อภินันทนาการ

สำนักหอสมุด

สวิตช์สัมผัสไร้สาย

WIRELESS PROXIMITY SWITCH

นายศุภณัฐ ค้ำชู รหัส 52362243

นายอรรถดร สุวรรณศรี รหัส 52362359

สำนักหอสมุด มหาวิทยาลัยนเรศวร
วันลงทะเบียน 24 ม.ค. 2561
เลขทะเบียน 1928137
เลขเรียกหนังสือ ปร
ศ191ส
ว556

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมไฟฟ้า ภาควิชาวิศวกรรมไฟฟ้าและคอมพิวเตอร์

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร

ปีการศึกษา 2556

ใบรับรองปริญญานิพนธ์

ชื่อหัวข้อโครงงาน	สวิตช์สัมผัสไร้สาย		
ผู้ดำเนินโครงงาน	นายศุภณัฐ	ค้ำชู	รหัส 52362243
	นายอรรถดร	สุวรรณศรี	รหัส 52362359
ที่ปรึกษาโครงงาน	รศ.ดร. ยงยุทธ ชนบดีเฉลิมรุ่ง		
สาขาวิชา	วิศวกรรมไฟฟ้า		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2556		

.....

คณะวิศวกรรมศาสตร์ มหาวิทยาลัยนเรศวร อนุมัติให้ปริญญานิพนธ์ฉบับนี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรวิศวกรรมศาสตรบัณฑิต สาขาวิชาวิศวกรรมไฟฟ้า

.....ที่ปรึกษาโครงงาน
(รศ.ดร. ยงยุทธ ชนบดีเฉลิมรุ่ง)

.....กรรมการ
(ดร. สุพรรณนิกา วัฒนะ)

.....กรรมการ
(ดร. พรพิศุทธิ์ วรจิรันตน์)

ชื่อหัวข้อโครงงาน	สวิทซ์สัมผัสไร้สาย		
ผู้ดำเนินโครงงาน	นายศุภณัฐ	ค้ำชู	รหัส 52362243
	นายอรรถดร	สุวรรณศรี	รหัส 52362359
ที่ปรึกษาโครงงาน	รศ.ดร. ยงยุทธ ชนบดีเฉลิมรุ่ง		
สาขาวิชา	วิศวกรรมไฟฟ้า		
ภาควิชา	วิศวกรรมไฟฟ้าและคอมพิวเตอร์		
ปีการศึกษา	2556		

บทคัดย่อ

วัตถุประสงค์ของโครงงานนี้ คือ การสาธิตระบบสวิตช์สัมผัสไร้สาย และศึกษาลักษณะการทำงานของระบบโดยการทำงานนั้นจะมีการทำงาน 3 แบบ คือการเปิดไฟ การปิดไฟ และการหรี่ไฟ และจะทำการทดลองการทำงานโดยใช้อุปกรณ์ให้แสงสว่าง (หลอดไฟ) 4 ชนิด คือ การทำงานของหลอดไส้ การทำงานของหลอดคอมแพคฟลูออเรสเซนต์ (แบบธรรมดา) การทำงานของหลอดคอมแพคฟลูออเรสเซนต์ (แบบหรี่ไฟได้) และการทำงานของหลอด LED 220V และทดสอบระยะการทำงานของเครือข่ายไร้สาย จากการทดลองพบว่า การเปิดไฟ และการปิดไฟนั้นจะสามารถทำงานได้กับหลอดไฟทุกชนิด ส่วนการหรี่ไฟนั้นจะสามารถทำงานได้กับหลอดไฟบางชนิดเท่านั้น

Project title	Wireless Proximity Switch	
Name	Mr. Supanat Kumchoo	ID. 52362243
	Mr. Attadon Suwansri	ID. 52362359
Project advisor	Assoc. Prof. Dr. Yongyut Chonbodeechalermroong	
Major	Electrical Engineering	
Department	Electrical and Computer Engineering	
Academic year	2013	

Abstract

The objective of this project is to demonstrate about the system of wireless proximity switch and to study about the working process of this system. The working process is divided into 3 categories: turning on, turning off and dimming the light. Moreover, in the experiment, 4 kinds of light bulbs were brought for testing the working process: the working process for incandescent light bulb, the working process for compact fluorescent lamp (normal), the working process for compact fluorescent lamp (dimmable lamp) and the working process for LED 220V bulb. This project also studies the working length of the wireless network. The result of the experiment indicates that it can work with all kind of the light bulbs when turning on and turning off the light. However, it can work with only some kinds of the light bulbs when dimming.

กิตติกรรมประกาศ

โครงงานนี้สำเร็จลุล่วงไปได้ด้วยความกรุณาช่วยเหลือของ รศ.ดร.ยงยุทธ ชนบดีเฉลิมรุ่ง อาจารย์ที่ปรึกษาโครงงานที่ได้ให้คำแนะนำ แนวคิด ตลอดจนแก้ไขปัญหาข้อบกพร่องต่างๆ มาโดยตลอดจนโครงงานเล่มนี้เสร็จสมบูรณ์ จึงขอกราบขอบพระคุณเป็นอย่างสูง

ขอกราบขอบพระคุณ ดร. สุพรรณนิกา วัฒนะ และ ดร. พรพิศุทธิ์ วรจิรันตน์ ที่ปรึกษาร่วมโครงงาน ที่ให้คำปรึกษาที่มีค่า

สุดท้ายนี้ขอขอบพระคุณทุกๆ ท่านที่ได้สนับสนุนการทำงาน และให้กำลังใจแก่คณะผู้ดำเนินโครงงานเสมอมา กระทั่งการดำเนินโครงงานครั้งนี้สำเร็จลุล่วงด้วยดี และความดีอันเกิดจากการดำเนินโครงงานครั้งนี้ คณะผู้ดำเนินโครงงานขอมอบแด่บิดา มารดา ครู อาจารย์ และผู้มีพระคุณทุกท่าน คณะผู้ดำเนินโครงงานมีความซาบซึ้งในความกรุณาอันดียิ่งจากทุกท่านที่ได้กล่าวนามมา และขอกราบขอบพระคุณมา ณ โอกาสนี้

นายศุภณัฐ ค้ำชู

นายอรรถดร สุวรรณศรี

สารบัญ

หน้า

ใบรับรองปริญญานิพนธ์ ..... ก

บทคัดย่อภาษาไทย ..... ข

บทคัดย่อภาษาอังกฤษ ..... ค

กิตติกรรมประกาศ ..... ง

สารบัญ ..... จ

สารบัญตาราง ..... ช

สารบัญรูป ..... ฌ

บทที่ 1 บทนำ 1

1.1 ที่มาและความสำคัญของโครงงาน 1

1.2 วัตถุประสงค์ของโครงงาน 2

1.3 ขอบเขตโครงงาน 2

1.4 ขั้นตอนและแผนการดำเนินงาน 2

1.5 ประโยชน์ที่คาดว่าจะได้รับจากโครงงาน 3

1.6 งบประมาณ 3

บทที่ 2 หลักการและทฤษฎีเบื้องต้น 4

2.1 พร็อกซิมิตี้เซนเซอร์ 4

2.1.1 เซนเซอร์แบบเหนี่ยวนำ 4

2.1.2 เซนเซอร์ชนิดเก็บประจุ 4

สารบัญ (ต่อ)

หน้า

2.2 Microcontroller PIC16F877 8
2.2.1 คุณสมบัติทางเทคนิคของ PIC16F877 8
2.2.2 หน้าที่ขาสัญญาณต่าง ๆ ของ PIC16F877 9
2.2.3 การออกแบบและเลือกใช้แหล่งกำเนิดสัญญาณนาฬิกา 13
2.2.4 การรีเซตไมโครคอนโทรลเลอร์ PIC 17
2.3 TRIAC 19
2.3.1 การควบคุมกำลังไฟแบบเฟสทริกเกอร์ 20
2.4 RF1100-232 Wireless RF 21
2.5 ชนิดของหลอดไฟ 22
2.5.1 หลอดไส้ หรือ หลอดอินแคนเดสเซนท์ 22
2.5.2 หลอดตะเกียบ หรือ หลอดคอมแพคฟลูออเรสเซนต์ 22
2.5.3 หลอดไฟ LED 23

บทที่ 3 วิธีดำเนินโครงงาน 24
3.1 ศึกษา และเก็บข้อมูล 24
3.2 ออกแบบการทดลอง 24
3.3 เขียนโปรแกรม 25
3.4 ทดสอบ และปรับปรุง 25
3.5 ขั้นตอนการทำงานของระบบ 25

สารบัญ (ต่อ)

หน้า

บทที่ 4 ผลการทดลอง 27

4.1 ผลการทดลองที่ 1 การทดลองระยะตรวจจับผ่านสิ่งกีดขวางของเซนเซอร์ 27

4.1.1 สรุปผลการทดลองที่ 1 30

4.2 ผลการทดลองที่ 2 การทดลองโดยใช้หลอดไฟแบบต่างๆ 30

4.2.1 สรุปผลการทดลองที่ 2 37

4.3 ผลการทดลองที่ 3 การทดลองระยะการทำงานของระบบไร้สาย 38

4.3.1 สรุปผลการทดลองที่ 3 38

บทที่ 5 สรุปผลการทดลอง และปัญหาที่พบ 39

5.1 สรุปผลการทดลอง 39

5.2 ปัญหาที่พบ 39

5.3 การพัฒนา และปรับปรุงแก้ไข 39

ภาคผนวก 40

เอกสารอ้างอิง 62

ประวัติผู้ดำเนินโครงงาน 63

สารบัญตาราง

ตารางที่ หน้า

2.1 หน้าที่ของขาสัญญาณของพอร์ต A ของไมโครคอนโทรลเลอร์ PIC16F877 10

2.2 หน้าที่ของขาสัญญาณของพอร์ต B ของไมโครคอนโทรลเลอร์ PIC16F877 11

2.3 หน้าที่ของขาสัญญาณของพอร์ต C ของไมโครคอนโทรลเลอร์ PIC16F877 11

2.4 หน้าที่ของขาสัญญาณของพอร์ต C ของไมโครคอนโทรลเลอร์ PIC16F877 (ต่อ) 12

2.5 หน้าที่ของขาสัญญาณของพอร์ต D ของไมโครคอนโทรลเลอร์ PIC16F877 12

2.6 หน้าที่ของขาสัญญาณของพอร์ต E ของไมโครคอนโทรลเลอร์ PIC16F877 13

2.7 การเลือกค่า C1 และ C2 ตามโหมด ค่าความถี่ของคริสตอล 14

2.8 การเลือกค่า C1 และ C2 ตามค่าความถี่ของเซรามิกเรโซเนเตอร์ 15

2.9 การเลือกค่า R และ C ค่าความถี่ออสซิลเลเตอร์ 16

4.1 ตารางแสดงการทำงานระยะการตรวจจับผ่านวัตถุชนิดไม้ 17

4.2 ตารางแสดงการทำงานระยะการตรวจจับผ่านวัตถุชนิดยางสังเคราะห์ 28

4.3 ตารางแสดงการทำงานของหลอดไฟชนิดต่างๆ 30

4.4 ตารางแสดงระยะการทำงานของระบบไร้สาย (Wireless) 38

สารบัญรูป

รูปที่ หน้า

2.1 ส่วนประกอบของ Capacitive Proximity Sensor 5

2.2 ส่วนประกอบและการกระจายสนามไฟฟ้าสถิตของ Capacitive Proximity Sensor 5

2.3 หลักการทำงานของ Capacitive Proximity Sensor 6

2.4 การต่อใช้งานเซนเซอร์แบบ PNP 7

2.5 การต่อใช้งานเซนเซอร์แบบ NPN 7

2.6 แสดงโครงสร้างการจัดขาของไมโครคอนโทรลเลอร์ PIC16F877 9

2.7 วงจรออสซิลเลเตอร์ใช้คริสตอลออสซิลเลเตอร์ 14

2.8 วงจรออสซิลเลเตอร์ใช้เซรามิกเรโซเนเตอร์ 15

2.9 วงจรออสซิลเลเตอร์ใช้ตัวต้านทานและตัวเก็บประจุ 16

2.10 วงจรออสซิลเลเตอร์จากแหล่งภายนอก 17

2.11 วงจรรีเซตแบบอัตโนมัติไมโครคอนโทรลเลอร์ PIC 18

2.12 วงจรรีเซตแบบสวิทช์กดและแบบอัตโนมัติไมโครคอนโทรลเลอร์ PIC 18

2.13 แสดงรูปคลื่นการเกิดการรีเซตภายในไมโครคอนโทรลเลอร์ช่วงเริ่มทำงาน 18

2.14 โครงสร้างและสัญลักษณ์ของไตรแอค 20

2.15 ลักษณะสมบัติกระแส – แรงดันของไตรแอค 20

2.16 การเปลี่ยนแปลงค่าของกำลังไฟฟ้าที่ป้อนให้แก่โหลด โดยกำหนดได้จากตำแหน่งเวลาของการทริกที่ให้แก่ไตรแอค 21

2.17 โครงสร้างของหลอดไส้ 22

2.18 โครงสร้างของหลอดคอมแพคฟลูออเรสเซนต์ 23

2.19 โครงสร้างของหลอดไฟ LED 23

3.1 แผนภาพแสดงภาพรวมของระบบ 24

3.2 โฟลว์ชาร์ตแสดงการทำงานของคอนโทรลเลอร์ฝั่งส่งสัญญาณ 25

สารบัญรูป (ต่อ)

รูปที่ หน้า

3.3 โฟลว์ชาร์ตแสดงการทำงานของคอนโทรลเลอร์ฝั่งรับสัญญาณ 26

4.1 ไม้หนา 3 มิลลิเมตร 27

4.2 ไม้หนา 6 มิลลิเมตร 28

4.3 ไม้หนา 9 มิลลิเมตร 28

4.4 ยางสังเคราะห์หนา 3 มิลลิเมตร 29

4.5 ยางสังเคราะห์หนา 6 มิลลิเมตร 29

4.6 ยางสังเคราะห์หนา 9 มิลลิเมตร 29

4.7 หลอดไฟชนิดต่างๆที่ใช้ในการทดลอง 30

4.8 ภาพแสดงอุปกรณ์ที่ใช้ทำการทดลอง 31

4.9 ภาพแสดงการปิดของหลอดไส้ 31

4.10 ภาพแสดงการเปิดของหลอดไส้ 32

4.11 ภาพแสดงการหรี่ไฟของหลอดไส้ 32

4.12 ภาพแสดงการปิดของหลอดตะเกียบ (แบบธรรมดา) 33

4.13 ภาพแสดงการเปิดของหลอดตะเกียบ (แบบธรรมดา) 33

4.14 ภาพแสดงการหรี่ไฟของหลอดตะเกียบ (แบบธรรมดา) 34

4.15 ภาพแสดงการปิดของหลอดตะเกียบ (แบบหรี่ไฟได้) 34

4.16 ภาพแสดงการเปิดของหลอดตะเกียบ (แบบหรี่ไฟได้) 35

4.17 ภาพแสดงการหรี่ไฟของหลอดตะเกียบ (แบบหรี่ไฟได้) 35

4.18 ภาพแสดงการปิดของหลอด LED 220V 36

4.19 ภาพแสดงการเปิดของหลอด LED 220V 36

บทที่ 1

บทนำ

1.1 ที่มาและความสำคัญของโครงงาน

ในปัจจุบันมีเทคโนโลยีที่สร้างมาเพื่ออำนวยความสะดวกสบายให้มนุษย์เกิดขึ้นมาอย่างมากมาย เพื่อช่วยให้การดำเนินชีวิตของมนุษย์นั้นง่ายและสะดวกสบายมากยิ่งขึ้น

สวิตช์สัมผัสเป็นนวัตกรรมใหม่ของสวิตช์ทางไฟฟ้าระบบสวิตช์สัมผัสเป็นสวิตช์อิเล็กทรอนิกส์ใช้การสื่อสารแบบรหัสข้อมูลซึ่งทั่วโลกยอมรับในความเที่ยงตรงแม่นในการใช้งาน คำสั่งเหล่านี้จะถูกประมวลผลเพื่อสั่งเปิดหรือปิดโคมไฟจากสวิตช์ระบบสวิตช์สัมผัสใช้ระบบไฟควบคุมเพียง 12VDC จึงตัดปัญหาเรื่องไฟช็อตไปได้เลย และมีความสวยงามควบคุมง่ายเพียงใช้แค่ปลายนิ้วสัมผัส

เครือข่ายไร้สาย หมายถึงระบบเครือข่ายที่เชื่อมต่อหรือแลกเปลี่ยนข้อมูลระหว่างกันโดยใช้สื่อกลางแบบไร้สายเช่นคลื่นวิทยุหรือคลื่นไมโครเวฟอินฟราเรดและ Bluetooth เป็นต้นวิวัฒนาการของเทคโนโลยีไร้สายเริ่มในปี ค.ศ.1896 โดยนำมาใช้ในการสื่อสารครั้งแรกเพื่อส่งโทรเลขแบบไร้สาย ต่อมาได้มีการพัฒนาเทคโนโลยีการสื่อสารและการส่งข้อมูลแบบไร้สายในรูปแบบต่าง ๆ เช่นคลื่นวิทยุดาวเทียมโทรศัพท์ไร้สายและการแลกเปลี่ยนข้อมูลด้วย Bluetooth เป็นต้น

ดังนั้นโครงงานนี้จึงได้นำสวิตช์สัมผัสและเครือข่ายไร้สายมาทำงานร่วมกันเพื่อเพิ่มประสิทธิภาพในการทำงานของอุปกรณ์เพื่อเพิ่มความสะดวกสบายในการดำเนินชีวิตโดยนำมาควบคุมอุปกรณ์ประเภทหลอดไฟ

1.2 วัตถุประสงค์ของโครงงาน

1.2.1 ออกแบบและสร้างระบบสาธิตการทำงานของระบบสวิตช์สัมผัสไร้สายกับหลอดไฟ

1.2.2 ทดสอบการทำงานของระบบสาธิตการทำงานของระบบสวิตช์สัมผัสไร้สายกับหลอดไฟ

1.3 ขอบเขตโครงงาน

1.3.1 ออกแบบและสร้างระบบสาธิตการทำงานของระบบสวิตช์สัมผัสไร้สาย

1.3.2 สร้างระบบสวิตช์สัมผัสเพื่อควบคุมการ เปิด/ปิดและหรี่ไฟได้

1.3.3 สร้างเครือข่ายไร้สายเพื่อเชื่อมต่อระหว่างสวิตช์สัมผัสและดวงไฟได้

1.4ขั้นตอนและแผนการดำเนินงาน

รายละเอียด	ปี 2556									ปี 2557	
	เม.ย.	พ.ค.	มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.
1.4.1 ศึกษาการทำงานของระบบสวิตช์สัมผัส											
1.4.2 ศึกษาการทำงานของระบบเครือข่ายไร้สาย											
1.4.3สร้างระบบจำลองต่างๆและปรับปรุงระบบ											
1.4.4จัดทำปริญญานิพนธ์ฉบับสมบูรณ์											

1.5ประโยชน์ที่คาดว่าจะได้รับจากโครงงาน

1.5.1 ได้ระบบสวิตช์สัมผัสไร้สาย

1.5.2 ได้เรียนรู้การสร้างระบบสวิตช์สัมผัสไร้สาย

1.5.3 นำไปประยุกต์ใช้งานได้จริง

1.6งบประมาณ

1.6.1 ค่าอุปกรณ์และเครื่องมือในการทำ Hardware	2,500 บาท
1.6.2 ค่าจัดทำปริญญานิพนธ์	700 บาท
1.6.3 รวมเป็นเงินทั้งสิ้น(สามพันสองร้อยบาทถ้วน)	3,200 บาท

1.6.4 หมายเหตุ: ถัวเฉลี่ยทุกรายการ

มหาวิทยาลัยนเรศวร

บทที่ 2

หลักการและทฤษฎีเบื้องต้น

ในบทนี้จะรวมหลักการและทฤษฎีขององค์ประกอบที่มีความจำเป็นต่อการทำงานของสวิตช์สัมผัสและระบบไร้สาย ซึ่งในแต่ละองค์ประกอบนั้นจะมีการทำงานที่สัมพันธ์กันของตัวเซนเซอร์ (Sensor) ระบบการควบคุม (Control System) และระบบเครือข่ายไร้สาย (Wireless Network)

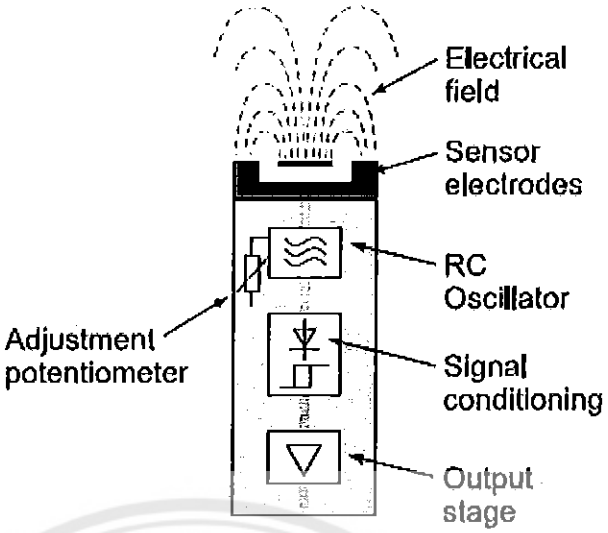
2.1 พร็อกซิมิตี้เซนเซอร์ (Proximity Sensor)

ประเภทของพร็อกซิมิตี้เซนเซอร์

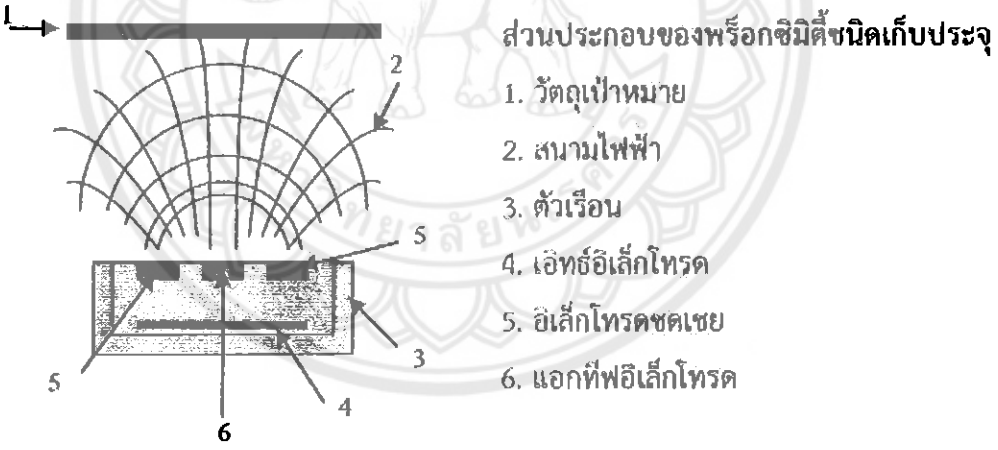
2.1.1 เซนเซอร์แบบเหนี่ยวนำ (Inductive Proximity Sensor) เป็นเซนเซอร์ที่ทำงานโดยอาศัยหลักการเปลี่ยนแปลงค่าความเหนี่ยวนำของขดลวดซึ่งการเปลี่ยนแปลงดังกล่าวจะมีผลต่อชิ้นงานหรือวัตถุที่เป็นโลหะเท่านั้นหรือเรียกกันทางภาษาเทคนิคว่า " อินดั๊กตีฟเซนเซอร์"

2.1.2 เซนเซอร์ชนิดเก็บประจุ (Capacitive Proximity Sensor) เซนเซอร์ประเภทนี้มีโครงสร้างทั้งภายนอกและภายในคล้ายกับแบบเหนี่ยวนำ การเปลี่ยนแปลงของค่าความจุ ซึ่งเนื่องมาจากการเคลื่อนที่ของวัตถุชนิดหนึ่งเข้ามาใกล้สนามไฟฟ้าสถิตย์ของคาปาซิเตอร์ เซนเซอร์ชนิดนี้สามารถตรวจจับอุปกรณ์ที่ไม่เป็นโลหะได้และเป็นโลหะได้ซึ่งในโครงงานนี้จะใช้เซนเซอร์ชนิดเก็บประจุ (Capacitive Proximity Sensor)

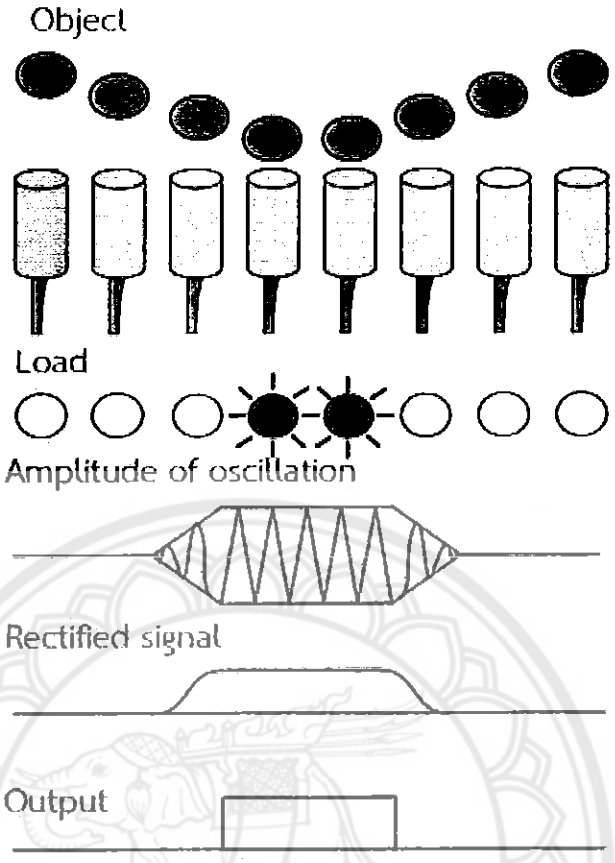
พร็อกซิมิตี้เซนเซอร์ชนิดเก็บประจุจะทำงานโดยอาศัยหลักการเปลี่ยนแปลงค่าความจุ เมื่อวัตถุเป้าหมายเคลื่อนที่เข้ามาใกล้สนามไฟฟ้าที่กำเนิดโดยแอกทีฟอิเล็กโทรดและเอิทธ์อิเล็กโทรด การเปลี่ยนแปลงดังกล่าวขึ้นอยู่กับระยะทางระหว่างหน้าพร็อกซิมิตี้และวัตถุเป้าหมาย ขนาดและรูปร่างของวัตถุ และชนิดของวัตถุเป้าหมาย (ค่าคงที่ไดอิเล็กตริก) เมื่อค่าความจุเปลี่ยนแปลงจนถึงค่า ๆ หนึ่ง ซึ่งเท่ากับค่าความต้านทานที่ปรับไว้ในตอนเริ่มต้นจะส่งผลให้เกิดการออสซิลเลทสัญญาณขึ้นและส่งต่อให้เอาต์พุตทำงาน



รูปที่ 2.1 ส่วนประกอบของ Capacitive Proximity Sensor



รูปที่ 2.2 ส่วนประกอบและการกระจายสนามไฟฟ้าสถิตของ Capacitive Proximity Sensor



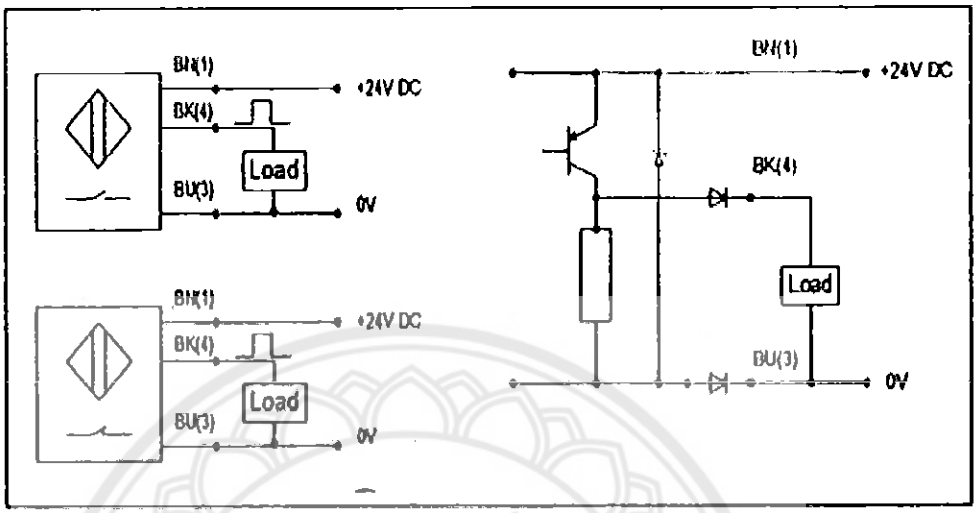
รูปที่ 2.3 หลักการทำงานของ Capacitive Proximity Sensor

ตัวตรวจจับแบบใช้หลักการความจุจะมีโพเทนซิโอมิเตอร์ (Potentiometer) สำหรับปรับความไวของระยะการตรวจจับอยู่ด้านท้ายตรงข้ามกับด้านส่วนตรวจจับ ซึ่งจะทำให้สามารถปรับเลือกให้ไม่ตรวจจับวัตถุที่ขวางกั้นอยู่ก่อนวัตถุที่ต้องการตรวจจับ ตัวอย่าง เช่นการตรวจจับน้ำที่อยู่ในภาชนะบรรจุหรือตรวจจับขวดในกล่องกระดาษ เป็นต้น ซึ่งตัวตรวจจับสามารถปรับไม่ให้ตรวจจับภาชนะบรรจุหรือกล่องกระดาษได้ง่ายมาก

ระยะการตรวจจับของพร็อกซิมิตี้ชนิดเก็บประจุขึ้นอยู่กับระยะห่างระหว่างตัวพร็อกซิมิตี้กับวัตถุและชนิดของวัตถุที่ต้องการตรวจจับ โดยวัตถุที่มีค่าคงที่ไดอิเล็กตริก (Dielectric Constant) สูงจะถูกตรวจจับได้ดีกว่าวัตถุที่มีค่าคงที่ไดอิเล็กตริกต่ำ ในกรณีที่วัตถุเป้าหมายเป็นโลหะระยะการตรวจจับจะเท่ากันหมดไม่ว่าจะเป็นโลหะชนิดใดก็ตาม [1]

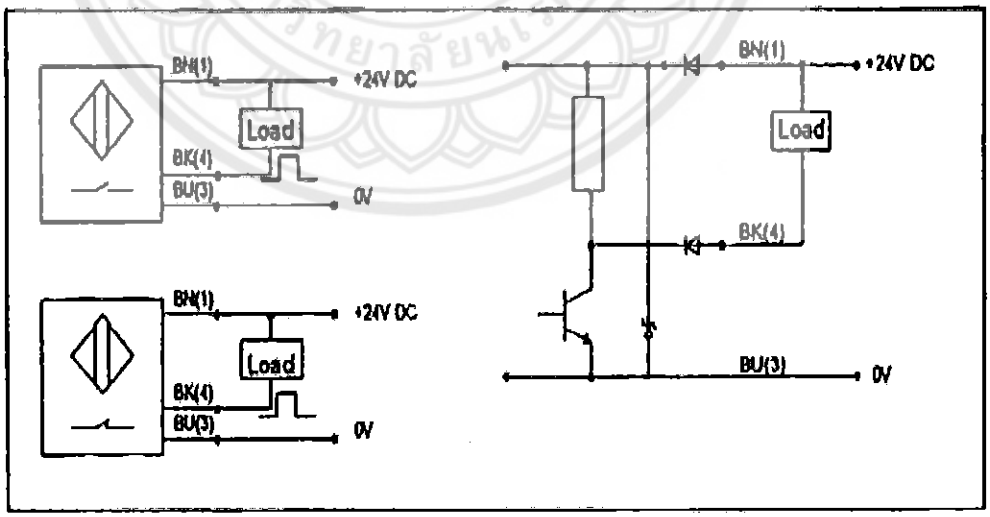
ซึ่งเซนเซอร์ที่ใช้จะเป็นเซนเซอร์แบบ 3 สาย จะมีสายไฟกระแสตรง 5V สายกราวด์ และสายเอาท์พุต ซึ่งเซนเซอร์แบบ3สายนั้นจะมีการทำงานเหมือนของทรานซิสเตอร์ดังนั้นเซนเซอร์ประเภทนี้จะมีการทำงาน 2 แบบคือ PNP และ NPN

แบบ PNP มีชื่อเรียกมากมายหลายแบบ เช่น Sensor แบบ ActiveHigh, Sensor แบบ Source ซึ่ง Sensor แบบนี้เมื่ออยู่ในสภาวะทำงาน จะทำการต่อสายสัญญาณเข้ากับสายไฟ ดังนั้นเมื่อวัดสายสัญญาณเทียบกับกราวด์ เราจะได้ความต่างศักดิ์เท่ากับไฟที่ให้กับ Sensor



รูปที่ 2.4 การต่อใช้งานเซนเซอร์แบบ PNP

แบบ NPN มีชื่อเรียกมากมายหลายแบบ เช่น Sensor แบบ ActiveLow, Sensor แบบ Sink ซึ่ง Sensor แบบนี้เมื่ออยู่ในสภาวะทำงาน จะทำการเชื่อมต่อสายสัญญาณเข้ากับกราวด์ ซึ่งเป็นที่มาของคำว่า ActiveLow คือเมื่อทำงานจะวัดไฟที่จุดนี้ได้เท่ากับกราวด์คือ 0V [2]



รูปที่ 2.5 การต่อใช้งานเซนเซอร์แบบ NPN

2.2 MicrocontrollerPIC16F877

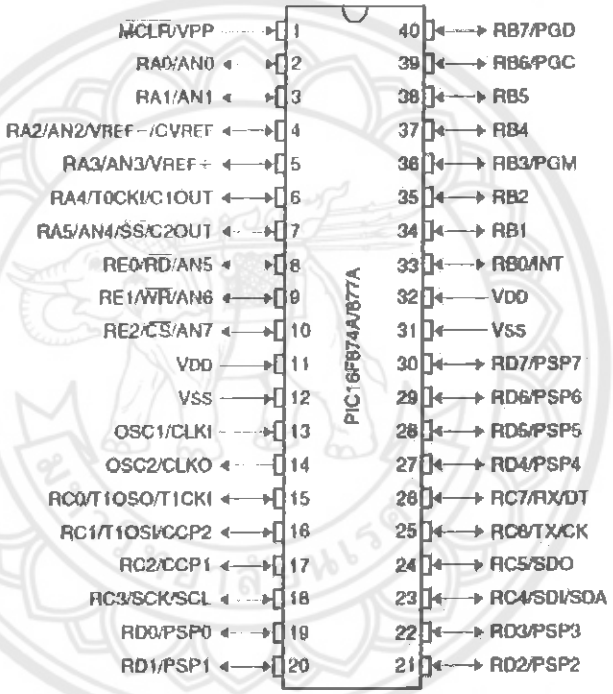
โดยปกติการขยายหน่วยความจำภายนอก PIC ไมโครคอนโทรลเลอร์เป็นเรื่องยากและหาอุปกรณ์ไม่ค่อยมี ในการนำไปใช้โครงงานที่มีขนาดกลางหรือขนาดใหญ่ PIC ตระกูลนี้จึงถูกออกแบบให้มีขนาดของหน่วยความจำข้อมูลแรมและขนาดของหน่วยความจำโปรแกรมเพิ่มขึ้น รองรับการเขียนโปรแกรมได้หลากหลายภาษา ภาษาแอสแซมบลี MPLAB, ภาษาซี (CCS,Hitech,C30, MicroC), ภาษาเบสิก (PICBASICPRO,MicroBasic), ภาษาโลโก้ และการเขียน FlowCode เป็นที่ทราบแล้วว่าไมโครคอนโทรลเลอร์ตระกูล 16Fxxx ได้รับความนิยมอย่างมากเพราะเป็นไมโครคอนโทรลเลอร์ที่มีหน่วยความจำโปรแกรมแบบแฟลช สามารถลบและเขียนคำสั่งแล้วทำการโปรแกรมเข้าใหม่ได้ง่าย มีหน่วยความจำ EEPROM ไว้ภายใน และมีหลายขนาดให้เลือกตามขนาดของงาน

2.2.1 คุณสมบัติทางเทคนิคของ PIC16F877

ที่จะศึกษาเรียนรู้ต่อไปนี้จะใช้ PIC ไมโครคอนโทรลเลอร์เบอร์ 16F877 เพราะหาได้ง่ายในท้องตลาดตลอดจนราคาไม่แพง มีคุณสมบัติเหมาะสำหรับผู้เริ่มต้นศึกษาวิชาไมโครคอนโทรลเลอร์ ภาษาที่ใช้ในการพัฒนาจะใช้ภาษาพิคเบสิกโปร ซึ่งเป็นภาษาที่ง่ายต่อการศึกษาในการเขียนโปรแกรมคำสั่งควบคุมซึ่งมีคุณสมบัติดังนี้

- ซีพียูแบบ RISC มีคำสั่งใช้งานเพียง35คำสั่ง
- ใน1คำสั่งใช้สัญญาณนาฬิกาเพียง 1 ลูก หรือ 2 ลูก
- ความถี่ของสัญญาณนาฬิกา ตั้งแต่ไฟตรงถึง 20 MHz
- ขนาดหน่วยความจำโปรแกรมแบบแฟลช มีขนาด 8 กิโลเวิร์ด (1 เวิร์ด = 14 บิต)
- หน่วยความจำข้อมูลแรม ขนาด 368 ไบต์
- มีหน่วยความจำข้อมูลอีอีพรอม ขนาด 128 ไบต์
- ตอบสนองการอินเตอร์รัปต์ได้ 14 แหล่ง
- มีสแต็ก 8 ระดับ
- มีวงจรเพาเวอร์ออนรีเซต (POR:PowerOnReset),เพาเวอร์อัปไทเมอร์ (PWRT:PowerUp Timer),และออสซิลเลเตอร์สตาร์ตอัปไทเมอร์ (OST:OscillatorStartUp)
- มีวอตช์ด็อกไทเมอร์ (WatchdogTimer)
- มีระบบป้องกันการคัดลอก มีโหมดประหยัดพลังงาน (SleepMode)
- เลือกสัญญาณนาฬิกาได้ 6 โหมดหลักคือ EC,ER,INTRC,LP,XT,HS
- สามารถโปรแกรมด้วยไฟ +5V ได้
- สามารถโปรแกรมในวงจรได้ (InCircuitSerialProgramming)
- ทำงานที่ไฟเลี้ยง +3V ถึง +5.5V

- กระแสซิงก์และซอร์สของพอร์ต 25 mA.
- มีไทเมอร์3 ตัว (ไทเมอร์ 0,8 บิต,ไทเมอร์ 1,16 บิตและไทเมอร์ 2,8 บิต)
- มีโมดูล CCP (Capture/Comparator/PWM:PulseWidthModulation) 2 ชุด
- มีวงจรแปลงสัญญาณอะนาลอกเป็นดิจิตอล (A/DConverter) ขนาด 10 บิต
- มีโมดูลสร้างแรงดันอ้างอิง
- มีโมดูลสื่อสารข้อมูลอนุกรม USARTแบบ RS-232
- มีวงจรตรวจจับระดับแรงดันไฟเลี้ยง (Brown Out Reset)
- มี I/O พอร์ตจำนวน 5 พอร์ต



รูปที่ 2.6 แสดงโครงสร้างการจัดขาของไมโครคอนโทรลเลอร์ PIC16F877

2.2.2 หน้าที่ขาสัญญาณต่างๆ ของ PIC16F877

ไมโครคอนโทรเลอร์ PIC เบอร์ 16F877 มีขาจำนวน 40 ขา หน้าที่ของขาต่างๆ ดังนี้

2.2.2.1 MCLR/VPP (MasterClearReset/ProgrammingVoltageInput) ทำหน้าที่เป็น

- ขารีเซตหลัก (Reset) เมื่อขานี้ได้รับสถานะลอจิก "0" ไมโครคอนโทรลเลอร์จะถูกรีเซตขานี้ ปกติจะมีสถานะลอจิก "1" ถูกต่อเข้าไฟเลี้ยงผ่านรีซิสเตอร์
- ขาอินพุตรับแรงดันสูงสำหรับการโปรแกรม

2.2.2.2 VDD ขาต่อไฟเลี้ยงบวก +3V ถึง +5.5V

2.2.2.3 VSS ต่อลงกราวด์

2.2.2.4 OSC1/CLKIN (OscillatorCrystal/ExternalClockSource)

- ขาอินพุตสัญญาณนาฬิกาหลักเมื่อทำงานในโหมด EC

- ต่อตัวต้านทานเพื่อกำหนดค่าความถี่ในโหมด ER

- ขาต่อคริสตอล เมื่อทำงานในโหมด LP,XTและHS

2.2.2.5 OSC2/CLKOUT (OscillatorCrystal/ExternalClockSource)

- เอาต์พุตสัญญาณนาฬิกาหลักเมื่อทำงานในโหมด EC ความถี่เท่ากับ ¼ ของความถี่ที่ขา OSC1

- ขาต่อคริสตอล เมื่อทำงานในโหมด LP,XT และ HS

2.2.2.6 ขาพอร์ตมีจำนวน 5 พอร์ต คือ พอร์ต A,พอร์ต B,พอร์ต C,พอร์ต D และพอร์ต E พอร์ตของไมโครคอนโทรลเลอร์ PIC16F877 เป็นได้ทั้งอินพุตพอร์ตและเอาต์พุตพอร์ต เป็นพอร์ตแบบสองทิศทาง (BiDirectionalInputOutputPort) ใช้ในการรับและส่งข้อมูลนอกจากนี้ยังมีหน้าที่อื่นๆ ดังตารางที่ 2.1–2.6

ตารางที่ 2.1 หน้าที่ของขาสัญญาณของพอร์ต A ของไมโครคอนโทรลเลอร์ PIC16F877

พอร์ต	ตำแหน่งขา	ชนิดของขา	รายละเอียดการทำงาน
RA0/AN0	2	อินพุต/เอาต์พุต	- ขาพอร์ต RA0 - อินพุตวงจรเปรียบเทียบแรงดันอะนาลอกช่อง 0
RA1/AN1	3	อินพุต/เอาต์พุต	- ขาพอร์ต RA1 - อินพุตวงจรเปรียบเทียบแรงดันอะนาลอกช่อง 1
RA2/AN2/VREF-	4	อินพุต/เอาต์พุต	- ขาพอร์ต RA2 - อินพุตวงจรเปรียบเทียบแรงดันอะนาลอกช่อง 2 - เอาต์พุตแรงดันอ้างอิงลบ
RA3/AN3/VREF+	5	อินพุต/เอาต์พุต	- ขาพอร์ต RA3 - อินพุตวงจรเปรียบเทียบแรงดันอะนาลอกช่อง 3 - เอาต์พุตแรงดันอ้างอิงบวก
RA4/TOCK1	6	อินพุต/เอาต์พุต	- ขาพอร์ต RA4 กรณีใช้งานเป็นพอร์ตเอาต์พุต - อินพุตสัญญาณนาฬิกาของไทเมอร์0
RA5/AN4/SS	7	อินพุต/เอาต์พุต	- ขาพอร์ต RA5 - อินพุตวงจรเปรียบเทียบแรงดันอะนาลอกช่อง 4 - อินพุตสัญญาณ SlaveSelectพอร์ตอนุกรม

ตารางที่ 2.2 หน้าที่ของขาสัญญาณของพอร์ต B ของไมโครคอนโทรลเลอร์ PIC16F877

พอร์ต	ตำแหน่งขา	ชนิดของขา	รายละเอียดการทำงาน
RB0/INT	33	อินพุต/เอาต์พุต	- ขาพอร์ต RB0 - อินพุตวงจรอินเตอร์รัปต์จากภายนอก
RB1	34	อินพุต/เอาต์พุต	- ขาพอร์ต RB1
RB2	35	อินพุต/เอาต์พุต	- ขาพอร์ต RB2
RB3/PGM	36	อินพุต/เอาต์พุต	- ขาพอร์ต RB3 - อินพุตแรงดันต่ำในการบันทึกโปรแกรม
RB4	37	อินพุต/เอาต์พุต	- ขาพอร์ต RB4
RB5	38	อินพุต/เอาต์พุต	- ขาพอร์ต RB5
RB6/PGC	39	อินพุต/เอาต์พุต	- ขาพอร์ต RB6 - ขาสัญญาณนาฬิกาของโปรแกรม
RB7/PGD	40	อินพุต/เอาต์พุต	- ขาพอร์ต RB7 - ขาสัญญาณข้อมูลของโปรแกรม

ตารางที่ 2.3 หน้าที่ของขาสัญญาณของพอร์ต C ของไมโครคอนโทรลเลอร์ PIC16F877

พอร์ต	ตำแหน่งขา	ชนิดของขา	รายละเอียดการทำงาน
RC0/T1OSO/ T1CKI	15	อินพุต/เอาต์พุต	- ขาพอร์ต RC0 - ขาสัญญาณเอาต์พุตของวงจรออสซิเลเตอร์ของไทเมอร์1 - ขาสัญญาณอินพุตของสัญญาณนาฬิกาไทเมอร์1
RC1/T1OSI/ CCP2	16	อินพุต/เอาต์พุต	- ขาพอร์ต RC1 - ขาสัญญาณอินพุตของวงจรออสซิเลเตอร์ไทเมอร์1 - ขาสัญญาณเอาต์พุตของโมดูล CCP2
RC2/CCP1	17	อินพุต/เอาต์พุต	- ขาพอร์ต RC2 - ขาสัญญาณเอาต์พุตของโมดูล CCP1
RC3/SCK/SCL	18	อินพุต/เอาต์พุต	- ขาพอร์ต RC3 - ขาสัญญาณนาฬิกาของวงจร SPI - ขาสัญญาณนาฬิกาของวงจร I^2C

ตารางที่ 2.4 หน้าที่ของขาสัญญาณของพอร์ต C ของไมโครคอนโทรลเลอร์ PIC16F877 (ต่อ)

RC4/SDI/SDA	23	อินพุต/เอาต์พุต	- ขาพอร์ต RC4 - ขาสัญญาณอินพุต SerialDataวงจร SPI - ขาข้อมูลของวงจร I^2C
RC5/SDO	24	อินพุต/เอาต์พุต	- ขาพอร์ต RC5 - ขาสัญญาณเอาต์พุต SerialDataวงจร SPI
RC6/TX/CK	25	อินพุต/เอาต์พุต	- ขาพอร์ต RC6 - ขาส่งข้อมูลแบบพอร์ตอนุกรม - ขาสัญญาณนาฬิกาแบบซิงค์โคนัส
RC7/RX/DT	26	อินพุต/เอาต์พุต	- ขาพอร์ต RC7 - ขารับข้อมูลแบบพอร์ตอนุกรม - ข้อมูลแบบซิงค์โคนัส

ตารางที่ 2.5 หน้าที่ของขาสัญญาณของพอร์ต D ของไมโครคอนโทรลเลอร์ PIC16F877

พอร์ต	ตำแหน่งขา	ชนิดของขา	รายละเอียดการทำงาน
RD0/PSP0	19	อินพุต/เอาต์พุต	- ขาพอร์ต RD0 - ขาขยายสัญญาณพอร์ตแบบขนาน บิต 0
RD1/PSP1	20	อินพุต/เอาต์พุต	- ขาพอร์ต RD1 - ขาขยายสัญญาณพอร์ตแบบขนาน บิต 1
RD2/PSP2	21	อินพุต/เอาต์พุต	- ขาพอร์ต RD2 - ขาขยายสัญญาณพอร์ตแบบขนาน บิต 2
RD3/PSP3	22	อินพุต/เอาต์พุต	- ขาพอร์ต RD3 - ขาขยายสัญญาณพอร์ตแบบขนาน บิต 3
RD4/PSP4	27	อินพุต/เอาต์พุต	- ขาพอร์ต RD4 - ขาขยายสัญญาณพอร์ตแบบขนาน บิต 4
RD5/PSP5	28	อินพุต/เอาต์พุต	- ขาพอร์ต RD5 - ขาขยายสัญญาณพอร์ตแบบขนาน บิต 5
RD6/PSP6	29	อินพุต/เอาต์พุต	- ขาพอร์ต RD6 - ขาขยายสัญญาณพอร์ตแบบขนาน บิต 6
RD7/PSP7	30	อินพุต/เอาต์พุต	- ขาพอร์ต RD7 - ขาขยายสัญญาณพอร์ตแบบขนานบิต 7

ตารางที่ 2.6 หน้าที่ของขาสัญญาณของพอร์ต E ของไมโครคอนโทรลเลอร์ PIC16F877

พอร์ต	ตำแหน่งขา	ชนิดของขา	รายละเอียดการทำงาน
RE0/AN5/RD	8	อินพุต/เอาต์พุต	- ขาพอร์ต RE0 - อินพุตวงจรเปรียบเทียบแรงดันอะนาลอกช่อง 5 - ขาขยายสัญญาณพอร์ตแบบขนานควบคุมการอ่าน
RE1/AN6/WR	9	อินพุต/เอาต์พุต	- ขาพอร์ต RE1 - อินพุตวงจรเปรียบเทียบแรงดันอะนาลอกช่อง 6 - ขาขยายสัญญาณพอร์ตแบบขนานควบคุมการเขียน
RE2/ AN7/CS	10	อินพุต/เอาต์พุต	- ขาพอร์ต RE2 - อินพุตวงจรเปรียบเทียบแรงดันอะนาลอกช่อง 7 - ขาขยายสัญญาณพอร์ตแบบขนานควบคุมการเลือกอุปกรณ์

2.2.3 การออกแบบและเลือกใช้แหล่งกำเนิดสัญญาณนาฬิกา

ไมโครคอนโทรลเลอร์จะต้องมีสัญญาณนาฬิกาคอยควบคุมจังหวะการทำงาน ซึ่งสามารถเลือกใช้ออสซิลเลเตอร์ภายในหรือภายนอกได้ สำหรับออสซิลเลเตอร์ภายในจะใช้ RC ออสซิลเลเตอร์ที่ความถี่คงที่ 4MHz ที่แรงดันไฟเลี้ยง 5 โวลต์

2.2.3.1 อุปกรณ์ที่ใช้สร้างสัญญาณนาฬิกา

อุปกรณ์ที่ใช้เป็นวงจรสร้างสัญญาณนาฬิกาหรือวงจรออสซิลเลเตอร์ภายนอกมี3แบบคือ

- เซรามิกเรโซเนเตอร์ (CeramicResonator) เหมาะใช้งานกับความถี่ไม่สูงมาก มีราคาถูก การต่อใช้งานขากลางต่อกราวด์ ขาที่เหลือต่อกับขา CLKIN และ CLKOUT ของ PIC

- คริสตอลออสซิลเลเตอร์ (CrystalOscillator) ความถี่สัญญาณนาฬิกามีความเที่ยงตรง วงจรการใช้งานต่อกับคาปาซิเตอร์ 2 ตัว กับขา OSC1/CLKIN และ OSC2/CLKOUT

- วงจรกำเนิดความถี่สำเร็จรูป (CrystalSquareWaveOscillator) จะมีคริสตอลและวงจรบรรจุไว้ภายในตัวคริสตอลออสซิลเลเตอร์ ทำให้ได้ความถี่ที่มีความเที่ยงตรงมาก

2.2.3.2 วงจรกำเนิดสัญญาณนาฬิกาให้ไมโครคอนโทรลเลอร์

วงจรออสซิลเลเตอร์ให้ไมโครคอนโทรลเลอร์ PIC โดยการต่อที่ขา OSC1 และขา OSC2 สามารถกำหนดเลือกใช้แหล่งสัญญาณนาฬิกาได้ 4 วิธี ดังนี้

- โหมด LP(LowPowerCrystalOscillator)ใช้คริสตอลกำลังงานต่ำ ความถี่สูงสุด ไม่เกิน 200KHz

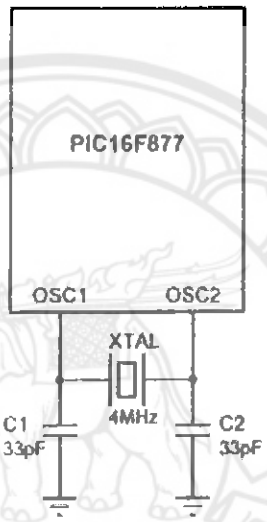
- โหมด XT (Crystal/Resonator) ใช้คริสตอลหรือเซรามิกเรโซเนเตอร์ ความถี่ตั้งแต่ 100KHz – 4MHz

- โหมด HS (HighSpeedCrystal/Resonator) ใช้คริสตอลความถี่สูง 4MHz--20MHz

- โหมด RC (ResistorCapacitorNetwork) สามารถกำหนดความถี่จากตัวต้านทานและตัวเก็บประจุร่วมกับแหล่งกำเนิดสัญญาณนาฬิกาภายในตัวไมโครคอนโทรลเลอร์เอง

1. วงจรออสซิลเลเตอร์ใช้คริสตอลออสซิลเลเตอร์

วงจรต่อใช้งานดังรูปที่ 2.7 ค่าคาปาซิเตอร์เลือกใช้ตรงความถี่ที่ต้องการตามตารางที่ 2.7



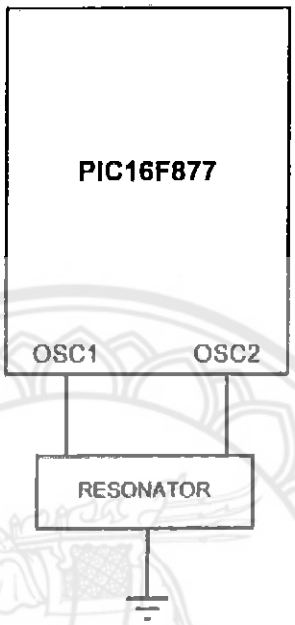
รูปที่ 2.7 วงจรออสซิลเลเตอร์ใช้คริสตอลออสซิลเลเตอร์

ตารางที่ 2.7 การเลือกค่า C1 และ C2 ตามโหมด ค่าความถี่ของคริสตอล

โหมด OSC.	ความถี่	C1	C2
LP	32KHz	68–100pF	68–100pF
	200KHz	15–33pF	15–33pF
XT	100KHz	100–150pF	100–150pF
	2MHz	15–33pF	15–33pF
	4MHz	15–33pF	15–33pF
HS	4MHz	15–33pF	15–33pF
	8MHz	15–33pF	15–33pF
	20MHz	15–33pF	15–33pF

2. วงจรออสซิลเลเตอร์ใช้เซรามิกเรโซเนเตอร์

วงจรออสซิเลเตอร์ต่อใช้งานดังรูปที่ 2.8 ค่าคาปาซิเตอร์เลือกใช้ตรงความถี่ที่ต้องการตามตารางที่ 2.8



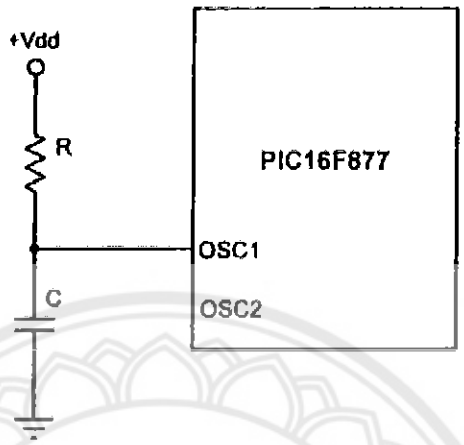
รูปที่ 2.8 วงจรออสซิลเลเตอร์ใช้เซรามิกเรโซเนเตอร์

ตารางที่ 2.8 การเลือกค่า C1 และ C2 ตามค่าความถี่ของเซรามิกเรโซเนเตอร์

โหมด OSC.	ความถี่	C1	C2
XT	455KHz	68–100 pF	68–100pF
	2MHz	15– 68pF	15–68pF
	4MHz	15–68pF	15–68pF
HS	8MHz	15–68pF	15–68pF
	16MHz	10 –22pF	10–22pF

3. วงจรออสซิลเลเตอร์ใช้ตัวต้านทานและตัวเก็บประจุ

วงจรต่อใช้งานดังรูปที่ 2.9 ค่าตัวต้านทานและคาปาซิเตอร์เลือกใช้ตรงความถี่ที่ต้องการตามตารางที่ 2.9



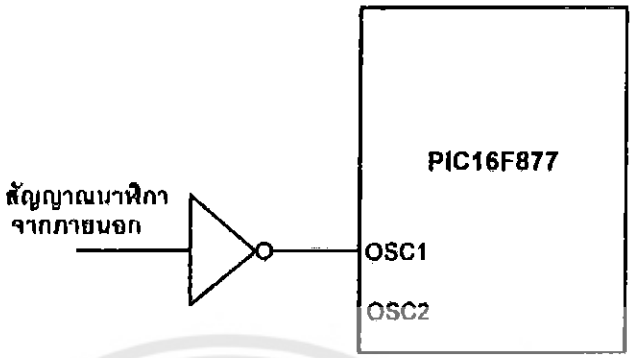
รูปที่ 2.9 วงจรออสซิลเลเตอร์ใช้ตัวต้านทานและตัวเก็บประจุ

ตารางที่ 2.9 การเลือกค่า R และ C ค่าความถี่ออสซิลเลเตอร์

ความถี่	R	C
4.61MHz	5K	20pF
2.66MHz	10K	20pF
311kHz	100K	20pF

4. วงจรออสซิลเลเตอร์จากแหล่งภายนอก

วงจรต่อใช้งานดังรูปที่ 2.8



รูปที่ 2.10 วงจรออสซิลเลเตอร์จากแหล่งภายนอก

2.2.4 การรีเซตไมโครคอนโทรลเลอร์ PIC

การรีเซตไมโครคอนโทรลเลอร์ PIC เป็นการทำให้ไมโครคอนโทรลเลอร์เริ่มต้นทำงานใหม่ โดยจะเริ่มทำคำสั่งแรกของโปรแกรมการรีเซตของ PIC มี 6 แบบ คือ

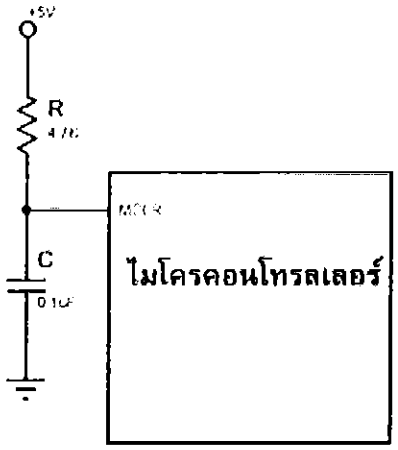
- เพาเวอร์ออนรีเซต (POR: PowerOnReset)
- บราวด์เอาต์รีเซต (BOR) เมื่อไฟเลี้ยงตกหรือต่ำกว่า 4.0V หรือต่ำกว่า 1.8V กรณีที่ให้ PIC ทำงานในโหมดแรงดันต่ำ
- รีเซตที่ขา MCLRในสภาวะปกติ
- รีเซตที่ขา MCLRในโหมดประหยัดพลังงาน (SleepMode)
- รีเซตวอตช์ด็อกไทเมอร์ (WDT)
- รีเซตวอตช์ด็อกไทเมอร์ (WDT)ในโหมดประหยัดพลังงาน

ในการสร้างสัญญาณรีเซต ซึ่งเป็นสัญญาณที่มีความสำคัญมากเมื่อ PIC เริ่มต้นของการทำงานของระบบ วงจรไมโครคอนโทรลเลอร์ต้องอยู่ในสภาพรีเซตเพื่อให้ระบบไฟต่างๆ คงที่อุปกรณ์ต่างๆ ที่ต่อพ่วงพร้อมที่จะทำงานก่อนตัวไมโครคอนโทรลเลอร์จึงจะเริ่มทำงาน

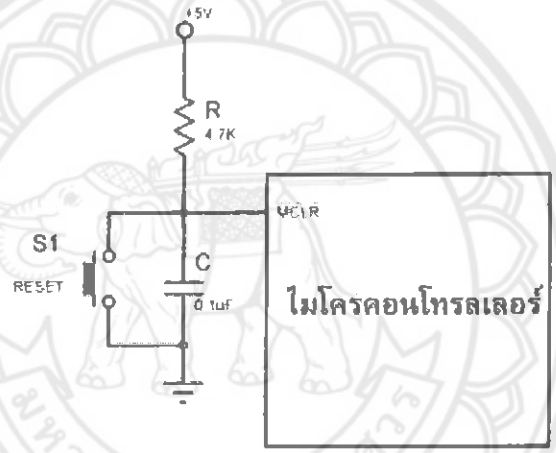
สัญญาณรีเซตของไมโครคอนโทรลเลอร์ ได้มาจากหลายแหล่ง คือ รีเซตขา MCLR ปกติขานี้จะต่อไฟเลี้ยงโดยผ่านตัวต้านทาน 4.7K เมื่อต้องการรีเซต PIC ให้ขานี้มีสภาวะทำงานเป็น ลอจิก “0”

การออกแบบวงจรรีเซตไมโครคอนโทรลเลอร์ PIC โดยทั่วไปจะมี 2 แบบคือ

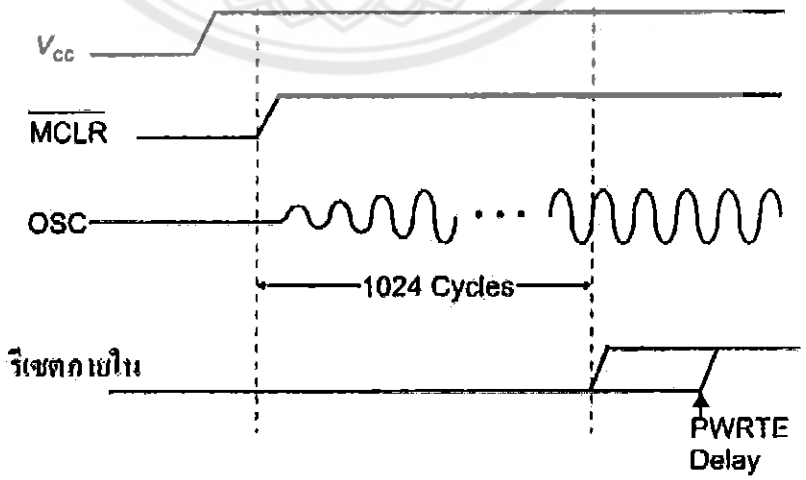
- รีเซตแบบอัตโนมัติ ดังวงจรรูปที่ 2.11
- รีเซตแบบสวิทช์กดและแบบอัตโนมัติ ดังวงจรรูปที่ 2.12



รูปที่ 2.11 วงจรรีเซตแบบอัตโนมัติไมโครคอนโทรลเลอร์ PIC



รูปที่ 2.12 วงจรรีเซตแบบสวิทช์กดและแบบอัตโนมัติไมโครคอนโทรลเลอร์ PIC



รูปที่ 2.13 แสดงรูปคลื่นการเกิดการรีเซตภายในไมโครคอนโทรลเลอร์ช่วงเริ่มทำงาน

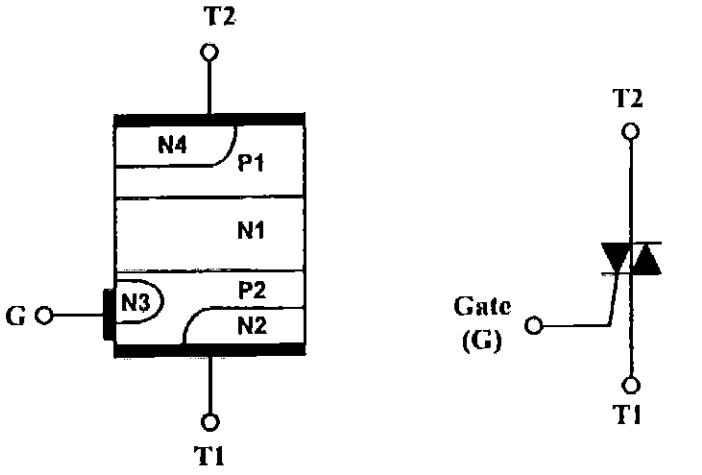
จากรูปที่ 2.13 เมื่อเริ่มจ่ายไฟเลี้ยง Vcc เข้าวงจร ตัว PIC ได้รับไฟเลี้ยง สภาวะขา MCLR จะเป็นลอจิก "0" (เมื่อ C เริ่มเก็บประจุ) จากนั้น MCLR เปลี่ยนสภาวะลอจิก "1" (Cเก็บประจุเต็ม) และสัญญาณนาฬิกา OSC จะถูกสร้างขึ้น มีจำนวนประมาณ 1024 ลูก เกิดสัญญาณรีเซตภายใน (PWRITE Delay) ประมาณ 72 ms. แล้ว PIC จึงจะเริ่มทำงานตามคำสั่งโปรแกรม [3]

2.3 TRIAC

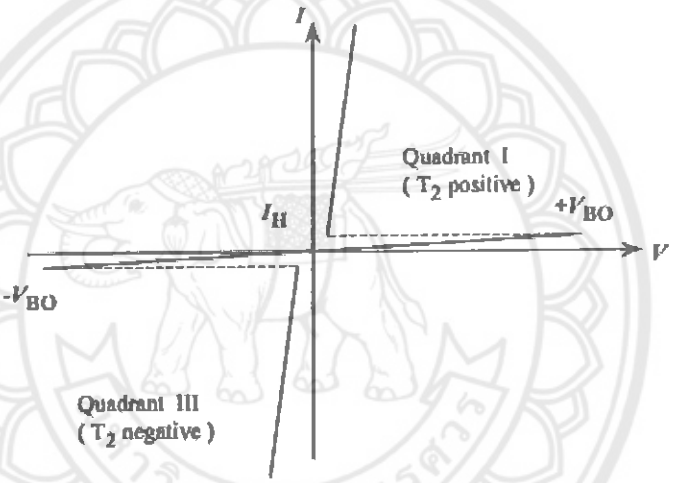
ไตรแอคเป็นสารกึ่งตัวนำประเภทไทรีสเตอร์ชนิดหนึ่งสามารถควบคุมการเปิด / ปิดกระแสไฟฟ้าได้เช่นเดียวกับ SCR แต่ SCR สามารถควบคุมการเปิด / ปิดกระแสไฟฟ้าได้เมื่อขั้วอาโนดแรงดันไฟฟ้าบวก ส่วน TRIAC สามารถควบคุมได้ทั้งแรงดันไฟฟ้าเป็นบวกหรือลบ ดังนั้นขั้วของไตรแอคจึงมีชื่อเรียกเป็น T2,T1 และ G แทนที่จะเป็น แอโนด, แคโทด, และเกท แบบกรณีของ SCR ในการบอกค่าแรงดันไฟฟ้าที่ขั้วต่างๆ ของไตรแอค เราจะถือเอา ขั้ว T1เป็นขั้วอ้างอิงเสมอ เช่น เมื่อบอกว่าแรงดันไฟฟ้าที่ขั้ว T2 เป็นบวก ก็หมายความว่าขั้ว T2 มีศักย์ไฟฟ้าสูงกว่าขั้ว T1 เป็นต้น

ทฤษฎีการทำงานของไตรแอค ไตรแอคเป็นไทรีสเตอร์ซึ่งสามารถปิดกั้นกระแสได้สองทิศทางและสามารถถูกสวิตช์ให้นำกระแสได้ทั้งสองทิศทาง โดยกระแสเกทที่ใช้จุดชนวนจะมีทิศทางเข้าหรือออกจากไตรแอคก็ได้ ดังนั้นการจุดชนวนไตรแอคจึงแบ่งออกได้เป็น 4 แบบคือ

- แบบ I^+ แรงดันไฟฟ้าที่ T2 เป็นบวก และกระแสเกทไหลเข้าไตรแอค
- แบบ I^- แรงดันไฟฟ้าที่ T2 เป็นบวก และกระแสเกทไหลออกจากไตรแอค
- แบบ III^+ แรงดันไฟฟ้าที่ T2 เป็นลบ และกระแสเกทไหลเข้าไตรแอค
- แบบ III^- แรงดันไฟฟ้าที่ T2 เป็นลบ และกระแสเกทไหลผ่านออกจากไตรแอค [4]



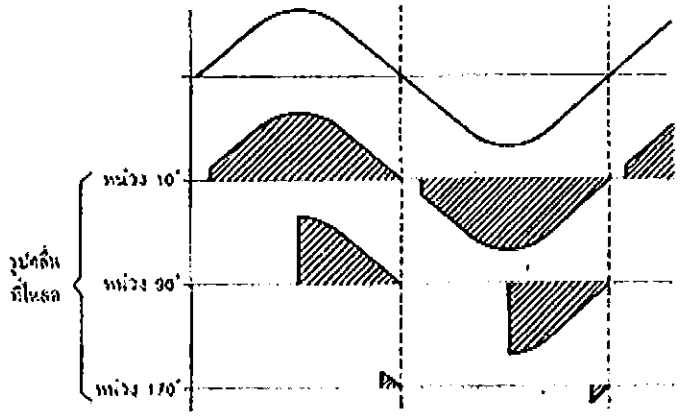
รูปที่ 2.14 โครงสร้างและสัญลักษณ์ของไตรแอค



รูปที่ 2.15 ลักษณะสมบัติกระแส –แรงดันของไตรแอค

2.3.1 การควบคุมกำลังไฟแบบเฟสทริกเกอร์

ไตรแอกที่กล่าวมาตั้งแต่ต้นนี้เป็นการใช้งานในลักษณะเป็นสวิตช์ เปิด / ปิด การจ่ายไฟให้แก่โหลดต่างๆ แต่ความจริงแล้วการใช้งานสามารถขยายออกไปได้อีกมาก เช่นใช้เป็นวงจรหรี่ความสว่างของหลอดไฟหรือเป็นวงจรควบคุมความเร็วของมอเตอร์เป็นต้นซึ่งก็ล้วนแล้วแต่เป็นการใช้งานควบคุมกำลังไฟที่จะจ่ายให้แก่โหลดในระบบที่เรียกว่าเฟส – ทริกเกอร์หลักการของวงจรที่มีลักษณะเป็นเฟส – ทริกเกอร์นี้ใช้ไตรแอกเป็นตัวควบคุมกำลังไฟที่จ่ายให้แก่โหลดโดยแทนที่จะทริกขาเกตด้วยสัญญาณไฟตรงนั้นตรงๆก็ทริกโดยมีการหน่วงของเฟสด้วยวงจรอีกส่วนหนึ่ง



รูปที่ 2.16 การเปลี่ยนแปลงค่าของกำลังไฟฟ้าที่ป้อนให้แก่โหลด โดยกำหนดได้จากตำแหน่งเวลาของการทริกที่ให้แก่ไตรแอก

การหน่วงเฟสมีผลดังนี้คือ ถ้าไตรแอกถูกทริกที่ตำแหน่งเฟส 10 องศาหลังจากที่ทุกๆ ครึ่งรูปคลื่นเริ่มเข้ามากำลังไฟเกือบทั้งหมดก็จะถูกป้อนให้แก่โหลดแต่ถ้าการทริกที่ตำแหน่งเฟส 90 องศาหลังจากทุกๆ ครึ่งคลื่นเริ่มเข้ามา จะทำให้กำลังไฟที่ป้อนให้แก่โหลดนั้นลดลงเหลือเพียงครึ่งหนึ่งของกำลังทั้งหมดและถ้าไปทริกที่ตำแหน่งเฟส 170 องศา หลังจากที่ทุกๆ ครึ่งรูปคลื่นเข้ามาแล้ว จะมีเพียงกำลังไฟส่วนน้อยเท่านั้นที่ป้อนให้แก่โหลด ขอให้ดูรูปที่ 2.16 ประกอบจะเข้าใจได้ยิ่งขึ้น [5]

2.4 RF1100-232 WirelessRF

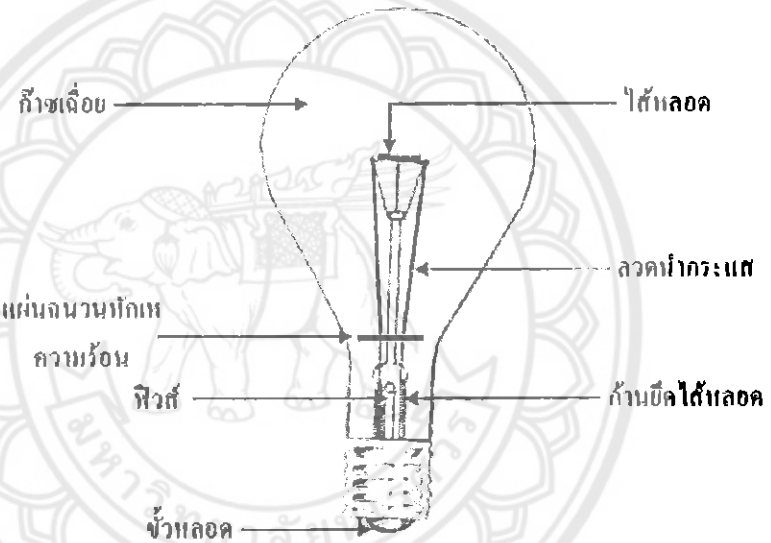
ระบบเครือข่ายไร้สาย (WirelessNetwork) คือ ระบบการสื่อสารข้อมูลที่นำมาใช้ทดแทนหรือเพิ่มกับระบบเครือข่ายแลนใช้สายแบบดั้งเดิม โดยใช้การส่งคลื่นความถี่วิทยุในย่านวิทยุ RF และคลื่นอินฟราเรด ในการรับและส่งข้อมูลนอกจากนี้ระบบเครือข่ายไร้สายจะมีคุณสมบัติครอบคลุมทุกอย่างเหมือนกับระบบแบบใช้สาย และที่สำคัญคือการที่ไม่ต้องใช้สายทำให้การเคลื่อนย้ายการใช้งานทำได้โดยสะดวก

RF1100-232 WirelessRF มีหน้าที่เป็นอุปกรณ์รับส่งข้อมูลส่งออกในรูปแบบไร้สายรับส่งสัญญาณใช่ช่วงความถี่ 433MHz ระยะการส่งในที่โล่ง 15 เมตร อัตราการส่ง 10 มิลลิวัตต์ ทำงานที่แรงดันไฟฟ้ากระแสตรง 5 โวลต์โมดูลที่ช่วยให้การส่งข้อมูลไร้สายที่พอร์ตอนุกรม โดยจะส่งขอมูลไร้สายที่ขา Tx และรับข้อมูลไร้สายที่ขา Rx รองรับอัตราการรับส่งข้อมูล 4800,9600 และ 19200 bit/s สามารถเลือกส่งได้ 256 Channels และส่วนที่น่าสนใจโมดูลนี้สามารถส่งสัญญาณแบบ Point–to-multipoint ได้ หมายความว่าเมื่อโมดูลตัวส่งส่งสัญญาณออกไป โมดูลตัวรับทั้งหมดที่ใช้ Channels เดียวกันก็จะได้รับข้อมูลที่เหมือนกัน และปัญหาที่พบคือโมดูลบางตัวอาจมีการรับส่งค่าผิดพลาดบ้างเป็นครั้งคราว

2.5 ชนิดของหลอดไฟ

2.5.1 หลอดไส้ หรือ หลอดอินแคนเดสเซนท์

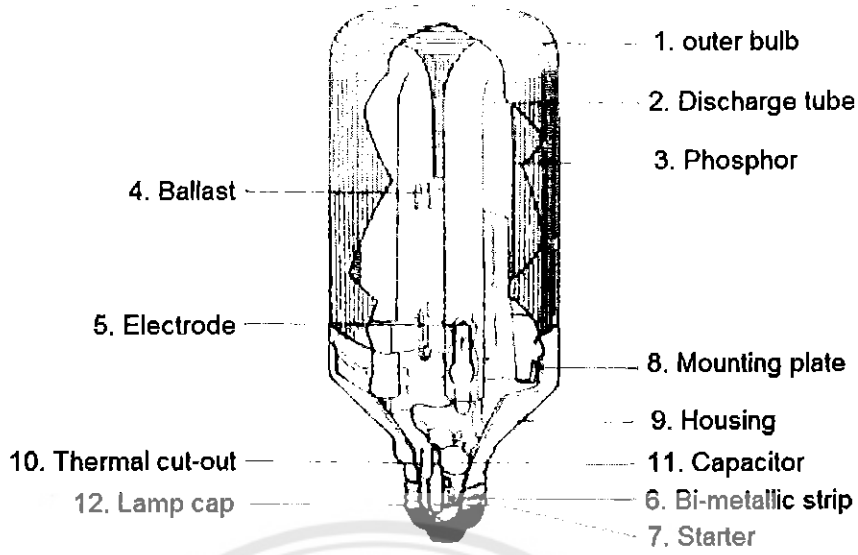
เป็นหลอดไฟที่ใช้กันในยุคแรกๆ บางทีเรียกกันว่าหลอดดวงเทียนเพราะมีแสงแดงๆ เหมือนแสงเทียนมีทั้งชนิดแก้วใส และแก้วฝ้า เมื่อกระแสไฟฟ้าไหลผ่านไส้หลอดจะเกิดความร้อนยิ่งความร้อนมากขึ้นเท่าใดแสงสว่างที่เปล่งออกมาจากไส้หลอดก็จะมากขึ้นหลอดไส้ร้อนแบบธรรมดาผลิตออกมาหลายขนาดกำลังส่องสว่างและอัตราทนความต่างศักย์ตั้งแต่ 1.5โวลต์ ถึงราว 300 โวลต์ หลอดประเภทนี้ไม่ต้องอาศัยอุปกรณ์ควบคุมภายนอก มีค่าบำรุงรักษาต่ำ และทำงานได้ดีเท่ากันทั้งไฟฟ้ากระแสสลับหรือกระแสตรง



รูปที่ 2.17 โครงสร้างของหลอดไส้

2.5.2 หลอดตะเกียบ หรือ หลอดคอมแพคฟลูออเรสเซนต์

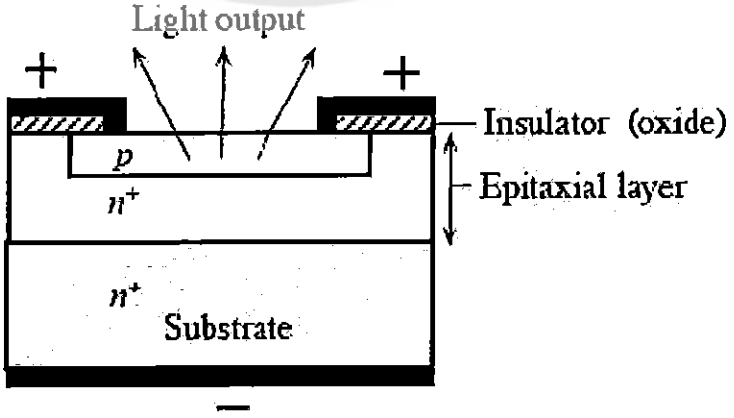
หลอดคอมแพคฟลูออเรสเซนต์ หรือที่รู้จักกันทั่วไปในชื่อของ หลอดตะเกียบถูกพัฒนาขึ้นมาเพื่อใช้แทนหลอดไส้แบบเก่า เพราะหลอดตะเกียบนอกจากจะมีขนาดกะทัดรัดแล้วยังเพิ่มระดับความสว่าง และมีอายุการใช้งานที่มากขึ้น โดยมีอายุการใช้งานเฉลี่ยถึง 7 ปี หรือประมาณ 8,000 ชั่วโมง อีกทั้งยังสามารถประหยัดพลังงานได้มากถึง 4 เท่าของหลอดไฟแบบเก่าด้วย ขนาดของหลอดตะเกียบแบ่งออกเป็น 5 ชนิดด้วยกันคือ 2U 3U 4U 5U และ 6U สามชนิดหลังเหมาะกับโรงงานหรืออาคารเชิงพาณิชย์มากกว่า ส่วนชนิดที่เหมาะกับบ้านเรือนที่พักอาศัยทั่วไปคือ 2U และ 3U



รูปที่ 2.18 โครงสร้างของหลอดคอมแพคฟลูออเรสเซนต์

2.5.3 หลอดไฟ LED

ด้วยคุณสมบัติการทำงานที่ไม่มีการเผาไส้หลอด จึงไม่เกิดความร้อนแสงสว่างเกิดขึ้นจากการเคลื่อนที่ของอิเล็กตรอนภายในสารกึ่งพลังงานเปลี่ยนเป็นแสงสว่างได้เต็มที่มีแสงหลายสีให้เลือกใช้งาน ขนาดที่เล็กทำให้ยืดหยุ่นในการออกแบบ การจัดเรียง นำไปใช้ด้านตกแต่งได้ดีมีความทนทาน ไม่ต้องห่วงเรื่องไส้หลอดขาด หรือหลอดแตก ด้านอายุการใช้งานก็อยู่ได้ถึง 50,000-60,000 ชั่วโมง และที่สำคัญปราศจากปรอท และสารกลุ่มฮาโลเจนที่เป็นพิษ แต่มีข้อเสีย คือในปัจจุบันหลอด LED มีราคาสูงกว่าหลอดธรรมดาทั่วไปและมีความสว่างไม่มากนัก



รูปที่ 2.19 โครงสร้างของหลอดไฟ LED

บทที่ 3
วิธีดำเนินโครงงาน

3.1 ศึกษา และเก็บข้อมูล

3.1.1 ศึกษาการใช้งานฟังก์ชั่นพื้นฐานโปรแกรม Microcontroller PIC16F877 และเครื่องมือและอุปกรณ์ในการปฏิบัติงาน และการเชื่อมต่อระบบไฟฟ้าคณะผู้ดำเนินโครงงานวิศวกรรมได้ทำการศึกษาการเขียนโปรแกรมลงในบอร์ด Microcontroller PIC16F877 เพื่อนำไปใช้ในการเขียนโปรแกรมควบคุมการเปิด – ปิดและปรับระดับความสว่างของหลอดไฟ

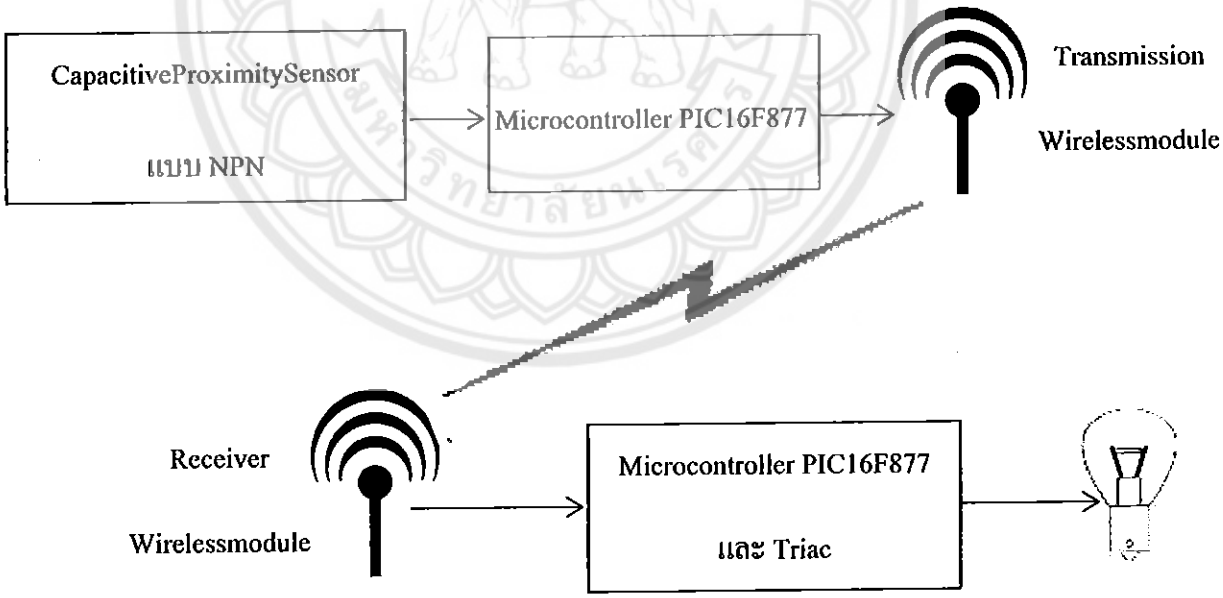
3.1.2 ศึกษาวิธีดำเนินงานจัดทำบอร์ดวงจร และการเขียนโปรแกรม Microcontroller PIC16F877 ในส่วนของรูปอุปกรณ์ผู้ดำเนินโครงงานได้นำไปไว้ในส่วนของภาคผนวก

3.1.2.1 จัดทำบอร์ดด้านตัวส่งสัญญาณ

3.1.2.2 จัดทำบอร์ดด้านตัวรับสัญญาณ

3.1.2.3 เขียนโปรแกรมลงในบอร์ด Microcontroller PIC16F877

3.1.2.4 บันทึกผล และวิเคราะห์ผลการทดลอง



รูปที่ 3.1 แผนภาพแสดงภาพรวมของระบบ

3.2 ออกแบบการทดลอง

ออกแบบขั้นตอน กระบวนการ วิธีการทดลอง

1799137
สำนักหอสมุด
24 ส.ค. 2561

3.3 เขียนโปรแกรม

3.3.1 เขียนโปรแกรมโดยใช้ Microcontroller PIC16F877 ในการควบคุมระบบ

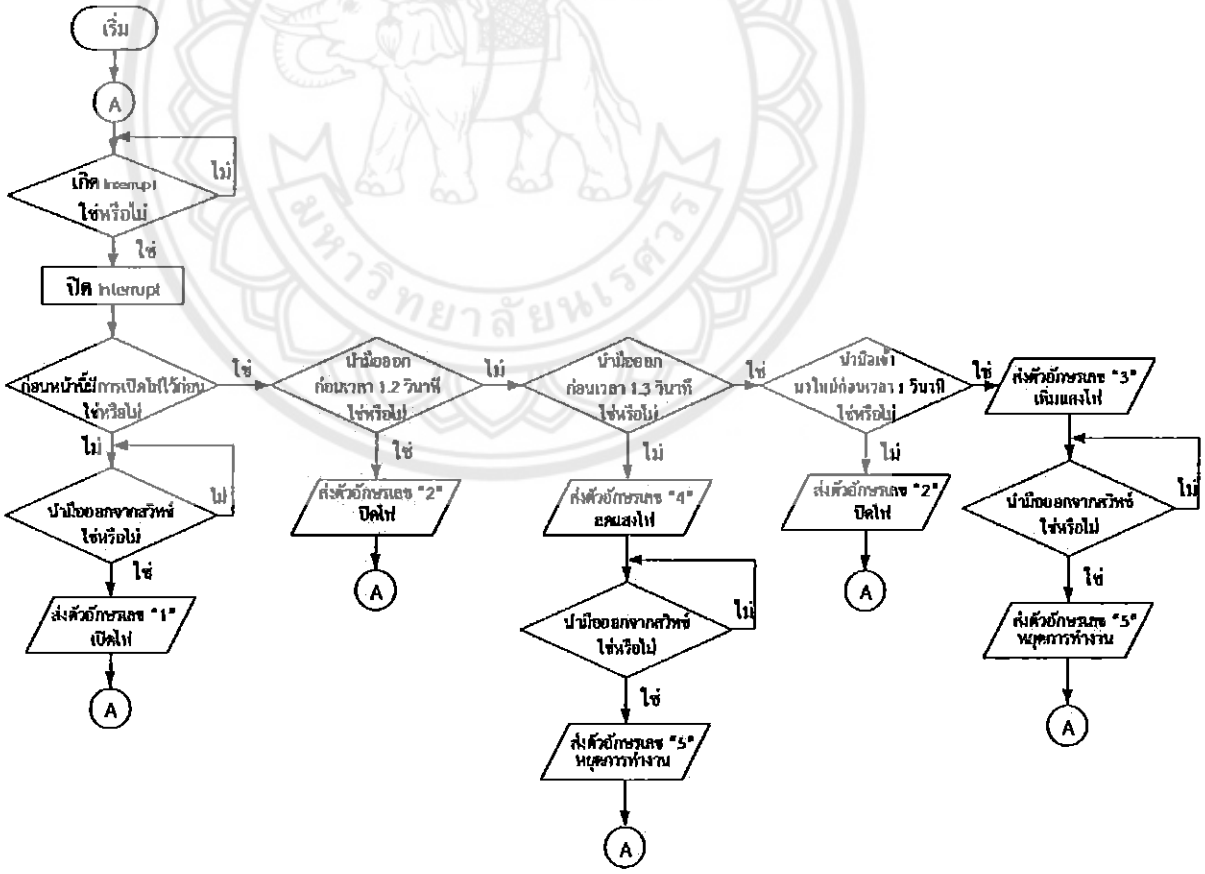
3.3.2 ในส่วนของโค้ดโปรแกรมผู้ดำเนินโครงงานได้นำไปไว้ในส่วนของภาคผนวก

3.4 ทดสอบและปรับปรุง

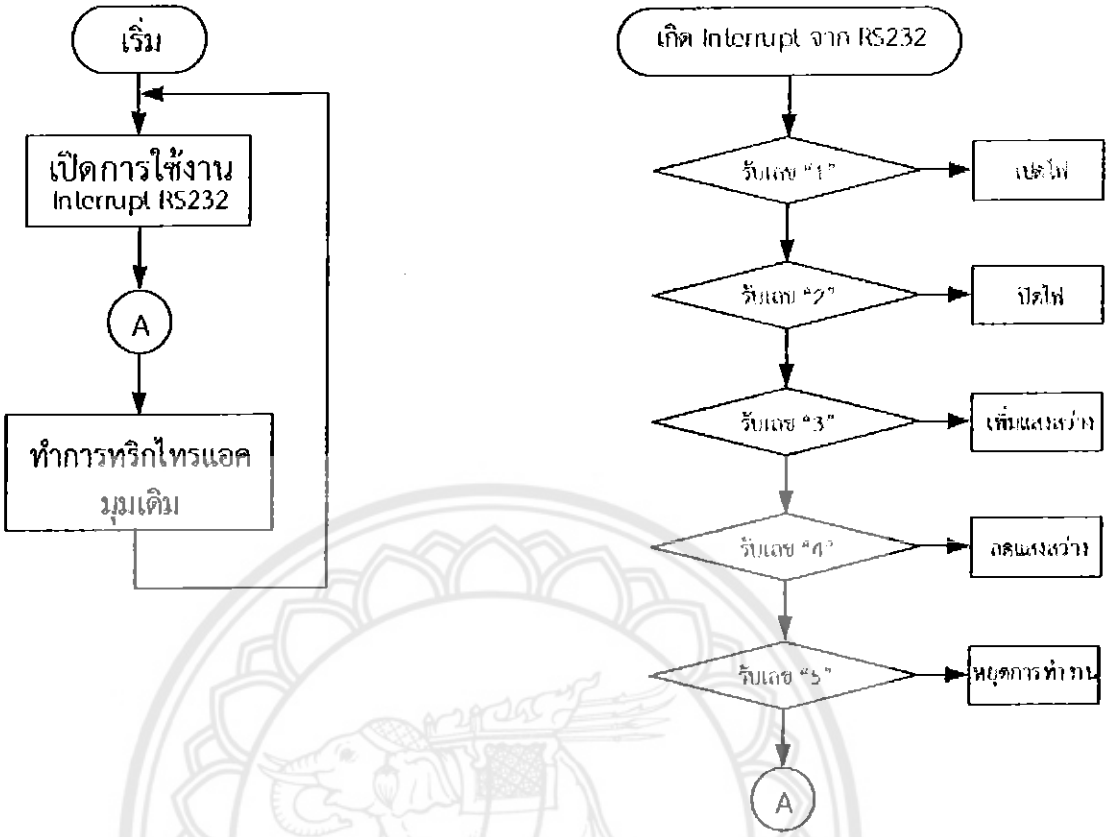
ทดสอบการใช้งานของอุปกรณ์ว่ามีข้อบกพร่องประการใด แล้วนำมาปรับปรุง

3.5 ขั้นตอนการทำงานของระบบ

ขั้นขั้นตอนการทำงานของสวิตซ์สัมผัสไร้สายนั้นจะทำการเปิด – ปิดและควบคุมแสงสว่างโดยการหรี่ซึ่งการทำงานทั้งหมดนั้นจะทำงานตามโปรแกรมที่เขียนไว้กับไมโครคอนโทรลเลอร์จะมีการทำงานดังนี้ทำงานดังนี้



รูปที่ 3.2 โฟลว์ชาร์ตแสดงการทำงานของคอนโทรลเลอร์ฝั่งส่งสัญญาณ



รูปที่ 3.3 โฟลว์ชาร์ตแสดงการทำงานของคอนโทรลเลอร์ฝั่งรับสัญญาณ

บทที่ 4

ผลการทดลอง

ในบทที่ 4 ได้แบ่งการทดลองออกเป็นการทดลองย่อย 3 การทดลอง ได้แก่

1. การทดลองระยะตรวจจับผ่านสิ่งกีดขวางของเซนเซอร์
2. การทดลองโดยใช้หลอดไฟแบบต่างๆ
3. การทดลองระยะการทำงานของระบบไร้สาย

4.1 ผลการทดลองที่ 1 การทดลองระยะตรวจจับผ่านสิ่งกีดขวางของเซนเซอร์

จากที่ทำการทดลองที่ 1 เพื่อหาระยะการตรวจจับผ่านสิ่งกีดขวางของเซนเซอร์ ผลที่ได้แสดงได้ดังตารางที่ 4.1 และตารางที่ 4.2

ตารางที่ 4.1 ตารางแสดงการทำงานระยะการตรวจจับผ่านวัตถุชนิดไม้

การทำงาน	ระยะวัตถุ (ไม้)		
	3 มิลลิเมตร	6 มิลลิเมตร	9 มิลลิเมตร
เปิดไฟ	ทำงาน	ทำงาน	ไม่ทำงาน
หรี่ไฟขึ้น	ทำงาน	ทำงาน	ไม่ทำงาน
หรี่ไฟลง	ทำงาน	ทำงาน	ไม่ทำงาน
ปิดไฟ	ทำงาน	ทำงาน	ไม่ทำงาน

รูปที่ 4.1 ไม้หนา 3 มิลลิเมตร

รูปที่ 4.2 ไม้หนา 6 มิลลิเมตร

รูปที่ 4.3 ไม้หนา 9 มิลลิเมตร

ตารางที่ 4.2 ตารางแสดงการทำงานระยะการตรวจจับผ่านวัตถุชนิดยางสังเคราะห์

การทำงาน	ระยะวัตถุ (ยางสังเคราะห์)		
	3 มิลลิเมตร	6 มิลลิเมตร	9 มิลลิเมตร
เปิดไฟ	ทำงาน	ทำงาน	ไม่ทำงาน
หรี่ไฟขึ้น	ทำงาน	ทำงาน	ไม่ทำงาน
หรี่ไฟลง	ทำงาน	ทำงาน	ไม่ทำงาน
ปิดไฟ	ทำงาน	ทำงาน	ไม่ทำงาน

รูปที่ 4.4 ยางสังเคราะห์หนา 3 มิลลิเมตร

รูปที่ 4.5 ยางสังเคราะห์หนา 6 มิลลิเมตร

รูปที่ 4.6 ยางสังเคราะห์หนา 9 มิลลิเมตร

4.1.1 สรุปผลการทดลองที่ 1

จากผลการทดลองด้านบนจะเห็นว่าทั้ง 2 วัสดุ เซนเซอร์จะมีการตรวจจับผ่านวัสดุที่ขวางได้ 6 มิลลิเมตร และเมื่อใช้วัสดุที่มีความหนามากกว่า 6 มิลลิเมตรขึ้นไป เซนเซอร์จะไม่สามารถทำการตรวจจับและไม่สามารถทำงานได้ซึ่งระยะตรวจจับนั้นสามารถปรับได้โดยการปรับค่าความต้านทานของโพเทนซิโอมิเตอร์ (Potentiometer) จากรูปที่ 2.1 เมื่อปรับค่าให้น้อยลงจะทำให้การออสซิลเลทในตัวเซนเซอร์ได้ไวขึ้น และพบว่าเซนเซอร์จะทำงานได้ในระยะที่ไกลขึ้นด้วย

4.2 ผลการทดลองที่ 2 การทดลองโดยใช้หลอดไฟแบบต่างๆ

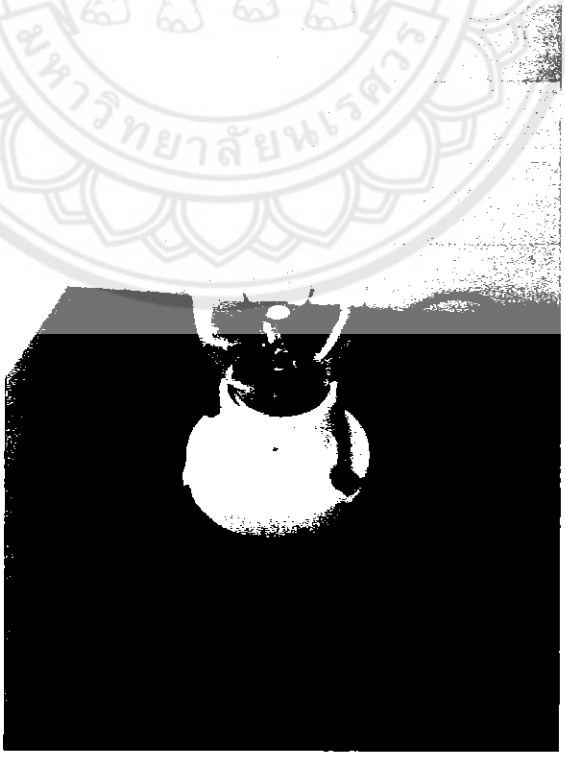
จากที่ทำการทดลองที่ 2 โดยการใช้หลอดไฟชนิดต่างๆ โดยการใช้หลอดไส้ หลอดคอมแพคฟลูออเรสเซนต์ (หลอดตะเกียบ) และหลอด LED 220V เพื่อดูการทำงานของอุปกรณ์ ผลที่ได้แสดงได้ดังตารางที่ 4.3

รูปที่ 4.7 หลอดไฟชนิดต่างๆที่ใช้ในการทดลอง

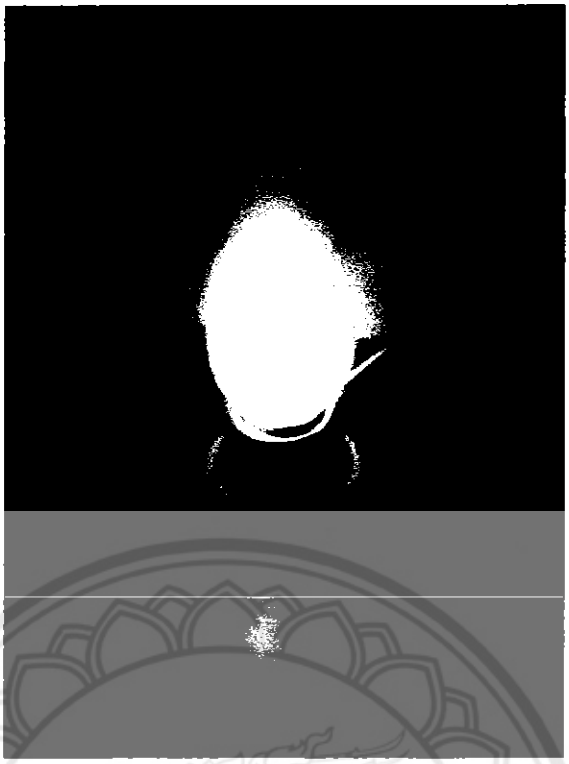
ตารางที่ 4.3 ตารางแสดงการทำงานของหลอดไฟชนิดต่างๆ

การทำงาน	หลอดไส้	หลอดตะเกียบ (แบบธรรมดา)	หลอดตะเกียบ (แบบหรี่ไฟได้)	หลอด LED 220V
เปิดไฟ	Stable	Stable	Stable	Stable
หรี่ไฟขึ้น	Stable	Unstable	Stable	Non
หรี่ไฟลง	Stable	Unstable	Stable	Non
ปิดไฟ	Stable	Stable	Stable	Stable

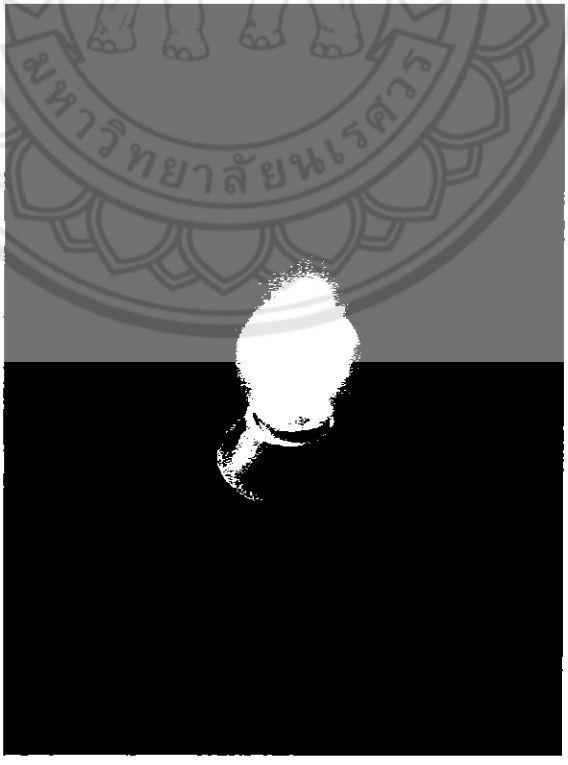
รูปที่ 4.8 ภาพแสดงอุปกรณ์ที่ใช้ทำการทดลอง



รูปที่ 4.9 ภาพแสดงการปิดของหลอดไส้



รูปที่ 4.10 ภาพแสดงการเปิดของหลอดไส้

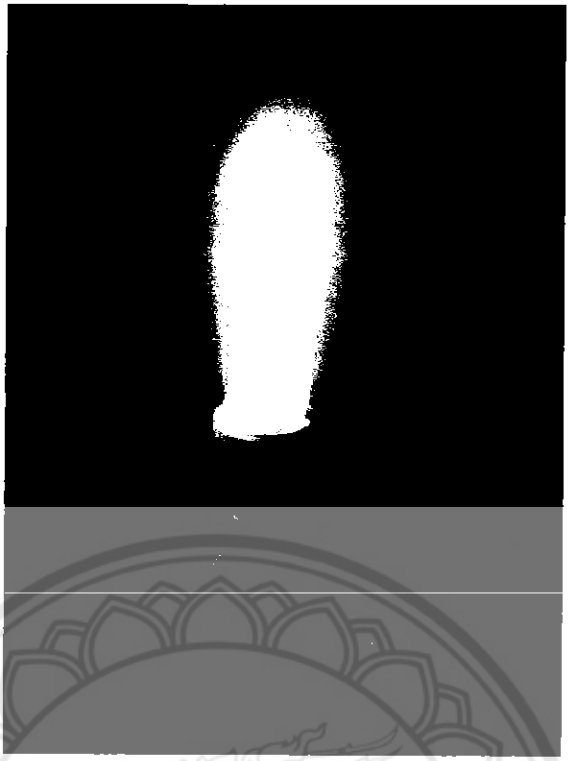


รูปที่ 4.11 ภาพแสดงการหรี่ไฟของหลอดไส้

มหาวิทยาลัยนเรศวร

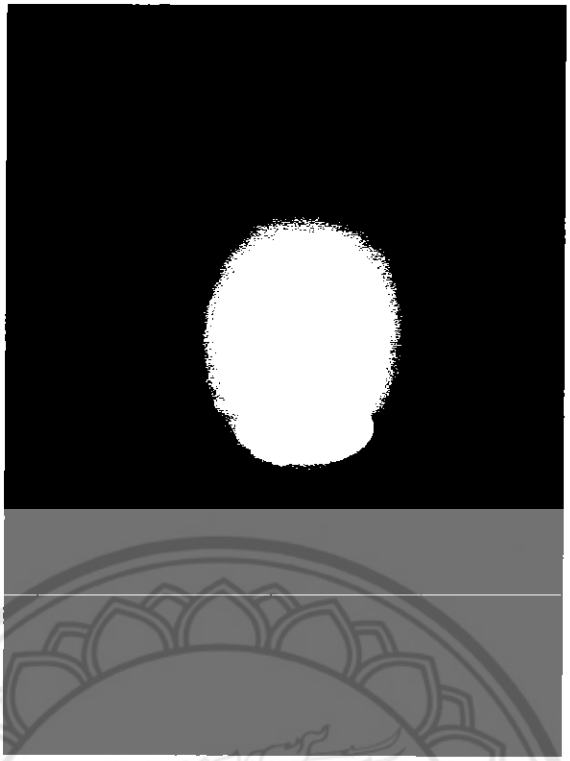
รูปที่ 4.12 ภาพแสดงการปิดของหลอดตะเกียบ (แบบธรรมดา)

รูปที่ 4.13 ภาพแสดงการเปิดของหลอดตะเกียบ (แบบธรรมดา)



รูปที่ 4.14 ภาพแสดงการหรี่ไฟของหลอดตะเกียบ (แบบธรรมดา)

รูปที่ 4.15 ภาพแสดงการปิดของหลอดตะเกียบ (แบบหรี่ไฟได้)



รูปที่ 4.16 ภาพแสดงการเปิดของหลอดตะเกียบ (แบบหรี่ไฟได้)

รูปที่ 4.17 ภาพแสดงการหรี่ไฟของหลอดตะเกียบ (แบบหรี่ไฟได้)

รูปที่ 4.18 ภาพแสดงการปิดของหลอด LED 220V

รูปที่ 4.19 ภาพแสดงการเปิดของหลอด LED 220V

4.2.1 สรุปผลการทดลองที่ 2

จากผลการทดลองด้านบน หลอดไส้จะทำงานได้ทั้งหมด ทั้งการเปิด / ปิด และหรี่ไฟ ส่วนหลอดตะเกียบ (แบบธรรมดา) ก็ทำงานได้ทั้งหมด ทั้งการเปิด / ปิด และหรี่ไฟ แต่การหรี่ไฟในหลอดตะเกียบ (แบบธรรมดา) นั้นจะพบปัญหาเรื่องการกระพริบในการหรี่ในการใช้งานจริงหลอดตะเกียบ (แบบธรรมดา) นั้นไม่แนะนำให้ใช้กับวงจรไฟหรี่ เนื่องจากหลอดตะเกียบจะมีวงจรจุดไส้หลอดแรงดันสูงที่มีลักษณะวงจรแบบสวิตชิ่งส่งผลคือ จะรักษาแรงดันที่จ่ายให้กับหลอดให้คงที่ โดยที่สามารถรับแรงดันเข้าจากขั้วหลอดได้กว้างมากๆซึ่งหลอดพวกนี้สามารถจุดไส้หลอดติดและรักษาแสงสว่างให้คงที่ได้ซึ่งหมายความว่าการลดแรงดันลงไปก็ไม่ส่งผลต่อแสงที่จะลดลง เพราะวงจรจะดึงกระแสเพิ่มเป็นอัตราส่วนที่แรงดันลดลงไป เพื่อรักษาแรงดันไฟที่ส่งให้กับหลอดคงที่ ส่งผลให้ไม่สามารถหรี่ไฟลงได้เพราะว่าไม่ว่าจะลดแรงดันไฟลง ตัววงจรก็จะทำการเพิ่มขึ้นให้อัตโนมัติด้วยการดึงกระแสเพิ่ม [6] ในส่วนของหลอดตะเกียบ (แบบหรี่ไฟได้) นั้นจะทำงานได้ปกติทั้งการเปิด / ปิด และหรี่ไฟ เนื่องจากมีตัวบัลลาสต์อิเล็กทรอนิกส์ ซึ่งบัลลาสต์อิเล็กทรอนิกส์โดยทั่วไปจะประกอบด้วยวงจรไฟฟ้า 2ส่วนส่วนแรกคือวงจรปรับปรุงค่าตัวประกอบกำลังไฟฟ้า (Power Factor Corrector) หรือ PFCซึ่งทำหน้าที่แปลงไฟฟ้ากระแสสลับจากระบบไฟฟ้าเป็นไฟฟ้ากระแสตรงส่วนที่สองคือวงจรแปรผันกำลังไฟฟ้า (Power Inverter) ที่ทำหน้าที่แปลงไฟฟ้ากระแสตรงที่ได้จากส่วน PFC เป็นไฟฟ้ากระแสสลับความถี่สูงเพื่อใช้ในการจุดและขับหลอด วงจรส่วนนี้คือหัวใจสำคัญของบัลลาสต์อิเล็กทรอนิกส์ ซึ่งจะต้องสามารถสร้างแรงดันสูงเพื่อจุดหลอดได้ในตอนแรก และรักษาระดับกระแสไฟฟ้าที่ไหลผ่านหลอดให้มีค่าตามที่ต้องการเมื่อหลอดติดแล้ว ซึ่งการจะหรี่แสงได้ ก็ต้องลดกำลังไฟฟ้าที่จ่ายให้หลอดที่จุดนี้ [7]ในส่วนของหลอด LED 220V จะทำงานได้ในการเปิด / ปิด แต่ไม่สามารถหรี่ไฟได้

4.3 ผลการทดลองที่ 3 การทดลองระยะการทำงานของระบบไร้สาย

จากที่ทำการทดลองแบบที่ 3 โดยวัดระยะการทำงานของระบบไร้สาย (Wireless) ผลที่ได้แสดงได้ดังตารางที่ 4.4

ตารางที่ 4.4 ตารางแสดงระยะการทำงานของระบบไร้สาย (Wireless)

ระยะทาง	การรับส่งสัญญาณ
10 เมตร	ทำงาน
11 เมตร	ทำงาน
12 เมตร	ทำงาน
13 เมตร	ทำงาน
14 เมตร	ทำงาน
15 เมตร	ทำงาน
16 เมตร	ไม่ทำงาน

4.3.1 สรุปผลการทดลองที่ 3

จากผลการทดลองด้านบนจะเห็นได้ว่า ระบบไร้สาย (Wireless) ส่งสัญญาณได้ไกลที่สุด 15 เมตร

บทที่ 5

สรุปผลของการทดลองและปัญหาที่พบ

5.1 สรุปผลการทดลอง

ในการดำเนินโครงงานนี้ผู้ดำเนินโครงงานมีจุดประสงค์ที่จะสร้างระบบสาธิตการทำงานของระบบสวิตช์สัมผัสไร้สาย จากนั้นศึกษาการทำงานการตรวจจับทะลุผ่านสิ่งกีดขวางของเซนเซอร์ การควบคุมการทำงานของหลอดไฟชนิดต่างๆ ด้วยการเปิด / ปิด และหรี่ไฟ และทดสอบระยะการทำงานของระบบไร้สาย โดยในโครงงานนี้ได้เลือกใช้ Microcontroller PIC16F877 เป็นตัวควบคุมการสั่งการทำงานของหลอดไฟโดยแบ่งการควบคุมเป็น 3 แบบ คือ การเปิดไฟ ปิดไฟ และหรี่ไฟและใช้ไตรแอคเบอร์ BTA10 มีค่าทนกระแส 10A ซึ่งจะสามารถใช้งานกับโหลดที่ใช้กำลังไฟฟ้าได้ 2200W แต่ควรใช้กับโหลดที่ใช้กำลังไฟฟ้าที่ 1100W เพื่อกันการกระชากตอนเริ่มทำงาน จากนั้นวิเคราะห์ผลการทดลอง

จากการทดลองพบว่าเซนเซอร์สามารถทำงานทะลุผ่านสิ่งกีดขวางได้ 6 มิลลิเมตร การเปิดและปิดไฟนั้นสามารถทำงานได้กับหลอดไฟทุกชนิดแต่การหรี่ไฟนั้นไม่สามารถใช้ได้กับหลอดไฟบางชนิด และระยะการส่งสัญญาณไร้สายสามารถส่งได้ไกลสุดที่ 15 เมตร

5.2 ปัญหาที่พบ คือ

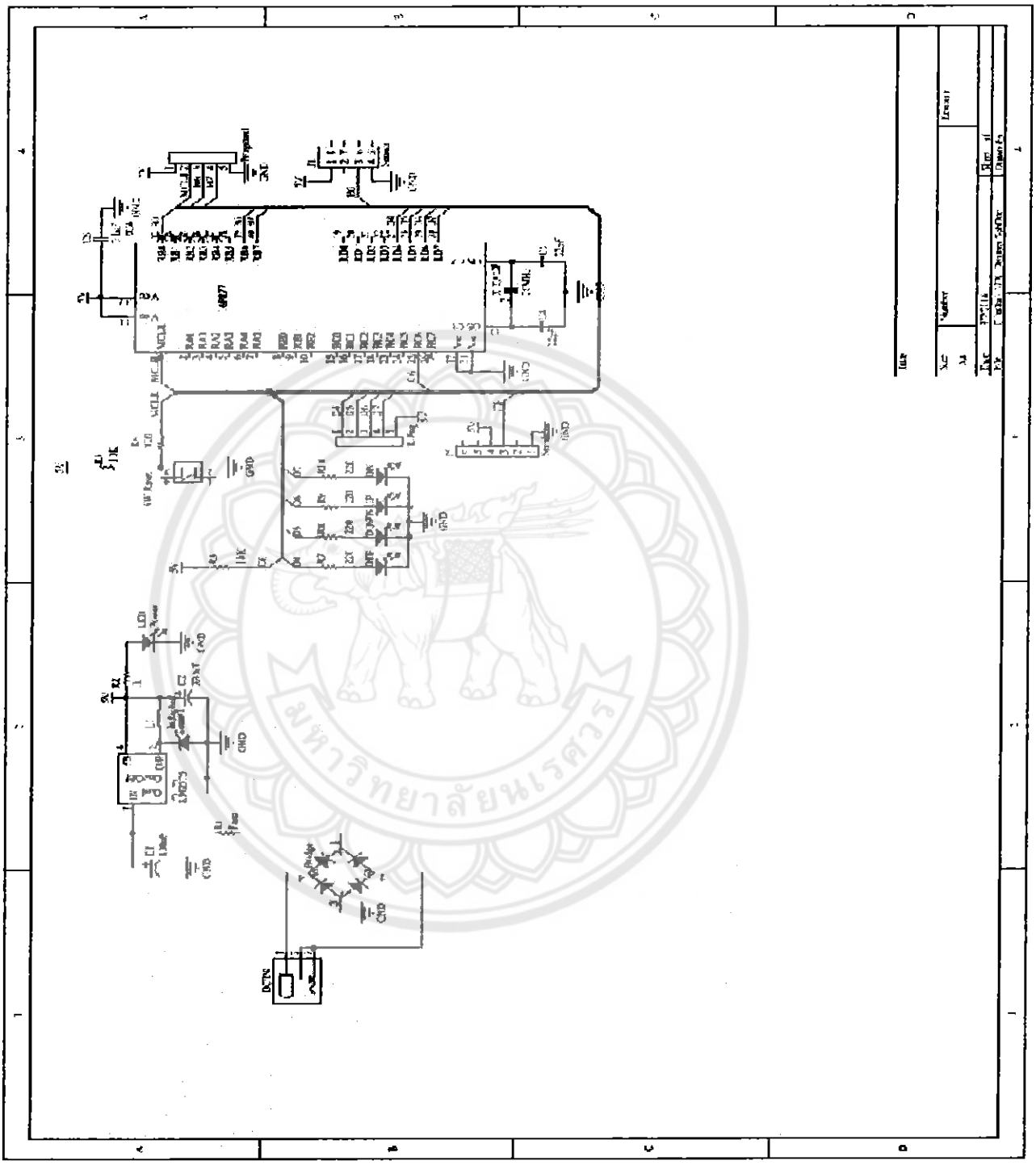
- เมื่อมีการวางสิ่งของทับหน้าสัมผัสของสวิตช์ จะทำให้สวิตช์ทำงาน
- เมื่อมีการสัมผัสสวิตช์ที่ไม่ชัดเจน จะทำให้บอร์ดทดลองค้างค่าที่ไม่เป็นปัจจุบันทำให้ภาคส่งและภาครับติดต่อกันไม่ได้ จึงต้องมีการรีเซ็ตบอร์ดทดลองบ่อยครั้ง
- หลอดไฟที่ไม่สามารถหรี่ไฟได้ อาจช็อตได้เมื่อเราทำการกดสวิตช์เพื่อหรี่ไฟ
- พอร์ท USB ไม่รองรับกับคอมพิวเตอร์ที่ใช้ จึงต้องเปลี่ยนคอมพิวเตอร์จนกว่าพอร์ท USB จะรองรับ เนื่องจากถ้าพอร์ท USB ไม่รองรับกับคอมพิวเตอร์จะทำให้ไม่สามารถเบริ์นโปรแกรมลงไปยังบอร์ด PIC16F877ได้

5.3 การพัฒนาและปรับปรุงแก้ไข

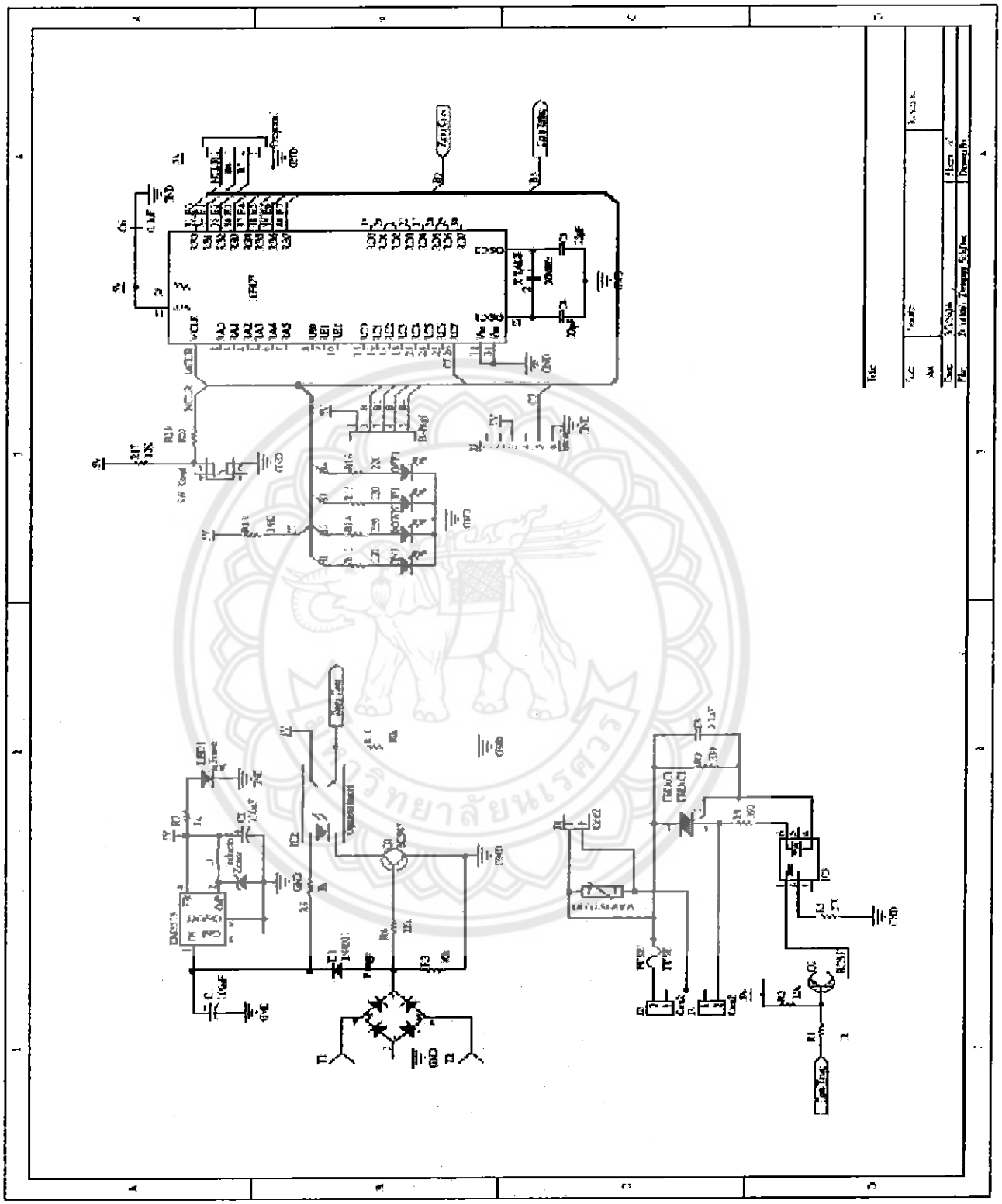
- การวางหน้าสัมผัสของสวิตช์ควรวางแนวตั้ง เพื่อแก้ปัญหาการวางของทับสวิตช์
- เพิ่มรูปแบบการทดลอง คือ อาจเปลี่ยนอุปกรณ์ที่ใช้ทดลองเช่น มอเตอร์ หรืออุปกรณ์กำเนิดความร้อน
- มีการเพิ่มระยะการทำงานของระบบไร้สายให้มีการทำงานได้ไกลขึ้น

ภาคผนวก

1.แบบวงจร



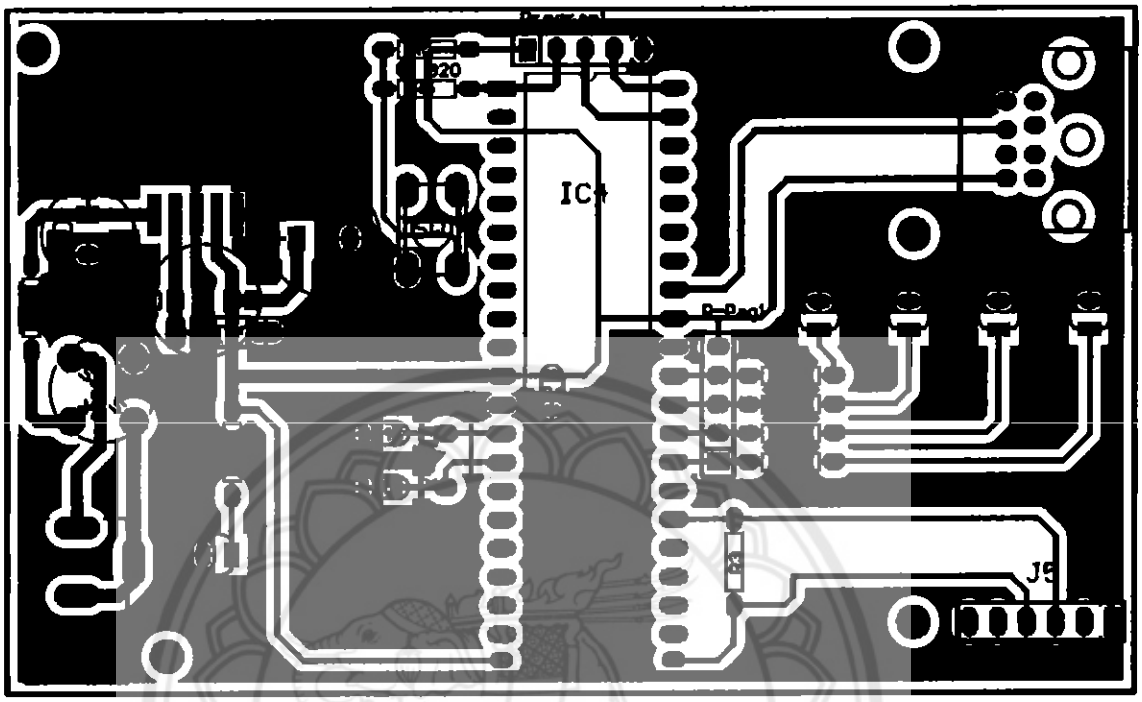
รูปที่ 1 วงจรของภาคส่งสัญญาณ



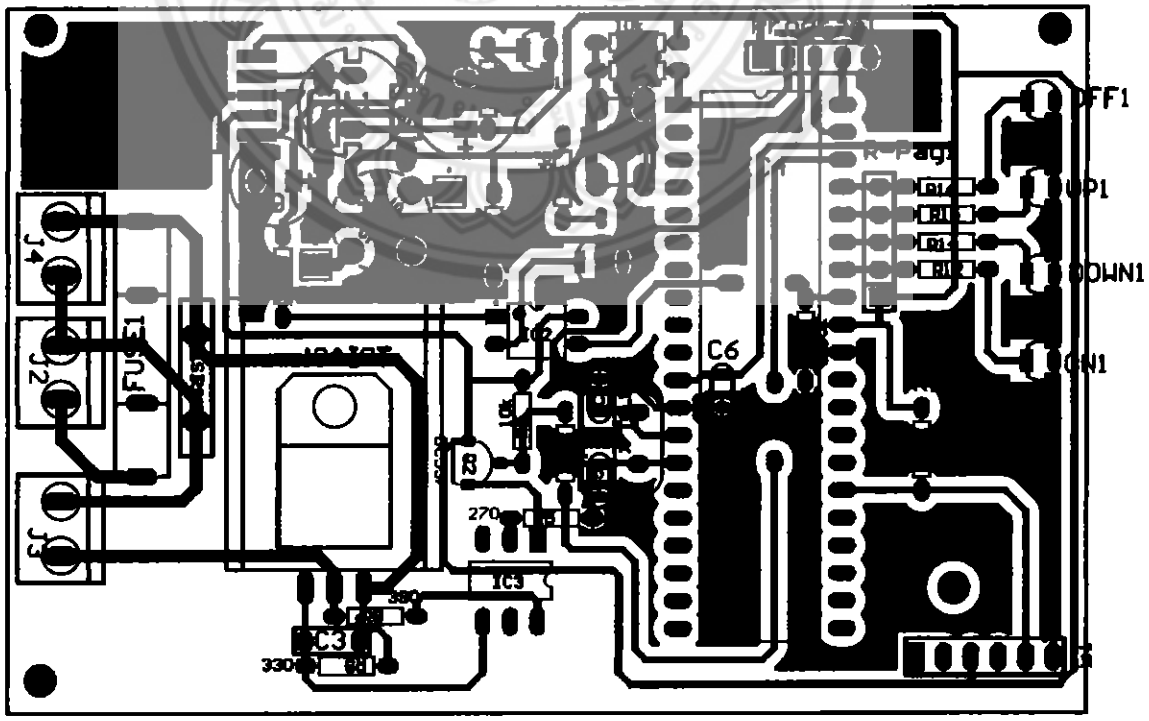
รูปที่ 2 วงจรของภาครับสัญญาณ

2. ลายทองแดง

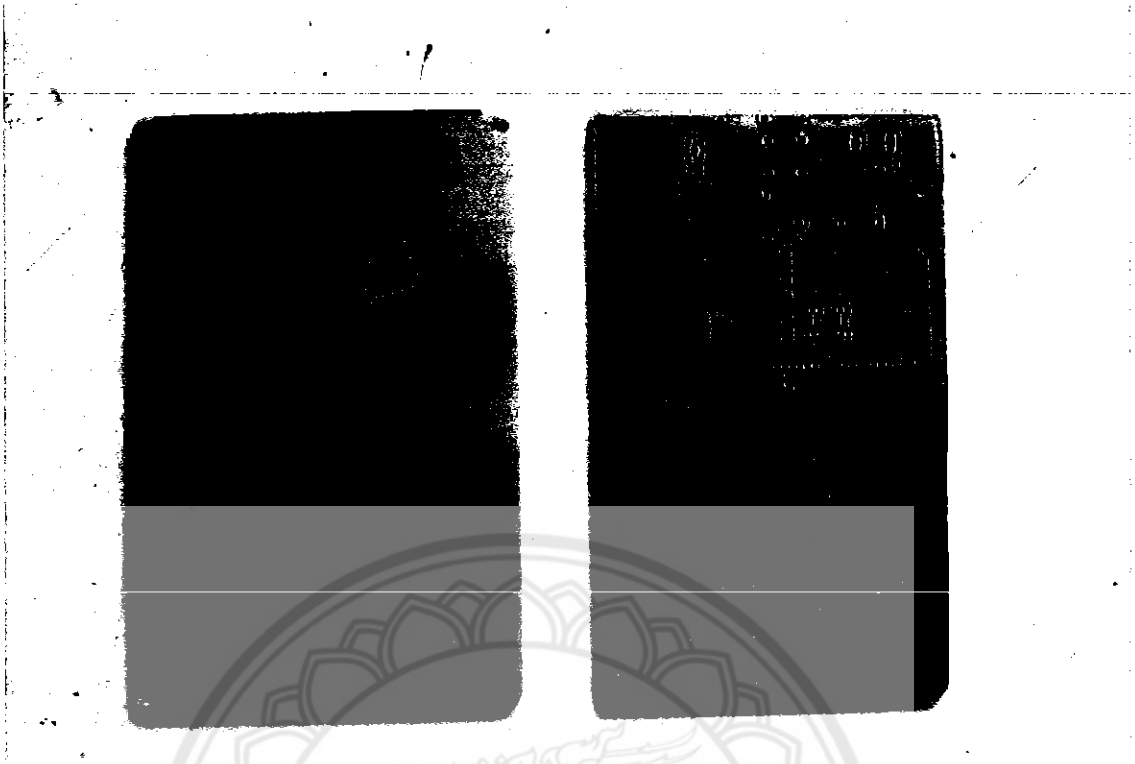
ใช้เป็นแบบเพื่อนำไปใช้ในการกัดแผ่นทองแดง



รูปที่ 3 ลายทองแดงของภาคส่งสัญญาณ



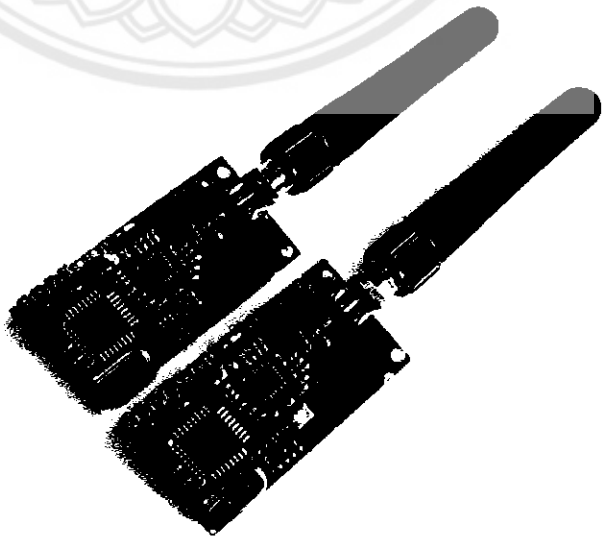
รูปที่ 4 ลายทองแดงของภาครับสัญญาณ



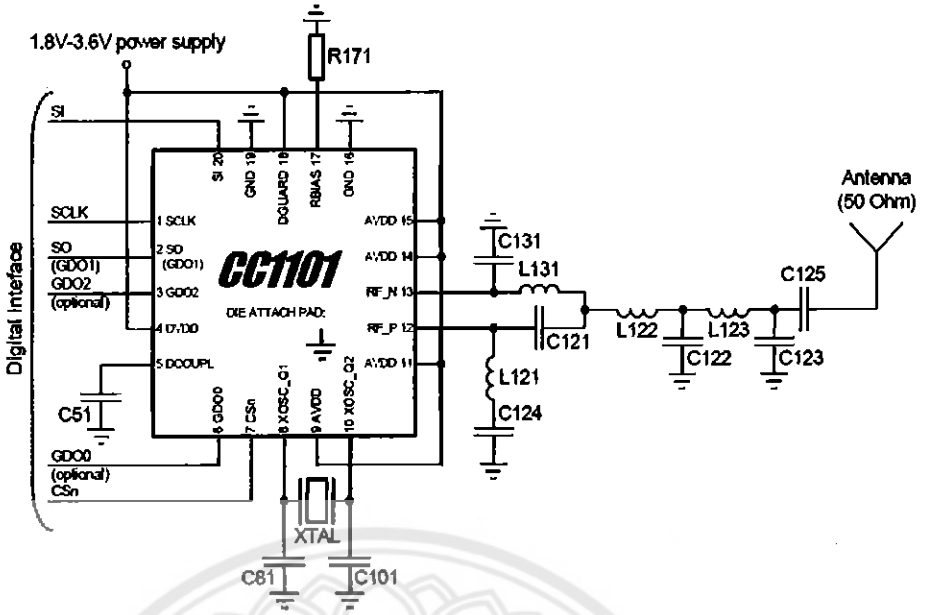
รูปที่ 5 แผ่นทองแดงที่กัดลายแล้ว

3. RF1100-232 Wireless RF Transceiver Module

เป็นโมดูลที่ใช้ชิป CC1100 โดยควบคุมการทำงานด้วยไมโครคอนโทรลเลอร์AVR ดังนั้น ไม่ต้องยุ่งยากในการเขียนโปรแกรมควบคุม CC1101 เพียงแค่ เขียนโปรแกรมแกรมรับ – ส่งผ่าน UART ธรรมดาก็สามารถใช้งานได้แล้ว โดยโมดูลนี้มีกำลังส่งถึง 10mWatt (10dbm)



รูปที่ 6 RF1100-232 Wireless RF Transceiver Module



รูปที่ 7 วงจรของชิป CC1101 ที่ใช้ในย่านความถี่ 433MHz

ตารางที่ 1 ค่าของอุปกรณ์ในรูปที่ 7

ส่วนประกอบ	ค่าที่ใช้ในย่านความถี่ 433MHz
C51	100 nF ± 10%, 0402 X5R
C81	27 pF ± 5%, 0402 NP0
C101	27 pF ± 5%, 0402 NP0
C121	3.9 pF ± 0.25 pF, 0402 NP0
C122	8.2 pF ± 0.5 pF, 0402 NP0
C123	5.6 pF ± 0.5 pF, 0402 NP0
C124	220 pF ± 5%, 0402 NP0
C125	220 pF ± 5%, 0402 NP0
C131	3.9 pF ± 0.25 pF, 0402 NP0
L121	27 nH ± 5%, 0402 monolithic
L122	22 nH ± 5%, 0402 monolithic
L123	27 nH ± 5%, 0402 monolithic
L131	27 nH ± 5%, 0402 monolithic
R171	Koa RK73 series
XTAL	26.0 MHz surface mount crystal

การต่อใช้งานRF1100-232 Wireless RF Transceiver Module เข้ากับไมโครคอนโทรลเลอร์จะอธิบายได้ดังนี้

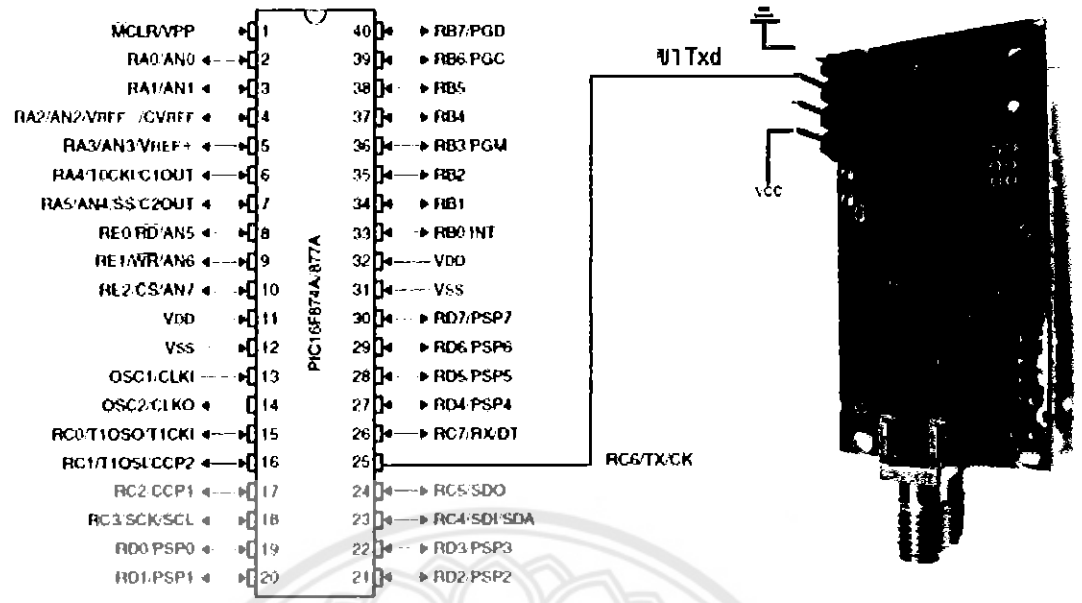
รูปที่ 8 แสดง PINของ RF1100-232Wireless RF Transceiver Module

PIN1 คือ ขา GNDใช้สำหรับต่อกราวด์ให้กับวงจรใน Module

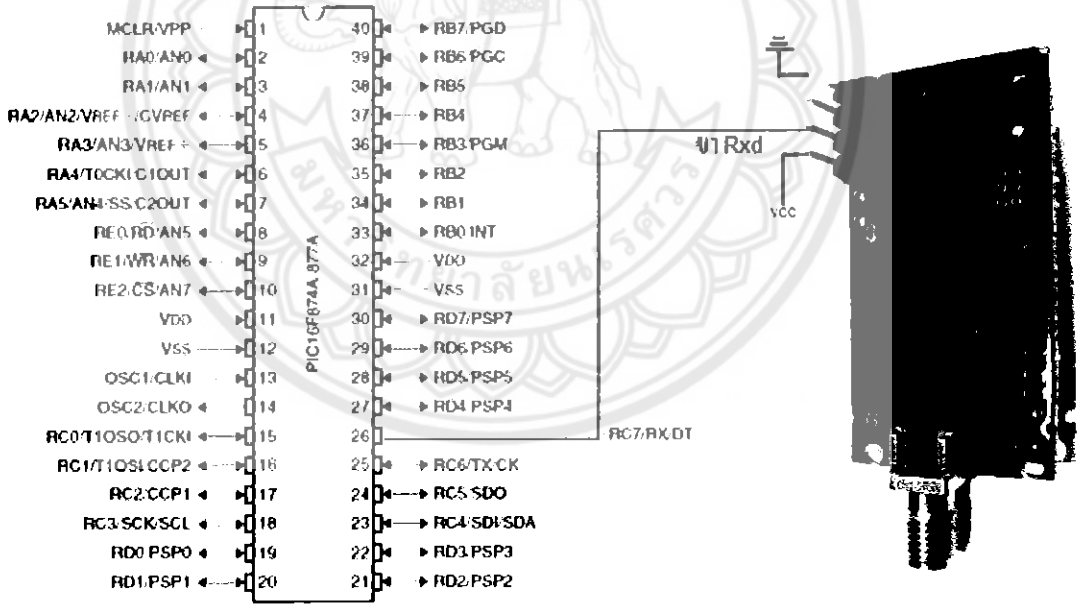
PIN2 คือ ขา Txd คือขาส่งสัญญาณของ Module เมื่อใช้เป็นตัวส่งสัญญาณ จะต่อกับขาส่งข้อมูลแบบพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ เช่น ต่อเข้ากับพอร์ต RC6 ของ PIC16F877 เป็นต้น

PIN3 คือ ขา Rxd คือขารับสัญญาณของ Module เมื่อใช้เป็นตัวรับสัญญาณ จะต่อกับขารับข้อมูลแบบพอร์ตอนุกรมของไมโครคอนโทรลเลอร์ เช่น ต่อเข้ากับพอร์ต RC7 ของ PIC16F877 เป็นต้น

PIN4 คือ ขา VCCใช้สำหรับต่อไฟเลี้ยงให้กับวงจรใน Module



รูปที่ 9 การต่อไมโครคอนโทรลเลอร์กับ Wireless module ในลักษณะการใช้ส่งสัญญาณ



รูปที่ 10 การต่อไมโครคอนโทรลเลอร์กับ Wireless module ในลักษณะการใช้รับสัญญาณ

4. หม้อแปลงไฟฟ้า

ใช้สำหรับแปลงไฟ 220V เป็นไฟ 5V เพื่อจ่ายให้กับบอร์ดไมโครคอนโทรลเลอร์

รูปที่ 11 หม้อแปลงไฟฟ้า

5. โค้ดโปรแกรมของไมโครคอนโทรลเลอร์

1.1 โค้ดโปรแกรมของไมโครคอนโทรลเลอร์ภาคส่งสัญญาณ

```
#include <16F877.h>
#fuses HS,NOWDT,NOPROTECT,NOLVP
#use delay(clock=20000000)
#use rs232(baud=9600,xmit=PIN_C6,rcv=PIN_C7)
#define LedTurnOnpin_D7
#define LedTurnOffpin_D4
#define LedUppin_D6
#define LedDownpin_D5
//#include "LCD.C"
```

```
//intNum_of_work;_
intChk_prees,Turn_on,Turn_off,Up,Down,OnOff,Exit_process;
int16 Times_Delay_switch;
#INT_EXT
void EXT_ISR()          // ฟังก์ชั่น interrupts ภายนอก
{
Chk_prees=1;            // มีการ interrupts ที่ Chk_prees=1
}
voidInit_led();         // ประกาศโปรแกรมย่อย Init_led
voidInit_MCU();         // ประกาศโปรแกรมย่อยInit_MCU
void main()
{

Init_MCU();             // เรียกใช้โปรแกรมย่อยInit_MCU
Init_led();             // เรียกใช้โปรแกรมย่อยInit_led
delay_ms(1);
Chk_prees=0;            // ตัวแปร Chk_prees คือการสัมผัสที่สวิทช์ มี 0 และ 1
Chk_prees=0; OnOff=0; Down=0; Turn_off=0;
while(1)
  {

Chk_prees=0;

while(!Chk_prees);                    // เมื่อมีการกลับค่า Chk_prees
disable_interrupts(INT_EXT);    // ปิด interrupts(INT_EXT)
Chk_prees=0;  Times_Delay_switch=0; Exit_process=0; Turn_off=0;
switch (OnOff)  //On=1,Off=0
    {
```

```
case 0:     // เมื่อยังไม่มีการเปิดไฟ
while(!input(pin_b0));    // เมื่อมี Input เข้ามาที่ pin_b0
delay_ms(1);
printf("1");                    // ให้ส่ง 1 ออกไปที่ขาส่งสัญญาณ pin_c6
output_low(LedTurnOff);
output_high(LedTurnOn);
OnOff=1;                        // ให้ค่าตัวแปรOnOff=1
enable_interrupts(INT_EXT);     // เปิดใช้งานinterrupts(INT_EXT)
delay_ms(1);
break;
case 1:{                    // เมื่อมีการเปิดไฟ
Times_Delay_switch=0; // ตัวแปร Times_Delay_switch คือเวลาที่สัมผัสกับสวิทซ์
Exit_process=0;
      Down=0;
while((!Exit_process)&&(Times_Delay_switch<=1200))
     // เมื่อมีการสัมผัสสวิทซ์ แอนเกจกับ Times_Delay_switch<=1.2 วินาที
      {
Times_Delay_switch++; // ถ้าเป็นจริงให้ค่า Times_Delay_switch เพิ่มขึ้นเรื่อยๆ
if(input(pin_b0))
        {
Times_Delay_switch=0;
while((!Exit_process)&&(Times_Delay_switch<=1300))
         // เมื่อเลิกสัมผัสสวิทซ์ แอนเกจกับ Times_Delay_switch<=1.3 วินาที
          {
Times_Delay_switch++; // ถ้าเป็นจริงให้ค่า Times_Delay_switch เพิ่มขึ้นเรื่อยๆ
if(!input(pin_b0))              // ถ้ามีการกลับค่าสัญญาณ input(pin_b0)
            {
printf("3");                // ให้ส่ง 3 ออกไปที่ขาส่งสัญญาณ pin_c6
```

```
output_high(LedUp);
output_low(LedDown);
delay_ms(1);
while(!input(pin_b0));    // เมื่อมีการกลับค่าสัญญาณ input(pin_b0)
delay_ms(10);
printf("5");              // ให้ส่ง 5 ออกไปที่ขาส่งสัญญาณ pin_c6
output_low(LedUp);
delay_ms(10);
Turn_off=1;
Exit_process=1;
            Down=1;
OnOff=1;
Chk_prees=0;
Times_Delay_switch=0;
enable_interrupts(INT_EXT);    // เปิดใช้งานinterrupts(INT_EXT)
delay_ms(100);
        }
delay_ms(1);
      }
//************************************************************
//*************** Turn OFF LOOP ************************
//************************************************************
if(Turn_off==0)           // ถ้าค่า Turn_off=0
       {
output_low(LedTurnOn);
output_high(LedTurnOff);
printf("2");              // ให้ส่ง 2 ออกไปที่ขาส่งสัญญาณ pin_c6
OnOff=0;
```

```
            Down=1;
Exit_process=1;
Chk_prees=0;
enable_interrupts(INT_EXT);     // เปิดใช้งานinterrupts(INT_EXT)
delay_ms(10);
          }
Turn_off=0;

        } //End pin_b0==1

delay_ms(1);
      }

//*********************************
//********* Step Down ************
//*********************************
if(Down==0)              // ถ้าค่า Down=0
   {
output_low(LedUp);
output_high(LedDown);
printf("4");       // ให้ส่ง 4 ออกไปที่ขาส่งสัญญาณ pin_c6
while(!input(pin_b0)); // เมื่อมีการกลับค่าสัญญาณ input(pin_b0)
printf("5");       // ให้ส่ง 5 ออกไปที่ขาส่งสัญญาณ pin_c6
output_low(LedDown);
Times_Delay_switch=0;
OnOff=1;
Chk_prees=0;
enable_interrupts(INT_EXT);     // เปิดใช้งานinterrupts(INT_EXT)
```

```
    }
    }
break ;
   }//End Switch(OnOff)
delay_ms(10);
  }

}
voidInit_MCU()          // โปรแกรมย่อย Init_MCU
{
set_tris_d(0b00000000); //Port_D Is Output
set_tris_b(0b11111111); //Port_B Is input
set_tris_a(0b00000000); //Port_A Is Output
enable_interrupts(GLOBAL); // เปิดใช้งานinterrupts ทั้งหมด
enable_interrupts(INT_EXT); // เปิดใช้งานinterrupts(INT_EXT)
ext_int_edge(H_TO_L); //ตรวจจับสัญญาณที่ขอบขาลง

}
voidInit_led()          // โปรแกรมย่อย Init_led
{
output_low(LedTurnOn);
output_low(LedTurnOff);
output_low(LedUp);
output_low(LedDown);

}
```

1.2 โค้ดโปรแกรมของไมโครคอนโทรลเลอร์ภาครับสัญญาณ

```
#include <16F877.h>
#fuses HS,NOWDT,NOPROTECT,NOLVP
#use delay(clock=20000000)
#use rs232(baud=9600,xmit=PIN_C6,rcv=PIN_C7)
//#define Rs232_Receive   pin_a0
//!#define LedTurnOff   pin_a1
//!#define LedUp       pin_a2
//!#define LedDown     pin_a3
//#include "LCD.C"
#define LedTurnOn    pin_B1
#define LedTurnOff   pin_B4
#define LedUp       pin_B2
#define LedDown     pin_B3
#include <stdlib.h>
#include <INPUT.c>
#define Port_Con_triac    PIN_B5
//intNum_of_work;_
//!intChk_prees,Turn_on,Turn_off,Up,Down,OnOff,Exit_process;
//!int16 Times_Delay_switch;
voidDetect_RX();          // ประกาศโปรแกรมย่อย Detect_RX
charch;                   // ประกาศ chเป็นตัวแปร char
int1 Exit_process,Stand_by,OnOff,Light,Dimmer;
int Working;
int16 Time_off;
shortint RX=0;
```

```
#INT_RDA
voidRxD_ISR()          // ฟังก์ชั่น interruptsRS232
{
ch=getc();             // รอรับค่า ch
  RX=1;
 // output_high(Rs232_Receive);
Detect_RX();           // ใช้งานโปรแกรมย่อย Detect_RX
}
//!void Init_led();
voidInit_MCU();        // ประกาศโปรแกรมย่อย Init_MCU
voidControl_Triac();   // ประกาศโปรแกรมย่อยControl_Triac
voidTurn_on();         // ประกาศโปรแกรมย่อยTurn_on
voidTurn_off();        // ประกาศโปรแกรมย่อยTurn_off
voidAdd_lighting();    // ประกาศโปรแกรมย่อยAdd_lighting
void Dimming();        // ประกาศโปรแกรมย่อยDimming
voidNormal_work();     // ประกาศโปรแกรมย่อยNormal_work
voidInit_led();        // ประกาศโปรแกรมย่อยInit_led
void main()
{
Init_MCU();            // เรียกใช้โปรแกรมย่อย Init_MCU
Init_led();            // เรียกใช้โปรแกรมย่อยInit_led
Turn_off();            // เรียกใช้โปรแกรมย่อยTurn_off
OnOff=0;               // ค่าตัวแปรOnOff คือค่าที่แสดงสถานะของไฟ
Time_off=0;            // ค่าตัวแปรTime_offคือค่าเวลาของZero crossing
  Working=0;
  RX=0;
Stand_by=0;
Exit_process=0; Light=0; Dimmer=0;
```

```
delay_ms(10);
while(1)
{
while(!RX)       // เมื่อมีการกลับค่า RX
 {
if((Time_off==0)&&(!(OnOff==1)))       // ถ้า Time_off=0 แอนเกจกับ OnOffไม่เท่ากับ 1
  {
Turn_off();              // เรียกใช้โปรแกรมย่อยTurn_off
  }
if((OnOff==0)&&(!(Time_off==0)))       // ถ้า OnOff=0 แอนเกจกับ Time_offไม่เท่ากับ 0
    {
Normal_work(); // เรียกใช้โปรแกรมย่อยNormal_work
    }
   }
 RX=0;



switch (Working)                    // ฟังก์ชั่น switch (Working)
    {
case 1 :  Turn_on();      //เมื่อค่า Working เป็น 1เรียกใช้โปรแกรมย่อยTurn_on
OnOff=1;                  // กำหนดค่า OnOff=1
break;
case 2 :  Turn_off();     // เมื่อค่า Working เป็น 2เรียกใช้โปรแกรมย่อยTurn_off
OnOff=0;                  // กำหนดค่า OnOff=0
break;
case 3 : output_high(LedUp);
Add_lighting();  // เมื่อค่า Working เป็น 3เรียกใช้โปรแกรมย่อยAdd_lighting
```

```
output_low(LedUp);
break;
case 4 :  output_high(LedDown);
Dimming();                    // เมื่อค่า Working เป็น 4เรียกใช้โปรแกรมย่อยDimming
output_low(LedDown);
break;
    }
}
}
//*************************************************************
//************sub  program*********************************
//*************************************************************
voidTurn_on()                 // โปรแกรมย่อย Turn_on
{
output_low(Port_Con_triac);   // ให้ลอจิก 0 ที่ Port_Con_triac
delay_us(100);
output_high(LedTurnOn);
output_low(LedTurnOff);
delay_us(10);
Time_off=0;
}
voidTurn_off()                // โปรแกรมย่อย Turn_off
{
output_high(Port_Con_triac);  // ให้ลอจิก 1 ที่ Port_Con_triac
delay_us(100);
output_low(LedTurnOn);
output_high(LedTurnOff);
delay_us(10);
```

```
}

voidInit_MCU()          // โปรแกรมย่อย Init_MCU
{
set_tris_d(0b00000000); //Port_D Is Output
set_tris_b(0b11000001); //Port_B Is Output
set_tris_a(0b00000000); //Port_A Is Output
output_low(Port_Con_triac);
enable_interrupts(GLOBAL);    // เปิดใช้ interrupts ทั้งหมด
enable_interrupts(INT_RDA);   // เปิดใช้interrupts RS232
}
voidAdd_lighting()        // โปรแกรมย่อย Add_lighting
{
if(Dimmer==1)  // ถ้าค่า Dimmer=1
  {
Time_off=Time_off-10; // ให้ค่าTime_off=Time_off-10
    Dimmer=0;          // กำหนดค่า Dimmer=0
  }
while(Exit_process==0)
  {
while(input(pin_b0));             // เมื่อมี input ที่ pin_b0
if(Time_off>0)           // ถ้าค่า Time_off>0
    {

delay_us(Time_off);
output_low(Port_Con_triac);     // ให้ลอจิก 0 ที่ Port_Con_triac
delay_us(100);
output_high(Port_Con_triac);    // ให้ลอจิก 1 ที่ Port_Con_triac
```

```
delay_us(10);
Time_off=Time_off-10;          // ให้ค่าTime_off=Time_off-10
    }
if(Time_off==0)          // ถ้าค่า Time_off=0
    {
Time_off=10000;                // ให้ค่า Time_off=10000
    }
  }
Exit_process=0; OnOff=0;  Light=1;  // กำหนดค่าตัวแปรตามลำดับ
}
void Dimming()                 // โปรแกรมย่อย Dimming
{
if(Light==1)                   // ถ้าค่า Light=1
  {
Time_off=Time_off+10; // ให้ค่าTime_off=Time_off+10
    Light=0;                   // กำหนดค่า Light=0
  }
while(Exit_process==0)
  {
while(input(pin_b0));          // เมื่อมี input ที่ pin_b0
if(Time_off>=0)          // ถ้าค่า Time_off>0
    {
delay_us(Time_off);
output_low(Port_Con_triac);    // ให้ลอจิก 0 ที่ Port_Con_triac
delay_us(100);
output_high(Port_Con_triac);   // ให้ลอจิก 1 ที่ Port_Con_triac
delay_us(10);
Time_off=Time_off+10;          // ให้ค่าTime_off=Time_off+10
```

```
    }
if(Time_off>=10000)                    // ถ้าค่า Time_off=10000
    {
Time_off=0;                            // ให้ค่า Time_off=0
    }
  }
Exit_process=0; OnOff=0; Dimmer=1;     // กำหนดค่าตัวแปรตามลำดับ
}
voidDetect_RX()                        // โปรแกรมย่อย Detect_RX
{
switch (ch)                            // เริ่มทำงานที่ฟังก์ชั่น switch (ch)
    {
case '1' :  Stand_by=1;    // เมื่อค่า chเป็น 1 กำหนดค่า Stand_by=1
            Working=1;             // กำหนดค่าWorking=1
break;
case '2' :  Stand_by=0;    // เมื่อค่า chเป็น 2 กำหนดค่า Stand_by=0
            Working=2;             // กำหนดค่า Working=2
break;
case '3' :  Stand_by=1;    // เมื่อค่า chเป็น 3 กำหนดค่า Stand_by=1
            Working=3;             // กำหนดค่า Working=3
Exit_process=0; // กำหนดค่า Exit_process=0
break;
case '4' :  Working=4;                 // เมื่อค่า chเป็น 4 กำหนดค่า Working=4
Exit_process=0; // กำหนดค่า Exit_process=0
Stand_by=1;                // กำหนดค่าStand_by=1
break;
case '5' :  Exit_process=1;            // เมื่อค่า chเป็น 4 กำหนดค่า Exit_process=1
            Working=0;             // กำหนดค่าWorking=0
```

```
        Stand_by=1;              // กำหนดค่าStand_by=1
        break;
            }
        }
        voidNormal_work()              // โปรแกรมย่อย Normal_work
        {
        while(input(pin_b0));     // เมื่อมีค่า input ที่ pin_b0
        delay_us(Time_off);
        output_low(Port_Con_triac); // ให้ลอจิก 0 ที่ Port_Con_triac
        delay_us(100);
           //   delay_us(Time_off-(Time_off-10));
        output_high(Port_Con_triac); // ให้ลอจิก 1 ที่ Port_Con_triac
        delay_us(10);
        }
        voidInit_led()                 // โปรแกรมย่อย Init_led
        {
        output_low(LedTurnOn);
output_low(LedTurnOff);
        output_low(LedUp);
        output_low(LedDown);
          }
```

เอกสารอ้างอิง

[1] ผศ.ดร.นวภัทรา หนูนาค.(2555). พร็อกซิมิตี้เซนเซอร์ชนิดเก็บประจุ (capacitive proximity sensor). **Food Network Solution ศูนย์เครือข่ายข้อมูลอาหารครบวงจร.**สืบค้นเมื่อ 5กุมภาพันธ์ 2557, จากhttp://www.foodnetworksolution.com/wiki/word/4346/

[2] Sensors for Automation. (27 กุมภาพันธ์ 2555). **ME-SUT.**สืบค้นเมื่อ 5กุมภาพันธ์ 2557, จาก http://eng.sut.ac.th/me/box/3_54/425311/17%20Sensors%20for%20Automation.pdf

[3] สันทนาสงครินทร์.(2548) วิชาช่างไมโครคอนโทรลเลอร์. ครูสันทนา สงครินทร์. สืบค้นเมื่อ 5 กุมภาพันธ์ 2557,จากhttp://www.sunthana.com/microcontroller.pdf

[4] ผศ.ดร. อาภรณ์ ธีรมงคลรัศมี.(7พฤษภาคม 2551).ไธรีสเตอร์และสิ่งประดิษฐ์สำหรับการจุดชนวน (Thyristor& Triggering devices , TTD). **2102384 Electronics Laboratory.**สืบค้นเมื่อ 5 กุมภาพันธ์ 2557,จาก http://pioneer.netserv.chula.ac.th/~tarporn/384/LabSheet/TTD.pdf

[5] อาร์ทคอมฯ. (19 พฤษภาคม2553). การควบคุมกำลังไฟแบบเฟสทริกเกอร์. **Living electronics.** สืบค้นเมื่อ 5กุมภาพันธ์ 2557,จาก http://living-electronics.blogspot.com/2010/05/1_19.html

[6] Satomi. (21 กุมภาพันธ์ 2553). **Guru Google.** สืบค้นเมื่อ 5กุมภาพันธ์ 2557, จากhttp://guru.google.co.th/guru/thread?tid=6b8ae21305f4265e

[7] วัชระ ปิ่นเพชร. (2549). **การวิเคราะห์และออกแบบบลลาสต์อิเล็กทรอนิกส์สำหรับหลอดคอมแพคฟลูออเรสเซนต์ที่สามารถปรับความสว่างได้ด้วยวงจรควบคุมชนิดปรับมุมเฟสแรงดันด้านเข้า.** วิทยานิพนธ์ วศ.บ.,สถาบันเทคโนโลยีพระจอมเกล้าพระนครเหนือ, กรุงเทพมหานคร. **สืบค้นเมื่อ 5กุมภาพันธ์ 2557, จาก http://www.gits.kmutnb.ac.th/ethesis/data/4510181094.pdf**