

คอนเวอร์เตอร์ฟลูบริดจ์สำหรับกังหันลมขนาดเล็ก  
Full-Bridge Converter for Small Wind Turbine

นายสุวิชา	วาทยานนท์
นายธเนศ	สุททหลวง
นายมาศธร	พรรณรัตน์

โครงการนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต  
สาขาวิศวกรรมไฟฟ้า คณะวิศวกรรมศาสตร์  
มหาวิทยาลัยศรีปทุม  
ปีการศึกษา 2555  
55EE125

หัวข้อโครงการ           คอนเวอร์เตอร์ฟลูบรีดจ์สำหรับกังหันลมขนาดเล็ก  
โดย                            นายสุวิชา            วาทยานนท์  
                                  นายชเนศ            สุทธหลวง  
                                  นายมาศธร         พรรณรัตน์  
สาขาวิชา                 วิศวกรรมไฟฟ้า  
อาจารย์ที่ปรึกษา         อ.กษิเดช ทิพย์อมรวิวัฒน์  
อาจารย์ที่ปรึกษาร่วม    ผู้ช่วยศาสตราจารย์ ดร.นิमित บุญภิรมย์

---

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยศรีปทุม อนุมัติให้นับโครงการ  
ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต

.....หัวหน้าสาขาวิชาวิศวกรรมไฟฟ้า  
( ผู้ช่วยศาสตราจารย์ พศวีร์ ศรีโหมด )

.....อาจารย์ที่ปรึกษา  
( อ.กษิเดช ทิพย์อมรวิวัฒน์ )

.....อาจารย์ที่ปรึกษาร่วม  
( ผู้ช่วยศาสตราจารย์ ดร.นิमित บุญภิรมย์ )

วันที่.....เดือน.....พ.ศ. 2555

รหัสโครงการ 55EE125

**คอนเวอร์เตอร์ฟลูบริดจ์สำหรับกังหันลมขนาดเล็ก**  
**Full-Bridge Converter for Small Wind Turbine**

**บทคัดย่อ (Abstract)**

โครงการนี้เป็นการนำเสนอวงจรคอนเวอร์เตอร์แบบฟลูบริดจ์สำหรับกังหันลมขนาดเล็ก ซึ่งมีวัตถุประสงค์ในการศึกษาและออกแบบวงจรคอนเวอร์เตอร์แบบฟลูบริดจ์เพื่อสร้างชุดวงจรคอนเวอร์เตอร์แบบฟลูบริดจ์ให้มีขนาด 100 W ที่สามารถนำมาใช้กับกังหันลมขนาดเล็กได้ โดยกังหันลมเป็นแบบแม่เหล็กถาวร 3 phase ขนาด 200 W-75 V เพื่อนำพลังงานไฟฟ้าที่ได้มาใช้ประโยชน์ โดยให้ความสว่างแก่พระวิษณุกรรม

## กิตติกรรมประกาศ

โครงการ“คอนเวอร์เตอร์ฟลูบรีดจ์สำหรับกังหันลมขนาดเล็ก”ได้ดำเนินการจนสำเร็จ ลุล่วงไปได้ด้วยดี ด้วยความกรุณาจากท่านอาจารย์ กษิเดช ทิพย์อมรวิวัฒน์ และ ท่านอาจารย์ผู้ช่วย ศาสตราจารย์ ดร.นิमित บุญภิรมย์ ซึ่งเป็นอาจารย์ที่ปรึกษาและให้คำแนะนำต่างๆในการปฏิบัติงาน คณะผู้จัดทำจึงขอกราบขอบพระคุณท่านอาจารย์เป็นอย่างสูงในความกรุณาครั้งนี้และขอกราบ ขอบพระคุณอาจารย์ทุกท่านรวมถึงเจ้าหน้าที่ในภาควิชาวิศวกรรมไฟฟ้าทุกฝ่ายที่ได้มีส่วน ช่วยเหลือจนโครงการสำเร็จได้ด้วยดีและขอบคุณเพื่อนๆทุกคนที่มีส่วนช่วยเหลือให้คำแนะนำ อื่นทั้งคอยให้กำลังใจ ประโยชน์และคุณค่าใดที่เกิดขึ้นจากปัญญานิพนธ์ฉบับนี้ผู้จัดทำขอมอบ ให้บิดา มารดาของคณะผู้จัดทำทุกท่านที่มีส่วนช่วยเป็นกำลังใจ กำลังทรัพย์ทำให้สามารถสำเร็จ ลุล่วงไปได้ด้วยดี

คณะผู้จัดทำ

2555

## สารบัญ

หน้า

บทคัดย่อ	ก
กิตติกรรมประกาศ	ข
สารบัญ	ค
สารบัญตาราง	จ
สารบัญภาพ	ฉ
บทที่ 1 บทนำ	
1.1 ความสำคัญของปัญหา	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตโครงการ	2
1.4 ประโยชน์ของโครงการ	2
1.5 ส่วนประกอบโครงการ	2
1.6 โครงสร้างโครงการ	3
บทที่ 2 ทฤษฎีที่เกี่ยวข้อง	
2.1 คอนเวอเตอร์แบบฟลูบริดจ์	4
2.2 การขับเพาเวอร์มอสเฟต	5
2.3 การควบคุมระดับแรงดันแบบพี ดับเบิ้ลยู เอ็ม	7
2.4 ตัวควบคุมอัตโนมัติ	8
2.5 ตัวควบคุมแบบสัดส่วนร่วมกับแบบปริพันธ์	11
2.6 กังหันลมชนิดแกนหมุนแนวนอน	13
2.7 แบตเตอรี่	16

## สารบัญ (ต่อ)

	หน้า
บทที่ 3 การออกแบบโครงงาน	
3.1 กำหนดหาพิกัดของไดโอดในวงจรเรียงกระแส	18
3.2 การออกแบบตัวเหนี่ยวนำ	20
3.3 การพันค่าความเหนี่ยวนำ (L) มีหน่วยเป็นเฮนรี่	22
3.4 การพันขดลวดหม้อแปลงสวิตชิงหรือหม้อแปลงต่างๆ	23
3.5 การพันหม้อแปลงขดไพรมารี	23
บทที่ 4 การทดลองและผลการทดลอง	
4.1 การทดลองกับโหลดที่เป็นความต้านทาน และผลการทดลอง	27
4.2 แสดงผลการทดลองวงจร	27
4.3 การเขียนไมโครคอนโทรลเลอร์	30
4.4 สรุปผลการทดลอง	43
บทที่ 5 สรุป	
เอกสารอ้างอิง	
ภาคผนวก	

## สารบัญตาราง

	หน้า
ตารางที่ 1.1 แสดงขนาดของกังหันลม	15
ตารางที่ 4.1 แสดงผลของการทดลอง	43

## สารบัญภาพ

	หน้า
ภาพที่ 1.1 แสดงโครงสร้างของโครงการ	3
ภาพที่ 2.1 วงจรคอนเวอร์เตอร์แบบฟลูบริดจ์	4
ภาพที่ 2.2 แสดงขาของมอเตอร์เฟส	5
ภาพที่ 2.3 โครงสร้างภายในของ IR2110	6
ภาพที่ 2.4 แสดงการไหลของกระแสไฟฟ้าในขณะ Bootstrap	6
ภาพที่ 2.5 กระบวนการสร้างสัญญาณ Pulse Width Modulation	7
ภาพที่ 2.6 บล็อกไดอะแกรมของตัวควบคุมแบบสัดส่วน	9
ภาพที่ 2.7 ผลตอบสนองต่อฟังก์ชันขั้นบันไดของตัวควบคุมแบบสัดส่วน	10
ภาพที่ 2.8 ฟังก์ชันถ่ายโอนและผลตอบสนองต่อฟังก์ชันขั้นบันไดแบบปริพันธ์	11
ภาพที่ 2.9 บล็อกไดอะแกรมสัดส่วนร่วมกับปริพันธ์	13
ภาพที่ 2.10 แสดงกัณฑ์ลมนแกนหมุนแวนอน	13
ภาพที่ 2.11 แสดงองค์ประกอบของกัณฑ์ลมนแกนหมุนแวนอน	14
ภาพที่ 3.1 แสดงวงจรเรียงกระแส	18
ภาพที่ 3.2 แสดงวงจรคอนเวอร์เตอร์แบบฟลูบริดจ์	20
ภาพที่ 3.3 แสดงตัวควบคุมวงจรฟลูบริดจ์ตัวที่ 1	24
ภาพที่ 3.4 แสดงตัวควบคุมวงจรฟลูบริดจ์ตัวที่ 2	24
ภาพที่ 3.5 แสดงวงจรเช็กระงัดันฝั่งเอาต์พุต	25
ภาพที่ 3.6 แสดงวงจรลดแรงดันลงให้เหลือ 12 V	25
ภาพที่ 3.7 แสดงวงจรที่เชื่อมต่อเข้าด้วยกันทั้งหมด	26
ภาพที่ 3.8 สัญญาณ PWM ช่วงเวลาที่เปิด Duty Cycle 47 %	27
ภาพที่ 3.9 สัญญาณ PWM เทียบกับสัญญาณ Hi Size Duty Cycle 47 %	28
ภาพที่ 4.10 สัญญาณ PWM เทียบกับสัญญาณ Lo Size Duty Cycle 47 %	28
ภาพที่ 4.11 สัญญาณ PWM เทียบกับสัญญาณ Out Put Duty Cycle 47 %	29
ภาพที่ 4.12 สัญญาณ Hi Size กับ หม้อแปลง Duty Cycle 47 %	29
ภาพที่ 4.13 วงจรฟลูบริดจ์คอนเวอร์เตอร์	30
ภาพที่ 4.14 ชุดไมโครคอนโทรลเลอร์	43



# บทที่ 1

## บทนำ

เป็นที่ทราบเป็นอย่างดีว่าการใช้ประโยชน์จากพลังงานลมมีมาอย่างยาวนานโดยการเปลี่ยนพลังงานลมหรือพลังงานกลให้เป็นพลังงานไฟฟ้าเพื่อใช้ประโยชน์ในด้านต่างๆ ด้วยความไม่แน่นอนของความเร็วลมที่เกิดขึ้น การทำงานของเครื่องกำเนิดไฟฟ้าขนาดเล็กไม่สามารถนำมาใช้ประโยชน์ได้โดยตรงดังนั้น จึงได้จัดทำชุดคอนเวอร์เตอร์แบบฟลูบริดจ์สำหรับเครื่องกำเนิดไฟฟ้าพลังงานลมขนาดจิ๋วเพื่อนำกระแสที่ได้มาอัดประจุลงในแบตเตอรี่ขนาดแรงดัน 12 V ความสามารถในการทำงานของวงจรไม่เกิน 100 W

### 1.1 ความสำคัญของปัญหา

1. พลังงานทางธรรมชาติยังไม่ได้ถูกนำมาใช้ให้เกิดประโยชน์มากนัก
2. ต้นทุนในการนำเข้พลังงานน้ำมันจากต่างประเทศมีราคาสูง
3. พลังงานลมมีทั้งกลางวันและกลางคืน สามารถนำมาใช้ประโยชน์ได้ตลอดเวลา
4. จำนวนกังหันลมยังไม่ได้มีการพัฒนาอย่างกว้างขวางในประเทศไทย
5. ต้นทุนการผลิตพลังงานไฟฟ้าจากลมมีต้นทุนต่ำ

### 1.2 วัตถุประสงค์ของโครงการ

1. เพื่อสร้างคอนเวอร์เตอร์แบบฟลูบริดจ์ของกังหันลมผลิตไฟฟ้าขนาดเล็กความเร็วลมต่ำ
2. เพื่อประยุกต์ใช้กับทฤษฎีวิชาอิเล็กทรอนิกส์กำลังในการควบคุมแรงดันและกำลังไฟฟ้า
3. เพื่อวิเคราะห์ปัจจัยของการผลิตพลังงานไฟฟ้าจากกังหันลมในบริเวณมหาวิทยาลัยศรีปทุม
4. เพื่อให้เกิดทักษะและประสบการณ์ในงานด้านวิศวกรรม

### 1.3 ขอบเขตโครงการ

1. ใช้กังหันลมผลิตไฟฟ้าและเครื่องกำเนิดไฟฟ้าแบบแม่เหล็กถาวร 3 phase ขนาด 200 W 75 V จากโครงการปี 2546
2. สร้างคอนเวอร์เตอร์แบบ Full Bridge ขนาด 100 W 12 V
3. จัดหาตัวควบคุมป้อนกลับแบบ PI ควบคุมระดับแรงดันเอาต์พุต

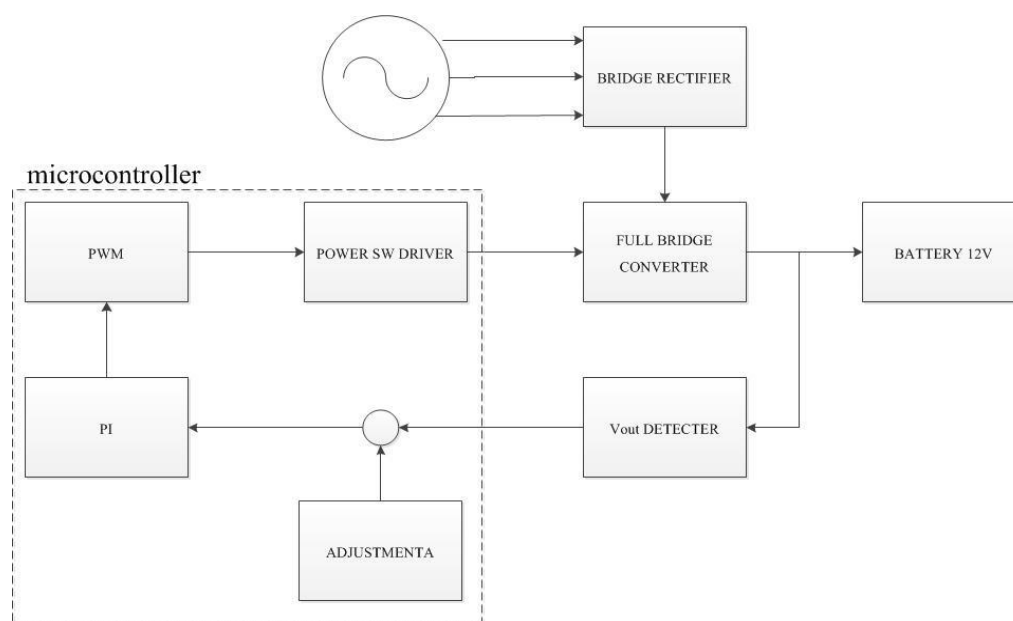
### 1.4 ประโยชน์ของโครงการ

1. ได้ศึกษาถึงหลักการทำงานของคอนเวอร์เตอร์แบบ Full Bridge
2. ได้ผลงานที่ปฏิบัติแล้วใช้งานได้จริง
3. ได้ทักษะในด้านการปฏิบัติการและสามารถนำไปใช้เป็นแนวทางในการพัฒนาด้านระบบควบคุมและระบบอื่นๆ

### 1.5 ส่วนประกอบโครงการ

1. กังหันลมผลิตไฟฟ้าแบบแกนหมุนแนวนอนและเครื่องกำเนิดไฟฟ้าแบบแม่เหล็กถาวร 3 phase ขนาด 200 W-75 V
2. ชุดคอนเวอร์เตอร์แบบ Full Bridge Converter 100 W 12 V
3. ชุด PI Controller

## 1.6 โครงสร้างโครงการ



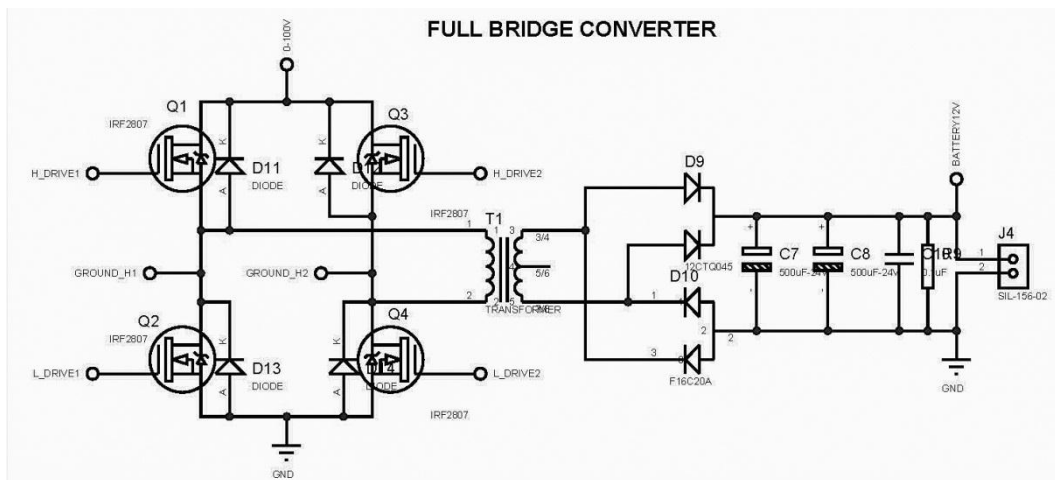
ภาพที่ 1.1 แสดงโครงสร้างของโครงการ

## บทที่ 2

### ทฤษฎีที่เกี่ยวข้อง

#### 2.1 คอนเวอร์เตอร์แบบฟลูบริดจ์ ( Full Bridge Converter ) [1] [4]

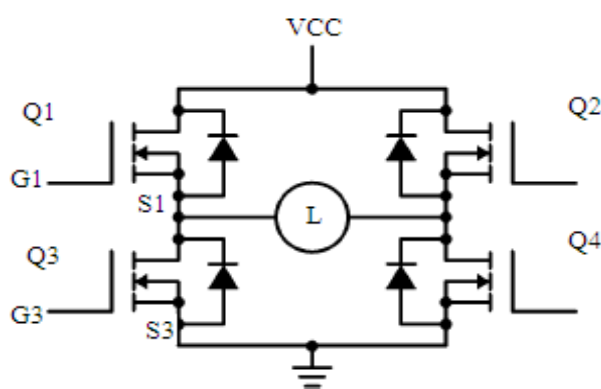
คอนเวอร์เตอร์แบบฟลูบริดจ์(Full-Bridge Converter)ขณะทำงานจะมีแรงดันตกคร่อมขดไฟโพรมารี่เท่ากับแรงดันอินพุต แต่แรงดันตกคร่อมเพาเวอร์ทรานซิสเตอร์มีค่าเพียงครึ่งหนึ่งของแรงดันอินพุตเท่านั้นและค่ากระแสสูงสุดที่เพาเวอร์ทรานซิสเตอร์แต่ละตัวนั้นมีค่าเป็นครึ่งหนึ่งของค่ากระแสสูงสุดในฮาล์ฟบริดจ์คอนเวอร์เตอร์ที่กำลังขาออกเท่ากัน เนื่องจากข้อกำหนดด้านเพาเวอร์ทรานซิสเตอร์ลดน้อยลงไปกำลังงานสูงสุดที่จะได้จากฟลูบริดจ์คอนเวอร์เตอร์จึงมีค่าสูงได้ตั้งแต่ 500 วัตต์ จนถึง 1000 วัตต์



ภาพที่ 2.1 วงจรคอนเวอร์เตอร์แบบฟลูบริดจ์

## 2.2 การขับ Power Mosfet (PMF)

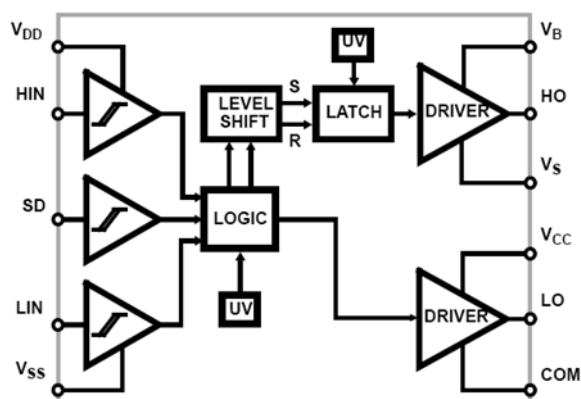
การขับ PMF ปกติจะขับด้วยโวลเตจ ด้วยการป้อนแรงดันไฟฟ้าประมาณ 10-20 V ระหว่างขา Gate กับขา Source ซึ่งถ้าเป็นชนิด N channel ก็ป้อนไฟบวกที่ขา Gate และขั้วลบก็ต่อเข้ากับขา Source แต่สำหรับการขับ PMF ของวงจร Full Bridge ดังแสดงในรูป ซึ่งมี PMF อยู่สี่ตัวคือ Q1 ถึง Q4 การขับจะแบ่งออกเป็นสองด้านคือ ด้าน High Side Switch (HSS) ก็คือ Q1 กับ Q2 และด้าน Low Side Switch (LSS) ก็คือ Q3 กับ Q4 ดังนั้นเทคนิคการขับก็จะแบ่งเป็นสองด้านด้วย



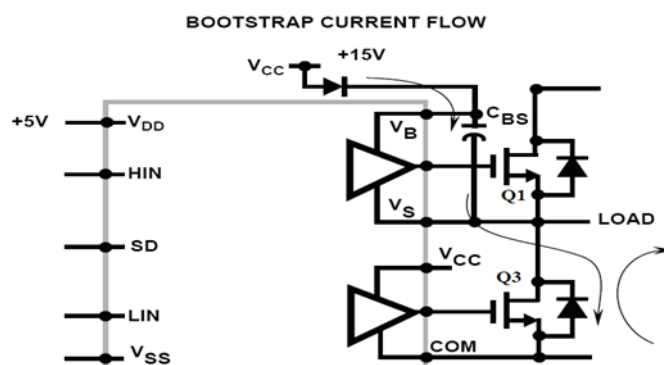
ภาพที่ 2.2 แสดงขาของมอดเฟส

การขับด้าน LSS ก็จะสามารถต่อได้ตามปกติคือป้อน โวลต์เตจที่ต้องการขับระหว่าง G3 กับ S3 หรือขั้วกราวด์ ดังนั้น โวลต์เตจที่ใช้ขับก็จะมีค่าเท่ากับประมาณ 15 V แต่สำหรับการขับทางด้าน HSS โวลต์เตจที่ใช้ขับจะต่อระหว่าง G1 กับ S1 ต้องมีลักษณะพิเศษเนื่องจากในขณะที่ Q1 นำกระแสค่าโวลต์เตจที่ S1 จะมีค่าเท่ากับ VCC ถ้า VCC มีค่าเท่ากับ 100V เมื่อเทียบกับกราวด์ ดังนั้น โวลต์เตจที่ใช้ขับ G1 จะต้องมีค่าเท่ากับ  $100 + 15 = 115V$  เมื่อเทียบกับกราวด์ซึ่งเป็นไปได้สำหรับการขับแบบปกติที่จะใช้ไฟสูง 115V ดังนั้นโครงการนี้ได้ใช้ไอซีขับพิเศษคือ IR2110 ซึ่งได้ใช้เทคนิคพิเศษที่เรียกว่า BOOTSTRAP ซึ่งเป็นการออกแบบที่ชาญฉลาดที่สามารถใช้ไฟจากแหล่งจ่ายไฟปกติ 15V เมื่อเทียบกับกราวด์ ให้สามารถขับ Q1 ที่อยู่ทางด้าน HSS ได้ด้วยการเก็บประจุไฟฟ้าไว้ในคาปาซิเตอร์ CBS ในขณะที่ Q3 นำกระแสแสดงแสดงในภาพที่ 2.2

จากรูปในขณะที  $Q_3$  นำกระแส กระแสไฟจาก  $V_{CC}(+15V)$  ไหลผ่านไดโอด ผ่าน  $C_{bs}$  และผ่าน  $Q_3$  ลงกราวด์ ดังนั้น  $C_{bs}$  จึงถูกเก็บประจุจนเต็มและมีพลังงานเพียงพอที่จะขับ  $Q_1$  ด้วยการจ่ายกระแสไฟผ่านขั้ว  $V_b$  ไปยังตัวขับ Driver ไปยังขั้ว  $HO$



ภาพที่ 2.3 โครงสร้างภายในของ IR2110



ภาพที่ 2.4 แสดงการไหลของกระแสไฟฟ้าในขณะ Bootstrap

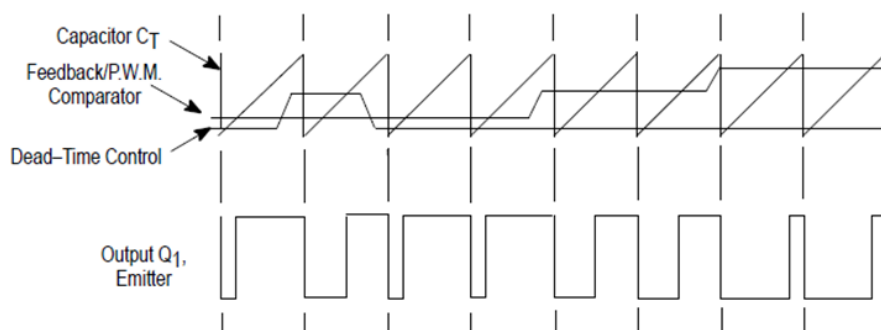
## 2.3 การควบคุมระดับแรงดันแบบ PWM

โดยทั่วไปแล้วคอนเวอร์เตอร์จะใช้เทคนิคการควบคุมความกว้างพัลส์ PWM ซึ่งเป็นการควบคุมโดยเปลี่ยนแปลงช่วงเวลาที่อยู่กรณีสวิตซ์อิเล็กทรอนิกส์ เช่น ทรานซิสเตอร์, มอสเฟต หรืออื่นๆ นั้นนำกระแสเป็นผลให้เกิดการควบคุมแรงดันที่เอาท์พุทให้ได้ค่าที่ต้องการซึ่งข้อดีของการควบคุมแรงดันแบบ PWM คือสามารถรักษาระดับแรงดันให้มีความคงที่สูง

### 2.3.1 หน้าที่ของวงจรถวาย (PWM)

หน้าที่ของวงจรถวาย PWM คือการสร้างสัญญาณแรงดันไฟฟ้าให้กลายเป็นค่าดิจิทัลที่ไซเคิลเพื่อไปขับขาคัดของมอสเฟตกำลังโดยค่าแรงดันไฟฟ้า (Dead-Time Control) จะถูกนำไปเปรียบเทียบกับสัญญาณฟันเลื่อย (Saw Tooth Waveform) ดังแสดงในภาพที่ 2.5 เมื่อแรงดันไฟฟ้า มีค่ามากกว่าแรงดันไฟฟ้าของสัญญาณฟันเลื่อยจะมีค่าเท่ากับศูนย์

ในกรณีที่แรงดันไฟฟ้าด้านออกของวงจรถวายคอนเวอร์เตอร์มีค่าลดต่ำกว่าแรงดันไฟฟ้าอ้างอิงแรงดันไฟฟ้าคลาดเคลื่อนที่ได้จากการเปรียบเทียบระหว่างแรงดันไฟฟ้าทั้งสองค่าจะมีค่าเพิ่มขึ้นเป็นผลทำให้แรงดันไฟฟ้าที่ควบคุมสัญญาณ PWM มีค่าเพิ่มขึ้นทำให้ค่าดิจิทัลที่ไซเคิลมีค่าเพิ่มขึ้นและด้วยค่าดิจิทัลที่เพิ่มขึ้นจะส่งผลให้แรงดันไฟฟ้าด้านออกของวงจรถวายคอนเวอร์เตอร์มีค่าเพิ่มขึ้นจนกระทั่งได้ค่าแรงดันคลาดเคลื่อนเท่ากับที่ออกแบบและในทำนองกลับกันกรณีที่แรงดันไฟฟ้าด้านออกของวงจรถวายคอนเวอร์เตอร์มีค่าเพิ่มขึ้นจะเป็นผลทำให้ค่าดิจิทัลที่ไซเคิลจะมีค่าลดลง



ภาพที่ 2.5 กระบวนการสร้างสัญญาณ Pulse Width Modulation

จากคุณสมบัติการสร้างสัญญาณแรงดันไฟฟ้าแบบ PWM สามารถสรุปข้อดีของการใช้เทคนิคการควบคุมแรงดันไฟฟ้าแบบ PWM ได้ดังนี้

- ตอบสนองการเปลี่ยนแปลงของแรงดันไฟฟ้าด้วยการควบคุมความกว้างของสัญญาณพัลส์ได้เป็นอย่างดี
- สามารถทำงานได้อย่างต่อเนื่องมีประสิทธิภาพสูงเนื่องจากมีค่าสูญเสียในการทำงานน้อยเมื่อเทียบกับการควบคุมไฟแบบเชิงเส้น

## 2.4 ตัวควบคุมอัตโนมัติ

ตัวควบคุมอัตโนมัติ ทำหน้าที่เป็นตัวเปรียบเทียบสัญญาณระหว่างสัญญาณทางออกของระบบกับสัญญาณทางเข้าหรือค่าที่ต้องการควบคุมหรือค่าอ้างอิงที่มีค่าคงที่ของระบบผลของการเปรียบเทียบที่ได้คือค่าเบี่ยงเบนหรือค่าผิดพลาดหลังจากนั้นตัวควบคุมอัตโนมัติจะนำค่าเบี่ยงเบนไปสร้างเป็นสัญญาณขึ้นมาใหม่เพื่อควบคุมให้ระบบลดการเบี่ยงเบนหรือลดค่าผิดพลาดลงให้เป็นศูนย์หรือน้อยที่สุดเท่าที่จะทำได้ลักษณะการทำงานของตัวควบคุมอัตโนมัติที่สร้างสัญญาณขึ้นมาควบคุมระบบดังกล่าวนี้ เรียกว่า “การทำงานควบคุม” ซึ่งสามารถแบ่งตัวควบคุมอัตโนมัติออกเป็นแบบต่าง ๆ ได้ดังนี้ ตัวควบคุมแบบสัดส่วน (P-Controller) ตัวควบคุมแบบปริพันธ์ (I-Controller), ตัวควบคุมแบบอนุพันธ์ (D-Controller), ตัวควบคุมแบบสัดส่วนร่วมกับแบบปริพันธ์ (PI-Controller), ตัวควบคุมแบบสัดส่วนร่วมกับแบบอนุพันธ์ (PD-Controller), ตัวควบคุมแบบสัดส่วนร่วมกับแบบปริพันธ์และอนุพันธ์ (PID-Controller)

### 2.4.1 ตัวควบคุมแบบสัดส่วน (P-Controller)

ตัวควบคุมแบบสัดส่วน (P-Controller) นี้เป็นตัวควบคุมที่ไม่มีหน่วง (Delay) ระหว่างสัญญาณทางเข้ากับสัญญาณทางออกแต่สัญญาณทางออกจะเป็นสัดส่วนกับสัญญาณทางเข้าซึ่งมีอัตราขยายเป็นพารามิเตอร์ที่ทำให้ขนาดของสัญญาณทางเข้าและสัญญาณทางออกเกิดความแตกต่างกันสามารถเขียนเป็นสมการได้ดังนี้



$$u_{out}(t) = g_p \times u_{in}(t) \quad (2.1)$$

โดยที่

$u_{out}(t)$  = สัญญาณทางออกของตัวควบคุม

$u_{in}(t)$  = สัญญาณผิดพลาดที่เกิดขึ้นหรือสัญญาณทางเข้าซึ่งเกิดจาก

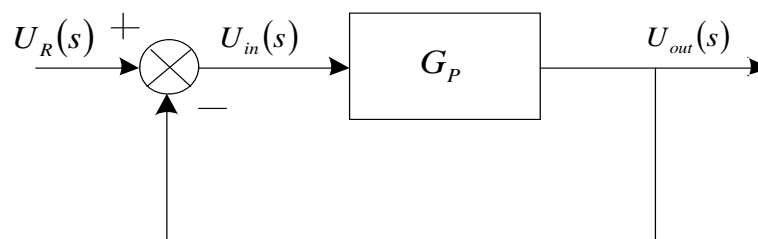
ความแตกต่างของค่าอ้างอิง ( $U_r$ ) กับค่าป้อนกลับ ( $U_{out}$ )

$g_p$  = ความไวของสัดส่วนหรืออัตราขยาย

เราสามารถเขียนเป็นฟังก์ชันถ่ายโอน (Transfer Function) ได้เป็น

$$F(s) = \frac{U_{out}(s)}{U_{in}(s)} = G_p \quad (2.2)$$

และจากฟังก์ชันถ่ายโอนในสมการที่ (2.2) นี้เราสามารถเขียนเป็นภาพบล็อก (Block Diagram) ได้ดังแสดงในรูปที่ 2.6

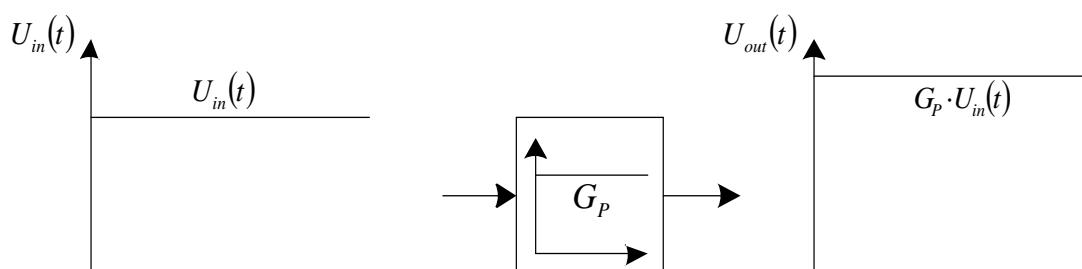


ภาพที่ 2.6 บล็อกไดอะแกรมของตัวควบคุมแบบสัดส่วน

เมื่อ

$U_R(s)$  = เป็นสัญญาณอ้างอิงหรืออาจจะเขียนในภาพของ

ผลตอบสนองต่อฟังก์ชันขั้นบันได (Step Response)



ภาพที่ 2.7 ผลตอบสนองต่อฟังก์ชันขั้นบันไดของตัวควบคุมแบบสัดส่วน

#### 2.4.2 ตัวควบคุมแบบปริพันธ์ (Integral Controller)

ตัวควบคุมแบบปริพันธ์จะเป็นตัวควบคุมที่ให้สัญญาณทางออกที่เป็นสัดส่วนโดยตรงกับการสะสมของสัญญาณผิดพลาดหรือสัญญาณทางเข้า ( $U_{in}$ ) ซึ่งก็คือพื้นที่ของสัญญาณผิดพลาดหรือสัญญาณเข้านั่นเองสามารถเขียนเป็นสมการเชิงอนุพันธ์ (Differential Equation) ได้ดังนี้

$$u_{out}(t) = \frac{1}{T_i} \int_0^t u_{in} dt$$

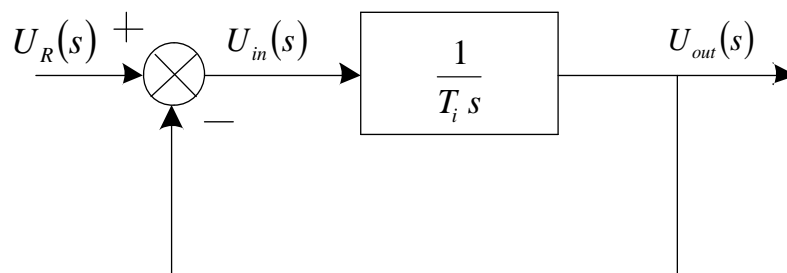
$$u_{out}(t) = u_{in}(t) \frac{1}{T_i} \quad (2.3)$$

เมื่อ

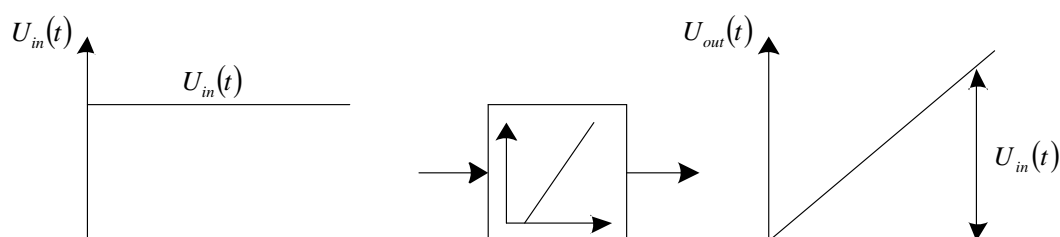
$T_i$  = เป็นค่าคงที่ปรับได้หรือเวลาคงตัวของปริพันธ์ (Integral Time) จากสมการดังกล่าวสามารถเขียนเป็นฟังก์ชันถ่ายโอน (Transfer Function) ได้ดังนี้

$$F(s) = \frac{U_{out}(s)}{U_{in}(s)} = \frac{1}{T_i(s)} = \frac{K_i}{S} \quad (2.4)$$

และฟังก์ชันถ่ายโอนนี้สามารถเขียนเป็นบล็อก (Block Diagram) และผลตอบสนองต่อฟังก์ชันขั้นบันไดดังแสดงในภาพที่ 2.8



ก. บล็อกของตัวควบคุมแบบปริพันธ์



ข. ผลตอบสนองต่อฟังก์ชันขั้นบันไดของตัวควบคุมแบบปริพันธ์

ภาพที่ 2.8 ฟังก์ชันถ่ายโอนและผลตอบสนองต่อฟังก์ชันขั้นบันไดแบบปริพันธ์

## 2.5 ตัวควบคุมแบบสัดส่วนร่วมกับแบบปริพันธ์ (PI-Controller)

เป็นการนำคุณสมบัติของตัวควบคุมแบบ P-Controller, I-Controller มารวมกันจะได้สัญญาณที่ออกจากตัวควบคุมชนิดนี้คือ

$$\begin{aligned}
 u_{out}(t) &= g_p u_{in}(t) + \frac{1}{T_i} \int_0^t u_{in}(t) \\
 &= g_p u_{in}(t) + \frac{1}{T_i} u_{in}(t)
 \end{aligned} \tag{2.5}$$

หรือเขียนในรูปของฟังก์ชันถ่ายโอน

$$U_{\text{out}}(s) = G_P U_{\text{in}}(s) + \frac{1}{T_i s} U_{\text{in}}(s)$$

$$= G_P U_{\text{in}}(s) \left( 1 + \frac{1}{T_i s G_P} \right)$$

$$U_{\text{out}}(s) = G_P \left( 1 + \frac{1}{T_i G_P s} \right) \quad (2.6)$$

เมื่อให้

$$U_{\text{in}} = T_i G_P$$

หรือเขียนในรูปของฟังก์ชันถ่ายโอน

$$F(s) = \frac{U_{\text{out}}(s)}{U_{\text{in}}(s)} = G_P \left( 1 + \frac{1}{T_i s} \right) \quad (2.7)$$

โดยที่

$$G_P = \text{Proportional Gain}$$

$$T_i = \text{Integral Time}$$

นำมาเขียนสมการฟังก์ชันถ่ายโอนในรูปของ Gain ได้ดังนี้

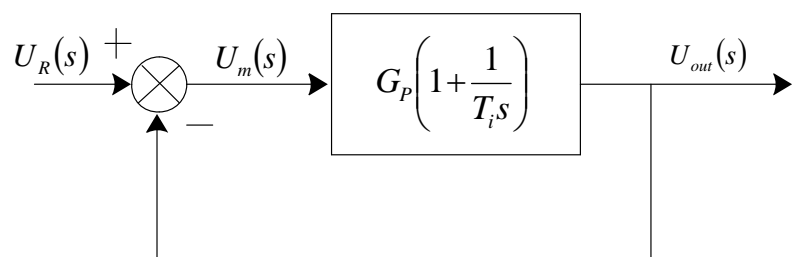
$$\begin{aligned} F(s) &= G_P + \frac{G_P}{T_i s} \\ &= G_P + \frac{K_i}{s} \end{aligned} \quad (2.8)$$

โดยที่

$$G_P = K$$

$$K_i = \frac{G_P}{T_i}$$

จากสมการฟังก์ชันถ่ายโอนนำมาเขียนเป็นภาพบล็อกไดอะแกรมได้ดังรูปที่ 2.9



ภาพที่ 2.9 บล็อกไดอะแกรมสัดส่วนร่วมกับปริพันธ์

## 2.6 กังหันลมชนิดแกนหมุนแนวนอน (Horizontal Axis Wind Turbine)

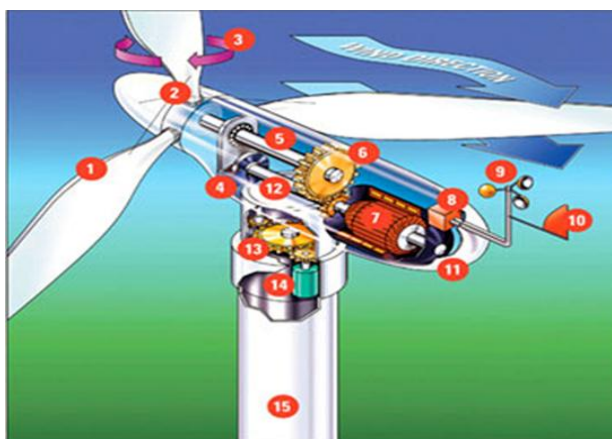
เป็นกังหันลมที่มีแกนหมุนขนานกับพื้นราบหรือขนานกับทิศทางการเคลื่อนที่ของลม โดยมีใบพัดยึดติดตั้งฉากกับแกนหมุน ทำหน้าที่รับแรงลมที่เคลื่อนตัวมากระทบทำให้เกิดการหมุนของใบพัด โดยกังหันลมชนิดแกนหมุนแนวนอนแบบสามใบพัดซึ่งมีการพัฒนาอย่างต่อเนื่อง เป็นกังหันลมที่ได้รับความนิยมใช้งานในเชิงพาณิชย์อย่างแพร่มากที่สุดถึงร้อยละ 75 ของกังหันลมที่มีการใช้งานในปัจจุบัน



ภาพที่ 2.10 แสดงกังหันลมแกนหมุนแนวนอน

### 2.6.1 ส่วนประกอบของกังหันลมผลิตไฟฟ้า (ชนิดแกนหมุนแนวนอน)

องค์ประกอบที่ประกอบขึ้นเป็นกังหันลมผลิตไฟฟ้าในหนึ่งชุด จะประกอบด้วย 4 องค์ประกอบหลักใหญ่ ๆ ได้แก่ชุดแกนหมุนใบพัด (Rotor Blade), ชุดห้องเครื่อง (Nacelle), ชุดเสา (Tower) และฐานราก(Foundation)



ภาพที่ 2.11 แสดงองค์ประกอบของกังหันลมแกนหมุนแนวนอน

1. ใบพัด เป็นตัวรับพลังงานลมและเปลี่ยนให้เป็นพลังงานกล ซึ่งยึดติดกับชุดแกนหมุนและส่งแรงจากแกนหมุนไปยังเพลากลานหมุน
2. เพลากลานหมุน รับแรงจากแกนหมุนใบพัด และส่งผ่านระบบกำลัง เพื่อหมุนและปั่นเครื่องกำเนิดไฟฟ้า
3. ห้องส่งกำลังเป็นระบบปรับเปลี่ยนและควบคุมความเร็วในการหมุน ระหว่างเพลากลานหมุนกับเพลากลางของเครื่องกำเนิดไฟฟ้า
4. ห้องเครื่อง ซึ่งมีขนาดใหญ่และมีความสำคัญต่อกังหันลม ใช้บรรจุระบบต่างๆ ของกังหันลม เช่น ระบบเกียร์เครื่องกำเนิดไฟฟ้า เบรก และระบบควบคุม
5. เครื่องกำเนิดไฟฟ้า ทำหน้าที่เปลี่ยนพลังงานกลเป็นพลังงานไฟฟ้า
6. ระบบควบคุมไฟฟ้า ใช้ระบบคอมพิวเตอร์เป็นตัวควบคุมการทำงาน และจ่ายกระแสไฟฟ้าเข้าสู่ระบบ
7. ระบบเบรก เป็นระบบกลไกเพื่อใช้ควบคุมการหยุดหมุนของใบพัดและเพลากลานหมุนของกังหัน เมื่อได้รับความเร็วลมเกินความสามารถของกังหันที่จะรับได้ และในระหว่างการซ่อมบำรุงรักษา

8. แกนคอกหมุนรับทิศทางลม เป็นตัวควบคุมการหมุนห้องเครื่อง เพื่อให้ใบพัดรับทิศทางลม โดยระบบอิเล็กทรอนิกส์ที่เชื่อมต่อให้มีความสัมพันธ์กับหางเสือรับทิศทางลมที่อยู่ด้านบนของเครื่อง
9. เครื่องวัดความเร็วลมและทิศทางลม เชื่อมต่อสายสัญญาณเข้าสู่ระบบคอมพิวเตอร์ เพื่อเป็นตัวชี้ขนาดของความเร็วและทิศทางของลม เพื่อที่คอมพิวเตอร์จะได้ควบคุมกลไกอื่นๆ ได้ถูกต้อง
10. เสาซึ่งตั้งอยู่ที่พื้นที่ที่ทำการก่อสร้างอย่างถูกวิธีตามหลักวิศวกรรม และเป็นตัวแบกรับส่วนที่เป็นตัวเครื่องที่อยู่ข้างบน

### 2.6.1.1 ขนาดของกังหันลมผลิตไฟฟ้า

ขนาดของกังหันลมผลิตไฟฟ้าถูกพัฒนาขึ้นเพื่อให้มีความสามารถในการผลิตไฟฟ้าได้ตามความต้องการใช้งาน ซึ่งจะขึ้นอยู่กับขนาดกำลังผลิตไฟฟ้า (Capacity) เส้นผ่านศูนย์กลางใบพัด (Rotor Diameter) และพื้นที่กวาดของใบพัด (Swept Area) ของกังหันลมผลิตไฟฟ้ารุ่นนั้น ๆ ดังแสดงรายละเอียดในตารางที่ 1.1

ตารางที่ 1.1 แสดงขนาดของกังหันลม

ขนาดของกังหันลม	ขนาดกำลังผลิต (kW)	เส้นผ่านศูนย์กลาง (m)	พื้นที่กวาด (m <sup>2</sup> )
ขนาดจิ๋ว (Micro Wind Turbine)	< 1.5	< 3	< 7
ขนาดเล็ก (Small Wind Turbine)	1.5-2.0	3-10	7-80
ขนาดกลาง (Medium Wind Turbine)	20-200	10-25	80 - 500
ขนาดใหญ่ (Large Wind Turbine)	200-1,500	25-70	500-3,850
ขนาดใหญ่่มาก (Very Large Wind Turbine)	> 1,500	> 70	> 3,850

1. กังหันลมขนาดเล็ก (Micro Wind Turbine) มีขนาดกำลังผลิตไฟฟ้าน้อยกว่า 1.5 กิโลวัตต์ เหมาะสำหรับติดตั้งผลิตไฟฟ้าในพื้นที่ห่างไกลเพื่อจัดเก็บกระแสไฟฟ้าลงในแบตเตอรี่ และมีภาระทางไฟฟ้าไม่มากนัก เช่น การใช้กับเครื่องมือสื่อสาร หรือแสงสว่างในบางเวลา
2. กังหันลมขนาดเล็ก (Small Wind Turbine) มีขนาดกำลังผลิตไฟฟ้าอยู่ระหว่าง 1.5-20 กิโลวัตต์ เหมาะสำหรับติดตั้งผลิตไฟฟ้าในพื้นที่ห่างไกลเพื่อจัดเก็บกระแสไฟฟ้าลงในแบตเตอรี่ และมีภาระทางไฟฟ้าไม่มากนัก เช่น ใช้ตามครัวเรือนหรือสำนักงานขนาดเล็กที่อยู่ห่างไกล
3. กังหันลมขนาดกลาง (Medium Wind Turbine) มีขนาดกำลังผลิตไฟฟ้าอยู่ระหว่าง 20-200 กิโลวัตต์ เหมาะสำหรับติดตั้งผลิตไฟฟ้าในระบบผสมผสานกับการผลิตไฟฟ้าชนิดอื่น เช่น ระบบผสมผสานดีเซล-เซลล์แสงอาทิตย์-กังหันลม เพื่อใช้ในระบบ Minigrid ตามชุมชนห่างไกล
4. กังหันลมขนาดใหญ่ (Large Wind Turbine) มีขนาดกำลังผลิตไฟฟ้าอยู่ระหว่าง 200-1,500 กิโลวัตต์ เหมาะสำหรับติดตั้งผลิตไฟฟ้าแบบทุ่งกังหันลมบนฝั่ง เพื่อเชื่อมต่อไฟฟ้าเข้ากับระบบสายส่ง (Grid Connection)
5. กังหันลมขนาดใหญ่มาก (Very Large Wind Turbine) มีขนาดกำลังผลิตมากกว่า 1,500 กิโลวัตต์ เหมาะสำหรับติดตั้งผลิตไฟฟ้าแบบทุ่งกังหันลมบนฝั่งและนอกชายฝั่ง เพื่อเชื่อมต่อไฟฟ้าเข้ากับระบบสายส่ง (Grid Connection)

## 2.7 แบตเตอรี่

แบตเตอรี่คือแหล่งจ่ายชนิดหนึ่งที่ผลิตพลังงานเป็นรูปแบบไฟฟ้ากระแสตรง โดยจะผลิตกระแสไฟฟ้าด้วยปฏิกิริยาเคมี สามารถที่จะรับการประจุพลังงานเข้าจากแหล่งจ่ายภายนอก และเป็นตัวจ่ายพลังงานไฟฟ้าให้กับอุปกรณ์หรือระบบ แบตเตอรี่จะมีหน้าที่หลักในการสำรองพลังงาน ดังนั้นการจ่ายพลังงานในระบบไฟฟ้ากระแสตรงในปัจจุบันนี้ จึงมีการนำแบตเตอรี่มาเป็นแหล่งจ่ายพลังงานสำรองให้แก่ระบบทางไฟฟ้า โดยเฉพาะในช่วงเวลาที่แหล่งจ่ายพลังงานหลักมีเหตุขัดข้อง หรือในสภาวะของระบบที่ทำหน้าที่การจ่ายพลังงานมีความไม่คงที่

### 2.7.1 ชนิดของแบตเตอรี่

แบตเตอรี่สามารถแบ่งชนิดออกตามลักษณะของการผลิตที่ส่งออกมาจากโรงงานผลิตอยู่ 2 ชนิด คือ



1. ชนิดประจุแห้ง แบตเตอรี่ชนิดนี้มีประจุไฟฟ้าเต็มมาจากโรงงานผลิต โดยยังไม่มี การเติมน้ำยาจึงมีสภาพแห้ง และเก็บไว้ใช้ได้ไม่นาน เมื่อต้องการนำไปใช้งานสามารถเติมน้ำยาแล้วทำการประจุไฟฟ้า การประจุไฟมักจะทำที่ตัวแทนจำหน่ายหรือในสถานประกอบการ
2. ชนิดประจุเปียก แบตเตอรี่ชนิดนี้มีการประจุไฟฟ้าเต็มจากโรงงานผลิต เติมน้ำยาแล้วจึงเป็นสภาพเปียก เก็บไว้ได้ไม่นาน ถ้าหากเก็บไว้นานเกินไปจะทำให้เกิดเกลือกำมะถันไปจับที่แผ่นตะกั่วมาก และจะหลุดได้ยากเพราะจะเกิดการจับอย่างถาวร จึงทำให้มีพื้นที่ในการทำปฏิกิริยาของแผ่นตะกั่ว(Plates)จะเหลือน้อยลง มีผลทำให้ประสิทธิภาพในการทำงานของแบตเตอรี่ลดลง

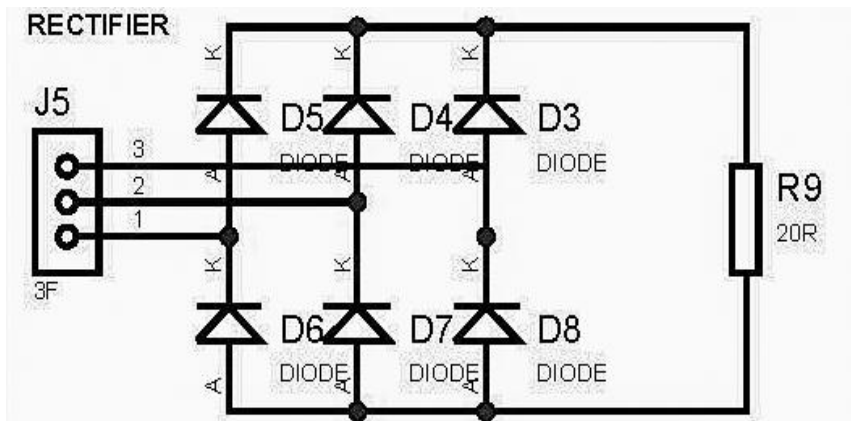
### 2.7.2 หลักการทำงานของแบตเตอรี่

แบตเตอรี่จะผลิตพลังงานไฟฟ้า ได้จากการทำปฏิกิริยาทางเคมีระหว่างสารเคลือบแผ่นตะกั่วและน้ำยาอิเล็กโทรไลต์ เมื่อมีการเกิดปฏิกิริยาเคมีพร้อมทั้งมีการต่อโหลด (Load) ครบวงจร แบตเตอรี่จะทำการคายประจุ หรือทำการกระจายกระแสไฟฟ้าให้กับโหลดเมื่อมีการเกิดปฏิกิริยาเคมีระหว่างกรดกำมะถันเจอกับแผ่นตะกั่วมีการต่อโหลดเข้ากับแบตเตอรี่จะทำให้แบตเตอรี่มีการคายประจุหรือจ่ายพลังงานไฟฟ้า และการเกิดปฏิกิริยาเคมีเช่นนี้เรื่อยๆตัวกรดกำมะถันจะค่อยๆ เจือจางลง และกลายเป็นน้ำธรรมดา ซึ่งมีผลทำให้เกิดค่าความถ่วงจำเพาะ(อัตราส่วนน้ำหนักของน้ำยาอิเล็กโทรไลต์กับน้ำมีปริมาณเท่ากันที่อุณหภูมิ 27 องศาเซลเซียส) ของกรดกำมะถันลดลง แบตเตอรี่หลังจากการนำไปใช้งานในระยะหนึ่ง พลังงานไฟฟ้าภายในแบตเตอรี่จะลดลงจึงจำเป็นต้องประจุไฟฟ้าใหม่จากแหล่งจ่ายภายนอก ขณะการประจุไฟฟ้าเข้าแบตเตอรี่ เกลือกำมะถันที่เกาะอยู่กับแผ่นตะกั่ว จะแยกตัวออกและจะละลายปนไปกับน้ำทำให้น้ำกรดมีความเข้มข้นเพิ่มขึ้นทำให้ค่าความถ่วงจำเพาะเพิ่มขึ้น แบตเตอรี่เมื่อมีการประจุเข้าจากแหล่งจ่าย หรือแหล่งกำเนิดพลังงานภายนอก ก็สามารถที่จะทำการจ่ายพลังงานได้ในเวลาพร้อมกัน

### บทที่ 3

#### การออกแบบโครงงาน

โครงงานคอนเวอร์เตอร์แบบฟลูบริดจ์สำหรับเครื่องกำเนิดไฟฟ้าพลังงานลมขนาดเล็กมี  
โครงสร้างของวงจรดังนี้



ภาพที่ 3.1 แสดงวงจรเรียงกระแส

- เครื่องกำเนิดไฟฟ้าพลังงานลม แบบ 3 เฟส 200 W- 75 V
- Bridge Rectifier แบบ 3 เฟส ขนาด 100 V 5 A
- Full Bridge Converter 300 W 12 V
- Power Switch Driver
- PWM Controller

#### 3.1 คำนวณหาพิคคของไดโอดในวงจรเรียงกระแส [1]

คำนวณแรงดันไฟฟ้าด้านเข้าจากพิคคเครื่องกำเนิดไฟฟ้า 75 Vac- 200 W  
แรงดันไฟฟ้าที่ผ่านวงจรเรียงกระแส =  $\sqrt{2} \times 75 = 106$  (3.1)

คำนวณพลังงานด้านเข้า  $P_{in}$  ดังภาพที่ 3.1

$$P_{in} = \frac{P_{out}}{\eta} \quad (3.2)$$

$$= \frac{200}{0.95}$$

$$= 210 \text{ W}$$

คำนวณกระแสไฟฟ้า (I)

$$I = \frac{P_{in}}{V_{in}} \quad (3.3)$$

$$= \frac{(210)}{75}$$

$$= 2.8 \text{ A}$$

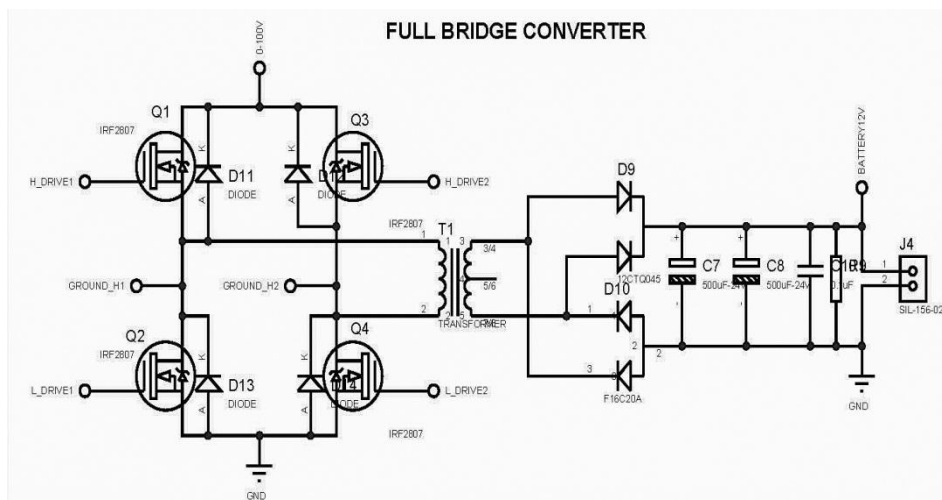
กระแสไฟฟ้าเฉลี่ยของไดโอดจะเท่ากับ

$$I_{D,av} = \frac{1}{2} I_{o,av} \quad (3.4)$$

$$= \frac{1}{2} \times 2.8$$

$$= 1.4 \text{ A}$$

ดังนั้นพิกัดของไดโอดที่เลือกใช้จะเท่ากับ 200 V - 2 A



ภาพที่ 3.2 แสดงวงจรคอนเวอร์เตอร์แบบฟูลบริดจ์

การคำนวณหาขนาดของมอสเฟตกำลัง

$$I_{D,av} = \frac{1}{2} I_{o,av} \quad (3.5)$$

$$= \frac{1}{2} \times 2.8$$

$$= 1.4 \text{ A}$$

### 3.2 การออกแบบตัวเหนี่ยวนำ

คำนวณพลังงานด้านเข้า ( $P_{in}$ )

$$P_{in} = \frac{P_{out}}{\eta} \quad (3.6)$$

$$= \frac{200}{0.95}$$

$$= 210 \text{ W}$$

คำนวณกระแสไฟฟ้าที่ไหลผ่านขดลวด

$$\begin{aligned} I &= \frac{P_{in}}{V_{in}} & (3.7) \\ &= \frac{(210)}{75} \\ &= 2.8 \text{ A} \end{aligned}$$

คำนวณอัตราระลอกคลื่นกระแส ( $\Delta I$ )

$$\begin{aligned} \Delta I &= 0.2I & (3.8) \\ &= 0.2(2.8) \\ &= 0.56 \text{ A} \end{aligned}$$

คำนวณจำนวนรอบการทำงานสูงสุด  $D_{(max)}$

$$\begin{aligned} D_{(max)} &= \left( \frac{V_{in}(\sqrt{2}) - (V_{out})}{V_{in}(\sqrt{2})} \right) & (3.9) \\ &= \left( \frac{106 - 12}{106} \right) \\ &= 0.89 \end{aligned}$$

คำนวณค่าตัวเหนี่ยวนำที่เหมาะสม (L)

$$\begin{aligned}
 L &= \left( \frac{(V_{in} \sqrt{2}) D_{(max)}}{(\Delta If)} \right) & (3.10) \\
 &= \left( \frac{(75 \times \sqrt{2}) \times (0.89)}{(0.56 \times 50,000)} \right) \\
 &= 3.371 \text{ mH}
 \end{aligned}$$

### 3.3 การพันค่าความเหนี่ยวนำ (L) มีหน่วยเป็นเฮนรี่

1H (เฮนรี่) หมายถึง การเปลี่ยนแปลงกระแส 1A/วินาที ทำให้เกิดแรงดันไฟฟ้า 1 Volte แกนเหล็กใช้กับความถี่สูง 50 Hz – 20 KHz แกนผงเหล็กอัดหรือแกนเฟอร์ไรต์ใช้กับความถี่สูง 10KHz-0.5MHz

$$L = \frac{\mu(N_2A)}{l}$$

เมื่อ

L	=	ค่าความเหนี่ยวนำ (H)
N	=	จำนวนรอบของขดลวด
A	=	พื้นที่หน้าตัดของแกนที่พันขดลวด (m <sup>2</sup> )
$\mu$	=	ค่าความซึมซาบได้ของชนิดของแกน
l	=	ความยาวของแกนที่พันขดลวด (m)

ค่าความซึมซาบได้ของแกนวัสดุชนิดต่างๆ ( $\mu$ )

อากาศหรือสูญญากาศ  $1.26 \times 10^{-5}$

นิกเกิล  $6.28 \times 10^{-5}$

เดบอลด์  $7.56 \times 10^{-5}$

เหล็กหล่อ  $1.1 \times 10^{-4}$

เหล็กแท่ง  $5.56 \times 10^{-4}$

แกนเหล็กของหม้อแปลงไฟฟ้า  $6.9 \times 10^{-3}$

เหล็กซิลิกอน (เฟอร์ไรต์)  $8.8 \times 10^{-3}$

เฟอร์มาลอย 0.126 ซุปเปอร์เฟอร์มาลอย 1.26

### 3.4 การพันขดลวดหม้อแปลง switching หรือหม้อแปลงต่างๆ

$$E_1 = 150 \text{ V}$$

$$F = 50000 \text{ Hz}$$

$$A = 7.85E^{-05} \text{ m}^2$$

$$\theta = 1.2 \text{ T}$$

$$\theta = 1.2 \text{ T} \quad (\text{คือค่าเส้นแรงแม่เหล็กที่พื้นที่ 1 ตารางเมตรของแกนเหล็ก}$$

หม้อแปลงส่วนแกนเฟอร์ไรต์มีค่ามากกว่าก็สามารถใช้ค่านี้นำนวนได้)

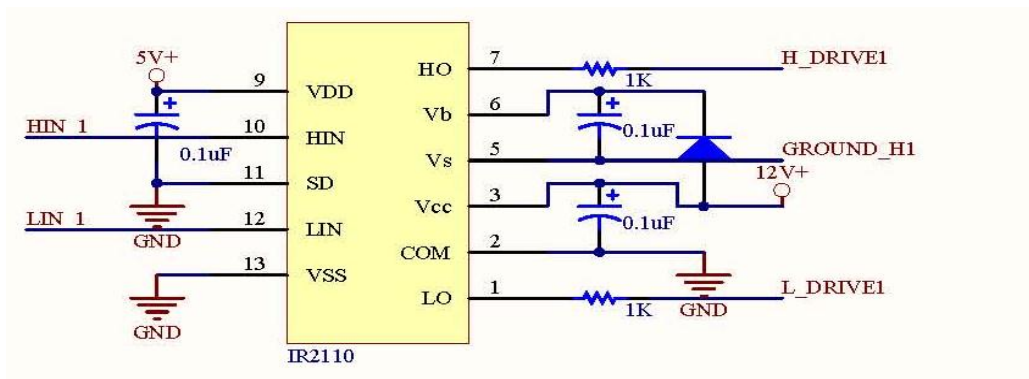
### 3.5 การพันหม้อแปลงขด Primary

$$N_1 = \frac{E_1}{(4.44 \times f \times A \times \theta)}$$

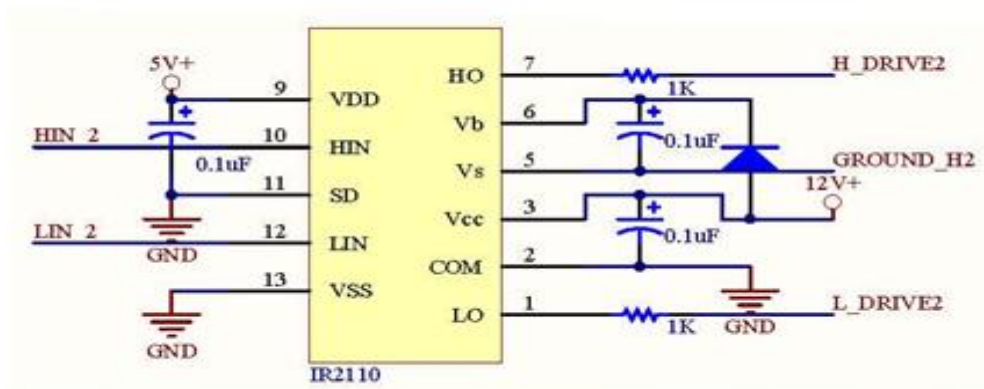
$$= 7.2 \text{ รอบ}$$

$$N_2 = N_1 \times \frac{E_2}{E_1}$$

ขนาดของลวดที่จะใช้พันต้องสอดคล้องกันกับจำนวนรอบที่คำนวณได้กับขนาดของแกนที่นำมาพันถ้าเราเลือกขนาดลวดใหญ่ไปก็จะพันรอบไม่ได้ (รอบที่คำนวณได้นั้นสามารถพันเกินได้แต่พันน้อยกว่าที่ไม่ได้เพราะมันมีผลกับคุณสมบัติของแกนด้วย) ถ้าเราเลือกขนาดของลวดเล็กไปก็ไม่มีผลในการพันจำนวนรอบแต่ว่ากระแสที่ได้จะถูกจำกัดด้วยขนาดของลวดต้องเลือกให้สอดคล้องกับขนาดของแกนที่พันความร้อนในขดลวด  $= I^2 \cdot R$  (Watt) สำหรับหม้อแปลงทั่วไปส่วนความร้อนในวงจร Switching ที่เกิดจากขดลวดแทบจะไม่มีเพราะความต้านทานในลวดต่ำแต่ความร้อนจะเกิดจากแกนมากกว่า เพราะใช้ความถี่สูงทำให้เกิดกระแสไหลวนภายในแกนสูง

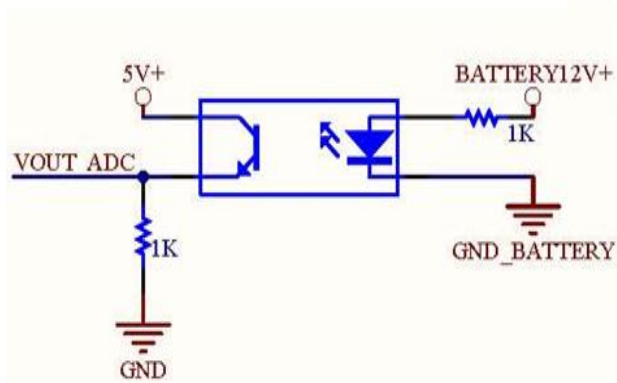


ภาพที่ 3.3 แสดงตัวควบคุมวงจรฟลูบริดจ์ตัวที่ 1

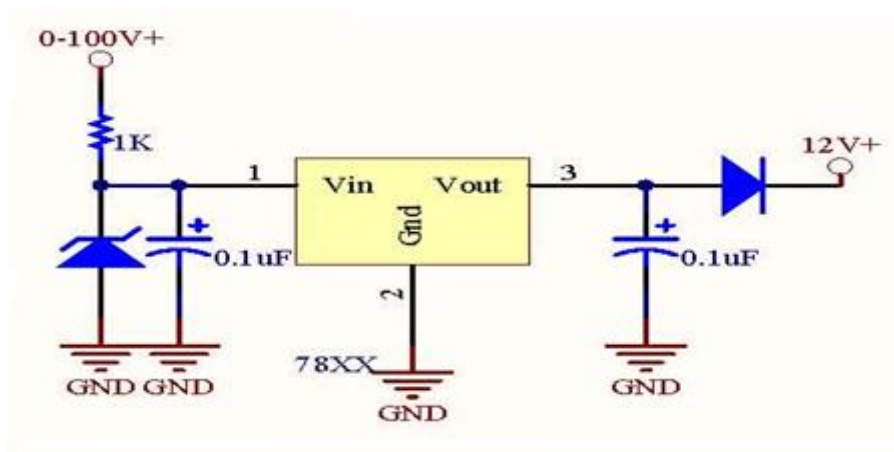


ภาพที่ 3.4 แสดงตัวควบคุมวงจรฟลูบริดจ์ตัวที่ 2

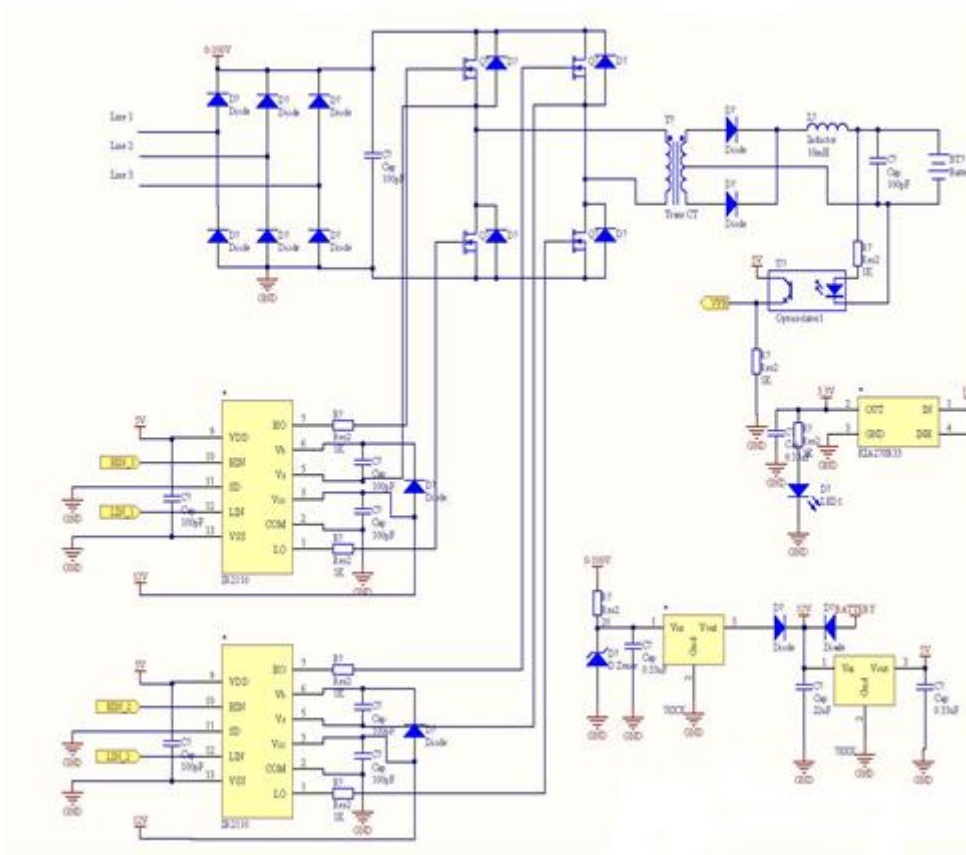




ภาพที่ 3.5 แสดงวงจรเช็คแรงดันฝั่งเอาต์พุต



ภาพที่ 3.6 แสดงวงจรลดแรงดันลงให้เหลือ 12 V



ภาพที่ 3.7 แสดงวงจรที่เชื่อมต่อเข้าด้วยกันทั้งหมด

## บทที่ 4

### การทดลองและผลการทดลอง

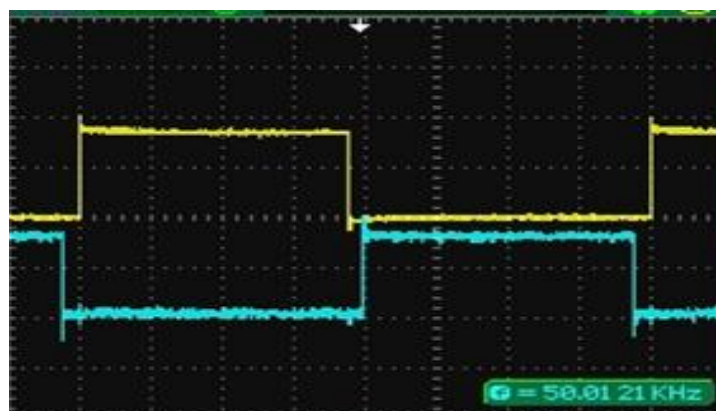
การทดลองโครงงานโดยแบ่งการดำเนินงานเป็น 2 ส่วน คือ 1.แสดงผลการทดลอง 2.การเขียนไมโครคอนโทรลเลอร์หรือโปรแกรมในการควบคุม

#### 4.1 การทดลองกับโหลดที่เป็นความต้านทาน และผลการทดลอง

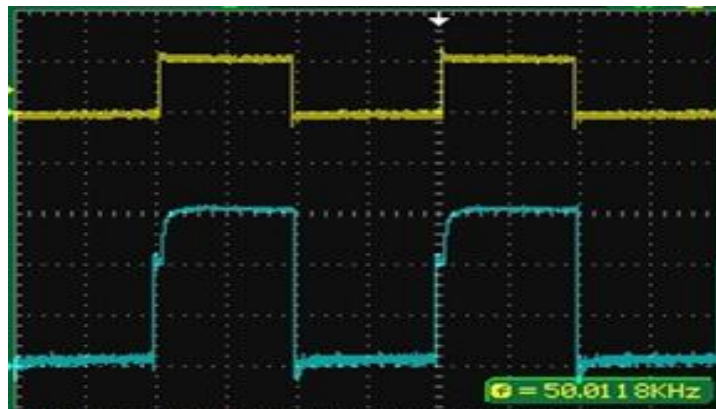
อุปกรณ์การทดลอง

1. ชุดคอนเวอร์เตอร์ฟลูบริดจ์
2. มิเตอร์ไฟฟ้า 2 ตัว
3. ออสซิลโลสโคป
4. แหล่งจ่ายไฟฟ้ากระแสสลับ
5. ชุดไมโครคอนโทรลเลอร์
6. สายไฟ
7. แบตเตอรี่ 12 V
8. หลอดไฟขนาด 55 W

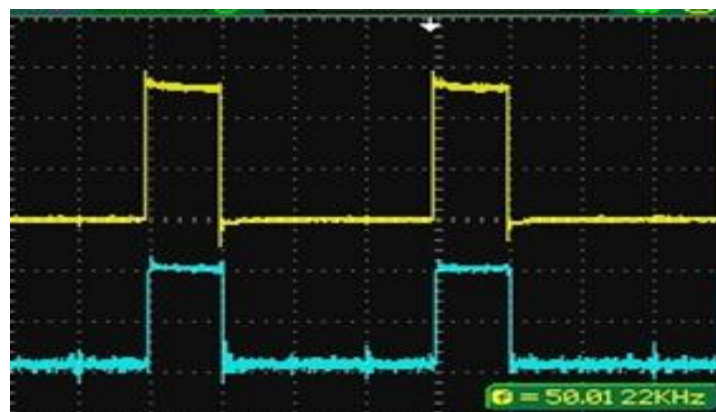
#### 4.2 แสดงผลการทดลองวงจร



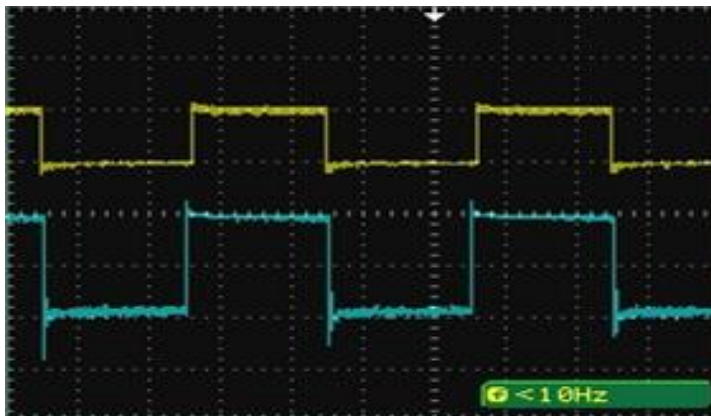
ภาพที่ 3.8 สัญญาณ PWM ช่วงเวลาที่เปิด Duty Cycle 47 %



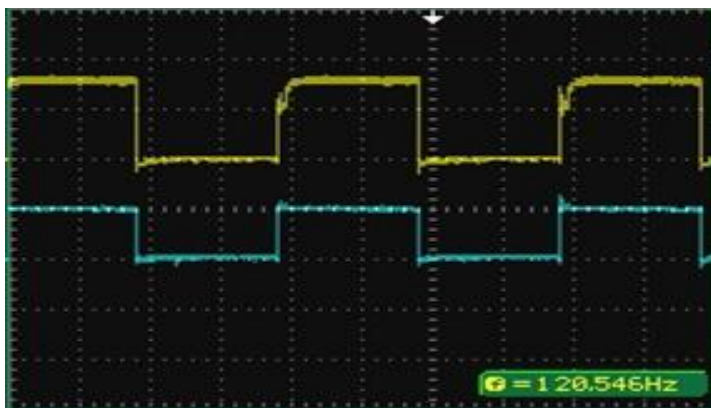
ภาพที่ 3.9 สัญญาณ PWM เทียบกับสัญญาณ Hi Size Duty Cycle 47 %



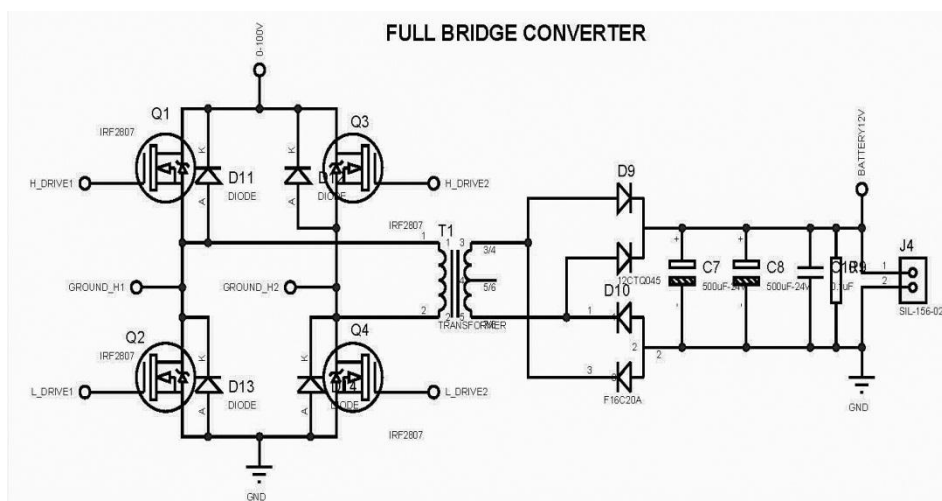
ภาพที่ 4.10 สัญญาณ PWM เทียบกับสัญญาณ Lo Size Duty Cycle 47 %



ภาพที่ 4.11 สัญญาณ PWM เทียบกับสัญญาณ Out Put Duty Cycle 47 %



ภาพที่ 4.12 สัญญาณ Hi Size กับ หม้อแปลง Duty Cycle 47 %



ภาพที่ 4.13 วงจรฟลูบริดจ์คอนเวอร์เตอร์

### 4.3 การเขียนไมโครคอนโทรลเลอร์

```
uint16_t CCR1Val = 200 , CCR1_Val = 30000 ,Duty_Cycle;
uint16_t PrescalerValue = 0,PrescalerValue1 = 0;
__IO uint16_t ADCConvertedValue;
float Voltage;
int Page_Number_LDC=1;
char Str_ADCConvertedValue[14],Str_Voltage[14],Str_Duty_Cycle[24];
/* Private function prototypes -----*/
void RCC_Configuration(void);
void GPIO_Configuration(void);
void NVIC_Configuration(void);
void TIM_Configuration(void);
void SETUP_TIM2(void);
void SETUP_TIM3(void);
void SETUP_TIM4(void);
void RCC_Configuration(void);
void SETUP_ADC(void);
```

```

void SETUP_USART(void);
void START_PWM(void);
void UPDATA_PWM_DUTY_CYCLE(void);
void Delay(__IO uint32_t nCount);
void FIRST_START(void);
void LCD_MENU(void);
void PI_CONTROL(void);
void usart1_putc(unsigned char c);
void usart1_puts(unsigned char *s);
int usart1_getc();
/* Private functions -----*/
/**
 * @brief Main program
 * @param None
 * @retval None
 */
int main(void)
{
    /*!< At this stage the microcontroller clock setting is already configured,
    this is done through SystemInit() function which is called from startup
    file (startup_stm32f10x_xx.s) before to branch to application main.
    To reconfigure the default setting of SystemInit() function, refer to
    system_stm32f10x.c file
    */
    /*RCC_Configuration*/
    RCC_Configuration();
    /*NVIC_Configuration*/
    NVIC_Configuration();
    /* GPIO Configuration */
    GPIO_Configuration();

```

```

/* Set TIM3,4 Configuration */
TIM_Configuration();
SETUP_TIM2();
/* Setup tim3 */
SETUP_TIM3();
/* Setup tim3 */
SETUP_TIM4();
/* Setup ADC1_IN14 */
SETUP_ADC();
/* Setup Usart */
SETUP_USART();
/* Start PWM */
START_PWM();
FIRST_START();

while (1)
{
LCD_MENU();
PI_CONTROL();
//          CCR1Val = ADCConvertedValue / 18;
//          UPDATA_PWM_DUTY_CYCLE(); //Updata Duty cycle
usart1_putc(Page_Number_LDC); // Send character from receive
}
}

void RCC_Configuration(void)
{
/* PCLK1 = HCLK/4 */
RCC_PCLK1Config(RCC_HCLK_Div4);
/* Enable peripheral clocks -----*/
/* Enable DMA1 clock */

```



```

RCC_AHBPeriphClockCmd(RCC_AHBPeriph_DMA1, ENABLE);

/* Enable TIM3 and GPIOC clock */

RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA|RCC_APB2Periph_GPIOC
|RCC_APB2Periph_GPIOD |RCC_APB2Periph_GPIOE |RCC_APB2Periph_ADC1
|RCC_APB2Periph_AFIO, ENABLE);

RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM2 |RCC_APB1Periph_TIM3
|RCC_APB1Periph_TIM4, ENABLE);
}

void GPIO_Configuration(void)
{
GPIO_InitTypeDef GPIO_InitStructure;

/* Remap TIM13_CH1 and TIM4_CH2 */

GPIO_PinRemapConfig(GPIO_FullRemap_TIM3 |GPIO_Remap_TIM4, ENABLE);

/* GPIOC Configuration: TIM3 Channel1 as alternate function push-pull */

GPIO_InitStructure.GPIO_Pin = GPIO_Pin_6;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;

GPIO_Init(GPIOC, &GPIO_InitStructure);

/* Configure PC.04 (ADC Channel1) as analog input -----*/

GPIO_InitStructure.GPIO_Pin = GPIO_Pin_3;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AIN;
GPIO_Init(GPIOC, &GPIO_InitStructure);

/* GPIOC Configuration: TIM4 Channel12 as alternate function push-pull */

GPIO_InitStructure.GPIO_Pin = GPIO_Pin_13;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_Init(GPIOD, &GPIO_InitStructure);

/* GPIOD Configuration: JoyStick */

GPIO_InitStructure.GPIO_Pin = GPIO_Pin_8 |GPIO_Pin_12 |GPIO_Pin_14;

```

```

GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_Init(GPIOD, &GPIO_InitStructure);
/* GPIOE Configuration: JoyStick */
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0 |GPIO_Pin_1;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_Init(GPIOE, &GPIO_InitStructure);
/* GPIOE Configuration:Pin2 as alternate function push-pull */
GPIO_InitStructure.GPIO_Pin = GPIO_Pin_2;
GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
GPIO_Init(GPIOE, &GPIO_InitStructure);
}
void NVIC_Configuration(void)
{
NVIC_InitTypeDef NVIC_InitStructure;
/* Enable the TIM2 global Interrupt */
NVIC_InitStructure.NVIC_IRQChannel = TIM2_IRQn;
NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
NVIC_InitStructure.NVIC_IRQChannelSubPriority = 1;
NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
NVIC_Init(&NVIC_InitStructure);
}
void TIM_Configuration(void)
{
/* -----
TIM3,4 Configuration: generate 1 PWM signal.
The TIM10CLK frequency is set to SystemCoreClock (72 MHz), to get TIM3 counter
clock at 24 MHz the Prescaler is computed as following:

```

```

- Prescaler = (TIM3CLK / TIM3 counter clock) - 1

The TIM3 is running at 36 KHz: TIM3 Frequency = TIM3 counter clock/(ARR + 1)
= 24 MHz / 480 = 50 KHz

TIM3 Channel1 duty cycle = (TIM3_CCR1/ TIM3_ARR)* 100 = 37.5%
----- */

/* Compute the prescaler value */
PrescalerValue = (uint16_t) (SystemCoreClock / 24000000) - 1;
/* Time base configuration */
TIM_TimeBaseStructure.TIM_Period = 479;
TIM_TimeBaseStructure.TIM_Prescaler = PrescalerValue;
TIM_TimeBaseStructure.TIM_ClockDivision = 0;
TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up;
TIM_TimeBaseInit(TIM4, &TIM_TimeBaseStructure);
TIM_TimeBaseInit(TIM3, &TIM_TimeBaseStructure);
//      /* Compute the prescaler value */
// PrescalerValue1 = (uint16_t) (SystemCoreClock / 12000000) - 1;
// /* Time base configuration */
// TIM_TimeBaseStructure.TIM_Period = 65535;
// TIM_TimeBaseStructure.TIM_Prescaler = 0;
// TIM_TimeBaseStructure.TIM_ClockDivision = 0;
// TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up;

TIM_TimeBaseInit(TIM2, &TIM_TimeBaseStructure);
/* Prescaler configuration */
TIM_PrescalerConfig(TIM2, PrescalerValue, TIM_PSCReloadMode_Immediate);
}
void SETUP_TIM2(void)
{
/* Output Compare Timing Mode configuration: Channel1 */
TIM_OCInitStructure.TIM_OCMode = TIM_OCMode_Timing;

```

```
TIM_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable;
TIM_OCInitStructure.TIM_Pulse = CCR1_Val;
TIM_OCInitStructure.TIM_OCPolarity = TIM_OCPolarity_High;
TIM_OC1Init(TIM2, &TIM_OCInitStructure);
TIM_OC1PreloadConfig(TIM2, TIM_OCPreload_Disable);
/* TIM IT enable */
TIM_ITConfig(TIM2, TIM_IT_CC1, ENABLE);
/* TIM2 enable counter */
TIM_Cmd(TIM2, ENABLE);
}

void SETUP_TIM3(void)
{
TIM_TimeBaseInit(TIM3, &TIM_TimeBaseStructure);
/* PWM1 Mode configuration: Channel1 */
TIM_OCInitStructure.TIM_OCMode = TIM_OCMode_PWM1;
TIM_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable;
TIM_OCInitStructure.TIM_Pulse = CCR1_Val;
TIM_OCInitStructure.TIM_OCPolarity = TIM_OCPolarity_High;
TIM_OC1Init(TIM3, &TIM_OCInitStructure);
TIM_OC1PreloadConfig(TIM3, TIM_OCPreload_Enable);
TIM_ARRPreloadConfig(TIM3, ENABLE);
}

void SETUP_TIM4(void)
{
TIM_TimeBaseInit(TIM4, &TIM_TimeBaseStructure);
/* PWM1 Mode configuration: Channel2 */
TIM_OCInitStructure.TIM_OCMode = TIM_OCMode_PWM1;
TIM_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable;
TIM_OCInitStructure.TIM_Pulse = CCR1_Val;
```

```

TIM_OCInitStruct.TIM_OCPolarity = TIM_OCPolarity_High;
TIM_OC2Init(TIM4, &TIM_OCInitStruct);
TIM_OC2PreloadConfig(TIM4, TIM_OCPreload_Enable);
TIM_ARRPreloadConfig(TIM4, ENABLE);
}

void SETUP_ADC(void)
{
/* DMA1 channel1 configuration -----*/
DMA_DeInit(DMA1_Channel1);
DMA_InitStructure.DMA_PeripheralBaseAddr = ADC1_DR_Address;
DMA_InitStructure.DMA_MemoryBaseAddr = (uint32_t)&ADCConvertedValue;
DMA_InitStructure.DMA_DIR = DMA_DIR_PeripheralSRC;
DMA_InitStructure.DMA_BufferSize = 1;
DMA_InitStructure.DMA_PeripheralInc = DMA_PeripheralInc_Disable;
DMA_InitStructure.DMA_MemoryInc = DMA_MemoryInc_Disable;
DMA_InitStructure.DMA_PeripheralDataSize = DMA_PeripheralDataSize_HalfWord;
DMA_InitStructure.DMA_MemoryDataSize = DMA_MemoryDataSize_HalfWord;
DMA_InitStructure.DMA_Mode = DMA_Mode_Circular;
DMA_InitStructure.DMA_Priority = DMA_Priority_High;
DMA_InitStructure.DMA_M2M = DMA_M2M_Disable;
DMA_Init(DMA1_Channel1, &DMA_InitStructure);
/* Enable DMA1 channel1 */
DMA_Cmd(DMA1_Channel1, ENABLE);
/* ADC1 configuration -----*/
ADC_InitStructure.ADC_Mode = ADC_Mode_Independent;
ADC_InitStructure.ADC_ScanConvMode = ENABLE;
ADC_InitStructure.ADC_ContinuousConvMode = ENABLE;
ADC_InitStructure.ADC_ExternalTrigConv = ADC_ExternalTrigConv_None;
ADC_InitStructure.ADC_DataAlign = ADC_DataAlign_Right;
ADC_InitStructure.ADC_NbrOfChannel = 1;

```

```
ADC_Init(ADC1, &ADC_InitStructure);
/* ADC1 regular channel14 configuration */
ADC_RegularChannelConfig(ADC1, ADC_Channel_3, 1, ADC_SampleTime_55Cycles5);
/* Enable ADC1 DMA */
ADC_DMACmd(ADC1, ENABLE);
/* Enable ADC1 */
ADC_Cmd(ADC1, ENABLE);
/* Enable ADC1 reset calibration register */
ADC_ResetCalibration(ADC1);
/* Check the end of ADC1 reset calibration register */
while(ADC_GetResetCalibrationStatus(ADC1));
/* Start ADC1 calibration */
ADC_StartCalibration(ADC1);
/* Check the end of ADC1 calibration */
while(ADC_GetCalibrationStatus(ADC1));
/* Start ADC1 Software Conversion */
ADC_SoftwareStartConvCmd(ADC1, ENABLE);
}
void SETUP_USART(void)
{
/* USARTx configured as follow:
- BaudRate = 115200 baud
- Word Length = 8 Bits
- One Stop Bit
- No parity
- Hardware flow control disabled (RTS and CTS signals)
- Receive and transmit enabled */
USART_InitStructure.USART_BaudRate = 115200;
USART_InitStructure.USART_WordLength = USART_WordLength_8b;
USART_InitStructure.USART_StopBits = USART_StopBits_1;
```

```

USART_InitStructure.USART_Parity = USART_Parity_No;
USART_InitStructure.USART_HardwareFlowControl = USART_HardwareFlowControl_None;
USART_InitStructure.USART_Mode = USART_Mode_Rx | USART_Mode_Tx;
STM_EVAL_COMInit(COM1, &USART_InitStructure);
}

void START_PWM(void)
{
/* TIM3 enable counter */
TIM_Cmd(TIM3, ENABLE);
Delay(0x00035);
/* TIM4 enable counter */
TIM_Cmd(TIM4, ENABLE);
}

void UPDATA_PWM_DUTY_CYCLE(void)
{
TIM_SetCompare2(TIM4,CCR1Val);
TIM_SetCompare1(TIM3,CCR1Val);
}

void Delay(__IO uint32_t nCount)
{
while(nCount--)
{
}
}

void FIRST_START(void)
{
char PASS,loop;
loop=1;
usart1_puts("STM32F103VBT6 ON!\r");
usart1_puts("PassWord :\r");
}

```

```

while(loop==1)
{
PASS = usart1_getc(); // Wait until receive character
if(PASS==0x53)
{
PASS = usart1_getc(); // Wait until receive character
if(PASS==0x50)
{
PASS = usart1_getc(); // Wait until receive character
if(PASS==0x55)
{
usart1_puts("\rOK\r");
loop=0;
}
}
}
void LCD_MENU(void)
{
if ((GPIO_ReadInputDataBit(GPIOE, GPIO_Pin_0) == 0)&&(Page_Number_LDC < 3))
{ Page_Number_LDC++;}
else if ((GPIO_ReadInputDataBit(GPIOE, GPIO_Pin_1) == 0)&&(Page_Number_LDC > 1))
{ Page_Number_LDC--;}
if(Page_Number_LDC == 1)
{
Voltage = (float)ADCConvertedValue / 1240;
Duty_Cycle = ((float)CCR1Val / 479) *100;
sprintf(Str_ADCConvertedValue,"AIN14: %d  ",ADCConvertedValue); // Convert analog
value to string
sprintf(Str_Voltage,"Voltage: %.1f V  ",Voltage); // Convert analog value to string
sprintf(Str_Duty_Cycle,"Duty Cycle: %d  ",Duty_Cycle); // Convert analog value to string
lcdString(1,1,Str_ADCConvertedValue); // Set message on line 1
lcdString(1,2,Str_Voltage); // Set message on line 2
}
}
}

```



```

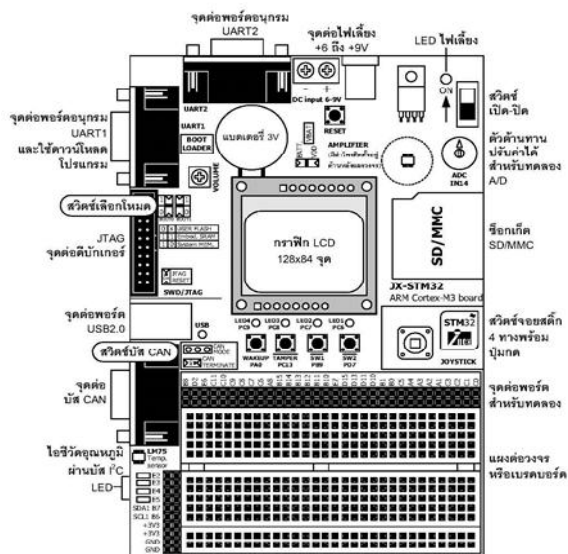
lcdString(1,3,Str_Duty_Cycle); // Set message on line 3
lcdUpdate(); // Display message
}
else if(Page_Number_LDC != 1)
{
lcdClear();
lcdUpdate(); // Display message
}
}

void PI_CONTROL(void)
{
int PI,ERROR;
PI = ADCConvertedValue/10;
ERROR = 29-PI;
CCR1Val = CCR1Val+ERROR;
if(CCR1Val>227)
{
CCR1Val = 227;
}
UPDATA_PWM_DUTY_CYCLE(); //Updata Duty cycle
}
//-----//
//----- Function USART1 send 1 character -----//
//-----//

void usart1_putc(unsigned char c)
{
// Wait until transmission ready
while(USART_GetFlagStatus(USART1,USART_FLAG_TXE)==RESET);
USART_SendData(USART1,(int)c); // Send character
}

```

```
//-----//  
//----- Function USART1 send string -----//  
//-----//  
void usart1_puts(unsigned char *s)  
{  
while(*s) // Check end of string  
{  
usart1_putc(*s++); // Send charracter 1 time  
}  
}  
//-----//  
//----- Function USART1 wait character -----//  
//-----//  
int usart1_getc()  
{  
// Wait until receive data  
while(USART_GetFlagStatus(USART1,USART_FLAG_RXNE)==RESET);  
return(USART_ReceiveData(USART1)); // Return character  
}
```



ภาพที่ 4.14 ชุดไมโครคอนโทรลเลอร์

### 4.4 สรุปผลการทดลอง

ตารางที่ 4.1 แสดงผลของการทดลอง

Load (W)	Vin (V)	Iin (A)	Pin (V)	Vout (V)	Iout (A)	Pout (W)	$\eta$ (%)
แบตเตอรี่	30	1.02	30.6	13.4	0.84	11.25	36.76
	40	1.24	49.6	13.37	0.99	13.23	26.67
	50	1.63	81.5	13.67	1.64	22.4	27.48
10 โอห์ม	30	1.21	36.3	13.27	0.97	12.87	35.45
	40	2.16	86.4	13.71	1.41	19.33	22.37
	50	2.37	118.5	13.8	1.74	24.01	20.26
5 โอห์ม	30	0.86	43	14.63	2.12	31.01	72.12
	40	0.91	36.4	14.45	2.07	29.91	77.36
	50	1.47	44.1	13.54	2.08	28.16	63.85
โหลด 55 W	60	4.48	213.9	13.36	3.83	50.3	23.52

## บทที่ 5

### สรุป

จากการวิเคราะห์ผลการทำงานของคอนเวอร์เตอร์ฟลูบริคส์สำหรับกังหันลมขนาดเล็ก สามารถนำมาใช้งานได้จริงแล้วได้ผลตามขอบเขตที่ตั้งไว้โดยสมบูรณ์และได้ทำการนำไปติดตั้งใช้ที่ตีวิศวกรรมศาสตร์เพื่อผลิตกระแสไฟฟ้านำมาให้ความสว่างแก่องค์พระวิทยุกรรมในการทำงานของวงจรคอนเวอร์เตอร์ฟลูบริคส์เป็นการลดแรงดันตั้งแต่ 13 V ถึง 14 V เพื่อการชาร์จแบตเตอรี่โดยใช้ไมโครคอนโทรลเลอร์ควบคุมการป้อนกลับแบบ PI คอนโทรล โดยใช้ความถี่ที่ 50 KHz

## เอกสารอ้างอิง

- [1] Colonel Wm.T.Mclyman “Transformer and Inductor design handbook” Kg Magnetic, Inc.  
Idyllwild, California, U.S.A, 2004
- [2] วิชชากร เสงศรีชวาช “เครื่องจักรกลไฟฟ้า 1” จัดพิมพ์โดย โรงพิมพ์ มหาวิทยาลัยศรีปทุม, 2549
- [3] รศ.ดร. วีระชัย ชันเงิน / วุฒิพล ชาราธิรเศรษฐ์ “อิเล็กทรอนิกส์กำลัง”  
ห.จ.ก วิ.เจ. พรินติ้ง, 2552
- [4] สุวัฒน์ คั่น “เทคนิคและการออกแบบสวิตชิงเพาเวอร์ซัพพลาย” จัดพิมพ์โดย บริษัท  
เอนเทลไทย จำกัด, 2537

**ภาคผนวก**



แสดงตู้วงจรคอนเวอร์เตอร์ฟลูบิรคค์

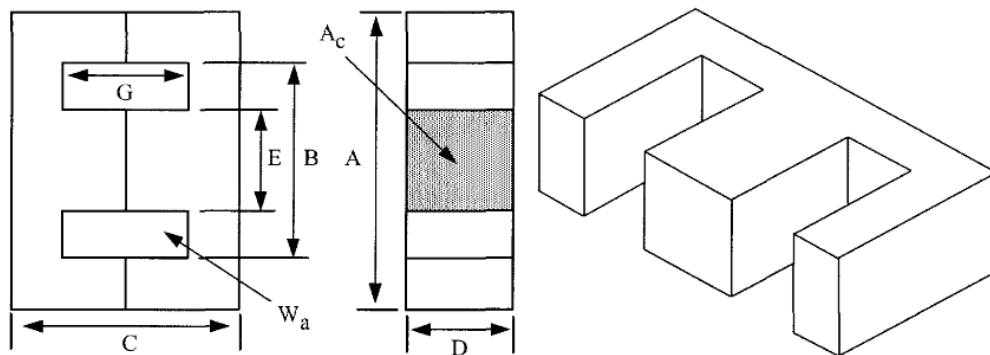
ตารางที่ 1.2 แสดงขนาดมาตรฐาน AWG และข้อมูลอื่นๆของลวดทองแดง

เบอร์ A W G (B.& S.)	ขนาดเส้นผ่าน ศูนย์กลาง (d )		ขนาดเส้นผ่านศูนย์กลางเมื่อ รวมฉนวน (do) mm	พื้นที่หน้าตัด ปกติ mm <sup>2</sup>	ค่าความต้านทานที่ 100 °C (Rdc) Ω/m	ระยะขีด ต่ำสุด (t <sub>min</sub> ) mm
	inch	mm				
44	0,00198	0,0503	0,06604	0,00199	11,180	0,071
43	0,00222	0,0564	0,07366	0,00250	8,899	0,079
42	0,00249	0,0633	0,08128	0,00314	7,073	0,087
41	0,00280	0,0711	0,09144	0,00397	5,594	0,098
40	0,00314	0,0798	0,1041	0,00500	4,448	0,111
39	0,00353	0,0897	0,1143	0,00631	3,519	0,122
38	0,00397	0,1008	0,1295	0,00799	2,783	0,138
37	0,00445	0,1130	0,1448	0,01003	2,215	0,154
36	0,00500	0,1270	0,1626	0,0127	1,754	0,172
35	0,0056	0,1422	0,1778	0,0159	1,398	0,188
34	0,0063	0,1600	0,1981	0,0201	1,105	0,209
33	0,0071	0,1803	0,2235	0,0255	0,8700	0,236
32	0,0080	0,2032	0,2489	0,0324	0,6853	0,261
31	0,0089	0,2261	0,2743	0,0401	0,5537	0,287
30	0,0100	0,2540	0,3048	0,0507	0,4386	0,319
29	0,0113	0,2870	0,3404	0,0647	0,3435	0,356
28	0,0126	0,3200	3,3759	0,0804	0,2762	0,393
27	0,0142	0,3607	0,4191	0,1022	0,2175	0,438
26	0,0159	0,4039	0,4699	0,128	0,1735	0,491
25	0,0179	0,4547	0,5232	0,162	0,1369	0,547
24	0,0201	0,5105	0,5817	0,205	0,10860	0,608
23	0,0226	0,5740	0,6502	0,259	0,08586	0,679
22	0,0253	0,6426	0,7214	0,324	0,06852	0,754
21	0,0285	0,7239	0,8052	0,412	0,05399	0,841
20	0,0320	0,8128	0,8966	0,519	0,04283	0,937
19	0,0359	0,9119	1,003	0,653	0,03403	1,048
18	0,0403	1,024	1,118	0,823	0,02700	1,168
17	0,0453	1,151	1,247	1,040	0,02137	1,303
16	0,0508	1,290	1,389	1,308	0,01699	1,452
15	0,0571	1,450	1,557	1,652	0,01345	1,627
14	0,0641	1,628	1,737	2,082	0,010670	1,815
13	0,0720	1,829	1,943	2,627	0,008460	2,030
12	0,0808	2,052	2,172	3,308	0,006717	2,270



11	0,0907	2,304	2,431	4,168	0,005331	2,540
10	0,1019	2,588	2,720	5,261	0,004224	2,842

ตารางที่ 1.3 แสดงตัวอย่างขนาดมาตรฐานของแกนเฟอร์ไรต์แบบ EE



EE, Ferrite Cores (Magnetics)											
Part No.	Wt <sub>cu</sub> grams	Wt <sub>fe</sub> grams	MLT cm	MPL cm	Wa AC	AC cm <sup>2</sup>	Wa cm <sup>2</sup>	Ap cm <sup>4</sup>	Kg cm <sup>5</sup>	At cm <sup>2</sup>	AL mh/1K
EE-187	6.8	4.4	3.8	4.01	2.219	0.228	0.506	0.116	0.0028	14.4	500
EE-2425	13.9	9.5	4.9	4.85	2.068	0.384	0.794	0.305	0.0095	23.5	767
EE-375	36.4	33.0	6.6	6.94	1.875	0.821	1.539	1.264	0.0624	45.3	1167
EE-21	47.3	57.0	8.1	7.75	1.103	1.490	1.643	2.448	0.1802	60.9	1967
EE-625	64.4	103.0	9.4	8.90	0.808	2.390	1.930	4.616	0.4700	81.8	2767
EE-75	111.1	179.0	11.2	10.70	0.826	3.390	2.799	9.487	1.1527	118.0	3467

