกับซีเนอริโคโอดที่ต่อในรูป สุดท้ายสัญญาณที่ได้จะส่งไปยังขาเกตของไอจีบีที



รูปที่ 6-8 วงจรขยายสัญญาณขั้วไอจีบีที

6.2.4 วงจรป้องกันกระแสเกิน



รูปที่ 6-9 แสดงวงจรป้องกัน

จากรูปที่ 3-10 จะใช้ current transformer ต่ออนุกรมกับโหลดตรวจจับสัญญาณกระแสที่ไหลผ่าน ไอจีบีทีขณะจ่ายโหลดเข้าวงจรบริดจ์เรกติฟายเออร์แปลงเป็นสัญญาณไฟฟ้ากระแสตรงผ่านชุดบัฟเฟอร์ หลังจากนั้นผ่านซีเนอร์ไดโอด เพื่อให้สามารถรักษาระดับแรงเกินเพื่อที่จะให้ไมโครคอนโทรลเลอร์ประมวลผลได้ โดยไมโครคอนโทรลเลอร์จะประมวลผลสัญญาณอนาล็อกเป็นดิจิตอล เพื่อควบคุมสัญญาณในการขับเคลื่อนที่ขา RE2

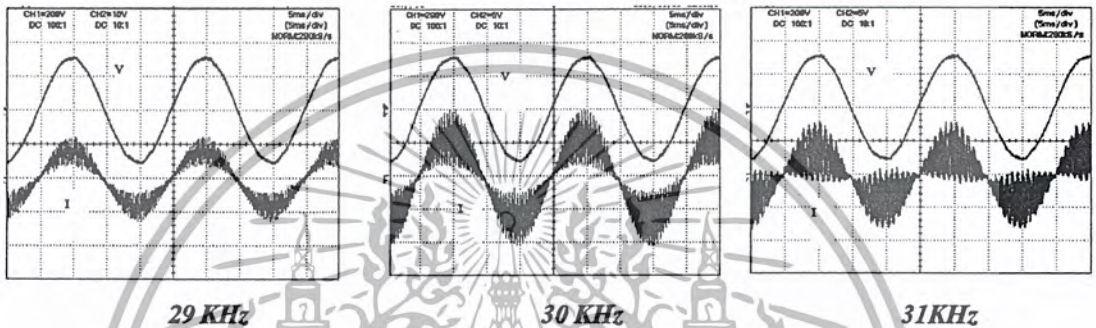
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 7

ผลการทดลอง

จากการทดสอบเตาหุงต้มเหนียวมาโดยเราจะทำการทดสอบทั้งทางด้านอินพุทของแหล่งจ่ายการไฟฟ้า โดยทำการปรับความถี่ จาก 29 KHz จนถึง 30 KHz โดยกำหนดความถี่รีโซแนนซ์ที่ 29 KHz โดยจะได้ลักษณะรูปคลื่นแรงดันและกระแสดังรูปที่ 7-1 โดยการทดสอบโครงการนี้พบว่าที่กำลังที่พิกัดที่ออกแบบไว้คือ 1.2 KW สามารถต้มน้ำเดือดได้ในเวลา 7 นาที

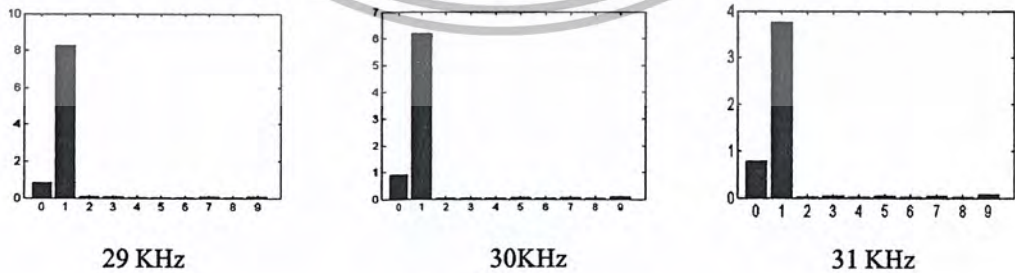
7.1 ผลการทดลองทางด้านอินพุทขณะเปลี่ยนความถี่ต่างๆ



รูปที่ 7-1 ลักษณะรูปคลื่นแรงดัน (Vs) และกระแส (Is) ของแหล่งจ่ายจากการไฟฟ้า



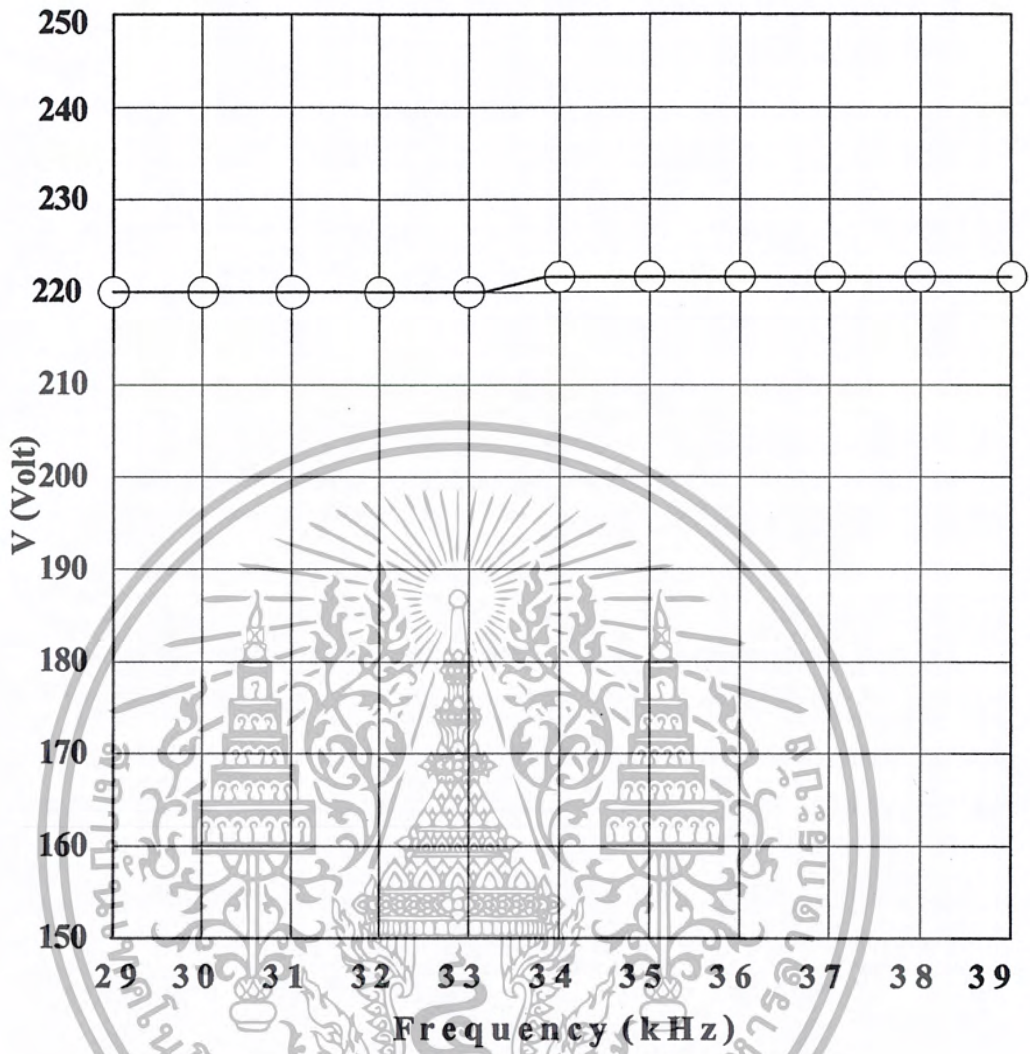
ก) สเปกตรัมของแรงดัน



ข) สเปกตรัมของกระแส

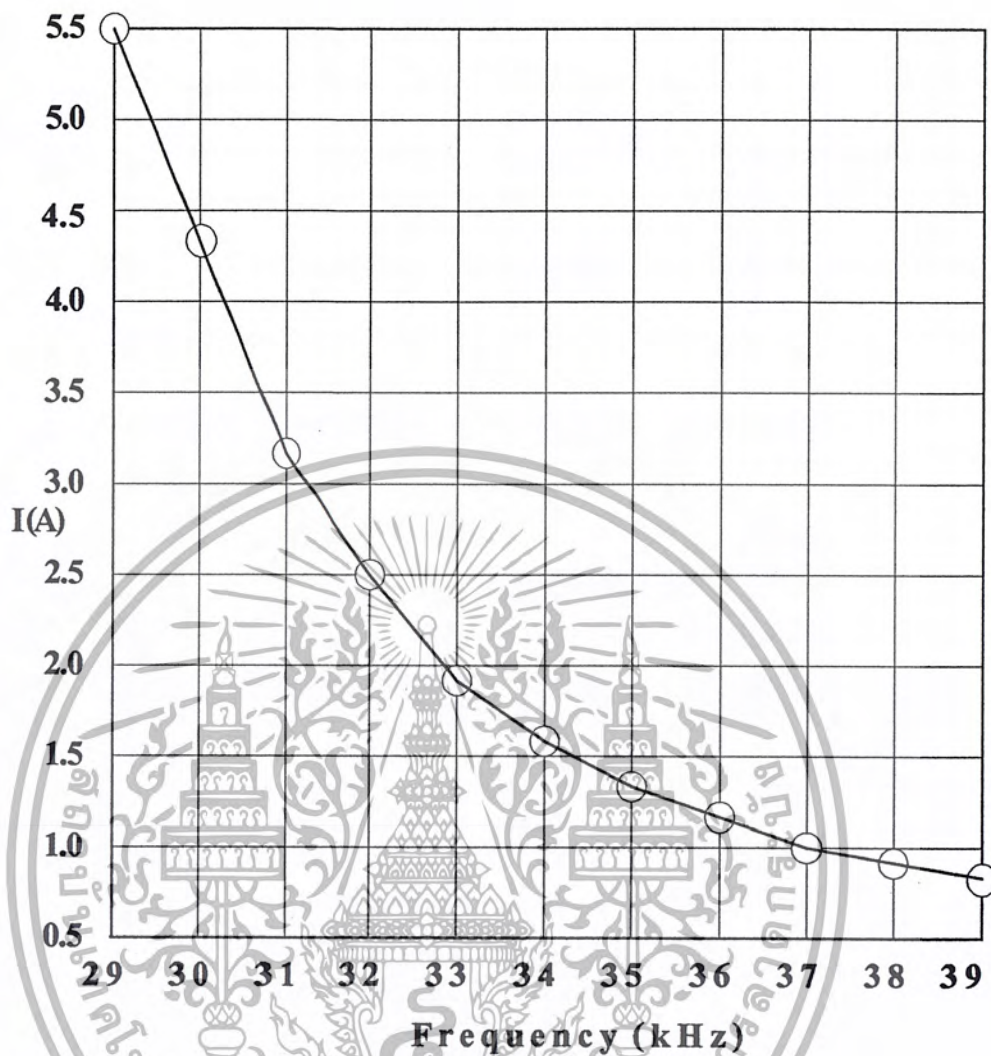
รูปที่ 7-2 สเปกตรัมของแรงดัน Vs และกระแส Is ของแหล่งจ่ายการไฟฟ้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



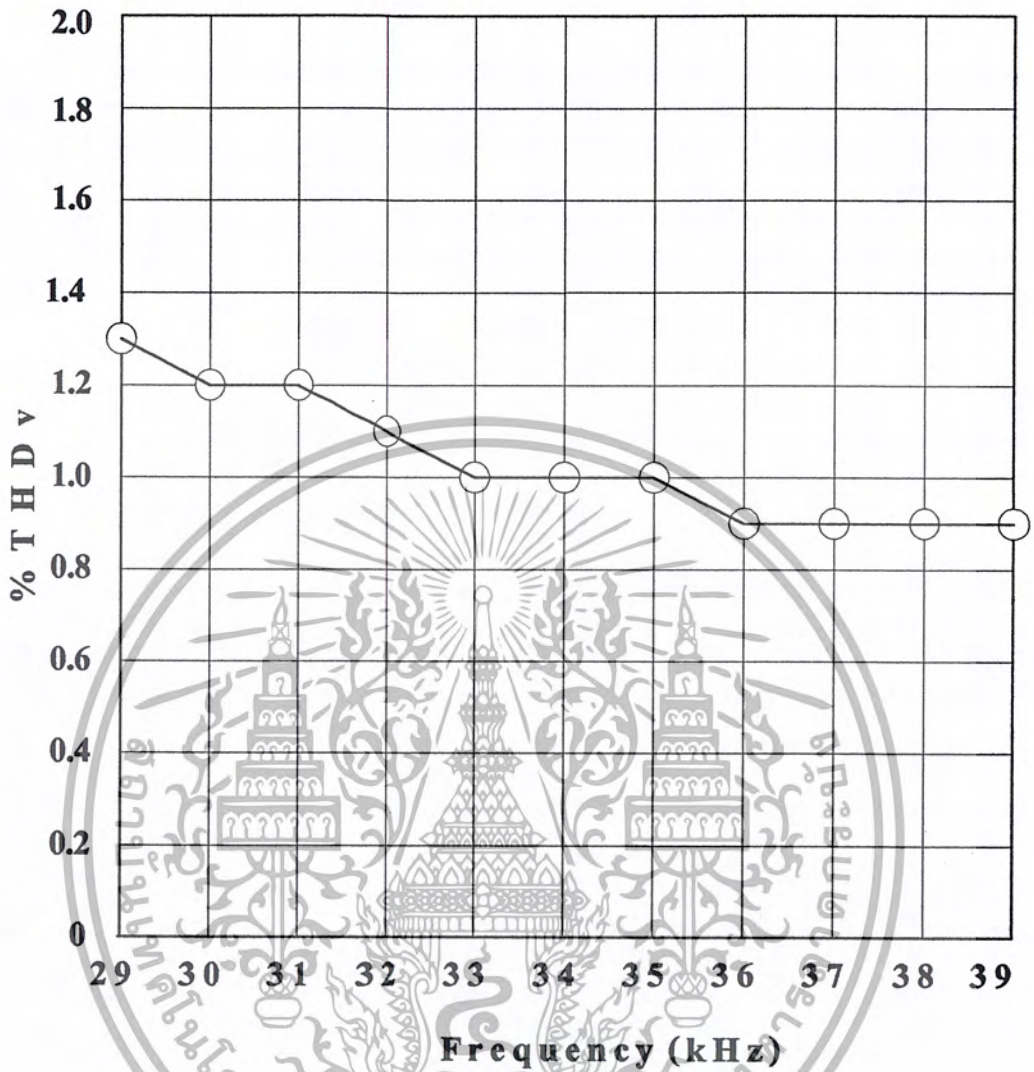
รูปที่ 7-1 แรงดันด้านแหล่งจ่ายการไฟฟ้า (V) ขณะปรับความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



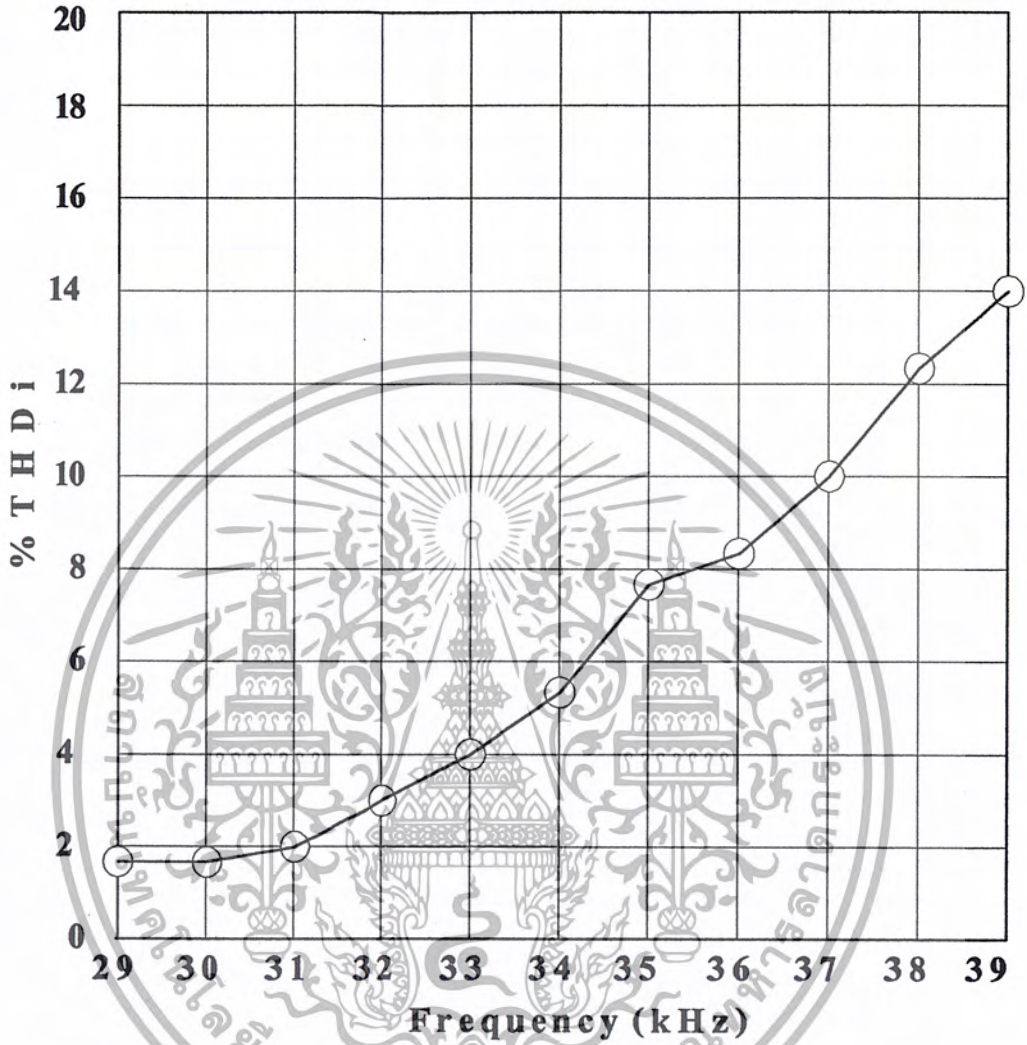
รูปที่ 7-2 กระแสต้านแหล่งจ่ายการไฟฟ้า (I) ขณะปรับความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



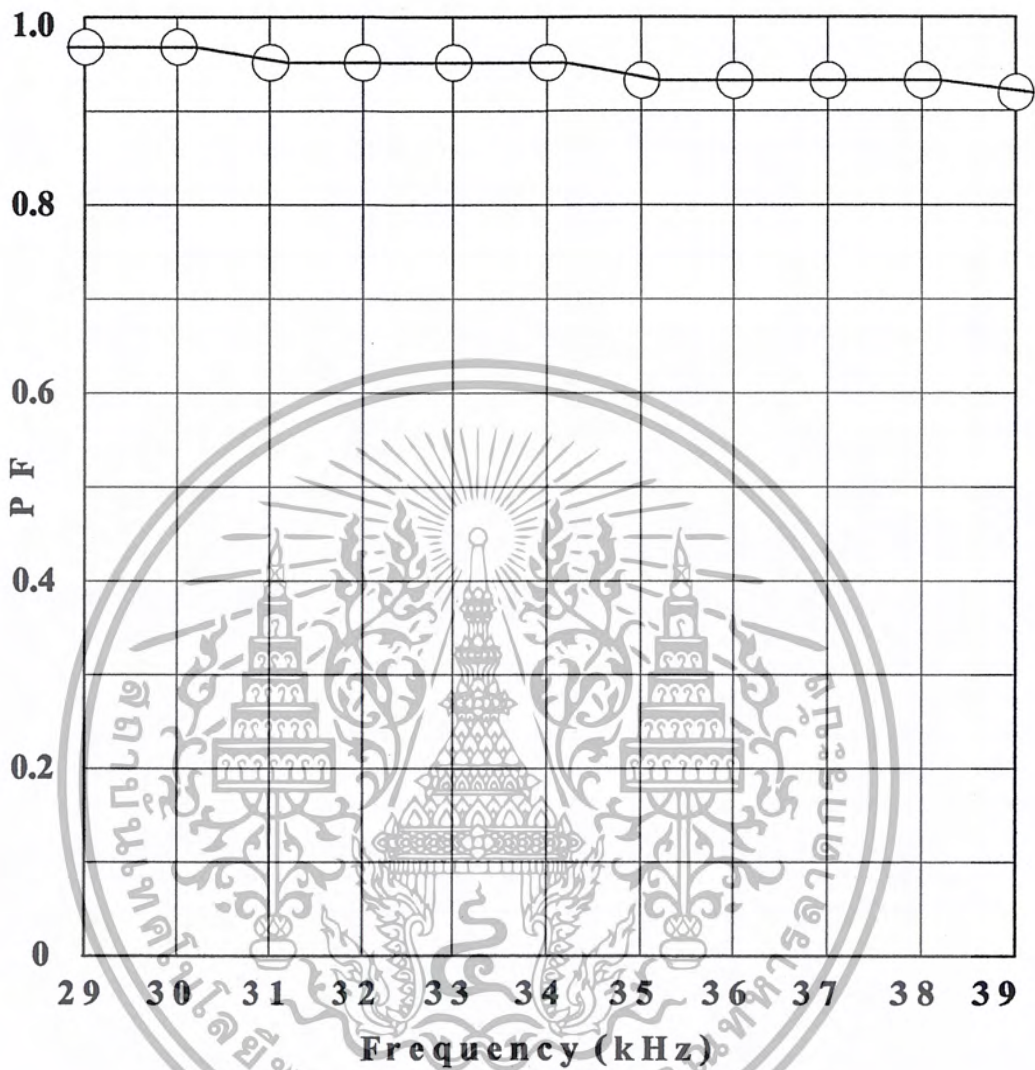
รูปที่ 7-3 ค่าความผิดเพี้ยนของแรงดันด้านแหล่งจ่ายการไฟฟ้า (% THD V) ขณะปรับความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



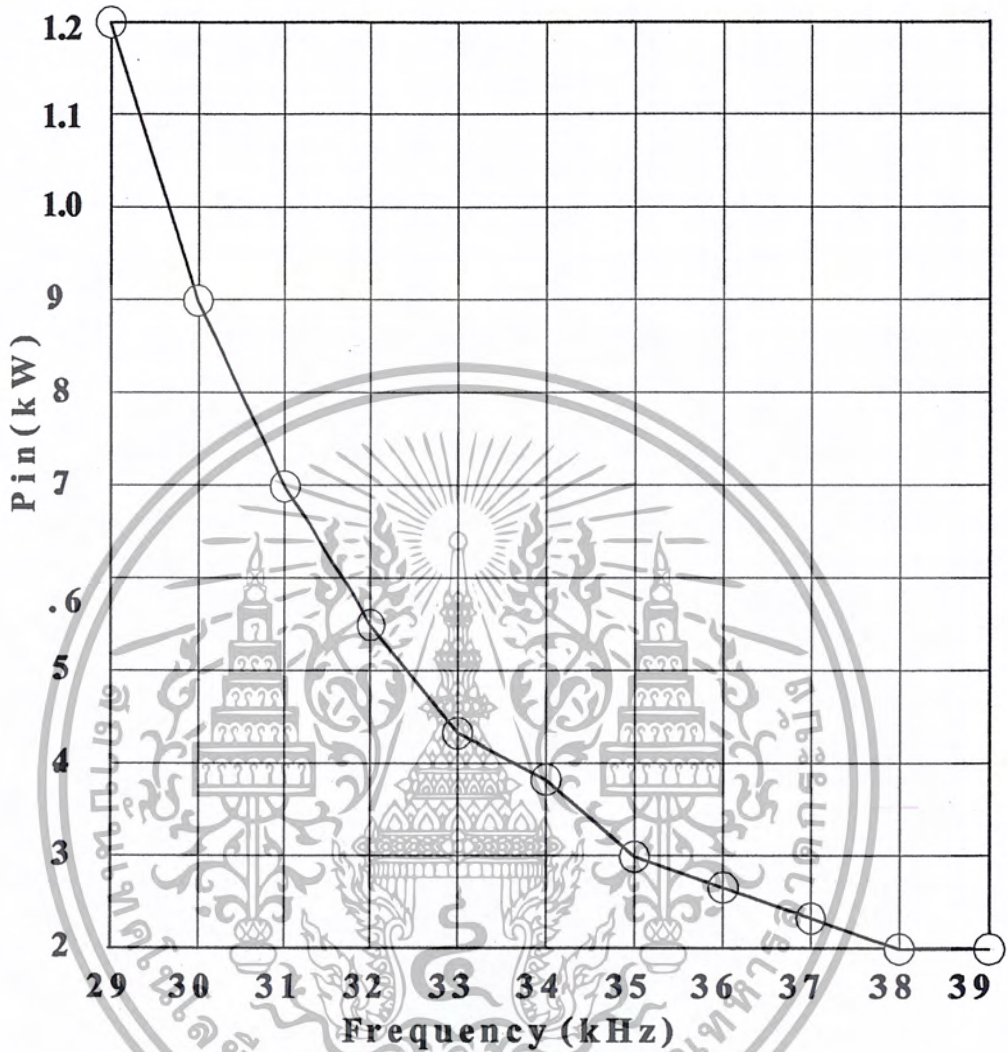
รูปที่ 7-4 ค่าความผิดเพี้ยนของกระแสด้านแหล่งจ่ายการไฟฟ้า (% THD i) ขณะปรับความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



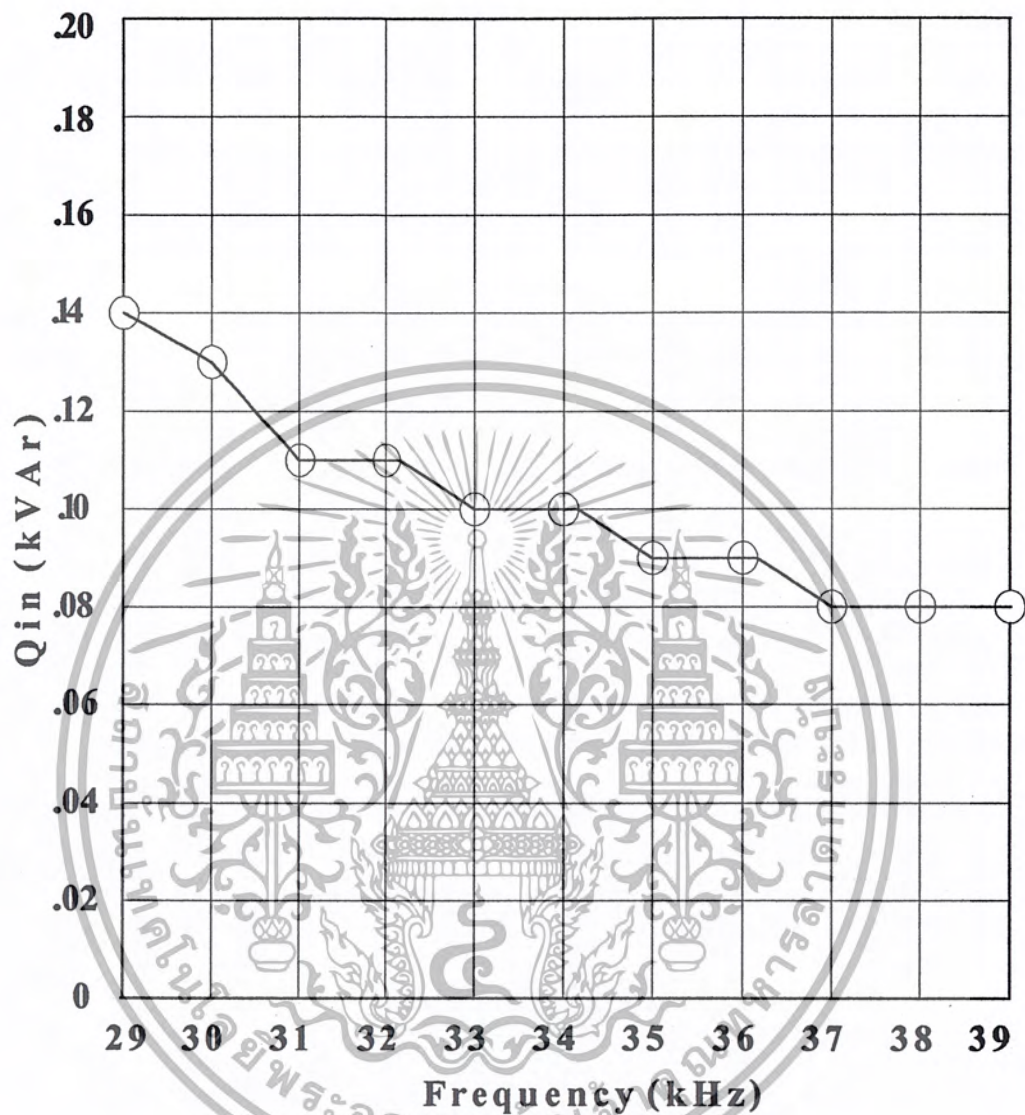
รูปที่ 7-5 เพาเวอร์แฟคเตอร์ด้านแหล่งจ่ายการไฟฟ้า (PF) ขณะปรับความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



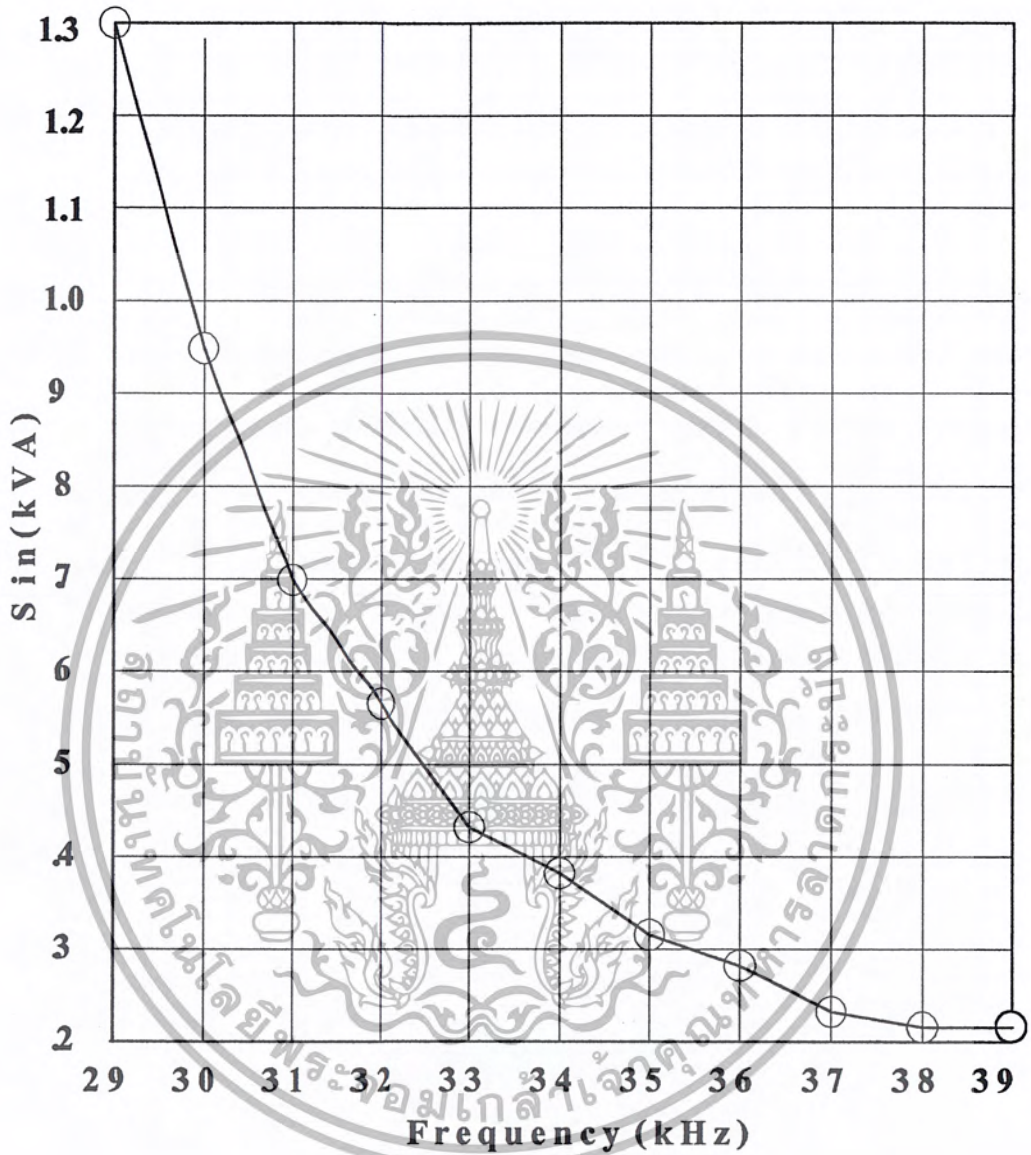
รูปที่ 7-6 กำลังไฟฟ้าอินพุตด้านแหล่งจ่ายการไฟฟ้า (P_{in}) ขณะปรับความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



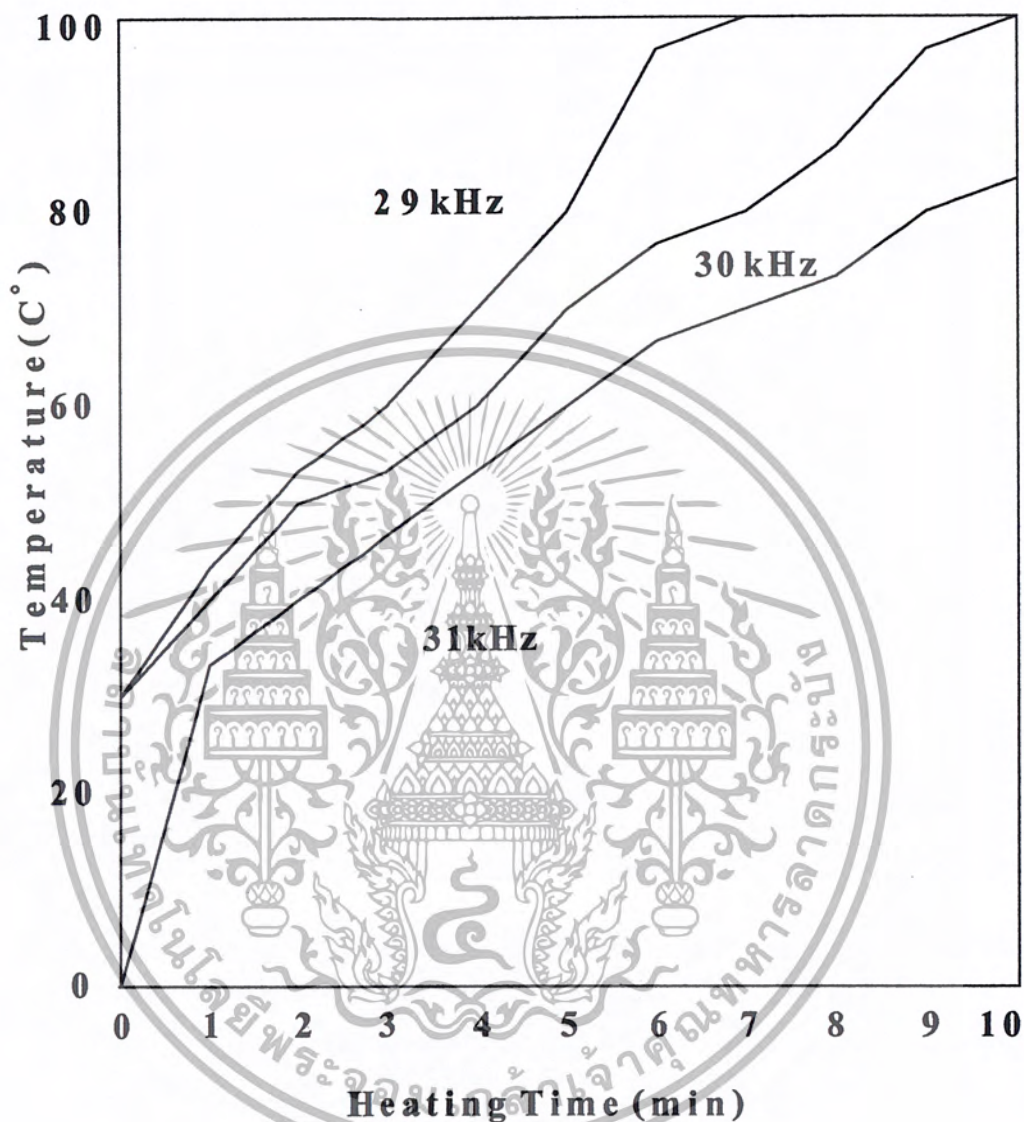
รูปที่ 7-7 กำลังไฟฟ้ารีแอกแตนซ์ด้านแหล่งจ่ายการไฟฟ้า (Q) ขณะปรับความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



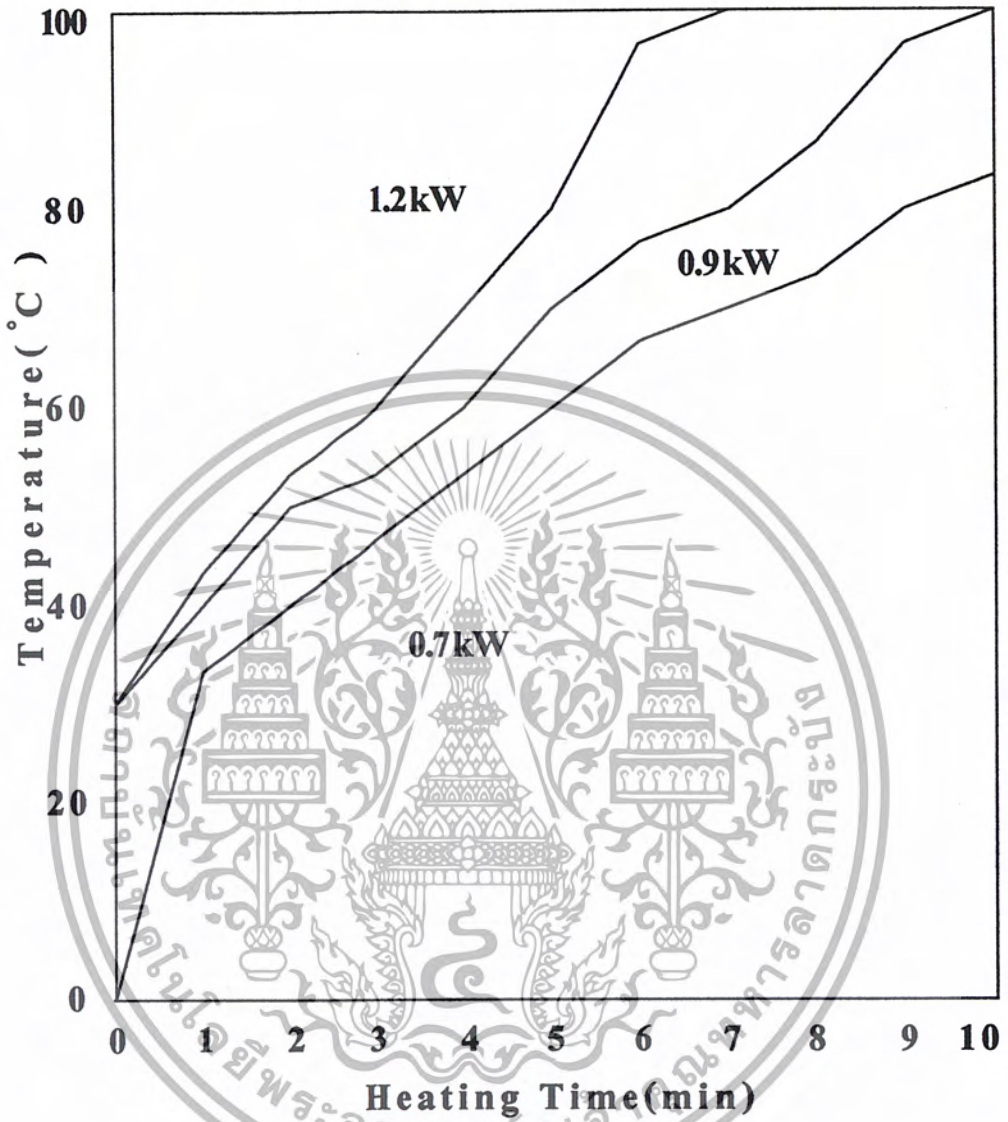
รูปที่ 7-8 กำลังไฟฟ้าอินพุตปรากฏด้านแหล่งจ่ายการไฟฟ้า (S) ขณะปรับความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7-9 อุณหภูมิของน้ำปริมาณ 1 ลิตร ขณะปรับความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7-10 อุณหภูมิของน้ำปริมาณ 1 ลิตร เมื่อปรับความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

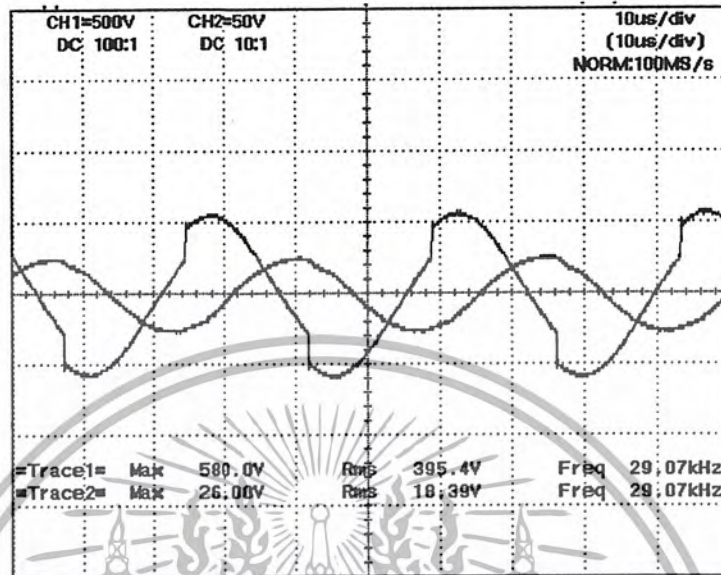
Frequency	kW	kVA	kVAR	PF	%THDI	%THDV	I	V
29	1.28	1.3	0.14	0.99	1.3	1.3	5.9	219
30	0.9	0.96	0.13	0.99	1.8	1.2	4.8	219
31	0.7	0.72	0.11	0.99	2.2	1.2	3.14	219
32	0.56	0.57	0.11	0.98	3.1	1.1	2.25	219
33	0.43	0.44	0.1	0.98	4.3	1	1.94	220
34	0.37	0.32	0.1	0.97	5.4	1	1.66	222
35	0.3	0.28	0.09	0.96	7.5	0.9	1.41	222
36	0.26	0.28	0.09	0.95	8.2	0.9	1.25	222
37	0.22	0.24	0.08	0.94	10.2	0.9	1.23	222
38	0.2	0.21	0.08	0.93	12.6	0.9	0.97	222
39	0.19	0.2	0.08	0.92	14	0.9	0.96	222

ตารางที่ 7-1 แสดงค่าต่างๆ ในขณะที่ทำการปรับความถี่ในช่วงความถี่ที่ 29-39 kHz

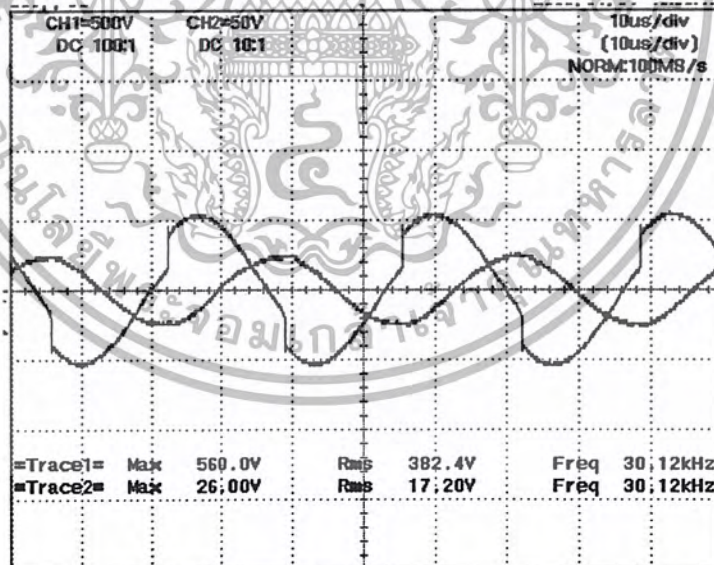
สรุปผลการทดลอง

จากการทดลองเราจะพบว่าเมื่อปรับความถี่เพิ่มขึ้น กระแสทางด้านอินพุทจะค่อยๆ ลดลง ซึ่งเมื่อกระแสลดลง กำลังไฟฟ้าอินพุทที่วัดได้จะลดลง นั่นคือเมื่อความถี่เพิ่มขึ้นขนาดกำลังไฟฟ้าอินพุทที่ได้ต่ำลง

7.2 ผลการทดลองทางด้านเอาท์พุทขณะปรับความถี่

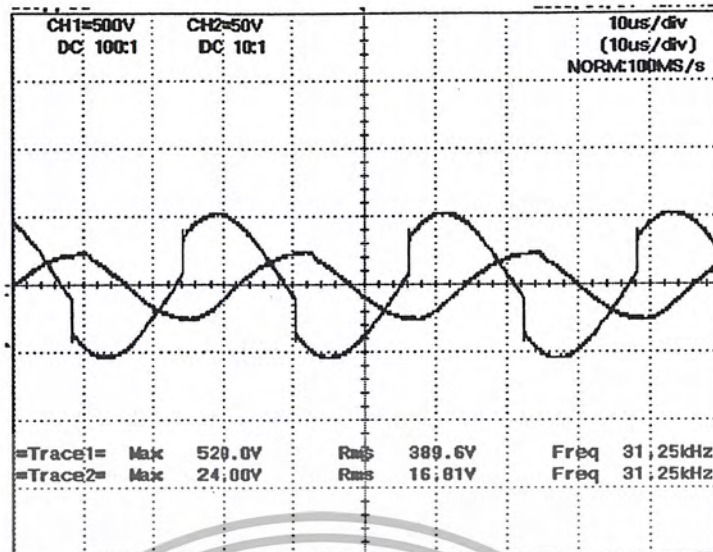


รูปที่ 7 – 10 รูปคลื่นกระแสและแรงดันที่ความถี่ 29 kHz



รูปที่ 7 – 11 รูปคลื่นกระแสและแรงดันที่ความถี่ 30 kHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 7-12 รูปคลื่นกระแสและแรงดันที่ความถี่ 31 kHz

ความถี่ (kHz)	V max (V)	V rms (V)	I rms (A)
29	580	395.4	18.3
30	560	382.4	17.2
31	529	389.6	16.81

ตารางที่ 7-2 ผลการทดลองขณะเปลี่ยนความถี่

สรุปผลการทดลอง

จากการทดลองทำให้เราพบว่าเมื่อความถี่เพิ่มขึ้น กระแสไหลจะค่อยๆ ลดลง เนื่องจากความถี่ห่างจากจุดรีโซแนนซ์ออกไปเรื่อยๆ และแรงดันก็จะลดลงเช่นกัน จากรูปที่ 7-10 ถึงรูปที่ 7-13 ความถี่ที่ใช้ยังมากกว่าความถี่รีโซแนนซ์

บทที่ 8 บทวิจารณ์และสรุป

จากโครงการนี้ได้ทำการสร้างเตาหุงต้มเหนียวน้ำ ทำงานที่ความถี่ 20 – 40 kHz. โดยมีระดับแรงดันไฟฟ้าอินพุท จากการไฟฟ้า 220 V. พิกัดกำลังทางค่านอินพุท 1.2 kW. ที่ความถี่ของสวิทซ์ซิ่ง 29 kHz. ความถี่ที่ใช้ในการสวิทซ์ จะถูกกำหนดโดยความถี่รีโซแนนซ์ แต่ในทางปฏิบัติจุดการทำงานจะเลือกให้ความถี่ในการสวิทซ์สูงกว่าความถี่รีโซแนนซ์เล็กน้อย ซึ่งจะทำให้กระแสไหลลดล้าหลังแรงดันเอาต์พุท โครงการนี้ได้ทำการแก้เพาเวอร์แฟคเตอร์ทางค่านอินพุท โดยการออกแบบสามารถทำให้ได้เพาเวอร์แฟคเตอร์ เท่ากับ 0.99 lagging และยังมีระบบการควบคุมกระแสทางค่านอินพุทและเอาต์พุทอีกด้วย

8.1 จุดเด่นของโครงการ

1. ขนาดของเตาหุงต้มมีขนาดเล็กกระทัดรัด ดูแลรักษาง่าย และปลอดภัย
2. มีการสูญเสียพลังงานน้อย เพราะเกิดความร้อนที่ก้นภาชนะ โดยตรง
3. มีค่าเพาเวอร์แฟคเตอร์ที่ดีมาก
4. ประหยัดพลังงาน

8.2 ปัญหาที่พบและข้อเสนอแนะเพื่อพัฒนา

1. เนื่องจากไมโครคอนโทรลเลอร์ที่ใช้ในการสร้างสัญญาณประมวลผลจากการควบคุมกระแส มีความไวต่ออุปกรณ์สวิทซ์ซิ่ง ดังนั้นเมื่อเตาหุงต้มเหนียวน้ำความถี่สูงทำงาน จะมีสัญญาณความถี่สูงเข้ามารบกวน (NOISE) ทำให้ไมโครคอนโทรลเลอร์ทำการประมวลผลผิดพลาด
2. โครงการนี้ใช้ขดลวดขนาดเบอร์ 30 จำนวน 16 เส้นพันตึกเกลียวรวมกัน แล้วนำมาขดเป็นรูปก้นหอยแบนจำนวน 14 รอบ และใช้ตัวเก็บประจุทางค่านไหลดต่อในลักษณะ C/2 โดยเลือกตัวเก็บประจุชนิดไมล้า ขนาด 0.1 μF 600 V. เมื่อทำการทดสอบให้เตาหุงต้มเหนียวน้ำความถี่สูงทำงานที่พิกัด 1.2 kW. ปรากฏว่าตัวเก็บประจุแบบที่ใช้ไม่ได้สามารถทนกระแสได้ ทำให้ตัวเก็บประจुरะเบิดขึ้น ดังนั้นจึงเปลี่ยนมาใช้ตัวเก็บประจุแบบ WIMA ขนาด 0.047 μF 1000 V. นำมาต่อแบบขนานกัน จำนวน 27 ตัว ซึ่งเมื่อเปลี่ยนมาใช้ตัวเก็บประจุแบบ WIMA แล้วเตาหุงต้มเหนียวน้ำความถี่สูง สามารถที่จะทำงานได้ตามพิกัดกำลังที่ออกแบบไว้
3. โครงการนี้ได้ใช้หลักการเหนียวน้ำความถี่สูง โดยใช้วงจร HALF BRIDGE INVERTERS เป็นตัวจ่ายแรงดันความถี่สูงให้กับไหลดตัวเหนียวน้ำและตัวเก็บประจุ โดยสัญญาณความถี่สูงจะถูกสร้างโดยไมโครคอนโทรลเลอร์ PIC18F458 คิวตี้ 50%กลับสัญญาณกันสองชุดนำสัญญาณที่ได้ผ่านวงจรแยกเคทไทยม์ 2.5 μS . เพื่อจ่ายให้ชุดขับสัญญาณ (Drive Circuit)

ป้องกันการ Short กันของสวิทซ์ จึงควรออกแบบให้เหมาะสม เพราะถ้าชุดขับสัญญาณ

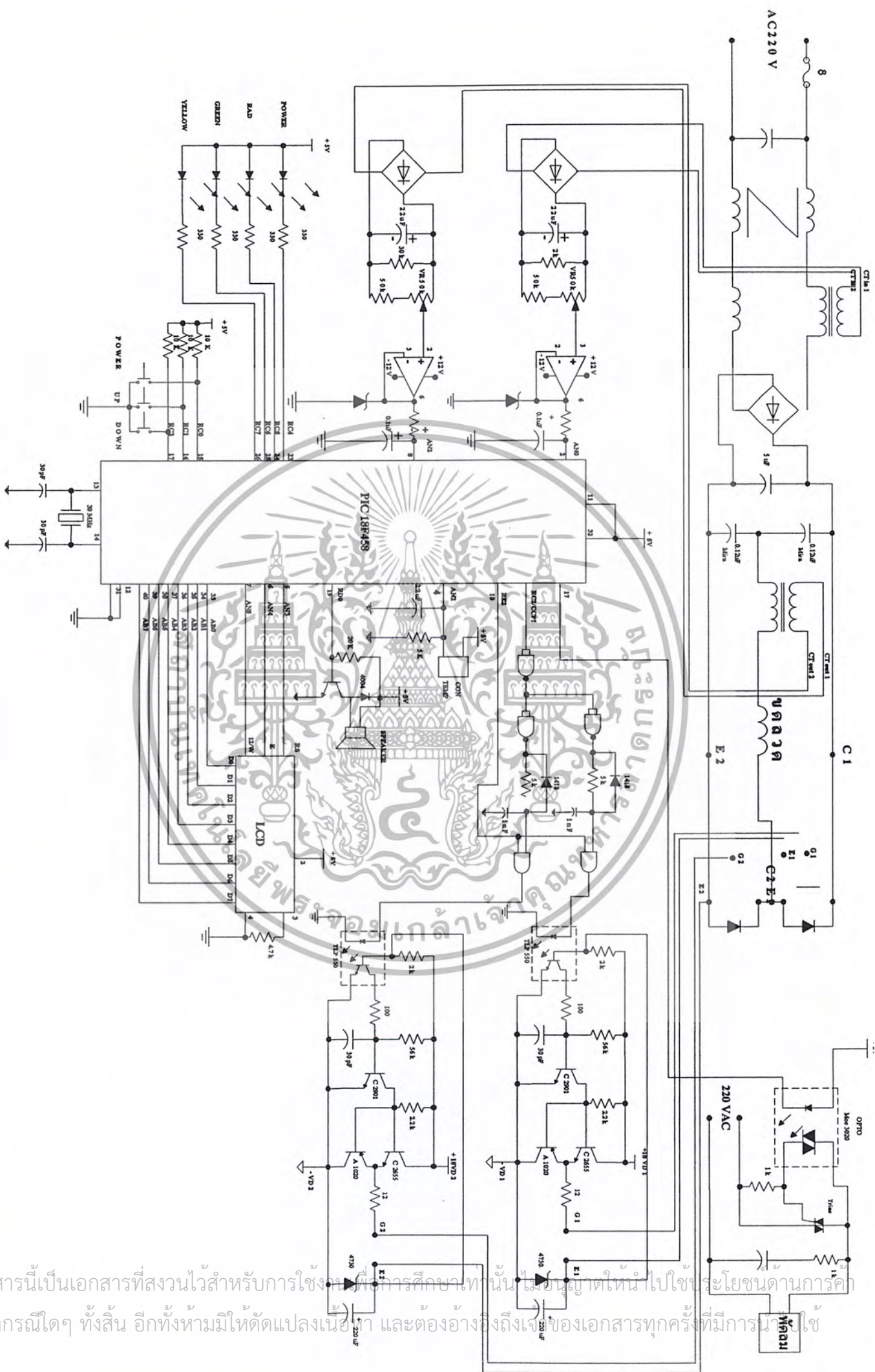
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อนำไปเผยแพร่โดยไม่ได้รับอนุญาตเป็นการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทำงานได้ไม่ดี อาจเป็นที่มาของการ Short ของสวิทช์ได้ ซึ่งโครงการนี้ได้เลือกใช้ IGBT เป็นอุปกรณ์สวิทช์ ซึ่งมีราคาแพงมาก

4. จากการทดลองพบว่า ถ้าใช้ตัวเก็บประจุที่กรองแรงดันมาก จะทำให้รีปเปิลมีค่าน้อยมีลักษณะแรงดันลงที่พาเวอร์เอาต์พุตสูง แต่ค่าพาเวอร์แฟคเตอร์ด้านอินพุตจะต่ำ แต่เมื่อทำการเปลี่ยนค่าตัวเก็บประจุ ให้มีค่าน้อยลง จะทำให้เกิดรีปเปิลมากมีลักษณะแรงดันไม่คงที่ที่พาเวอร์เอาต์พุตน้อยลง แต่ค่าพาเวอร์แฟคเตอร์ทางด้านอินพุตจะมีค่าสูงขึ้น ดังนั้นจึงใช้เป็นข้อเปรียบเทียบได้ถึงข้อดีข้อเสียในจุดนี้ได้

8.3 ประโยชน์ที่ได้รับ

1. สามารถจัดทำเตาหุงต้มเหนียวนำความถี่สูง ที่ประหยัดพลังงาน และมีประสิทธิภาพสูง เหมาะสำหรับนำไปใช้ในครัวเรือน หรือธุรกิจต่างๆ ได้ เพราะมีความปลอดภัยสูงด้วย
2. สามารถทำให้เข้าใจถึงระบบการทำงานต่างๆ ของเตา เพื่อนำไปสู่การประยุกต์ใช้กับงานอื่นๆ ได้ เช่น เครื่องเชื่อมไฟฟ้ากระแสตรง เตาอบเหนียวนำและเครื่องหุงแป้ง โลหะ เป็นต้น
3. โครงการนี้ได้เลือกใช้ ไอจีบีที (IGBT) เป็นอุปกรณ์สวิทช์ซึ่ง ซึ่งในโครงการอื่นได้มีการเลือกใช้มอสเฟต เป็นอุปกรณ์สวิทช์ซึ่ง จะเห็นได้ว่าในแต่ละครั้งจะมีการเลือกใช้อุปกรณ์สวิทช์ซึ่งที่มีการพัฒนาขึ้นเรื่อยๆ ดังนั้นโครงการนี้จึงเป็นข้อมูลที่สำคัญ ที่เป็นต้นแบบของการใช้อุปกรณ์สวิทช์ซึ่งชนิดหนึ่ง เพื่อการวิจัยและพัฒนาต่อไป
4. สามารถออกแบบผลิตภัณฑ์ที่ประหยัดพลังงาน เพื่อเป็นทางเลือกอีกทางหนึ่งในปัจจุบัน ที่มวลดมนุษย์กำลังแสวงหาพลังงานทดแทนมาสนองความต้องการของมนุษย์ ที่มีอยู่มากมาย จนเกินปริมาณที่พลังงานมีอยู่บนโลก
5. โครงการนี้ได้ฝึกให้คณะผู้จัดทำ มีความวิริยะ อุตสาหะ อดทน และความพยายาม ที่จะมุ่งไปสู่ความสำเร็จจุดต่างลงด้วยดี เพื่อเป็นการฝึกตนให้เป็นคนดีก่อนที่จะออกไปสู่การทำงาน และการอยู่ร่วมกันในสังคมต่อไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้ภายในของนักศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. ผศ. ถิรวัฒน์ ประกอบผล, “ การพัฒนาไมโครคอนโทรลเลอร์ด้วยภาษาซี “, สำนักพิมพ์ ศ.ส.ท
2. “ เรียนรู้การปฏิบัติการ ไมโครคอนโทรลเลอร์ PIC18FXXX “, บริษัท อินโนวทีฟ เอ็กเพอริเมนต์ จำกัด
3. สติธิโชค สินรัตน์, “การวิเคราะห์เตาหุงต้มเหนียวนำความถี่สูงชนิดควบคุมกำลังไฟฟ้าด้วยความถี่”, ภาควิชาวิศวกรรมไฟฟ้า สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบังลาดกระบัง



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

High Performance, 28/40-Pin Enhanced FLASH Microcontrollers with CAN

High Performance RISC CPU:

- Linear program memory addressing up to 2 Mbytes
- Linear data memory addressing to 4 Kbytes
- Up to 10 MIPS operation
- DC - 40 MHz clock input
- 4 MHz - 10 MHz osc./clock input with PLL active
- 16-bit wide instructions, 8-bit wide data path
- Priority levels for interrupts
- 8 x 8 Single Cycle Hardware Multiplier

Peripheral Features:

- High current sink/source 25 mA/25 mA
- Three external interrupt pins
- Timer0 module: 8-bit/16-bit timer/counter with 8-bit programmable prescaler
- Timer1 module: 16-bit timer/counter
- Timer2 module: 8-bit timer/counter with 8-bit period register (time-base for PWM)
- Timer3 module: 16-bit timer/counter
- Secondary oscillator clock option - Timer1/Timer3
- Capture/Compare/PWM (CCP) modules CCP pins can be configured as:
 - Capture input: 16-bit, max resolution 6.25 ns
 - Compare: 16-bit, max resolution 100 ns (T_{cy})
 - PWM output: PWM resolution is 1- to 10-bit
Max. PWM freq. @:8-bit resolution = 156 kHz
10-bit resolution = 39 kHz
- Enhanced CCP module which has all the features of the standard CCP module, but also has the following features for advanced motor control:
 - 1, 2, or 4 PWM outputs
 - Selectable PWM polarity
 - Programmable PWM deadtime
- Master Synchronous Serial Port (MSSP) with two modes of operation:
 - 3-wire SPI™ (Supports all 4 SPI modes)
 - I²C™ Master and Slave mode
- Addressable USART module: Supports Interrupt on Address bit

Advanced Analog Features:

- 10-bit, up to 8-channel Analog-to-Digital Converter module (A/D) with:
 - Conversion available during SLEEP
 - Up to 8 channels available
- Analog Comparator Module:
 - Programmable input and output multiplexing
- Comparator Voltage Reference Module
- Programmable Low Voltage Detection (LVD) module
 - Supports interrupt on low voltage detection
- Programmable Brown-out Reset (BOR)

CAN bus Module Features:

- Message bit rates up to 1 Mbps
- Conforms to CAN 2.0B ACTIVE Spec with:
 - 29-bit Identifier Fields
 - 8-byte message length
 - 3 Transmit Message Buffers with prioritization
 - 2 Receive Message Buffers
 - 6 full 29-bit Acceptance Filters
 - Prioritization of Acceptance Filters
 - Multiple Receive Buffers for High Priority Messages to prevent loss due to overflow
 - Advanced Error Management Features

Special Microcontroller Features:

- Power-on Reset (POR), Power-up Timer (PWRT), and Oscillator Start-up Timer (OST)
- Watchdog Timer (WDT) with its own on-chip RC oscillator
- Programmable code protection
- Power saving SLEEP mode
- Selectable oscillator options, including:
 - 4X Phase Lock Loop (of primary oscillator)
 - Secondary Oscillator (32 kHz) clock input
- In-Circuit Serial Programming™ (ICSP™) via two pins

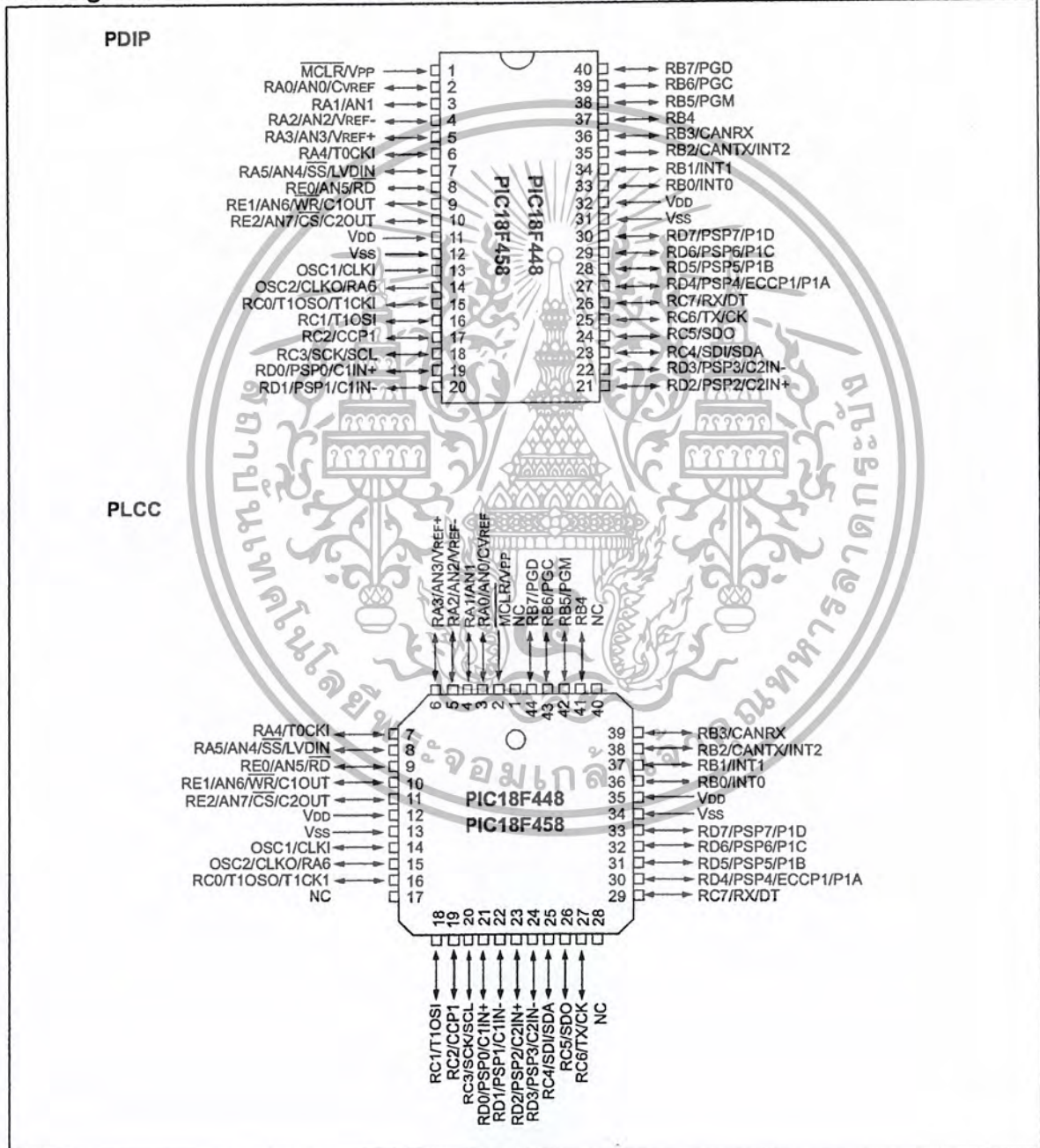
FLASH Technology:

- Low power, high speed Enhanced FLASH technology
- Fully static design
- Wide operating voltage range (2.0V to 5.5V)
- Industrial and Extended temperature ranges

PIC18FXX8

Device	Program Memory		Data Memory		I/O	10-bit A/D (ch)	Comparators	CCP/ ECCP (PWM)	MSSP		USART	Timers 8/16-bit
	FLASH (bytes)	# Single Word Instructions	SRAM (bytes)	EEPROM (bytes)					SPI	Master I ² C		
PIC18F248	16K	8192	768	256	22	5	—	1/0	Y	Y	Y	1/3
PIC18F258	32K	16384	1536	256	22	5	—	1/0	Y	Y	Y	1/3
PIC18F448	16K	8192	768	256	33	8	2	1/1	Y	Y	Y	1/3
PIC18F458	32K	16384	1536	256	33	8	2	1/1	Y	Y	Y	1/3

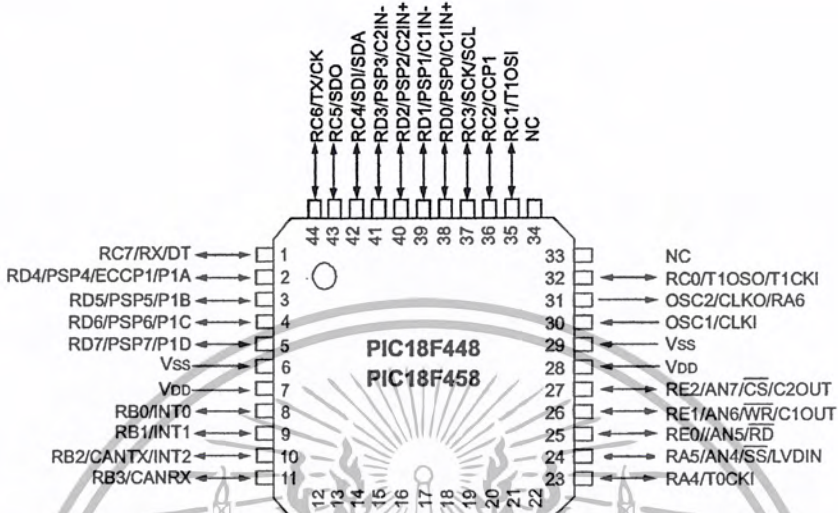
Pin Diagrams



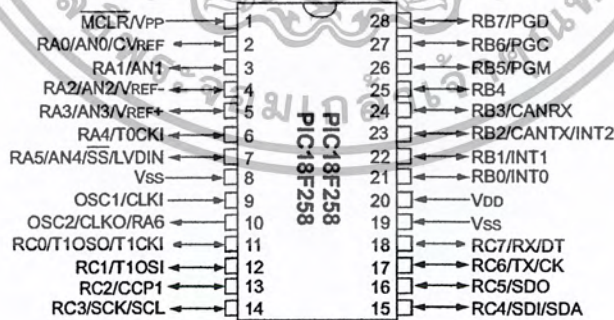
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Diagrams (Continued)

TQFP



SPDIP, SOIC



1.0 DEVICE OVERVIEW

This document contains device specific information for the following devices:

1. PIC18F248
2. PIC18F258
3. PIC18F448
4. PIC18F458

These devices are available in 28-pin, 40-pin and 44-pin packages. They are differentiated from each other in four ways:

1. PIC18FX58 devices have twice the FLASH program memory and data RAM of PIC18FX48 devices (32 Kbytes and 1536 bytes vs. 16 Kbytes and 768 bytes, respectively).

2. PIC18F2X8 devices implement 5 A/D channels, as opposed to 8 for PIC18F4X8 devices.
3. PIC18F2X8 devices implement 3 I/O ports, while PIC18F4X8 devices implement 5.
4. Only PIC18F4X8 devices implement the Enhanced CCP module, analog comparators and the Parallel Slave Port.

All other features for devices in the PIC18FXX8 family, including the serial communications modules, are identical. These are summarized in Table 1-1.

Block diagrams of the PIC18F2X8 and PIC18F4X8 devices are provided in Figure 1-1 and Figure 1-2, respectively. The pinouts for these device families are listed in Table 1-2.

TABLE 1-1: PIC18FXX8 DEVICE FEATURES

Features	PIC18F248	PIC18F258	PIC18F448	PIC18F458
Operating Frequency	DC - 40 MHz	DC - 40 MHz	DC - 40 MHz	DC - 40 MHz
Internal Program Memory	Bytes	16K	32K	16K
	# of Single Word Instructions	8192	16384	8192
Data Memory (Bytes)	768	1536	768	1536
Data EEPROM Memory (Bytes)	256	256	256	256
Interrupt Sources	17	17	21	21
I/O Ports	Ports A, B, C	Ports A, B, C	Ports A, B, C, D, E	Ports A, B, C, D, E
Timers	4	4	4	4
Capture/Compare/PWM Modules	1	1	1	1
Enhanced Capture/Compare/PWM Modules	—	—	1	1
Serial Communications	MSSP, CAN, Addressable USART	MSSP, CAN, Addressable USART	MSSP, CAN, Addressable USART	MSSP, CAN, Addressable USART
Parallel Communications (PSP)	No	No	Yes	Yes
10-bit Analog-to-Digital Converter	5 input channels	5 input channels	8 input channels	8 input channels
Analog Comparators	No	No	2	2
Analog Comparators VREF Output	N/A	N/A	Yes	Yes
RESETS (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow (PWRT, OST)
Programmable Low Voltage Detect	Yes	Yes	Yes	Yes
Programmable Brown-out Reset	Yes	Yes	Yes	Yes
CAN Module	Yes	Yes	Yes	Yes
In-Circuit Serial Programming™ (ICSP™)	Yes	Yes	Yes	Yes
Instruction Set	75 Instructions	75 Instructions	75 Instructions	75 Instructions
Packages	28-pin SPDIP 28-pin SOIC	28-pin SPDIP 28-pin SOIC	40-pin PDIP 44-pin PLCC 44-pin TQFP	40-pin PDIP 44-pin PLCC 44-pin TQFP

PIC18FXX8

FIGURE 1-1: PIC18F248/258 BLOCK DIAGRAM

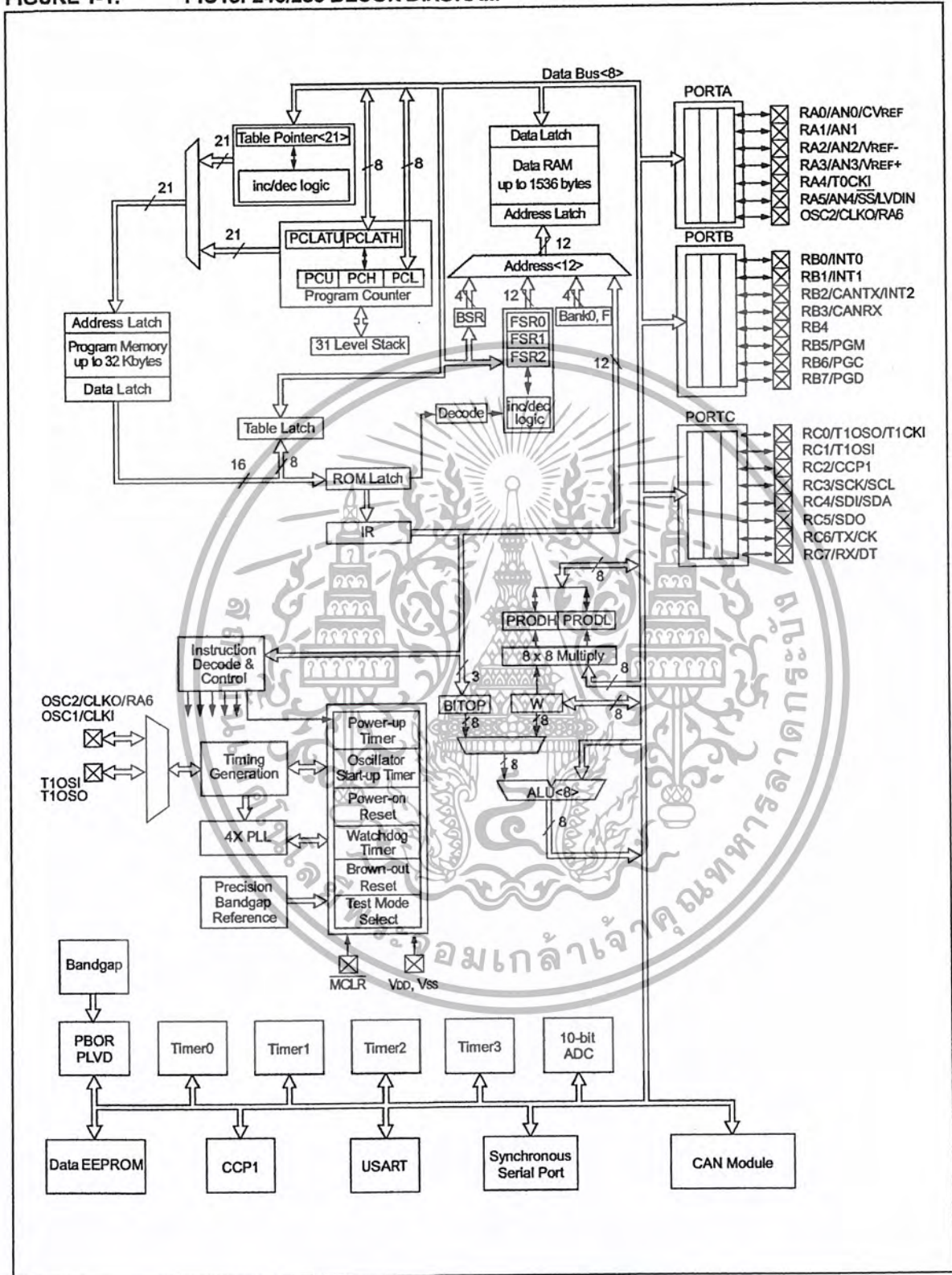
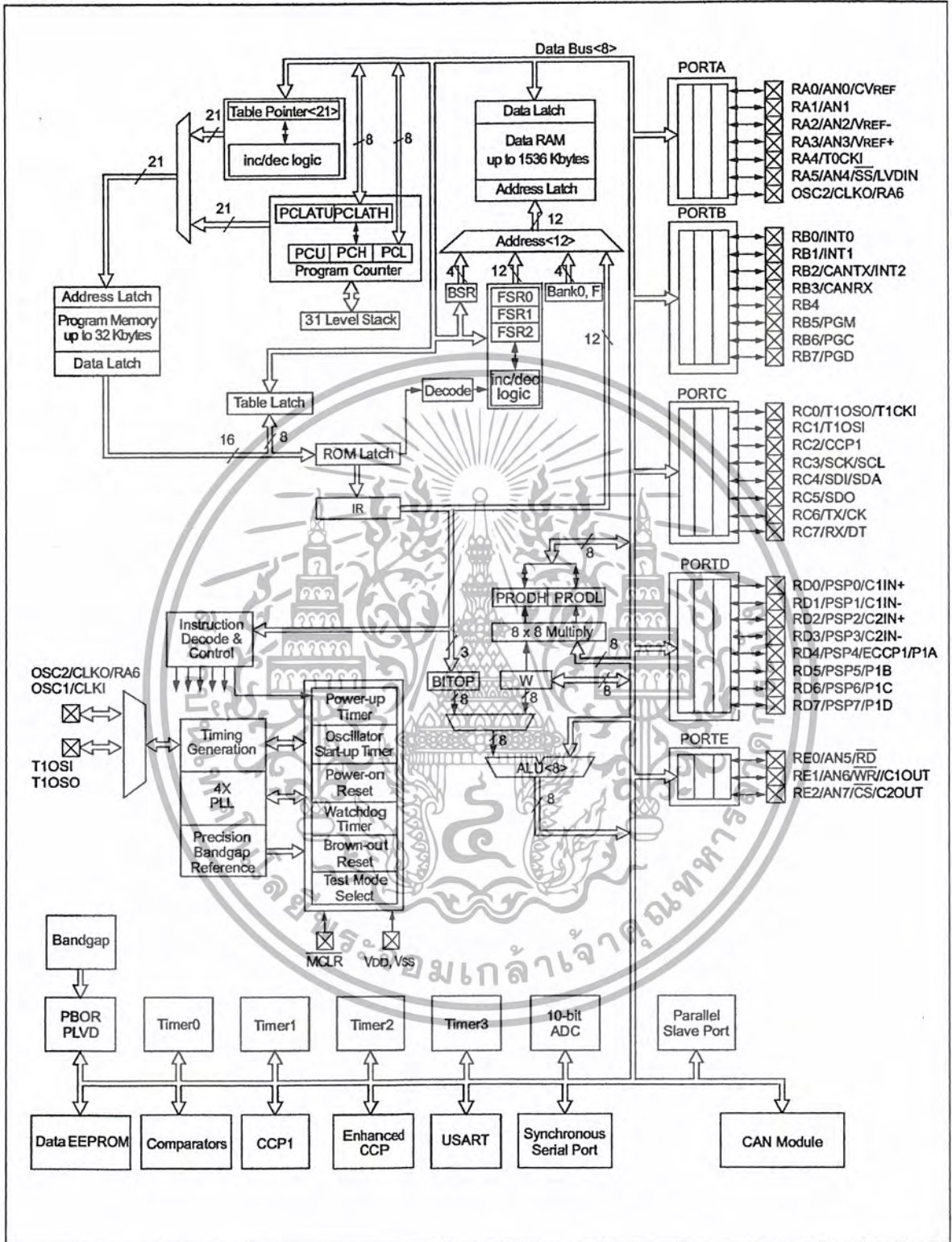


FIGURE 1-2: PIC18F448/458 BLOCK DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIC18FXX8

TABLE 1-2: PIC18FXX8 PINOUT I/O DESCRIPTIONS

Pin Name	Pin Number				Pin Type	Buffer Type	Description
	PIC18F248/258	PIC18F448/458					
	SPDIP, SOIC	PDIP	TQFP	PLCC			
MCLR/VPP MCLR VPP	1	1	18	2	I P	ST —	Master Clear (input) or programming voltage (output). Master Clear (Reset) input. This pin is an active low RESET to the device. Programming voltage input.
NC	—	—	12, 13, 33, 34	1, 17, 28, 40	—	—	These pins should be left unconnected.
OSC1/CLKI OSC1 CLKI	9	13	30	14	I I	CMOS/ST CMOS	Oscillator crystal or external clock input. Oscillator crystal input or external clock source input. ST buffer when configured in RC mode. Otherwise CMOS. External clock source input. Always associated with pin function OSC1 (see OSC1/CLKI, OSC2/CLKO pins).
OSC2/CLKO/RA6 OSC2 CLKO RA6	10	14	31	15	O O I/O	— — TTL	Oscillator crystal or clock output. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In RC mode, OSC2 pin outputs CLKO, which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate. General purpose I/O pin.

Legend: TTL = TTL compatible input CMOS = CMOS compatible input or output
 ST = Schmitt Trigger input with CMOS levels Analog = Analog input
 I = Input O = Output
 P = Power OD = Open Drain (no P diode to VDD)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TABLE 1-2: PIC18FXX8 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number				Pin Type	Buffer Type	Description
	PIC18F248/258		PIC18F448/458				
	SPDIP, SOIC	PDIP	TQFP	PLCC			
RA0/AN0/CVREF RA0 AN0 CVREF	2	2	19	3	I/O I O	TTL Analog Analog	PORTA is a bi-directional I/O port. Digital I/O. Analog input 0. Comparator voltage reference output.
RA1/AN1 RA1 AN1	3	3	20	4	I/O I	TTL Analog	Digital I/O. Analog input 1.
RA2/AN2/VREF- RA2 AN2 VREF-	4	4	21	5	I/O I I	TTL Analog Analog	Digital I/O. Analog input 2. A/D reference voltage (Low) input.
RA3/AN3/VREF+ RA3 AN3 VREF+	5	5	22	6	I/O I I	TTL Analog Analog	Digital I/O. Analog input 3. A/D reference voltage (High) input.
RA4/T0CKI RA4 T0CKI	6	6	23	7	I/O I	TTL/OD ST	Digital I/O - open drain when configured as output. Timer0 external clock input.
RA5/AN4/SS/LVDIN RA5 AN4 SS LVDIN	7	7	24	8	I/O I I I	TTL Analog ST Analog	Digital I/O. Analog input 4. SPI slave select input. Low voltage detect input.
RA6							See the OSC2/CLKO/RA6 pin.

Legend: TTL = TTL compatible input CMOS = CMOS compatible input or output
 ST = Schmitt Trigger input with CMOS levels Analog = Analog input
 I = Input O = Output
 P = Power OD = Open Drain (no P diode to VDD)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIC18FXX8

TABLE 1-2: PIC18FXX8 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number				Pin Type	Buffer Type	Description
	PIC18F248/258	PIC18F448/458					
	SPDIP, SOIC	PDIP	TQFP	PLCC			
RB0/INT0 RB0 INT0	21	33	8	36	I/O I	TTL ST	PORTB is a bi-directional I/O port. PORTB can be software programmed for internal weak pull-ups on all inputs. Digital I/O. External interrupt 0.
RB1/INT1 RB1 INT1	22	34	9	37	I/O I	TTL ST	Digital I/O. External interrupt 1.
RB2/CANTX/INT2 RB2 CANTX INT2	23	35	10	38	I/O O I	TTL TTL ST	Digital I/O. Transmit signal for CAN bus. External interrupt 2.
RB3/CANRX RB3 CANRX	24	36	11	39	I/O I	TTL TTL	Digital I/O. Receive signal for CAN bus.
RB4	25	37	14	41	I/O	TTL	Digital I/O. Interrupt-on-change pin.
RB5/PGM RB5 PGM	26	38	15	42	I/O I	TTL ST	Digital I/O. Interrupt-on-change pin. Low voltage ICSP programming enable.
RB6/PGC RB6 PGC	27	39	16	43	I/O I	TTL ST	Digital I/O. In-Circuit Debugger pin. Interrupt-on-change pin. ICSP programming clock.
RB7/PGD RB7 PGD	28	40	17	44	I/O I/O	TTL ST	Digital I/O. In-Circuit Debugger pin. Interrupt-on-change pin. ICSP programming data.

Legend: TTL = TTL compatible input
 ST = Schmitt Trigger input with CMOS levels
 I = Input
 P = Power
 CMOS = CMOS compatible input or output
 Analog = Analog input
 O = Output
 OD = Open Drain (no P diode to V_{DD})

TABLE 1-2: PIC18FXX8 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number				Pin Type	Buffer Type	Description
	PIC18F248/258	PIC18F448/458					
	SPDIP, SOIC	PDIP	TQFP	PLCC			
RC0/T1OSO/T1CKI RC0 T1OSO T1CKI	11	15	32	16	I/O O I	ST — ST	PORTC is a bi-directional I/O port. Digital I/O. Timer1 oscillator output. Timer1/Timer3 external clock input.
RC1/T1OSI RC1 T1OSI	12	16	35	18	I/O I	ST CMOS	Digital I/O. Timer1 oscillator input.
RC2/CCP1 RC2 CCP1	13	17	36	19	I/O I/O	ST ST	Digital I/O. Capture1 input/Compare1 output/PWM1 output.
RC3/SCK/SCL RC3 SCK SCL	14	18	37	20	I/O I/O I/O	ST ST ST	Digital I/O. Synchronous serial clock input/output for SPI mode. Synchronous serial clock input/output for I ² C mode.
RC4/SDI/SDA RC4 SDI SDA	15	23	42	25	I/O I I/O	ST ST ST	Digital I/O. SPI data in. I ² C data I/O.
RC5/SDO RC5 SDO	16	24	43	26	I/O O	ST —	Digital I/O. SPI data out.
RC6/TX/CK RC6 TX CK	17	25	44	27	I/O O I/O	ST — ST	Digital I/O. USART asynchronous transmit. USART synchronous clock (see RX/DT).
RC7/RX/DT RC7 RX DT	18	26	1	29	I/O I I/O	ST ST ST	Digital I/O. USART asynchronous receive. USART synchronous data (see TX/CK).

Legend: TTL = TTL compatible input
 ST = Schmitt Trigger input with CMOS levels
 I = Input
 P = Power
 CMOS = CMOS compatible input or output
 Analog = Analog input
 O = Output
 OD = Open Drain (no P diode to VDD)

PIC18FXX8

TABLE 1-2: PIC18FXX8 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number				Pin Type	Buffer Type	Description
	PIC18F248/258	PIC18F448/458					
	SPDIP, SOIC	PDIP	TQFP	PLCC			
RD0/PSP0/C1IN+ RD0 PSP0 C1IN+	—	19	38	21	I/O I/O I	ST TTL Analog	PORTD is a bi-directional I/O port. These pins have TTL input buffers when external memory is enabled. Digital I/O. Parallel slave port data. Comparator 1 input.
RD1/PSP1/C1IN- RD1 PSP1 C1IN-	—	20	39	22	I/O I/O I	ST TTL Analog	Digital I/O. Parallel slave port data. Comparator 1 input.
RD2/PSP2/C2IN+ RD2 PSP2 C2IN+	—	21	40	23	I/O I/O I	ST TTL Analog	Digital I/O. Parallel slave port data. Comparator 2 input.
RD3/PSP3/C2IN- RD3 PSP3 C2IN-	—	22	41	24	I/O I/O I	ST TTL Analog	Digital I/O. Parallel slave port data. Comparator 2 input.
RD4/PSP4/ECCP1/P1A RD4 PSP4 ECCP1 P1A	—	27	2	30	I/O I/O I/O O	ST TTL ST —	Digital I/O. Parallel slave port data. ECCP1 capture/compare. ECCP1 PWM output A.
RD5/PSP5/P1B RD5 PSP5 P1B	—	28	3	31	I/O I/O O	ST TTL —	Digital I/O. Parallel slave port data. ECCP1 PWM output B.
RD6/PSP6/P1C RD6 PSP6 P1C	—	29	4	32	I/O I/O O	ST TTL —	Digital I/O. Parallel slave port data. ECCP1 PWM output C.
RD7/PSP7/P1D RD7 PSP7 P1D	—	30	5	33	I/O I/O O	ST TTL —	Digital I/O. Parallel slave port data. ECCP1 PWM output D.

Legend: TTL = TTL compatible input
 ST = Schmitt Trigger input with CMOS levels
 I = Input
 P = Power
 CMOS = CMOS compatible input or output
 Analog = Analog input
 O = Output
 OD = Open Drain (no P diode to VDD)

PIC18FXX8

TABLE 1-2: PIC18FXX8 PINOUT I/O DESCRIPTIONS (CONTINUED)

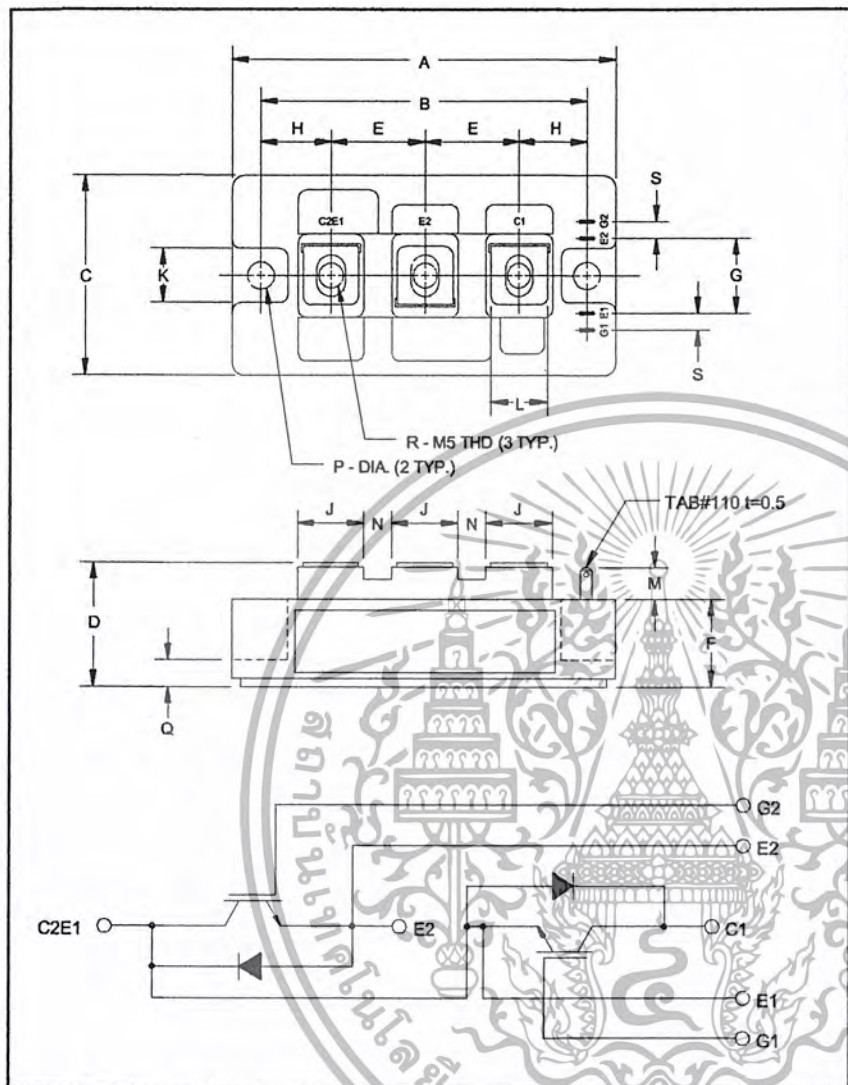
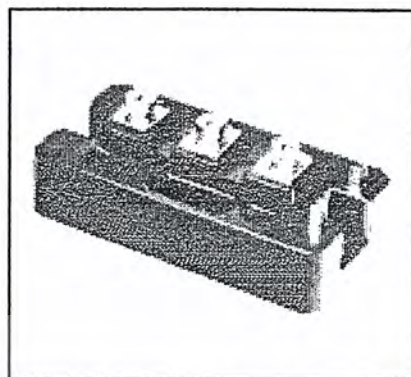
Pin Name	Pin Number				Pin Type	Buffer Type	Description
	PIC18F248/258	PIC18F448/458					
	SPDIP, SOIC	PDIP	TQFP	PLCC			
RE0/AN5/ \overline{RD} RE0 AN5 \overline{RD}	—	8	25	9	I/O I I	ST Analog TTL	<p>PORTE is a bi-directional I/O port.</p> <p>Digital I/O. Analog input 5. Read control for parallel slave port (see \overline{WR} and \overline{CS} pins).</p>
RE1/AN6/ \overline{WR} /C1OUT RE1 AN6 \overline{WR} C1OUT	—	9	26	10	I/O I I O	ST Analog TTL Analog	<p>Digital I/O. Analog input 6. Write control for parallel slave port (see \overline{CS} and \overline{RD} pins). Comparator 1 output.</p>
RE2/AN7/ \overline{CS} /C2OUT RE2 AN7 \overline{CS} C2OUT	—	10	27	11	I/O I I O	ST Analog TTL Analog	<p>Digital I/O. Analog input 7. Chip select control for parallel slave port (see \overline{RD} and \overline{WR} pins). Comparator 2 output.</p>
Vss	19, 8	12, 31	6, 29	13, 34	—	—	Ground reference for logic and I/O pins.
VDD	20	11, 32	7, 28	12, 35	—	—	Positive supply for logic and I/O pins.

Legend: TTL = TTL compatible input CMOS = CMOS compatible input or output
 ST = Schmitt Trigger input with CMOS levels Analog = Analog input
 I = Input O = Output
 P = Power OD = Open Drain (no P diode to VDD)

MITSUBISHI IGBT MODULES

CM100DY-24H

HIGH POWER SWITCHING USE
INSULATED TYPE



Outline Drawing and Circuit Diagram

Dimensions	Inches	Millimeters
A	3.70	94.0
B	3.150±0.01	80.0±0.25
C	1.89	48.0
D	1.18 Max.	30.0 Max.
E	0.90	23.0
F	0.83	21.2
G	0.71	18.0
H	0.67	17.0
J	0.62	16.0

Dimensions	Inches	Millimeters
K	0.51	13.0
L	0.47	12.0
M	0.30	7.5
N	0.28	7.0
P	0.256 Dia.	Dia. 6.5
Q	0.31	8.0
R	M5 Metric	M5
S	0.16	4.0

Description:

Mitsubishi IGBT Modules are designed for use in switching applications. Each module consists of two IGBTs in a half-bridge configuration with each transistor having a reverse-connected super-fast recovery free-wheel diode. All components and interconnects are isolated from the heat sinking baseplate, offering simplified system assembly and thermal management.

Features:

- Low Drive Power
- Low $V_{CE(sat)}$
- Discrete Super-Fast Recovery Free-Wheel Diode
- High Frequency Operation
- Isolated Baseplate for Easy Heat Sinking

Applications:

- AC Motor Control
- Motion/Servo Control
- UPS
- Welding Power Supplies

Ordering Information:

Example: Select the complete part module number you desire from the table below -i.e. CM100DY-24H is a 1200V (V_{CES}), 100 Ampere Dual IGBT Module.

Type	Current Rating Amperes	V_{CES} Volts (x 50)
CM	100	24

Sep.1998

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



CM100DY-24H

HIGH POWER SWITCHING USE
INSULATED TYPEAbsolute Maximum Ratings, $T_j = 25^\circ\text{C}$ unless otherwise specified

Ratings	Symbol	CM100DY-24H	Units
Junction Temperature	T_j	-40 to 150	$^\circ\text{C}$
Storage Temperature	T_{stg}	-40 to 125	$^\circ\text{C}$
Collector-Emitter Voltage (G-E SHORT)	V_{CES}	1200	Volts
Gate-Emitter Voltage (C-E SHORT)	V_{GES}	± 20	Volts
Collector Current ($T_C = 25^\circ\text{C}$)	I_C	100	Amperes
Peak Collector Current	I_{CM}	200*	Amperes
Emitter Current** ($T_C = 25^\circ\text{C}$)	I_E	100	Amperes
Peak Emitter Current**	I_{EM}	200*	Amperes
Maximum Collector Dissipation ($T_C = 25^\circ\text{C}$, $T_j \leq 150^\circ\text{C}$)	P_C	780	Watts
Mounting Torque, M5 Main Terminal	-	1.47 ~ 1.96	$\text{N} \cdot \text{m}$
Mounting Torque, M6 Mounting	-	1.96 ~ 2.94	$\text{N} \cdot \text{m}$
Weight	-	270	Grams
Isolation Voltage (Main Terminal to Baseplate, AC 1 min.)	V_{iso}	2500	V_{rms}

*Pulse width and repetition rate should be such that the device junction temperature (T_j) does not exceed $T_{j(max)}$ rating.

**Represents characteristics of the anti-parallel, emitter-to-collector free-wheel diode (FWDi).

Static Electrical Characteristics, $T_j = 25^\circ\text{C}$ unless otherwise specified

Characteristics	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Collector-Cutoff Current	I_{CES}	$V_{CE} = V_{CES}$, $V_{GE} = 0V$	-	-	1.0	mA
Gate Leakage Current	I_{GES}	$V_{GE} = V_{GES}$, $V_{CE} = 0V$	-	-	0.5	μA
Gate-Emitter Threshold Voltage	$V_{GE(th)}$	$I_C = 10\text{mA}$, $V_{CE} = 10V$	4.5	6.0	7.5	Volts
Collector-Emitter Saturation Voltage	$V_{CE(sat)}$	$I_C = 100A$, $V_{GE} = 15V$	-	2.5	3.4**	Volts
		$I_C = 100A$, $V_{GE} = 15V$, $T_j = 150^\circ\text{C}$	-	2.25	-	Volts
Total Gate Charge	Q_G	$V_{CC} = 600V$, $I_C = 100A$, $V_{GE} = 15V$	-	500	-	nC
Emitter-Collector Voltage	V_{EC}	$I_E = 100A$, $V_{GE} = 0V$	-	-	3.5	Volts

** Pulse width and repetition rate should be such that device junction temperature rise is negligible.

Dynamic Electrical Characteristics, $T_j = 25^\circ\text{C}$ unless otherwise specified

Characteristics	Symbol	Test Conditions	Min.	Typ.	Max.	Units	
Input Capacitance	C_{ies}		-	-	20	nF	
Output Capacitance	C_{oes}	$V_{GE} = 0V$, $V_{CE} = 10V$	-	-	7	nF	
Reverse Transfer Capacitance	C_{res}		-	-	4	nF	
Resistive	Turn-on Delay Time	$t_{d(on)}$	-	-	250	ns	
Load	Rise Time	t_r	$V_{CC} = 600V$, $I_C = 100A$,	-	-	350	ns
Switching	Turn-off Delay Time	$t_{d(off)}$	$V_{GE1} = V_{GE2} = 15V$, $R_G = 3.1\Omega$	-	-	300	ns
Time	Fall Time	t_f		-	-	350	ns
Diode Reverse Recovery Time	t_{rr}	$I_E = 100A$, $di_E/dt = -200A/\mu\text{s}$	-	-	250	ns	
Diode Reverse Recovery Charge	Q_{rr}	$I_E = 100A$, $di_E/dt = -200A/\mu\text{s}$	-	0.74	-	μC	

Thermal and Mechanical Characteristics, $T_j = 25^\circ\text{C}$ unless otherwise specified

Characteristics	Symbol	Test Conditions	Min.	Typ.	Max.	Units
Thermal Resistance, Junction to Case	$R_{th(j-c)}$	Per IGBT	-	-	0.16	$^\circ\text{C/W}$
Thermal Resistance, Junction to Case	$R_{th(j-c)}$	Per FWDi	-	-	0.35	$^\circ\text{C/W}$
Contact Thermal Resistance	$R_{th(c-f)}$	Per Module, Thermal Grease Applied	-	-	0.065	$^\circ\text{C/W}$

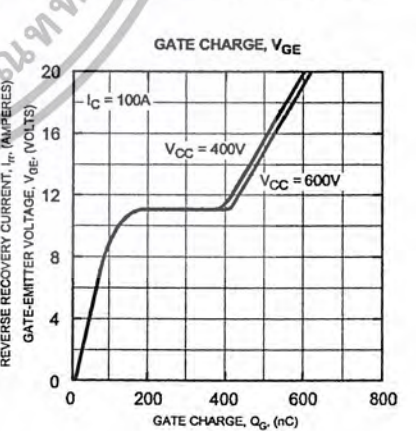
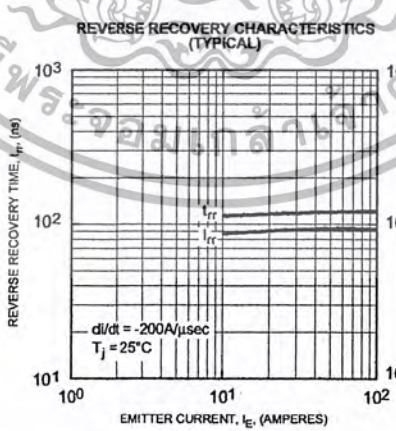
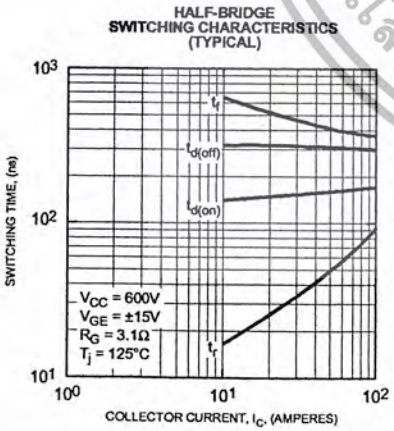
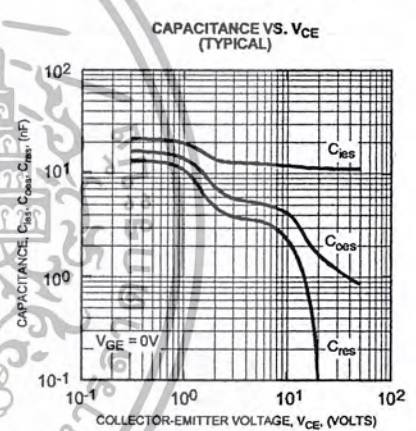
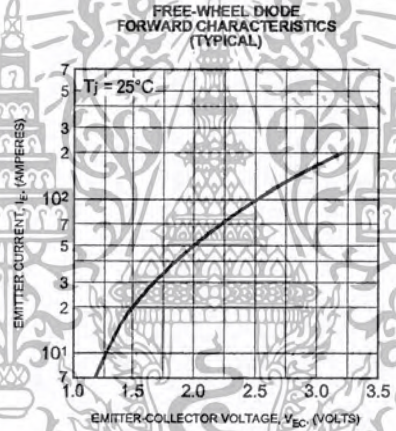
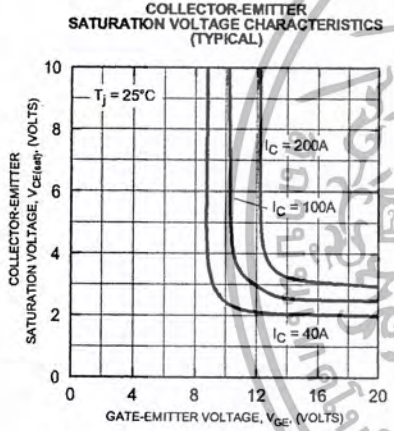
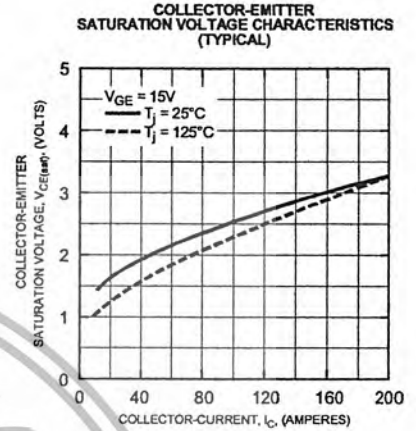
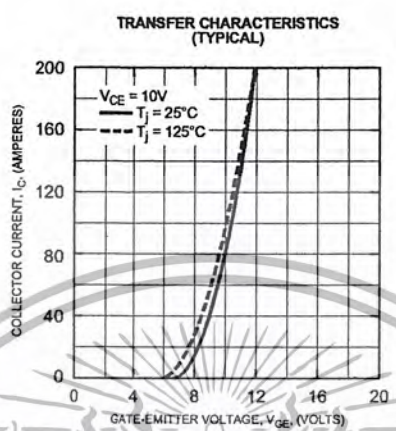
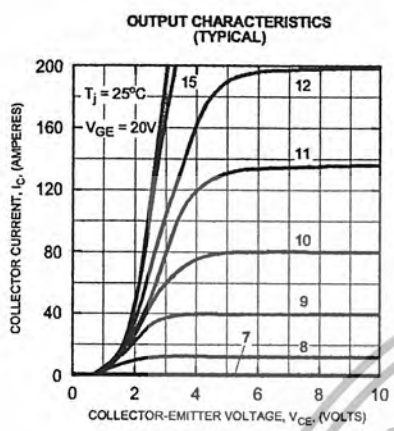
Sep.1998

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



CM100DY-24H

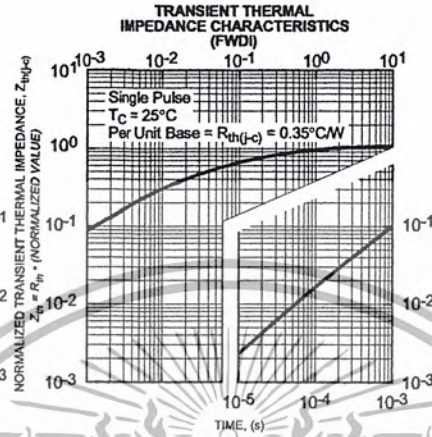
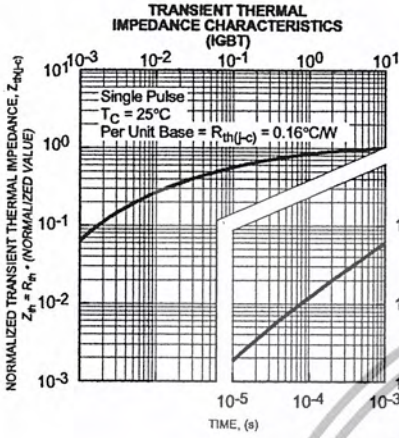
HIGH POWER SWITCHING USE
INSULATED TYPE



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานี้เท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

CM100DY-24H

**HIGH POWER SWITCHING USE
INSULATED TYPE**



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษานั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้