อุปกรณ์ควบคุมการเปิด-ปิด หลอดไฟ
CONTROL DEVICE TO TURN ON-OFF THE LAMP

โดย

นายศุภกร บุญเติม

นายสถาพร เหลืองทรงชัย

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

อุปกรณ์ควบคุมการเปิด-ปิด หลอดไฟ

CONTROL DEVICE TO TURN ON-OFF THE LAMP

โดย

นายศุภกร บุญเติม

นายสถาพร เหลืองทรงชัย

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ควบคุมการเปิด-ปิด หลอดไฟ

CONTROL DEVICE TO TURN ON-OFF THE LAMP

โดย

นายศุภกร บุญเติม 53011588

นายสถาพร เหลืองทรงชัย 53011629

อาจารย์ที่ปรึกษา

ดร.สิรภพ ตู้ประกาย

รศ.ดร.กอบชัย เดชหาญ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผ่านการตรวจรูปเล่มแล้ว

(.....)

อาจารย์ที่ปรึกษา

4/3/57

KMITL วิศวกรรมโทรคมนาคม Telecommunications Engineering

ผ่านการตรวจชิ้นงานแล้ว

(.....)

กรรมการผู้ตรวจชิ้นงาน

12/3/57

KMITL วิศวกรรมโทรคมนาคม Telecommunications Engineering

ปริญญานิพนธ์ปีการศึกษา 2556

สาขาวิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง อุปกรณ์ควบคุมการเปิด-ปิด หลอดไฟ

CONTROL DEVICE TO TURN ON-OFF THE LAMP

ผู้จัดทำ

1. นายศุภกร บุญเติม 53011588
2. นายสถาพร เหลืองทรงชัย 53011629

..... **อาจารย์ที่ปรึกษา**

(ดร.สิรภพ ตู้ประกาย)

..... **อาจารย์ที่ปรึกษาร่วม**

(รศ.ดร.กอบชัย เดชหาญ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์เล่มนี้คงไม่อาจสำเร็จได้ด้วยดี หากไม่ได้รับความช่วยเหลือและความร่วมมือจากหลายๆฝ่ายด้วยกัน บุคคลที่ต้องกล่าวถึงอันดับแรกเพราะเป็นบุคคลสำคัญที่ทำให้โครงงานนี้สำเร็จได้ก็คือ ดร.สิรภพ ตู้ประกาย อาจารย์ที่ปรึกษาปริญญานิพนธ์ ที่คอยให้คำปรึกษา เอาใจใส่ดูแล ให้คำแนะนำ คอยย้ำเตือนช่วงระยะเวลาและการวางแผนการดำเนินงานมาโดยตลอดและให้ความช่วยเหลือเป็นอย่างดีโดยตลอดมา

ขอขอบคุณพระคุณคณาจารย์ สาขาวิชาวิศวกรรมโทรคมนาคม คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง ที่ได้ประสิทธิ์ประสาทวิชาให้แก่คณะผู้จัดทำ

ขอขอบคุณรุ่นพี่ เพื่อนๆและน้องๆทุกคนที่คอยให้ความช่วยเหลือและเป็นกำลังใจให้กันโดยตลอดมา จนทำให้ปริญญานิพนธ์เล่มนี้ลุล่วงไปได้ด้วยดี

และสุดท้ายต้องขอกราบขอบพระคุณบุคคลที่สำคัญที่สุดที่ทำให้มีวันนี้ได้ ก็คือ บิดามารดาอันเป็นที่เคารพรักที่สุด ที่ได้ให้กำเนิดและเลี้ยงดูเป็นอย่างดี พร้อมทั้งให้โอกาสในการศึกษาอย่างเต็มที่ และให้กำลังใจ เอาใจใส่และให้ความช่วยเหลือสนับสนุนในทุกๆด้าน

คุณค่าและประโยชน์อันพึงได้มาจากโครงงานนี้ คณะผู้จัดทำขอมอบแด่ผู้มีพระคุณทุกท่าน

นายศุภกร บุญเติม

นายสถาพร เหลืองทรงชัย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ควบคุมการเปิด-ปิด หลอดไฟ
CONTROL DEVICE TO TURN ON-OFF THE LAMP

โดย นายศุภกร บุญเติม 53011588
นายสถาพร เหลืองทรงชัย 53011629

อาจารย์ที่ปรึกษา ดร.สิรภพ ตู้ประกาย
อาจารย์ที่ปรึกษาร่วม รศ.ดร.กอบชัย เดชหาญ

บทคัดย่อ

อุปกรณ์ควบคุมการเปิด-ปิดหลอดไฟ ขึ้นอยู่กับหลักการทำงานของ ZIGBEE ในการเป็นโหนดส่งสัญญาณ และใช้ PIC16F877 เป็นตัวไมโครคอนโทรลเลอร์ในการควบคุมการทำงานของวงจร เริ่มแรกเราจะทำการกดปุ่มที่อุปกรณ์ควบคุมเพื่อส่งสัญญานไปยังไมโครคอนโทรลเลอร์ตัวที่ 1 จากนั้นจะส่งคำสั่งไปยัง ZIGBEE1 เมื่อ ZIGBEE2 ได้รับข้อมูลที่ส่งมาจาก ZIGBEE1 จากนั้นจะนำไปประมวลผลที่ไมโครคอนโทรลเลอร์ตัวที่ 2 ท้ายที่สุดไมโครคอนโทรลเลอร์ตัวที่ 2จะส่งคำสั่งควบคุม RELAY เพื่อควบคุมการเปิด-ปิดหลอดไฟ

ABSTRACT

A control device to turn on-off the lamp depends on the principle of zigbees which are nodes for sending signal, and PIC16F877, which is the microcontroller to control circuit. First, we push a button at a control device for sending signal to a microcontroller#1. Then, the microcontroller sends command to zigbee#1 to send data to a zigbee#2. After a zigbee#2 receives data from a zigbee#1, all data is processed by a microcontroller#2. Finally, the microcontroller#2 sends command to control relay for turning on-off the lamp.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

		หน้า
กิตติกรรมประกาศ		I
บทคัดย่อ		II
สารบัญ		III
สารบัญรูป		V
สารบัญตาราง		VIII
บทที่ 1	**บทนำ**	**1**
	1.1 ความเป็นมาและความสำคัญของปัญหา	1
	1.2 วัตถุประสงค์	1
บทที่ 2	**ทฤษฎีและหลักการที่เกี่ยวข้อง**	**2**
	2.1 ZIGBEE	2
	2.2 PIC16F877	11
	2.3 RELAY	14
บทที่ 3	**การออกแบบและการจัดทำปริญญานิพนธ์**	**19**
	3.1 การออกแบบวงจรในโครงงาน	19
	3.2 เครื่องมือที่ใช้ในการทดลอง	33
	3.3 การตั้งค่า ZIGBEE ผ่านโปรแกรม X-CTU	39
	3.4 การออกแบบการจัดเก็บผลของปริญญานิพนธ์	42

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

		หน้า
บทที่ 4	**ผลการทดลอง**	**44**
	4.1 ผลการทดสอบการส่งข้อมูลผ่าน ZIGBEE ทั้ง 3 ตัว	44
	4.2 ผลการทดสอบการส่งข้อมูลของฝั่งส่งโดยใช้โปรแกรม X-CTU	45
	4.3 ผลการทดสอบผ่าน SPECTRUM ANALYZER	46
	4.4 ผลการทดสอบการวัดค่าผ่าน MIXED SIGNAL OSCILLOSCOPE	47
	4.5 ผลการทดสอบการเปิด – ปิดหลอดไฟด้วยอุปกรณ์จริง	66
บทที่ 5	**สรุปผลและข้อเสนอแนะ**	**63**
	5.1 สรุปผล	63
	5.2 ข้อเสนอแนะ	64
บรรณานุกรม		**72**
ภาคผนวก ก	ผังงานโปรแกรม (PROGRAM FLOWCHART)	73
ภาคผนวก ข	ขาของ PIC 16F877	74
ภาคผนวก ค	CODE PROGRAM	76

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

รูปที่		หน้า
2.1	CHANNEL ของ ZIGBEE	4
2.2	ZIGBEE PROTOCOL	5
2.3	ZIGBEE แบบ WIRE ANTENNA	5
2.4	ZIGBEE แบบ CHIP ANTENNA	6
2.5	ZIGBEE แบบ UFL CONNECTOR	6
2.6	ZIGBEE แบบ SMA CONNNECTOR	7
2.7	STAR (BROADCAST)	8
2.8	CLUSTER TREE	9
2.9	MESH	10
2.10	ลักษณะทางกายภาพของ ZIGBEE	10
2.11	ลักษณะทาง MACHANICLE ของ ZIGBEE	11
2.12	PIC16F877	13
2.13	RELAY	15
2.14	หน้าสัมผัสของ RELAY	16
3.1	BLOCK DIAGRAM	19
3.2	FLOWCHART (TRANSMITTER)	20
3.3	FLOWCHART (1^{ST} RECEIVER)	22
3.4	FLOWCHART แสดงการส่งข้อมูลกลับจาก 1^{ST} RECEIVER	24
3.5	FLOWCHART (2^{ND} RECEIVER)	25
3.6	FLOWCHART แสดงการส่งข้อมูลกลับจาก 2^{ND} RECEIVER	27
3.7	วงจรทั้ง 3 วงจร	28
3.8	หลอดไฟจริง 4 ดวง	28
3.9	วงจรแปลงแรงดันเข้า PIC16F877	29
3.10	วงจรแปลงแรงดันเข้า ZIGBEE	29
3.11	วงจรฝั่งส่ง	30

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.12	วงจรฝั่งรับที่ 1	31
3.13	วงจรฝั่งรับที่ 2	32
3.14	การจัดขาของ PIC16F877	34
3.15	IC7805	36
3.16	LM1117T	36
3.17	การทำงานของRELAY	37
3.18	RELAY	38
3.19	ULN2003APG	38
3.20	หน้าเริ่มต้นของ X-TCU	39
3.21	หลังจากการกด TEST	40
3.22	หน้า MODEM CONFIGURATION ของ X-CTU	41
4.1	ผลการทดสอบการส่งข้อมูลของ ZIGBEE#1 และ ZIGBEE#2	44
4.2	ผลการทดสอบของฝั่งส่งโดย X-CTU	45
4.3	ผลการวัดสัญญาณผ่าน SPECTRUM ANALYZER	46
4.4	สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘A’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ	47
4.5	สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘B’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ	48
4.6	สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘C’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ	48
4.7	สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘D’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ	49
4.8	สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘E’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ	49
4.9	สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘F’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ	50
4.10	สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘G’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ	50
4.11	สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘H’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ	51
4.12	สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘I’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ	51
4.13	สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘J’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ	52
4.14	สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘K’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ	52
4.15	สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘L’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ	53
4.16	สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘M’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ	53
4.17	สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘N’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ	54

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.18 สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘O’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ 54
4.19 สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘P’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ 55
4.20 สัญญาณที่เกิดจากตัวอักษร ‘a’ จากการส่งข้อมูลกลับมา 57
4.21 สัญญาณที่เกิดจากตัวอักษร ‘b’ จากการส่งข้อมูลกลับมา 57
4.22 สัญญาณที่เกิดจากตัวอักษร ‘c’ จากการส่งข้อมูลกลับมา 58
4.23 สัญญาณที่เกิดจากตัวอักษร ‘d’ จากการส่งข้อมูลกลับมา 58
4.24 สัญญาณที่เกิดจากตัวอักษร ‘e’ จากการส่งข้อมูลกลับมา 59
4.25 สัญญาณที่เกิดจากตัวอักษร ‘f’ จากการส่งข้อมูลกลับมา 59
4.26 สัญญาณที่เกิดจากตัวอักษร ‘g’ จากการส่งข้อมูลกลับมา 60
4.27 สัญญาณที่เกิดจากตัวอักษร ‘h’ จากการส่งข้อมูลกลับมา 60
4.28 สัญญาณที่เกิดจากตัวอักษร ‘i’ จากการส่งข้อมูลกลับมา 61
4.29 สัญญาณที่เกิดจากตัวอักษร ‘j’ จากการส่งข้อมูลกลับมา 61
4.30 สัญญาณที่เกิดจากตัวอักษร ‘k’ จากการส่งข้อมูลกลับมา 62
4.31 สัญญาณที่เกิดจากตัวอักษร ‘l’ จากการส่งข้อมูลกลับมา 62
4.32 สัญญาณที่เกิดจากตัวอักษร ‘m’ จากการส่งข้อมูลกลับมา 63
4.33 สัญญาณที่เกิดจากตัวอักษร ‘n’ จากการส่งข้อมูลกลับมา 63
4.34 สัญญาณที่เกิดจากตัวอักษร ‘o’ จากการส่งข้อมูลกลับมา 64
4.35 สัญญาณที่เกิดจากตัวอักษร ‘p’ จากการส่งข้อมูลกลับมา 64
4.36 ตัวอย่างผลการทดสอบเมื่อส่ง ‘A’ ไปยังวงจรฝั่งรับ 66
4.37 ตัวอย่างผลการทดสอบเมื่อส่ง ‘I’ ไปยังวงจรฝั่งรับ 67
4.38 ตัวอย่างผลการทดสอบเมื่อส่ง ‘J’ ไปยังวงจรฝั่งรับ 68
4.39 ตัวอย่างผลการทดสอบเมื่อส่ง ‘K’ ไปยังวงจรฝั่งรับ 69

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

ตารางที่		หน้า
2.1	ย่านความถี่ใช้งานของ ZIGBEE	3
3.1	การทำงานแต่ละขาของโมดูล ZIGBEE	33
3.2	คุณสมบัติของ ZIGBEE SERIES2	34
4.1	ค่าต่างๆของข้อมูลที่ฝั่งส่งและฝั่งรับ	56
4.2	ค่าข้อมูลที่ส่งกลับมา	62

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

เนื่องจากในชีวิตประจำวันมีเทคโนโลยีมากมายที่ทำให้มนุษย์เราใช้ชีวิตสะดวกสบายอย่างมาก ไม่ว่าจะเป็นในด้านไหนก็ตาม ทางผู้จัดทำเล็งเห็นในจุดนี้จึงคิดค้นอุปกรณ์เปิด-ปิดหลอดไฟ โดยใช้อุปกรณ์คล้ายรีโมตในการควบคุม เพื่อเอื้ออำนวยความสะดวกแก่ผู้ใช้งาน โดยผู้ใช้งานไม่ต้องเดินไปเปิดหลอดไฟแต่สามารถกดที่อุปกรณ์นี้แทนได้ ซึ่งจะใช้อุปกรณ์ Zigbee เป็นอุปกรณ์ส่งสัญญาณข้อมูลจากตัวหนึ่งไปยังอีกตัวหนึ่งและใช้ Microcontroller คือ PIC18F877 เป็นตัวควบคุมการทำงานของวงจรทั้งหมด

1.2 วัตถุประสงค์

โครงงานนี้เป็นโครงงานเปิด-ปิดหลอดไฟโดยใช้ Zigbee โดยมีวัตถุประสงค์ดังนี้

1.) เพื่อศึกษาหลักการทำงานของ Zigbee

2.) เพื่อศึกษาการเขียนคำสั่งบน Microcontroller

3.) เพื่ออำนวยความสะดวกสบายในการใช้ชีวิตประจำวัน

1.3 ขอบเขตของปริญญานิพนธ์

1.) เขียนคำสั่งเพื่อใช้ติดต่อระหว่าง Microcontroller และ Zigbee

2.) สร้างอุปกรณ์ที่ใช้สำหรับส่งคำสั่งไปยังอุปกรณ์ภาคส่งและรับ

3.) สร้างอุปกรณ์ภาคส่งและรับเพื่อควบคุมการเปิด-ปิดหลอดไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 Zigbee

Zigbee เป็นเทคโนโลยีไร้สายที่ร่วมกันสื่อสารข้อมูลผ่านเซ็นเซอร์ขนาดเล็กจิ๋ว จำนวนเป็นพันๆ หมื่นๆ ชิ้นที่ฝังอยู่ตามส่วนต่างๆ ในอาคาร สำนักงาน โรงงาน หรือแม้แต่ในบ้าน การทำงานของมันจะเป็นการรับ-ส่งคลื่นสัญญาณข้อมูล ผ่านชิปเล็กจิ๋วนี้จุดต่อจุดไปเรื่อยๆ จนถึงปลายทางที่ต้องการดาวน์โหลดข้อมูลลงในเครื่องคอมพิวเตอร์เพื่อใช้ในการ วิเคราะห์ข้อมูล ข้อมูลที่ได้อาจจะเป็นการวัดอุณหภูมิ การเคลื่อนไหวของสิ่งมีชีวิต จับปริมาณมลพิษในอากาศ ปริมาณน้ำท่อแก๊สโดยใช้พลังงานแสงอาทิตย์หรือแบตเตอรี่ขนาดเล็กที่กินไฟน้อยมาก จึงสามารถฝังทิ้งไว้ในที่ห่างไกลได้เป็น 10 ปี ว่ากันว่าเทคโนโลยี Zigbee นี้จะช่วยทำให้บริษัทที่เกี่ยวข้องกับการส่งพลังงาน เช่น น้ำมัน ประปา น้ำในเขื่อน ท่อแก๊ส สามารถประหยัดการสูญเสียได้อย่างน้อย10-15%และในอนาคตอันใกล้นี้เมื่อเทคโนโลยีนาโนก้าวหน้ามากขึ้น เซ็นเซอร์ Zigbee จะมีขนาดเล็กเท่าหัวเข็มหมุด สามารถฝังได้แม้กับในร่างกายของสิ่งมีชีวิตก็ได้

ชื่อ ZigBee ได้มาจากพฤติกรรมการสื่อสารของผึ้ง โดยผึ้งจะบินแบบซิกแซ็ก และจะให้ข้อมูลข่าวสารระหว่างผึ้งด้วยกันที่เกี่ยวกับตำแหน่ง ระยะทาง และทิศทางของอาหารที่พวกมันกำลังหาอยู่

ZigBee ถูกสร้างขึ้นในการทำระบบเครือข่ายไร้สายส่วนบุคคล (WPAN) อยู่ภายใต้มาตรฐาน IEEE 802.15.4 โดยมาตรฐานนี้ใช้งานสำหรับการสื่อสารความเร็วต่ำ ใช้กำลังไฟฟ้าน้อย อุปกรณ์ราคาถูก และมีคุณสมบัติการจัดการตัวเองได้ เป็นเทคโนโลยีไร้สายที่ร่วมกันสื่อสารข้อมูลผ่านเซ็นเซอร์ขนาดเล็กจิ๋ว จำนวนเป็นพันๆ หมื่นๆ ชิ้นที่ฝังอยู่ตามส่วนต่างๆ ในอาคาร สำนักงาน โรงงาน หรือแม้แต่ในบ้าน

การทำงานของ ZigBee จะเป็นการรับ-ส่งคลื่นสัญญาณข้อมูล ผ่านชิปเล็กจิ๋วนี้จุดต่อจุดไปเรื่อยๆจนถึงปลายทางที่ต้องการดาวน์โหลดข้อมูลลงในเครื่องคอมพิวเตอร์เพื่อใช้ในการวิเคราะห์ข้อมูล ข้อมูลที่ได้อาจจะเป็นการวัดอุณหภูมิ การเคลื่อนไหวของสิ่งมีชีวิต จับปริมาณมลพิษในอากาศ ปริมาณน้ำ ท่อแก๊สโดยใช้พลังงานแสงอาทิตย์หรือแบตเตอรี่ขนาดเล็กที่กินไฟน้อยมาก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.1 ลักษณะการทำงานของZigbee

1. Coordinator

มีหน้าที่สร้างการสื่อสารเชื่อมโยงเครือข่ายระหว่าง End Device กับ Router หรือ Coordinator กับ Coordinator ด้วยกัน หรือ Coordinator กับ Router กำหนด address ให้กับ device ที่อยู่ในวงเครือข่าย ไม่ให้ซ้ำกัน ดูแลจัดการเรื่องการ Routing เส้นทาง ซึ่งเทียบได้กับ FFD

2. End Device

เป็นอุปกรณ์ปลายทางสุด ซึ่งจะใช้รับสัญญาณจาก Sensor ที่ปลายทาง โดยที่ใช้พลังงานต่ำในการทำงาน เทียบได้กับ RFD หรือ FFD บางกรณี ขึ้นอยู่กับ sensor ที่จะใช้งาน

3. Router

มีหน้าที่ รับส่งข้อมูล ในเส้นทางต่าง ๆ ของเครือข่ายซึ่งเทียบได้กับ FFD

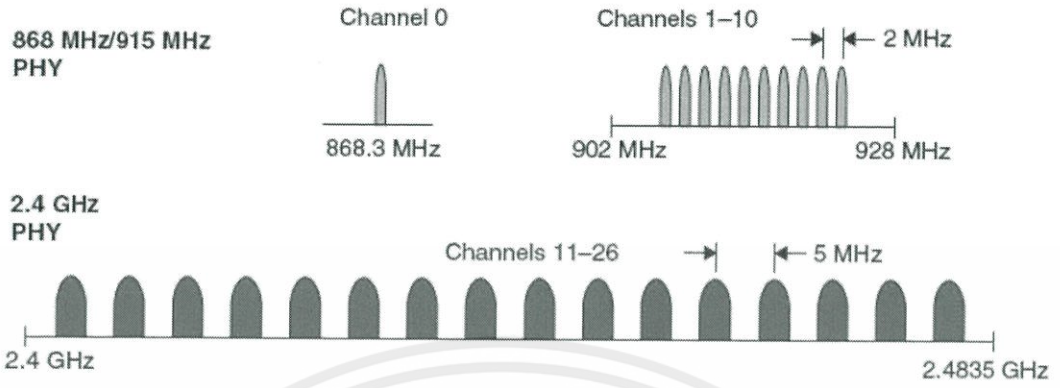
2.1.2 ย่านความถี่ใช้งานของ Zigbee

Zigbee กำหนด ย่านความถี่ใช้งานตามมาตรฐานไว้ 3 ย่านความถี่คือ ย่าน 2.4 GHz , ย่าน 915 MHz และย่าน 868 MHz โดยแต่ละย่านจะมีช่องสัญญาณ 16 ช่อง 10 ช่อง และ 1 ช่อง ตามลำดับ ส่วน อัตรารับส่งข้อมูล (ทางอากาศ) จะอยู่ที่ 250 Kbps,40 Kbps,20 Kbps

ตารางที่ 2.1 ย่านความถี่ใช้งานของ Zigbee

Channel	Frequency	Data rate
1 (0)	868 MHz	20 Kbps
10 (1-10)	915 MHz	40 Kbps
16 (11-26)	2.4 GHz	250 Kbps

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.1 Channel ของ Zigbee [3]

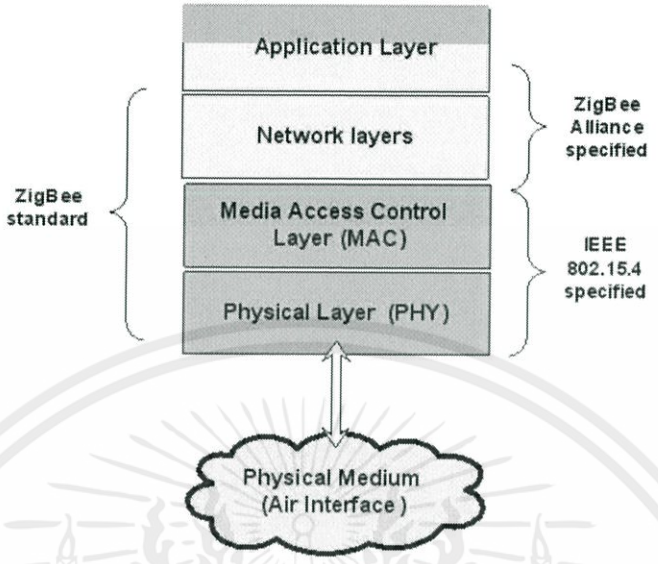
จากรูป 2.1 จะเห็นได้ว่า ที่ Channel 11-26 ของ Zigbee จะเพิ่มทีละ 5 MHz โดยที่จุดเริ่มต้น จะมีความถี่ 2.40 GHz ดังนั้นจะสรุปได้ว่า

$$\text{Frequency} = 2.40\ \text{GHz} + (\text{Ch} - 10) \times 5\ \text{MHz} \qquad (2.1)$$

2.1.3 มาตรฐานโปรโตคอล Zigbee

มาตรฐานIEEE802.15.4เป็นมาตรฐานสำหรับการสื่อสารระยะใกล้ที่เน้นการใช้พลังงานต่ำ และเน้นการสื่อสารที่ทนต่อสภาพสัญญาณรบกวนสูง และเน้นที่มีความง่ายเพื่อให้ชิบการสื่อสารมีราคาถูก ส่วนมาตรฐาน Zigbee เป็นมาตรฐานที่ออกแบบมาเฉพาะสำหรับการติดต่อเครื่องเซ็นเซอร์ชนิดต่างๆเพื่อการพัฒนาแอบพลิเคชัน เช่น เครื่องเซ็นเซอร์สำหรับ Home Automation การสื่อสารแบบ Zigbee เป็นแบบ multi-hop routing ที่สามารถส่งข้อมูลไปยังเครื่องเซ็นเซอร์ที่ต้องการโดยผ่านเครื่องเซ็นเซอร์ตัวอื่นๆ ซึ่งคุณสมบัตินี้ไม่ได้รับการสนับสนุนในบลูทูธ การสื่อสารแบบ Zigbee ช่วยให้ขยายรัศมีการส่งของข้อมูลได้ออกไปเรื่อย เมื่อมีการเพิ่มจำนวนของเครื่องเซ็นเซอร์ด้วย Zigbee มาตรฐาน IEEE 802.15.4 ที่ถูกกำหนดไว้ใน ชั้น Physical layer มีอยู่ 3 ความถี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.2 Zigbee Protocol [4]

2.1.4 **สายอากาศของ Zigbee**

1. แบบ Wire Antenna

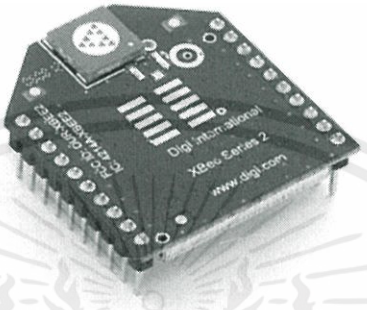
แบบ Wire Antenna นั้นจะสามารถส่งสัญญาณได้ไกลกว่าแบบ Chip Antenna แต่ถ้านำไปประกอบใส่กล่อง จะมีข้อเสียตรงที่ไม่สามารถใส่กล่องเล็กๆ ได้จะต้องเหลือที่ให้สายอากาศ

รูปที่ 2.3 Zigbee แบบ Wire Antenna [5]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. แบบ Chip Anttena

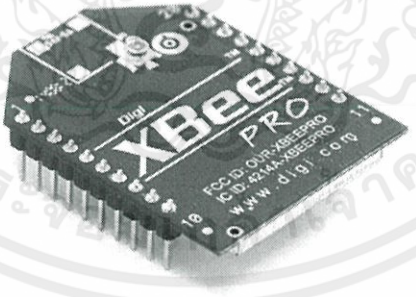
แบบ Chip Antenna นั้นจะส่งสัญญาณได้ระยะทางสั้นกว่าแบบอื่นจากคุณสมบัติที่กำหนดมาให้ แต่สามารถที่จะซ่อนไว้เก็บไว้ในกล่องขนาดเล็กได้อย่างสวยงาม



รูปที่ 2.4 Zigbee แบบ Chip Antenna [5]

3. แบบ UFL connector

แบบ UFL นั้นจะสามารถที่จะซ่อนตัว Zigbee ไว้ในกล่องได้ แล้วต่อสายอากาศโดยใช้ตัวแปลง UFL to SMA ต่อออกมา แล้วต่อกับสายอากาศ 2.4 Ghz ก็จะใช้งานได้ ข้อดีของ UFL คือ เราสามารถเก็บอุปกรณ์ Zigbee ให้ปลอดภัยจากฝน ฝุ่น น้ำ ต่างๆ ได้ อีกทั้งยังสามารถตั้งสายอากาศและขยายสัญญาณเองให้ส่งได้ไกลขึ้นอีกด้วย

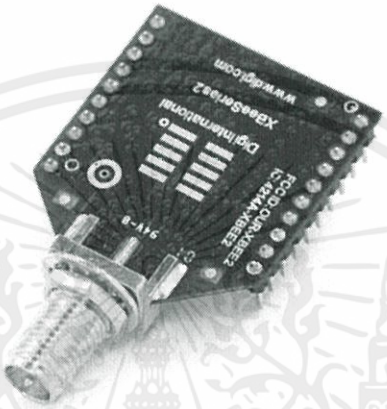


รูปที่ 2.5 Zigbee แบบ UFL connector [5]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4. แบบ SMA Connnector

แบบ SMA Connector แบบนี้จะคล้ายๆ กับแบบ UFL Connector แต่ว่าไม่ต้องซื้อสายแปลง เวลาทำกล่องต้องให้ Zigbee อยู่ริมกล่อง แต่ถ้า Connector SMA นี้เสียหรือขึ้นสนิมหรือเกิดอะไรขึ้นมา จะต้องบัดกรีออกเปลี่ยนเท่านั้น



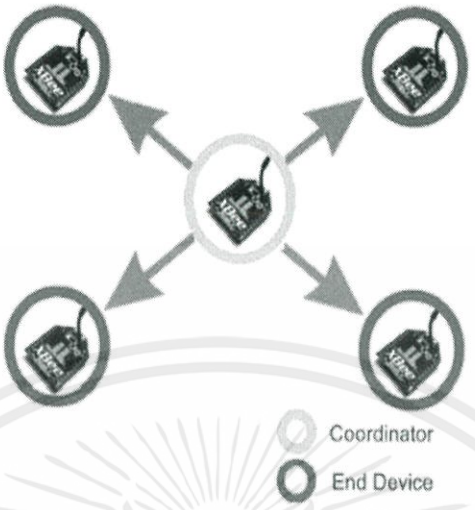
รูปที่ 2.6 Zigbee แบบ SMA Connnector [5]

2.1.5 รูปแบบเครือข่ายของ Zigbee

1. Star (Broadcast)

การเชื่อมต่อแบบ Star (Broadcast) เป็นการรับส่งข้อมูลแบบ ไม่เฉพาะเจาะจง จุดหมายปลายทาง หรือ Zigbee ทุกตัวที่อยู่ในระบบเครือข่ายเดียวกันสามารถ รับข้อมูลทุกข้อมูลได้ทุกตัว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



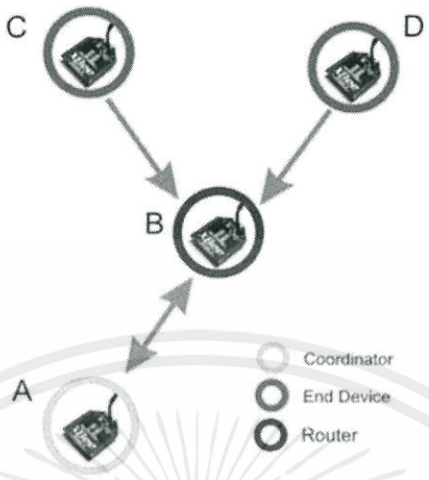
รูปที่ 2.7 Star (Broadcast) [5]

จากรูปที่ 2.7 การใช้งานแบบ Star จะประกอบไปด้วย Zigbee ที่ทำงานเป็น 2 รูปแบบคือ แบบที่1 เป็น Coordinator ทำหน้าที่สร้างเครือข่าย และ แบบที่ 2 เป็น End Device ทำหน้าที่เป็นลูกข่าย

2. Cluster Tree

เป็นการรับส่งข้อมูลแบบ ส่งผ่าน เช่น A ต้องการติดต่อ กับ C แต่ C อยู่ไกลจาก A จน A ไม่สามารถ ติดต่อกับ C ได้ แต่มี B ที่อยู่ระหว่าง A กับ C ดังนั้น Cluster Tree จะใช้ B เป็นเหมือน ตัวกลาง เชื่อมการติดต่อ (Repeater) ระหว่าง A กับ C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



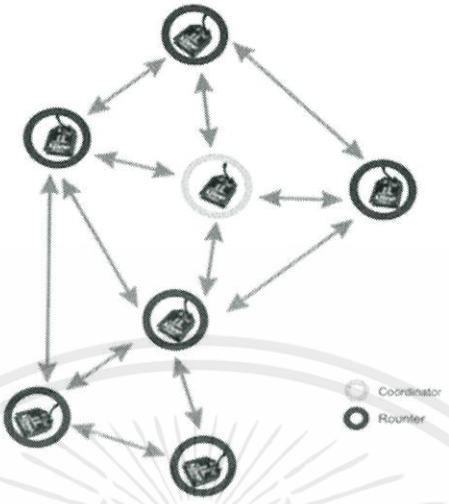
รูปที่ 2.8 Cluster Tree [5]

จากรูปที่2.8 จะพบว่า มีการใช้งาน Zigbee อยู่ 3 ลักษณะด้วยกันคือ Coordinator ทำหน้าที่สร้างเครือข่าย, End Device ทำหน้าที่เป็นลูกข่าย, Routers ทำหน้าที่รับส่งข้อมูล ในเส้นทางต่าง ๆ ของเครือข่าย

3. Mesh

การเชื่อต่อเครือข่ายแบบ Mesh เป็นโครงข่ายที่มีประสิทธิภาพสูงเนื่องจากข้อมูลสามารถส่งไปถึงเป้าหมายได้หลายทาง ทำให้ ระบบนี้สามารถรับส่งข้อมูลไปยังจุดหมายปลายทางได้ แม้จะเกิดความเสียหายของระบบในบางส่วนก็ตาม (ขึ้นอยู่กับ การออกแบบระบบของผู้ใช้ด้วย) ระบบนี้จึงเป็นระบบที่ได้รับความนิยมเป็นอย่างมาก

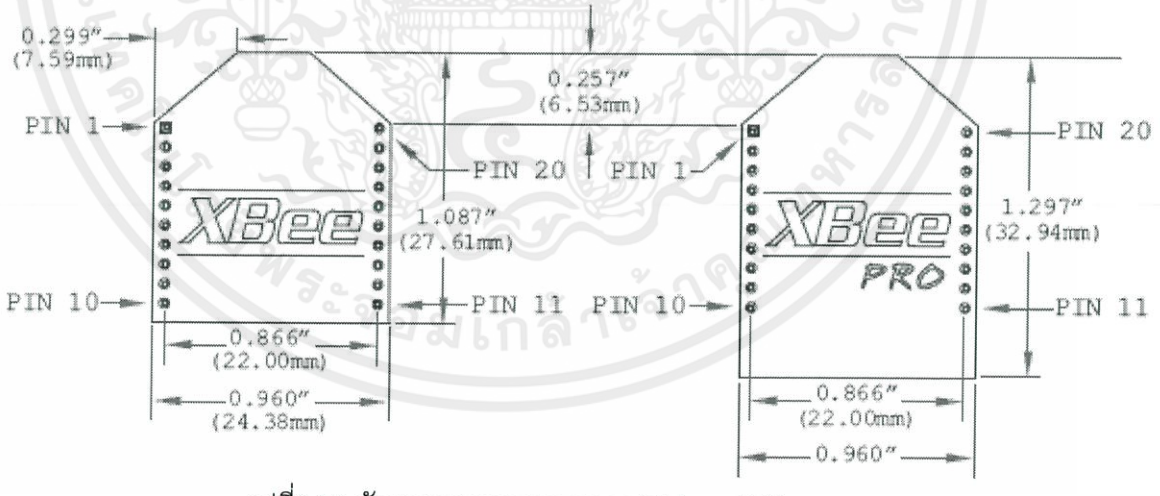
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 Mesh [5]

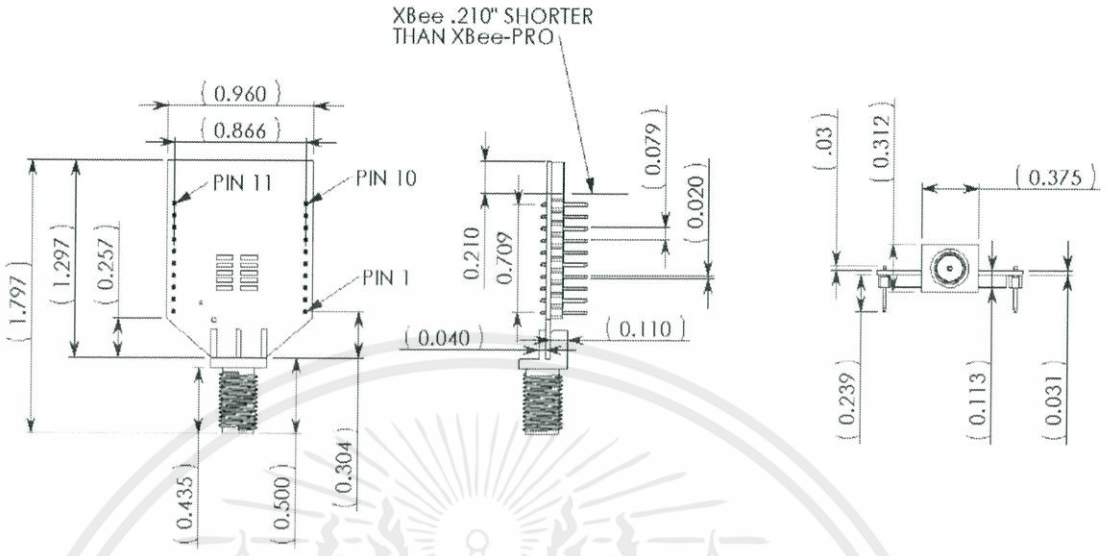
จากรูปที่ 2.9 จะพบว่า มีการใช้งาน Zigbee อยู่ 2 ลักษณะด้วยกันคือ Coordinator ทำหน้าที่สร้างเครือข่าย Routers รับส่งข้อมูล ในเส้นทางต่าง ๆ ของเครือข่าย

2.1.6 ลักษณะทางกายภาพ



รูปที่2.10 ลักษณะทางกายภาพของ Zigbee [10]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 ลักษณะทาง Machanical ของ Zigbee [10]

2.2 PIC16F877

Microcontroller ตระกูล PIC ของบริษัทไมโครชิพ (MICROCHIP) โดยเฉพาะเบอร์ PIC16F877 เป็น Microcontroller ที่กำลังได้รับความนิยมอย่างแพร่หลาย เนื่องจากเป็น Microcontroller แบบแรกๆที่มีการประมวลผลแบบ RISC Processor (RISC : Reduced Instruction Set Computer) โดยใช้คำสั่งการประมวลผลเพียง 33-35 คำสั่ง และใช้เวลาในการประมวลคำสั่งเพียง 1 หรือ 2 machine cycle ต่อคำสั่งเท่านั้น การประมวลผลคำสั่งเป็นลักษณะ Pipe Line คือ ขณะประมวลผลคำสั่งแรกจะทำการโหลดคำสั่งถัดไปมาเตรียมรอไว้ ทำให้การทำงานที่รวดเร็วมาก นอกจากนั้นในตัวโครงสร้างยังประกอบด้วยฮาร์ดแวร์ฟังก์ชั่นโมดูลสำหรับใช้งานพิเศษต่างๆมากมาย ได้แก่ โมดูล Analog to Digital Converter , USART , Timer/Counter , SPI , Compare/Capture/PWM , I2C เป็นต้น โมดูลเหล่านี้ล้วนเพิ่มขีดความสามารถให้นำไปใช้งานได้อย่างหลากหลาย ประกอบกับมีเครื่องมือในการพัฒนาที่เพรียบพร้อม คุณสมบัติของ PIC16F877 มีดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- มีคำสั่งให้ใช้งาน 35 คำสั่ง
- ใชความถี่ออสซิลเลเตอรไดสูงสุด 20 MHz
- มีหนวยความจำโปรแกรม Flash Memory ขนาด 8 K word (14 – bit words)
- มีหนวยความจำขอมูลแบบ RAM 368 Bytes
- มีหนวยความจำขอมูลแบบ EEPROM 256 Byte
- มีการตอบสนอง interrupt ทั้งหมด 14 แหลง
- สามารถเลือกระดับการปองกันขอมูล (Code Protection) ได
- มีโหมดประหยัดพลังงาน (Sleep Mode)
- สามารถเลือกแหลงสัญญาณนาฬิกาได้หลายโหมด XT RC และออสซิเลเตอรพลังงานต่ำ
- มีฟงกชั่นการรักษาเสถียรภาพการทำงาน ไดแกPOR , PWRT, OST, BOR และWDT
- การโปรแกรมตัวชิพแบบ ICSP (ICSP : In-Circuit Serial Programming)
- สามารถทำงานที่ไฟเลี้ยงวงจรตั้งแต2.0 V ถึง 5.5 V
- ขาพอรท I/O แตละขา สามารถรับและปลอยกระแสไดสูงสุด 25 mA
- มีโมดูล Timer / Counter ใชงานทั้งหมด3 ตัว Timer 0 , Timer 1, และ Timer 2
- มีโมดูลCCP (CCP : Compare / Capture / PWM) จำนวน 2 ชุด
- มีโมดูล Analog to Digital Converter ความละเอียดขนาด 8 บิท และ 10 บิท จำนวน 8 ชอง ภายในตัวชิพ
- มีโมดูลสื่อสารอนุกรมแบบ USART (USART : Universal Synchronous Asynchronous Receiver / Transmitter)
- มีพอรท I/O จำนวน 5 พอรท ไดแก พอรท A , B , C , D , และ E มีขาI/O รวมกัน 33 ขา ดังนี้คือ

PORTA มี5 ขา RA0 ถึง RA 5 6 7

PORTB มี8 ขา RB0 ถึง RB7

PORTC มี8 ขา RC0 ถึง RC7

PORTD มี8 ขา RD0 ถึง RD7

PORTE มี3 ขา RE0 ถึง RE2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

รูปที่ 2.12 PIC16F877 [7]

2.2.1 RS232

RS-232C เป็นมาตรฐานที่ตั้งขึ้นมานานมาก (“C” เป็นเวอร์ชันปัจจุบัน) ที่อธิบายการอินเตอร์ทางกายภาคและโปรโตคอลสำหรับการสื่อสารข้อมูลความเร็วต่ำระหว่างเครื่องคอมพิวเตอร์กับอุปกรณ์เกี่ยวข้อง สิ่งนี้ได้รับการกำหนดโดยกลุ่มการค้าอุตสาหกรรมคือ Electronic Industries Association (EIA) เริ่มต้นสำหรับเครื่องพิมพ์ทางไกล RS-232C เป็นการอินเตอร์เฟซที่เครื่องคอมพิวเตอร์ใช้สนทนากับและแลกเปลี่ยนข้อมูลด้วยโมเด็มและอุปกรณ์อนุกรม บางแห่งเครื่องคอมพิวเตอร์ส่วนบุคคล ตามปกติบนชิป Universal Asynchronous Receiver/Transmitter (UART) บนแผ่นเมนบอร์ด ข้อมูลจากเครื่องคอมพิวเตอร์ผู้ใช้ได้รับการส่งไปโมเด็มภายในหรือภายนอก (หรืออุปกรณ์อนุกรม) จากอินเตอร์เฟซ Data Terminal Equipment (DTE) เนื่องจากข้อมูลในเครื่องคอมพิวเตอร์ไหลผ่านวงจรขนานและอุปกรณ์อนุกรมจัดการเพียงหนึ่งบิตในแต่ละครั้ง ชิป UART แปลงกลุ่มของบิตในการขนานกับ serial stream ของบิต ตัวแทน DTE ของเครื่องคอมพิวเตอร์ส่วนบุคคล สิ่งนี้สื่อสารกับโมเด็มหรืออุปกรณ์อนุกรมด้วยตามมาตรฐานของมาตรฐาน RS-232C มีอินเตอร์เฟซอย่างสมบูรณ์ที่เรียกว่าอินเตอร์เฟซ Data Communications Equipment (DCE)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3 Relay

เป็นอุปกรณ์ควบคุมวงจรไฟฟ้าที่มีการทำงานในลักษณะเป็นเครื่องกลไฟฟ้าที่นิยมใช้ในวงจรควบคุมแบบบางๆกันอย่างแพร่หลายโดยโครงสร้างพื้นฐาน และการทำงานของรีเลย์จะประกอบไปด้วยขดลวดตัวนำ และแกนโลหะที่สามารถเคลื่อนที่ขึ้นลงได้ เรียกว่าอาร์มาเจอร์ โดยอาร์มาเจอร์ จะมีหน้าที่เปิดปิดหน้าสัมผัสของรีเลย์การทำงานของรีเลย์ จะเริ่มทำงานได้เมื่อมีกระแสไฟฟ้าไหลผ่านไปที่ขดลวดตัวนำทำให้เกิดสนามแม่เหล็ก ไปดึงดูดแกนของอาร์มาเจอร์ถ้าแรงดึงดูดที่เกิดจาก สนามแม่เหล็กสามารถชนะแรงดึงของสปริงได้ก็จะดึงแกนของอาร์มาเจอร์ให้หน้าสัมผัสของรีเลย์มาอยู่ในตำแหน่งอีกทางหนึ่ง แต่ถ้าแรงดึงดูดที่เกิดจากสนามแม่เหล็ก ไม่สามารถชนะแรงดึงของสปริงได้หน้าสัมผัส ของรีเลย์ก็จะอยู่ในตำแหน่งเดิมรีเลย์ จะมีหน้าสัมผัสอยู่สองแบบ คือแบบปกติเปิดและแบบปกติปิด รีเลย์แบบปกติเปิดหน้าสัมผัสของรีเลย์ จะเปิดเมื่อไม่มีกระแสไฟฟ้าไหลผ่านไปยังขดลวดของรีเลย์ และหน้าสัมผัสจะปิดเมื่อมี กระแสไฟฟ้าไหลผ่านไปที่ขดลวดของรีเลย์ซึ่งการทำงานก็จะตรงกันข้ามกัน ในรีเลย์แบบปกติปิดรีเลย์มีหลายชนิดด้วยกัน โดยมากรีเลย์จะถูกนำมาใช้ในวงจรการควบคุมมอเตอร์ไฟฟ้า ในลักษณะของการหน่วงเวลาเพื่อทำให้เกิดการทำงานของวงจรควบคุมเป็นไปตามลำดับหรือใช้เพื่อป้องกันอันตรายที่จะเกิดขึ้นภายในวงจรควบคุม

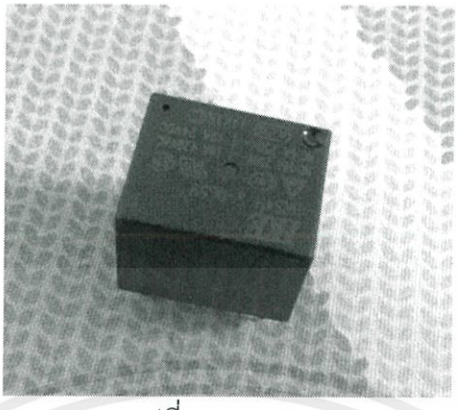
2.3.1 หน้าที่ของรีเลย์

หน้าที่ของรีเลย์ คือ เป็นอุปกรณ์ไฟฟ้าที่ใช้ตรวจสอบสภาพการณ์ของทุกส่วนในระบบกำลังไฟฟ้าอยู่ตลอดเวลา หากระบบมีการทำงานที่ผิดปกติ รีเลย์จะเป็นตัวสั่งการให้ตัดส่วนที่ลัดวงจรหรือส่วนที่ทำงานผิดปกติออกจากระบบทันที โดยเซอร์กิตเบรกเกอร์ จะเป็นตัวที่ตัดส่วนที่เกิดความผิดพลาดออกจากระบบจริงๆ ซึ่งจะเห็นว่ารีเลย์เป็นเพียงตัวส่งสัญญาณสั่งการเท่านั้นและจะต้องทำงานร่วมกับเซอร์กิตเบรกเกอร์

กรณีที่มีเหตุผิดปกติเกิดขึ้นเพียงเล็กน้อย รีเลย์จะเป็นเพียงอุปกรณ์ที่ทำหน้าที่ส่งสัญญาณเตือนภัยให้พนักงานที่ทำหน้าที่ควบคุมทราบ เพื่อให้พนักงานทำการแก้ไขให้ระบบสามารถทำงานได้ตามปกติ โดยรีเลย์จะยังไม่สั่งการให้เซอร์กิตเบรกเกอร์ตัดวงจรที่ผิดปกติออก เพียงแต่ส่งสัญญาณเตือนภัยเท่านั้น

รีเลย์ยังสามารถบอกตำแหน่งที่เกิดฟอลต์ และสาเหตุของฟอลต์โดยใช้ข้อมูลจากรีเลย์ไปพิจารณาร่วมกับผลของออสซิลโลสโคบ เพื่อวิเคราะห์หาสาเหตุของฟอลต์ได้อย่างถูกต้องและรวดเร็ว

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 Relay

2.3.2 ข้อคำถึงในการใช้งานรีเลย์ทั่วไป

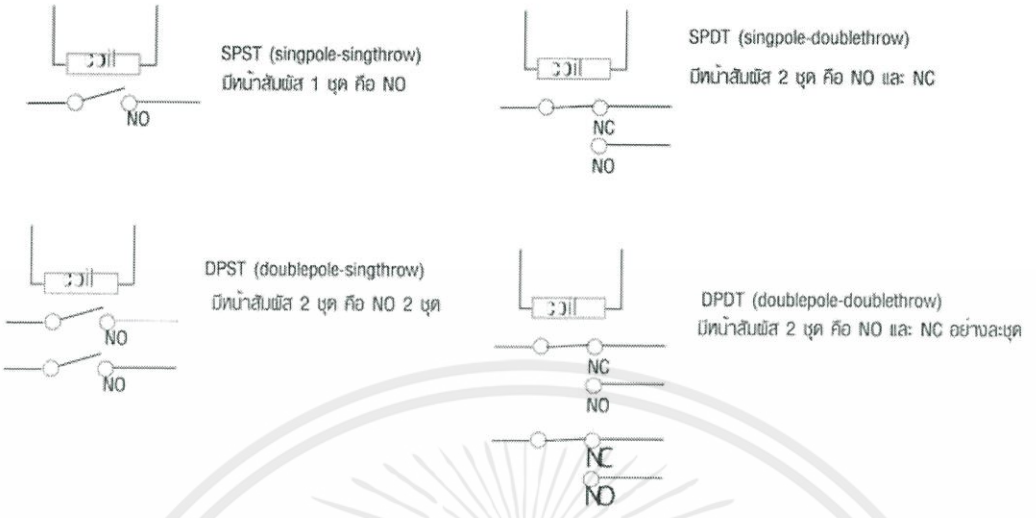
1. แรงดันใช้งาน หรือแรงดันที่ทำให้รีเลย์ทำงานได้ หากเราดูที่ตัวรีเลย์จะระบุค่าแรงดันใช้งานไว้ (หากใช้ในงานอิเล็กทรอนิกส์ ส่วนมากจะใช้แรงดันกระแสตรงในการใช้งาน) เช่น 12VDC คือต้องใช้แรงดันที่ 12 VDC เท่านั้นหากใช้มากกว่านี้ ขดลวดภายใน ตัวรีเลย์อาจจะขาดได้ หรือหากใช้แรงดันต่ำกว่ามาก รีเลย์จะไม่ทำงาน ส่วนในการต่อวงจรนั้นสามารถต่อขั้วใดก็ได้ครับ เพราะตัวรีเลย์ จะไม่ระบุขั้วต่อไว้ (นอกจากชนิดพิเศษ)

2. การใช้งานกระแสผ่านหน้าสัมผัส ซึ่งที่ตัวรีเลย์จะระบุไว้ เช่น 10A 220AC คือหน้าสัมผัสของรีเลย์นั้นสามาถทนกระแสได้ 10 A ที่ 220VAC ครับ แต่การใช้ก็ควรจะใช้งานที่ระดับกระแสต่ำกว่านี้จะเป็นการดีกว่าครับ เพราะถ้ากระแสมากหน้าสัมผัส ของรีเลย์จะละลายเสียหายได

3. จำนานหน้าสัมผัสการใช้งาน ควรดูว่ารีเลย์นั้นมีหน้าสัมผัสให้ใช้งานจำนวนกี่อันและมีขั้วคอมมอนด้วยหรือเปล่า

จำนวนหน้าสัมผัสของรีเลย์ ปกติแล้วรีเลย์จะมีหน้าสัมผัสและการเรียกจำนวนหน้าสัมผัสดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 หน้าสัมผัสของ Relay [9]

2.3.3 การจัดขาของรีเลย์

ประกอบไปด้วยตำแหน่งต่างๆ ดังนี้

- ขาจ่ายแรงดัน มี 2 ขาสำหรับแรงดันใช้งาน
- ขา C (Common) เป็นขาต่อระหว่าง No และ Nc
- ขา No (Normally Opened) ปกติขานี้จะเปิดเอาไว้ จะทำงานเมื่อเราป้อนแรงดันให้รีเลย์
- ขา Nc (Normally Closed) ปกติขานี้จะต่อกับขา C ในกรณีที่เราไม่ได้จ่ายแรงดัน หน้าสัมผัสของ C และ Nc จะต่อถึงกัน

2.3.4 ประเภทของรีเลย์แบ่งตามการใช้งาน

เป็นอุปกรณ์ทำหน้าที่เป็นสวิตช์มีหลักการทำงานคล้ายกับ ขดลวดแม่เหล็กไฟฟ้าหรือโซลินอยด์ (Solenoid) รีเลย์ใช้ในการควบคุมวงจร ไฟฟ้าได้อย่างหลากหลาย รีเลย์เป็นสวิตช์ควบคุมที่ทำงานด้วยไฟฟ้า แบ่งออกตามลักษณะการใช้งานได้เป็น 2 ประเภทคือ

1. รีเลย์กำลัง (Power relay) หรือมักเรียกกันว่าคอนแทกเตอร์ (Contactor or Magneticcontactor) ใช้ในการควบคุมไฟฟ้ากำลัง มีขนาดใหญ่กว่ารีเลย์ธรรมดา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. รีเลย์ควบคุม (Control Relay) มีขนาดเล็กกำลังไฟฟ้าต่ำ ใช้ในวงจรควบคุมทั่วไปที่มีกำลังไฟฟ้าไม่มากนัก หรือเพื่อการควบคุมรีเลย์หรือคอนแทกเตอร์ขนาดใหญ่ รีเลย์ควบคุม บางทีเรียกกันง่าย ๆ ว่า "รีเลย์"

2.3.5 ชนิดของรีเลย์

การแบ่งชนิดของรีเลย์สามารถแบ่งได้ 11 แบบ คือ ชนิดของรีเลย์แบ่งตามลักษณะของคอยล์ หรือแบ่งตามลักษณะการใช้งาน (Application) ได้แก่รีเลย์ดังต่อไปนี้

1. รีเลย์กระแส (Current relay) คือ รีเลย์ที่ทำงานโดยใช้กระแสมีทั้งชนิดกระแสขาด (Under- current) และกระแสเกิน (Over current)

2. รีเลย์แรงดัน (Voltage relay) คือ รีเลย์ ที่ทำงานโดยใช้แรงดันมีทั้งชนิดแรงดันขาด (Under-voltage) และ แรงดันเกิน (Over voltage)

3. รีเลย์ช่วย (Auxiliary relay) คือ รีเลย์ที่เวลาใช้งานจะต้องประกอบเข้ากับรีเลย์ชนิดอื่น จึงจะทำงานได้

4. รีเลย์กำลัง (Power relay) คือ รีเลย์ที่รวมเอาคุณสมบัติของรีเลย์กระแส และรีเลย์แรงดันเข้าด้วยกัน

5. รีเลย์เวลา (Time relay) คือ รีเลย์ที่ทำงานโดยมีเวลาเข้ามาเกี่ยวข้องด้วย ซึ่งมีอยู่ด้วยกัน 4 แบบ คือ

- รีเลย์กระแสเกินชนิดเวลาผกผันกับกระแส (Inverse time over current relay) คือ รีเลย์ ที่มีเวลาทำงานเป็นส่วนกลับกับกระแส
- รีเลย์กระแสเกินชนิดทำงานทันที (Instantaneous over current relay) คือรีเลย์ที่ทำงานทันทีทันใดเมื่อมีกระแสไหลผ่านเกินกว่าที่กำหนดที่ตั้งไว้
- รีเลย์แบบดิฟฟินิตไทม์เล็ก (Definite time lag relay) คือ รีเลย์ ที่มีเวลาการทำงานไม่ขึ้นอยู่กับความมากน้อยของกระแสหรือค่าไฟฟ้าอื่นๆ ที่ทำให้เกิดงานขึ้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- รีเลย์แบบอินเวอสดิฟฟินิตมินิมั่มไทม์เล็ก (Inverse definite time lag relay) คือ รีเลย์ ที่ทำงานโดยรวมเอาคุณสมบัติของเวลาผกผันกับกระแส (Inverse time) และ แบบดิฟฟินิตไทม์แล็ก (Definite time lag relay) เข้าด้วยกัน

6. รีเลย์กระแสต่าง (Differential relay) คือ รีเลย์ที่ทำงานโดยอาศัยผลต่างของกระแส

7. รีเลย์มีทิศ (Directional relay) คือรีเลย์ที่ทำงานเมื่อมีกระแสไหลผิดทิศทาง มีแบบรีเลย์กำลังมีทิศ (Directional power relay) และรีเลย์กระแสมีทิศ (Directional current relay)

8. รีเลย์ระยะทาง (Distance relay) คือ รีเลย์ระยะทางมีแบบต่างๆ ดังนี้

- รีแอกแตนซ์รีเลย์ (Reactance relay)
- อิมพีแดนซ์รีเลย์ (Impedance relay)
- โมห์รีเลย์ (Mho relay)
- โอห์มรีเลย์ (Ohm relay)
- โพลาไรซ์โมห์รีเลย์ (Polaized mho relay)
- ออฟเซทโมห์รีเลย์ (Off set mho relay)

9. รีเลย์อุณหภูมิ (Temperature relay) คือ รีเลย์ที่ทำงานตามอุณหภูมิที่ตั้งไว้

10. รีเลย์ความถี่ (Frequency relay) คือ รีเลย์ที่ทำงานเมื่อความถี่ของระบบต่ำกว่าหรือมากกว่าที่ตั้งไว้

11. บูคโฮลซ์รีเลย์ (Buchholz ‘s relay) คือรีเลย์ที่ทำงานด้วยก๊าซ ใช้กับหม้อแปลงที่แช่อยู่ในน้ำมันเมื่อเกิด ฟอลต์ ขึ้นภายในหม้อแปลง จะทำให้น้ำมันแตกตัวและเกิดก๊าซขึ้นภายในไปดันหน้าสัมผัส ให้รีเลย์ทำงาน

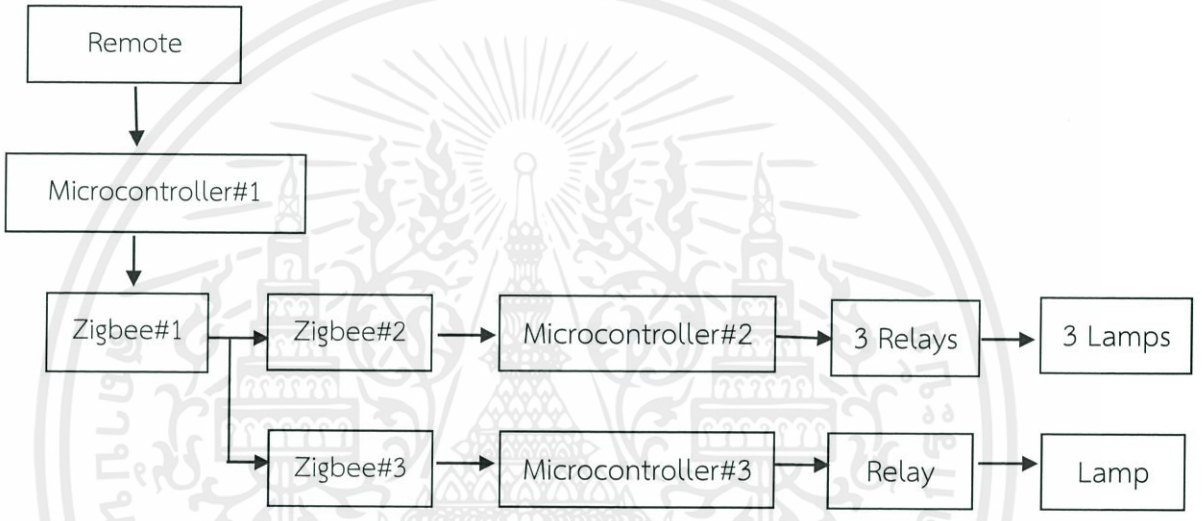
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

การออกแบบและการจัดทำปริญญานิพนธ์

3.1 การออกแบบวงจรในโครงงาน

3.1.1 บล็อกไดอะแกรมของโครงงาน



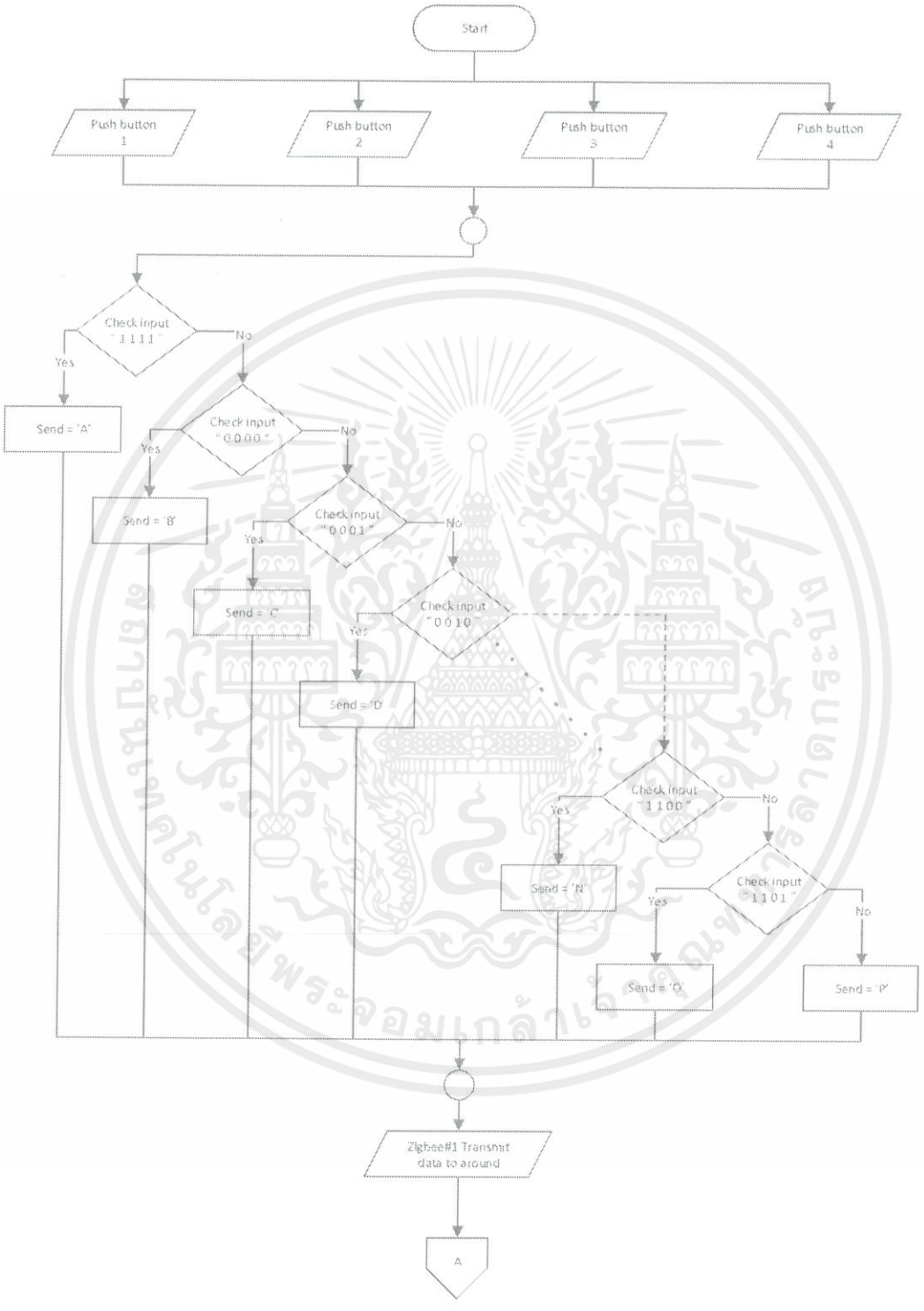
รูปที่ 3.1 Block Diagram

หลักการทำงานของบล็อกไดอะแกรม

1. Push Button ที่ Remote เพื่อส่งสัญญาณไป Microcontroller#1
2. Microcontroller#1 ส่งข้อมูลไปยัง Zigbee#1
3. Zigbee#1 ส่งข้อมูลไปยัง Zigbee#2 และ Zigbee#3
4. Zigbee#2 และ Zigbee#3 ส่งข้อมูลที่ได้รับมาไปยัง Microcontroller เพื่อนำไปประมวลผล
5. Microcontroller จะส่งคำสั่งควบคุม Relay เพื่อควบคุมการเปิด-ปิดหลอดไฟ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 Flowchart



รูปที่ 3.2 Flowchart (Transmitter)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

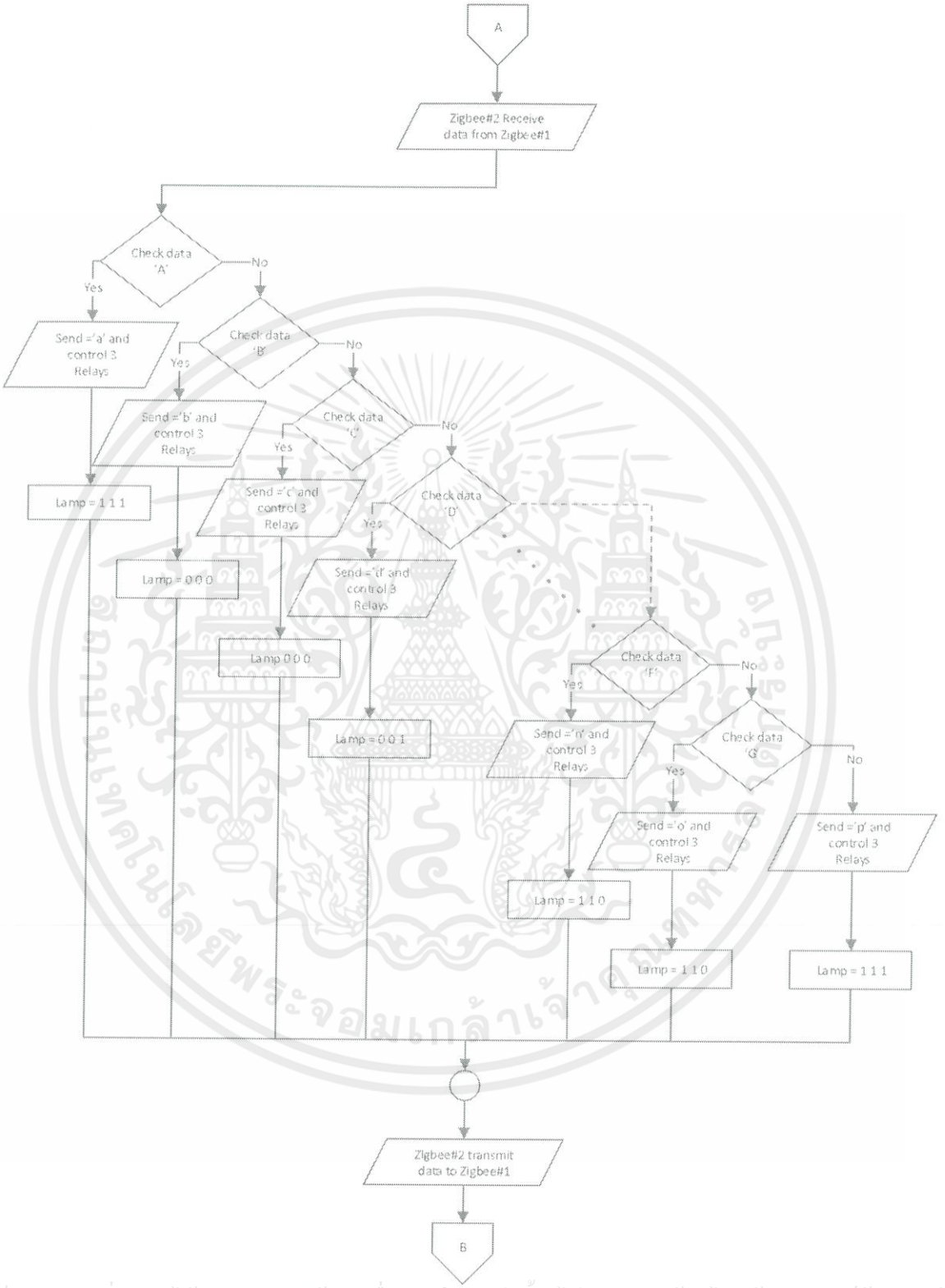
ขั้นตอนการทำงานของ Flowchart (Transmitter)

1. เมื่อทำการเริ่มต้นของวงจรจะทำการกดปุ่ม 1, 2, 3, 4

2. ทำการเช็คข้อมูลขาเข้า 1111, 0000, 0001, 0010, ..., 1100, 1101

3. ส่งข้อมูลเป็นตัวอักษร ‘A’, ‘B’, ‘C’, ‘D’, ‘E’, ‘F’, ‘G’, ‘H’, ’I’, ’J’, ’K’, ’L’, ’M’, ’N’, ’O’ และ ’P’ ไปยังวงจรฝั่งส่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 Flowchart (1st Receiver)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

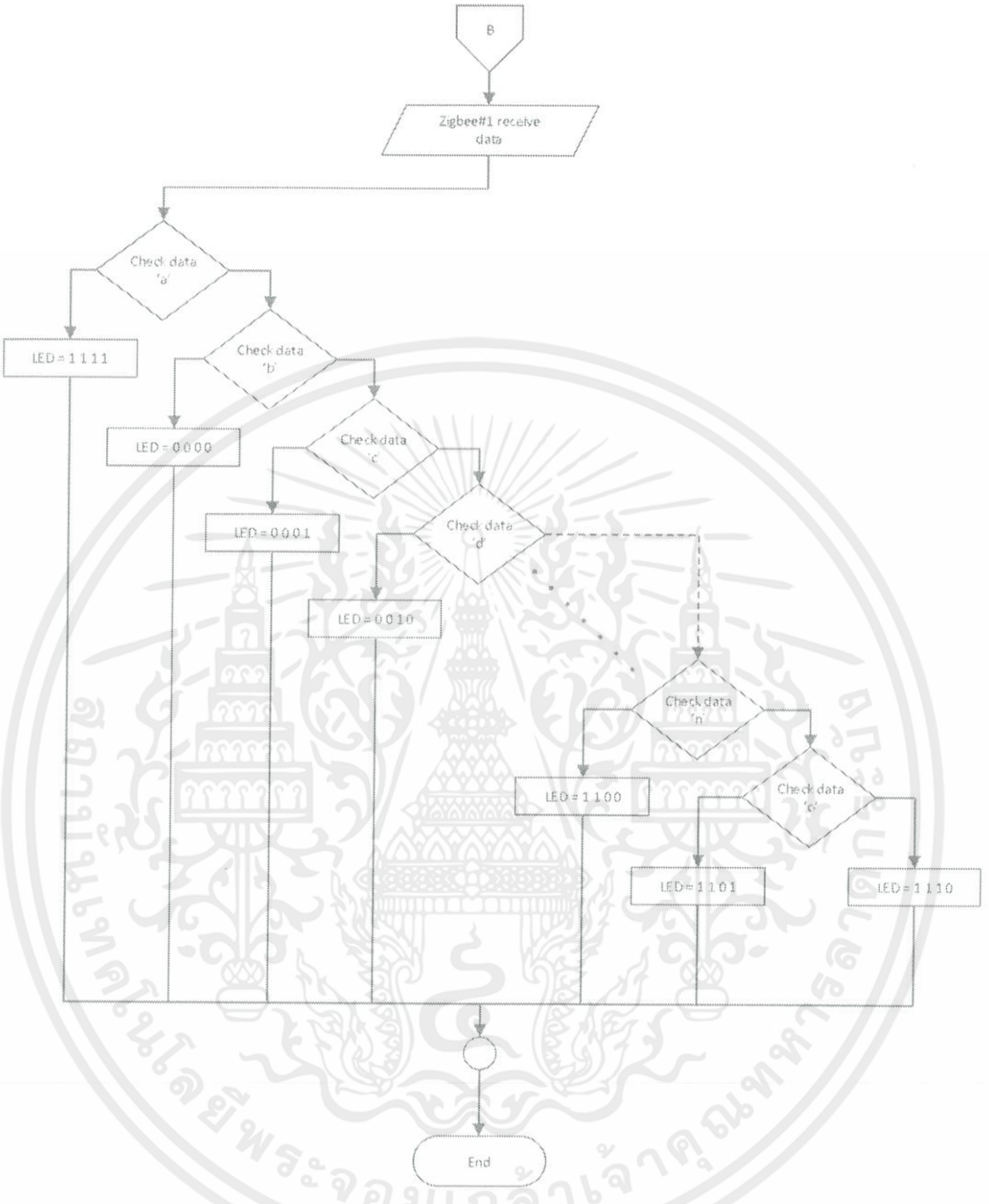
ขั้นตอนการทำงานของ Flowchart (1st Receiver)

1. Zigbee#2 จะทำการรับข้อมูลมากจาก Zigbee#1 และทำการเช็คข้อมูลตัวอักษร ‘A’, ‘B’, ‘C’, ‘D’, ‘E’, ‘F’, ‘G’, ‘H’, ’I’, ’J’, ’K’, ’L’, ’M’, ’N’, ’O’ และ ’P’

2. ทำการส่งตัวอักษรตรวจสอบการส่งข้อมูล ‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘f’, ‘g’, ‘h’, ‘I’, ‘j’, ‘k’, ‘l’, ‘m’, ‘n’, ‘o’, และ ‘p’ และควบคุมรีเลย์ทั้ง 3 ตัว สำหรับควบคุมการเปิด – ปิดหลอดไฟ

3. Zigbee#2 ส่งข้อมูลกลับไปยัง Zigbee#1 เพื่อทำการทดสอบว่ามีการส่งข้อมูลกลับมาจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



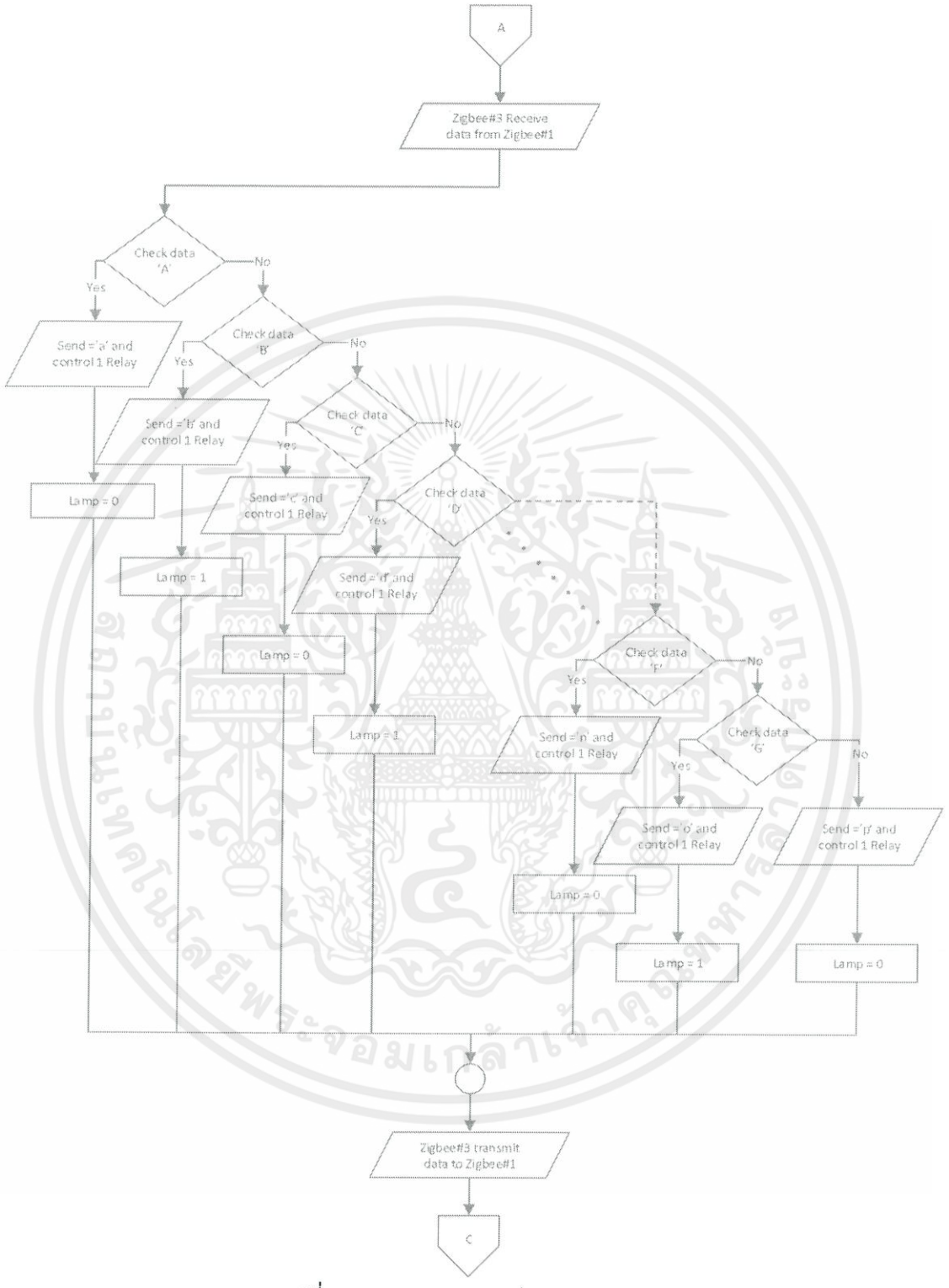
รูปที่ 3.4 Flowchart แสดงการส่งข้อมูลกลับจาก 1st Receiver

ขั้นตอนการทำงานของ Flowchart แสดงการส่งข้อมูลกลับจาก 1st Receiver

1. Zigbee#1 จะทำการรับข้อมูลมากจาก Zigbee#2 (วงจรฝั่งรับที่ 1) และทำการเช็คข้อมูลตัวอักษร ‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘f’, ‘g’, ‘h’, ‘I’, ‘j’, ‘k’, ‘l’, ‘m’, ‘n’, ‘o’, และ ‘p’

2. ทำการส่งตัวอักษรตรวจสอบการส่งข้อมูลไปที่ไฟ LED เพื่อแสดงว่ามีการส่งข้อมูลกลับมาจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.5 Flowchart (2nd Receiver)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

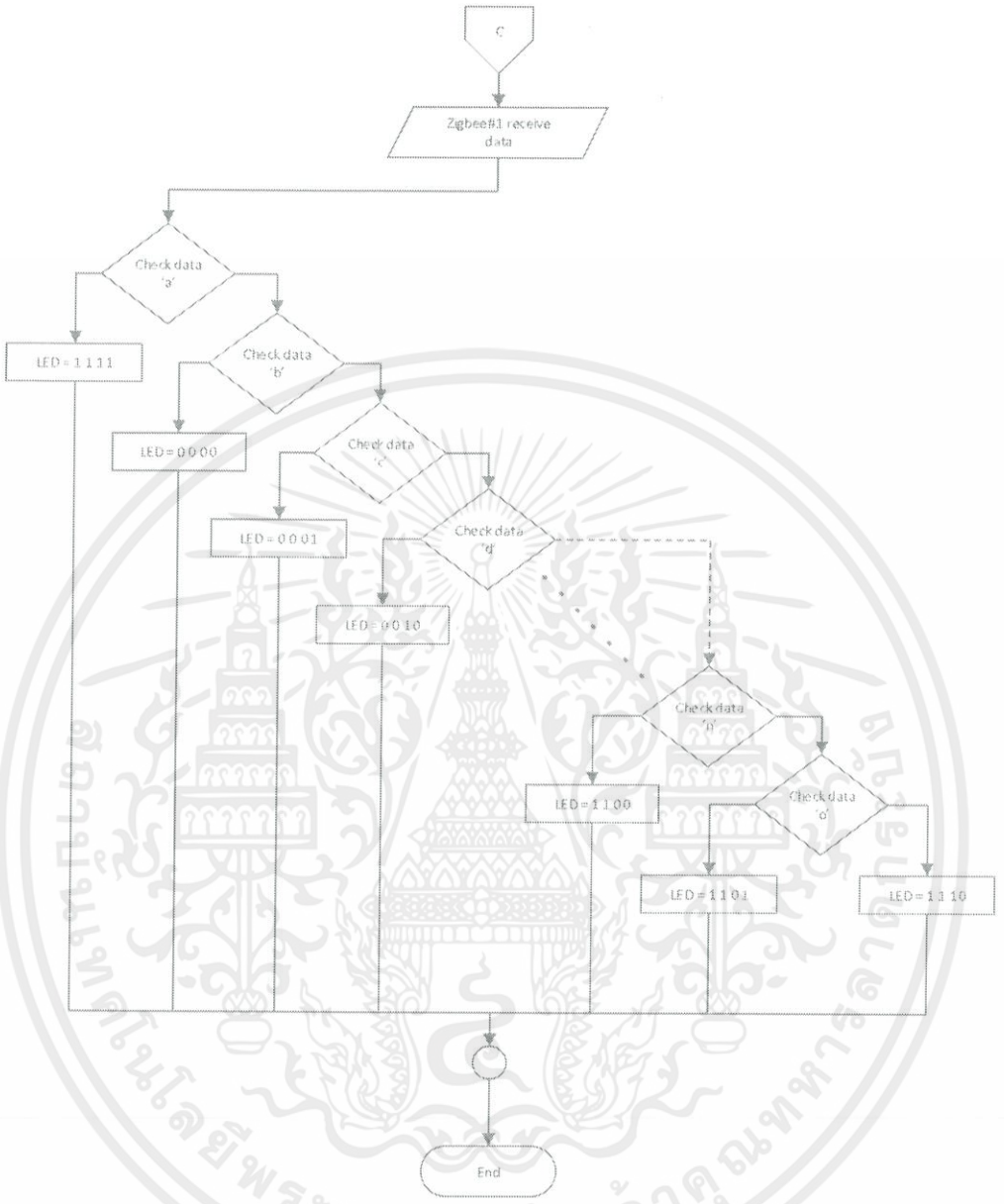
ขั้นตอนการทำงาน Flowchart (2nd Receiver)

1. Zigbee#3 จะทำการรับข้อมูลมากจาก Zigbee#1 และทำการเช็คข้อมูลตัวอักษร ‘A’, ‘B’, ‘C’, ‘D’, ‘E’, ‘F’, ‘G’, ‘H’, ’I’, ’J’, ’K’, ’L’, ’M’, ’N’, ’O’ และ ’P’

2. ทำการส่งตัวอักษรตรวจสอบการส่งข้อมูล ‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘f’, ‘g’, ‘h’, ‘I’, ‘j’, ‘k’, ‘l’, ‘m’, ‘n’, ‘o’, และ ‘p’ และควบคุมรีเลย์ สำหรับควบคุมการเปิด – ปิดหลอดไฟ

3. Zigbee#3 ส่งข้อมูลกลับไปยัง Zigbee#1 เพื่อทำการทดสอบว่ามีการส่งข้อมูลกลับมาจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.6 Flowchart แสดงการส่งข้อมูลกลับจาก 2nd Receiver

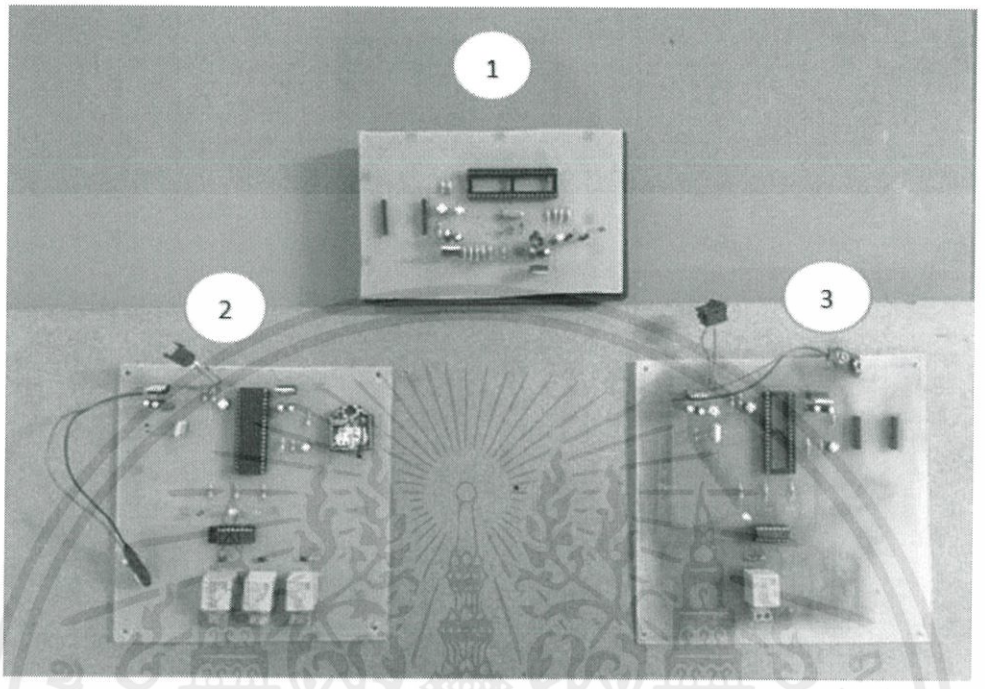
ขั้นตอนการทำงานของ Flowchart แสดงการส่งข้อมูลกลับจาก 2nd Receiver

1. Zigbee#1 จะทำการรับข้อมูลมากจาก Zigbee#3 (วงจรฝั่งรับที่ 2) และทำการเช็คข้อมูลตัวอักษร ‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘f’, ‘g’, ‘h’, ‘I’, ‘j’, ‘k’, ‘l’, ‘m’, ‘n’, ‘o’, และ ‘p’

2. ทำการส่งตัวอักษรตรวจสอบการส่งข้อมูลไปที่ไฟ LED เพื่อแสดงว่ามีการส่งข้อมูลกลับมาจริง

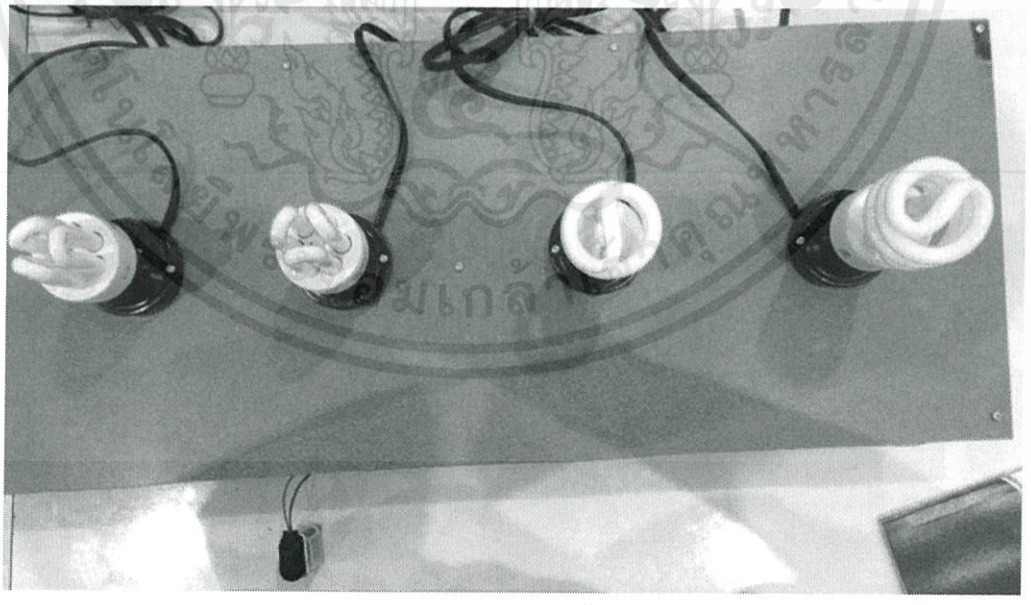
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.3 ชิ้นงาน



รูปที่ 3.7 วงจรทั้ง 3 วงจร

จากรูปที่ 3.7 แสดงวงจรทั้ง 3 วงจร ได้แก่ 1. วงจรฝั่งส่ง 2. วงจรฝั่งรับตัวแรกมีไฟ 3 ดวง 3. วงจรฝั่งรับตัวที่สองมีไฟ 1 ดวง

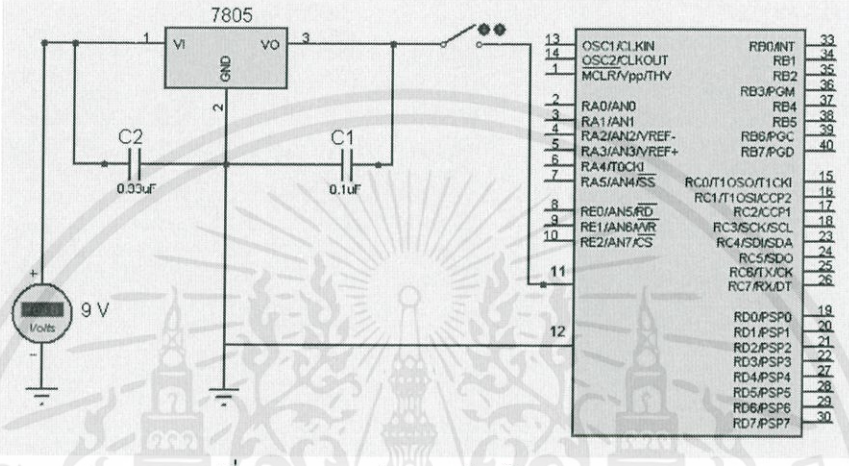


รูปที่ 3.8 หลอดไฟจริง 4 ดวง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.4 วงจรแปลงแรงดันจาก 9V เป็น 5V

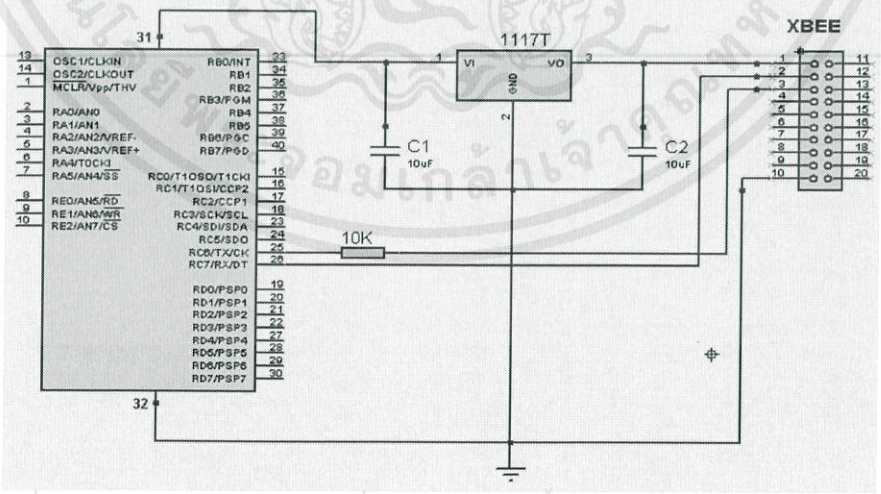
เนื่องจาก PIC16F877 ใช้แรงดันไฟในการทำงานที่ 5V จึงจำเป็นต้องมีการแปลงแรงดันจาก 9V เป็น 5V เสียก่อนจึงจะทำงานได้ เพราะมิเช่นนั้นอาจจะทำให้ตัวอุปกรณ์เกิดการเสียหายได้ โดยเราใช้ IC7805 ในการแปลงแรงดันตามวงจรดังรูป



รูปที่ 3.9 วงจรแปลงแรงดันเข้า PIC16F877

3.1.5 วงจรแปลงแรงดันเป็น 3.3V

ในการทำงานของ Zigbee จะทำงานได้ก็ต่อเมื่อป้อนไฟเลี้ยงขนาด 3.3V ให้กับตัว Zigbee โดยเราจะป้อนไฟเลี้ยงให้ผ่านไอซี LM1117T จากนั้นไอซีจะปรับแรงดันเป็น 3.3V เพื่อให้เหมาะสมกับการใช้งานกับ Zigbee

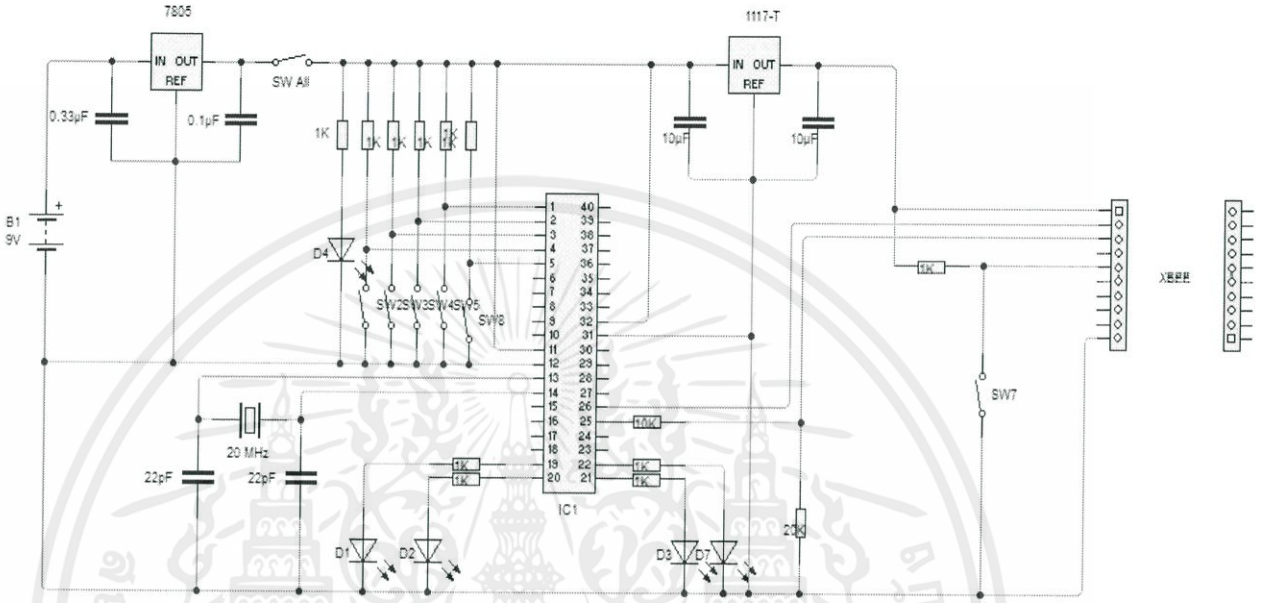


รูปที่ 3.10 วงจรแปลงแรงดันเข้า Zigbee

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับ [illegible] ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.6 วงจรด้านรับและด้านส่งของโมดูล Zigbee

1.) วงจรด้านส่ง

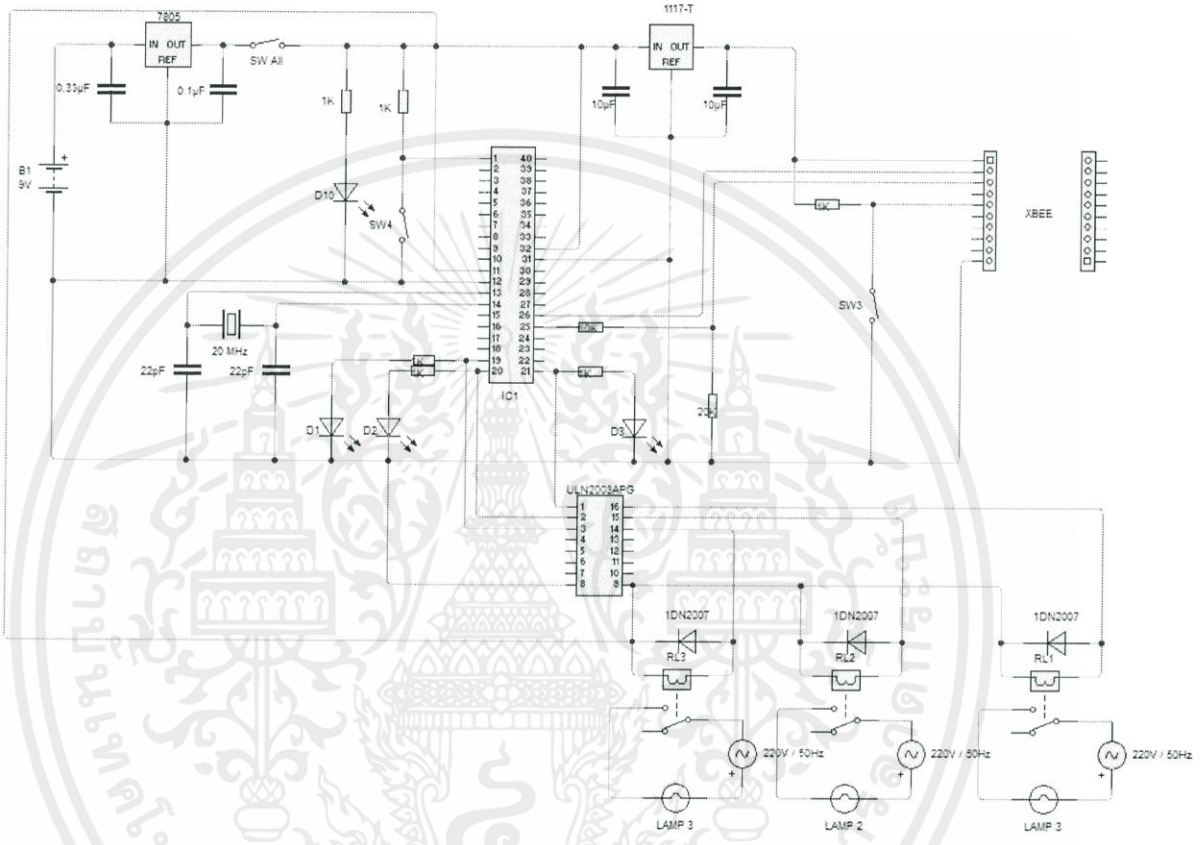


รูปที่ 3.11 วงจรฝั่งส่ง

จากรูปที่ 3.11 จะใช้ IC7805 ในการแปลงแรงดันไฟจาก 9V เป็น 5V เข้าขาที่ 11 ของ PIC16F877 ส่วนขาที่ 12 ต่อลง Ground จากนั้นขาที่ 32 ของ PIC16F877 จะทำหน้าเป็นข้อมูลขาออกไฟ 5V เพื่อไปแปลงแรงดันเข้า Zigbee โดยจะใช้ LM1117T แปลงแรงดันไฟที่มาจาก PIC16F877 5V เป็น 3.3V ต่อเข้าขาที่ 1 ของ Zigbee โดยจะต่อขาที่ 25 ของ PIC16F877 ซึ่งทำหน้าที่เป็นขา Tx เข้ากับขาที่ 3 ของ Zigbee ซึ่งทำหน้าที่เป็นขา Rx และต่อขาที่ 26 ของ PIC16F877 ซึ่งทำหน้าที่เป็นขา Rx เข้ากับขาที่ 2 ของ Zigbee ซึ่งทำหน้าที่เป็นขา Tx โดยวงจรฝั่งส่งจะใช้สวิตช์ทั้งหมด 4 ตัว และมีไฟ LED จำนวน 4 ดวง ไว้แสดงไฟสถานะว่ามีการส่งข้อมูลกลับมาจากวงจรฝั่งรับทั้งสองวงจรจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.) วงจรด้านรับที่ 1

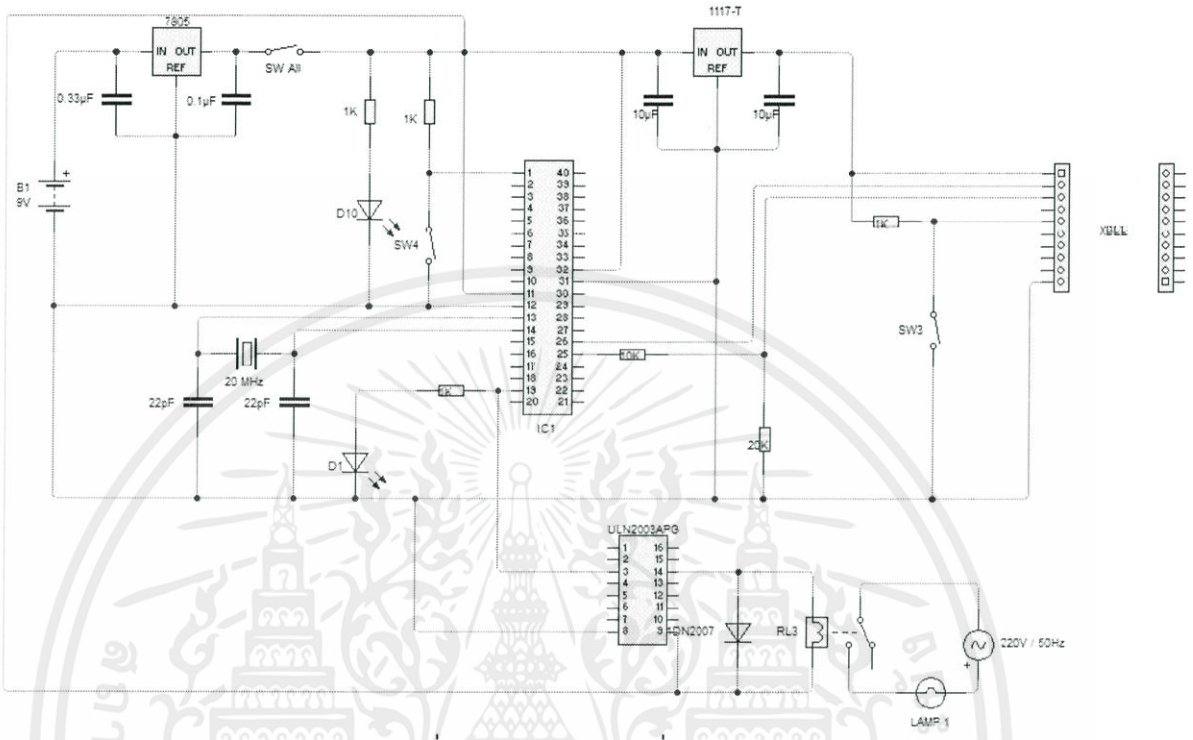


รูปที่ 3.12 วงจรด้านรับที่ 1

จากรูปที่ 3.12 จะใช้ IC7805 ในการแปลงแรงดันไฟจาก 9V เป็น 5V เข้าขาที่ 11 ของ PIC16F877 ส่วนขาที่ 12 ต่อลง Ground จากนั้นขาที่ 32 ของ PIC16F877 จะทำหน้าเป็นข้อมูลขาออกไฟ 5V เพื่อไปแปลงแรงดันเข้า Zigbee โดยจะใช้ LM1117T แปลงแรงดันไฟที่มาจาก PIC16F877 5V เป็น 3.3V ต่อเข้าขาที่ 1 ของ Zigbee โดยจะต่อขาที่ 25 ของ PIC16F877 ซึ่งทำหน้าที่เป็นขา Tx เข้ากับขาที่ 3 ของ Zigbee ซึ่งทำหน้าที่เป็นขา Rx และต่อขาที่ 26 ของ PIC16F877 ซึ่งทำหน้าที่เป็นขา Rx เข้ากับขาที่ 2 ของ Zigbee ซึ่งทำหน้าที่เป็นขา Tx โดยวงจรฝั่งรับที่ 1 จะมีรีเลย์จำนวน 3 ตัว และหลอดไฟจำนวน 3 ดวง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.) วงจรด้านรับที่ 2



รูปที่ 3.13 วงจรด้านรับที่ 2

จากรูปที่ 3.13 จะใช้ IC7805 ในการแปลงแรงดันไฟจาก 9V เป็น 5V เข้าขาที่ 11 ของ PIC16F877 ส่วนขาที่ 12 ต่อลง Ground จากนั้นขาที่ 32 ของ PIC16F877 จะทำหน้าเป็นข้อมูลขาออกไฟ 5V เพื่อไปแปลงแรงดันเข้า Zigbee โดยจะใช้ LM1117T แปลงแรงดันไฟที่มาจาก PIC16F877 5V เป็น 3.3V ต่อเข้าขาที่ 1 ของ Zigbee โดยจะต่อขาที่ 25 ของ PIC16F877 ซึ่งทำหน้าที่เป็นขา Tx เข้ากับขาที่ 3 ของ Zigbee ซึ่งทำหน้าที่เป็นขา Rx และต่อขาที่ 26 ของ PIC16F877 ซึ่งทำหน้าที่เป็นขา Rx เข้ากับขาที่ 2 ของ Zigbee ซึ่งทำหน้าที่เป็นขา Tx โดยวงจรฝั่งรับที่ 2 จะมีรีเลย์จำนวน 1 ตัว และหลอดไฟจำนวน 1 ดวง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 เครื่องมือที่ใช้ในการทดลอง

3.2.1 Zigbee Serie2

1. วิธีการเลือกใช้ Zigbee

- Zigbee ที่หาได้ตามท้องตลาดจะมีอยู่ 2 Series ได้แก่ Series1 และ Series 2 โดย Series 2 จะใช้งานได้ง่ายกว่า Series 1 ข้อที่สำคัญคือ Zigbee Series1 กับ Series 2 นี้จะไม่สามารถใช้งานร่วมกันได้ สำหรับ Series 2 จะมีข้อดีกว่า Series1 คือ สามารถที่จะสื่อสารกันในแบบ Mesh และสามารถที่จะทำ Configuration ผ่านระบบไร้สายโดยที่ไม่จำเป็นต้องไปแกะตัวอุปกรณ์ Zigbee มาเชื่อมต่อบอร์ดสำหรับตั้งค่า Zigbee
- ความไกลในการส่งสัญญาณสำหรับการใช้งาน โดย Zigbee แบบธรรมดา จะมีกำลัง ส่ง อยู่ที่ 1-2 mW จะรับ-ส่งสื่อสารระยะทางอยู่ที่ 100 ถึง 120 เมตร ส่วน Zigbee PRO จะมีกำลังส่ง 50 - 60 mW จะส่งได้ไกลที่ 1500 เมตร ซึ่งระยะทางนี้คำนวณแบบ Line Of Sight หรือเป็นแบบไม่มีสิ่งใดมาขวางกั้นระยะทางมีใช้ในการสื่อสารระหว่างอุปกรณ์ 2 ตัว แต่เมื่อนำไปใช้งานจริงก็จะเจอสิ่งกีดขวางมากมาย เช่น ผนัง หลังคา ต้นไม้ เป็นต้น ซึ่งอาจจะทำให้ไม่ได้ระยะทางตามที่ต้องการแต่ไม่เราสามารถใช้ Zigbee ส่งต่อข้อมูลกันได้ ส่งไปเรื่อยๆ จนถึงปลายทางได้
- ชนิดของเสาอากาศที่เหมาสำหรับใช้งาน ควรเลือกตามความสะดวกในการจัดทำชิ้นงาน เพื่อความสวยงามของชิ้นงาน และ ไม่ทำให้เกะกะเมื่อทำการทดลอง

2. การใช้งานแต่ละขาของโมดูล Zigbee

ตารางที่ 3.1 การทำงานแต่ละขาของโมดูล Zigbee [8]

Pin #	Name	Direction	Description
1	VCC	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN / $\overline{\text{CONFIG}}$	Input	UART Data In
4	DIO8*	Output	Digital Output 8
5	RESET	Input	Module Reset (reset pulse must be at least 200 ns)
6	PWM0 / RSSI	Output	PWM Output 0 / RX Signal Strength Indicator
7	PWM1	Output	PWM Output 1
8	[reserved]	-	Do not connect
9	$\overline{\text{DTR}}$ / SLEEP_RQ / DI8	Input	Pin Sleep Control Line or Digital Input 8
10	GND	-	Ground
11	AD4 / DIO4	Either	Analog Input 4 or Digital I/O 4
12	$\overline{\text{CTS}}$ / DIO7	Either	Clear-to-Send Flow Control or Digital I/O 7
13	ON / SLEEP	Output	Module Status Indicator
14	VREF	Input	Voltage Reference for A/D Inputs
15	Associate / AD5 / DIO5	Either	Associated Indicator, Analog Input 5 or Digital I/O 5
16	RTS / AD6 / DIO6	Either	Request-to-Send Flow Control, Analog Input 6 or Digital I/O 6
17	AD3 / DIO3	Either	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0	Either	Analog Input 0 or Digital I/O 0

เอกสารนี้เป็นเอกสาร... ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. คุณสมบัติของ Zigbee Series2

ตารางที่ 3.2 คุณสมบัติของ Zigbee Series2

Specification	Zigbee Series2
1. Indoor/Urban Range	up to 133 ft. (40 m)
2. Outdoor RF line-of-sight Range	up to 400 ft. (120 m)
3. Transmit Power Output	2mW (+3dBm), boost mode enabled 1.25mW (+1dBm), boost mode disabled
4. RF Data Rate	250,000 bps
5. Serial Interface Data Rate (software selectable)	1200 - 230400 bps (non-standard baud rates also supported)
6. Receiver Sensitivity	-96 dBm, boost mode enabled -95 dBm, boost mode disabled

3.2.2 PIC16F877

1. การจัดขาของไอซี

ในรูปนั้นถ้าขาใดเป็น Active Low จะมีเครื่องหมาย ’ เอาไว้ด้านหลัง เช่น MCLR’ ก็คือ ขาที่ Active แบบ Low

รูปที่ 3.14 การจัดขาของ PIC16F877 [7]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. คุณสมบัติของPIC16F877

1. มีคำสั่งให้ใช้งาน 35 คำสั่ง
2. คำสั่งหนึ่งๆใช้เวลาทำงาน 1 ถึง 2 Cycle
3. ทำงานได้สูงสุดที่ 20MHz
4. ทำงานแบบ Pipe-line (มี 2 ท่อ) ทำให้ ณ เวลาหนึ่งทำงาน 2 อย่างพร้อมๆกันได้
5. หน่วยความจำโปรแกรมเป็นแบบ Flash มีขนาด 8KWord (1 word=14 บิต)
6. มี RAM ขนาด 368 ไบต์ ให้ใช้งาน
7. มี EEPROM ขนาด 256 ไบต์
8. ตอบสนองกับอินเตอร์รัพได้ทั้งหมด 14 แหล่ง
9. มี Stack ให้ใช้ได้สูงสุด 8 ระดับ
10. มีระบบ Power On Reset, Power Up Timer, Oscillator Start-up timer
11. Watchdog timer
12. มีระบบ Code Protection ป้องกันการลอกเลียนแบบ
13. มีโหมดประหยัดพลังงานช่วยในการประหยัดค่าใช้จ่าย
14. สัญญาณนาฬิกามีหลายโหมดให้เลือกใช้งาน คือ อาจจะใช้ XTAL หรือ วงจร RC ก็ได้
15. สามารถโปรแกรมด้วยไฟ +5VDC ได้
16. ใช้การโปรแกรมแบบ In-Circuit Serial Programming
17. ทำงานที่ไฟเลี้ยง 2VDC ถึง 5.5VDC
18. Current Sink และ Current Source อยู่ที่ 25mA
19. มี Timer/Counter 3 ตัว
20. มีโมดูล Capture/Compare/PWM อีก 2 ชุด
21. มี A-TO-D Converter แบบ 10 บิต จำนวน 8 ช่องนำเข้า ในตัวเอง
22. มีระบบ USART สำหรับต่อกับ การสื่อสารแบบ RS232 หรือดีกว่า
23. มีระบบตรวจระดับไปเลี้ยง (Brown-out reset)
24. มี I/O พอร์ตทั้งหมด 5 พอร์ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

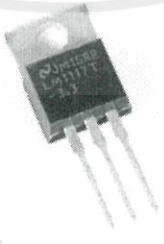
3.2.3 IC7805

7805 เป็น IC เรียงกระแสแบบคงที่ +5V โดย IC ตัวนี้มีคุณสมบัติหลายอย่าง เช่น ป้องกันความปลอดภัยในส่วนของพื้นที่นั้น ๆ ,ทำการตัดวงจรเมื่อมีความร้อนสูง ,กระแสภายในจะถูกจำกัดเมื่อเกิดปัญหาใน IC เนื่องจาก PIC16F877 จะทำงานที่แรงดันไฟฟ้า 5V แต่จ่ายแรงดันไฟฟ้า 9V ที่จุดกำเนิด ทำให้ต้องแปลงแรงดันไฟฟ้าเสียก่อน ดังนั้น IC7805 จะทำหน้าที่แปลงแรงดันไฟฟ้าจาก 9V เป็น 5V เพื่อนำไปใช้งานในอุปกรณ์ต่อไป

รูปที่ 3.15 IC7805

3.2.4 LM1117T

ในวงจรทั้งทางด้านรับและด้านส่งนั้นจะประกอบไปด้วย Zigbee ทั้งสองด้าน ประกอบด้วยด้านละ 1 ตัว โดยการที่ Zigbee จะทำงานได้นั้น จะต้องจ่ายแรงดันไฟฟ้าขนาด 3.3V ให้แก่ตัวอุปกรณ์ เพราะถ้าแรงดันไฟฟ้าที่จ่ายให้กับ Zigbee มีค่ามากกว่า 3.3V จะทำให้ตัว Zigbee เกิดความเสียหายได้ ดังนั้นจึงต้องผ่านวงจรแปลงแรงดันไฟฟ้าให้มีขนาด 3.3V โดยในวงจรแปลงแรงดันไฟฟ้านี้จะประกอบด้วย IC Regulator ที่ใช้ในวงจรคือ LM1117T โดยไฟที่ส่งมาจาก PIC16F877 จะมีขนาด 5V เมื่อผ่านวงจรแปลงแรงดันไฟฟ้าจะมีขนาด 3.3V ต่อเข้าที่ขาที่ 1 ของ Zigbee ทั้งสองตัว



รูปที่ 3.16 LM1117T

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2.5 Relay HRS4H-S-DC5V

รีเลย์ (Relay) เป็นอุปกรณ์ที่เปลี่ยนพลังงานไฟฟ้าให้เป็นพลังงานแม่เหล็ก เพื่อใช้ในการดึงดูดหน้าสัมผัสของคอนแทคให้เปลี่ยนสภาวะ โดยการป้อนกระแสไฟฟ้าให้กับขดลวด เพื่อทำการปิดหรือเปิดหน้าสัมผัสคล้ายกับสวิตช์อิเล็กทรอนิกส์ ซึ่งเราสามารถนำรีเลย์ไปประยุกต์ใช้ในการควบคุมวงจรต่าง ๆ ในงานช่างอิเล็กทรอนิกส์มากมาย

การเลือกซื้อรีเลย์มีหลักที่ควรพิจารณาในการเลือกซื้อ โดยให้ระบุความต้องการเป็นข้อ ๆ ว่ารีเลย์ที่กำลังจะซื้อสามารถที่จะสนองความต้องการทั้ง 9 ข้อดังนี้

1.กระแสไฟฟ้าที่ใช้ป้อนให้กับขดลวดเป็นไฟตรงหรือไฟสลับ

2.แรงเคลื่อนและความถี่ของไฟฟ้าที่จะใช้กับขดลวดของรีเลย์

3.ความต้านทานของขดลวดมีค่าเท่าใด

4.อุณหภูมิสูงสุดเท่าใด

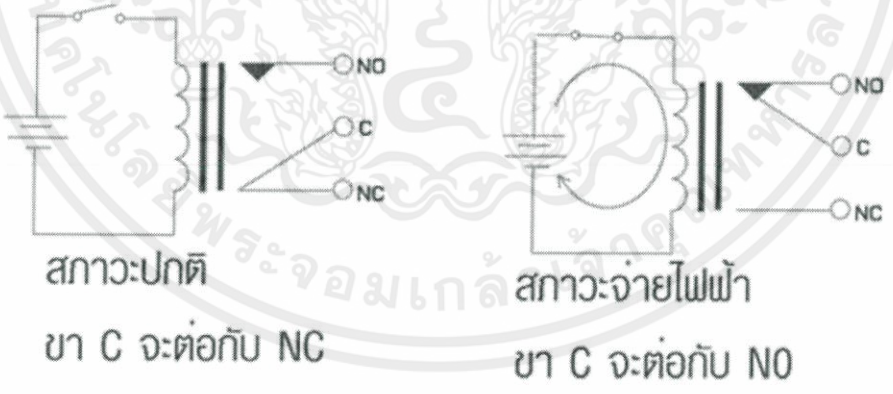
5.หน้าสัมผัสต้องใช้กับแรงเคลื่อนและกระแสเท่าใด

6.หน้าสัมผัสเป็นแบบใด

7.แหล่งจ่ายไฟฟ้าที่ใช้มีช่วงแรงดันเปลี่ยนแปลงไปมากน้อยเพียงใด

8.ต้องการเวลาสัมผัสและจากของหน้าสัมผัสเร็วหรือช้าเพียงใด

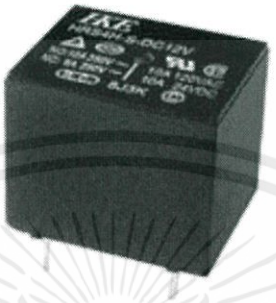
9.ลักษณะรูปร่างภายนอกเป็นอย่างไร และจะติดตั้งเข้ากับวงจรอย่างไร



รูปที่ 3.17 การทำงานของRelay [9]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

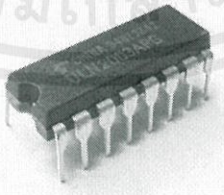
จากรูปเมื่ออยู่ในสภาวะปกติ ขา C ต่ออยู่กับขา NC ไฟในวงจรจะไม่ติดเพราะไม่มีกระแสไฟฟ้าไหลผ่านอุปกรณ์ แต่ในรูปทางด้านขวา เมื่อทำการจ่ายไฟให้กับตัวรีเลย์ ขา C จะต่ออยู่กับขา NO บ่งชี้ว่ามีกระแสไฟฟ้าไหลผ่านตัวอุปกรณ์ ดังนั้นหลอดไฟจะติด



รูปที่ 3.18 Relay [9]

3.2.6 ULN2003APG

ไอซี ULN2003APG เป็นไอซีทำหน้าที่ขยายแรงดันและกระแสแล้วส่งสัญญาณออกทางเอาต์พุตเพื่อขับโหลด เนื่องจากกระแสไฟที่ส่งออกมาจากวงจรส่งข้อมูลนั้นไม่เพียงพอที่จะทำให้หลอดไฟติดได้ จึงต้องใช้ IC ULN2003APG ทำหน้าที่ทำให้รีเลย์ทำงาน ซึ่ง IC ULN2003APG มีคุณสมบัติ คือ สามารถขับโหลดได้ถึง 500 mA และทนแรงดันออกได้สูงถึง 50 โวลท์ นอกจากนี้ยังมีไดโอดต่อป้องกันไว้ภายในสำหรับขับรีเลย์ได้เลย และใช้สำหรับอินพุทแบบ TTL หรือ CMOS ขนาด 9V เมื่อกระแสและแรงดันถูกขยายแล้ว รีเลย์จะทำงานและส่งผลให้หลอดไฟทำงาน

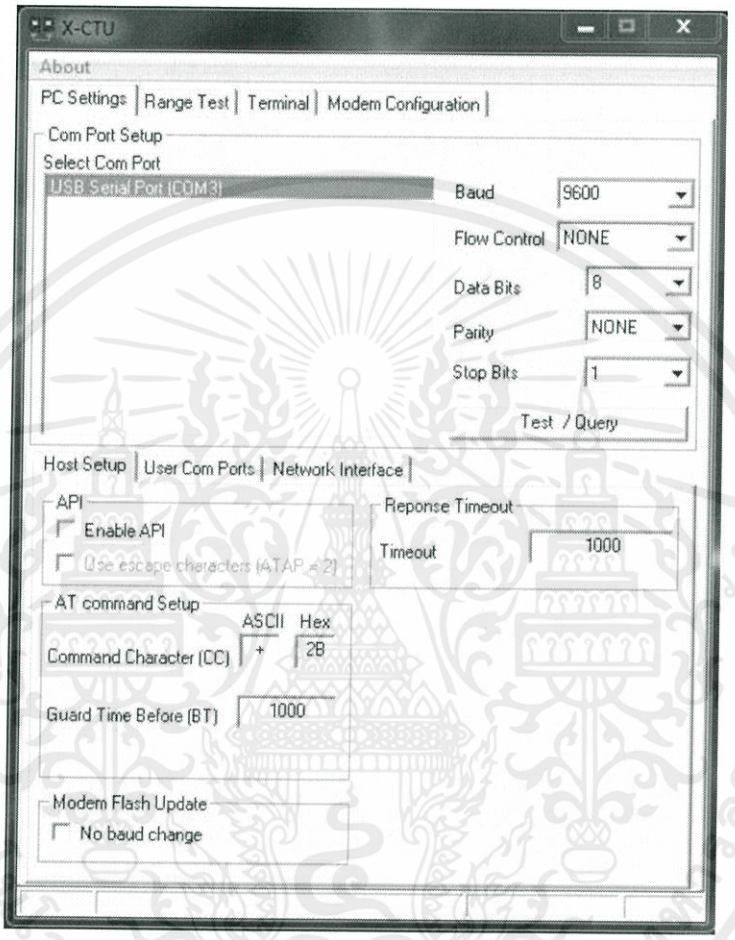


รูปที่ 3.19 ULN2003APG [5]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3 การตั้งค่า Zigbee ผ่านโปรแกรม X-CTU

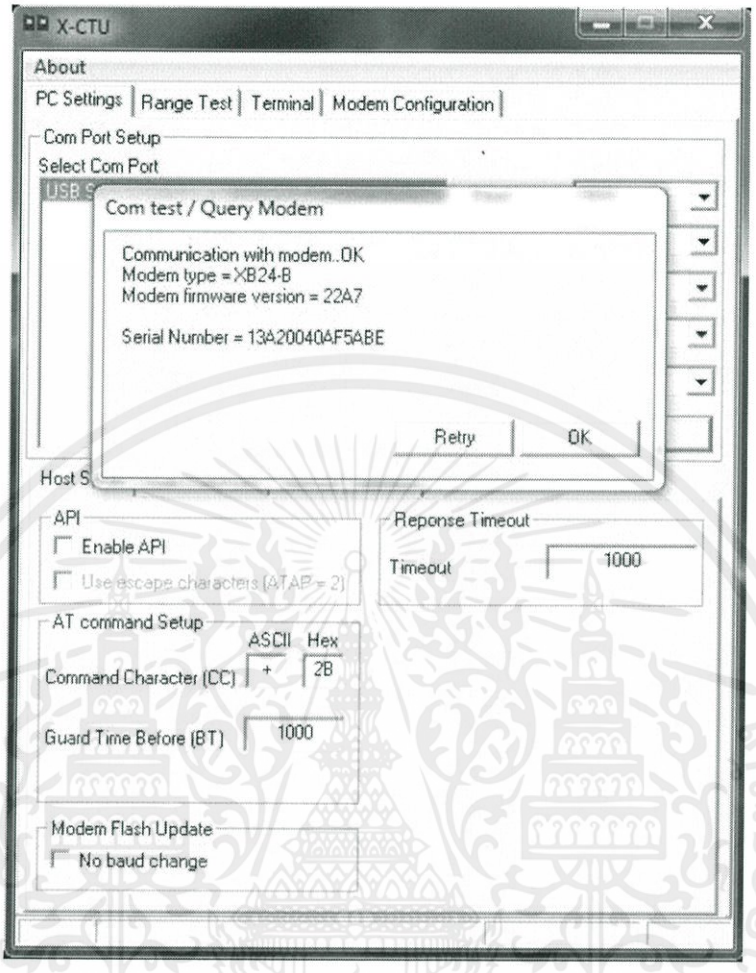
3.3.1 การเพิ่มความปลอดภัยให้กับระบบเครือข่าย



รูปที่ 3.20 หน้าเริ่มต้นของ X-CTU

1. เชื่อมต่อ module Zigbee เข้ากับ PC
2. เรียกใช้ โปรแกรม X-CTU ขึ้นมา ตั้งค่า Com Port ที่เชื่อมต่ออยู่กับ Module ตั้งค่า baud, Flow Control, Data Bit, Parity, Stop Bit ตามรูป
3. ทำการ Test ว่า Zigbee ได้เชื่อมต่อสมบูรณ์แล้วตามรูป

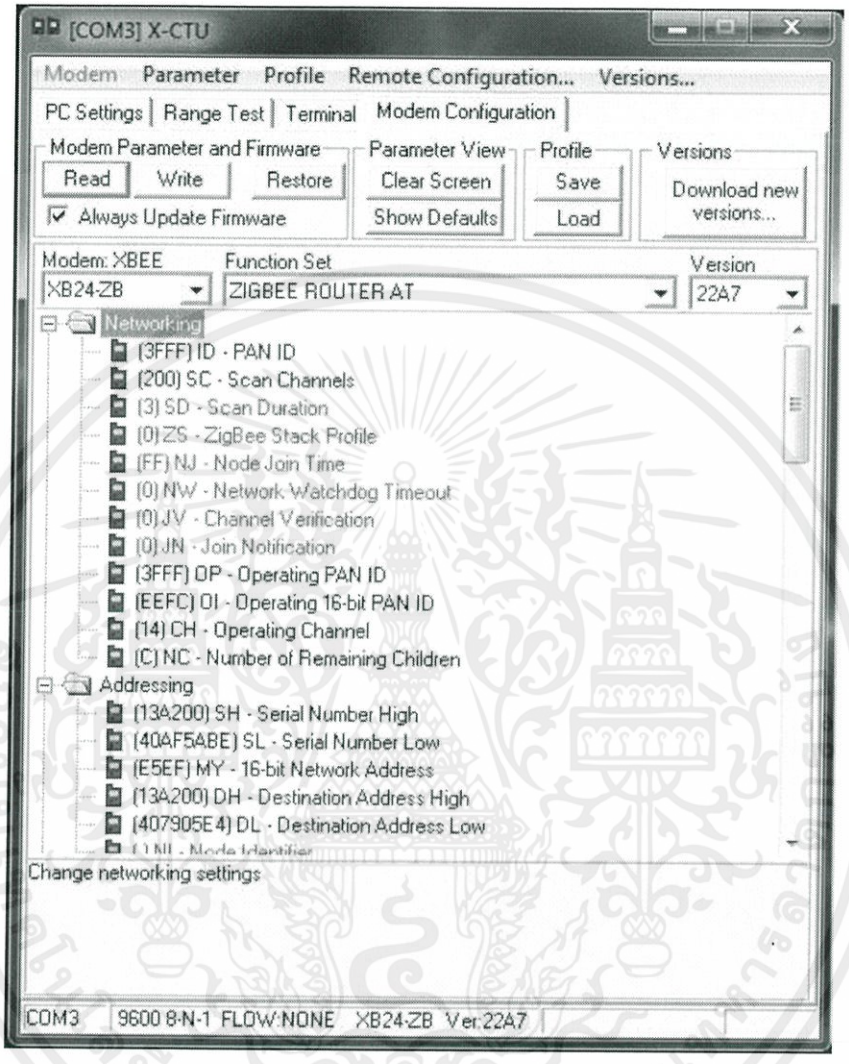
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.21 หลังจากกด test

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.2 การตั้งค่าให้ฟังก์ชัน Zigbee



รูปที่3.22 หน้า Modem Configuration ของ X-CTU

1. เข้าไปที่ Modem Configuration
2. เลือก Modem XBee ให้ตรงตามรุ่นที่ใช้
3. ตั้งคั่งก์ชันตามที่ต้องการใช้งาน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4 การออกแบบการจัดเก็บผลของปริญญานิพนธ์

3.4.1 การทดสอบการส่งข้อมูลผ่าน Zigbee ทั้ง 2 ตัว

1.ต่อ Zigbee#1 และZigbee#2 กับ Mini Xbee USB Dongle แล้วต่อเข้าคอมพิวเตอร์

2.เปิดโปรแกรม X-CTU มา 2 หน้าต่างเพื่อเช็คว่า Zigbee ทั้ง 2 ตัวที่เราตั้งค่าไว้สามารถสื่อสารได้จริง

3.พิมพ์ข้อมูลลงไปหน้าต่างใดหน้าต่างหนึ่ง

4.ต่อ Zigbee#1 และ Zigbee#3 กับ Mini Xbee USB Dongle แล้วทำตามขั้นตอนที่ 2 และ 3

5.เก็บผลการทดลอง

3.4.2 การทดสอบการส่งข้อมูลของฝั่งส่งโดยใช้โปรแกรม X-CTU

1.ต่อ Zigbee#1 กับวงจรฝั่งส่งและ Zigbee#2 กับ Mini Xbee USB Dongle ต่อกับ คอมพิวเตอร์

2.เปิดโปรแกรม X-CTU

3.ทำการกดปุ่มเพื่อตรวจสอบฝั่งส่งว่าส่งข้อมูลตามที่กำหนดไว้หรือไม่

4.เก็บผลการทดลอง

3.4.3 การทดสอบผ่าน Spectrum Analyzer

1.ทำการต่อวงจรฝั่งส่งและรับ

2.ทำการเซ็ตค่า Center Frequency ที่เครื่อง Spectrum Analyzer ไว้ที่ เท่ากับความถี่ของอุปกรณ์ เพื่อดูความถี่ของ Zigbee ว่าตรงตามทฤษฎีหรือไม่ โดยจะคำนวณจาก Center Frequency = 2.405 + (CH - 10) x 5 MHz โดย CH ในที่นี้เป็นฐาน 10

3.เก็บผลการทดลอง

3.4.4 การทดสอบการรับค่าผ่าน Mixed Signal Oscilloscope

1.ต่อวงจรฝั่งส่งและฝั่งรับ

2.ทำการวัดค่าที่ขา 25 ของ PIC16F877 ที่วงจรฝั่งส่ง ซึ่งทำหน้าที่เป็นขา Tx ของวงจร เพื่อดูรูปสัญญาณเป็นไปตามที่กำหนดไว้หรือไม่ ซึ่งจะมีทั้งหมด 16 ค่า คือ ‘A’, ‘B’, ‘C’, ‘D’, ‘E’, ‘F’, ‘G’, ‘H’, ’I’, ’J’, ’K’, ’L’, ’M’, ’N’, ’O’ และ ’P’ เป็นการส่งคำสั่งเพื่อควบคุมหลอดไฟทั้ง 4 ดวง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.ทำการวัดค่าที่ขา 26 ของ PIC16F877 ที่วงจรฝั่งส่งที่ส่งมาจากวงจรฝั่งรับเพื่อเช็คว่ามีการส่งข้อมูลกลับมาจริง ซึ่งทำหน้าที่เป็นขา Rx ของวงจร เพื่อดูรูปสัญญาณที่รับมาจาก Zigbee#2 และ Zigbee#3 ว่ามีค่าเท่าไร

4.เก็บผลการทดลอง

3.4.5 การทดสอบการเปิด – ปิดหลอดไฟด้วยอุปกรณ์จริง

1.ป้อนแรงดันไฟฟ้าขนาด 9V เข้าที่วงจรฝั่งส่ง วงจรฝั่งรับที่ 1 และวงจรฝั่งรับที่ 2

2.กดสวิตช์ทดสอบการเปิด – ปิดหลอดไฟ ว่าสอดคล้องกับที่เรากำหนดหรือไม่

3.ตรวจสอบไฟสถานะว่ามีการส่งข้อมูลกลับมาและสอดคล้องกับหลอดไฟจริงหรือไม่

4.เก็บผลการทดลอง

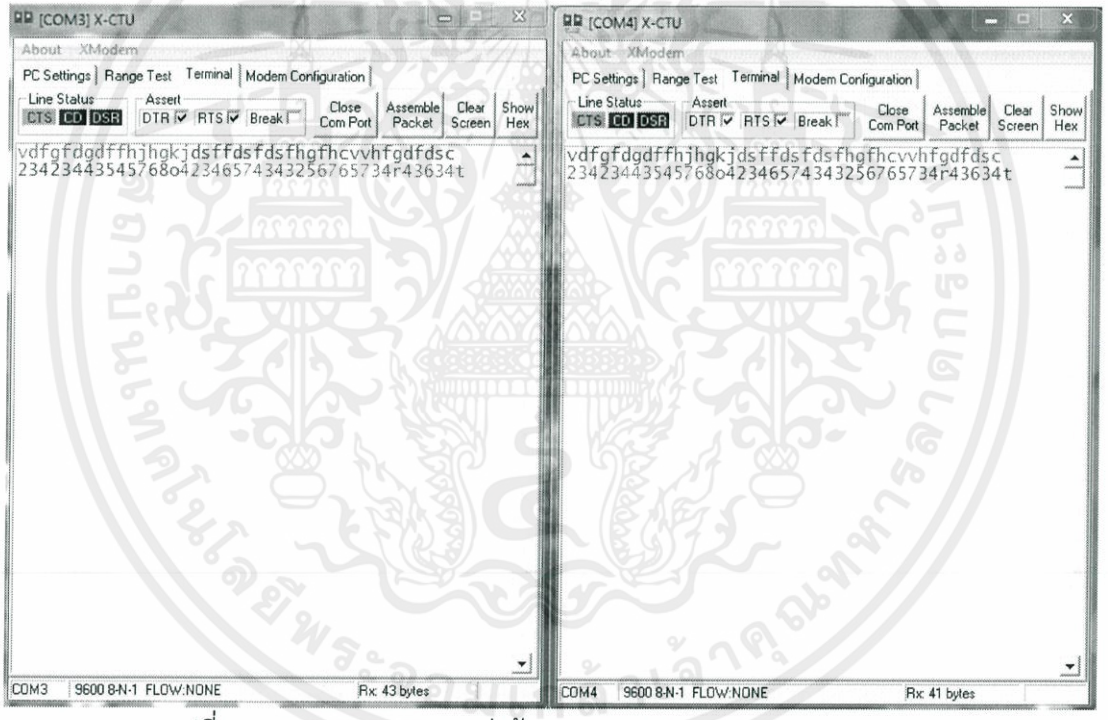
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

4.1 ผลการทดสอบการส่งข้อมูลผ่าน Zigbee ทั้ง 2 ตัว

การที่ Zigbee 2 ตัว จะติดต่อส่งข้อมูลกันได้นั้น จำเป็นจะต้องตั้งค่าฟังก์ชันของ Zigbee แต่ละตัวว่าให้ทำหน้าที่อะไรบ้าง โดยในปริญญานิพนธ์นี้จะกำหนด Zigbee ตัวแรกทำหน้าที่ Coordinator คือ มีหน้าที่สร้างการสื่อสารเชื่อมโยงเครือข่ายระหว่าง Coordinator กับ Router และ Zigbee อีกตัวทำหน้าที่เป็น Router คือ มีหน้าที่รับส่งข้อมูลในเส้นทางต่าง ๆ ของเครือข่าย จากรูปที่ 4.1 เป็นผลการทดสอบการส่งข้อมูลของ Zigbee#1 และ Zigbee#2



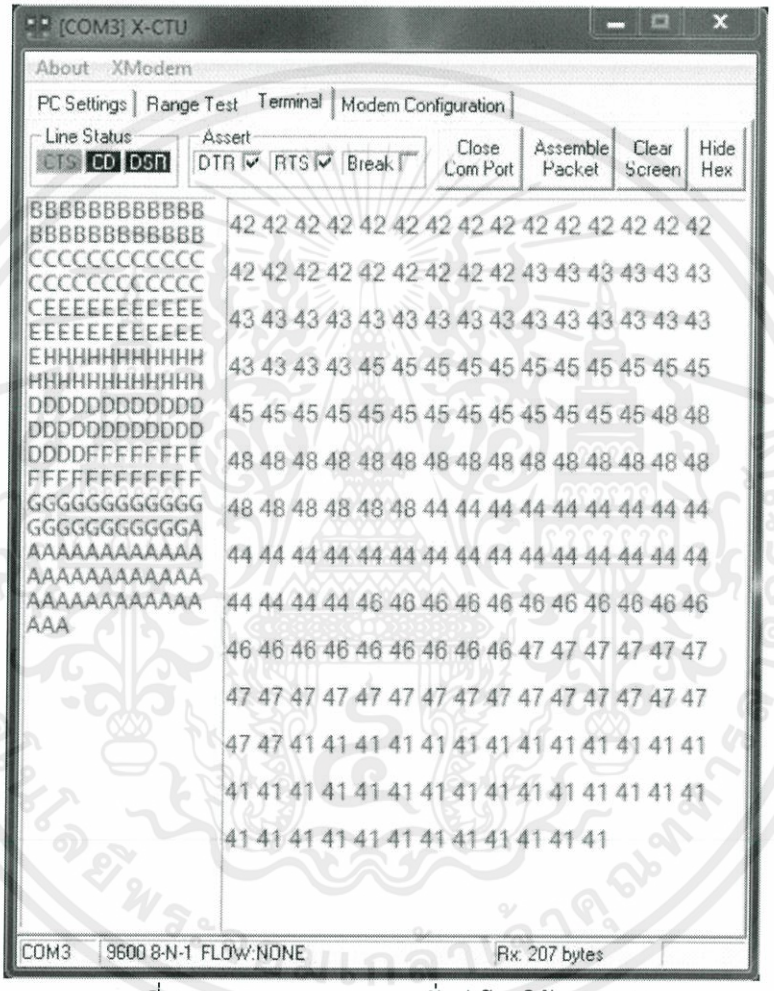
รูปที่ 4.1 ผลการทดสอบการส่งข้อมูลของ Zigbee#1 และ Zigbee#2

จากรูปที่ 4.1 พบว่าเมื่อ Zigbee ทั้ง 2 ตัว สามารถส่งข้อมูลหรือติดต่อกันโดยสมบูรณ์ สังเกตจอแสดงผลของโปรแกรมX-CTU

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 ผลการทดสอบการส่งข้อมูลของฝั่งส่งโดยใช้โปรแกรม X-CTU

จากการทดสอบเป็นการตรวจสอบว่าข้อมูลที่ถูกส่งออกจากฝั่งส่งเป็นไปตามที่เรากำหนดไว้จริง โดยการที่ให้ Zigbee#1 ต่อกับวงจรฝั่งส่งเพื่อเป็นอุปกรณ์ที่ใช้ส่งข้อมูลแล้วใช้ Zigbee#2 ต่อกับ Mini Xbee USB Dongle เชื่อมต่อกับคอมพิวเตอร์ จากนั้นเปิดโปรแกรม X-CTU เพื่อทำการทดสอบ



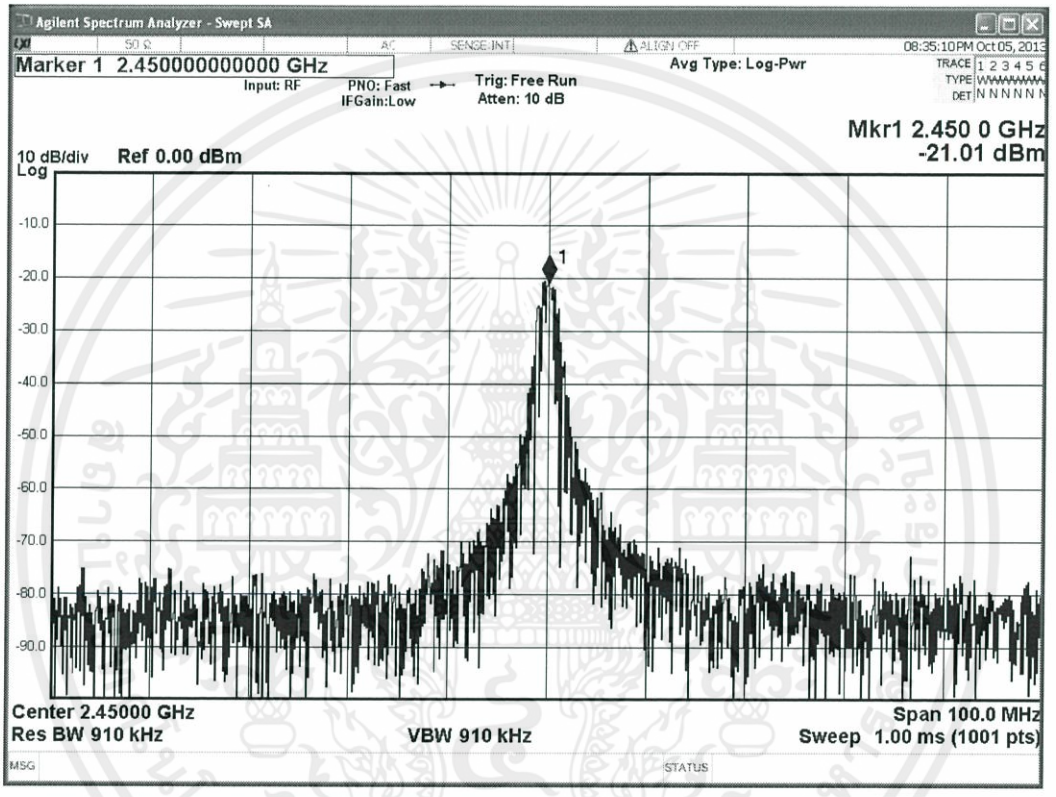
รูปที่ 4.2 ผลการทดสอบของฝั่งส่งโดยใช้ X-CTU

จากรูป 4.2 จะเห็นได้ว่าข้อมูลที่ได้จากการใช้โปรแกรม X-CTU เป็นไปตามที่เรากำหนดไว้ คือ ‘A’, ‘B’, ‘C’, ‘D’, ‘E’, ‘F’, ‘G’, ‘H’, ’I’, ’J’, ’K’, ’L’, ’M’, ’N’, ’O’ และ ’P’ โดยหน้าต่างฝั่งซ้ายคือข้อมูลตัวอักษรที่ป้อนเข้าไปตามที่กำหนด ส่วนหน้าต่างฝั่งขวาคือค่าของเลขฐาน 16 ที่สอดคล้องกับค่าในหน้าต่างฝั่งซ้าย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.3 ผลการทดสอบผ่าน Spectrum Analyzer

เป็นการทดสอบดูความเข้มของสัญญาณและความถี่ของ Zigbee ว่าตรงกับคุณสมบัติทางเทคนิคหรือไม่ โดย Zigbee ที่ใช้มีค่าความถี่อยู่ที่ 2.45 GHz อยู่ที่ channel 14 โดยกำหนดให้สถานที่ที่ทำการทดลองไม่มีคลื่นใดๆ รบกวน โดยผลการทดสอบเป็นไปตามรูปที่ 4.3

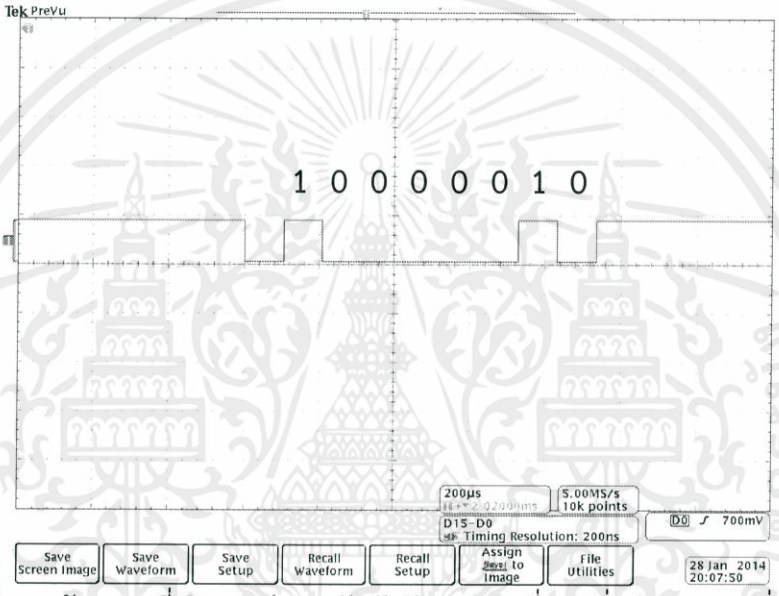


รูปที่ 4.3 ผลการวัดสัญญาณผ่าน Spectrum Analyzer

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 ผลการทดสอบการวัดค่าผ่าน Mixed Signal Oscilloscope

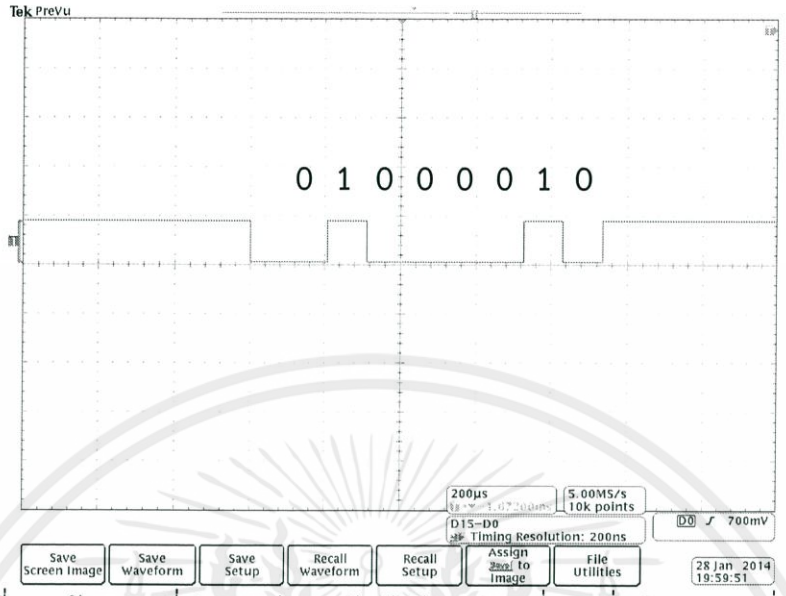
ในการทดสอบนี้จะทดสอบคล้าย 4.2 แต่จะใช้อุปกรณ์คือ Oscilloscope วัดค่าออกมาเป็น pulse สัญญาณซึ่งจะวัดที่ขา 25 คือขา Tx ของ PIC16F877 ที่ฝั่งส่งและจะวัดที่ขา 26 คือขา Rx ของ PIC16F877 ของฝั่งรับที่ส่งมาจาก Zgibee#2 และ Zigbee#3 ว่ามีค่าตามที่เรากำหนดไว้หรือไม่ ซึ่งค่าเหล่านั้นคือ ‘A’, ‘B’, ‘C’, ‘D’, ‘E’, ‘F’, ‘G’, ‘H’, ‘I’, ‘J’, ‘K’, ‘L’, ‘M’, ‘N’, ‘O’ และ ‘P’



รูปที่ 4.4 สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘A’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ

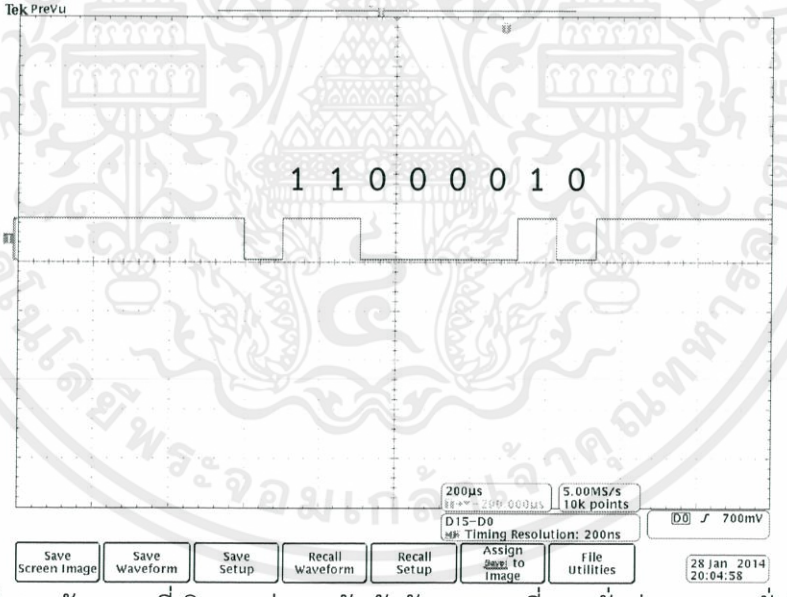
จากรูปที่ 4.4 สัญญาณที่ได้คือ 0 1 0 0 0 0 0 1 เมื่อแปลงเป็น ฐาน 16 จะได้ 41 ซึ่งคือค่า ‘A’ ที่เรากำหนดไว้ ซึ่งค่าที่ได้นี้ตรงกับการทดสอบจาก X-CTU ทำให้ไฟทั้ง 4 ดวงติดทั้งหมด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.5 สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘B’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ

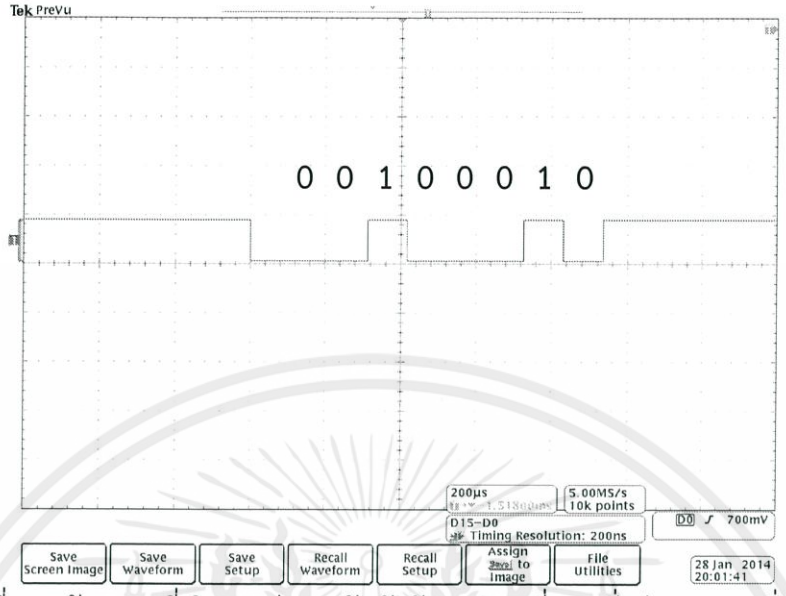
จากรูปที่ 4.5 สัญญาณที่ได้คือ 0 1 0 0 0 0 1 0 เมื่อแปลงเป็น ฐาน 16 จะได้ 42 ซึ่งคือค่า ‘B’ ที่เรากำหนดไว้ ซึ่งค่าที่ได้นี้ตรงกับการทดสอบจาก X-CTU ทำให้ไฟทั้ง 4 ดวงดับทั้งหมด



รูปที่ 4.6 สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘C’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ

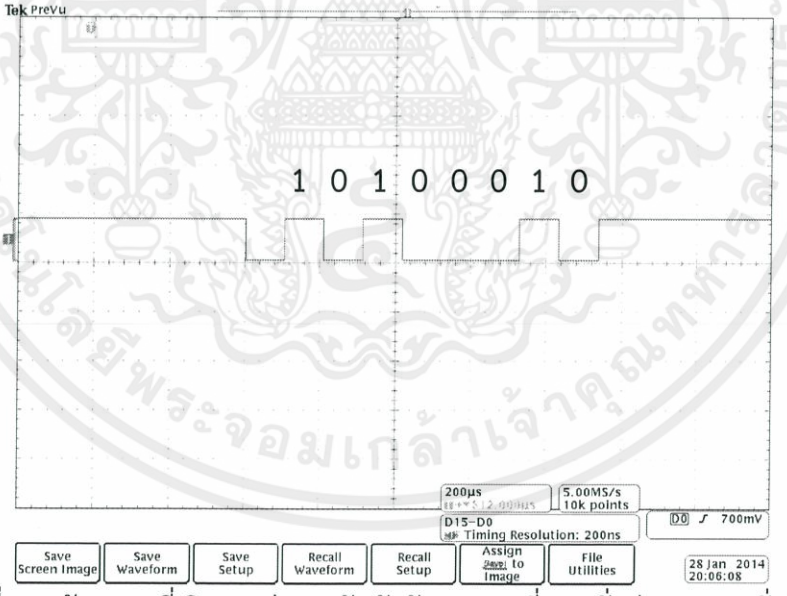
จากรูปที่ 4.6 สัญญาณที่ได้คือ 0 1 0 0 0 0 1 1 เมื่อแปลงเป็น ฐาน 16 จะได้ 43 ซึ่งคือค่า ‘C’ ที่เรากำหนดไว้ ซึ่งค่าที่ได้นี้ตรงกับการทดสอบจาก X-CTU ทำให้ไฟดวงที่ 4 ติด แต่ดวงที่ 1 ดวงที่ 2 และดวงที่ 3 จะดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.7 สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘D’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ

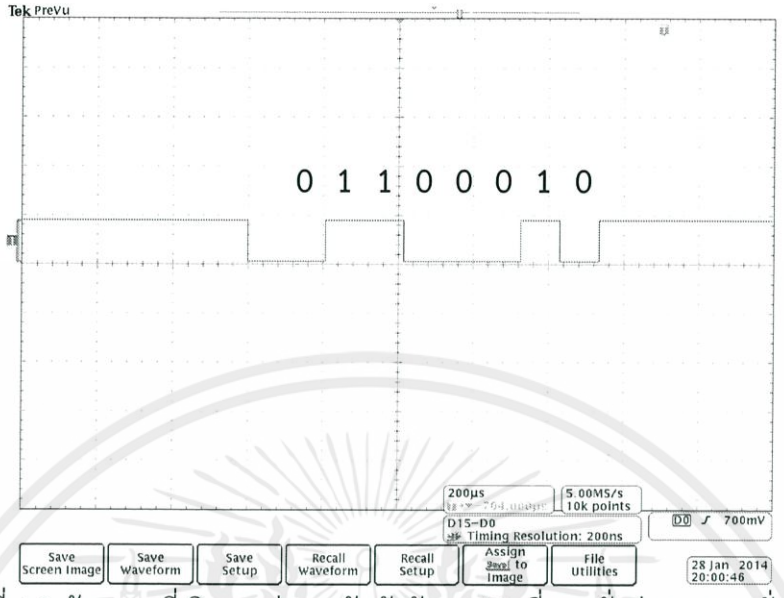
จากรูปที่ 4.7 สัญญาณที่ได้คือ 0 1 0 0 0 1 0 0 เมื่อแปลงเป็น ฐาน 16 จะได้ 44 ซึ่งคือค่า ‘D’ ที่เรากำหนดไว้ ซึ่งค่าที่ได้นี้ตรงกับการทดสอบจาก X-CTU ทำให้ไฟดวงที่ 3 ติด ส่วนดวงที่ 1 ดวงที่ 2 และดวงที่ 3 จะดับ



รูปที่ 4.8 สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘E’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ

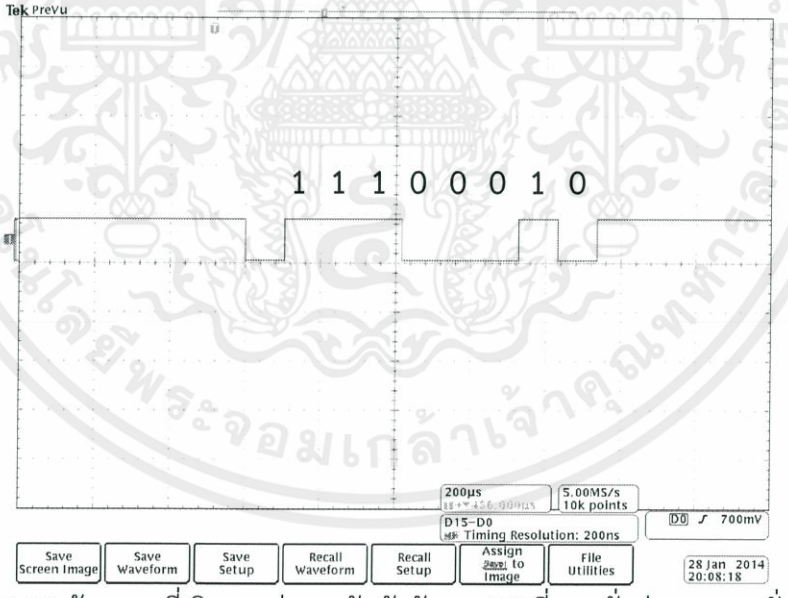
จากรูปที่ 4.8 สัญญาณที่ได้คือ 0 1 0 0 0 1 0 1 เมื่อแปลงเป็น ฐาน 16 จะได้ 45 ซึ่งคือค่า ‘E’ ที่เรากำหนดไว้ ซึ่งค่าที่ได้นี้ตรงกับการทดสอบจาก X-CTU ทำให้ไฟดวงที่ 3 และดวงที่ 4 ติด ส่วนดวงที่ 1 และดวงที่ 2 จะดับ

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.9 สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘F’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ

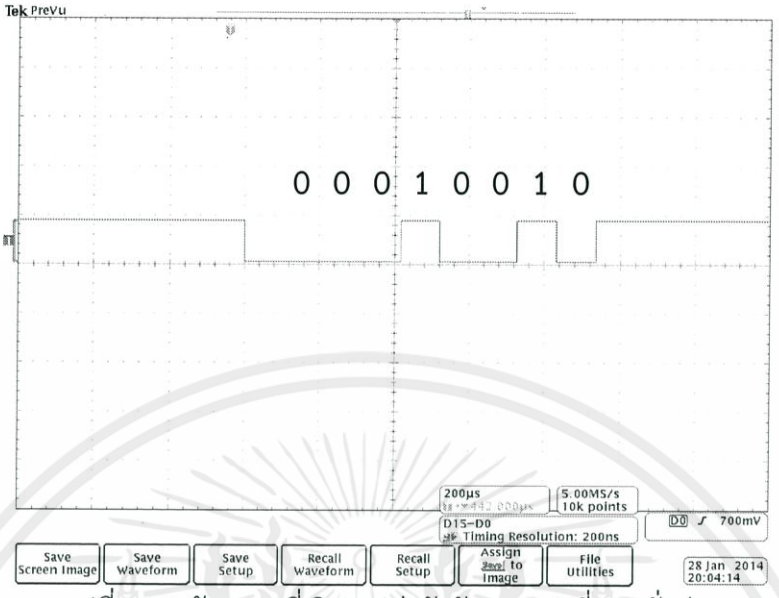
จากรูปที่ 4.9 สัญญาณที่ได้คือ 0 1 0 0 0 1 1 0 เมื่อแปลงเป็น ฐาน 16 จะได้ 46 ซึ่งคือค่า ‘F’ ที่เรากำหนดไว้ ซึ่งค่าที่ได้นี้ตรงกับการทดสอบจาก X-CTU ทำให้ไฟดวงที่ 2 ติด ส่วนดวงที่ 1 ดวงที่ 3 และดวงที่ 4 จะดับ



รูปที่ 4.10 สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘G’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ

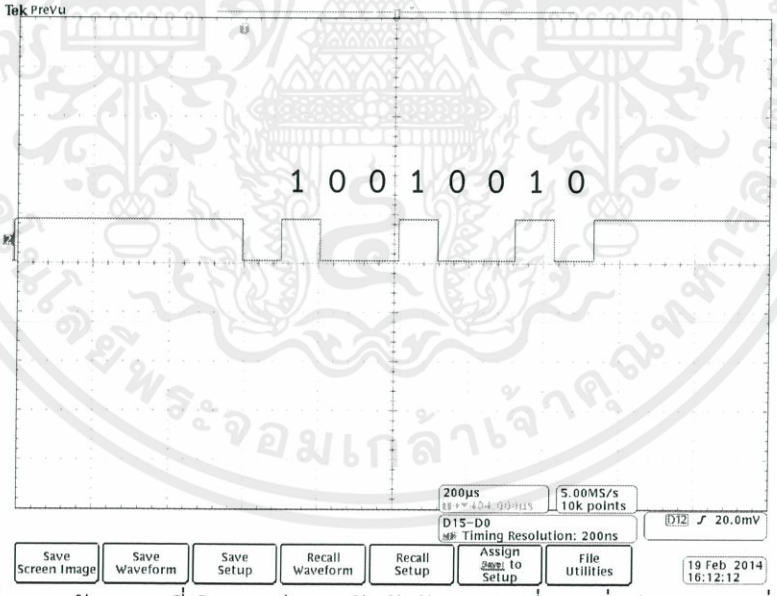
จากรูปที่ 4.10 สัญญาณที่ได้คือ 0 1 0 0 0 1 1 1 เมื่อแปลงเป็น ฐาน 16 จะได้ 47 ซึ่งคือค่า ‘G’ ที่เรากำหนดไว้ ซึ่งค่าที่ได้นี้ตรงกับการทดสอบจาก X-CTU ทำให้ไฟดวงที่ 2 และดวงที่ 4 จะติด ส่วนดวงที่ 1 และดวงที่ 3 จะดับ

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.11 สัญญาณที่เกิดจากส่งตัวอักษร ‘H’ ที่วงจรฝั่งส่ง

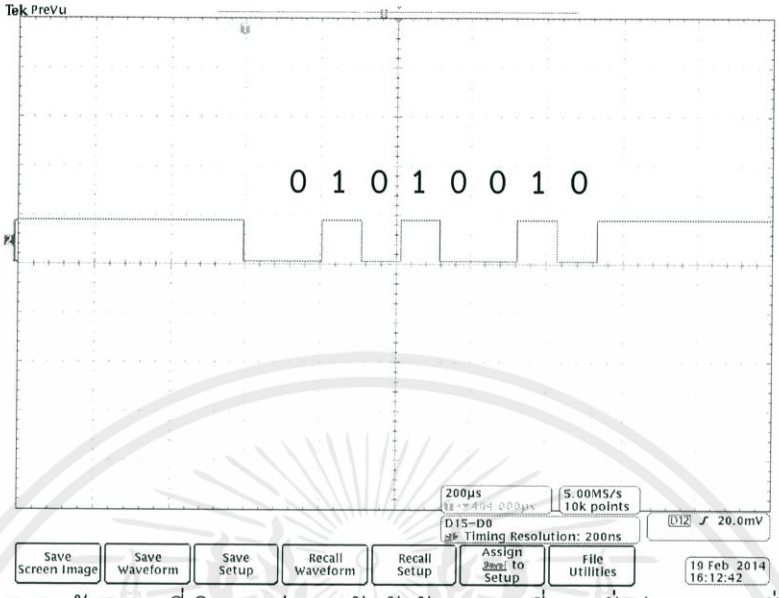
จากรูปที่ 4.11 สัญญาณที่ได้คือ 0 1 0 0 1 0 0 0 เมื่อแปลงเป็น ฐาน 16 จะได้ 48 ซึ่งคือค่า ‘H’ ที่เรากำหนดไว้ ซึ่งค่าที่ได้นี้ตรงกับการทดสอบจาก X-CTU ทำให้ไฟดวงที่ 2 และดวงที่ 3 จะติด ส่วนดวงที่ 1 และดวงที่ 4 จะดับ



รูปที่ 4.12 สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘I’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ

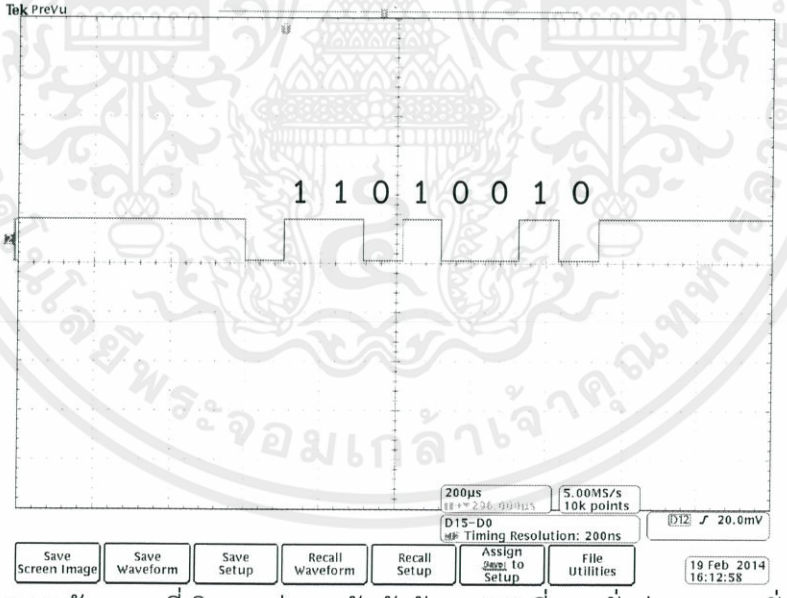
จากรูปที่ 4.12 สัญญาณที่ได้คือ 0 1 0 0 1 0 0 1 เมื่อแปลงเป็น ฐาน 16 จะได้ 49 ซึ่งคือค่า ‘I’ ที่เรากำหนดไว้ ซึ่งค่าที่ได้นี้ตรงกับการทดสอบจาก X-CTU ทำให้ไฟดวงที่ 2 ดวงที่ 3 และดวงที่ 4 จะติด ส่วนดวงที่ 1 จะดับ

เอกสารนี้มีลิขสิทธิ์ สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.13 สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘J’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ

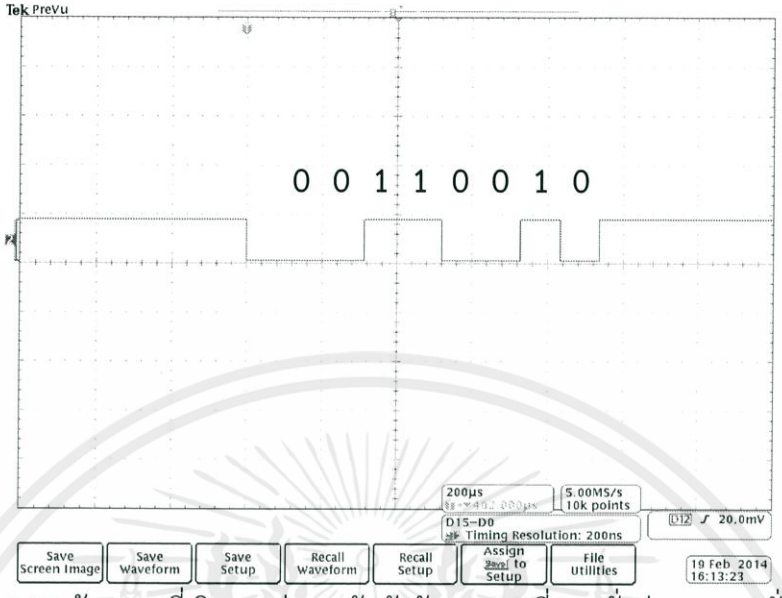
จากรูปที่ 4.13 สัญญาณที่ได้คือ 0 1 0 0 1 0 1 0 เมื่อแปลงเป็น ฐาน 16 จะได้ 4A ซึ่งคือค่า ‘J’ ที่เรากำหนดไว้ ซึ่งค่าที่ได้นี้ตรงกับการทดสอบจาก X-CTU ทำให้ไฟดวงที่ 1 ติด ส่วนดวงที่ 2 ดวงที่ 3 และดวงที่ 4 จะดับ



รูปที่ 4.14 สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘K’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ

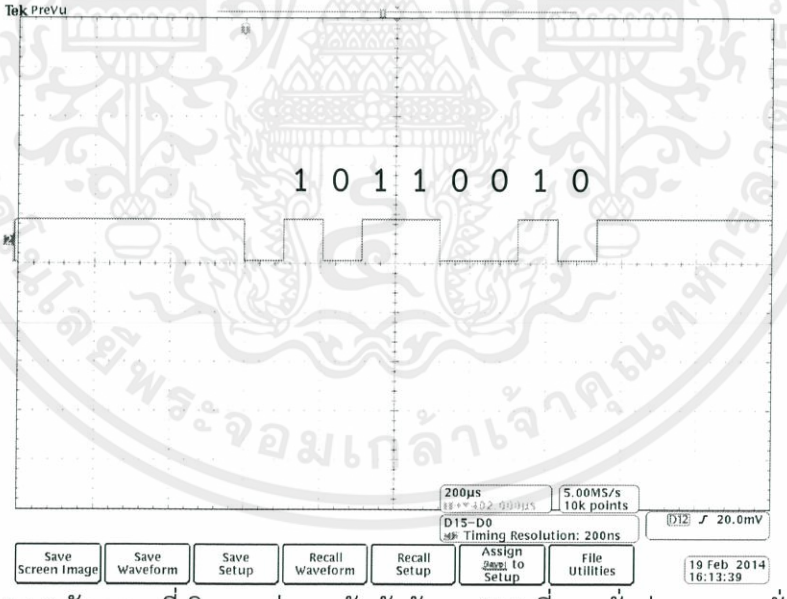
จากรูปที่ 4.14 สัญญาณที่ได้คือ 0 1 0 0 1 0 1 1 เมื่อแปลงเป็น ฐาน 16 จะได้ 4B ซึ่งคือค่า ‘K’ ที่เรากำหนดไว้ ซึ่งค่าที่ได้นี้ตรงกับการทดสอบจาก X-CTU ทำให้ไฟดวงที่ 1 และดวงที่ 4 ติด ส่วนดวงที่ 2 และดวงที่ 3 จะดับ

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.15 สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘L’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ

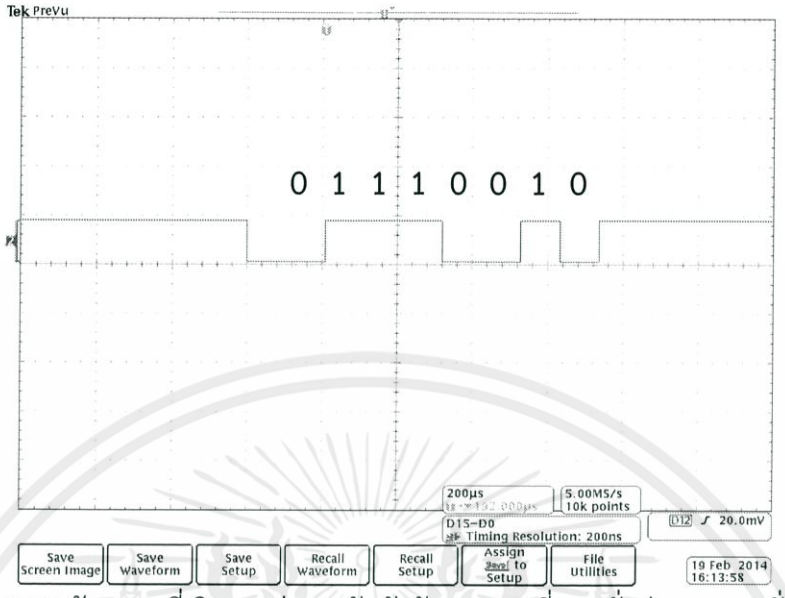
จากรูปที่ 4.15 สัญญาณที่ได้คือ 0 1 0 0 1 1 0 0 เมื่อแปลงเป็น ฐาน 16 จะได้ 4C ซึ่งคือค่า ‘L’ ที่เรากำหนดไว้ ซึ่งค่าที่ได้นี้ตรงกับการทดสอบจาก X-CTU ทำให้ไฟดวงที่ 1 และดวงที่ 3 ติด ส่วนดวงที่ 2 และดวงที่ 4 จะดับ



รูปที่ 4.16 สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘M’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ

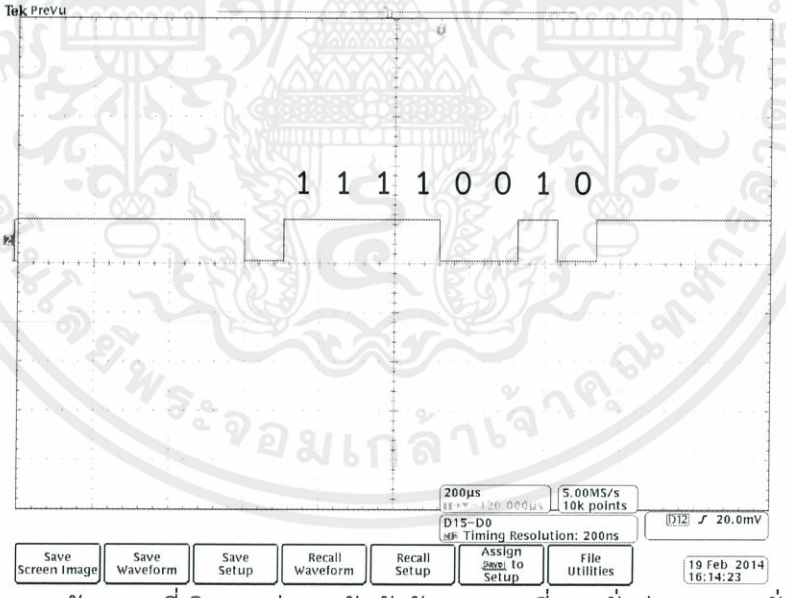
จากรูปที่ 4.16 สัญญาณที่ได้คือ 0 1 0 0 1 1 0 1 เมื่อแปลงเป็น ฐาน 16 จะได้ 4D ซึ่งคือค่า ‘M’ ที่เรากำหนดไว้ ซึ่งค่าที่ได้นี้ตรงกับการทดสอบจาก X-CTU ทำให้ไฟดวงที่ 1 ดวงที่ 3 และดวงที่ 4 ติด ส่วงดวงที่ 2 จะดับ

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.17 สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘N’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ

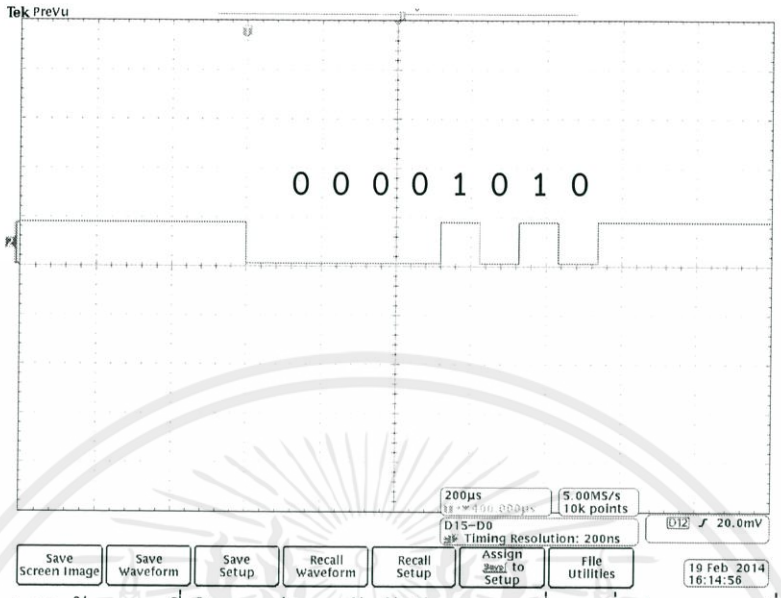
จากรูปที่ 4.17 สัญญาณที่ได้คือ 0 1 0 0 1 1 1 0 เมื่อแปลงเป็น ฐาน 16 จะได้ 4E ซึ่งคือค่า ‘N’ ที่เรากำหนดไว้ ซึ่งค่าที่ได้นี้ตรงกับการทดสอบจาก X-CTU ทำให้ไฟดวงที่ 1 และดวงที่ 2 ติด ส่วนดวงที่ 3 และดวงที่ 4 จะดับ



รูปที่ 4.18 สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘O’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ

จากรูปที่ 4.18 สัญญาณที่ได้คือ 0 1 0 0 1 1 1 1 เมื่อแปลงเป็น ฐาน 16 จะได้ 4F ซึ่งคือค่า ‘O’ ที่เรากำหนดไว้ ซึ่งค่าที่ได้นี้ตรงกับการทดสอบจาก X-CTU ทำให้ไฟดวงที่ 1 ดวงที่ 3 และดวงที่ 4 ติด ส่วนดวงที่ 2 จะดับ

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.19 สัญญาณที่เกิดจากส่งและรับตัวอักษร ‘P’ ที่วงจรฝั่งส่งและวงจรฝั่งรับ

จากรูปที่ 4.19 สัญญาณที่ได้คือ 0 1 0 1 0 0 0 0 เมื่อแปลงเป็น ฐาน 16 จะได้ 50 ซึ่งคือค่า ‘P’ ที่เรากำหนดไว้ ซึ่งค่าที่ได้นี้ตรงกับการทดสอบจาก X-CTU ทำให้ไฟดวงที่ 1 ดวงที่ 2 และดวงที่ 3 ติด ส่วนดวงที่ 4 จะดับ

จากการทดสอบค่าที่วัดได้จากขา Tx ฝั่งส่งและขา Rx ฝั่งรับ พบว่า มีค่าเหมือนกัน และมีค่าเหมือนกับการทดสอบที่ 4.2 และตรงกับที่เรากำหนดไว้ คือ ‘A’, ‘B’, ‘C’, ‘D’, ‘E’, ‘F’, ‘G’, ‘H’, ‘I’, ‘J’, ‘K’, ‘L’, ‘M’, ‘N’, ‘O’ และ ‘P’ ซึ่งค่าที่ได้จากการทดลองนี้จะอยู่ในรูปของฐาน 2 เมื่อแปลงไปเป็น ฐาน 16 แล้วจะได้ค่าที่เป็นจริง ข้อมูลเหล่านี้จะไปกำหนดทำให้ดวงไฟทั้ง 4 ติดหรือดับ ตามตารางที่ 4.1

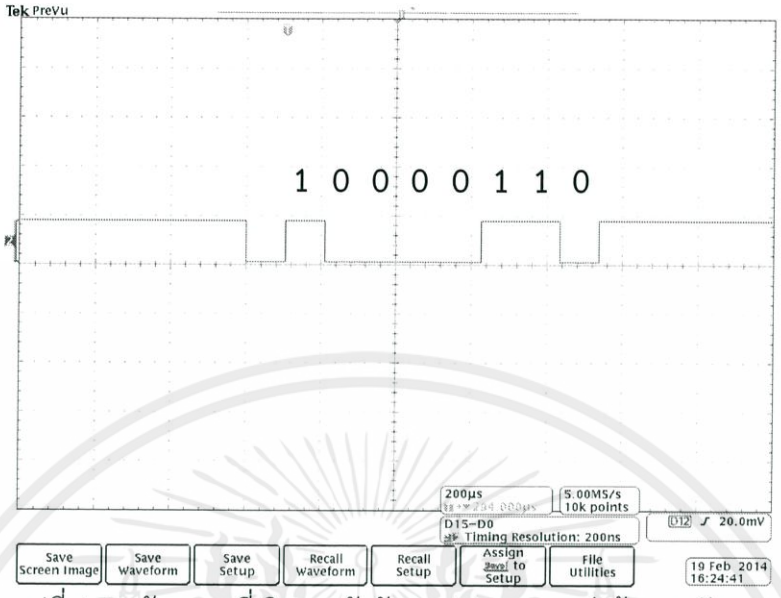
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.1 ค่าต่างๆของข้อมูลที่ฝั่งส่งและฝั่งรับ

ข้อมูล	เลขฐาน 2	เลขฐาน 16	ดวงที่ 1	ดวงที่ 2	ดวงที่ 3	ดวงที่ 4
A	01000001	41	ติด	ติด	ติด	ติด
B	01000010	42	ดับ	ดับ	ดับ	ดับ
C	01000011	43	ติด	ติด	ติด	ดับ
D	01000100	44	ดับ	ดับ	ติด	ดับ
E	01000101	45	ติด	ติด	ดับ	ดับ
F	01000110	46	ดับ	ติด	ดับ	ดับ
G	01000111	47	ดับ	ติด	ดับ	ติด
H	01001000	48	ดับ	ติด	ติด	ดับ
I	01001001	49	ดับ	ติด	ติด	ติด
J	01001010	4A	ติด	ดับ	ดับ	ดับ
K	01001011	4B	ติด	ดับ	ดับ	ติด
L	01001100	4C	ติด	ดับ	ติด	ดับ
M	01001101	4D	ติด	ดับ	ติด	ติด
N	01001110	4E	ติด	ติด	ดับ	ดับ
O	01001111	4F	ติด	ดับ	ติด	ดับ
P	01010000	50	ติด	ติด	ดับ	ดับ

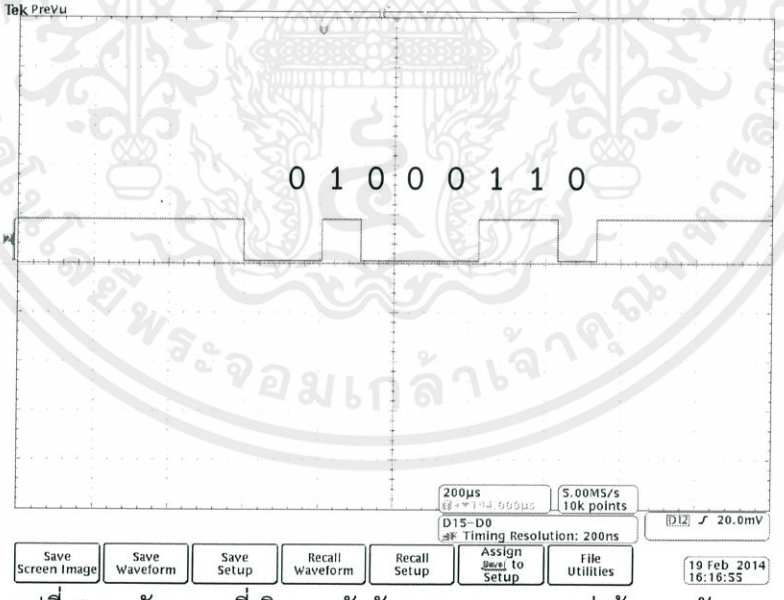
นอกจากนี้ยังมีผลการทดลองที่ได้จากการวัดค่าไฟสถานะที่แสดงว่ามีการส่งข้อมูลกลับมาจริง โดยใช้ Mixed Signal Oscilloscope เป็นเครื่องมือที่ใช้ในการวัด โดยทำการวัดค่าที่ขา 26 ของ PIC16F877 ที่วงจรฝั่งส่งที่ส่งมาจากวงจรฝั่งรับ ซึ่งทำหน้าที่เป็นขา Rx ของวงจร โดยค่าที่ส่งกลับมาได้แก่ ‘a’, ‘b’, ‘c’, ‘d’, ‘e’, ‘f’, ‘g’, ‘h’, ‘I’, ‘j’, ‘k’, ‘l’, ‘m’, ‘n’, ‘o’, และ ‘p’ โดยผลการทดลองที่ได้เป็นดังต่อไปนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.20 สัญญาณที่เกิดจากตัวอักษร ‘a’ จากการส่งข้อมูลกลับมา

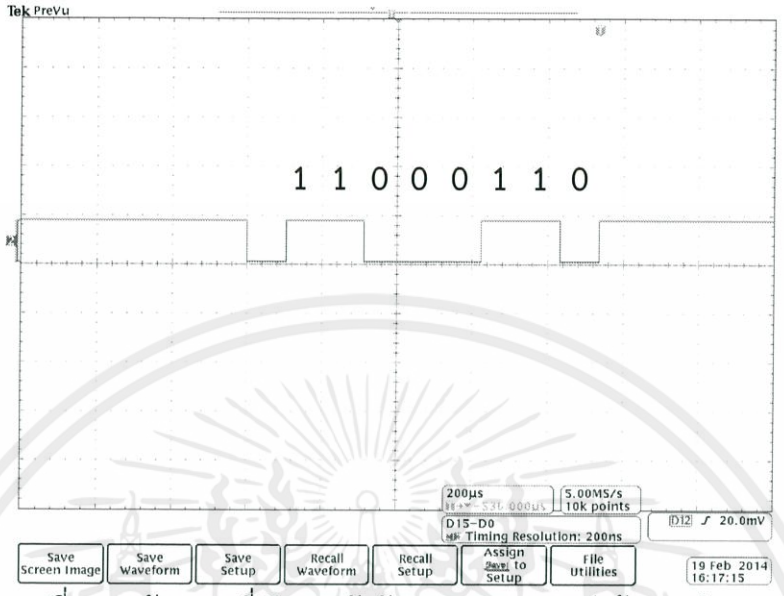
จากรูปที่ 4.20 สัญญาณที่ได้คือ 0 1 1 0 0 0 0 1 เมื่อแปลงเป็น ฐาน 16 จะได้ 61 ซึ่งคือค่า ‘a’ ที่เรากำหนดไว้



รูปที่ 4.21 สัญญาณที่เกิดจากตัวอักษร ‘b’ จากการส่งข้อมูลกลับมา

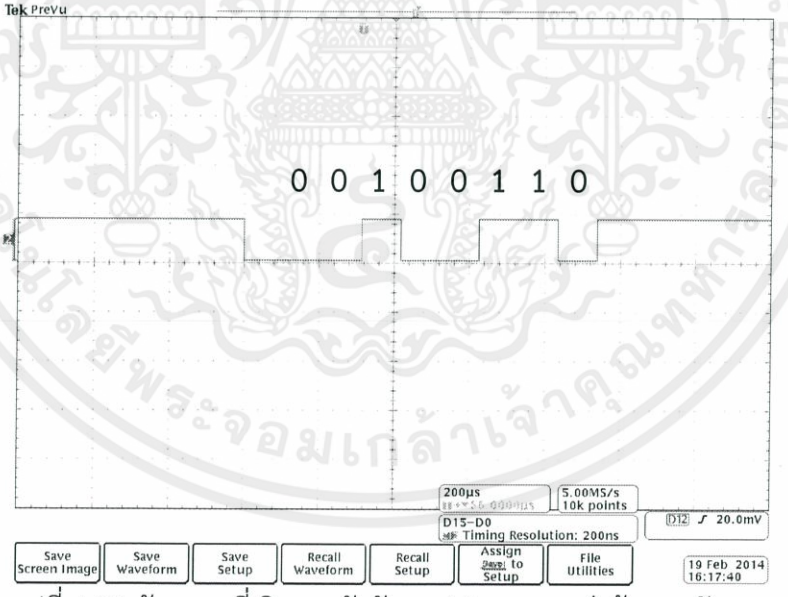
จากรูปที่ 4.21 สัญญาณที่ได้คือ 0 1 1 0 0 0 1 0 เมื่อแปลงเป็น ฐาน 16 จะได้ 62 ซึ่งคือค่า ‘b’ ที่เรากำหนดไว้

เอกสารนี้สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.22 สัญญาณที่เกิดจากตัวอักษร ‘c’ จากการส่งข้อมูลกลับมา

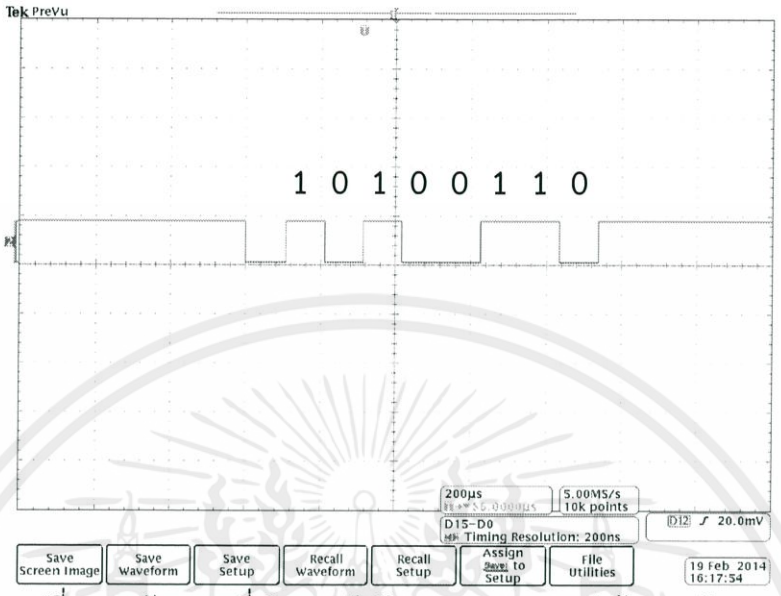
จากรูปที่ 4.22 สัญญาณที่ได้คือ 0 1 1 0 0 0 1 1 เมื่อแปลงเป็น ฐาน 16 จะได้ 63 ซึ่งคือค่า ‘c’ ที่เรากำหนดไว้



รูปที่ 4.23 สัญญาณที่เกิดจากตัวอักษร ‘d’ จากการส่งข้อมูลกลับมา

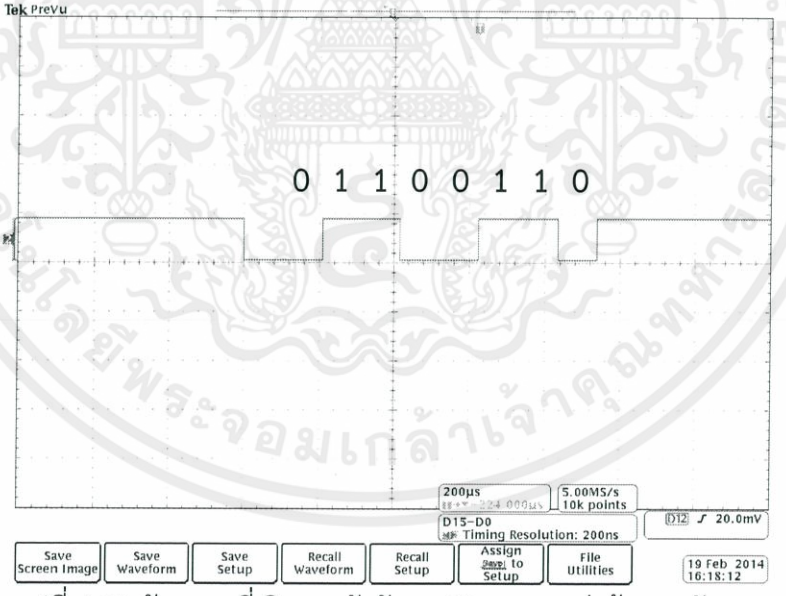
จากรูปที่ 4.23 สัญญาณที่ได้คือ 0 1 1 0 0 1 0 0 เมื่อแปลงเป็น ฐาน 16 จะได้ 64 ซึ่งคือค่า ‘d’ ที่เรากำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.24 สัญญาณที่เกิดจากตัวอักษร ‘e’ จากการส่งข้อมูลกลับมา

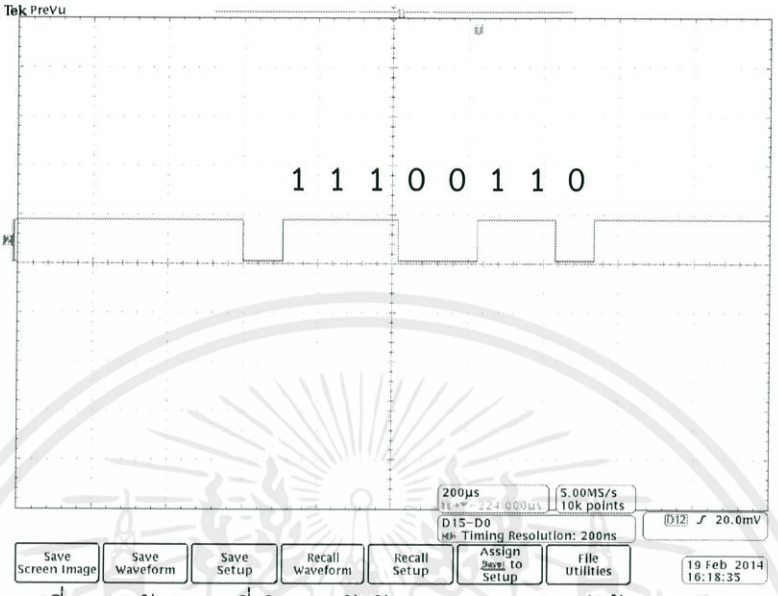
จากรูปที่ 4.24 สัญญาณที่ได้คือ 0 1 1 0 0 1 0 1 เมื่อแปลงเป็น ฐาน 16 จะได้ 65 ซึ่งคือค่า ‘e’ ที่เรากำหนดไว้



รูปที่ 4.25 สัญญาณที่เกิดจากตัวอักษร ‘f’ จากการส่งข้อมูลกลับมา

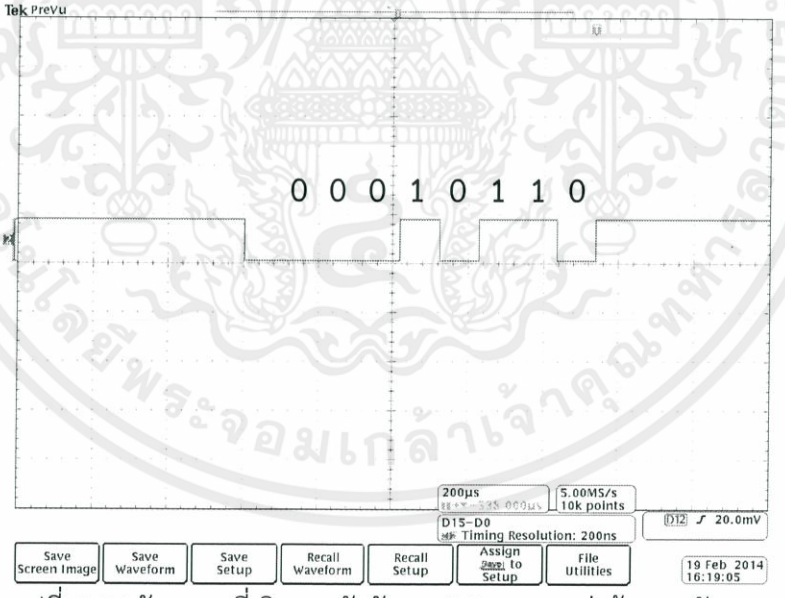
จากรูปที่ 4.25 สัญญาณที่ได้คือ 0 1 1 0 0 1 1 0 เมื่อแปลงเป็น ฐาน 16 จะได้ 66 ซึ่งคือค่า ‘f’ ที่เรากำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.26 สัญญาณที่เกิดจากตัวอักษร ‘g’ จากการส่งข้อมูลกลับมา

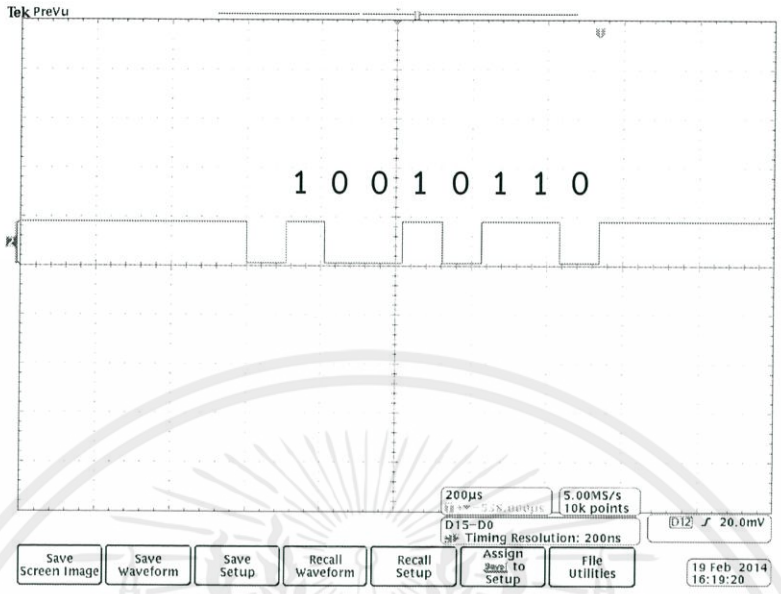
จากรูปที่ 4.26 สัญญาณที่ได้คือ 0 1 1 0 0 1 1 1 เมื่อแปลงเป็น ฐาน 16 จะได้ 67 ซึ่งคือค่า ‘g’ ที่เรากำหนดไว้



รูปที่ 4.27 สัญญาณที่เกิดจากตัวอักษร ‘h’ จากการส่งข้อมูลกลับมา

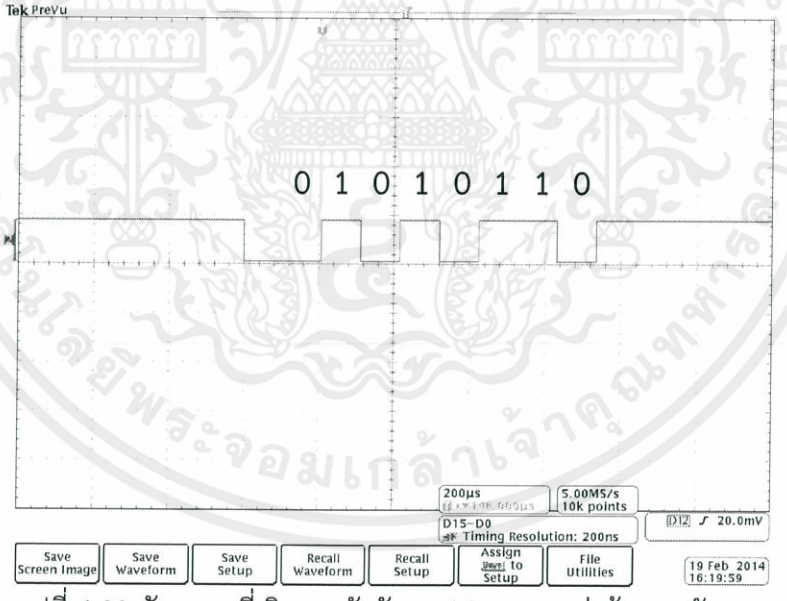
จากรูปที่ 4.27 สัญญาณที่ได้คือ 0 1 1 0 1 0 0 0 เมื่อแปลงเป็น ฐาน 16 จะได้ 68 ซึ่งคือค่า ‘h’ ที่เรากำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.28 สัญญาณที่เกิดจากตัวอักษร ‘i’ จากการส่งข้อมูลกลับมา

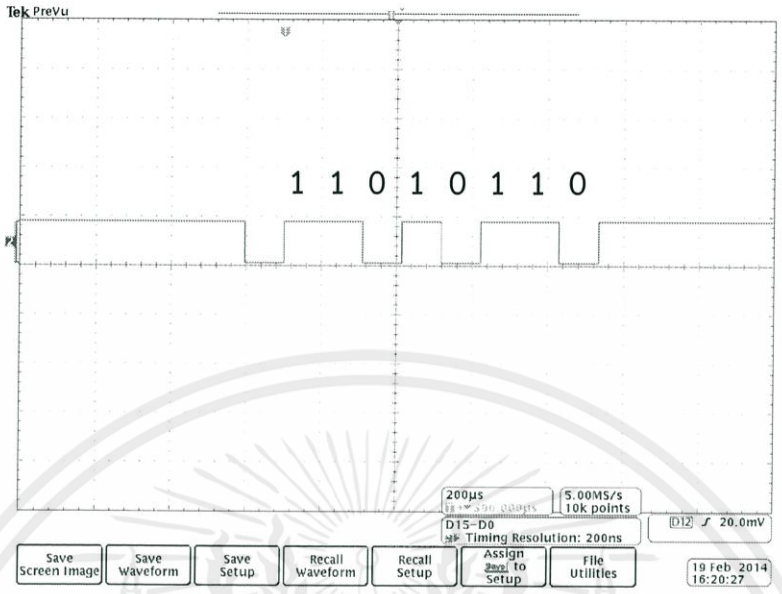
จากรูปที่ 4.28 สัญญาณที่ได้คือ 0 1 1 0 1 0 0 1 เมื่อแปลงเป็น ฐาน 16 จะได้ 69 ซึ่งคือค่า ‘i’ ที่เรากำหนดไว้



รูปที่ 4.29 สัญญาณที่เกิดจากตัวอักษร ‘j’ จากการส่งข้อมูลกลับมา

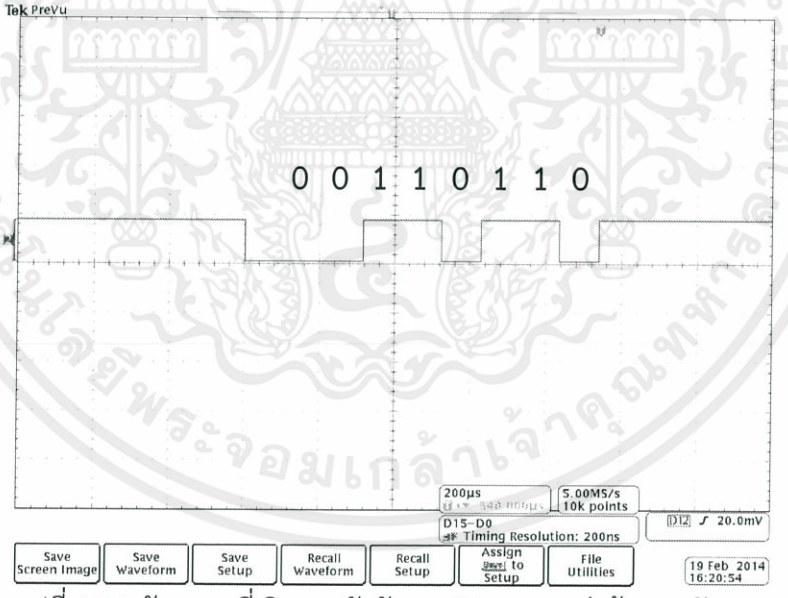
จากรูปที่ 4.29 สัญญาณที่ได้คือ 0 1 1 0 1 0 1 0 เมื่อแปลงเป็น ฐาน 16 จะได้ 6A ซึ่งคือค่า ‘j’ ที่เรากำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.30 สัญญาณที่เกิดจากตัวอักษร ‘k’ จากการส่งข้อมูลกลับมา

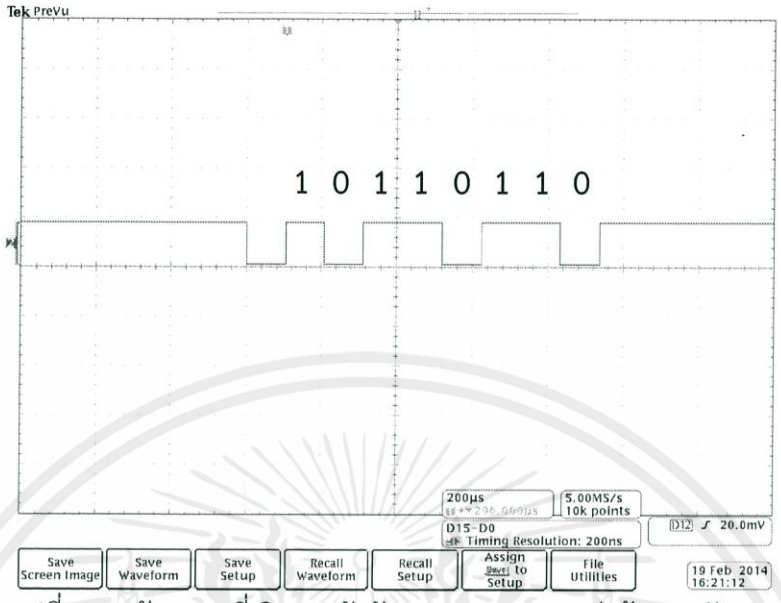
จากรูปที่ 4.30 สัญญาณที่ได้คือ 0 1 1 0 1 0 1 1 เมื่อแปลงเป็น ฐาน 16 จะได้ 6B ซึ่งคือค่า ‘k’ ที่เรากำหนดไว้



รูปที่ 4.31 สัญญาณที่เกิดจากตัวอักษร ‘l’ จากการส่งข้อมูลกลับมา

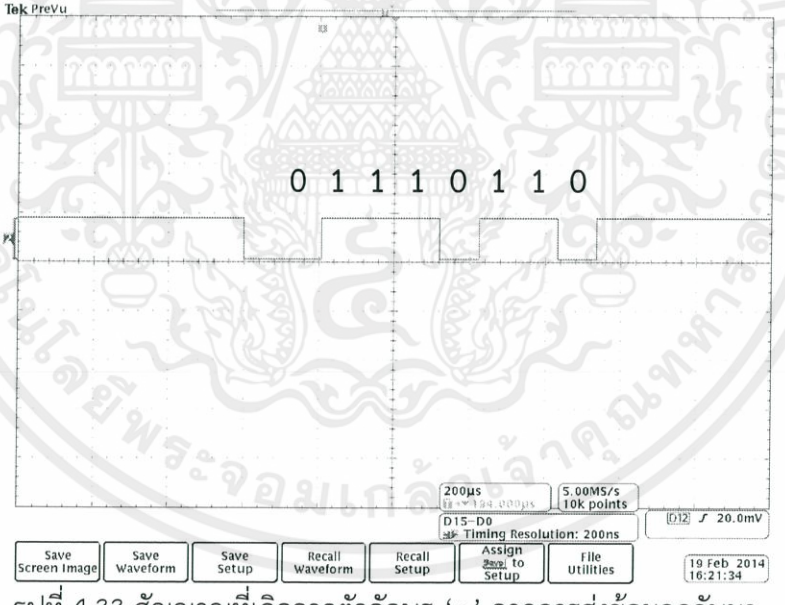
จากรูปที่ 4.31 สัญญาณที่ได้คือ 0 1 1 0 1 1 0 0 เมื่อแปลงเป็น ฐาน 16 จะได้ 6C ซึ่งคือค่า ‘l’ ที่เรากำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.32 สัญญาณที่เกิดจากตัวอักษร ‘m’ จากการส่งข้อมูลกลับมา

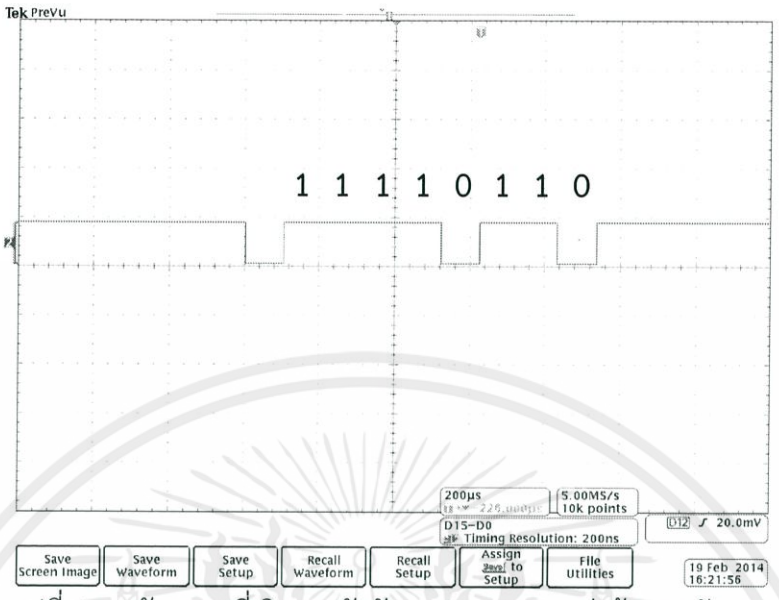
จากรูปที่ 4.32 สัญญาณที่ได้คือ 0 1 1 0 1 1 0 1 เมื่อแปลงเป็น ฐาน 16 จะได้ 6D ซึ่งคือค่า ‘m’ ที่เรากำหนดไว้



รูปที่ 4.33 สัญญาณที่เกิดจากตัวอักษร ‘n’ จากการส่งข้อมูลกลับมา

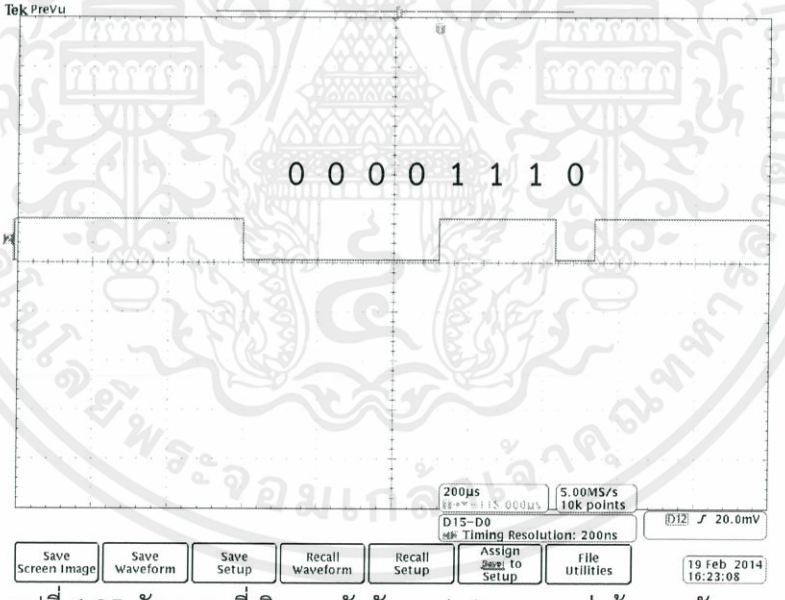
จากรูปที่ 4.33 สัญญาณที่ได้คือ 0 1 1 0 1 1 1 0 เมื่อแปลงเป็น ฐาน 16 จะได้ 6E ซึ่งคือค่า ‘n’ ที่เรากำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.34 สัญญาณที่เกิดจากตัวอักษร ‘o’ จากการส่งข้อมูลกลับมา

จากรูปที่ 4.34 สัญญาณที่ได้คือ 0 1 1 0 1 1 1 1 เมื่อแปลงเป็น ฐาน 16 จะได้ 6F ซึ่งคือค่า ‘o’ ที่เรากำหนดไว้



รูปที่ 4.35 สัญญาณที่เกิดจากตัวอักษร ‘p’ จากการส่งข้อมูลกลับมา

จากรูปที่ 4.35 สัญญาณที่ได้คือ 0 1 1 1 0 0 0 0 เมื่อแปลงเป็น ฐาน 16 จะได้ 70 ซึ่งคือค่า ‘p’ ที่เรากำหนดไว้

ซึ่งเราสามารถสรุปข้อมูลต่างๆได้ดังตารางที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 ค่าข้อมูลที่ส่งกลับมา

ข้อมูล	เลขฐาน 2	เลขฐาน 16	LED#1	LED#2	LED#3	LED#4
a	01100001	61	ติด	ติด	ติด	ติด
b	01100010	62	ดับ	ดับ	ดับ	ดับ
c	01100011	63	ติด	ติด	ติด	ดับ
d	01100100	64	ดับ	ดับ	ติด	ดับ
e	01100101	65	ติด	ติด	ดับ	ดับ
f	01100110	66	ดับ	ติด	ดับ	ดับ
g	01100111	67	ดับ	ติด	ดับ	ติด
h	01101000	68	ดับ	ติด	ติด	ดับ
i	01101001	69	ดับ	ติด	ติด	ติด
j	01101010	6A	ติด	ดับ	ดับ	ดับ
k	01101011	6B	ติด	ดับ	ดับ	ติด
l	01101100	6C	ติด	ดับ	ติด	ติด
m	01101101	6D	ติด	ดับ	ติด	ติด
n	01101110	6E	ติด	ติด	ดับ	ดับ
o	01101111	6F	ติด	ดับ	ติด	ดับ
p	01110000	70	ติด	ติด	ดับ	ดับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

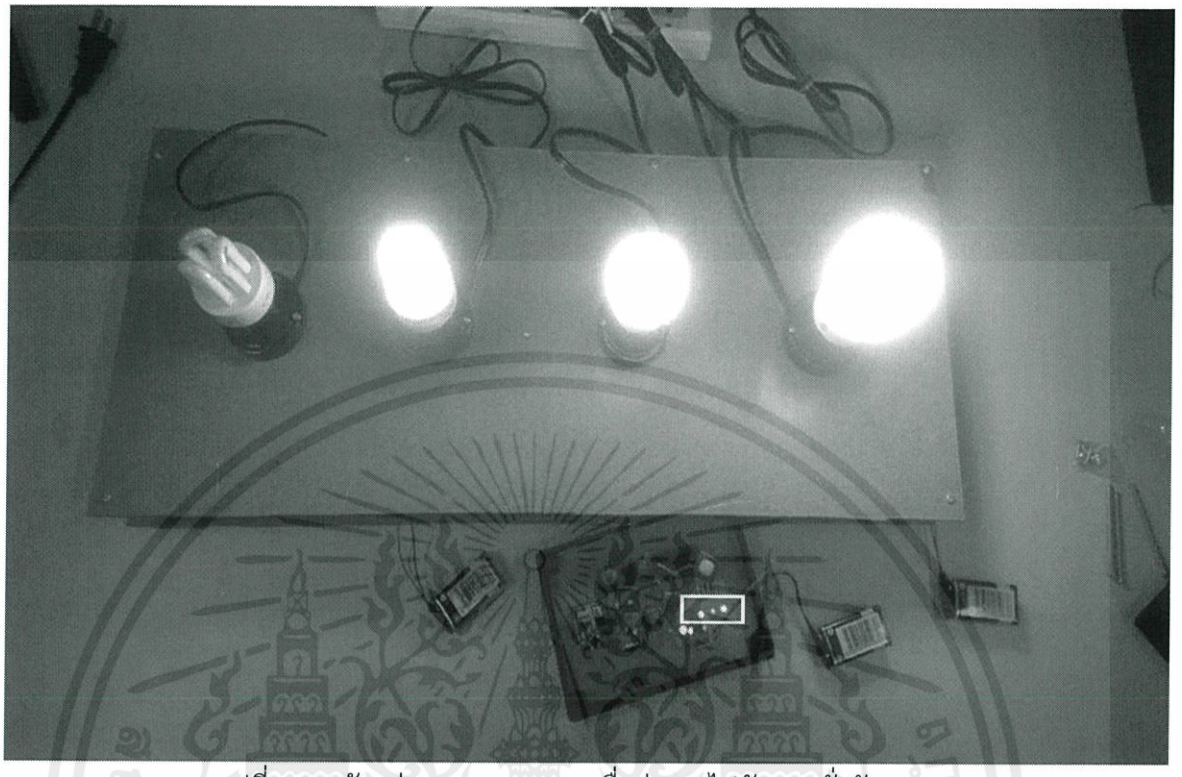
4.5 ผลการทดสอบการเปิด – ปิดหลอดไฟด้วยอุปกรณ์จริง

จากการทดสอบการเปิด – ปิดหลอดไฟด้วยอุปกรณ์จริง เพื่อทดสอบว่าอุปกรณ์ที่สร้างขึ้นนั้นสามารถนำไปใช้งานได้จริง โดยจะทำการป้อนแรงดันไฟฟ้าขนาด 9V เข้าที่วงจรฝั่งส่ง วงจรฝั่งรับที่ 1 และวงจรฝั่งรับที่ 2 โดยจะทำการตรวจสอบว่าหลอดไฟติดหรือดับสอดคล้องกับค่าที่กำหนดไว้หรือไม่ และเป็นการแสดงผลไฟสถานะของวงจรฝั่งส่งว่ามีการส่งข้อมูลกลับมาจริง โดยผลการทดสอบเป็นไปดังตัวอย่างต่อไปนี้

รูปที่ 4.36 ตัวอย่างผลการทดสอบเมื่อส่ง ‘A’ ไปยังวงจรฝั่งรับ

จากการทดสอบการส่งค่า ‘A’ ไปยังวงจรฝั่งส่ง สัญญาณที่ได้คือ 0 1 0 0 0 0 0 1 เมื่อแปลงเป็น ฐาน 16 จะได้ 41 ผลปรากฏว่าหลอดไฟทั้ง 4 ดวงติดหมด และส่งค่ากลับมายังวงจรฝั่งส่ง โดยสังเกตได้จากไฟ LED ติดหมดทั้ง 4 ดวง ซึ่งสัญญาณที่ได้คือ 0 1 1 0 0 0 0 1 เมื่อแปลงเป็น ฐาน 16 จะได้ 61 ซึ่งคือค่า ‘a’ ที่เรากำหนดไว้

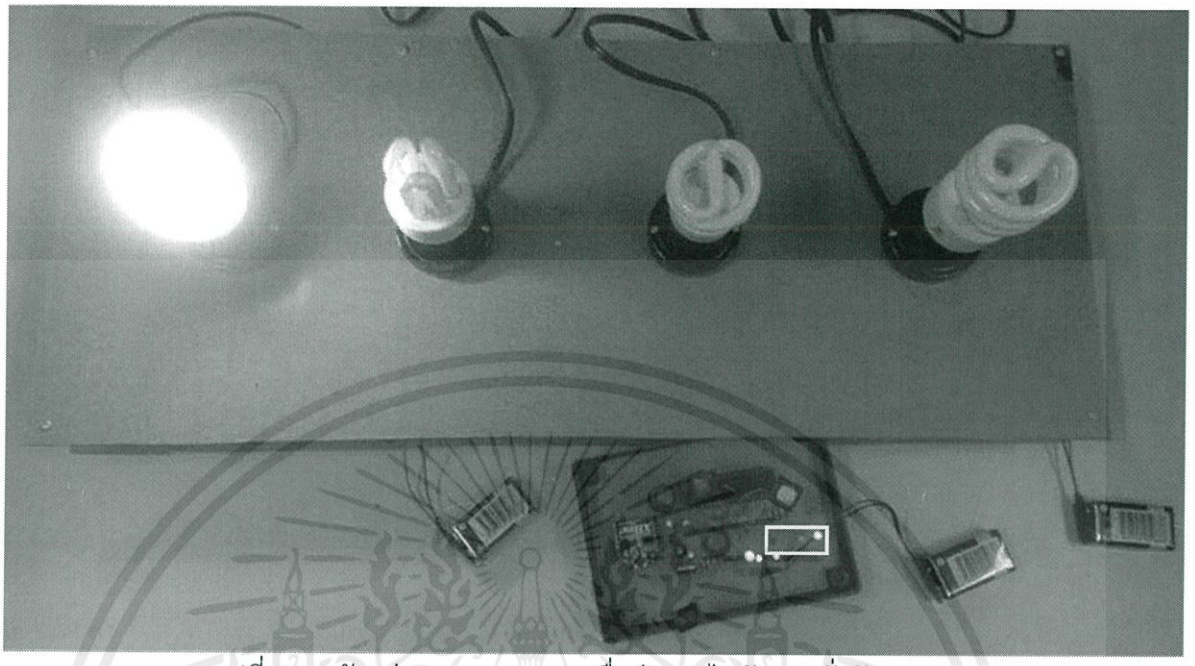
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.37 ตัวอย่างผลการทดสอบเมื่อส่ง ‘I’ ไปยังวงจรฝั่งรับ

จากการทดสอบการส่งค่า ‘I’ ไปยังวงจรฝั่งส่ง สัญญาณที่ได้คือ 0 1 0 0 1 0 0 1 เมื่อแปลงเป็น ฐาน 16 จะได้ 49 ผลปรากฏว่าหลอดไฟดวงที่ 1 ดับ ส่วนดวงที่ 2 ดวงที่ 3 และดวงที่ 4 จะติด และส่งค่ากลับมายังวงจรฝั่งส่ง โดยสังเกตได้จากไฟ LED ดวงที่ 1 ดับ ส่วนดวงที่ 2 ดวงที่ 3 และดวงที่ 4 จะดับ ซึ่งสัญญาณที่ได้คือ 0 1 1 0 1 0 0 1 เมื่อแปลงเป็น ฐาน 16 จะได้ 69 ซึ่งคือค่า ‘i’ ที่เรากำหนดไว้

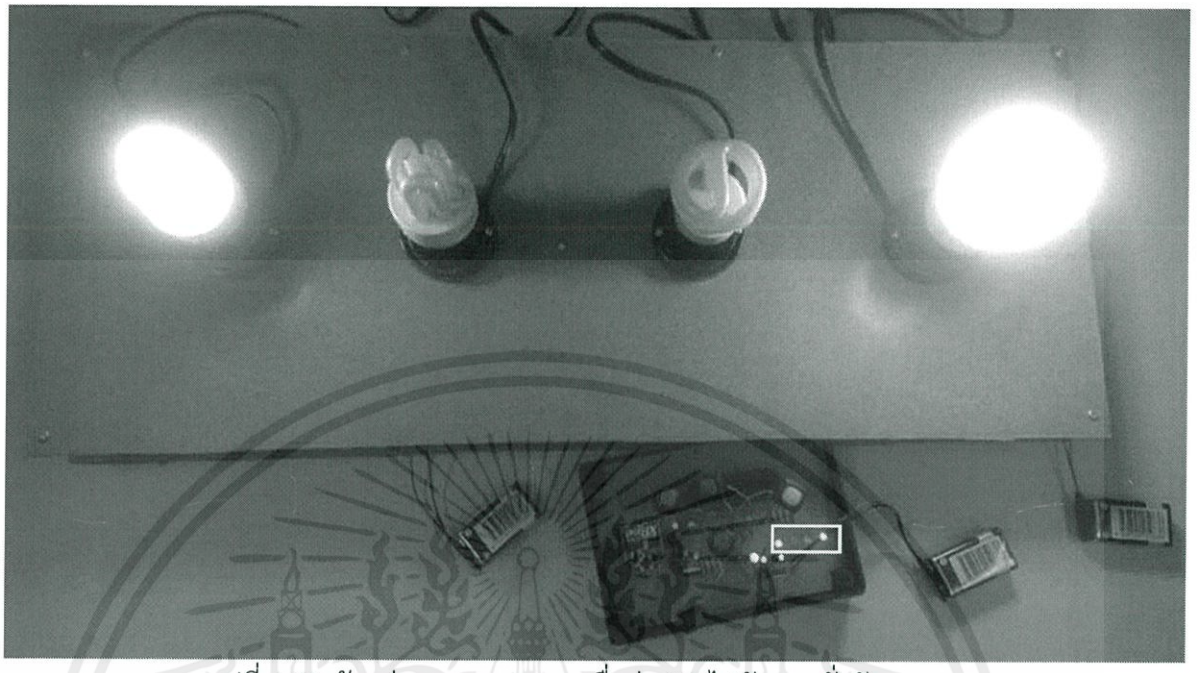
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.38 ตัวอย่างผลการทดสอบเมื่อส่ง ‘J’ ไปยังวงจรฝั่งรับ

จากการทดสอบการส่งค่า ‘J’ ไปยังวงจรฝั่งส่ง สัญญาณที่ได้คือ 0 1 0 0 1 0 0 1 เมื่อแปลงเป็น ฐาน 16 จะได้ 4A ผลปรากฏว่าหลอดไฟดวงที่ 1 ติด ส่วนดวงที่ 2 ดวงที่ 3 และดวงที่ 4 จะดับ และส่งค่ากลับมายังวงจรฝั่งส่ง โดยสังเกตได้จากไฟ LED ดวงที่ 1 ติด ส่วนดวงที่ 2 ดวงที่ 3 และดวงที่ 4 จะดับ ซึ่งสัญญาณที่ได้คือ 0 1 1 0 1 0 1 0 เมื่อแปลงเป็น ฐาน 16 จะได้ 6A ซึ่งคือค่า ‘j’ ที่เรากำหนดไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.39 ตัวอย่างผลการทดสอบเมื่อส่ง ‘K’ ไปยังวงจรฝั่งรับ

จากการทดสอบการส่ง ‘K’ ไปยังวงจรฝั่งส่ง สัญญาณที่ได้คือ 0 1 0 0 1 0 1 1 เมื่อแปลงเป็น ฐาน 16 จะได้ 49 ผลปรากฏว่าหลอดไฟดวงที่ 1 และดวงที่ 2 ติด ส่วนดวงที่ 3 และดวงที่ 4 จะดับ และส่งค่ากลับมายังวงจรฝั่งส่ง โดยสังเกตได้จากไฟ LED ดวงที่ 1 และดวงที่ 4 จะติด ส่วนดวงที่ 2 และดวงที่ 3 จะดับ ซึ่งสัญญาณที่ได้คือ 0 1 1 0 1 0 1 1 เมื่อแปลงเป็น ฐาน 16 จะได้ 6B ซึ่งคือค่า ‘k’ ที่เรากำหนดไว้

จากผลการทดสอบการเปิด – ปิดหลอดไฟด้วยอุปกรณ์จริงพบว่า หลอดไฟติด – ดับสอดคล้องกับผลการทดลองตามตารางที่ 4.1 และไฟสถานะตรวจสอบข้อมูลที่ส่งกลับมาสอดคล้องกับผลการทดลองตามตารางที่ 4.2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลและข้อเสนอแนะ

5.1 สรุปผล

ปริญญานิพนธ์หัวข้อเรื่องอุปกรณ์ควบคุมการเปิด-ปิด หลอดไฟ จะใช้สวิตช์ชนิดกดติด-กดดับเป็นอุปกรณ์กำหนดการติด-ดับของหลอดไฟ และ Microcontroller เป็นตัวควบคุมการทำงานทั้งหมดซึ่งจะมีอยู่ 3 ตัวคือที่วงจรฝั่งส่ง วงจรฝั่งรับที่ 1 และวงจรฝั่งรับที่ 2 จากนั้นเราจะใช้ Zigbee เป็นอุปกรณ์ในการส่งข้อมูลจากฝั่งส่งไปยังฝั่งรับ เมื่อฝั่งรับได้รับข้อมูลจะนำมาประมวลผลว่าหลอดไฟดวงใดติดบ้างซึ่งจะควบคุมการติด-ดับด้วย Relay และในการทำปริญญานิพนธ์นี้ทางผู้จัดทำได้แบ่งการเก็บผลการทดลองออกเป็น 5 ส่วนใหญ่ๆ ได้แก่

1. ผลการทดสอบการส่งข้อมูลผ่าน Zigbee ทั้ง 3 ตัว เพื่อทดสอบว่า Zigbee ทั้ง 3 ตัว สามารถติดต่อกันและสามารถนำมาใช้งานได้จริงตามที่เราได้ตั้งค่าไว้

2. ผลการทดสอบการส่งข้อมูลของฝั่งส่งโดยใช้โปรแกรม X-CTU โดยต่อ Zigbee#1 กับวงจรฝั่งส่ง และต่อ Zigbee#2 กับคอมพิวเตอร์ เพื่อทดสอบว่าค่าที่ส่งมายัง Zigbee#2 สามารถส่งได้และถูกต้องแม่นยำโดยไม่มีการตกหล่นของข้อมูล โดยแสดงผลบนโปรแกรม X-CTU

3. ผลการทดสอบผ่าน Spectrum Analyzer เพื่อทดสอบว่าความถี่ของ Zigbee ตรงตามทฤษฎีหรือไม่ โดยคำนวณจาก Center Frequency = 2.405 + (CH - 10) x 5 MHz โดย CH ในที่นี้เป็นฐาน 10 โดยตั้งค่า Center Frequency ให้เท่ากับความถี่ของอุปกรณ์

4. ผลการทดสอบผ่าน Mixed Signal oscilloscope เพื่อทดสอบดูจำนวนบิตที่ทางฝั่งส่งได้ส่งไปยังฝั่งรับ ว่าตรงตามที่กำหมดค่าไว้หรือไม่ พร้อมทั้งตรวจสอบว่า ฝั่งรับรับค่าที่ถูกต้องหรือไม่ จากนั้นทำการเช็คค่าที่ส่งกลับมาว่ามีการส่งค่ากลับมาจริงตามที่กำหนดไว้

5. ผลการทดสอบอุปกรณ์ควบคุมการเปิด - ปิดหลอดไฟ เพื่อทดสอบว่าหลอดไฟติดดับตรงตามที่กำหนดหรือไม่ และตรวจสอบว่าไฟสถานะตรวจสอบค่าที่ส่งกลับมาหรือไม่

สรุปจากการทดลองทั้ง 5 โดยเมื่อเราเช็คข้อมูลที่ขา 25 คือ Tx ของฝั่งส่งเทียบกับขา 26 คือ Rx ของฝั่งรับจากตัว Microcontroller มีค่าเท่ากันและมีค่าตรงไปตามที่เรากำหนดไว้ ทั้งการเช็คจากโปรแกรม X-CTU หรือการเช็คจาก Oscilloscope ก็ตาม รวมไปถึงการเช็คความถี่ของ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อุปกรณ์ก็ตรงตามทฤษฎีทุกประการ นอกจากนี้อุปกรณ์ควบคุมการเปิด – ปิดหลอดไฟที่สร้างขึ้นสามารถนำไปใช้งานได้จริงและทำงานได้ตามเงื่อนไขที่กำหนดอย่างถูกต้อง

5.2 ข้อเสนอแนะ

ผู้จัดทำปริญญานิพนธ์นี้คาดหวังว่า งานชิ้นนี้จะสามารถอำนวยความสะดวกให้แก่ผู้ใช้ไม่มากก็น้อย ทางผู้จัดทำยังจะคิดค้นพัฒนาให้ชิ้นงานมีความสมบูรณ์มากขึ้นไปอีก เนื่องจากงานชิ้นนี้ยังมีจุดบกพร่องอยู่ ได้แก่

1. Zigbee ที่ใช้เป็น Zigbee Series 2 ทำให้ไม่สามารถกำหนดความถี่ในแต่ละช่วงสัญญาณได้ ต้องใช้ Zigbee Pro Series 1 ถึงจะสามารถกำหนดความถี่ในแต่ละช่วงสัญญาณได้ ส่งผลให้เมื่อมีการใช้อุปกรณ์ที่มีย่านความถี่เดียวกันทำให้เกิดปัญหาได้
2. อุปกรณ์ที่ใช้ในการควบคุมมีขนาดใหญ่เกินไป ไม่สะดวกต่อการพกพา ควรจะมีการปรับปรุงแก้ไขเพิ่มเติม
3. แหล่งจ่ายไฟมีพลังงานไม่พอเพียงต่อการใช้งานเป็นเวลานานๆ ทำให้สิ้นเปลืองพลังงานอย่างมาก ควรจะมีการจัดหาแหล่งจ่ายไฟที่มีขนาดเล็กและมีพลังงานเพียงพอต่ออุปกรณ์ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม

[1] ดอนสัน ปงผาบ และทิพวัลย์ คำน้ำนอง, *ไมโครคอนโทรเลอร์ PICและการประยุกต์ใช้งาน.* พิมพ์ครั้งที่ 5, กรุงเทพฯ : สมาคมส่งเสริมเทคโนโลยี (ไทย-ญี่ปุ่น), 2550.

[2] อรพิน ประวัติบริสุทธิ์, *คู่มือเรียนภาษาซี ฉบับปรับปรุงใหม่,* กรุงเทพฯ : โปรวิชั่น, 2552.

[3] Zigbee and Wifi RF Channels. [ออนไลน์]. เข้าถึงจาก : http://fosiao.com/taxonomy/term/18. [สืบค้นวันที่ 12 พฤศจิกายน 2556]

[4] Xbee ใช้ง่ายกว่าที่คิด [ออนไลน์]. เข้าถึงจาก : http://wara.com/article-736.html. [สืบค้นวันที่ 15 พฤศจิกายน 2556]

[5] การต่อไมโครคอนโทรลเลอร์กับ Xbee เบื้องต้น. [ออลไลน์]. เข้าถึงจาก : http://www.thaieasyelec.com/Review-Product-Article/xbee-with-microcontroller-PIC16F877.html. [สืบค้นวันที่ 20 พฤศจิกายน 2556]

[6] การเขียนโปรแกรมควบคุมไมโครคอนโทรลเลอร์เบื้องต้นด้วยภาษาซี PIC C. [ออลไลน์]. เข้าถึงจาก : http://idea2make.com/?p=261. [สืบค้นวันที่ 3 ธันวาคม 2556]

[7] PIC16F87X DataSheet – MICROSHIP. [ออนไลน์]. เข้าถึงได้จาก: ww1.microchip.com/downloads/en/devicedoc/30292c.pdf. [สืบค้นวันที่ 9 ธันวาคม 2556]

[8] Xbee Series 2 DataSheet . [ออนไลน์]. เข้าถึงได้จาก : http://ftp1.digi.com/support/documentation/90000866_A.pdf. [สืบค้นวันที่ 11 ธันวาคม 2556]

[9] หน้าที่ของรีเลย์. [ออนไลน์]. เข้าถึงจาก : http://www.semi-shop.com/main.php. [สืบค้นวันที่ 6 มกราคม 2557]

[10] ลักษณะทางกายภาพของ Xbee. [ออนไลน์]. เข้าถึงได้จาก : http://roy.vanegas.org. [สืบค้นวันที่ 24 พฤศจิกายน 2556]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ก

ผังงานโปรแกรม (Program Flowchart)

การเขียนผังโปรแกรมจะประกอบไปด้วยการใช้สัญลักษณ์มาตรฐานต่าง ๆ ที่เรียกว่า สัญลักษณ์ ANSI (American National Standards Institute) ในการสร้างผังงาน ดังตัวอย่างที่แสดงในรูปต่อไปนี้

ตารางที่ ก สัญลักษณ์ต่างๆในFlow Chart

	จุดเริ่มต้น / สิ้นสุดของโปรแกรม
	ลูกศรแสดงทิศทางการทำงานของโปรแกรมและการไหลของข้อมูล
	ใช้แสดงคำสั่งในการประมวลผล หรือการกำหนดค่าข้อมูลให้กับตัวแปร
	แสดงการอ่านข้อมูลจากหน่วยเก็บข้อมูลสำรองเข้าสู่หน่วยความจำหลักภายในเครื่องหรือการแสดงผลลัพธ์จากการประมวลผลออกมา
	การตรวจสอบเงื่อนไขเพื่อตัดสินใจ โดยจะมีเส้นออกจารรูปเพื่อแสดงทิศทางการทำงานต่อไป เงื่อนไขเป็นจริงหรือเป็นเท็จ
	แสดงจุดเชื่อมต่อของผังงานภายใน หรือเป็นที่บรรจบของเส้นหลายเส้นที่มาจากหลายทิศทางเพื่อจะไปสู่การทำงานอย่างใดอย่างหนึ่งที่เหมือนกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ข

ขาของ PIC 16F877

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
OSC1/CLKIN	13	14	30	I	ST/CMOS(4)	Oscillator crystal input/external clock source input.
OSC2/CLKOUT	14	15	31	O	—	Oscillator crystal output. Connects to crystal or resonator in crystal oscillator mode. In RC mode, OSC2 pin outputs CLKOUT which has 1/4 the frequency of OSC1, and denotes the instruction cycle rate.
$\overline{MCLR}$/VPP	1	2	18	I/P	ST	Master Clear (Reset) input or programming voltage input. This pin is an active low RESET to the device.
						PORTA is a bi-directional I/O port.
RA0/AN0	2	3	19	I/O	TTL	RA0 can also be analog input0.
RA1/AN1	3	4	20	I/O	TTL	RA1 can also be analog input1.
RA2/AN2/VREF-	4	5	21	I/O	TTL	RA2 can also be analog input2 or negative analog reference voltage.
RA3/AN3/VREF+	5	6	22	I/O	TTL	RA3 can also be analog input3 or positive analog reference voltage.
RA4/T0CKI	6	7	23	I/O	ST	RA4 can also be the clock input to the Timer0 timer/ counter. Output is open drain type.
RA5/$\overline{SS}$/AN4	7	8	24	I/O	TTL	RA5 can also be analog input4 or the slave select for the synchronous serial port.
						PORTB is a bi-directional I/O port. PORTB can be soft-ware programmed for internal weak pull-up on all inputs.
RB0/INT	33	36	8	I/O	TTL/ST(1)	RB0 can also be the external interrupt pin.
RB1	34	37	9	I/O	TTL	
RB2	35	38	10	I/O	TTL	
RB3/PGM	36	39	11	I/O	TTL	RB3 can also be the low voltage programming input.
RB4	37	41	14	I/O	TTL	Interrupt-on-change pin.
RB5	38	42	15	I/O	TTL	Interrupt-on-change pin.
RB6/PGC	39	43	16	I/O	TTL/ST(2)	Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming clock.
RB7/PGD	40	44	17	I/O	TTL/ST(2)	Interrupt-on-change pin or In-Circuit Debugger pin. Serial programming data.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as an external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Pin Name	DIP Pin#	PLCC Pin#	QFP Pin#	I/O/P Type	Buffer Type	Description
						PORTC is a bi-directional I/O port.
RC0/T1OSO/T1CKI	15	16	32	I/O	ST	RC0 can also be the Timer1 oscillator output or a Timer1 clock input.
RC1/T1OSI/CCP2	16	18	35	I/O	ST	RC1 can also be the Timer1 oscillator input or Capture2 input/Compare2 output/PWM2 output.
RC2/CCP1	17	19	36	I/O	ST	RC2 can also be the Capture1 input/Compare1 output/PWM1 output.
RC3/SCK/SCL	18	20	37	I/O	ST	RC3 can also be the synchronous serial clock input/ output for both SPI and I^2C modes.
RC4/SDI/SDA	23	25	42	I/O	ST	RC4 can also be the SPI Data In (SPI mode) or data I/O (I^2C mode).
RC5/SDO	24	26	43	I/O	ST	RC5 can also be the SPI Data Out (SPI mode).
RC6/TX/CK	25	27	44	I/O	ST	RC6 can also be the USART Asynchronous Transmit or Synchronous Clock.
RC7/RX/DT	26	29	1	I/O	ST	RC7 can also be the USART Asynchronous Receive or Synchronous Data.
						PORTD is a bi-directional I/O port or parallel slave port when interfacing to a microprocessor bus.
RD0/PSP0	19	21	38	I/O	ST/TTL(3)	
RD1/PSP1	20	22	39	I/O	ST/TTL(3)	
RD2/PSP2	21	23	40	I/O	ST/TTL(3)	
RD3/PSP3	22	24	41	I/O	ST/TTL(3)	
RD4/PSP4	27	30	2	I/O	ST/TTL(3)	
RD5/PSP5	28	31	3	I/O	ST/TTL(3)	
RD6/PSP6	29	32	4	I/O	ST/TTL(3)	
RD7/PSP7	30	33	5	I/O	ST/TTL(3)	
						PORTE is a bi-directional I/O port.
RE0/$\overline{RD}$/AN5	8	9	25	I/O	ST/TTL(3)	RE0 can also be read control for the parallel slave port, or analog input5.
RE1/$\overline{WR}$/AN6	9	10	26	I/O	ST/TTL(3)	RE1 can also be write control for the parallel slave port, or analog input6.
RE2/$\overline{CS}$/AN7	10	11	27	I/O	ST/TTL(3)	RE2 can also be select control for the parallel slave port, or analog input7.
VSS	12,31	13,34	6,29	P	—	Ground reference for logic and I/O pins.
VDD	11,32	12,35	7,28	P	—	Positive supply for logic and I/O pins.
NC	—	1,17,28, 40	12,13, 33,34		—	These pins are not internally connected. These pins should be left unconnected.

Legend: I = input O = output I/O = input/output P = power
— = Not used TTL = TTL input ST = Schmitt Trigger input

Note 1: This buffer is a Schmitt Trigger input when configured as an external interrupt.
2: This buffer is a Schmitt Trigger input when used in Serial Programming mode.
3: This buffer is a Schmitt Trigger input when configured as general purpose I/O and a TTL input when used in the Parallel Slave Port mode (for interfacing to a microprocessor bus).
4: This buffer is a Schmitt Trigger input when configured in RC oscillator mode and a CMOS input otherwise.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก ค

Code program

ด้านส่ง (Transmitter)

```
#include <16f877.h>

#fuses HS,NOPUT,NOWDT,NOPROTECT

#use delay(clock=20000000)

#use rs232(baud=9600,xmit=pin_c6,rcv=pin_c7,stream=txdata1)

char receive;

#INT_RDA

void serial_isr()

{

   receive = fgetc(txdata1);

}

void main()

{

   enable_interrupts(INT_RDA);

   enable_interrupts(GLOBAL);

     while(true)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
{
  if(!input(pin_a0)==1&&!input(pin_a1)==1&&!input(pin_a2)==1&&!input(pin_a3)==1)
          // 1 1 1 1
   {
     fputc('A', txdata1);
      if(receive=='a')
     {
       output_high(pin_d0);
       output_high(pin_d1);
       output_high(pin_d2);
       output_high(pin_d3);
     }
  }
  else
if(!input(pin_a0)==0&&!input(pin_a1)==0&&!input(pin_a2)==0&&!input(pin_a3)==0)  //
0 0 0 0
  {
   fputc('B', txdata1);
      if(receive=='b')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
          output_low(pin_d0);
          output_low(pin_d1);
          output_low(pin_d2);
          output_low(pin_d3);
        }
      }
    else
if(!input(pin_a0)==0&&!input(pin_a1)==0&&!input(pin_a2)==0&&!input(pin_a3)==1)
            //0 0 0 1
    {
     fputc('C', txdata1);
       if(receive=='c')
      {
        output_low(pin_d0);
        output_low(pin_d1);
        output_low(pin_d2);
        output_high(pin_d3);
      }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }

   else
if(!input(pin_a0)==0&&!input(pin_a1)==0&&!input(pin_a2)==1&&!input(pin_a3)==0)
            //0 0 1 0

   {

    fputc('D', txdata1);

      if(receive=='d')

     {

      output_low(pin_d0);

      output_low(pin_d1);

      output_high(pin_d2);

      output_low(pin_d3);

     }

   }

   else
if(!input(pin_a0)==0&&!input(pin_a1)==0&&!input(pin_a2)==1&&!input(pin_a3)==1)
            //0 0 1 1

   {

    fputc('E', txdata1);

      if(receive=='e')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
          output_low(pin_d0);
          output_low(pin_d1);
          output_high(pin_d2);
          output_high(pin_d3);
        }
      }
    else
if(!input(pin_a0)==0&&!input(pin_a1)==1&&!input(pin_a2)==0&&!input(pin_a3)==0)
            // 0 1 0 0
    {
     fputc('F', txdata1);
       if(receive=='f')
      {
        output_low(pin_d0);
        output_high(pin_d1);
        output_low(pin_d2);
        output_low(pin_d3);
      }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
}

else
if(!input(pin_a0)==0&&!input(pin_a1)==1&&!input(pin_a2)==0&&!input(pin_a3)==1)
            //0 1 0 1

{

 fputc('G', txdata1);

   if(receive=='g')

  {

   output_low(pin_d0);

   output_high(pin_d1);

   output_low(pin_d2);

   output_high(pin_d3);

  }

}

else
if(!input(pin_a0)==0&&!input(pin_a1)==1&&!input(pin_a2)==1&&!input(pin_a3)==0)
            // 0 1 1 0

{

 fputc('H', txdata1);

   if(receive=='h')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
          output_low(pin_d0);
          output_high(pin_d1);
          output_high(pin_d2);
          output_low(pin_d3);
        }
      }
    else
if(!input(pin_a0)==0&&!input(pin_a1)==1&&!input(pin_a2)==1&&!input(pin_a3)==1)
          // 0 1 1 1
    {
     fputc('l', txdata1);
       if(receive=='i')
      {
        output_low(pin_d0);
        output_high(pin_d1);
        output_high(pin_d2);
        output_high(pin_d3);
      }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }

   else
if(!input(pin_a0)==1&&!input(pin_a1)==0&&!input(pin_a2)==0&&!input(pin_a3)==0)
            // 1 0 0 0

   {

    fputc('J', txdata1);

      if(receive=='j')

     {

      output_high(pin_d0);

      output_low(pin_d1);

      output_low(pin_d2);

      output_low(pin_d3);

     }

   }

   else
if(!input(pin_a0)==1&&!input(pin_a1)==0&&!input(pin_a2)==0&&!input(pin_a3)==1)
            // 1 0 0 1

   {

    fputc('K', txdata1);

      if(receive=='k')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
          output_high(pin_d0);
          output_low(pin_d1);
          output_low(pin_d2);
          output_high(pin_d3);
        }
      }
    else
if(!input(pin_a0)==1&&!input(pin_a1)==0&&!input(pin_a2)==1&&!input(pin_a3)==0)
            // 1 0 1 0
    {
     fputc('L', txdata1);
       if(receive=='l')
      {
        output_high(pin_d0);
        output_low(pin_d1);
        output_high(pin_d2);
        output_low(pin_d3);
      }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    }

   else
if(!input(pin_a0)==1&&!input(pin_a1)==0&&!input(pin_a2)==1&&!input(pin_a3)==1)
            // 1 0 1 1

   {

    fputc('M', txdata1);

      if(receive=='m')

     {

      output_high(pin_d0);

      output_low(pin_d1);

      output_high(pin_d2);

      output_high(pin_d3);

     }

   }

  else
if(!input(pin_a0)==1&&!input(pin_a1)==1&&!input(pin_a2)==0&&!input(pin_a3)==0)
            // 1 1 0 0

   {

    fputc('N', txdata1);

      if(receive=='n')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        {
          output_high(pin_d0);
          output_high(pin_d1);
          output_low(pin_d2);
          output_low(pin_d3);
        }
      }
    else
if(!input(pin_a0)==1&&!input(pin_a1)==1&&!input(pin_a2)==0&&!input(pin_a3)==1)
          // 1 1 0 1
    {
     fputc('O', txdata1);
       if(receive=='o')
      {
        output_high(pin_d0);
        output_high(pin_d1);
        output_low(pin_d2);
        output_high(pin_d3);
      }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      }

     else
if(!input(pin_a0)==1&&!input(pin_a1)==1&&!input(pin_a2)==1&&!input(pin_a3)==0)
              // 1 1 1 0

     {

      fputc('P', txdata1);

        if(receive=='p')

       {

        output_high(pin_d0);

        output_high(pin_d1);

        output_high(pin_d2);

        output_low(pin_d3);

       }

     }

   delay_ms(100);

  }

}
```

ด้านรับที่ 1 (1st Receiver)

```
#include <16f877.h>
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#fuses HS,NOPUT,NOWDT,NOPROTECT

#use delay(clock=20000000)

#use rs232(baud=9600,xmit=PIN_C6,rcv=PIN_C7,stream=PC)

char receive1;

#INT_RDA

void serial_isr()

{

   receive1 = fgetc(PC);

}

void main()

{

   enable_interrupts(INT_RDA);

   enable_interrupts(GLOBAL);



   while(true)

   {

     if(receive1=='A')

       {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    output_high(pin_d0);
    output_high(pin_d1);
    output_high(pin_d2);
    fputc('a',PC);
  }
 else if(receive1=='B')
  {
   output_low(pin_d0);
   output_low(pin_d1);
   output_low(pin_d2);
   fputc('b',PC);
  }
 else if(receive1=='C')
 {
   output_low(pin_d0);
   output_low(pin_d1);
   output_low(pin_d2);
   fputc('c',PC);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 }
else if(receive1=='D')
 {
  output_low(pin_d0);
  output_low(pin_d1);
  output_high(pin_d2);
   fputc('d',PC);
 }
else if(receive1=='E')
{
 output_low(pin_d0);
 output_low(pin_d1);
 output_high(pin_d2);
  fputc('e',PC);
}
else if(receive1=='F')
{
 output_low(pin_d0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    output_high(pin_d1);
    output_low(pin_d2);
     fputc('f',PC);
   }
  else if(receive1=='G')
   {
    output_low(pin_d0);
    output_high(pin_d1);
   output_low(pin_d2);
    fputc('g',PC);
  }
 else if(receive1=='H')
 {
  output_low(pin_d0);
  output_high(pin_d1);
  output_high(pin_d2);
   fputc('h',PC);
 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else if(receive1=='I')
 {
  output_low(pin_d0);
  output_high(pin_d1);
  output_high(pin_d2);
   fputc('i',PC);
 }
else if(receive1=='J')
 {
  output_high(pin_d0);
  output_low(pin_d1);
  output_low(pin_d2);
   fputc('j',PC);
 }
else if(receive1=='K')
 {
  output_high(pin_d0);
  output_low(pin_d1);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    output_low(pin_d2);
     fputc('k',PC);
   }
  else if(receive1=='L')
   {
    output_high(pin_d0);
    output_low(pin_d1);
    output_high(pin_d2);
     fputc('l',PC);
  }
 else if(receive1=='M')
 {
  output_high(pin_d0);
  output_low(pin_d1);
  output_high(pin_d2);
   fputc('m',PC);
 }
else if(receive1=='N')
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 {
  output_high(pin_d0);
  output_high(pin_d1);
  output_low(pin_d2);
   fputc('n',PC);
 }
else if(receive1=='O')
 {
 output_high(pin_d0);
 output_high(pin_d1);
 output_low(pin_d2);
  fputc('o',PC);
}
else if(receive1=='P')
 {
 output_high(pin_d0);
 output_high(pin_d1);
 output_high(pin_d2);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        fputc('p',PC);
      }
   delay_ms(100);
   }
}
```

ด้านรับที่ 2 (2nd Receiver)

```
#include <16f877.h>
#fuses HS,NOPUT,NOWDT,NOPROTECT
#use delay(clock=20000000)
#use rs232(baud=9600,xmit=PIN_C6,rcv=PIN_C7,stream=PC)
char receive1;
#INT_RDA
void serial_isr()
{
   receive1 = fgetc(PC);
}
void main()
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
enable_interrupts(INT_RDA);
enable_interrupts(GLOBAL);
while(true)
{
  if(receive1=='A')
   {
     output_high(pin_d0);
     fputc('a',PC);
   }
 else if(receive1=='B')
 {
   output_low(pin_d0);
    fputc('b',PC);
 }
 else if(receive1=='C')
 {
   output_high(pin_d0);
    fputc('c',PC);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
 }
else if(receive1=='D')
 {
  output_low(pin_d0);
   fputc('d',PC);
 }
else if(receive1=='E')
 {
  output_high(pin_d0);
   fputc('e',PC);
}
else if(receive1=='F')
 {
  output_low(pin_d0);
   fputc('f',PC);
 }
else if(receive1=='G')
 {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  output_high(pin_d0);
   fputc('g',PC);
 }
else if(receive1=='H')
 {
  output_low(pin_d0);
   fputc('h',PC);
 }
else if(receive1=='I')
{
 output_high(pin_d0);
  fputc('i',PC);
 }
else if(receive1=='J')
 {
 output_low(pin_d0);
  fputc('j',PC);
 }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
else if(receive1=='K')
 {
  output_high(pin_d0);
   fputc('k',PC);
 }
else if(receive1=='L')
 {
  output_low(pin_d0);
   fputc('l',PC);
 }
else if(receive1=='M')
 {
  output_high(pin_d0);
   fputc('m',PC);
 }
else if(receive1=='N')
 {
  output_low(pin_d0);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        fputc('n',PC);
       }
      else if(receive1=='O')
       {
        output_high(pin_d0);
         fputc('o',PC);
       }
      else if(receive1=='P')
      {
       output_low(pin_d0);
        fputc('p',PC);
      }
   delay_ms(100);
   }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้