

การพัฒนาเครื่องกำเนิดกระแสเกินชั่วคราว

นางสาวศุภนิดา ชาญหัตถศิลป์

นางสาวชนาวดี แซ่เอ็ง

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตร

ปริญญาวិชาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

ปีการศึกษา 2558

# Development of Temporary Overcurrent Generator

Ms. Suphanida    Chanhatthasin

Ms. Thanavadee    Sae-oueng

A PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS  
FOR THE DEGREE OF BACHELOR OF ELECTRICAL ENGINEERING  
DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING  
FACULTY OF ENGINEERING  
KING MONGKUT'S UNIVERSITY OF TECHNOLOGY NORTH BANGKOK  
ACADEMIC YEAR 2015

ชื่อปริญญาบัตร : การพัฒนาเครื่องกำเนิดกระแสเกินชั่วคราว  
ชื่อ : นางสาวศุภนิตา ชาญหัตถศิลป์  
นางสาวธนาดี แซ่เอ็ง  
สาขาวิชา : วิศวกรรมไฟฟ้า  
ภาควิชา : วิศวกรรมไฟฟ้าและคอมพิวเตอร์  
คณะ : วิศวกรรมศาสตร์  
อาจารย์ที่ปรึกษา : รองศาสตราจารย์ไชยันต์ สุวรรณชีวะศิริ  
ผู้ช่วยศาสตราจารย์ ดร.นภดล วิวัชร โกเศศ  
ปีการศึกษา : 2558

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ อนุมัติให้ปริญญาบัตรนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

..... หัวหน้าภาควิชาวิศวกรรมไฟฟ้า  
(ผู้ช่วยศาสตราจารย์ ดร. นภดล วิวัชร โกเศศ) และคอมพิวเตอร์  
..... ประธานกรรมการ  
(รองศาสตราจารย์ไชยันต์ สุวรรณชีวะศิริ)  
..... กรรมการ  
(ผู้ช่วยศาสตราจารย์ ดร. นภดล วิวัชร โกเศศ)  
..... กรรมการ  
(รองศาสตราจารย์ ดร.พิสิษฐ์ ถั่วชนกุล)

ลิขสิทธิ์ของภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์  
มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ

Project Report Title : Development of Temporary Overcurrent Generator

Name : Ms. Suphanida Chanhathasin  
Ms. Thanavadee Sae-oueng

Major Field : Electrical Engineering

Department : Electrical and Computer Engineering

Faculty : Engineering

Project Advisor(s) : Assoc. Prof. Chaiyan Suwanchewasiri  
Asst. Prof. Dr. Nophadon Wiwatcharagoses

Academic Year : 2015

Accepted by the Faculty of Engineering, King Mongkut's University of Technology North Bangkok in Partial Fulfillment of the Requirements for the Degree of Bachelor of Electrical Engineering

.....  
(Asst. Prof. Dr. Nophadon Wiwatcharagoses) Chairperson of Department of Electrical and Computer Engineering

.....  
(Assoc. Prof. Chaiyan SuwanCheewasiri) Chairperson

.....  
(Asst. Prof. Dr. Nophadon Wiwatcharagoses) Member

.....  
(Assoc. Prof. Dr. Pisit Liutanakul ) Member

Copyright of the Department of Electrical and Computer Engineering, Faculty of Engineering  
King Mongkut's University of Technology North Bangkok

## บทคัดย่อ

บทความนี้นำเสนอการศึกษา ออกแบบพัฒนา และสร้างเครื่องกำเนิดกระแสเกินชั่วคราว เพื่อที่จะสร้างกระแสเกินชั่วคราวสำหรับการทดสอบเครื่องวัดกระแสเกินชั่วคราว ในส่วนการออกแบบจะไม่โครคอนโทรลเลอร์ ในการควบคุมการส่งสัญญาณกระตุ้นให้ขาเกตของไทรแอกที่เป็นสวิตช์ให้วงจรภาคกำลัง และใช้อุปกรณ์เชื่อมต่อทางแสงกับหม้อแปลงพัลส์ เป็นอุปกรณ์แยกวงจรภาคกำลังกับภาคควบคุม โดยผลที่ได้จะเป็นกระแสปกติและกระแสเกินรูปคลื่นไซน์จำนวนลูกตามที่ใช้กำหนด หรือเป็นระยะเวลาในหน่วยวินาทีตามที่ใช้กำหนดเช่นกัน

## **Abstract**

This paper to present about studies design and construction of Temporary Overcurrent Generator. In order to generate a temporary overcurrent for testing temporary overcurrent measure. In part of design is used microcontroller to trigger and generate control signal for Triac, switch of power part. Used the Opto-isolator and Pulse Transformers for separate the controller from power circuit. The result is the number of sine waves or a period of seconds that come from user.

## กิตติกรรมประกาศ

ผู้จัดทำปริญญานิพนธ์ฉบับนี้ ขอกราบขอบพระคุณ รองศาสตราจารย์ไชยันต์ สุวรรณชีวะศิริและ คณะกรรมการในการควบคุมการสอบทุกท่าน ที่ให้คำปรึกษาแนวทางแก้ไขปัญหาข้อเสนอนั้นต่าง ๆ และให้แนวคิดริเริ่มที่ดี ขอบคุณอาจารย์ทุกท่านที่ประสิทธิ์ประสาทวิชาความรู้ซึ่งเป็นพื้นฐานที่สำคัญ อย่างยิ่งที่นำมาประยุกต์ใช้ในการทำโครงการนี้ ขอบคุณภาควิชาวิศวกรรม ไฟฟ้าและคอมพิวเตอร์ คณะ วิศวกรรมศาสตร์ มหาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ขอบคุณ บิดา มารดา และเพื่อน ๆ พี่ ๆ ทุกคน ที่คอยให้คำปรึกษา และเป็นกำลังใจที่ดีเสมอมา

ศุภนิดา ชาญหัตถศิลป์

ธนาวิดี แซ่ฮึ้ง

## สารบัญ

	หน้า
บทคัดย่อภาษาไทย	จ
บทคัดย่อภาษาอังกฤษ	ฉ
กิตติกรรมประกาศ	ช
สารบัญตาราง	ญ
สารบัญภาพ	ฎ
บทที่ 1. บทนำ	1
1.1 ที่มาและความสำคัญ	1
1.2 วัตถุประสงค์	2
1.3 ขอบเขตของการทำโครงการ	2
1.4 แนวเหตุผล ทฤษฎีสำคัญหรือสมมติฐาน	2
1.5 วัตถุประสงค์ของการทำโครงการ	3
บทที่ 2. ทฤษฎีพื้นฐานที่เกี่ยวข้อง	4
2.1 บทนำ	4
2.2 อาร์ดูโน (Arduino)	4
2.3 แอลซีดีดีสเพลย์ (LCD Display)	5
2.4 คีย์แพด (Keypad)	6
2.5 อุปกรณ์เชื่อมต่อทางแสง (Opto-Coupler)	7
2.6 หม้อแปลงไฟฟ้า (Transformer)	9
2.7 ไตรแอก (Triac)	11
2.8 วงจรตรวจจับจุดผ่านศูนย์ (Zero-crossing Detector)	13
บทที่ 3. ขั้นตอนและวิธีการดำเนินงาน	16
3.1 บล็อกไดอะแกรมรวมของเครื่องกำเนิดกระแสเกินชั่วคราว	16
3.2 การจำลองการทำงานส่วนต่าง ๆ ของเครื่องกำเนิดกระแสเกินชั่วคราว	16
3.3 วงจรทั้งหมดของเครื่องกำเนิดกระแสเกินชั่วคราว	26
บทที่ 4. ผลการดำเนินงาน	29
4.1 วงจรจริงของเครื่องกำเนิดกระแสเกินชั่วคราว	29



## สารบัญ (ต่อ)

	หน้า
4.2 การทดสอบการทำงานส่วนต่าง ๆ ของเครื่องกำเนิดกระแสเกินชั่วคราว	29
4.3 การทดสอบการทำงานของเครื่องกำเนิดกระแสเกินชั่วคราว	33
4.4 ผลการทดสอบการทำงานของเครื่องกำเนิดกระแสเกินชั่วคราว เมื่อนำไปใช้กับ Surge Counter	41
บทที่ 5. สรุปผล ปัญหาและข้อเสนอแนะ	43
5.1 สรุปผล	43
5.2 ปัญหา แนวทางแก้ไข และพัฒนา	43
5.3 ข้อเสนอแนะ	43
เอกสารอ้างอิง	44
ภาคผนวก ก การประกอบวงจร	46
ภาคผนวก ข โปรแกรมในไมโครคอนโทรลเลอร์	53
ประวัติผู้แต่ง	92

## สารบัญตาราง

ตารางที่	หน้า
3-1 รายการและราคาโดยประมาณของอุปกรณ์ที่ใช้ในโครงการ	28

## สารบัญภาพ

ภาพที่	หน้า
1-1 ลูกคลื่นไซน์ที่มีกระแสเกินเกิดขึ้นบนกระแสปกติ	1
1-2 บล็อกไดอะแกรมของเครื่องกำเนิดกระแสเกินชั่วคราว	3
2-1 บอร์ดอาคูน	5
2-2 บอร์ดอาคูนต่อกับอุปกรณ์เสริมต่าง ๆ (กับไดโอดเปล่งแสง)	5
2-3 จอแอลซีดี 16 ตัวอักษร 2 บรรทัด	6
2-4 ด้านหน้าของจอแอลซีดี 16 ตัวอักษร 2 บรรทัด	6
2-5 4x4 Matrix Keypad	7
2-6 บล็อกไดอะแกรมของออปโตคัปเปลอร์	8
2-7 ออปโตคัปเปลอร์ชนิดมีขาเบส	9
2-8 โครงสร้างของหม้อแปลงไฟฟ้า	9
2-9 การส่งถ่ายพลังงานไฟฟ้าของหม้อแปลงจากวงจรหนึ่งไปยังอีกวงจรหนึ่ง	10
2-10 (ก) โครงสร้าง	11
(ข) สัญลักษณ์	11
(ค) วงจรสมมูลของไตรแอก	11
2-11 กราฟลักษณะสมบัติของไตรแอก	12
2-12 รูปคลื่นแรงดันตกคร่อมโหลด	13
2-13 จุดผ่านศูนย์	14
2-14 วงจรตรวจจับจุดผ่านศูนย์ด้วยอุปกรณ์เชื่อมต่อทางแสง	14
2-15 รูปสัญญาณของวงจรตรวจจับจุดผ่านศูนย์	15
3-1 บล็อกไดอะแกรมรวมเครื่องกำเนิดกระแสเกินชั่วคราว	16
3-2 วงจรตรวจจับจุดผ่านศูนย์	17
3-3 ผลการจำลองการทำงานของวงจรตรวจจับจุดผ่านศูนย์	17
3-4 บล็อกไดอะแกรมของโปรแกรมในไมโครคอนโทรลเลอร์ 1	19
3-5 วงจรไมโครคอนโทรลเลอร์ 1	20
3-6 ผลการจำลองการทำงานของไมโครคอนโทรลเลอร์ 1	21
3-7 วงจรขับเคลื่อน	21
3-8 ผลการจำลองการทำงานของวงจรขับเคลื่อน	22

## สารบัญภาพ (ต่อ)

ภาพที่	หน้า
3-9 วงจรเซนเซอร์กระแส	23
3-10 ผลการจำลองการทำงานของเซนเซอร์กระแส	23
3-11 บล็อกไดอะแกรมโปรแกรมไมโครคอนโทรลเลอร์2	24
3-12 วงจรไมโครคอนโทรลเลอร์2	25
3-13 ผลการจำลองการทำงานของไมโครคอนโทรลเลอร์2	26
3-14 วงจรทั้งหมดของเครื่องกำเนิดกระแสเกินชั่วคราวที่จะนำไปสร้างจริง	27
4-1 วงจรจริงของเครื่องกำเนิดกระแสเกินชั่วคราว	29
4-2 ผลการทดสอบการทำงานของวงจรตรวจจับจุดผ่านศูนย์ ที่แรงดันอินพุต 10 V <sub>RMS</sub>	30
4-3 ผลการทดสอบการทำงานของวงจรตรวจจับจุดผ่านศูนย์ ที่แรงดันอินพุต 100 V <sub>RMS</sub>	30
4-4 ผลการทดสอบการทำงานของไมโครคอนโทรลเลอร์1 ของกระแสปกติ	31
4-5 ผลการทดสอบการทำงานของไมโครคอนโทรลเลอร์1 ของกระแสเกิน	31
4-6 ผลการทดสอบการทำงานของวงจรขั้วเกท	32
4-7 ผลการทดสอบการทำงานของเซนเซอร์กระแส	32
4-8 ผลการทดสอบการทำงานของไมโครคอนโทรลเลอร์2	33
4-9 การต่อวงจรทดสอบการทำงานของเครื่องกำเนิดกระแสเกินชั่วคราว	33
4-10 การทดสอบครั้งที่ 1 เลือกกำหนดจำนวนลูกคลื่น กระแสปกติ 3 ลูกคลื่น และกระแสเกิน 2 ลูกคลื่น	34
4-11 การทดสอบครั้งที่ 2 เลือกกำหนดจำนวนลูกคลื่น กระแสปกติ 5 ลูกคลื่น และกระแสเกิน 10 ลูกคลื่น	35
4-12 การทดสอบครั้งที่ 3 เลือกกำหนดระยะเวลา กระแสปกติ 1 วินาที และกระแสเกิน 1 วินาที	35
4-13 การทดสอบครั้งที่ 4 เลือกกำหนดระยะเวลา กระแสปกติ 2 วินาที และกระแสเกิน 3 วินาที	36
4-14 การทดสอบครั้งที่ 5 เลือกกำหนดจำนวนลูกคลื่น กระแสปกติ 3 ลูกคลื่น และกระแสเกิน 2 ลูกคลื่น	37

## สารบัญภาพ (ต่อ)

ภาพที่	หน้า
4-15 การทดสอบครั้งที่ 6 เลือกกำหนดจำนวนลูกคลื่น กระแสปกติ 5 ลูกคลื่น และกระแสเกิน 10 ลูกคลื่น	37
4-16 การทดสอบครั้งที่ 7 เลือกกำหนดระยะเวลา กระแสปกติ 1 วินาที และกระแสเกิน 1 วินาที	38
4-17 การทดสอบครั้งที่ 8 เลือกกำหนดระยะเวลา กระแสปกติ 2 วินาที และกระแสเกิน 3 วินาที	38
4-18 การทดสอบครั้งที่ 9 เลือกกำหนดจำนวนลูกคลื่น กระแสปกติ 3 ลูกคลื่น และกระแสเกิน 2 ลูกคลื่น	39
4-19 การทดสอบครั้งที่ 10 เลือกกำหนดจำนวนลูกคลื่น กระแสปกติ 5 ลูกคลื่น และกระแสเกิน 10 ลูกคลื่น	40
4-20 การทดสอบครั้งที่ 11 เลือกกำหนดระยะเวลา กระแสปกติ 1 วินาที และกระแสเกิน 1 วินาที	40
4-21 การทดสอบครั้งที่ 12 เลือกกำหนดระยะเวลา กระแสปกติ 2 วินาที และกระแสเกิน 3 วินาที	41
4-22 เอาท์พุทของเครื่องกำเนิดกระแสเกินชั่วคราว เมื่อเลือกกำหนดจำนวนลูกคลื่น กระแสปกติ 3 ลูกคลื่น และกระแสเกิน 3 ลูกคลื่น	42
4-23 ผลการนับของ Surge Counter	42
ก-1 ไดอะแกรมรวมของเครื่องกำเนิดกระแสเกินชั่วคราว	47
ก-2 ตำแหน่งอุปกรณ์บนลายทองแดงของเครื่องกำเนิดกระแสเกินชั่วคราว ในโปรแกรมออกแบบลายวงจร	48
ก-3 ลายทองแดงด้านบนของเครื่องกำเนิดกระแสเกินชั่วคราว ในโปรแกรมออกแบบลายวงจร	48
ก-4 ลายทองแดงด้านล่างของเครื่องกำเนิดกระแสเกินชั่วคราว ในโปรแกรมออกแบบลายวงจร	49
ก-5 ลายทองแดงด้านบนของเครื่องกำเนิดกระแสเกินชั่วคราว	49

## สารบัญภาพ (ต่อ)

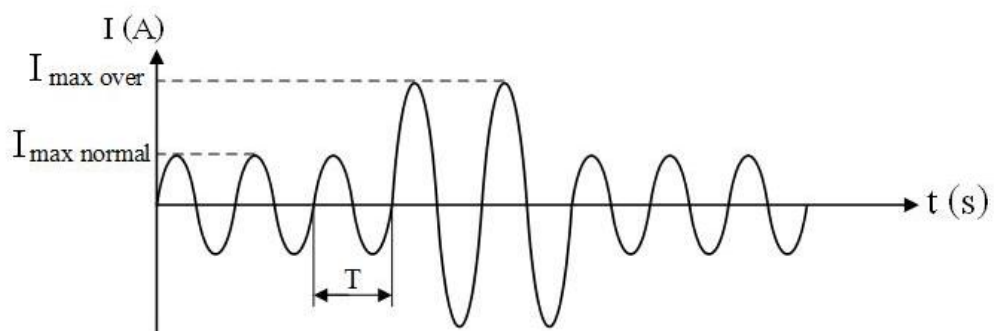
ภาพที่	หน้า
ก-6 ลายทองแดงด้านล่างของเครื่องกำเนิดกระแสเกินชั่วคราว	50
ก-7 แผ่นวงจรด้านบนของเครื่องกำเนิดกระแสเกินชั่วคราวหลังบัดกรี	50
ก-8 แผ่นวงจรด้านล่างของเครื่องกำเนิดกระแสเกินชั่วคราวหลังบัดกรี	51
ก-9 เครื่องกำเนิดกระแสเกินชั่วคราวหลังประกอบอุปกรณ์ลงกล่อง	52

# บทที่ 1

## บทนำ

### 1.1 ความจำเป็นและความสำคัญ

เนื่องจากในปัจจุบันไฟฟ้านับเป็นพลังงานสำคัญอันดับต้น ๆ ทั้งในอุตสาหกรรม และในชีวิตประจำวัน ในแต่ละสถานที่จะมีการใช้พลังงานไฟฟ้าในปริมาณมาก จึงอาจเกิดกรณีที่กระแสไฟฟ้าเกิน ซึ่งอาจทำให้เกิดความเสียหายกับอุปกรณ์เครื่องใช้ไฟฟ้าและสายไฟ หรืออาจทำให้เกิดอัคคีภัยได้ โดยการเกิดกระแสเกิน สามารถเกิดได้จากหลายสาเหตุ เช่น อาจเกิดจากการที่ค่าความต้านทานของโหลดลดลง เมื่อมีการต่อโหลดหลาย ๆ ตัว (ขนาน) เข้าด้วยกัน ทำให้เกิดการใช้พลังงานมากขึ้น หรืออาจเกิดจากการที่ค่าความต้านทานของโหลดลดลงกะทันหัน เช่น เกิดการลัดวงจรของอุปกรณ์ หรืออาจมาจากการดึงกระแสของมอเตอร์ไฟฟ้าขณะสตาร์ท โดยกระแสเกินจะมีลักษณะคล้ายกับรูปกระแสปกติที่จะเป็นลูกคลื่นไซน์ที่มีความถี่คงที่ แต่จะมีความสูงเพิ่มขึ้นตามขนาดของกระแสที่เพิ่มขึ้น ซึ่งสามารถแบ่งออกได้เป็น 2 แบบ ตามระยะเวลาที่เกิดกระแสเกินได้ คือ แบบชั่วคราวกับแบบถาวร ซึ่งกระแสเกินแบบถาวร สามารถแก้ปัญหาได้ด้วยการใช้อุปกรณ์ประเภทตัดวงจรทั่วไป ส่วนแบบชั่วคราวนั้นจะสามารถเห็นด้วยการวัดกระแสด้วยออสซิลโลสโคป แต่ในการทำงานจริงไม่สามารถใช้ออสซิลโลสโคปเพื่อจับกระแสเกินตลอดเวลาได้ จึงมีเครื่องวัดกระแสเกินชั่วคราวเพื่อทดแทนการทำงานในส่วนนี้ ซึ่งโดยทั่วไปจะเป็นเครื่องนับจำนวนครั้งที่เกิดลูกคลื่นกระแสเกิน คือเมื่อมีลูกคลื่นกระแสที่มีขนาดเกินกว่าค่าสูงสุดที่กำหนดไว้ เครื่องวัดจะทำการนับจำนวนลูกคลื่นที่เกิดกระแสเกิน และเพื่อที่จะทำการทดสอบการทำงานของเครื่องวัดกระแสเกินชั่วคราว จึงได้คิดที่จะสร้างเครื่องกำเนิดกระแสไฟฟ้าเกินชั่วคราวขึ้น โดยให้ได้รูปสัญญาณเป็นลูกคลื่นไซน์ที่มีกระแสเกินเกิดขึ้นบนกระแสปกติ แสดงดังภาพที่ 1-1



ภาพที่ 1-1 ลูกคลื่นไซน์ที่มีกระแสเกินเกิดขึ้นบนกระแสปกติ

เนื่องจากที่ห้องปฏิบัติการไฟฟ้าแรงสูงของภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ มีเครื่องกำเนิดกระแสเกินชั่วคราวอยู่ก่อนแล้ว แต่เนื่องด้วยมีการใช้งานมาเป็นเวลานาน และเกิดการชำรุดเสียหาย จนทำให้ไม่สามารถใช้งานได้ปกติและมีขนาดและน้ำหนักมากเกินไป โครงการนี้จึงเกิดขึ้นมาเพื่อสร้างเครื่องกำเนิดกระแสเกินชั่วคราวขึ้นมาใหม่

ดังนั้นในโครงการนี้มีเป้าหมายเพื่อออกแบบและสร้างเครื่องกำเนิดกระแสเกินชั่วคราวให้ขนาดเล็กระทัดรัด และใช้งานได้สะดวกขึ้นกว่าเดิม

## 1.2 วัตถุประสงค์ของการทำโครงการ

- 1.2.1 เพื่อศึกษาและทำความเข้าใจการทำงานของเครื่องกำเนิดกระแสเกินชั่วคราว
- 1.2.2 เพื่อออกแบบ และสร้างเครื่องกำเนิดกระแสเกินชั่วคราว ที่สามารถกำหนดจำนวนลูกคลื่นและระยะเวลาการเกิดได้ พร้อมชุดวัดกระแส

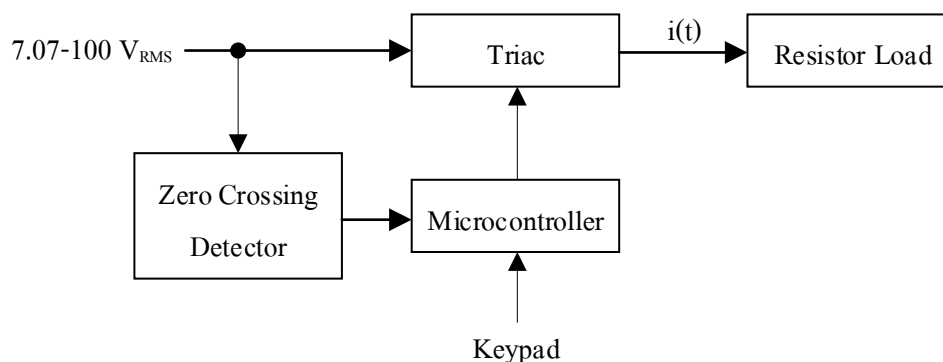
## 1.3 ขอบเขตของการทำโครงการ

- 1.3.1 กระแสปกติ และกระแสเกินชั่วคราวเป็นกระแสสลับรูปไซน์ ความถี่ 50 Hz
- 1.3.2 ลูกคลื่นที่สร้างจะเริ่มต้นที่ค่ากระแสเป็นศูนย์
- 1.3.3 พิกัดกระแสเกินชั่วคราว  $20 A_{RMS}$  สามารถปรับค่ากระแสเป็น  $1 - 20 A_{RMS}$  ได้
- 1.3.4 สามารถกำหนดค่า จำนวนลูกคลื่น หรือระยะเวลาของกระแสปกติได้
- 1.3.5 สามารถกำหนดจำนวนลูกคลื่น หรือตำแหน่งการเกิดกระแสเกินบนกระแสปกติ
- 1.3.6 มีชุดวัดกระแส และชุดแสดงผลพร้อมสามารถนำไปแสดงผลด้วยออสซิลโลสโคป (Oscilloscope) ได้

## 1.4 แนวเหตุผล ทฤษฎีสำคัญหรือสมมติฐาน

กระแสปกติและกระแสเกินชั่วคราวที่สร้างขึ้น จะเป็นกระแสสลับรูปไซน์ที่ผ่านตัวต้านทานขณะจ่ายแรงดันไฟสลับรูปไซน์ความถี่ 50 Hz การปรับค่ากระแสทำได้โดยการปรับค่าแรงดันแหล่งจ่าย หรือค่าความต้านทาน การควบคุมจำนวนลูกคลื่นกระแสเกินทำผ่านอุปกรณ์ไทรสเตอร์ (Thyristor) หรืออุปกรณ์สวิตช์กำลัง (Power Switch) ส่วนการวัดกระแสจะใช้ โมดูลเซนเซอร์หม้อแปลงกระแส (Current Transformer Sensor Module) และมีการขยายเป็นสัญญาณแรงดันเพื่อสามารถนำไปแสดงผลได้ด้วยออสซิลโลสโคป





ภาพที่ 1-2 บล็อกไดอะแกรมของเครื่องกำเนิดกระแสเกินชั่วคราว

จากภาพที่ 1-2 จะเห็นได้ว่าส่วนไมโครคอนโทรลเลอร์จะทำการรับคำสั่งจำนวนลูกคลื่นหรือระยะเวลาที่เกิดกระแสปกติ กระแสเกิน และสัญญาณเริ่มการทำงานจากผู้ใช้งานทางคีย์แพด (Keypad) และรับสัญญาณจากวงจรจับจุดตัดศูนย์ (Zero Crossing Detector) เพื่อทำการสร้างขบวนพัลส์ (Pulse Train) ที่ใช้ในการขับขาเกตของไทรแอก (Triac) ให้ทำการเปิดวงจรจนเกิดกระแสปกติและกระแสเกินชั่วคราวไหลผ่านตัวต้านทานตามที่ต้องการ

## 1.5 วัตถุประสงค์ของการทำโครงการ

- 1.5.1 ศึกษาแล้วทำความเข้าใจโปรแกรม ภาษาและการทำงานของไมโครคอนโทรลเลอร์ (Microcontroller)
- 1.5.2 ศึกษาแล้วทำความเข้าใจวงจรและอุปกรณ์ที่ใช้ในการสร้างเครื่องกำเนิดกระแสเกินชั่วคราว
- 1.5.3 ใช้เป็นแหล่งกำเนิดกระแสเกินชั่วคราว สำหรับงานทดสอบที่เกี่ยวข้อง

## บทที่ 2

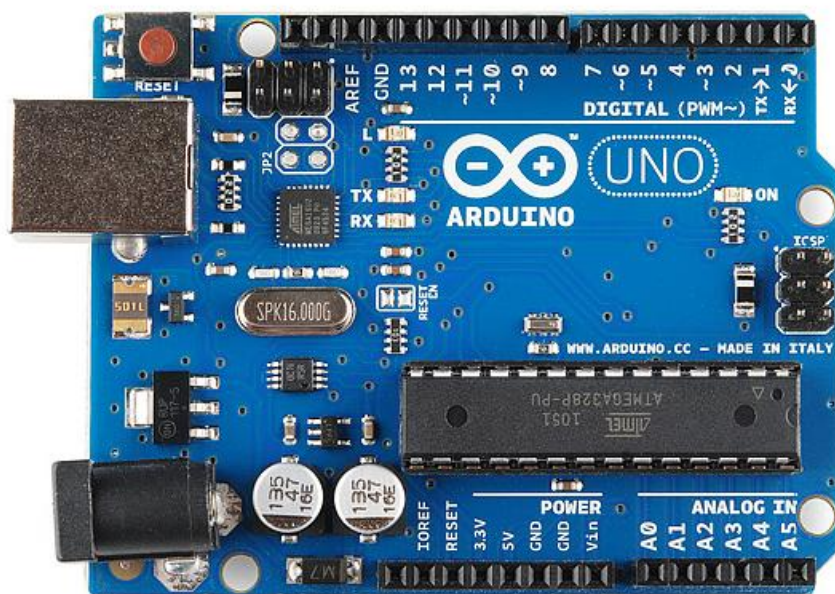
### ทฤษฎีพื้นฐานที่เกี่ยวข้อง

#### 2.1 บทนำ

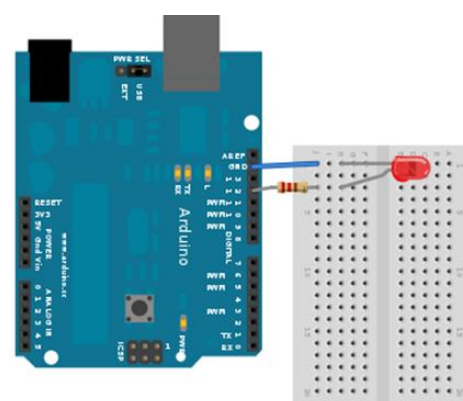
ในการพัฒนาสร้างเครื่องกำเนิดกระแสเกินชั่วคราว มีการใช้ไมโครคอนโทรลเลอร์และวงจรอิเล็กทรอนิกส์มาประกอบเป็นเครื่องกำเนิดกระแสเกินชั่วคราว ซึ่งจะประกอบไปด้วย ส่วนของไมโครคอนโทรลเลอร์ ที่จะเป็นศูนย์กลางควบคุมใหญ่ เช่น การหาจุดตัดศูนย์ (Zero-crossing) และควบคุมส่วนของคีย์แพดควบคุมกับจอแอลซีดี (LCD) ที่ใช้ในการกำหนดจำนวนลูกคลื่นหรือจำนวนเวลาที่เกิดกระแสเกินและกระแสปกติได้ ส่วนของหม้อแปลงสัญญาณพัลส์ (Pulse transformer) วงจรขับเคลื่อน (Gate Drive) รวมไปถึงไดรเวอร์ที่ทำหน้าที่เป็นสวิตช์กำลัง และเป็นอุปกรณ์ที่สำคัญที่สุด จึงจำเป็นที่จะต้องศึกษาเกี่ยวกับหลักการทำงาน และวิธีการใช้งานอุปกรณ์ที่นำมาสร้างเป็นวงจรต่างๆ และหลักการทำงานของแต่ละส่วน

#### 2.2 อาดูยโน (Arduino)

Arduino อ่านว่า “อา-ดู-คิ-โน้ หรือ อาดูยโน”<sup>[1]</sup> เป็น Open-Source แพลตฟอร์ม คือมีการเปิดเผยข้อมูลทั้งด้าน ฮาร์ดแวร์ (Hardware) และซอฟต์แวร์ (Software) สำหรับใช้สร้างโครงงานทางด้านอิเล็กทรอนิกส์ อาดูยโนประกอบไปด้วย 2 ส่วนคือ บอร์ดวงจร (บอร์ดไมโครคอนโทรลเลอร์) และส่วนของซอฟต์แวร์ หรือที่เรียกว่า IDE (Integrated Development Environment) เป็นโปรแกรมที่จะทำงานบนคอมพิวเตอร์ (Computer) ใช้สำหรับเขียน หรืออัปโหลดโปรแกรมลงไปในบอร์ด อาดูยโนนั้นเป็นบอร์ดไมโครคอนโทรลเลอร์ตระกูลเอวีอาร์ (AVR) ถูกออกแบบมาเพื่อให้ใช้งานได้ง่าย สามารถเชื่อมต่อกับคอมพิวเตอร์ ด้วยสาย USB และโปรแกรม IDE ก็ใช้ภาษา C++ ในการเขียนโปรแกรม ทำให้ง่ายต่อการเขียนโปรแกรม ผู้ใช้งานสามารถต่อวงจรอิเล็กทรอนิกส์จากภายนอกแล้วเชื่อมต่อเข้ามาที่ขาอินพุต/เอาต์พุตของบอร์ดอาดูยโน และอาดูยโนสามารถต่อใช้กับอุปกรณ์อิเล็กทรอนิกส์ต่าง ๆ เช่น ตัววัดเซนเซอร์ ตัววัดอุณหภูมิ มอเตอร์เซอร์โว รีเลย์ไดโอดเปล่งแสง (ดูตัวอย่างผังภาพที่ 2-2) สัญญาณแอนะล็อกเป็นสัญญาณดิจิทัล (A/D) การปรับความกว้างของพัลส์โดยการนำเอาสองสัญญาณมาเปรียบเทียบกับกัน (PWM : Pulse Width Modulation) เป็นต้น



ภาพที่ 2-1 บอร์ดอาดูอิน<sup>[2]</sup>



ภาพที่ 2-2 บอร์ดอาดูอินต่อกับอุปกรณ์เสริมต่าง ๆ (กับไดโอดเปล่งแสง)<sup>[1]</sup>

## 2.3

### แอลซีดีดีสเพลย์

### (LCD

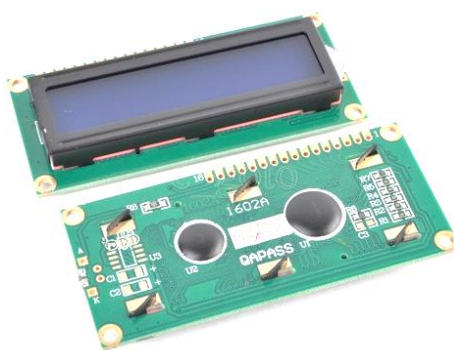
### Display)

จอแอลซีดี (Liquid Crystal Display)<sup>[3]</sup> เป็นจอแสดงผลชนิดหนึ่ง ที่นิยมนำมาใช้งานกับระบบสมองกลฝังตัวอย่างแพร่หลาย จอแอลซีดีมีทั้งแบบแสดงผลเป็นตัวอักษรเรียกว่า คาแรคเตอร์แอลซีดี (Character LCD) ซึ่งมีการกำหนดตัวอักษรหรืออักขระที่สามารถแสดงผลไว้ได้อยู่แล้ว และแบบที่สามารถแสดงผลออกมาเป็นรูปภาพหรือสัญลักษณ์ได้ตามความต้องการของผู้ใช้งาน เรียกว่า กราฟฟิกแอลซีดี (Graphic LCD) นอกจากนี้บางชนิดเป็นจอที่มีการผลิตขึ้นมาใช้เฉพาะงาน ทำให้มีรูปแบบ

และรูปร่างเฉพาะเจาะจงในการแสดงผล เช่น นาฬิกาดิจิทัล เครื่องคิดเลข หรือ หน้าปัดวิทยุ เป็นต้น

โครงสร้างของจอแอลซีดีทั่วไปจะประกอบขึ้นด้วยแผ่นแก้ว 2 แผ่นประกบกันอยู่โดยเว้นช่องว่างตรงกลางไว้ 6-10 ไมโครเมตร ผิวด้านในของแผ่นแก้วจะเคลือบด้วยตัวนำไฟฟ้าแบบใสเพื่อใช้แสดงตัวอักษร ตรงกลางระหว่างตัวนำไฟฟ้าแบบใสกับผลึกเหลวจะมีชั้นของสารที่ทำให้โมเลกุลของผลึกรวมตัวกันในทิศทางที่แสงส่องมากระทบเรียกว่า Alignment Layer และผลึกเหลวที่ใช้โดยทั่วไปจะเป็นแบบแมกเนติก (Magnetic)

จอแอลซีดีที่แสดงผลเป็นอักขระหรือตัวอักษร โดยทั่วไปจะมีหลายแบบด้วยกัน มีทั้ง 16 ตัวอักษร 20 ตัวอักษรหรือมากกว่า และจำนวนบรรทัดจะมีตั้งแต่ 1 บรรทัด 2 บรรทัด 4 บรรทัดหรือมากกว่าตามแต่ความต้องการและลักษณะของงานที่ใช้ หรืออาจจะมีแบบสั่งทำเฉพาะงานก็ได้ ในที่นี้เราจะยกตัวอย่างจอแอลซีดีขนาด 16x2 คาแรคเตอร์ (16 ตัวอักษร 2 บรรทัด) (ภาพที่ 2-3) สามารถหาซื้อได้ง่ายและมีราคาไม่สูง เหมาะสมกับการใช้งานแสดงผลไม่มากในหน้าจอเดียว



ภาพที่ 2-3 จอแอลซีดี 16 ตัวอักษร 2 บรรทัด<sup>[3]</sup>

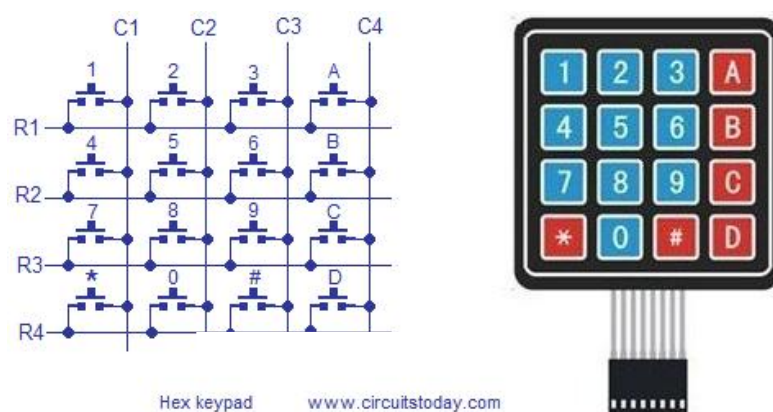


ภาพที่ 2-4 ด้านหน้าของจอแอลซีดี 16 ตัวอักษร 2 บรรทัด<sup>[3]</sup>

## 2.4

## คีย์แพด

แป้นปุ่มกดหรือคีย์แพดเป็นอุปกรณ์สำหรับรับอินพุตจากผู้ใช้งาน<sup>[4]</sup> มีลักษณะเป็นปุ่มกดหลายปุ่ม ซึ่งเป็นอุปกรณ์อ่านค่าอินพุตที่ต่อเข้ากับปุ่มกด จำนวนของคีย์แพดจะเท่ากับขาพอร์ตของที่ใช้งาน ซึ่งเป็นการสิ้นเปลืองขาพอร์ต จึงมีการต่อใช้สวิทช์จำนวนมากนี้ในรูปแบบของ “เมตริกสวิทช์” (Metrix Switch) จึงทำให้ลดการใช้พอร์ตได้อย่างดี การต่อใช้งานคีย์แพดเมตริกสวิทช์จะต่อใช้งานแบบพูลอัพ (Pull-up) ดังนั้นสถานะของสวิทช์จะมีสถานะเป็น “1” ทั้งหมด เมื่อต้องการอ่านค่าของคีย์แพด เราจะต้องกำหนดค่าของหลัก (Column) ให้สถานะเป็น “0” เพราะ หลัก นั้นจะเป็นการควบคุมสัญญาณ หากสวิทช์แถวไหน มีการเปลี่ยนแปลงแสดงว่าแถวนั้นมีการกดสวิทช์ วิธีการนี้เรียกว่า การสแกนปุ่มกด (Key scan)<sup>[5]</sup>

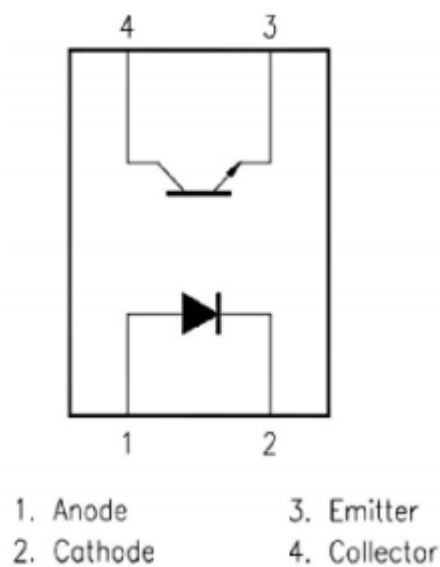


ภาพที่ 2-5 4x4 Matrix Keypad<sup>[6]</sup>

## 2.5 อุปกรณ์เชื่อมต่อทางแสง (Opto-Coupler)

อุปกรณ์เชื่อมต่อทางแสง หรือที่เรียกว่า “ออปโตคัปเปิลอร์” หรืออาจเรียกว่า อุปกรณ์แยกสัญญาณทางแสง (Opto-Isolator) เป็นอุปกรณ์อิเล็กทรอนิกส์สำหรับการเชื่อมต่อทางแสง โดยการเปลี่ยนสัญญาณไฟฟ้าให้เป็นแสงแล้วเปลี่ยนกลับเป็นสัญญาณไฟฟ้าตามเดิม นิยมใช้สำหรับการเชื่อมต่อสัญญาณระหว่างสองวงจรและต้องการแยกกันทางไฟฟ้าโดยเด็ดขาด เพื่อป้องกันการรบกวนกันทางไฟฟ้าระหว่างสองวงจร ภายในของอุปกรณ์ประเภทนี้ประกอบด้วยไดโอดเปล่งแสง ซึ่งทำหน้าที่เป็นตัวส่งแสง (Optical Transmitter) เช่น แสงอินฟราเรด (Infrared) และสำหรับตัวรับแสง (Optical Receiver) ซึ่งมักนิยมใช้โฟโตทรานซิสเตอร์ (Phototransistor) เป็นตัวรับโดยจะถูกผลิตรวมอยู่ในตัวเดียวกัน

โฟโตทรานซิสเตอร์ทำงานได้ในลักษณะเดียวกับทรานซิสเตอร์รอยต่อคู่แบบ NPN แต่ไม่มีขาเบส (B) และถูกแทนที่ด้วยส่วนรับแสง เมื่อได้รับแสงหรืออนุภาคของแสง หรือที่เรียกว่า โฟตอน (Photons) ในปริมาณมากพอ จะทำให้เกิดอนุภาคอิสระที่มีประจุในบริเวณรอยต่อระหว่างเบสและคอลเลกเตอร์ (Base-Collector Region) และให้ผลเหมือนมีกระแสไหลเข้าที่ขาเบส ออปโตคัปเปิลอร์แบบ 4 ขา (เบอร์ PC817)<sup>[7]</sup> แสดงดังภาพที่ 2-6

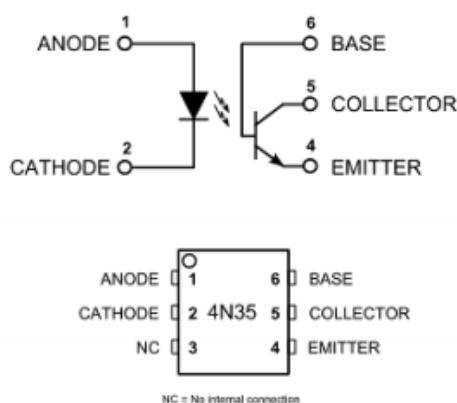


ภาพที่ 2-6 บล็อกไดอะแกรมของออปโตคัปเปิลอร์<sup>[8]</sup>

ออปโตคัปเปิลอร์สามารถรับสัญญาณอินพุตเปิด-ปิดไดโอดเปล่งแสงที่อยู่ภายในตัวอุปกรณ์ เพื่อควบคุมการทำงานของโฟโตทรานซิสเตอร์ด้วยแสง ดังนั้นจึงนำไปใช้ในลักษณะเป็นอุปกรณ์สวิตช์เปิด-ปิด หรือนำไปต่อกับวงจรทรานซิสเตอร์ภายนอกเพื่อให้สามารถขับกระแสได้ในปริมาณที่สูงขึ้น

เมื่อแรงดันอินพุตอยู่ในระดับที่สูงกว่าแรงดันไบอัสตรงของไดโอดเปล่งแสง ( $V_F$ ) จะทำให้เกิดกระแสไหล หรือที่เรียกว่า กระแสอินพุต หรือ กระแสไบอัสตรง ( $I_F$ ) ทำให้ไดโอดเปล่งแสงตามปริมาณกระแสไฟฟ้าที่ไหลในการต่อวงจรจะต้องมีตัวต้านทานต่ออนุกรมอยู่ด้วย เพื่อจำกัดปริมาณของกระแสที่ไหลไม่ให้สูงเกิน ซึ่งขึ้นอยู่กับอุปกรณ์แต่ละตัวที่ใช้ แต่โดยทั่วไปแล้วควรจะให้อยู่ในช่วง 5-50 มิลลิแอมป์ (mA) เมื่อโฟโตทรานซิสเตอร์ได้รับแสงจะทำให้สามารถนำไฟฟ้าระหว่างคอลเลกเตอร์ (Collector) และอิมิตเตอร์ (Emitter) ได้ ซึ่งให้ผลเหมือนในกรณีที่จ่ายกระแสเข้าที่ขาเบส (Base) ของทรานซิสเตอร์รอยต่อคู่แบบ NPN และถ้ามีแรงดันตกคร่อมที่ขาคอลเลกเตอร์และขาอิมิตเตอร์หรือ  $V_{CE}$  มีค่ามากกว่าศูนย์ก็จะทำให้มีกระแสเอาต์พุตไหล

ตัวถังของออปโตคัปเปลอร์ที่พบเห็นได้บ่อย คือ ตัวถังแบบ 4 ขาและแบบ 6 ขา บางตัวที่มี ไดโอดเปล่งแสงและโฟโตทรานซิสเตอร์เพียงหนึ่งคู่ แต่ออปโตคัปเปลอร์ตัวถังแบบ 6 ขา จะมีขาเบส (Base Connection Pin) ที่เชื่อมต่อมาจากบริเวณเบสของโฟโตทรานซิสเตอร์ที่อยู่ภายใน และใช้ในการ ปรับความไวในการตอบสนองเชิงเวลา โดยการนำขาเบสไปต่อกับตัวต้านทานที่มีค่า อยู่ในช่วง 200 k ถึง 1 M ไปยังกราวด์ของวงจรเอาต์พุต แต่ถ้าไม่สนใจเรื่องความไวในการตอบสนองก็ไม่จำเป็นต้องต่อ ขาเบส ออปโตคัปเปลอร์ชนิดมีขาเบสแสดงดังภาพที่ 2-7



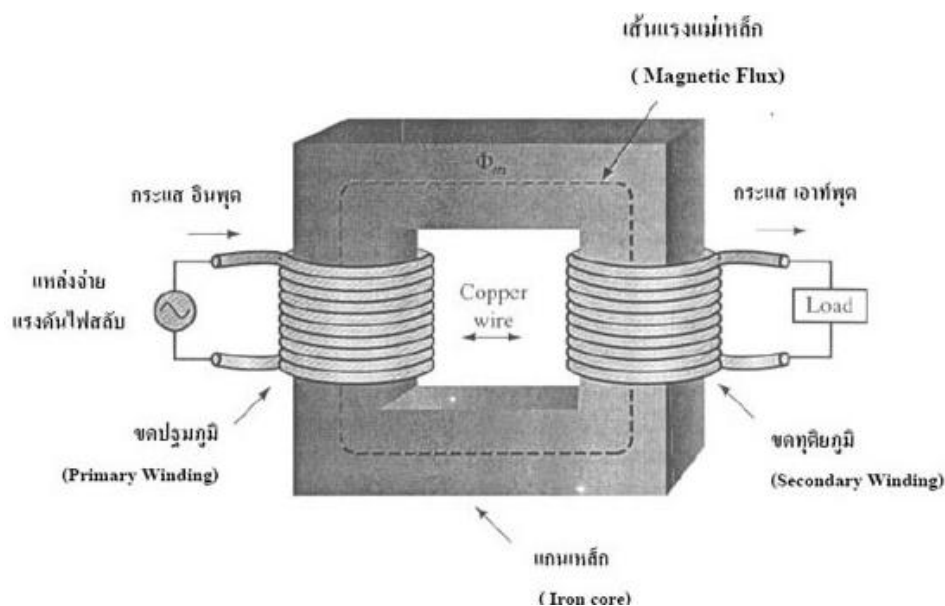
ภาพที่ 2-7 ออปโตคัปเปลอร์ชนิดมีขาเบส<sup>[7]</sup>

## 2.6 หม้อแปลงไฟฟ้า (Transformer)

หม้อแปลงไฟฟ้า คือ เครื่องกลไฟฟ้าชนิดหนึ่งที่ใช้เปลี่ยนพลังงานไฟฟ้าเป็นพลังงานไฟฟ้าโดย สามารถเปลี่ยนแรงดันไฟฟ้าให้เพิ่มขึ้นเรียกว่า “Step up Transformer” และให้ลดลงเรียกว่า “Step down Transformer” แต่ไม่เปลี่ยนกำลังไฟฟ้า (Power/Watt) และความถี่ (Frequency/Hz)<sup>[9]</sup>

### 2.6.1 โครงสร้างของหม้อแปลงไฟฟ้า

โครงสร้างของหม้อแปลงไฟฟ้าแสดงดังภาพที่ 2-8



ภาพที่ 2-8 โครงสร้างของหม้อแปลงไฟฟ้า<sup>[10]</sup>

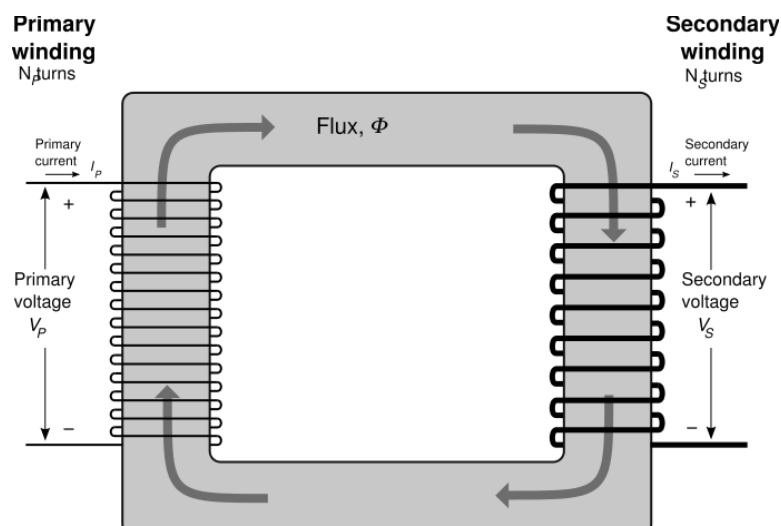
จากภาพที่ 2-8 จะเห็นว่าหม้อแปลงไฟฟ้ามีส่วนประกอบที่สำคัญ 3 ส่วน คือ

- แกนเหล็ก แกนเหล็กของหม้อแปลงจะมีลักษณะเป็นแผ่นบาง ๆ เคลือบด้วยฉนวน เรียกว่า แผ่นลามิเนต
- ขดลวดตัวนำ ขดลวดตัวนำของหม้อแปลงจะมีลักษณะเป็นขดลวดทองแดงหรือ อลูมิเนียมหุ้มด้วยฉนวน โดยทั่วไปหม้อแปลงจะมีขดลวด 2 ชุด คือ ขดลวดปฐมภูมิ (Primary Winding) และขดลวดทุติยภูมิ (Secondary Winding)
- ฉนวน ฉนวนของหม้อแปลงมีไว้เพื่อป้องกันไม่ให้ขดลวดสัมผัสกับส่วนที่เป็นแกนเหล็ก และป้องกันไม่ให้ขดลวดแต่ละชั้นสัมผัสกัน

#### 2.6.2 หลักการทำงานของหม้อแปลงไฟฟ้า

การทำงานของหม้อแปลง ใช้การส่งถ่ายพลังงานไฟฟ้าจากวงจรหนึ่งไปยังอีกวงจรหนึ่ง แสดงดังภาพที่ 2-9





ภาพที่ 2-9 การส่งถ่ายพลังงานไฟฟ้าของหม้อแปลงจากวงจรหนึ่งไปยังอีกวงจรหนึ่ง<sup>[11]</sup>

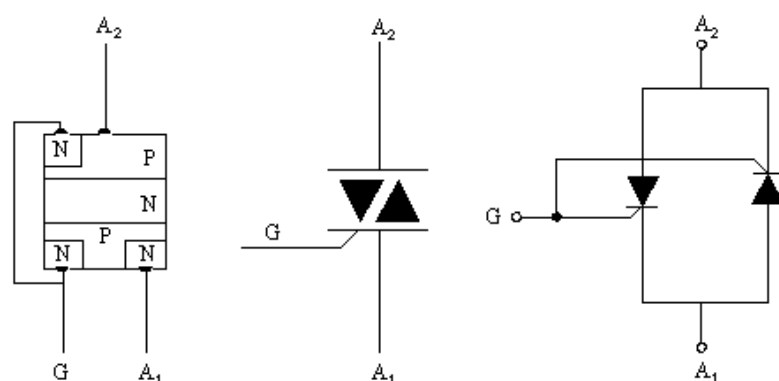
จากภาพที่ 2-9 จะเห็นว่า การส่งถ่ายพลังงานจากขดลวดปฐมภูมิ ซึ่งกระแสไฟฟ้าที่ป้อนเข้า มาจะสร้างเส้นแรงแม่เหล็ก (Flux) และแรงแม่เหล็ก (Magnetomotive Force) ไปยังขดลวดทุติยภูมิ เพื่อ สร้างเป็นแรงดันไฟฟ้า และกระแสไฟฟ้าออกมา โดยมีพลังงานไฟฟ้าคงที่และความถี่ไฟฟ้าเท่ากับ ความถี่ไฟฟ้าที่ป้อนเข้ามาทางขดปฐมภูมิ

การทำงานของหม้อแปลงไฟฟ้าจะไม่มีส่วนใดเคลื่อนที่เหมือนมอเตอร์จึงมีการสูญเสีย กำลังงานในขณะที่ทำงานน้อยกว่ามอเตอร์

## 2.7 ไตรแอก

### 2.7.1 โครงสร้างของไตรแอก

ไตรแอก เป็นอุปกรณ์จำพวกสารกึ่งตัวนำในกลุ่มของทรานซิสเตอร์มีลักษณะโครงสร้าง ภายในคล้ายกับไดโอด แต่มีขาเกตเพิ่มขึ้นมาอีก 1 ขา ไตรแอกถูกสร้างขึ้นเพื่อแก้ไขข้อบกพร่องของ เอสซีอาร์ (SCR : Silicon Control Rectifier) ซึ่งไม่สามารถนำกระแสในซีกลบของไฟฟ้าสลับได้ การ นำไตรแอกไปใช้งานส่วนใหญ่จะใช้ทำเป็นวงจรควบคุมการทำงานเป็นสวิตช์ต่อแรงดันไฟสลับ ไตร แอกถูกสร้างขึ้นมาให้ใช้งานกระแสสูง ๆ ดังนั้นต้องระวังเรื่องของการระบายความร้อน โดย โครงสร้างสัญลักษณ์และวงจรสมมูลของไตรแอกแสดงดังภาพที่ 2-10

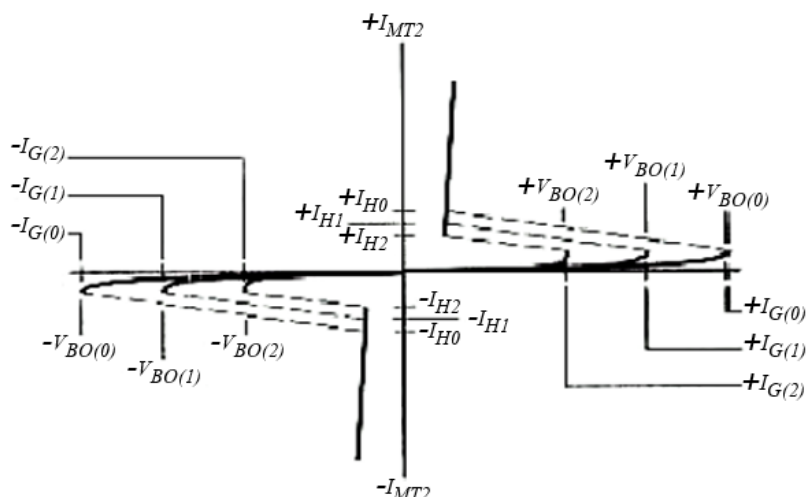


ภาพที่ 2-10 (ก) โครงสร้าง (ข) สัญลักษณ์ (ค) วงจรสมมูลของไทรแอก<sup>[12]</sup>

ภาพที่ 2-10 (ก) โครงสร้างของไทรแอก จะประกอบด้วยสารกึ่งตัวนำตอนใหญ่ 3 ตอน คือ PNP และในสารกึ่งตัวนำตอนใหญ่จะมีสารกึ่งตัวนำตอนย่อยชนิด N อีก 3 ตอนต่อรวมในสารกึ่งตัวนำ P ทั้ง 2 ตอนมีขาต่อออกมาใช้งาน 3 ขา เหมือน SCR คือขาแอโนด 1 (A1) เรียกว่า ขาเทอร์มินอล 1 (Main terminal 1, MT1) ขาแอโนด 2 (A2) เรียกว่า ขาเทอร์มินอล 2 (Main terminal 2, MT2) และขาเกต (Gate, G)

### 2.7.2 กราฟคุณสมบัติของไทรแอก

ลักษณะสมบัติของไทรแอกแรงดันไบอัสตรงกับไทรแอก ( $V_{BO(0)}$ ) และจ่ายกระแสเกตบวก ( $I_G$ ) ให้กับขาเกตของไทรแอก จะทำให้ไทรแอกนำกระแสได้โดยกระแสจะไหลจากขั้ว A1 ไปยังขั้ว A2 และเมื่อป้อนแรงดันไบอัสลบให้กับไทรแอก ( $-V_{BO(0)}$ ) โดยไม่ให้เกินกว่าค่าแรงดันพังทลาย สามารถจ่ายกระแสเกตลบ ( $-I_G$ ) กระแสจะไหลจากขั้ว A2 ไปยังขั้ว A1 แสดงดังภาพที่ 2-11



ภาพที่ 2-11 กราฟลักษณะสมบัติของไตรแอก

### 2.7.3 การหยุดการนำกระแสของไตรแอก

เมื่อนำกระแสแล้วจะไม่จำเป็นต้องคงค่าแรงดันที่จ่ายกระตุ้นขา G เพราะไตรแอกจะนำกระแสต่อเนื่องได้เหมือนกับ SCR การจะหยุดนำกระแสทำได้ 2 วิธีเหมือน SCR คือ

- ตัดแหล่งจ่ายแรงดัน  $V_{AA}$  ที่ป้อนให้ขา A2 และขา A1 ของไตรแอกออกชั่วขณะ
- ลดแรงดันไบอัสตรงที่จ่ายให้ขา A2 และ A1 ลง จนทำให้มีกระแสไหลผ่านตัวไตรแอกต่ำกว่ากระแสโฮลดิ้ง (Holding current,  $I_H$ ) ของไตรแอก

### 2.7.4 รายละเอียดและขีดจำกัดของไตรแอก<sup>[13]</sup>

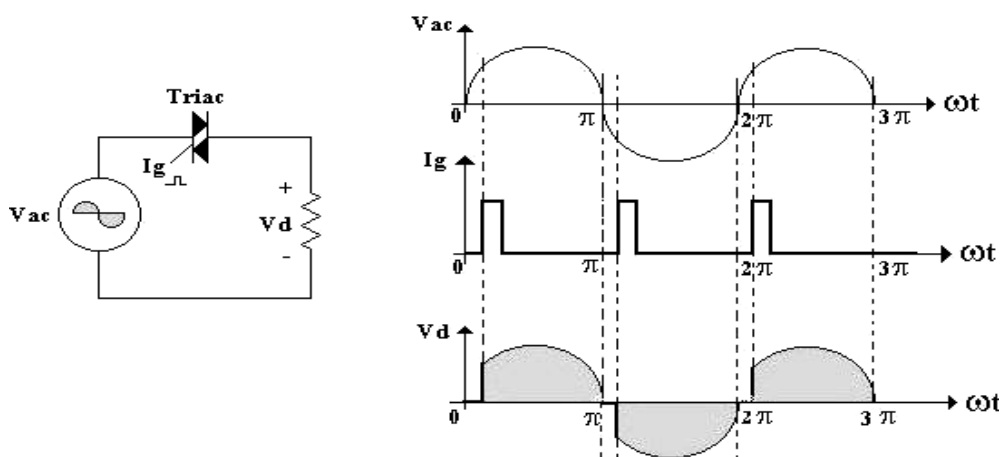
- แรงดันบล็อกกิ้งสูงสุดซ้ำ (Repetitive Peak Blocking Voltage) ใช้ตัวย่อเป็น  $V_{DRM}$  หรือ  $V_{RPM}$  คือแรงดันสูงสุดที่ป้อนเฉพาะขา A2 และ A1 ที่ตัวไตรแอกยังไม่นำกระแสโดยไม่มีแรงดันกระตุ้นที่ขา G
- กระแสไบอัสตรงสูงสุดเป็น RMS (RMS Max Forward Current) ใช้ตัวย่อ  $I_{T(MAX)}$  คือ ค่ากระแสสูงสุดเป็น RMS ที่ไตรแอกทนได้เมื่อมีกระแสไบอัสตรงไหลผ่าน ถ้ากระแสไหลผ่านไตรแอกเกินกว่าค่านี้ไตรแอกจะชำรุดเสียหาย
- กระแสโฮลดิ้ง (Holding Current) ใช้ตัวย่อ  $I_H$  คือกระแสต่ำสุดที่ไหลผ่านตัวไตรแอกระหว่างขา A2 และ A1 แล้วไตรแอกยังคงนำกระแสได้
- กระแสกระตุ้นเกต (Gate Trigger Current) ใช้ตัวย่อ  $I_{GT}$  คือกระแสที่ใช้กระตุ้นขา G ของไตรแอกเพื่อให้ไตรแอกทำงานมักบอกค่ากระแสไว้ที่ต่ำสุด

- แรงดันกระตุ้นเกต (Gate Trigger Voltage) ใช้ตัวย่อ  $V_{GT}$  คือแรงดันที่ป้อนให้ขา G เทียบกับขา A1 เพื่อกระตุ้นให้ไทรแอกนำกระแสมีค่าสูงสุด
- กระแสกระตุ้นเกตสูงสุด (Peak Gate Trigger Current) ใช้ตัวย่อ  $I_{GTM}$  คือ ค่ากระแสกระตุ้นที่ไหลผ่านเกตค่าสูงสุดยังคงทำให้ไทรแอกทำงานได้
- ย่านอุณหภูมิในการทำงาน (Operating Junction Temperature) ใช้ตัวย่อ  $T_j$  คือ ย่าน อุณหภูมิตรงรอยต่อที่ไทรแอกทำงานได้โดยไม่ชำรุดเสียหาย บอกค่าตั้งแต่ต่ำสุด (-) จนถึงสูงสุดเป็นค่าบวก (+)

#### 2.7.5 การใช้ไทรแอกควบคุมเฟสของแรงดันไฟสลับ

ไทรแอกสามารถนำกระแสในไฟสลับได้ 2 ซีก คือซีกบวกและซีกลบในการควบคุมไฟสลับสามารถทำได้กับไฟสลับ 1 เฟส ดังนี้

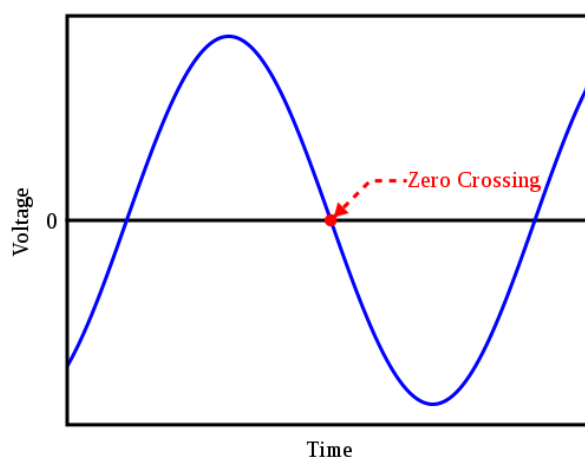
- ควบคุมแรงดันไฟสลับชนิดเฟสเดียว ถ้าใช้ไทรแอกจะใช้เพียงตัวเดียว แต่ถ้าใช้ SCR จะต้องใช้ 2 ตัวแสดงดังภาพที่ 2-12



ภาพที่ 2-12 รูปคลื่นแรงดันตกคร่อมโหลด

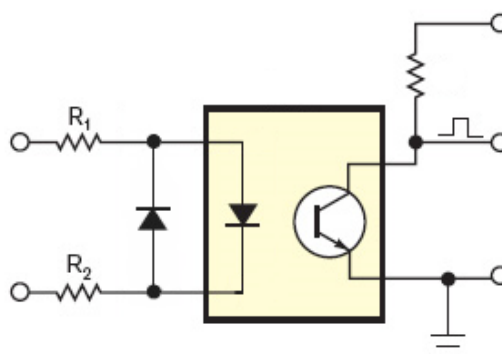
#### 2.8 วงจรตรวจจับจุดผ่านศูนย์

ในไฟกระแสสลับ จุดผ่านศูนย์ คือ จุดที่แรงดันเปลี่ยนจากค่าบวกเป็นค่าลบ หรือเวลาชั่วขณะที่มีแรงดันเป็นศูนย์ ของสัญญาณคลื่นไซน์และสัญญาณพื้นฐานอื่น ๆ โดยปกติแล้วจะมีสองครั้งต่อหนึ่งลูกคลื่น แสดงดังภาพที่ 2-13 ในการตรวจจับจุดผ่านศูนย์สามารถทำได้หลายแบบ เช่น ใช้โอปแอมป์ในการเปรียบเทียบแรงดัน หรือใช้อุปกรณ์เชื่อมต่อทางแสง เป็นต้น



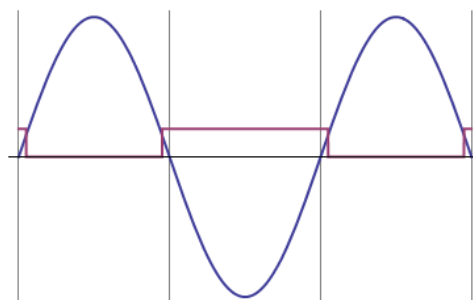
ภาพที่ 2-13 จุดผ่านศูนย์<sup>[13]</sup>

จากภาพที่ 2-13 จะเห็นว่าที่จุดตัดศูนย์ก็คือจุดที่อุปกรณ์เชื่อมต่อทางแสงจะไม่ทำงาน เมื่อนำอุปกรณ์เชื่อมต่อทางแสงมาต่อโดยให้เอาต์พุตต่อแบบคอลเลกเตอร์เอาต์พุต แสดงดังภาพที่ 2-14



ภาพที่ 2-14 วงจรตรวจจับจุดผ่านศูนย์ด้วยอุปกรณ์เชื่อมต่อทางแสง<sup>[14]</sup>

จากภาพที่ 2-14 จะเห็นว่ามีการต่อตัวต้านทานฝั่งอินพุตเพื่อจำกัดกระแสเข้าอุปกรณ์ และมีการต่อไดโอดขนานแบบกลับขั้วกับตัวไดโอดภายในให้กระมีความต่อเนื่อง และต่อเอาต์พุตระหว่างตัวต้านทานกับขาคอลเลกเตอร์ จะทำให้ได้เอาต์พุตในช่วงที่แรงดันฝั่งอินพุตมีค่าน้อยกว่าแรงดันขั้นต่ำที่ทำให้ตัวไดโอดเปล่งแสงออกมามากพอให้ทรานซิสเตอร์เปิดวงจร โดยเอาต์พุตที่ได้จะมีค่าเท่ากับแรงดันแหล่งจ่ายฝั่งเอาต์พุต แสดงดังภาพที่ 2-15



ภาพที่ 2-15 รูปสัญญาณของวงจรตรวจจับจุดผ่านศูนย์<sup>[14]</sup>

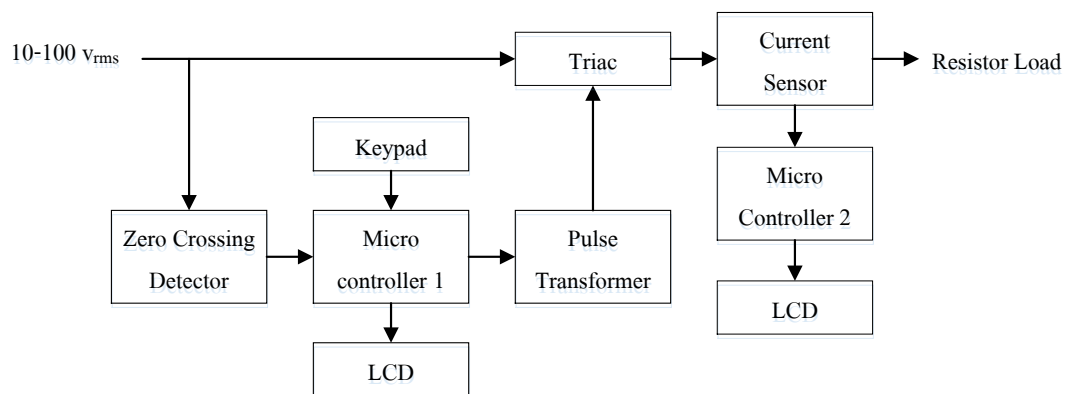
จากภาพที่ 2-15 จะเห็นว่าเมื่อแรงดันรูปคลื่นไซน์เข้ามาฝั่งอินพุต มีค่ามากพอจะทำให้ทรานซิสเตอร์เปิดวงจร เอาต์พุตที่ได้จะเป็นศูนย์

## บทที่ 3

### ขั้นตอนและวิธีการดำเนินงาน

#### 3.1 บล็อกไดอะแกรมรวมของเครื่องกำเนิดกระแสเกินชั่วคราว

เครื่องกำเนิดกระแสเกินชั่วคราวที่สร้างขึ้นในโครงการนี้ มีส่วนประกอบหลายส่วน สามารถแสดงโดยบล็อกไดอะแกรม ดังภาพที่ 3-1



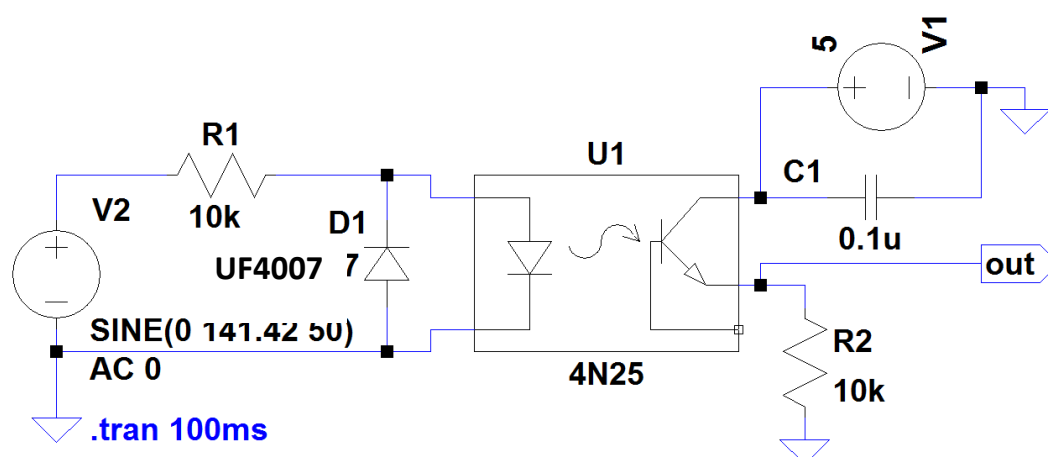
ภาพที่ 3-1 บล็อกไดอะแกรมรวมเครื่องกำเนิดกระแสเกินชั่วคราว

จากภาพที่ 3-1 จะเห็นได้ว่ามีส่วนประกอบต่าง ๆ ดังนี้คือ ส่วนวงจรตรวจจับจุดผ่านศูนย์ ส่วนไมโครคอนโทรลเลอร์1 ส่วนวงจรจับเกต ส่วนเซนเซอร์กระแส และส่วนไมโครคอนโทรลเลอร์2 โดยจะแสดงรายละเอียดของแต่ละส่วนในหัวข้อต่อไป

#### 3.2 การจำลองการทำงานส่วนต่าง ๆ ของเครื่องกำเนิดกระแสเกินชั่วคราว

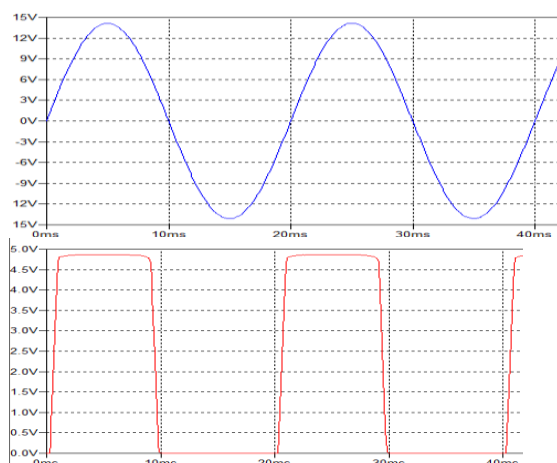
##### 3.2.1 วงจรตรวจจับจุดผ่านศูนย์

วงจรตรวจจับจุดผ่านศูนย์มีหน้าที่ตรวจจับจุดที่แรงดันเป็นศูนย์ แล้วส่งสัญญาณไปให้ส่วนไมโครคอนโทรลเลอร์1 ซึ่งในวงจรส่วนนี้ใช้อุปกรณ์เชื่อมต่อทางแสง 1 ตัว โดยมีตัวต้านทาน 1 ตัวทำหน้าที่จำกัดกระแสที่ไหลเข้าฟัองอินพุต และมีแหล่งจ่ายแรงดันไฟกระแสตรง 5 โวลต์ ซึ่งต่อขนานกับตัวเก็บประจุ ต่อเข้ากับขาคอลเลกเตอร์ของทรานซิสเตอร์ แสดงดังภาพที่ 3-2

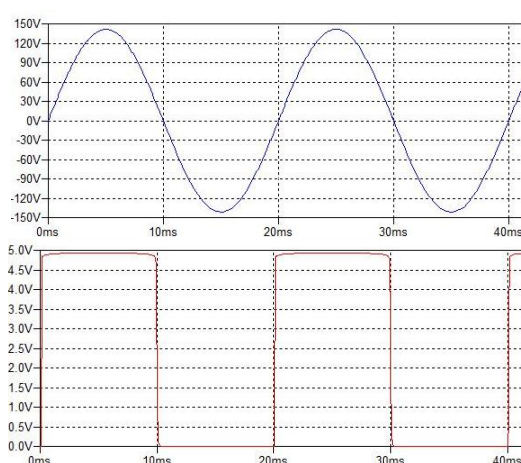


ภาพที่ 3-2 วงจรตรวจจับจุดผ่านศูนย์

จากภาพที่ 3-2 จะเห็นว่ากำหนดค่าตัวต้านทาน  $R_1 = R_2 = 10 \text{ k}\Omega$  โดยสามารถหาค่าความต้านทานได้จากสมการกฎของโอห์ม ( $V=IR$ ) เมื่อ  $V_{\max} = 100 \times \sqrt{2} = 141.42 \text{ V}$ ,  $I_{\text{rate}} = 50 \text{ mA}$  จะได้  $R_1 \geq 2.828 \text{ k}$  ไดโอด D1 ใช้เบอร์ UF4007 ตัวเก็บประจุ  $C_1 = 0.1 \text{ uF}$  และอุปกรณ์เชื่อมต่อทางแสง หรือ ออปโตไเบอร์ 4N25 ผลการจำลองการทำงานของวงจรตรวจจับจุดผ่านศูนย์ที่ได้ แสดงดังภาพที่ 3-3



(ก.) ที่อินพุต  $10 \text{ V}_{\text{RMS}}$



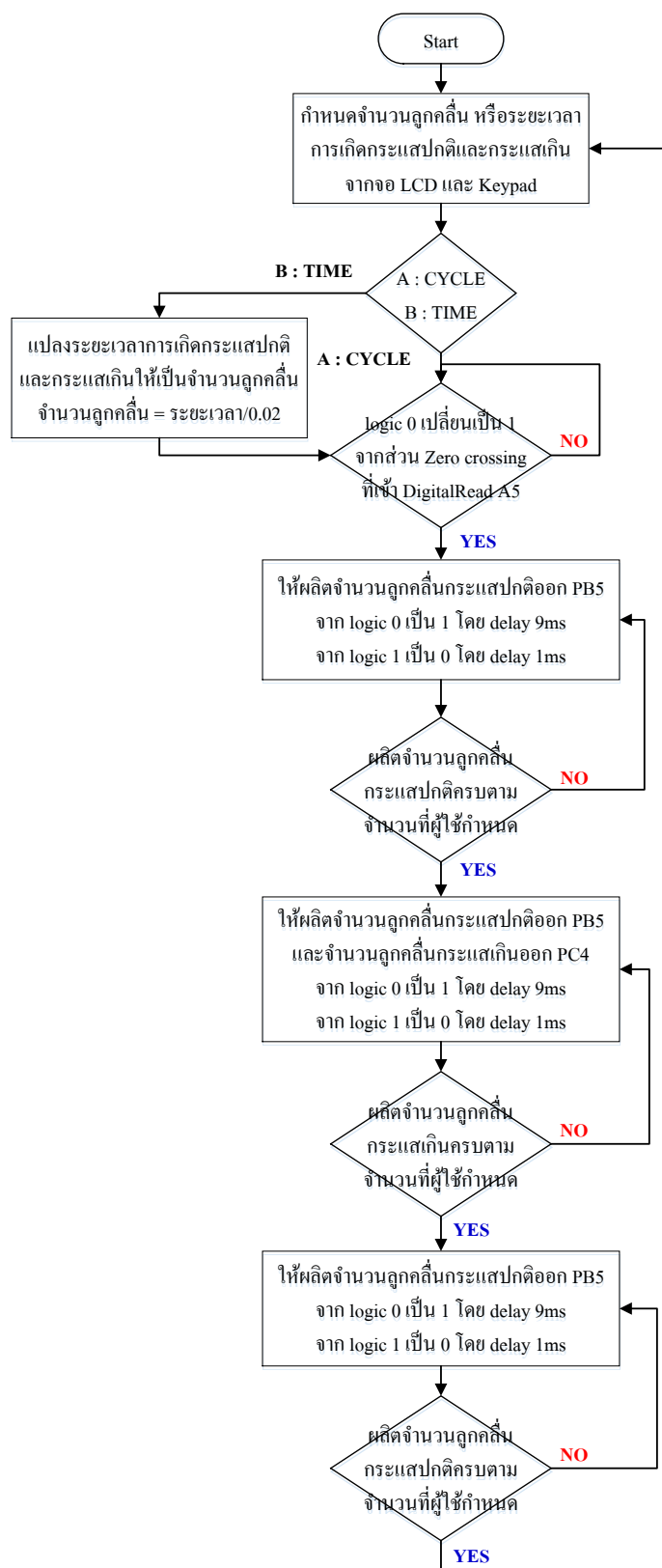
(ข.) ที่อินพุต  $100 \text{ V}_{\text{RMS}}$

ภาพที่ 3-3 ผลการจำลองการทำงานของวงจรตรวจจับจุดผ่านศูนย์



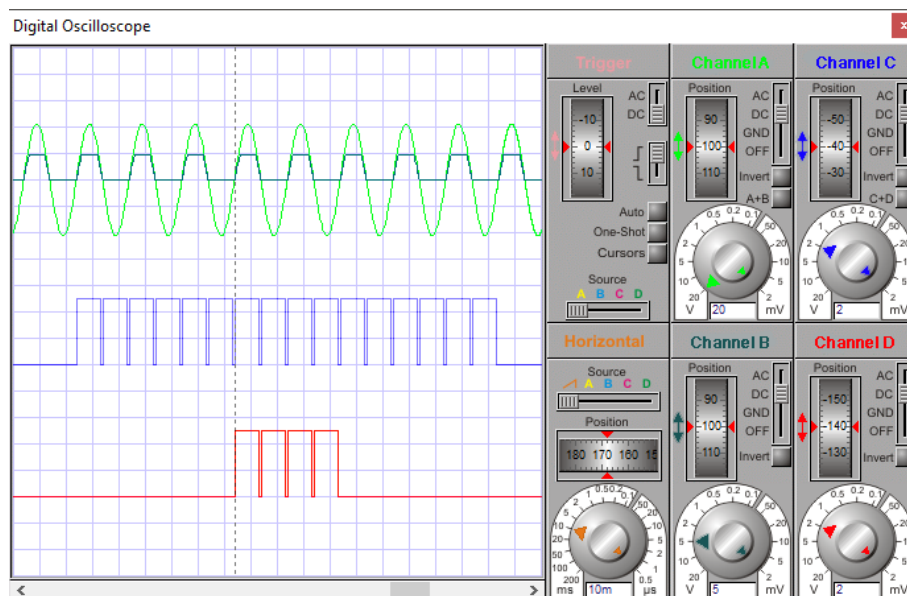
### 3.2.2 ไมโครคอนโทรลเลอร์1

ไมโครคอนโทรลเลอร์1 นี้ จะเป็นการใช้บอร์ดอาduino และภาษา C++ เขียนลงในโปรแกรมอาduino เพื่อออกแบบให้ตัวเครื่องมีฟังก์ชันการใช้งานตามต้องการ โดยเครื่องกำเนิดกระแสเกินชั่วคราวจะแสดงข้อความผ่านทางหน้าจอ LCD และทำการรับค่าจากคีย์แพดโดยผู้ใช้งาน เริ่มแรกจะให้ผู้ใช้งานเลือกว่าจะกำหนดค่า จำนวนลูกกลิ้ง หรือ ระยะเวลาที่เกิด (วินาที) ของทั้งกระแสปกติและกระแสเกิน หลังจากนั้นก็สามารถป้อนค่าจำนวนตัวเลขตามที่ต้องการได้ เสมือนเป็นวงจรสร้างสัญญาณขบวนพัลส์ โดยบล็อกไดอะแกรมของโปรแกรมในไมโครคอนโทรลเลอร์1 แสดงดังภาพที่ 3-4 และวงจรไมโครคอนโทรลเลอร์1 แสดงดังภาพที่ 3-5



ภาพที่ 3-4 บล็อกไดอะแกรมของโปรแกรมในไมโครคอนโทรลเลอร์ 1

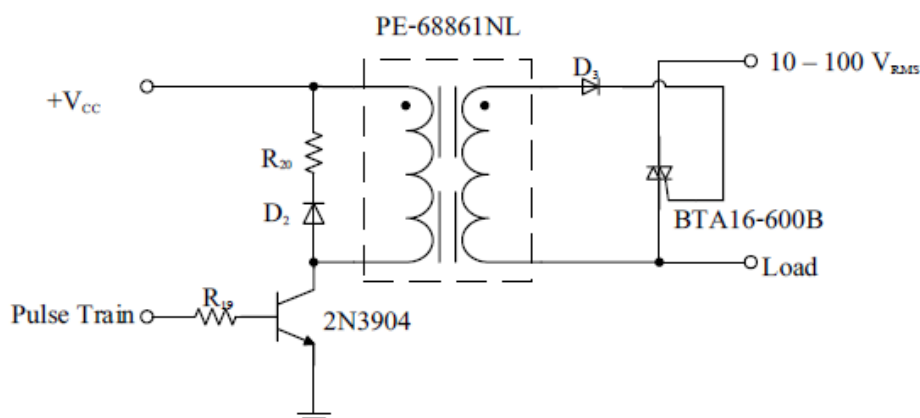




ภาพที่ 3-6 ผลการจำลองการทำงานของไมโครคอนโทรลเลอร์ 1

### 3.2.3 วงจรขับเคลื่อน

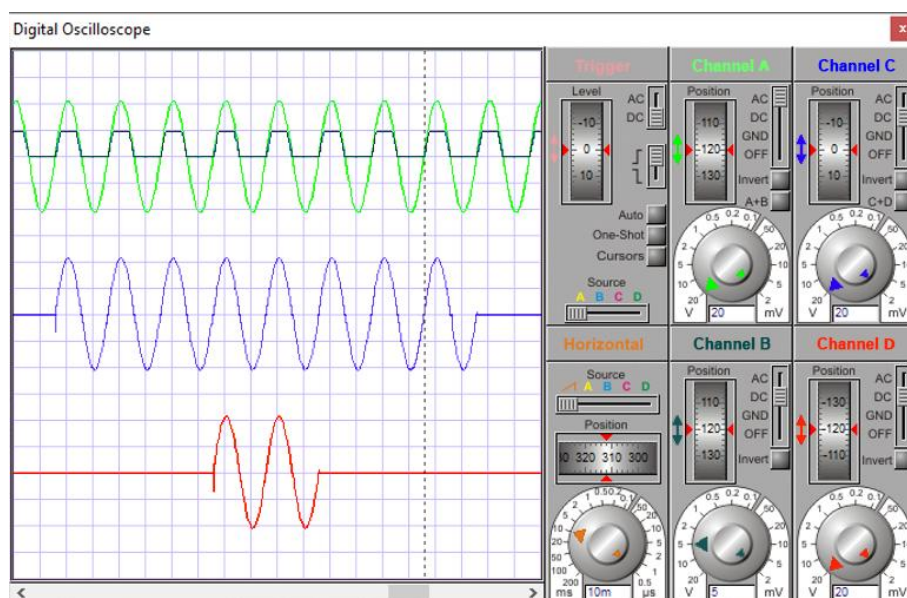
วงจรขับเคลื่อนมีหน้าที่ขยายสัญญาณพัลส์ที่ได้จากไมโครคอนโทรลเลอร์ 1 เพื่อนำมาขับเคลื่อนของไตรแอก ด้วยความสามารถในการขยายสัญญาณของทรานซิสเตอร์ โดยต่อสัญญาณพัลส์ผ่านตัวต้านทานจำกัดกระแสเข้าขาเบส และต่อแรงดันแหล่งจ่ายผ่านหม้อแปลงพัลส์ฟลักซ์ปฐมภูมิเข้าขาคอลเลกเตอร์ของทรานซิสเตอร์ โดยมีไดโอดต่ออนุกรมกับตัวต้านทานต่อขนานอยู่ และทางฝั่งทุติยภูมิก็ต่อผ่านไดโอดเข้าขาเกตของไตรแอก แสดงดังภาพที่ 3-7



ภาพที่ 3-7 วงจรขับเคลื่อน

จากภาพที่ 3-7 จะเห็นว่ากำหนดค่าตัวต้านทาน  $R19 = 100\ \Omega$  และ  $R20 = 27\ \Omega$  ไดโอด D2 และ D3 ใช้เบอร์ UF4007 ทรานซิสเตอร์ชนิด NPN เบอร์ 2N3904 หม้อแปลงฟัลส์เบอร์ PE-68861 และไดรแอกเบอร์ BTA26-600B ทำหน้าที่เป็นสวิตช์สำหรับจ่ายกระแสให้โหลด

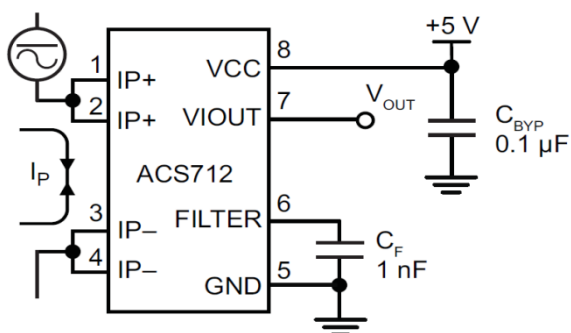
ผลการจำลองการทำงานของวงจรจับเกต โดยเลือกกำหนดจำนวนลูกคลื่นให้เกิดกระแสปกติ 3 ลูก และกระแสเกิน 2 ลูก ใช้แรงดันอินพุตเป็นไฟกระแสสลับ  $30\ V_{RMS}$  แสดงโดย Channel A ( $20\ V/div$ ) และ  $Time/div = 10\ ms/div$  ผลจากวงจรตรวจจับจุดผ่านศูนย์ แสดงโดย Channel B ( $5\ V/div$ ) และ  $Time/div = 10\ ms/div$  ได้แรงดันขนาด  $3.3\ V_{RMS}$  ผลการทดสอบเอาต์พุตวงจรจับเกตของกระแสปกติ แสดงโดย Channel C ( $20\ V/div$ ) และ  $Time/div = 10\ ms/div$  และของกระแสเกิน แสดงโดย Channel D ( $20\ V/div$ ) และ  $Time/div = 10\ ms/div$  แสดงดังภาพที่ 3-8



ภาพที่ 3-ค ผลการจำลองการทำงานของวงจรจับเกต

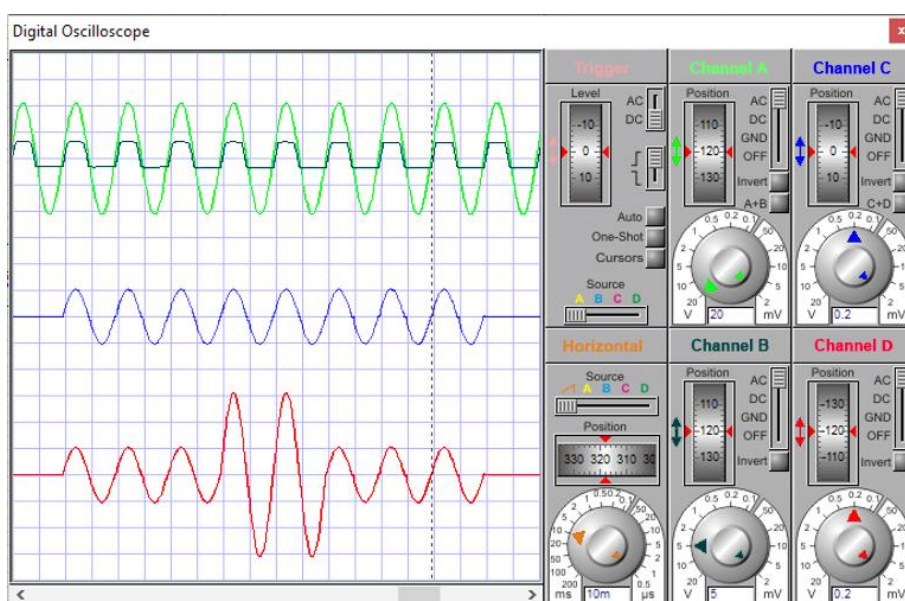
### 3.2.4 เซนเซอร์กระแส

เซนเซอร์กระแสมีหน้าที่แปลงค่ากระแสจริงที่วัดได้ให้เป็นระดับแรงดันที่มีอัตราส่วน  $1A = 0.1V$  เพื่อให้บอร์ดไมโครคอนโทรลเลอร์ 2 สามารถอ่านค่าได้ โดยจะต่อเซนเซอร์กระแสแบบอนุกรมด้านหลังโหลดตัวต้านทาน แสดงดังภาพที่ 3-9



ภาพที่ 3-9 วงจรเซนเซอร์กระแส

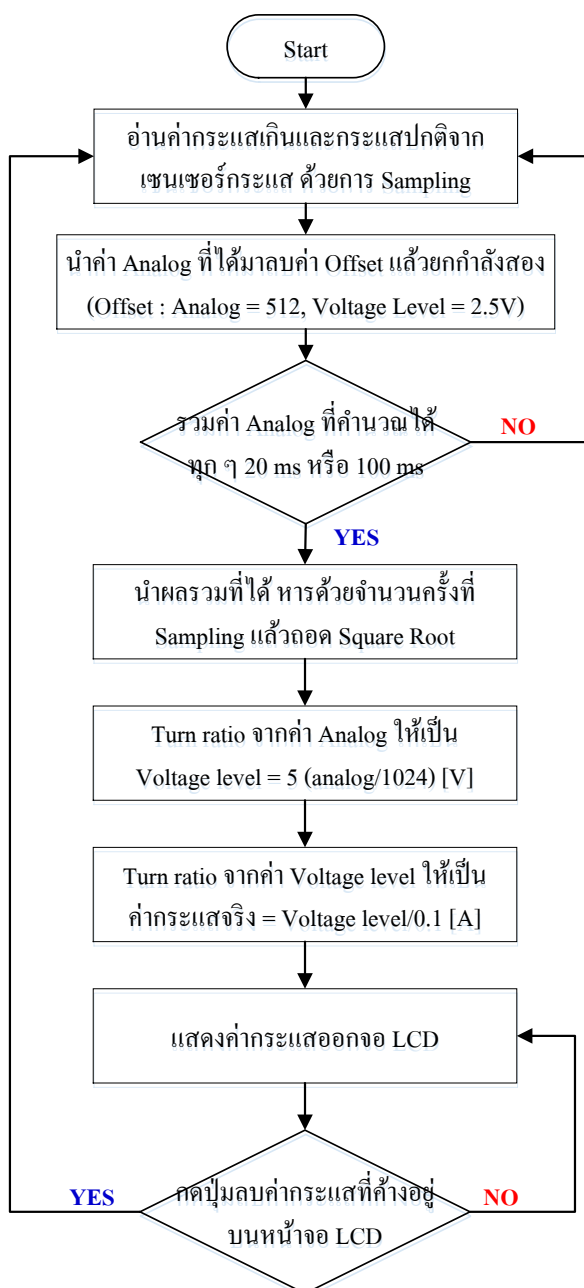
ผลการจำลองการทำงานของเซนเซอร์กระแส โดยเลือกกำหนดจำนวนลูกคลื่นให้เกิดกระแสปกติ 3 ลูก และกระแสเกิน 2 ลูก ใช้แรงดันอินพุตเป็นไฟกระแสสลับ 30 V<sub>RMS</sub> แสดงโดย Channel A (20 V/div) และ Time/div = 10 ms/div ต่อกับโหลดตัวต้านทานของกระแสปกติ 20 และ โหลดตัวต้านทานของกระแสเกิน 10 ผลจากวงจรตรวจจับจุดผ่านศูนย์ แสดงโดย Channel B (5 V/div) และ Time/div = 10 ms/div ได้แรงดันขนาด 3.3 V<sub>RMS</sub> ผลการทดสอบเอาต์พุตเซนเซอร์กระแสของกระแสปกติ แสดงโดย Channel C (0.2 V/div) และ Time/div = 10 ms/div และของกระแสปกติรวมกับกระแสเกิน แสดงโดย Channel D (0.2 V/div) และ Time/div = 10 ms/div แสดงดังภาพที่ 3-10



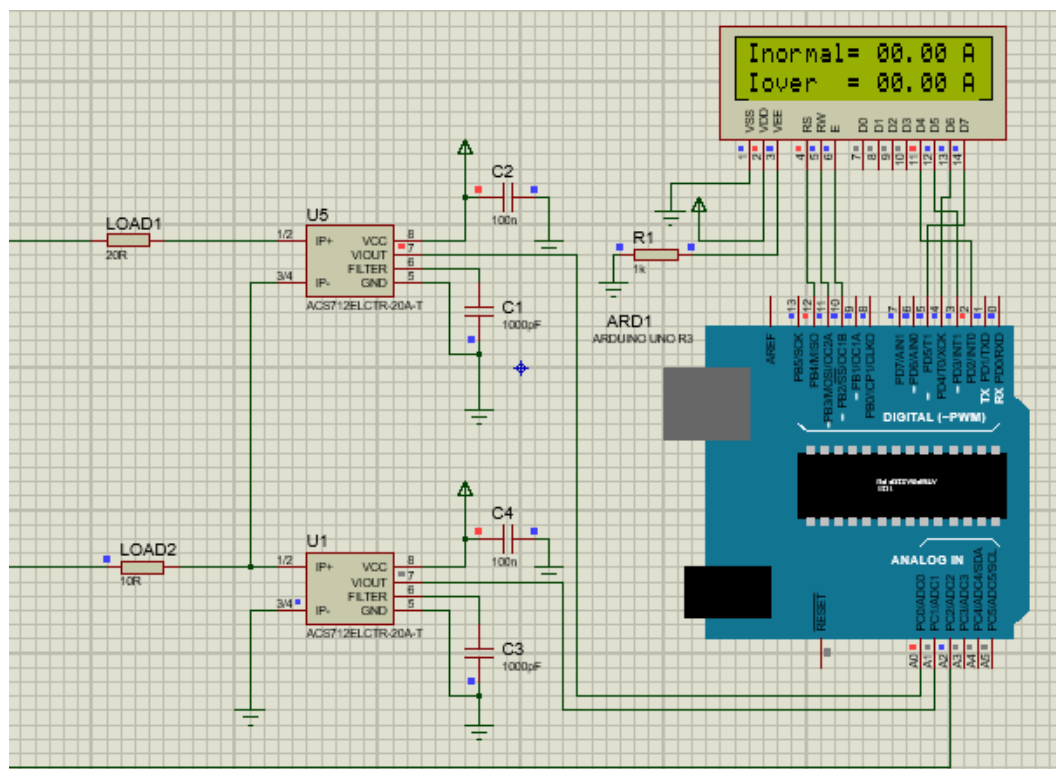
ภาพที่ 3-10 ผลการจำลองการทำงานของเซนเซอร์กระแส

### 3.2.5 ไมโครคอนโทรลเลอร์ 2

ไมโครคอนโทรลเลอร์ 2 นี้ จะเป็นการใช้บอร์ดอาδυโน้ และภาษา C++ เขียนลงในโปรแกรมอาδυโน้เช่นเดิม เพื่อแปลงอัตราส่วนแรงดันที่เซนเซอร์กระแสส่งให้บอร์ดไมโครคอนโทรลเลอร์ 2 ให้เป็นค่ากระแสจริงทั้งค่ากระแสปกติและกระแสเกิน แล้วแสดงค่ากระแสผ่านทางหน้าจอ LCD โดยบล็อกไดอะแกรมโปรแกรมไมโครคอนโทรลเลอร์ 2 แสดงดังภาพที่ 3-11 และวงจรไมโครคอนโทรลเลอร์ 2 แสดงดังภาพที่ 3-12



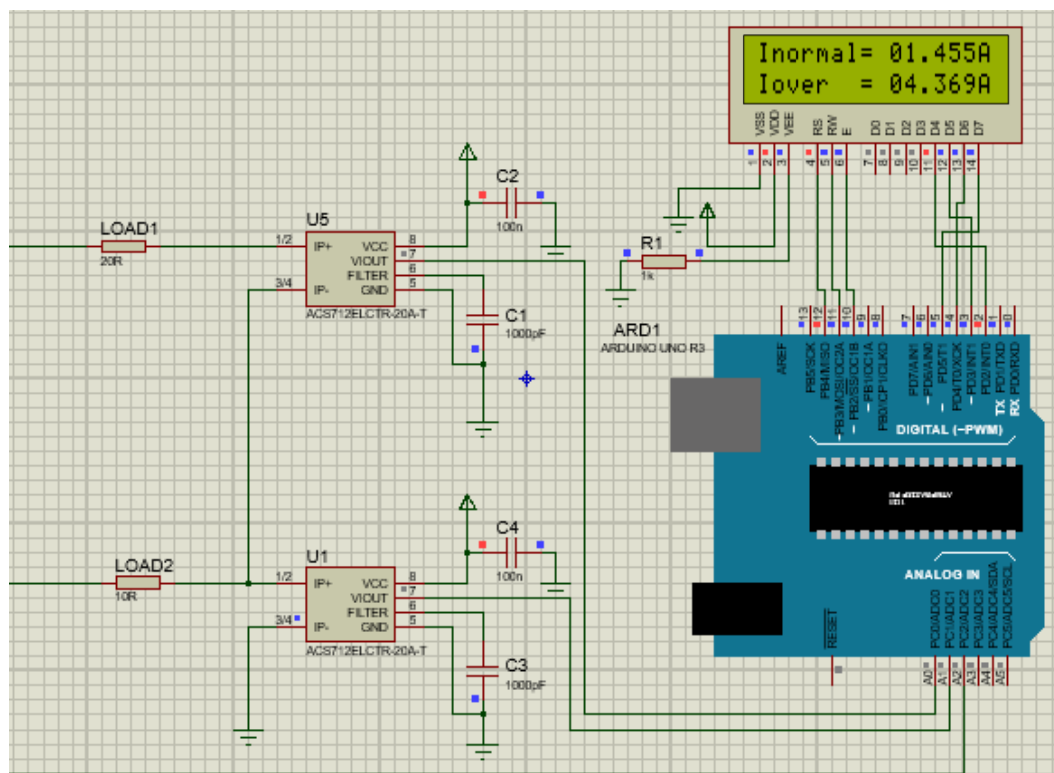
ภาพที่ 3-11 บล็อกไดอะแกรมโปรแกรมไมโครคอนโทรลเลอร์ 2



ภาพที่ 3-12 วงจรไมโครคอนโทรลเลอร์ 2

ผลการจำลองการทำงานของไมโครคอนโทรลเลอร์ 2 โดยเลือกกำหนดจำนวนลูกคลื่นให้เกิดกระแสปกติ 3 ลูก และกระแสเกิน 2 ลูก ใช้แรงดันอินพุตเป็นไฟกระแสสลับ  $30\text{ V}_{\text{RMS}}$  ต่อกับโหลดตัวต้านทานของกระแสปกติ 20 และโหลดตัวต้านทานของกระแสเกิน 10 ได้ผลการทดสอบเอาต์พุตไมโครคอนโทรลเลอร์ 2 แสดงดังภาพที่ 3-13

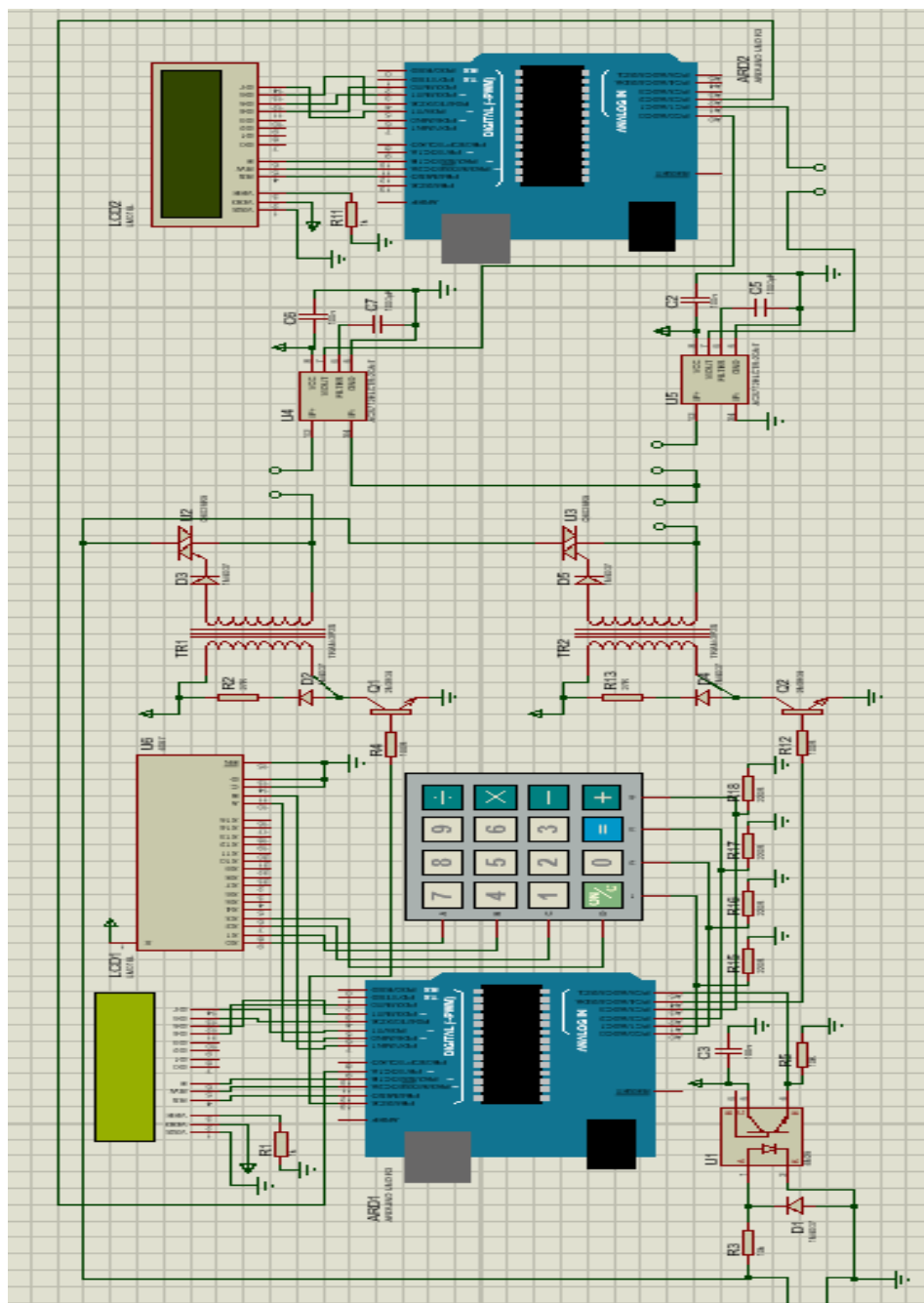




ภาพที่ 3-13 ผลการจำลองการทำงานของไมโครคอนโทรลเลอร์ 2

### 3.3 วงจรทั้งหมดของเครื่องกำเนิดกระแสเกินชั่วคราว

วงจรทั้งหมดของเครื่องกำเนิดกระแสเกินชั่วคราวที่จะนำไปสร้างจริง แสดงดังภาพที่ 3-14



ภาพที่ 3-14 วงจรทั้งหมดของเครื่องกำเนิดกระแสเกินชั่วคราวที่จะนำไปสร้างจริง

ตารางที่ 3-1 รายการและราคาโดยประมาณของอุปกรณ์ที่ใช้ในโครงการ

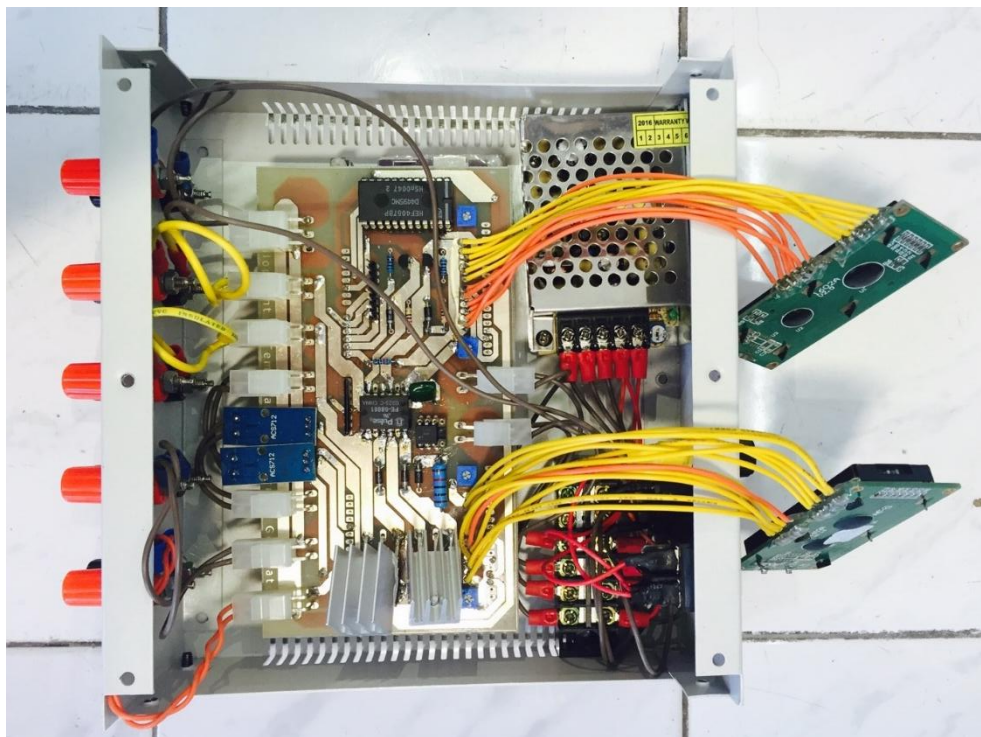
ลำดับ	รายการ	จำนวน	ราคา
1	ไครแอค เบอร์ BTA26-600B	2	104
2	หม้อแปลงฟัลส์ เบอร์ PE-68861NL	1	85
3	ออปโต้ เบอร์ 4N25	1	6
4	บอร์ดอาคูนี UNO R3	2	1,450
5	คีย์แพด 4x4	1	40
6	จอแอลซีดี 16x2	2	200
7	เซนเซอร์กระแส เบอร์ ACS712	2	300
8	ทรานซิสเตอร์ เบอร์ 2N3904	2	2
9	ไอโอด เบอร์ UF4007	5	20
10	ตัวเก็บประจุ ขนาด 100 nF 100 V	1	2
11	ตัวต้านทานแบบเน็ทเวอร์ค 330 $\Omega$	1	2
12	ตัวต้านทานปรับค่าได้ 10 k $\Omega$	4	32
13	ตัวต้านทาน ขนาด 10 k $\Omega$ 2 W	1	5
14	ตัวต้านทาน ขนาด 27 $\Omega$ 1/2 W	2	2
15	ตัวต้านทาน ขนาด 100 $\Omega$ 1/2 W	2	2
16	ตัวต้านทาน ขนาด 10 k $\Omega$ 1/2 W	1	1
17	แหล่งจ่ายไฟตรง 5 V	1	175
18	แผ่นวงจร (พร้อมลายวงจร)	1	200
19	กล่องใส่บอร์ดอาคูนี UNO R3	2	900
20	เทอมินอล	10	90
21	กล่องและอุปกรณ์อื่น ๆ	1	400
	รวม		4,018

## บทที่ 4

### ผลการดำเนินงาน

#### 4.1 วงจรจริงของเครื่องกำเนิดกระแสเกินชั่วคราว

หลังจากทำการจำลองผลการทำงานของวงจรแต่ละส่วนจนได้ผลที่ตรงตามต้องการแล้ว จึงทำการประกอบวงจรรวมของเครื่องกำเนิดกระแสเกินชั่วคราวขึ้น แสดงดังภาพที่ 4-1 และทำการทดสอบวงจรที่ได้ โดยผลการทดสอบการทำงานของวงจรจะกล่าวในหัวข้อต่อไป

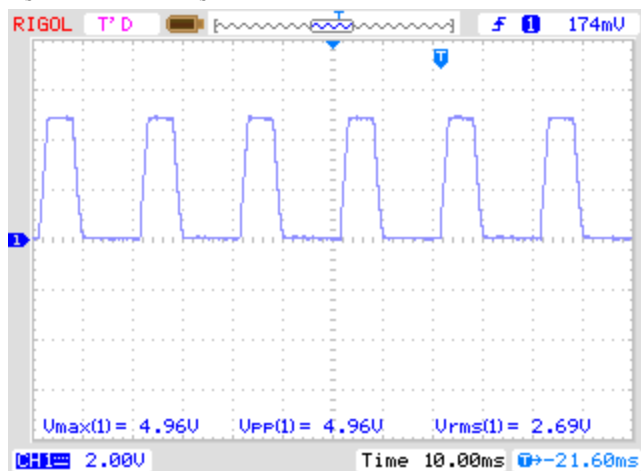


ภาพที่ 4-1 วงจรจริงของเครื่องกำเนิดกระแสเกินชั่วคราว

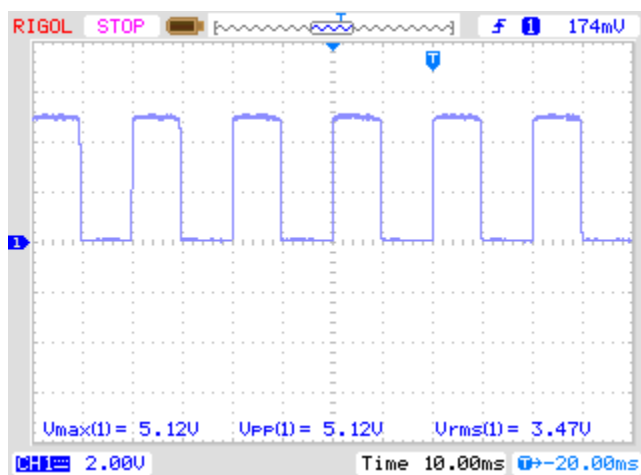
#### 4.2 การทดสอบการทำงานส่วนต่าง ๆ ของเครื่องกำเนิดกระแสเกินชั่วคราว

##### 4.2.1 วงจรตรวจจับจุดผ่านศูนย์

ผลการทดสอบการทำงานของวงจรตรวจจับจุดผ่านศูนย์ โดยใช้แรงดันอินพุตเป็นไฟ  
กระแสสลับ  $10\text{ V}_{\text{RMS}}$  และ  $100\text{ V}_{\text{RMS}}$  แสดงโดย Ch.1 ( $2\text{ V/div}$ ) และ  $\text{Time/div} = 10\text{ ms/div}$  ได้แรงดัน  
เอาต์พุตขนาด  $2.69\text{ V}_{\text{RMS}}$  และ  $3.47\text{ V}_{\text{RMS}}$  แสดงดังภาพที่ 4-2 และภาพที่ 4-3 ตามลำดับ



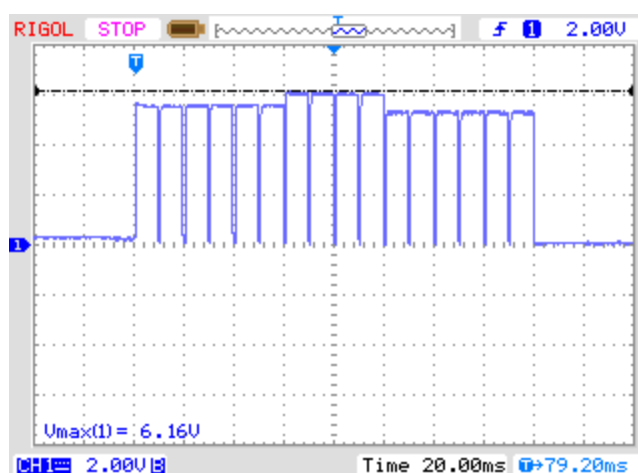
ภาพที่ 4-2 ผลการทดสอบการทำงานของวงจรตรวจจับจุดผ่านศูนย์ ที่แรงดันอินพุต  $10\text{ V}_{\text{RMS}}$



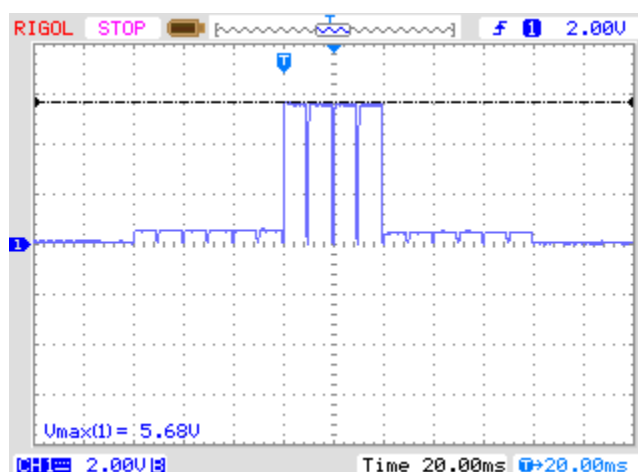
ภาพที่ 4-3 ผลการทดสอบการทำงานของวงจรตรวจจับจุดผ่านศูนย์ ที่แรงดันอินพุต  $100\text{ V}_{\text{RMS}}$

#### 4.2.2 ไมโครคอนโทรลเลอร์ 1

ผลการทดสอบการทำงานของไมโครคอนโทรลเลอร์ 1 โดยเลือกกำหนดจำนวนลูกคลื่นให้  
เกิดกระแสปกติ 3 ลูก และกระแสเกิน 2 ลูก ได้เอาต์พุตของกระแสปกติ และกระแสเกิน แสดงโดย  
Ch.1 ( $2\text{ V/div}$ ) และ  $\text{Time/div} = 20\text{ ms/div}$  แสดงดังภาพที่ 4-4 และภาพที่ 4-5 ตามลำดับ



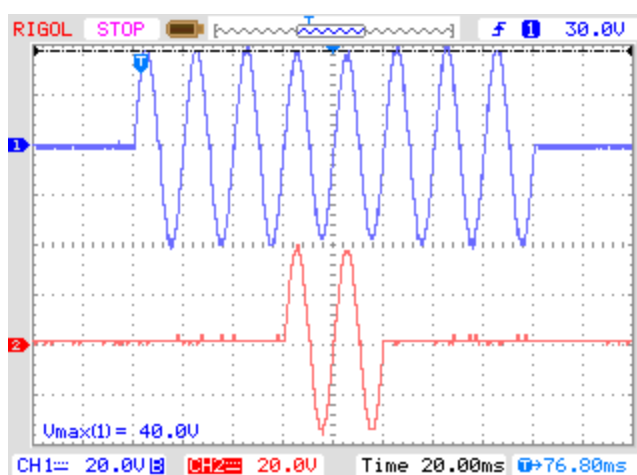
ภาพที่ 4-4 ผลการทดสอบการทำงานของไมโครคอนโทรลเลอร์ 1 ของกระแสปกติ



ภาพที่ 4-5 ผลการทดสอบการทำงานของไมโครคอนโทรลเลอร์ 1 ของกระแสเกิน

#### 4.2.3 วงจรขั้วเกต

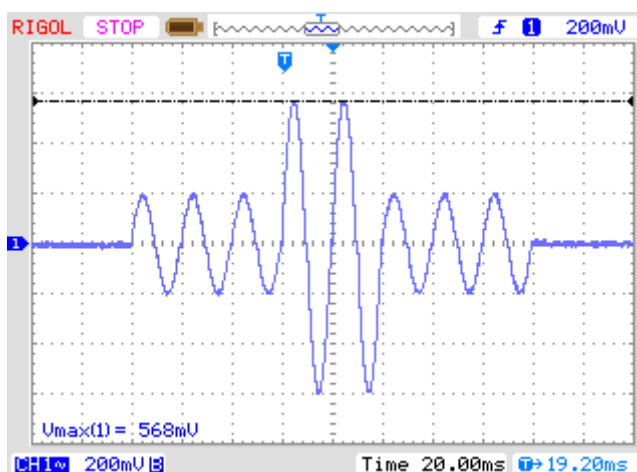
ผลการทดสอบการทำงานของวงจรขั้วเกต โดยเลือกกำหนดจำนวนลูกคลื่นให้เกิดกระแสปกติ 3 ลูก และกระแสเกิน 2 ลูก ได้เอาที่พูดของกระแสปกติแสดงโดย Ch.1 และของกระแสเกินแสดงโดย Ch.2 (20 V/div) และ Time/div = 20 ms/div แสดงดังภาพที่ 4-6



ภาพที่ 4-6 ผลการทดสอบการทำงานของวงจรขับเกา

#### 4.2.4 เซนเซอร์กระแส

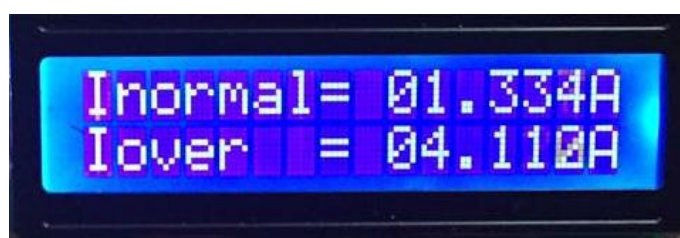
ผลการทดสอบการทำงานของเซนเซอร์กระแส โดยเลือกกำหนดจำนวนลูกคลื่นให้เกิดกระแสปกติ 3 ลูก และกระแสเกิน 2 ลูก ได้เอาต์พุตของกระแสปกติและกระแสปกติรวมกับกระแสเกินแสดงโดย Ch.1 (0.2 V/div) และ Time/div = 20 ms/div แสดงดังภาพที่ 4-7



ภาพที่ 4-7 ผลการทดสอบการทำงานของเซนเซอร์กระแส

#### 4.2.5 ไมโครคอนโทรลเลอร์ 2

ผลการทดสอบการทำงานของไมโครคอนโทรลเลอร์ 2 ได้ค่ากระแสจริงทั้งค่ากระแสปกติและกระแสเกินจากการอ่านค่าของเซนเซอร์กระแส แสดงดังภาพที่ 4-8



ภาพที่ 4-8 ผลการทดสอบการทำงานของไมโครคอนโทรลเลอร์ 2

#### 4.3 การทดสอบการทำงานของเครื่องกำเนิดกระแสเกินชั่วคราว

การต่อวงจรทดสอบการทำงานของเครื่องกำเนิดกระแสเกินชั่วคราว แสดงดังภาพที่ 4-9



ภาพที่ 4-9 การต่อวงจรทดสอบการทำงานของเครื่องกำเนิดกระแสเกินชั่วคราว

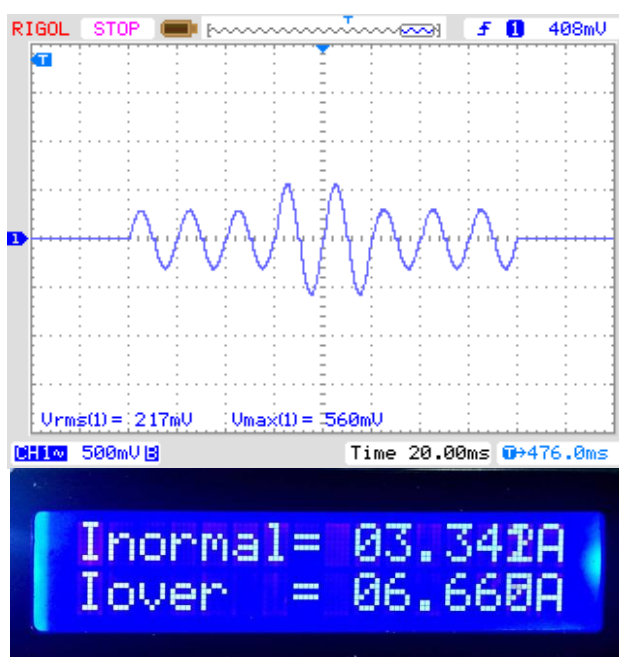
##### 4.3.1 การทดสอบทำในเงื่อนไขดังนี้

- แหล่งจ่ายไฟกระแสสลับขนาด 20 V<sub>RMS</sub>
- โหลดความต้านทานขนาด 5 Ω, 1,200 Watts 2 ชุด
- แสดงแรงดันเอาต์พุตด้วย Ch.1 (0.5 V/div)

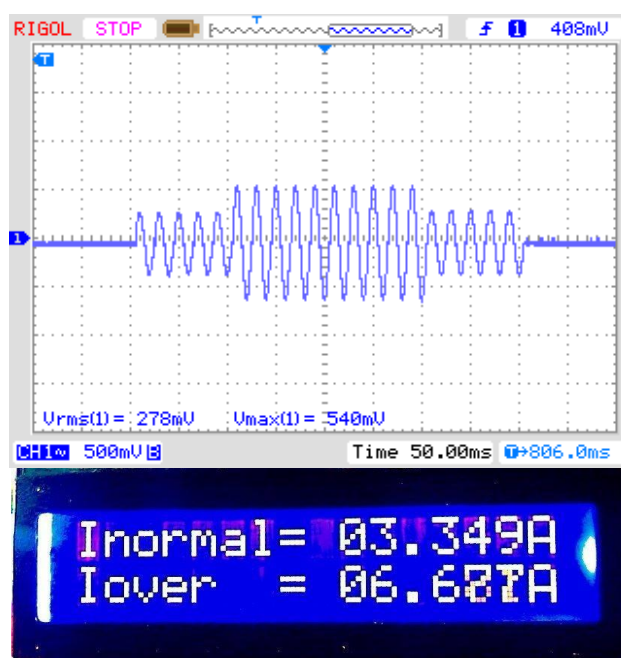


- ทำการทดสอบ 4 ครั้ง โดยกำหนดจำนวนลูกคลื่นหรือระยะเวลาของเอาต์พุตกระแส การกำหนดระยะเวลาการกระตุ้น คือ

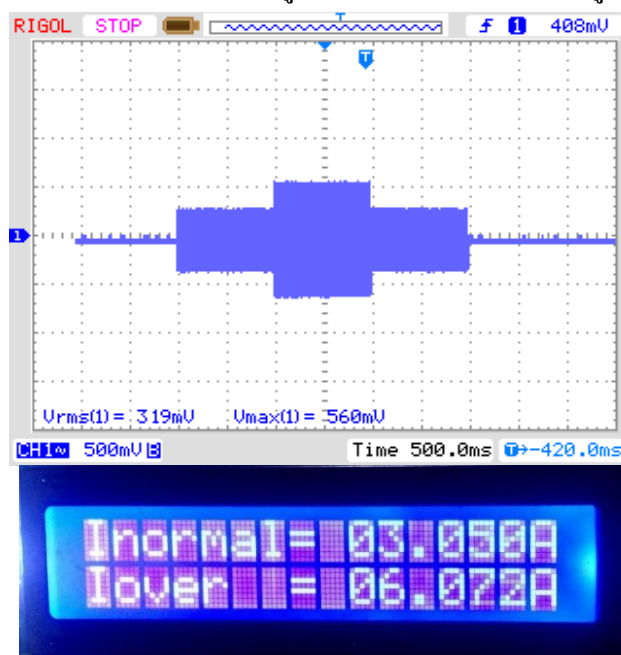
- ครั้งที่ 1 กำหนดจำนวนลูกคลื่นกระแสปกติ 3 ลูกคลื่น และกระแสเกิน 2 ลูกคลื่น ;  
160 ms , Time/div = 20 ms/div แสดงดังภาพที่ 4-10
- ครั้งที่ 2 กำหนดจำนวนลูกคลื่นกระแสปกติ 5 ลูกคลื่น และกระแสเกิน 10 ลูกคลื่น;  
400 ms, Time/div = 50 ms/div แสดงดังภาพที่ 4-11
- ครั้งที่ 3 กำหนดระยะเวลากระแสปกติ 1 วินาที และกระแสเกิน 1 วินาที ;  
3 s, Time/div = 500 ms/div แสดงดังภาพที่ 4-12
- ครั้งที่ 4 กำหนดระยะเวลากระแสปกติ 2 วินาที และกระแสเกิน 3 วินาที ;  
7 s, Time/div = 1 s/div แสดงดังภาพที่ 4-13



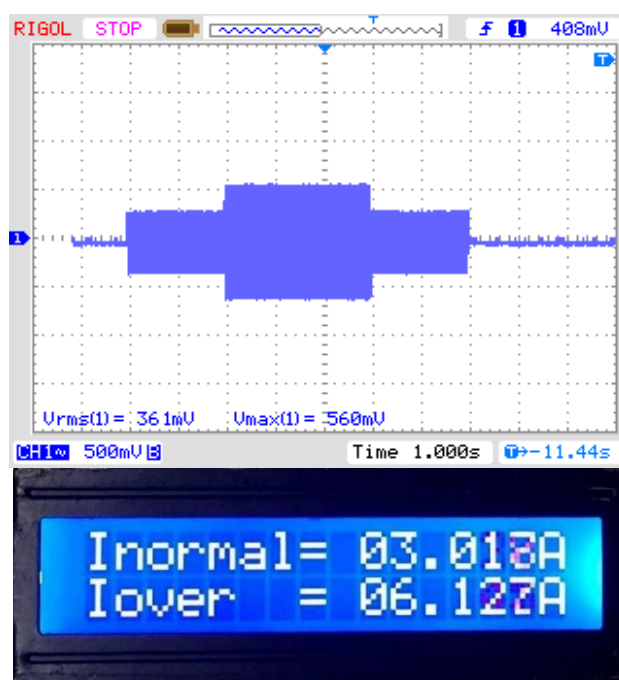
ภาพที่ 4-10 การทดสอบครั้งที่ 1 เลือกกำหนดจำนวนลูกคลื่น  
กระแสปกติ 3 ลูกคลื่น และกระแสเกิน 2 ลูกคลื่น



ภาพที่ 4-11 การทดสอบครั้งที่ 2 เลือกกำหนดจำนวนลูกคลื่น  
กระแสปกติ 5 ลูกคลื่น และกระแสเกิน 10 ลูกคลื่น



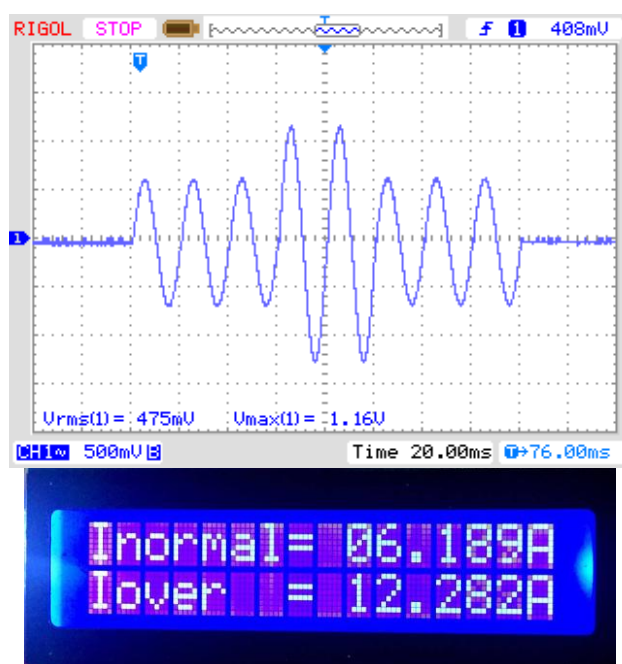
ภาพที่ 4-12 การทดสอบครั้งที่ 3 เลือกกำหนดระยะเวลา  
กระแสปกติ 1 วินาที และกระแสเกิน 1 วินาที



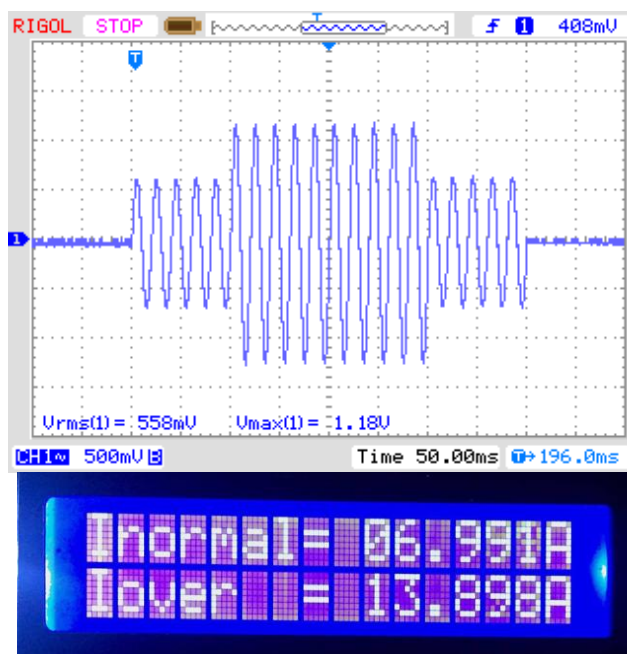
ภาพที่ 4-13 การทดสอบครั้งที่ 4 เลือกกำหนดระยะเวลา  
กระแสปกติ 2 วินาที และกระแสเกิน 3 วินาที

#### 4.3.2 การทดสอบทำในเงื่อนไขดังนี้

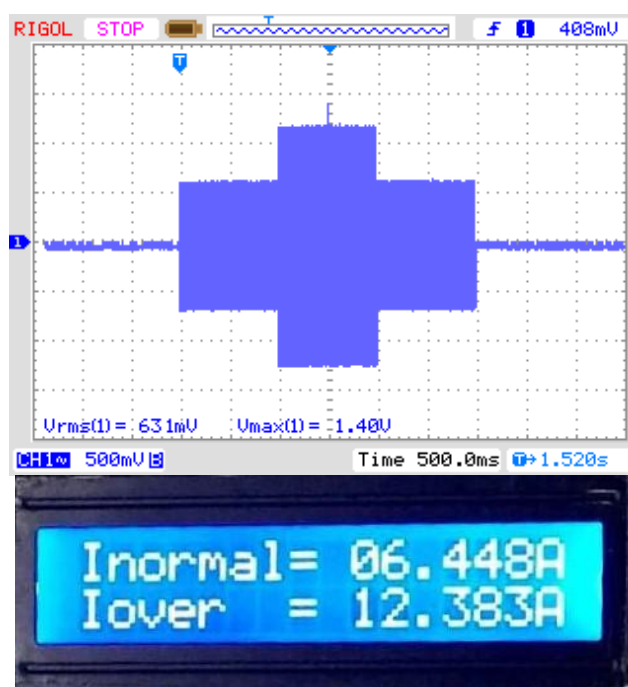
- แหล่งจ่ายไฟกระแสสลับขนาด 40 V<sub>RMS</sub>
- โหลดความต้านทานขนาด 5 Ω, 1,200 Watts 2 ชุด
- แสดงแรงดันเอาต์พุตด้วย Ch.1 (0.5 V/div)
- ทำการทดสอบ 4 ครั้ง โดยกำหนดจำนวนลูกคลื่นหรือระยะเวลาของเอาต์พุตกระแส การกำหนดระยะเวลาการกระตุ้น คือ
  - ครั้งที่ 5 กำหนดจำนวนลูกคลื่นกระแสปกติ 3 ลูกคลื่น และกระแสเกิน 2 ลูกคลื่น ;  
160 ms , Time/div = 20 ms/div แสดงดังภาพที่ 4-14
  - ครั้งที่ 6 กำหนดจำนวนลูกคลื่นกระแสปกติ 5 ลูกคลื่น และกระแสเกิน 10 ลูกคลื่น;  
400 ms, Time/div = 50 ms/div แสดงดังภาพที่ 4-15
  - ครั้งที่ 7 กำหนดระยะเวลากระแสปกติ 1 วินาที และกระแสเกิน 1 วินาที ;  
3 s, Time/div = 500 ms/div แสดงดังภาพที่ 4-16
  - ครั้งที่ 8 กำหนดระยะเวลากระแสปกติ 2 วินาที และกระแสเกิน 3 วินาที ;  
7 s, Time/div = 1 s/div แสดงดังภาพที่ 4-17



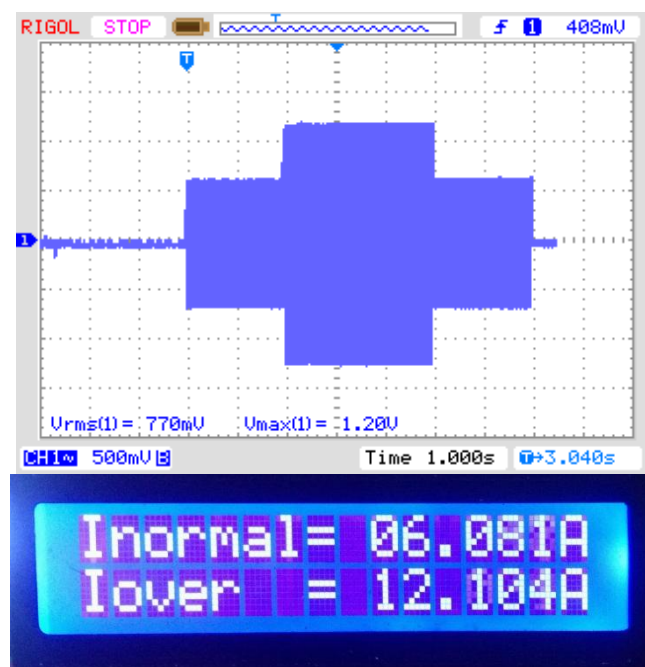
ภาพที่ 4-14 การทดสอบครั้งที่ 5 เลือกกำหนดจำนวนลูกคลื่น  
กระแสปกติ 3 ลูกคลื่น และกระแสเกิน 2 ลูกคลื่น



ภาพที่ 4-15 การทดสอบครั้งที่ 6 เลือกกำหนดจำนวนลูกคลื่น  
กระแสปกติ 5 ลูกคลื่น และกระแสเกิน 10 ลูกคลื่น



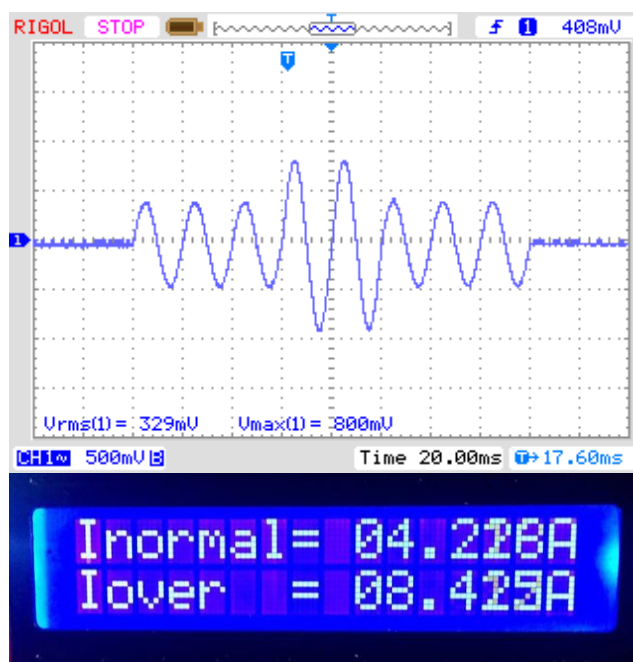
ภาพที่ 4-16 การทดสอบครั้งที่ 7 เลือกกำหนดระยะเวลา  
กระแสปกติ 1 วินาที และกระแสเกิน 1 วินาที



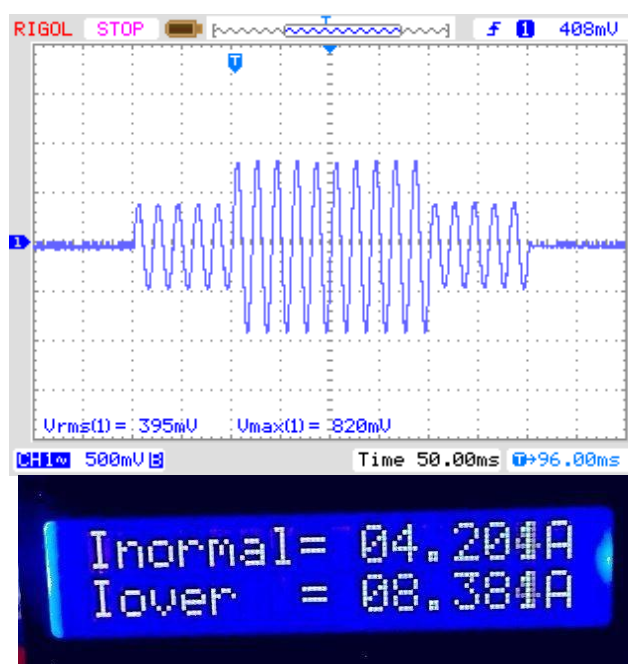
ภาพที่ 4-17 การทดสอบครั้งที่ 8 เลือกกำหนดระยะเวลา  
กระแสปกติ 2 วินาที และกระแสเกิน 3 วินาที

### 4.3.3 การทดสอบทำในเงื่อนไขดังนี้

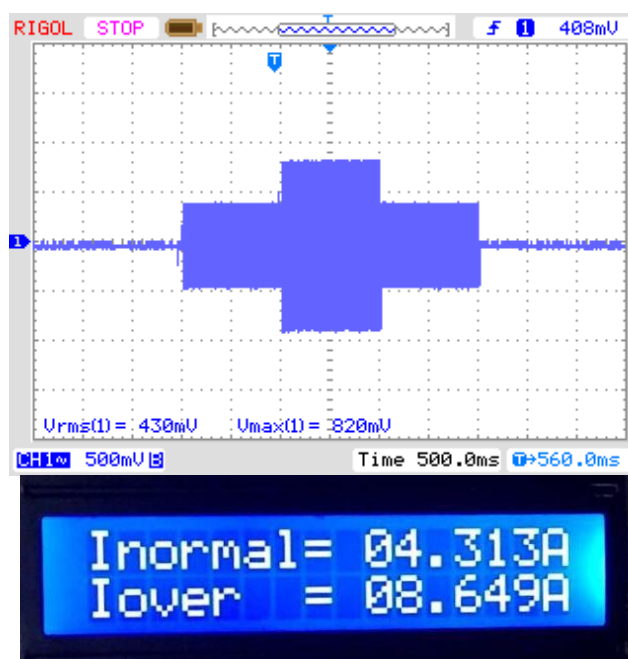
- แหล่งจ่ายไฟกระแสสลับขนาด 50 V<sub>RMS</sub>
- โหลดความต้านทานขนาด 10  $\Omega$ , 2400 Watts 2 ชุด
- แสดงแรงดันเอาต์พุตด้วย Ch.1 (0.5 V/div)
- ทำการทดสอบ 4 ครั้ง โดยกำหนดจำนวนลูกคลื่นหรือระยะเวลาของเอาต์พุตกระแส การกำหนดระยะเวลาการกระตุ้น คือ
  - ครั้งที่ 9 กำหนดจำนวนลูกคลื่นกระแสปกติ 3 ลูกคลื่น และกระแสเกิน 2 ลูกคลื่น ;  
160 ms , Time/div = 20 ms/div แสดงดังภาพที่ 4-18
  - ครั้งที่ 10 กำหนดจำนวนลูกคลื่นกระแสปกติ 5 ลูกคลื่นและกระแสเกิน 10 ลูกคลื่น  
400 ms, Time/div = 50 ms/div แสดงดังภาพที่ 4-19
  - ครั้งที่ 11 กำหนดระยะเวลากระแสปกติ 1 วินาที และกระแสเกิน 1 วินาที ;  
3 s, Time/div = 500 ms/div แสดงดังภาพที่ 4-20
  - ครั้งที่ 12 กำหนดระยะเวลากระแสปกติ 2 วินาที และกระแสเกิน 3 วินาที ;  
7 s, Time/div = 1 s/div แสดงดังภาพที่ 4-21



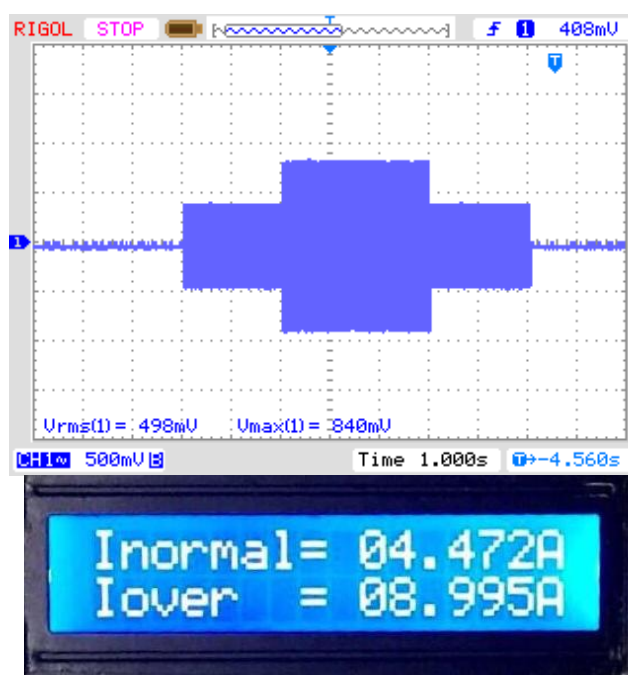
ภาพที่ 4-18 การทดสอบครั้งที่ 9 เลือกกำหนดจำนวนลูกคลื่น  
กระแสปกติ 3 ลูกคลื่น และกระแสเกิน 2 ลูกคลื่น



ภาพที่ 4-19 การทดสอบครั้งที่ 10 เลือกกำหนดจำนวนลูกคลื่น  
กระแสปกติ 5 ลูกคลื่น และกระแสเกิน 10 ลูกคลื่น



ภาพที่ 4-20 การทดสอบครั้งที่ 11 เลือกกำหนดระยะเวลา  
กระแสปกติ 1 วินาที และกระแสเกิน 1 วินาที

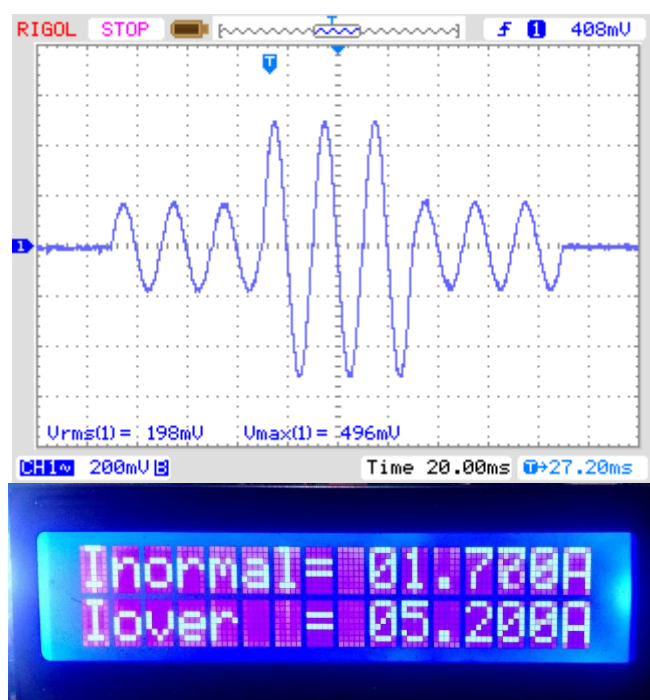


ภาพที่ 4-21 การทดสอบครั้งที่ 12 เลือกกำหนดระยะเวลา  
กระแสปกติ 2 วินาที และกระแสเกิน 3 วินาที

#### 4.4 ผลการทดสอบการทำงานของเครื่องกำเนิดกระแสเกินชั่วคราว เมื่อนำไปใช้กับ Surge Counter

ผลการทดสอบเครื่องกำเนิดกระแสเกินชั่วคราวเมื่อนำไปใช้กับ Surge Counter เกิดการนับขึ้นของตัวเลขที่ต่อเมื่อกระแสเกินมีค่าตั้งแต่ 5 A โดยเลือกกำหนดจำนวนลูกคลื่นให้เกิดกระแสปกติ 3 ลูก และกระแสเกิน 3 ลูก ใช้แรงดันอินพุตเป็นไฟกระแสสลับ 40 V<sub>RMS</sub> ต่อกับโหลดตัวต้านทานของกระแสปกติ 20 และโหลดตัวต้านทานของกระแสเกิน 10 แสดงโดย Ch.1 (0.2 V/div) และ Time/div = 20 ms/div แสดงดังภาพที่ 4-22 และผลการนับของ Surge Counter แสดงดังภาพที่ 4-23





ภาพที่ 4-22 เอาต์พุตของเครื่องกำเนิดกระแสเกินชั่วคราว เมื่อเลือกกำหนดจำนวนลูกคลื่น  
กระแสปกติ 3 ลูกคลื่น และกระแสเกิน 3 ลูกคลื่น



ภาพที่ 4-23 ผลการนับของ Surge Counter

## บทที่ 5

### สรุปผล ปัญหาและข้อเสนอแนะ

#### 5.1 สรุปผล

ปริญญานิพนธ์เล่มนี้ได้ศึกษา ออกแบบ และสร้างเครื่องกำเนิดกระแสเกิน ขนาดไม่เกิน 20 A<sub>RMS</sub> โดยใช้อุปกรณ์เชื่อมต่อทางแสงในการตรวจจับจุดตัดสัญญาณแรงดันอินพุต ใช้บอร์ดไมโครคอนโทรลเลอร์ในการรับค่าจากผู้ใช้งานทางจอแอลซีดี และคีย์แพด ใช้ไทรแอกเป็นสวิตช์เปิด – ปิดวงจร และใช้โหลดตัวต้านทาน เพื่อให้ได้รูปสัญญาณกระแสตามที่ต้องการ

จากการจำลองการทำงานวงจรของเครื่องกำเนิดกระแสเกินชั่วคราวด้วยโปรแกรม Proteus และ LTspice แสดงให้เห็นว่าสามารถสร้างกระแสปกติ และกระแสเกินตามจำนวนลูกคลื่นหรือระยะเวลาที่ผู้ใช้งานกำหนดได้ โดยมีขนาดขึ้นอยู่กับแหล่งจ่ายแรงดันไฟฟ้า และโหลดตัวต้านทานที่ต่อให้กับวงจร ซึ่งหลังจากการจำลองการทำงานแล้วก็นำผลที่ได้มาใช้เป็นแนวทางในการสร้างเครื่องกำเนิดกระแสเกินชั่วคราวจริง

จากการทดสอบเครื่องกำเนิดกระแสเกินชั่วคราวที่ได้สร้างขึ้นพบว่า เครื่องกำเนิดกระแสเกินชั่วคราวสามารถสร้างกระแสปกติและกระแสเกินได้ที่ขนาดแรงดันอินพุต 7.07 – 100 V<sub>RMS</sub> โดยมีพิกัดกระแสที่ตัวไทรแอกคือ 25 A<sub>RMS</sub> และสามารถสร้างตามจำนวนลูกคลื่นหรือระยะเวลาที่ผู้ใช้งานกำหนดได้

#### 5.2 ปัญหา แนวทางแก้ไข และพัฒนา

ปัญหาที่พบจากการทดสอบวงจรคือ ค่ากระแสที่วัดได้จากออกออสซิลโลสโคปและจอแอลซีดี มีค่าต่ำกว่าค่าที่ได้ทางทฤษฎีเล็กน้อย ซึ่งอาจเกิดจากแรงดันและกระแสที่ไปเลี้ยงวงจรไม่พอหรือไม่เสถียร ฉะนั้นจึงจำเป็นต้องปรับแต่งวงจรที่จ่ายแรงดันและกระแสไฟเลี้ยงให้กับอุปกรณ์ต่างๆภายในเครื่อง เพื่อให้สามารถจ่ายไฟเลี้ยงได้อย่างเสถียรและเพียงพอต่อความต้องการของวงจร

#### 5.3 ข้อเสนอแนะ

5.3.1 เปลี่ยนชนิดหรือรุ่นของบอร์ดไมโครคอนโทรลเลอร์ ให้มีจำนวนขาเชื่อมต่อเพียงพอภายในบอร์ดเดียว

5.3.2 เพิ่มพิกัดกระแสที่เครื่องกำเนิดกระแสเกินชั่วคราวสามารถสร้างได้

### เอกสารอ้างอิง

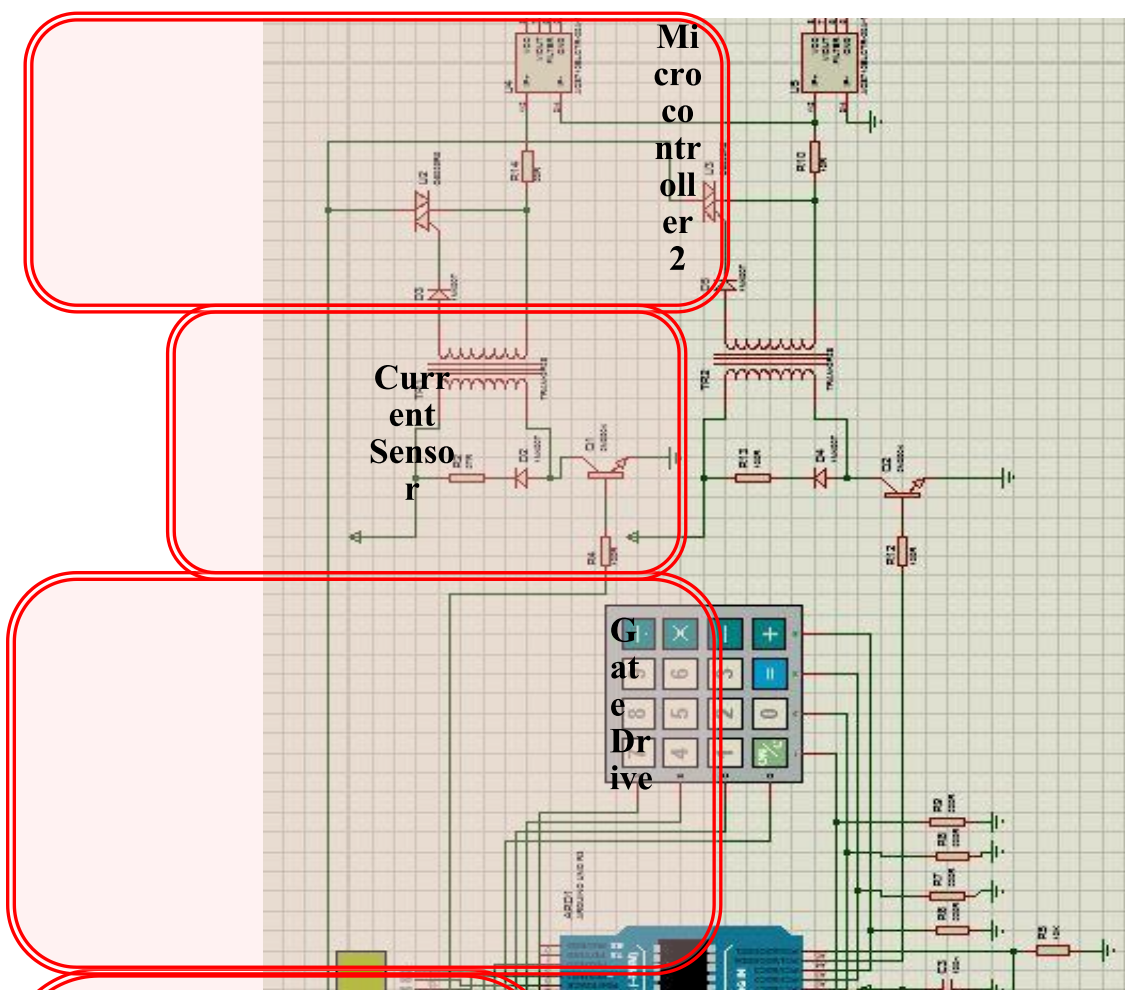
1. B\_E\_N. (2556). *What is an Arduino?*. วันที่ค้นข้อมูล 29 ธันวาคม 2559. จาก SparkFun  
เว็บไซต์ : <https://learn.sparkfun.com/tutorials/what-is-an-arduino>
2. บริษัท วินัส ชัพพลาย จำกัด. (2549). บทความ *Arduino* คืออะไร. วันที่ค้นข้อมูล 17  
พฤศจิกายน 2559. จาก บริษัท วินัส ชัพพลาย จำกัด เว็บไซต์ : <http://www.thaieasyelec.com/article-wiki/basic-electronics/บทความ-arduino-คืออะไร-เริ่มต้นใช้งาน-arduino.html>
3. บริษัท วินัส ชัพพลาย จำกัด. (2549). บทความ การใช้งาน *Character LCD Display* กับ  
*Arduino*. วันที่ค้นข้อมูล 17 พฤศจิกายน 2559. จาก บริษัท วินัส ชัพพลาย จำกัด เว็บไซต์ :  
<http://www.thaieasyelec.com/article-wiki/review-product-article/การใช้งาน-character-lcd-display-กับ-arduino-ตอนที่1-รูปแบบการเชื่อมต่อแบบ-parallel.html>
4. ห้องปฏิบัติการระบบสมองกลฝังตัว ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ คณะ  
วิศวกรรมศาสตร์ มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ. (2556). *การใช้งานเป็น  
ปุ่มกด (Keypad) แบบ 4x4 ปุ่ม*. วันที่ค้นข้อมูล 17 พฤศจิกายน 2559. จาก ห้องปฏิบัติการระบบ  
สมองกลฝังตัว ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ มหาวิทยาลัย  
เทคโนโลยีพระจอมเกล้าพระนครเหนือ เว็บไซต์ : <http://cpre.kmutnb.ac.th/esl/learning/index.php?article=4x4-keypad>
5. คุณสรายุทธ แยมประยูร. (2555). การทดลองที่ 6 การใช้งาน *Matrix Switch (Keypad)*. วันที่ค้น  
ข้อมูล 17 พฤศจิกายน 2559. จาก บริษัท อิเล็กทรอนิกส์โปร ดีไซน์ จำกัด เว็บไซต์ :  
<http://www.eopro.co.th/index.php?mo=3&art=42039092>
6. Praveen. (2557). *Interfacing hex keypad to Arduino*. วันที่ค้นข้อมูล 17 พฤศจิกายน 2559. จาก  
CircuitsToday Electronic Solutions เว็บไซต์ : <http://www.circuitstoday.com/interfacing-hex-keypad-to-arduino>
7. Esc Rc servo. (2558). *แนะนำอุปกรณ์ อุปกรณ์เชื่อมต่อทางแสง (Optocouplers)*. วันที่ค้น  
ข้อมูล 17 พฤศจิกายน 2559. จาก Esc Rc servo เว็บไซต์ : <http://brushlesservocontrolrccar.blogspot.com/2013/08/optocouplers.html>
8. arduino-info.wikispaces.com. (2558). *PC817-PinOut*. วันที่ค้นข้อมูล 17 พฤศจิกายน 2559.  
เว็บไซต์ : <https://arduino-info.wikispaces.com/file/view/PC817-PinOut.jpg/329170116/PC817-PinOut.jpg>

## เอกสารอ้างอิง (ต่อ)

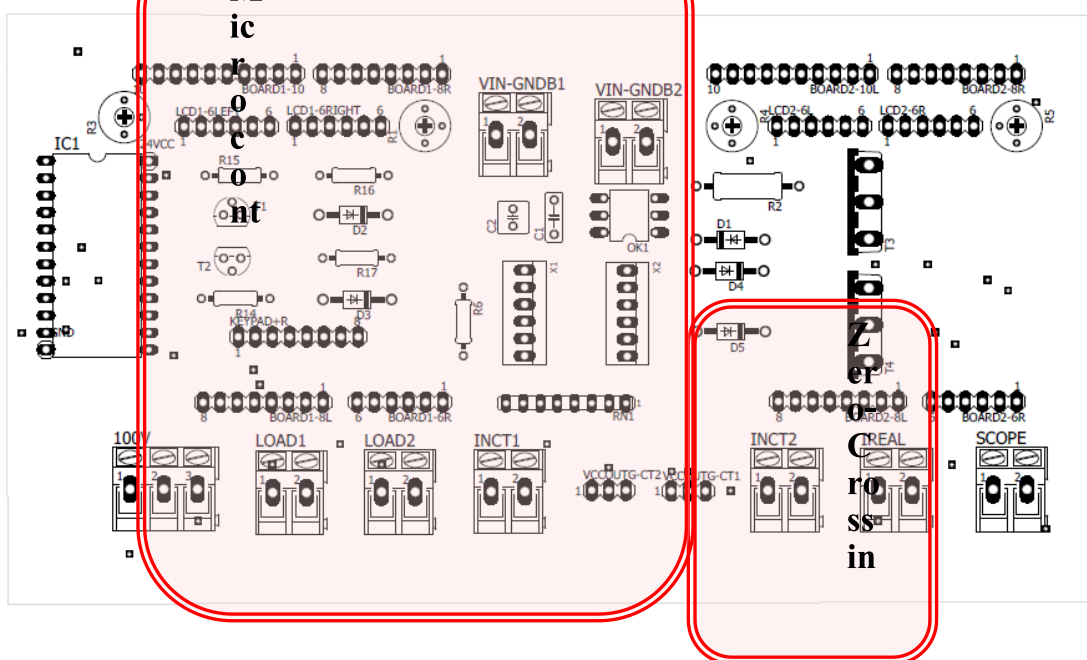
9. 49210666. (2551). ผลการทดสอบหม้อแปลง. วันที่ค้นข้อมูล 17 พฤศจิกายน 2559. จาก oknation เว็บไซต์ : <http://www.oknation.net/blog/49210666>
10. WWW.STABLE.CO.TH. (2558). หม้อแปลงไฟฟ้า (Transformer). วันที่ค้นข้อมูล 17 พฤศจิกายน 2559. จาก บริษัท สเตเบิล อิเล็กทรอนิกส์ จำกัด เว็บไซต์ : <http://www.Stable.co.th/index.php?lay=show&ac=article&Id=539185370&Ntype=17>
11. <http://www.pspstech.co.th>. (2558). หลักการทำงานของหม้อแปลงไฟฟ้า (Transformer) วันที่ค้นข้อมูล 17 พฤศจิกายน 2559. จาก บริษัท พีเอสพี เทคโนโลยี จำกัด เว็บไซต์ : <http://www.pspstech.co.th/หลักการทำงานของหม้อแปลงไฟฟ้าtransformer-16777.page>
12. ฟิสิกส์ราชมงคล. (2552). โครงสร้างของไดรแอก. วันที่ค้นข้อมูล 17 พฤศจิกายน 2559. เว็บไซต์ : <http://www.neutron.rmutphysics.com/physicsboard/forum/index.php?action=dlattach;topic=898.0;attach=4078;image>
13. Wikipedia. (2558). Zero crossing. วันที่ค้นข้อมูล 17 พฤศจิกายน 2559. เว็บไซต์ : [https://en.wikipedia.org/wiki/Zero\\_crossing](https://en.wikipedia.org/wiki/Zero_crossing)
14. Mehrdad Kamelzadeh & Stevenvh. (2555). Digital Dimmer with Microcontroller. วันที่ค้นข้อมูล 17 พฤศจิกายน 2559. เว็บไซต์ : <http://electronics.stackexchange.com/questions/35685/digital-dimmer-with-microcontroller>
15. Vikas Sood. (2558). How to correctly simulate Zero Cross Detector Circuit?. วันที่ค้นข้อมูล 17 พฤศจิกายน 2559. เว็บไซต์ : <http://electronics.stackexchange.com/questions/160883/how-to-correctly-simulate-zero-crossdetector-circuit>
16. Xsumirx. (2557). Zero Cross Detector (using Optocoupler 4N35). วันที่ค้นข้อมูล 17 พฤศจิกายน 2559. เว็บไซต์ : <https://elexidea.wordpress.com/2014/10/01/zero-cross-detector-using-optocoupler-4n35/>

ภาคผนวก ก

การประกอบวงจร

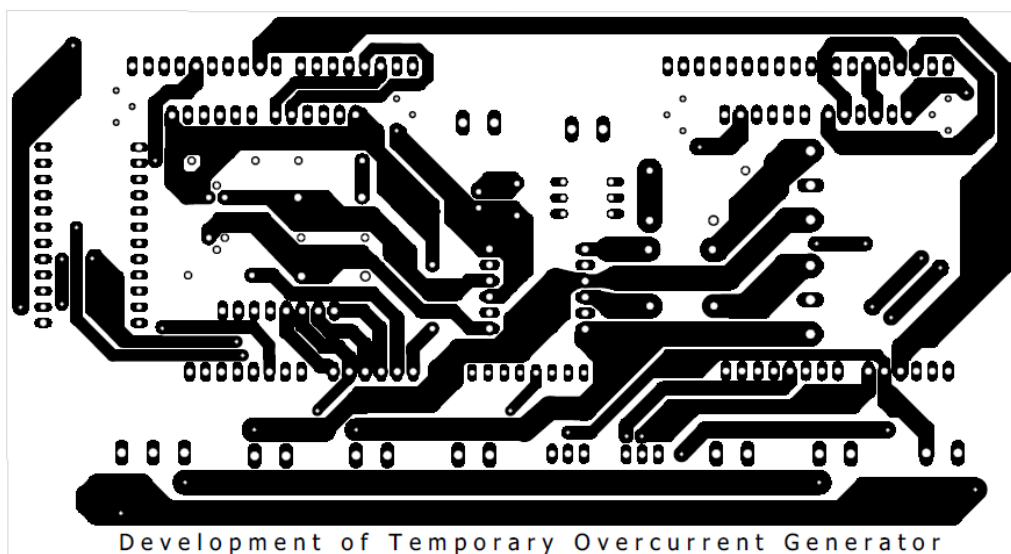


ภาพที่ ก-1 ไดอะแกรมรวมของเครื่องกำเนิดกระแสเกินชั่วคราว

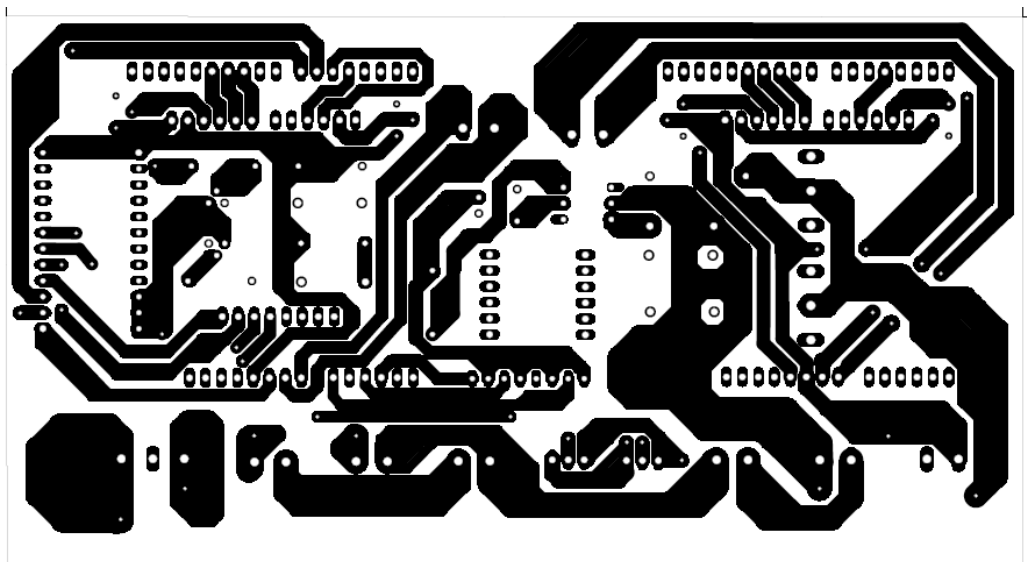


ภาพที่ ก-2 ตำแหน่งอุปกรณ์บนลายทองแดงของเครื่องกำเนิดกระแสเกินชั่วคราว

ในโปรแกรมออกแบบลายวงจร

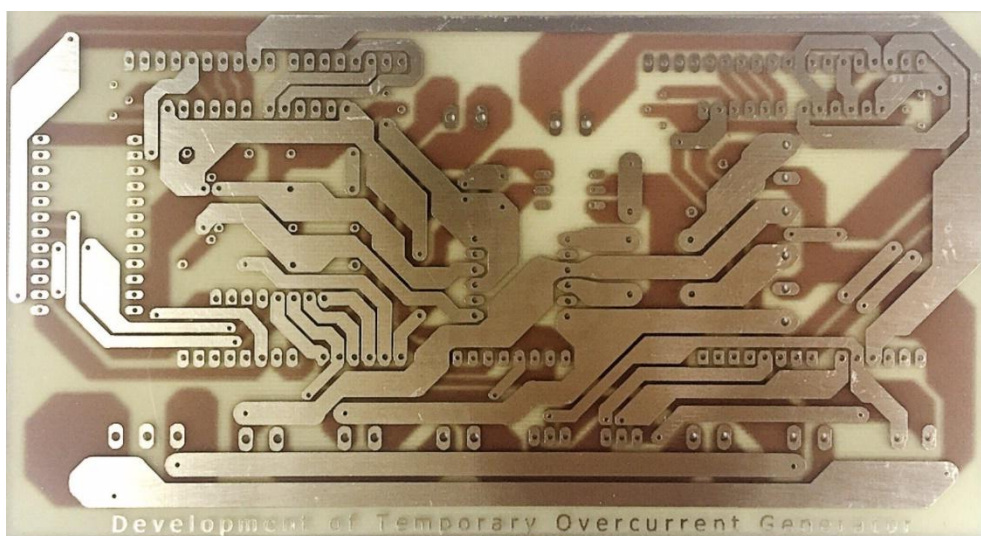


ภาพที่ ก-3 ลายทองแดงด้านบนของเครื่องกำเนิดกระแสเกินชั่วคราว  
ในโปรแกรมออกแบบลายวงจร

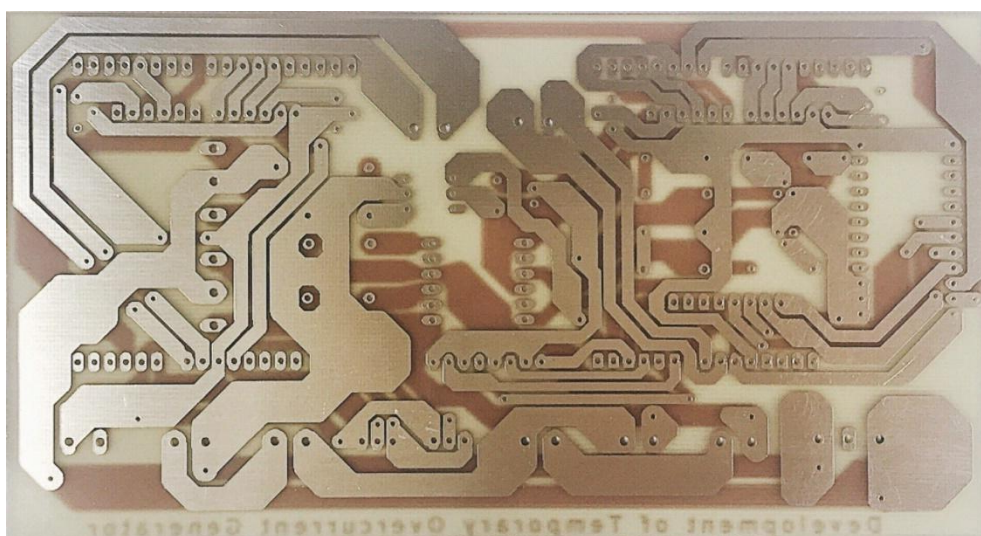


ภาพที่ ก-4 ลายทองแดงด้านล่างของเครื่องกำเนิดกระแสเกินชั่วคราว  
ในโปรแกรมออกแบบลายวงจร



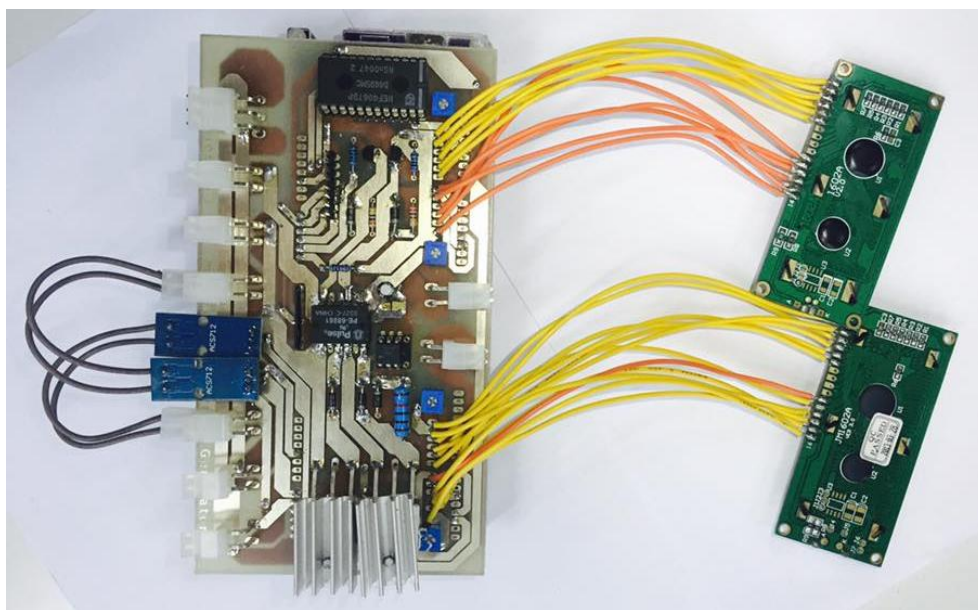


ภาพที่ ก-5 ลายทองแดงด้านบนของเครื่องกำเนิดกระแสเกินชั่วคราว

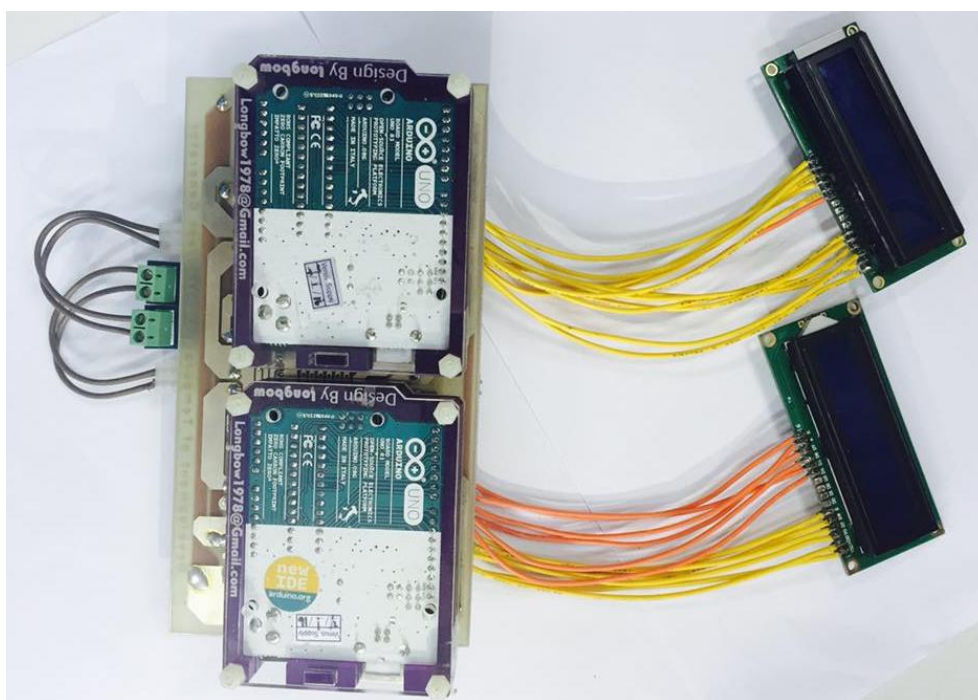


ภาพที่ ก-6 ลายทองแดงด้านล่างของเครื่องกำเนิดกระแสเกินชั่วคราว





ภาพที่ ก-7 แผ่นวงจรด้านบนของเครื่องกำเนิดกระแสเกินชั่วคราวหลังบัดกรี



ภาพที่ ก-8 แผ่นวงจรด้านล่างของเครื่องกำเนิดกระแสเกินชั่วคราวหลังบัดกรี



ภาพที่ ก-9 เครื่องกำเนิดกระแสเกินชั่วคราวหลังประกอบอุปกรณ์ดังกล่าว

ภาคผนวก ข

โปรแกรมในไมโครคอนโทรลเลอร์

## 1. โปรแกรมในไมโครคอนโทรลเลอร์ 1

```
#define F_CPU 16000000UL //MPU Clock Freq.

#include <avr/io.h>

#include <util/delay.h> //For _Delay_us() and _Delay_ms() function

int A = 0;

int B = 0;

int C = 0;

int z = 13;

int num = 0, num2 = 0;

int e, f, g, h,m,n,t=0;

int R = 0;

int D,pulse;

#define LCD_DPRT PORTD //PortD.5-D.12

#define LCD_DDDR DDRD //in Arduino Pin5-2

#define LCD_DPIN PIND //LCD Data Pin11-14

#define LCD_BPRT PORTB //PortB.4-B.2

#define LCD_BDDR DDRB //

#define LCD_BPIN PINB //

#define LCD_RS 4 //PB.4 Arduino Pin 12

#define LCD_RW 3 //PB.3 Arduino Pin 11

#define LCD_EN 2 //PB.2 Arduino Pin 10

//LCD module information

//LCD Instructions

#define LCD_Clear 0b00000001 //Replace all characters with ASCII 'Space'

#define LCD_Home 0b00000010 //Return cursor to first position on first line

#define LCD_EntryMode 0b00000110 //Shift cursor from left to right on read/write

#define LCD_DisplayOff 0b00001000 //Turn display

#define LCD_DisplayOn 0b00001100 //Display on, cursor off, don't blink character

#define LCD_FunctionReset 0b00110010 //Reset the LCD

#define LCD_FunctionSet4bit 0b00101000 //4-bit data, 2-line display, 5x7 font
```

```

#define LCD_SetCursor 0b10000000 //Set cursor position
#define LCD_CGRAM_ADDRESS 0b01000000 //Set CGRAM_ADDRESS
#define LCD_DDRAM_ADDRESS 0b10000000 //Set DDRAM_ADDRESS

void LCD_CMNDWRT42(unsigned char cmnd) //Subprogram Write Command 4-bit Data 2 times
{
    unsigned char A2 = 0; //Define A2
    A2 = cmnd; //Back up Data(cmnd)
    cmnd &= 0b11110000; //Mask-Low Nibble Clear
    LCD_DPRT = cmnd >> 2; //Shift Right 2 bit and Send
    //(High Nibble) to PORTD
    LCD_BPRT &= 0b11100100; //Set LCD_RS=0 for Select Instruction register
    //and Set LCD_RW=0 for write
    LCD_BPRT |= 0b00000100; //Set LCD_EN=1 for Starts data read/write
    _delay_us(1); //wait about 1us.
    LCD_BPRT &= 0b11111000; //Set LCD_EN=0 for Ends data read/write
    cmnd = A2 << 4; //Load back up Data
    cmnd &= 0b11110000; //Mask-Low Nibble Clear
    LCD_DPRT = cmnd >> 2; //Shift Right 2 bit and Send
    //(High Nibble) to PORTD
    LCD_BPRT |= 0b00000100; //Set LCD_EN=1 for Starts data read/write
    _delay_us(1); //wait about 1us.
    LCD_BPRT &= 0b11111000; //Set LCD_EN=0 for Ends data read/write
    _delay_us(100); //wait about 100us.
}

void LCD_CLS(void) //Subprogram Clear LCD
{
    unsigned char x = 0x01; //Define x
    LCD_CMNDWRT42(x); //Write Command for Clear
    _delay_ms(2); //wait about 2us.
}

```

```

void LCD_CMNDWRT41(unsigned char cmnd) //Subprogram Write Command 4-bit Data 1 time
{
    unsigned char A1 = 0; //Define A1
    A1 = cmnd; //Back up Data(cmnd)
    cmnd &= 0b11110000; //Mask-Low Nibble Clear
    LCD_DPRT = cmnd >> 2; //Shift Right 2 bit and Send
    //(High Nibble) to PORTD
    LCD_BPRT &= 0b11100100; //Set LCD_RS=0 for Select Instruction register
    //and Set LCD_RW=0 for write
    LCD_BPRT |= 0b00000100; //Set LCD_EN=1 for Starts data read/write
    _delay_us(1); //wait about 1us.
    LCD_BPRT &= 0b11111000; //Set LCD_EN=0 for Ends data read/write
}

void Init_LCD4D(void) //Subprogram Initialize LCD 4-bit data Mode
{
    _delay_ms(50); //wait about 50ms.
    LCD_CMNDWRT41(0x30); //special case of Function Set
    _delay_ms(5); //wait about 5ms.
    LCD_CMNDWRT41(0x30); //special case of Function Set
    _delay_us(100); //wait about 100us.
    LCD_CMNDWRT41(0x30); //special case of Function Set
    _delay_us(100); //wait about 100us.
    LCD_CMNDWRT41(0x20); //Initial Function Set to Change Interface
    _delay_us(100); //wait about 100us.
    LCD_CMNDWRT42(0x28); //Function Set N=1,F=0 2line and 5*8 Dot
    LCD_CMNDWRT42(0x0C); //Display ON D=1,C=1 Display on and Cursor On
    LCD_CLS(); //Clear LCD
    LCD_CMNDWRT42(0x06); //Entry Mode Set 1/D=1, S=0
}

void LCD_DATWRT42(unsigned char dat) //Subprogram Write Data 4-bit Data 2 times
{

```

```

unsigned char A3 = 0; //Define A3

A3 = dat; //Back up Data(dat)

dat &= 0b11110000; //Mask-Low Nibble Clear

LCD_DPRT = dat >> 2; //Shift Right 2 bit and Send
//(High Nibble) to PORTD

LCD_BPRT &= 0b11110100; //Set LCD_RW=0 for write
LCD_BPRT |= 0b00010100; //Set LCD_RS=1 for Select Data register
//and Set LCD_EN=1 for Starts data read/write

_delay_us(1); //wait about 1us.

LCD_BPRT &= 0b11111000; //Set LCD_EN=0 for Ends data read/write

dat = A3 << 4; //Load back up Data

dat &= 0b11110000; //Mask-Low Nibble Clear

LCD_DPRT = dat >> 2; //Shift Right 2 bit and Send
//(High Nibble) to PORTD

LCD_BPRT |= 0b00000100; //Set LCD_EN=1 for Starts data read/write

_delay_us(1); //wait about 1us.

LCD_BPRT &= 0b11111000; //Set LCD_EN=0 for Ends data read/write

_delay_us(100); //wait about 100us.
}

void LCD_gotoxy(uint8_t col, uint8_t row) //Subprogram Set Position Cursor
{
    unsigned int s[] = {0x80, 0xC0}; //Define s[]

    LCD_CMNDWRT42(s[row - 1] + (col - 1)); //Write Command for Set Cursor

    _delay_us(1000); //wait about 1000us.
}

void LCD_Print(char*str) //Subprogram Print String
{
    unsigned char y = 0; //Define y

    while (str[y] != 0) //ทำงานเมื่อมีข้อมูลใน str[]
    {

```

```

    LCD_DATWRT42(str[y]); //Write Data from str[]
    y++;      //เลื่อน Pointer
}
}

void LCD_Printnum(int num) //Subprogram Print number
{
    int y = num; //Define y
    int C, D;
    D = y % 10;
    D = D + 0x30;
    C = y / 10;
    C = C + 0x30;

    LCD_DATWRT42(C);
    LCD_DATWRT42(D);
}

void LCD_Printnum2(int num2)
{
    int y = num2;
    y = y + 0x30;
    LCD_DATWRT42(y);
}

void LCDhomepage(void)
{
    while (A == 0) {
        LCD_gotoxy(2, 1); //Set cursor to low 2,col 1
        LCD_Print("Press ENTER");//Send String to Screen
        LCD_gotoxy(2, 2); //Set cursor to low 2,col 2
        LCD_Print("to continue");//Send String to Screen
        R = button();
    }
}

```



```

if (R == 15) {
    A = A + 1;
    pulse = 0;
    D = 0;
    LCD_CLS();
}
}
}

void MenuValue(void){
    while (A == 1) {
        LCD_gotoxy(1, 1); //Set cursor to low 2,col 1
        LCD_Print("A: CYCLE");//Send String to Screen
        LCD_gotoxy(1, 2); //Set cursor to low 2,col 2
        LCD_Print("B: TIME");//Send String to Screen
        R = button();
        if ((R == 10)&&(B==0)) {
            A = 2;
            LCD_CLS();
            C = 0;
        }
        if ((R == 10)&&(B==1)) {
            A = 3;
            LCD_CLS();
            C = 0;
        }
        if ((R == 11)&&(C==0)) {
            A = 6 ;
            LCD_CLS();
            B = 0;
        }
        if ((R == 11)&&(C==1)) {

```

```

    A = 7;

    LCD_CLS();

    B = 0;

}

}

}

void Entervalue(void)
{
    while (A == 2) {
        num = 0;

        z = 11;

        LCD_gotoxy(1, 1); //Set cursor to low 2,col 2
        LCD_Print("CYCLE "); //Send String to Screen
        LCD_gotoxy(1, 2); //Set cursor to low 2,col 2
        LCD_Print("NORMAL I= "); //Send String to Screen
        R = buttom();

        LCD_gotoxy(z, 2); //Set cursor to low 1,col 1
        LCD_Printnum2(R);

        if (R == 12) {
            LCD_CLS();

            A = 1;

            break;
        }

        z = z + 1;

        e = R;

        R = buttom();

        if (R == 12) {
            LCD_CLS();

            A = 1;

            break;
        }
    }
}

```

```

LCD_gotoxy(z, 2); //Set cursor to low 1,col 1
LCD_Printnum2(R);
z = z + 1;
f = R;
A = A + 1;
}
while(A == 3){
z = 11;
LCD_gotoxy(1, 1); //Set cursor to low 2,col 2
LCD_Print("CYCLE "); //Send String to Screen
LCD_gotoxy(1, 2); //Set cursor to low 2,col 2
LCD_Print("NORMAL I= "); //Send String to Screen
LCD_gotoxy(z, 2); //Set cursor to low 1,col 1
LCD_Printnum2(e);
z = z + 1;
LCD_gotoxy(z, 2); //Set cursor to low 1,col 1
LCD_Printnum2(f);
R = button();
if ((R == 13) && (B != 1)) {
    LCD_CLS();
    e = e * 10;
    num = e + f;
    A = 4;
}
if ((R == 13) && (B == 1)) {
    LCD_CLS();
    e = e * 10;
    num = e + f;
    A = 5;
}
if (R == 14) {

```

```

    LCD_CLS();

    A = 2;

    B = 0;

}

if (R == 12) {

    LCD_CLS();

    A = 1;

    B = 1;

}

}

}

void Entervalue2(void)
{
    while (A == 4) {
        num2 = 0;

        z = 9;

        LCD_gotoxy(1, 1); //Set cursor to low 2,col 2
        LCD_Print("CYCLE "); //Send String to Screen
        LCD_gotoxy(1, 2); //Set cursor to low 2,col 2
        LCD_Print("OVER I= "); //Send String to Screen
        R = button();

        if (R == 12) {

            LCD_CLS();

            A = 3;

            e=e/10;

            break;

        }

        LCD_gotoxy(z, 2); //Set cursor to low 1,col 1
        LCD_Printnum2(R);

        z = z + 1;
    }
}

```

```

m = R;
R = button();
if (R == 12) {
    LCD_CLS();
    A = 3;
    e=e/10;
    break;
}
LCD_gotoxy(z, 2); //Set cursor to low 1,col 1
LCD_Printnum2(R);
z = z + 1;
n = R;
A = A+1;
}
while(A == 5){
    z = 9;
    LCD_gotoxy(1, 1); //Set cursor to low 2,col 2
    LCD_Print("CYCLE "); //Send String to Screen
    LCD_gotoxy(1, 2); //Set cursor to low 2,col 2
    LCD_Print("OVER I= "); //Send String to Screen
    LCD_gotoxy(z, 2); //Set cursor to low 1,col 1
    LCD_Printnum2(m);
    z = z + 1;
    LCD_gotoxy(z, 2); //Set cursor to low 1,col 1
    LCD_Printnum2(n);
    R = button();
    if (R == 12) {
        LCD_CLS();
        A = 3;
        e=e/10;
        B = 1;

```

```

    }

    if (R == 14) {

        LCD_CLS();

        A = 4;

    }

    if (R == 15) {

        LCD_CLS();

        m = m * 10;

        num2 = m + n;

        if((num2 == 0)|| (num == 0)){

            A = 20;

        }else{

            A = 10;

        }

    }

}

if (R == 13) {

    LCD_CLS();

    A = 5;

}

}

}

void Entervalue3(void)
{
    while (A == 6) {

        num = 0;

        z = 11;

        LCD_gotoxy(1, 1); //Set cursor to low 2,col 2

        LCD_Print("TIME "); //Send String to Screen

        LCD_gotoxy(1, 2); //Set cursor to low 2,col 2

        LCD_Print("NORMAL I= "); //Send String to Screen

```

```

R = button();
if (R == 12) {
    LCD_CLS();
    A = 1;
    break;
}
LCD_gotoxy(z, 2); //Set cursor to low 1,col 1
LCD_Printnum2(R);
z = z + 1;
e = R;
R = button();
if (R == 12) {
    LCD_CLS();
    A = 1;
    break;
}
LCD_gotoxy(z, 2); //Set cursor to low 1,col 1
LCD_Printnum2(R);
z = z + 1;
f = R;
A = A + 1;
}
while(A == 7){
    z = 11;
    LCD_gotoxy(1, 1); //Set cursor to low 2,col 2
    LCD_Print("TIME "); //Send String to Screen
    LCD_gotoxy(1, 2); //Set cursor to low 2,col 2
    LCD_Print("NORMAL I= "); //Send String to Screen
    LCD_gotoxy(z, 2); //Set cursor to low 1,col 1
    LCD_Printnum2(e);
    z = z + 1;

```

```

LCD_gotoxy(z, 2); //Set cursor to low 1,col 1
LCD_Printnum2(f);
R = button();
if ((R == 13)&&(C==0)) {
    LCD_CLS();
    e = e * 10;
    num = e + f;
    A = 8;
}
if ((R == 13)&&(C==1)) {
    LCD_CLS();
    e = e * 10;
    num = e + f;
    A = 9;
}
if (R == 14) {
    LCD_CLS();
    A = 6;
    C=0;
}
if (R == 12) {
    LCD_CLS();
    A = 1;
    C=1;
}
}

void Entervalue4(void)
{
    while (A == 8) {

```



```

num2 = 0;

z = 9;

LCD_gotoxy(1, 1); //Set cursor to low 1,col 1
LCD_Print("TIME "); //Send String to Screen
LCD_gotoxy(1, 2); //Set cursor to low 1,col 2
LCD_Print("OVER I= "); //Send String to Screen
R = button();
if (R == 12) {
    LCD_CLS();
    A = 7;
    e=e/10;
    break;
}
LCD_gotoxy(z, 2);
LCD_Printnum2(R);
z = z + 1;
m = R;
R = button();
if (R == 12) {
    LCD_CLS();
    A = 7;
    e=e/10;
    break;
}
LCD_gotoxy(z, 2);
LCD_Printnum2(R);
z = z + 1;
n = R;
A = A+1;
}
while(A == 9){

```

```

z = 9;

LCD_gotoxy(1, 1); //Set cursor to low 2,col 2
LCD_Print("TIME "); //Send String to Screen
LCD_gotoxy(1, 2); //Set cursor to low 2,col 2
LCD_Print("OVER I= "); //Send String to Screen
LCD_gotoxy(z, 2); //Set cursor to low 1,col 1
LCD_Printnum2(m);

z = z + 1;

LCD_gotoxy(z, 2); //Set cursor to low 1,col 1
LCD_Printnum2(n);

R = button();
if (R == 12) {
    LCD_CLS();

    A = 7;

    e=e/10;

    C = 1;
}

if (R == 14) {
    LCD_CLS();

    A = 8;
}

if (R == 15) {
    LCD_CLS();

    m = m * 10;

    num2 = m + n;

    if((num2 == 0)|| (num == 0)){

        A = 20;

    }else{

        A = 11;

    }

}

```

```

if (R == 13) {
    LCD_CLS();
    A = 9;
}
}
}

void show(void)
{
    while (A == 10) {
        LCD_gotoxy(1, 1); //Set cursor to low 1,col 1
        LCD_Print("NORMAL I= ");//Send String to Screen
        LCD_Printnum(num);
        LCD_gotoxy(1, 2); //Set cursor to low 2,col 2
        LCD_Print("OVER I = ");//Send String to Screen
        LCD_Printnum(num2);
        PORTB ^= (1 << 1);
        A = 12;
        delay(200);
        PORTB ^= (1 << 1);
    }
    while (A == 11) {
        LCD_gotoxy(1, 1); //Set cursor to low 1,col 1
        LCD_Print("NORMAL I= ");//Send String to Screen
        LCD_Printnum(num);
        LCD_Print(" s");
        LCD_gotoxy(1, 2); //Set cursor to low 2,col 2
        LCD_Print("OVER I = ");//Send String to Screen
        LCD_Printnum(num2);
        LCD_Print(" s");
    }
}

```

```

num =(num*1000)/20 ;
num2 =(num2*1000)/20;
A = 16;
}
}

```

```

void zerocrossing(void)
{
while (A == 12) {
do{
pulse = digitalRead(19);
if((pulse)==0)
{
while(pulse==0){
pulse= digitalRead(19);
if(pulse==1){
PORTB ^= 1 << 5;
delay(9);
PORTB ^= 1 << 5;
delay(1);
PORTB ^= 1 << 5;
delay(9);
PORTB ^= 1 << 5;
t++;
}
}
}
} while(t<num);
A=13;
t=0;
}
}

```

```

while (A == 13) {
    do{
        pulse = digitalRead(19);
        if((pulse)==0)
        {
            while(pulse==0){
                pulse= digitalRead(19);
                if(pulse==1){
                    PORTB ^= 1 << 5;
                    PORTC ^= 1 << 4;
                    delay(9);
                    PORTB ^= 1 << 5;
                    PORTC ^= 1 << 4;
                    delay(1);
                    PORTB ^= 1 << 5;
                    PORTC ^= 1 << 4;
                    delay(9);
                    PORTB ^= 1 << 5;
                    PORTC ^= 1 << 4;
                    t++;
                }
            }
        }
    } while(t<num2);
    A=14;
    t=0;
}

while (A == 14) {
    do{
        pulse = digitalRead(19);
        if((pulse)==0)

```

```

{
    while(pulse==0){
        pulse= digitalRead(19);
        if(pulse==1){
            PORTB ^= 1 << 5;
            delay(9);
            PORTB ^= 1 << 5;
            delay(1);
            PORTB ^= 1 << 5;
            delay(9);
            PORTB ^= 1 << 5;
            t++;
        }
    }
}

} while(t<num);

A=15;

t=0;

}

while(A == 15){
    R = buttom2();
    if (R == 14) {
        LCD_CLS();
        A = 1;
    }
}

}

}

void zerocrossing2(void)
{
    while (A == 16) {

```

```

do{
pulse = digitalRead(19);
if((pulse)==0)
{
while(pulse==0){
pulse= digitalRead(19);
if(pulse==1){
PORTB ^= 1 << 5;
delay(9);
PORTB ^= 1 << 5;
delay(1);
PORTB ^= 1 << 5;
delay(9);
PORTB ^= 1 << 5;
t++;
}
}
}
} while(t<num);
A=17;
t=0;
}

while (A == 17) {
do{
pulse = digitalRead(19);
if((pulse)==0)
{
while(pulse==0){
pulse= digitalRead(19);
if(pulse==1){
PORTB ^= 1 << 5;

```

```

    PORTC ^= 1 << 4;
    delay(9);
    PORTB ^= 1 << 5;
    PORTC ^= 1 << 4;
    delay(1);
    PORTB ^= 1 << 5;
    PORTC ^= 1 << 4;
    delay(9);
    PORTB ^= 1 << 5;
    PORTC ^= 1 << 4;
    t++;
  }
}
}
} while(t<num2);
A=18;
t=0;
}

while (A == 18) {
  do{
    pulse = digitalRead(19);
    if((pulse)==0)
    {
      while(pulse==0){
        pulse= digitalRead(19);
        if(pulse==1){
          PORTB ^= 1 << 5;
          delay(9);
          PORTB ^= 1 << 5;
          delay(1);
          PORTB ^= 1 << 5;

```



```

    delay(9);
    PORTB ^= 1 << 5;
    t++;
}
}
}
} while(t<num);
A=19;
t=0;
}
while(A == 19){
    R = button2();
    if (R == 14) {
        LCD_CLS();
        A = 1;
    }
}
}

void inputzero(void)
{
    while(A==20){
        LCD_gotoxy(6, 1); //Set cursor to low 1,col 1
        LCD_Print("Error");//Send String to Screen
        R = button();
        if (R == 14) {
            LCD_CLS();
            A = 1;
        }
    }
}
}

```

```

int button2(void)
{
    int data[4][4] = {
        {1, 2, 3, 10},
        {4, 5, 6, 11},
        {7, 8, 9, 12},
        {14, 0, 15, 13}
    };
    int i,in, result;
    while (1) {
        i=3;
        PORTD = (3 << 6);
        in = digitalRead(14);
        if ((in == 1)&&(i==3))
        {
            PORTB ^= (1 << 1);
            result = data[3][0];
            delay(200);
            while (digitalRead(14) == 1);
            in = 0;
            PORTB ^= (1 << 1);
            return result;
            break;
        }
    }
}

int button(void)
{
    int data[4][4] = {
        {1, 2, 3, 10},
        {4, 5, 6, 11},

```

```

    {7, 8, 9, 12},
    {14, 0, 15, 13}
};

int i, j, out, in, result;
while (1) {
    for (i = 0; i < 4; i++)
    {
        PORTD = (i << 6);
        for (j = 14; j < 18; j++)
        {
            in = digitalRead(j);
            if (in == 1)
            {
                result = data[i][j - 14];
                delay(200);
                while (digitalRead(j) == 1);
                in = 0;
                return result;
                break;
            }
        }
    }
}

void setup() {
    Serial.begin(9600);
    DDRC = 0b00010000;
    LCD_DDDR = 0xFF;    //Set PortD for Output
    LCD_BDDR = 0xFF;    //Set PortB for Output
    Init_LCD4D();       //Set-up LCD
}

```

```

}

void loop() {
  LCDhomepage();
  MenuValue();
  Entervalue();
  Entervalue2();
  Entervalue3();
  Entervalue4();
  show();
  zerocrossing();
  zerocrossing2();
  inputzero();
}

```

## 2. โปรแกรมในไมโครคอนโทรลเลอร์ 2

```

#define F_CPU 16000000UL //MPU Clock Freq.
#include <avr/io.h>
#include <util/delay.h> //For _Delay_us() and _Delay_ms() function
#define LCD_DPRT PORTD //PortD.5-D.12
#define LCD_DDDR DDRD //in Arduino Pin5-2
#define LCD_DPIN PIND //LCD Data Pin11-14
#define LCD_BPRT PORTB //PortB.4-B.2
#define LCD_BDDR DDRB //
#define LCD_BPIN PINB //
#define LCD_RS 4 //PB.4 Arduino Pin 12
#define LCD_RW 3 //PB.3 Arduino Pin 11
#define LCD_EN 2 //PB.2 Arduino Pin 10
//LCD module information
//LCD Instructions

```

```

#define LCD_Clear    0b00000001    //Replace all characters with ASCII 'Space'
#define LCD_Home     0b00000010    //Return cursor to first position on first line
#define LCD_EntryMode 0b00000110    //Shift cursor from left to right on read/write
#define LCD_DisplayOff 0b00001000 //Turn display
#define LCD_DisplayOn 0b00001100    //Display on, cursor off, don't blink character
#define LCD_FunctionReset 0b00110010 //Reset the LCD
#define LCD_FunctionSet4bit 0b00101000 //4-bit data, 2-line display, 5x7 font
#define LCD_SetCursor 0b10000000    //Set cursor position
#define LCD_CGRAM_ADDRESS 0b01000000 //Set CGRAM_ADDRESS
#define LCD_DDRAM_ADDRESS 0b10000000 //Set DDRAM_ADDRESS
void LCD_CMNDWRT42(unsigned char cmnd) //Subprogram Write Command 4-bit Data 2 times
{
    unsigned char A2 = 0;    //Define A2
    A2 = cmnd;                //Back up Data(cmnd)
    cmnd &= 0b11110000;      //Mask-Low Nibble Clear
    LCD_DPRT = cmnd >> 2;    //Shift Right 2 bit and Send
                             //(High Nibble) to PORTD
    LCD_BPRT &= 0b11100100; //Set LCD_RS=0 for Select Instruction register
                             //and Set LCD_RW=0 for write
    LCD_BPRT |= 0b00000100; //Set LCD_EN=1 for Starts data read/write
    _delay_us(1);           //wait about 1us.
    LCD_BPRT &= 0b11111000; //Set LCD_EN=0 for Ends data read/write
    cmnd = A2 << 4;          //Load back up Data
    cmnd &= 0b11110000;      //Mask-Low Nibble Clear
    LCD_DPRT = cmnd >> 2;    //Shift Right 2 bit and Send
                             //(High Nibble) to PORTD
    LCD_BPRT |= 0b00000100; //Set LCD_EN=1 for Starts data read/write
    _delay_us(1);           //wait about 1us.
    LCD_BPRT &= 0b11111000; //Set LCD_EN=0 for Ends data read/write
    _delay_us(100);         //wait about 100us.
}

```

```

void LCD_CLS(void)      //Subprogram Clear LCD
{
    unsigned char x = 0x01; //Define x
    LCD_CMNDWRT42(x);      //Write Command for Clear
    _delay_ms(2);          //wait about 2us.
}

void LCD_CMNDWRT41(unsigned char cmnd) //Subprogram Write Command 4-bit Data 1 time
{
    unsigned char A1 = 0; //Define A1
    A1 = cmnd;             //Back up Data(cmnd)
    cmnd &= 0b11110000;    //Mask-Low Nibble Clear
    LCD_DPRT = cmnd >> 2;  //Shift Right 2 bit and Send
    //(High Nibble) to PORTD
    LCD_BPRT &= 0b11100100; //Set LCD_RS=0 for Select Instruction register
    //and Set LCD_RW=0 for write
    LCD_BPRT |= 0b00000100; //Set LCD_EN=1 for Starts data read/write
    _delay_us(1);          //wait about 1us.
    LCD_BPRT &= 0b11111000; //Set LCD_EN=0 for Ends data read/write
}

void Init_LCD4D(void) //Subprogram Initialize LCD 4-bit data Mode
{
    _delay_ms(50);        //wait about 50ms.
    LCD_CMNDWRT41(0x30);  //special case of Function Set
    _delay_ms(5);         //wait about 5ms.
    LCD_CMNDWRT41(0x30);  //special case of Function Set
    _delay_us(100);       //wait about 100us.
    LCD_CMNDWRT41(0x30);  //special case of Function Set
    _delay_us(100);       //wait about 100us.
    LCD_CMNDWRT41(0x20);  //Initial Function Set to Change Interface
    _delay_us(100);       //wait about 100us.
    LCD_CMNDWRT42(0x28);  //Function Set N=1,F=0 2line and 5*8 Dot

```

```

LCD_CMNDWRT42(0x0C); //Display ON D=1,C=1 Display on and Cursor On
LCD_CLS();          //Clear LCD
LCD_CMNDWRT42(0x06); //Entry Mode Set 1/D=1, S=0
}

void LCD_DATWRT42(unsigned char dat) //Subprogram Write Data 4-bit Data 2 times
{
    unsigned char A3 = 0; //Define A3
    A3 = dat;             //Back up Data(dat)
    dat &= 0b11110000;    //Mask-Low Nibble Clear
    LCD_DPRT = dat >> 2;  //Shift Right 2 bit and Send
    //(High Nibble) to PORTD
    LCD_BPRT &= 0b11110100; //Set LCD_RW=0 for write
    LCD_BPRT |= 0b00010100; //Set LCD_RS=1 for Select Data register
    //and Set LCD_EN=1 for Starts data read/write
    _delay_us(1);         //wait about 1us.
    LCD_BPRT &= 0b11111000; //Set LCD_EN=0 for Ends data read/write
    dat = A3 << 4;         //Load back up Data
    dat &= 0b11110000;    //Mask-Low Nibble Clear
    LCD_DPRT = dat >> 2;  //Shift Right 2 bit and Send
    //(High Nibble) to PORTD
    LCD_BPRT |= 0b00000100; //Set LCD_EN=1 for Starts data read/write
    _delay_us(1);         //wait about 1us.
    LCD_BPRT &= 0b11111000; //Set LCD_EN=0 for Ends data read/write
    _delay_us(100);       //wait about 100us.
}

void LCD_gotoxy(uint8_t col, uint8_t row) //Subprogram Set Position Cursor
{
    unsigned int s[] = {0x80, 0xC0}; //Define s[]
    LCD_CMNDWRT42(s[row - 1] + (col - 1)); //Write Command for Set Cursor
    _delay_us(1000);         //wait about 1000us.
}

```

```

void LCD_Print(char*str) //Subprogram Print String
{
    unsigned char y = 0; //Define y
    while (str[y] != 0) //ทำงานเมื่อมีข้อมูลใน str[]
    {
        LCD_DATWRT42(str[y]); //Write Data from str[]
        y++; //เลื่อน Pointer
    }
}

void LCD_Printnum(float num) //Subprogram Print number
{
    int y = num; //Define y
    int A,B,C,D,F,G,H,I,J,K,L;
    float E = num;
    D = y % 10;
    B = D;
    D = D + 0x30;
    C = y / 10;
    A = C;
    C = C + 0x30;
    F = (A*10)+B;
    E = E-F;
    E = (E*1000);
    J = E;
    K = J /100;
    L = K*100;
    K = K + 0x30;
    J = J-L;
    G = J / 10;
    G = G + 0x30;

```



```

H = J % 10;
H = H + 0x30;
I = I + 0x2E;
LCD_DATWRT42(C);
LCD_DATWRT42(D);
LCD_DATWRT42(I);
LCD_DATWRT42(K);
LCD_DATWRT42(G);
LCD_DATWRT42(H);

}

int offset = 512 ;
int readValue,readValue2;
double Amp = 0,Amp2=0;
double Real = 0,Real2 = 0;
double result,result2;
long sample = 0,sample2 = 0;
int mVperAmp = 66;
long sqI=0,sqI2=0;
int A=0;
double sumdelay,sumdelay2;
double avgsumdelay = 0,avgsumdelay2 = 0;
double Ampdelay=0,Ampdelay2=0;
double Realdelay=0,Realdelay2=0;
uint32_t d = 0,e = 0;
int in,input,B=0;

void setup() {
LCD_DDDR = 0xFF;    //Set PortD for Output

```

```

LCD_BDDR = 0xFF;    //Set PortB for Output
DDRC = 0x00;        //Set PortC for Input
Init_LCD4D();
Serial.begin(9600);
}

```

```

void loop() {
while(B==0){
input = digitalRead(16);
if(input == 1)
{
delay(200);
while (digitalRead(16) == 1);
input = 0;
B=1;
}
int n = 0;
long sumI = 0;
long sumI2 = 0;
uint32_t start_time = millis();
while(((millis())-start_time) < 100){
readValue=analogRead(A0);
sample=readValue-offset;
sqI=sample*sample;
sumI = sumI+sqI;
readValue2=analogRead(A1);
sample2=readValue2-offset;
sqI2=sample2*sample2;
sumI2 = sumI2 + sqI2;
n++;
}
}

```

```

result = (sqrt(sumI/n));
result2 = (sqrt((sumI2)/n));
Amp = (result*5.0)/1024.0;
Amp2 = ((result2)*5.0)/1024.0;
Real = (Amp*1000)/mVperAmp;
Real2 = ((Amp2)*1000)/mVperAmp;
if((result < 4)&&(A==0)){
    A=1;
}
if ( ((result >= 4)&&(A==1)) || ((result >= 2)&&(A==2)) )
{
    if(result2 >= (result*1.2)){
        sumdelay = sumdelay;
    }else{
        sumdelay = sumdelay+result;
        d++;
    }
}
if (((result >= 4)&&(A==1)&&(result2 > (result*1.2)))||((result >= 4)&&(A==2)&&(result2 >
(result*1.2))))
{
    sumdelay2 = sumdelay2+result2;
    e++;
    A=2;
}

if((result <4)&&(A==2))
{
    avgsumdelay = sumdelay/d;
    avgsumdelay2 = sumdelay2/e;
    Ampdelay = (avgsumdelay*5.0)/1024.0;

```

```

Ampdelay2 = ((avgsumdelay2)*5.0)/1024.0;
Realdelay = (Ampdelay*1000)/mVperAmp;
Realdelay2 = ((Ampdelay2)*1000)/mVperAmp;
A=3;
}
while(A==3)
{
    LCD_gotoxy(1, 1);      //Set cursor to row 1,col 1
    LCD_Print("Inormal= "); //Send String to Screen
    LCD_Printnum(Realdelay);
    LCD_Print("A");
    LCD_gotoxy(1, 2);      //Set cursor to row 2,col 1
    LCD_Print("Iover = "); //Send String to Screen
    LCD_Printnum(Realdelay2);
    LCD_Print("A");
    A=4;
}
while(A==4)
{
    in = digitalRead(16);
    if(in == 1)
    {
        Serial.println("Have Input");
        delay(200);
        while (digitalRead(16) == 1);
        in = 0;
        sumdelay = 0;
        sumdelay2 = 0;
        d=0;
        e=0;
        A=0;
    }
}

```

```

}
}

LCD_gotoxy(1, 1);      //Set cursor to row 1,col 1
LCD_Print("Inormal= "); //Send String to Screen
LCD_Printnum(Real);
LCD_Print("A");
LCD_gotoxy(1, 2);      //Set cursor to row 2,col 1
LCD_Print("Iover = "); //Send String to Screen
LCD_Printnum(Real2);
LCD_Print("A");
}

```

```

while(B==1){
int n = 0;
long sumI = 0;
long sumI2 = 0;
uint32_t start_time = millis();
while((((millis())-start_time) < 20){
readValue=analogRead(A0);
sample=readValue-offset;
sqI=sample*sample;
sumI = sumI+sqI;
readValue2=analogRead(A1);
sample2=readValue2-offset;
sqI2=sample2*sample2;
sumI2 = sumI2 + sqI2;
n++;
}
result = (sqrt(sumI/n));
result2 = (sqrt((sumI2)/n));
Amp = (result*5.0)/1024.0;

```

```

Amp2 = ((result2)*5.0)/1024.0;
Real = (Amp*1000)/mVperAmp;
Real2 = ((Amp2)*1000)/mVperAmp;
if((result < 4)&&(A==0)){
    A=1;
}
if ( ((result >= 4)&&(A==1)) || ((result >= 4)&&(A==2)) )
{
    if(result2 >= (result*1.2)){
        sumdelay = sumdelay;
    }else{
        sumdelay = sumdelay+result;
        d++;
    }
}
if (((result >= 4)&&(A==1)&&(result2 > (result*1.2))||((result >= 4)&&(A==2)&&(result2 >
(result*1.2))))
{
    sumdelay2 = sumdelay2+result2;
    e++;
    A=2;
}

if((result <4)&&(A==2))
{
    avgsumdelay = sumdelay/d;
    avgsumdelay2 = sumdelay2/e;
    Ampdelay = (avgsumdelay*5.0)/1024.0;
    Ampdelay2 = ((avgsumdelay2)*5.0)/1024.0;
    Realdelay = (Ampdelay*1000)/mVperAmp;
    Realdelay2 = ((Ampdelay2)*1000)/mVperAmp;
}

```

```

    A=3;
}
while(A==3)
{
    LCD_gotoxy(1, 1);      //Set cursor to row 1,col 1
    LCD_Print("Inormal= "); //Send String to Screen
    LCD_Printnum(Realdelay);
    LCD_Print("A");
    LCD_gotoxy(1, 2);      //Set cursor to row 2,col 1
    LCD_Print("Iover = "); //Send String to Screen
    LCD_Printnum(Realdelay2);
    LCD_Print("A");
    A=4;
}
while(A==4)
{
    in = digitalRead(16);
    if(in == 1)
    {
        Serial.println("Have Input");
        delay(200);
        while (digitalRead(16) == 1);
        in = 0;
        sumdelay = 0;
        sumdelay2 = 0;
        d=0;
        e=0;
        A=0;
        B=0;
    }
}

```

```
LCD_gotoxy(1, 1);      //Set cursor to row 1,col 1
LCD_Print("Inormal= "); //Send String to Screen
LCD_Printnum(Real);
LCD_Print("A");
LCD_gotoxy(1, 2);      //Set cursor to row 2,col 1
LCD_Print("Iover = "); //Send String to Screen
LCD_Printnum(Real2);
LCD_Print("A");
}
}
```



### ประวัติผู้แต่ง

ปริญญานิพนธ์ : เครื่องกำเนิดกระแสเกินชั่วคราว

สาขาวิชา : วิศวกรรมไฟฟ้า

ภาควิชา : วิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะ : วิศวกรรมศาสตร์

ชื่อ : นางสาวสุกนิดา ชาญหัตถศิลป์

#### ประวัติ

เกิดเมื่อวันที่ 7 พฤษภาคม 2536 ปัจจุบันอาศัยอยู่บ้านเลขที่ 144/57 ซอย 13 หมู่บ้านเมืองประชา ตำบลบางคูวัด อำเภอเมือง จังหวัดปทุมธานี สำเร็จศึกษาระดับประกาศนียบัตรวิชาชีพ จากโรงเรียนเตรียมวิศวกรรมศาสตร์ วิทยาลัยเทคโนโลยีอุตสาหกรรม มหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ ปีการศึกษา 2554 และสำเร็จการศึกษาระดับปริญญาตรี สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จากมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ กรุงเทพมหานคร ปีการศึกษา 2559

ชื่อ : นางสาวธนาวดี แซ่อึ้ง

#### ประวัติ

เกิดเมื่อวันที่ 14 กุมภาพันธ์ 2537 ปัจจุบันอาศัยอยู่บ้านเลขที่ 1747/9 หมู่ 9 ถนนสุขุมวิท ตำบลเทพารักษ์ อำเภอเมือง จังหวัดสมุทรปราการ สำเร็จการศึกษาระดับมัธยมศึกษาตอนปลาย จากโรงเรียนเซนต์โยเซฟ บางนา ปีการศึกษา 2554 และสำเร็จการศึกษาระดับปริญญาตรี สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์ คณะวิศวกรรมศาสตร์ จากมหาวิทยาลัยเทคโนโลยีพระจอมเกล้าพระนครเหนือ กรุงเทพมหานคร ปีการศึกษา 2559