

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

มาตรวัดน้ำไร้สาย

WIRELESS WATER METER



โดย
นางสาวเพ็ญพิชชา คำมา
นายประนอม นภัท
นายวิจิตร เรืองกาญจน์

รพ.
พ892ม
2549

เลขหมู่.....
เลขทะเบียน 72917
วัน,เดือน,ปี 25 ส.ย. 2550

b. 117246๗๔
i.

ปฏิญานิทรรศน์นี้เป็นส่วนหนึ่งของการศึกษาค้นคว้าหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2549

ผ่านการตรวจรูปเล่มแล้ว
เอกสารนี้เป็นเอกสารที่จัดทำขึ้นไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
(ลงชื่อ).....ผู้ตรวจ
ผ่านการตรวจชิ้นงานแล้ว
(ลงชื่อ).....ผู้ตรวจ

มาตรวัดน้ำไร้สาย
WIRELESS WATER METER

โดย

นางสาวเพ็ญพิชชา คำมา 47015016

นายประนอม นำทัพ 47015017

นายวิจิตร เรืองกาญจน์ 47015023

อาจารย์ที่ปรึกษา

ผศ.กริยงไกร วงศ์โรจนกรณ์

รศ.ดร.สุวิพล สาทิชีวะภาค

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมโทรคมนาคม

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญาโทปีการศึกษา 2549

ภาควิชาวิศวกรรมโทรคมนาคม

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง **มาตรวัดน้ำไร้สาย**

WIRELESS WATER METER

ผู้จัดทำ

1. นางสาวเพ็ญทิชชา คำมา 47015016
2. นายประนอม นำทัพ 47015017
3. นายวิจิตร เรืองกาญจน์ 47015023

.....
อาจารย์ที่ปรึกษา
(ศศ.เกรียงไกร วงศ์โรจนกรณ์)

.....
อาจารย์ที่ปรึกษา
(รศ.ดร.สุวิพล ลิทธิชีวภาค)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

มาตรวัดน้ำไร้สาย

WIRELESS WATER METER

โดย นางสาวเพ็ญพิชชา คำมา 47015016

นายประนอม นำทัพ 47015017

นายวิจิตร เรืองกาญจน์ 47015023

อาจารย์ที่ปรึกษา ศศ.เกรียงไกร วงศ์โรจนภรณ์

รศ.ดร.สุวิพล สิริชวีภาค

บทคัดย่อ

โครงการนี้เป็นการสร้างอุปกรณ์อ่านค่าตัวเลขของมาตรวัดน้ำโดยใช้ตัวตรวจจับสนามแม่เหล็ก (Hall Effect Sensor) เป็นตัวตรวจจับ ระบบจะทำการส่งค่าไปเก็บยังหน่วยความจำและแสดงผลทางหน้าจอคอมพิวเตอร์

ABSTRACT

This project builds the reading equipment for water meter .Working by Hall Effect Sensor, the rating point was saved in memory and show on the monitor.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

ปริญญานิพนธ์ฉบับนี้สำเร็จลงได้ด้วยความช่วยเหลือจากบุคคลหลายๆท่านทางผู้จัดทำขอกราบ
ขอบพระคุณอาจารย์ที่ปรึกษา ศศ.เกรียงไกร วงศ์โรจนภรณ์ และ รศ.ดร.สุวิพล ลิทธิชีวะภาค ที่ได้ให้
คำปรึกษา คำแนะนำ และให้ความรู้แก่ผู้จัดทำ

ขอขอบคุณ นายอภิชาติ กำลิ่งแพทย์ ที่คอยให้ความช่วยเหลือ ให้คำปรึกษา คำแนะนำในสิ่ง
ต่างๆแก่ผู้จัดทำ

ขอขอบคุณเพื่อนๆและน้องๆทุกคนที่ให้คำแนะนำ ให้ยืมอุปกรณ์ เอื้อเฟื้อสถานที่ คอยเป็น
กำลังใจและให้ความช่วยเหลือแก่ผู้จัดทำ

ที่ลืมนมิได้เลยขอกราบขอบพระคุณ คุณพ่อ คุณแม่ อาจารย์ และญาติพี่น้องทุกท่านที่ให้การ
สนับสนุน และเป็นกำลังใจให้แก่ผู้จัดทำมาโดยตลอด



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

	หน้า
บทที่ 1 บทนำ	
1.1 วัตถุประสงค์ของโครงการ	1
1.2 ขอบเขตของโครงการ	1
บทที่ 2 ทฤษฎีและหลักการ	
2.1 ทฤษฎีเกี่ยวกับมาตรวัดน้ำ	2
2.1.1 ความหมายของค่าที่ใช้ในมาตรวัดน้ำ	2
2.1.2 ขนาดของมาตรวัดน้ำ	3
2.1.3 เกลียว	3
2.1.4 วัสดุและการทำ	3
2.1.5 ส่วนประกอบ	5
2.1.6 คุณสมบัติ	6
2.1.7 การคำนวณค่าบริการ	6
2.1.8 การใช้งานมาตรวัดน้ำร่วมกับตัวตรวจจับสนามแม่เหล็ก	6
2.2 ทฤษฎีเครื่องวัดสนามแม่เหล็ก	7
2.2.1 ตัวตรวจจับสนามแม่เหล็ก (Hall Effect Sensor)	7
2.2.2 ปรากฏการณ์ฮอลล์ (Hall Effect)	9
2.3 การสื่อสารข้อมูลอนุกรม	11
2.3.1 ความเร็วของการสื่อสารข้อมูลอนุกรม	11
2.3.2 รูปแบบของการส่งข้อมูลอนุกรม	12
2.4 ไมโครคอนโทรลเลอร์ MCS-51	12
2.4.1 การใช้งานพอร์ตสื่อสารอนุกรม	12
2.4.2 กระบวนการรับและส่งข้อมูลอนุกรมของ MCS-51	13
2.4.3 รีจิสเตอร์ที่ใช้ควบคุมการรับส่งข้อมูลอนุกรม SCON	19
2.4.4 หน่วยความจำและการเชื่อมต่อกับ MCS – 51	19
2.4.5 ระบบบัสแบบ I ² C และการเชื่อมต่อ	22
2.4.5.1 ฟังก์ชันที่ช่วยในการเขียนโปรแกรมภาษา C เพื่อติดต่ออุปกรณ์บนระบบบัส I ² C ดังนี้	25
2.4.6 การสร้างฐานเวลา	26
2.5 โครงสร้างแอลซีดี โมดูล (LCD Module)	30
2.5.1 ส่วนประกอบหลักของแอลซีดี โมดูล	30
2.5.2 การเชื่อมต่อ แอลซีดี โมดูล เข้ากับไมโครคอนโทรลเลอร์	31
2.5.3 ชุดคำสั่งของ แอลซีดี โมดูล	31

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

	หน้า
2.5.4 ฟังก์ชันการเขียนโปรแกรมภาษา C เพื่อติดต่อกับ แอลซีดี โมดูล	34
2.6 การใช้งานคีย์แพด (Keypad)	35
2.6.1 การใช้ไอซีสแกนคีย์สวิตช์	36
2.7 มาตรฐานพอร์ทอนุกรมแบบ RS – 232	37
2.7.1 คอนเน็คเตอร์สำหรับพอร์ท RS – 232 และการเชื่อมต่อ	38
2.8 โมดูลความถี่วิทยุ 2.4 GHz	40
2.8.1 การจัดขาของโมดูลความถี่วิทยุ (TRW - 2.4 GHz)	42
2.8.2 โหมคการทำงานของ (TRW - 2.4 GHz)	42
2.8.3 ส่วนประกอบของชุดข้อมูล	44
2.8.4 ตำแหน่งบิตข้อมูลของโมดูลความถี่วิทยุ	47
บทที่ 3 การสร้างและการออกแบบ	
3.1 บล็อกไดอะแกรมการทำงานของโครงการระบบมาตรวัดน้ำไร้สายและคำอธิบาย	48
3.2 การสร้างและการออกแบบวงจร	50
3.2.1 วงจรเปรียบเทียบแรงดันและตัวตรวจจับสนามแม่เหล็ก	50
3.2.2 วงจรไมโครคอนโทรลเลอร์ในอุปกรณ์เก็บข้อมูล	51
3.2.3 วงจรไมโครคอนโทรลเลอร์ในมาตรวัดน้ำไร้สาย	52
3.3.4 วงจรพาวเวอร์ซัพพลาย (Power Supply)	55
3.3.5 วงจรแหล่งจ่ายไฟสำรอง (Battery Backup)	55
3.3 วิธีการใช้งานและตัวอย่างการใช้งานระบบมาตรวัดน้ำไร้สาย	56
3.4 ส่วนของซอฟต์แวร์	57
3.4.1 ส่วนของอุปกรณ์เก็บข้อมูล	57
3.4.2 ส่วนของมาตรวัดน้ำไร้สาย	64
3.4.3 ส่วนของคอมพิวเตอร์	66
บทที่ 4 การทดลองและผลการทดลอง	
4.1 อุปกรณ์เก็บข้อมูล	67
4.2 มาตรวัดน้ำไร้สาย	70
4.2.1 การเปรียบเทียบสัญญาณต่างๆของวงจรเปรียบเทียบแรงดัน	71
4.2.2 การเปรียบเทียบจำนวนครั้งที่แม่เหล็กตัดผ่านตัวตรวจจับสนามแม่เหล็ก จะเปรียบเทียบกับปริมาณน้ำ 0.01ลูกบาศก์เมตร	73
4.3 ส่วนประกอบของมาตรวัดน้ำ	74
4.4 การรับส่งข้อมูลแบบไร้สาย	75
4.5 การต่อใช้งานจริง	75

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.6 การแสดงผลทางหน้าจอคอมพิวเตอร์

บทที่ 5 บทวิจารณ์และสรุป

ภาคผนวก

กิตติกรรมประกาศ

หนังสืออ้างอิง



สารบัญรูป

	หน้า
รูปที่ 2.1 ตัวตรวจจับสนามแม่เหล็ก	7
รูปที่ 2.2 การต่อตัวตรวจจับสนามแม่เหล็กกับแหล่งจ่ายไฟกระแสตรงและ โวลต์มิเตอร์	8
รูปที่ 2.3 การวัดความเข้มของสนามแม่เหล็ก	8
รูปที่ 2.4 (ก - ค) แสดงการเกิดปรากฏการณ์ฮอลล์	9
รูปที่ 2.5 การวัดความต่างศักย์ฮอลล์ V_H	10
รูปที่ 2.6 ผังการทำงานโหมด 0	15
รูปที่ 2.7 ผังการทำงานโหมด 1	16
รูปที่ 2.8 ผังการทำงาน โหมด 2	17
รูปที่ 2.9 ผังการทำงาน โหมด 3	18
รูปที่ 2.10 แสดงกลุ่มสัญญาณที่ใช้ในการติดต่อกับหน่วยความจำ	20
รูปที่ 2.11 แสดงสัญญาณในการอ่านข้อมูล	20
รูปที่ 2.12 แสดงสัญญาณในการเขียนข้อมูล	21
รูปที่ 2.13 แสดงสัญญาณของ MCS - 51 ที่ใช้ติดต่อกับตำแหน่งของหน่วยความจำ	21
รูปที่ 2.14 แสดงการต่อหน่วยความจำประเภท ROM และ RAM กับ MCS - 51	22
รูปที่ 2.15 รูปแบบการเชื่อมต่อมาตรฐานของระบบบัส I ² C	22
รูปที่ 2.16 จังหวะสัญญาณเวลาบนบัส	23
รูปที่ 2.17 รูปแบบข้อมูลมาตรฐานในการอ้างแอดเดรสของการเชื่อมต่อบนระบบบัส I ² C	24
รูปที่ 2.18 รูปแบบของข้อมูลในการอ้างแอดเดรสแบบ 7 บิตบนระบบบัส I ² C	25
รูปที่ 2.19 รูปแบบของข้อมูลในการอ้างแอดเดรสแบบ 10 บิตบนระบบบัส I ² C	25
รูปที่ 2.20 บล็อกไดอะแกรมภายในของไอซี DS1307	27
รูปที่ 2.21 แผนผังหน่วยความจำ	28
รูปที่ 2.22 แผนผังรีจิสเตอร์ตำแหน่งต่างๆ ในการจัดการด้านฐานข้อมูลเวลา	28
รูปที่ 2.23 วงจรการเชื่อมต่อกับ MCS-51 อย่างง่าย	29
รูปที่ 2.24 โครงสร้างแอลซีดี โมดูล	30
รูปที่ 2.25 วงจรสวิตช์แบบแมทริกซ์หรือคีย์แพด	36
รูปที่ 2.26 แสดงไอซีสแกนคีย์สวิตช์	36
รูปที่ 2.27 แสดงการจัดขาของคอนเน็คเตอร์พอร์ทอนุกรมตามมาตรฐาน RS-232	38
รูปที่ 2.28 แสดงการต่ออุปกรณ์ภายนอกกับพอร์ทอนุกรมของคอมพิวเตอร์ในลักษณะต่าง ๆ	39
รูปที่ 2.29 แสดงลักษณะขาของ โมดูลความถี่วิทยุ (TRW - 2.4 GHz)	41
รูปที่ 2.30 แสดงรายละเอียดทางด้านบน, ด้านข้างและด้านหน้าของตัวโมดูลความถี่วิทยุ	41
รูปที่ 2.31 โฟลว์ชาร์ต Shock Burst Transmit ของโมดูลความถี่วิทยุ (TRW - 2.4 GHz)	45
รูปที่ 2.32 โฟลว์ชาร์ต Shock Burst Receive ของโมดูลความถี่วิทยุ (TRW - 2.4 GHz)	46

สารบัญรูป (ต่อ)

	หน้า	
รูปที่3.1	บล็อกไดอะแกรมการทำงานของมาตรวัดน้ำไร้สาย	48
รูปที่3.2	บล็อกไดอะแกรมการทำงานของอุปกรณ์เก็บข้อมูล	49
รูปที่3.3	วงจรเปรียบเทียบแรงดัน	50
รูปที่3.4	วงจรเปรียบเทียบแรงดันและตัวตรวจจับสนามแม่เหล็ก	51
รูปที่3.5	วงจรไมโครคอนโทรลเลอร์ในอุปกรณ์เก็บข้อมูล	51
รูปที่3.6	วงจรไมโครคอนโทรลเลอร์ในมาตรวัดน้ำไร้สาย	52
รูปที่3.7	วงจรรวมของอุปกรณ์เก็บข้อมูล	52
รูปที่3.8	วงจรรวมของมาตรวัดน้ำไร้สาย	53
รูปที่3.9	วงจรการต่อใช้งานจริง	54
รูปที่3.10	วงจรพาวเวอร์ซัพพลาย (Power Supply)	55
รูปที่3.11	วงจรแหล่งจ่ายไฟสำรอง	56
รูปที่3.12	โพลีชาร์ตโปรแกรมหลักของอุปกรณ์เก็บข้อมูล	58
รูปที่3.13	โพลีชาร์ตการตั้งเวลาและวันที่	59
รูปที่3.14	โพลีชาร์ตการอ่านค่าจากค่าปริมาณน้ำที่บันทึกไว้ในหน่วยความจำ	60
รูปที่3.15	โพลีชาร์ตการอ่านค่าปริมาณน้ำจากมาตรวัดน้ำไร้สาย	61
รูปที่3.16	โพลีชาร์ตการแก้ไขค่าปริมาณน้ำที่มาตรวัดน้ำไร้สาย	62
รูปที่3.17	โพลีชาร์ตการขัดจังหวะ (Interrupt) ที่ขา Rx ใช้ในการติดต่อไมโครคอนโทรลเลอร์กับคอมพิวเตอร์	63
รูปที่3.18	โพลีชาร์ตการวัดปริมาณน้ำโดยอาศัยการตัดผ่านของสนามแม่เหล็กที่ตัวตรวจจับสนามแม่เหล็ก	64
รูปที่3.19	โพลีชาร์ตการขัดจังหวะ (Interrupt) เพื่ออ่านหรือแก้ไขปริมาณน้ำที่มาตรวัดน้ำไร้สาย	65
รูปที่3.20	โพลีชาร์ตการแสดงผลปริมาณน้ำโดยรับข้อมูลจากอุปกรณ์เก็บข้อมูล	66
รูปที่4.1	เป็นการแสดงคำสั่งเพื่อเลือกการตั้งเวลาหรือการตั้งวันที่	67
รูปที่4.2	คำสั่งการตั้งเวลา	67
รูปที่4.3	แสดงผลของการตั้งเวลาผิด	67
รูปที่4.4	คำสั่งการตั้งวันที่	67
รูปที่4.5	แสดงผลของการตั้งวันที่ผิด	67
รูปที่4.6	แสดงเวลาและวันที่	68
รูปที่4.7	คำสั่งในการตั้งค่าตำแหน่งของมาตรวัดน้ำไร้สาย	68
รูปที่4.8	แสดงตำแหน่งของมาตรวัดน้ำไร้สายที่กำหนด	68
รูปที่4.9	แสดงตำแหน่งที่มาตรวัดน้ำไร้สาย	68
รูปที่4.10	คำสั่งการอ่านข้อมูลจากมาตรวัดน้ำไร้สายที่ระบุตำแหน่ง	69

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

	หน้า
รูปที่4.11 เป็นการแสดงผลเมื่อไม่ได้รับการตอบสนองของมาตรวัดน้ำไร้สายตำแหน่ง 07	69
รูปที่4.12 กำหนดตำแหน่งของมาตรวัดน้ำไร้สายที่ต้องการ	69
รูปที่4.13 เป็นการแสดงผลเมื่อได้รับการตอบสนองและปริมาณการใช้น้ำที่มาตรวัดน้ำไร้สายที่ระบุตำแหน่ง 00 ที่วัดได้	69
รูปที่4.14 คำสั่งในการอ่านข้อมูลจากหน่วยความจำ	69
รูปที่4.15 แสดงตำแหน่งของมาตรวัดน้ำไร้สายและปริมาณน้ำใช้ไป	69
รูปที่4.16 คำสั่งในการแก้ไขค่าที่มาตรวัดน้ำไร้สาย	70
รูปที่4.17 แสดงการระบุตัวเลข	70
รูปที่4.18 แสดงค่าตัวเลขที่แก้ไขที่มาตรวัดน้ำไร้สาย	70
รูปที่4.19 แสดงการทำงานเสร็จสิ้นสมบูรณ์	70
รูปที่4.20 เป็นรูปสัญญาณที่อุปกรณ์เก็บข้อมูลเรียกมาตรวัดน้ำไร้สายที่ต้องการติดต่อ โดยเรียกตามรหัสเครื่อง	70
รูปที่4.21 เป็นรูปสัญญาณที่มาตรวัดน้ำไร้สายตามรหัสที่อุปกรณ์เก็บข้อมูลส่งสัญญาณกลับมา	71
รูปที่4.22 รูปสัญญาณเอาต์พุตของตัวตรวจจับสนามแม่เหล็กขณะที่ไม่มีสนามแม่เหล็กมาตัดผ่านตัวตรวจจับสนามแม่เหล็ก	71
รูปที่4.23 รูปสัญญาณเอาต์พุตของตัวตรวจจับสนามแม่เหล็ก ขณะที่ไม่มีสนามแม่เหล็กมาตัดผ่านตัวตรวจจับสนามแม่เหล็ก	72
รูปที่4.24 รูปสัญญาณเอาต์พุตของฮอปแอมป์ เมื่อแรงดันที่ขา 2 (เอาต์พุตของตัวตรวจจับสนามแม่เหล็ก) มากกว่าแรงดันขา 3 (แรงดันของความต้านทานปรับค่าได้)	72
รูปที่4.25 รูปสัญญาณเอาต์พุตของฮอปแอมป์ เมื่อแรงดันที่ขา 2 (เอาต์พุตของตัวตรวจจับสนามแม่เหล็ก) น้อยกว่าแรงดันขา 3 (แรงดันของความต้านทานปรับค่าได้)	73
รูปที่4.26 ส่วนประกอบของมาตรวัดน้ำ	74
รูปที่4.27 สัญญาณสเปกตรัมของตัวโมดูเลต-ส่ง (TRW – 2.4 GHz)	75
รูปที่4.28 อุปกรณ์เก็บข้อมูล	76
รูปที่4.29 มาตรวัดน้ำไร้สาย	76
รูปที่4.30 สายนำสัญญาณ	77
รูปที่4.31 การติดตั้ง Hall Effect Sensor เข้ากับมาตรวัดน้ำจริง	77
รูปที่4.32 การต่อใช้งานมาตรวัดน้ำไร้สายกับมาตรวัดน้ำจริง	78
รูปที่4.33 การแสดงผลหน้าจอหลัก	78
รูปที่4.34 การ Setup	79
รูปที่4.35 แสดงการระบุตำแหน่งของมาตรวัดน้ำไร้สาย	79
รูปที่4.36 แสดงค่าตำแหน่งของมาตรวัดน้ำไร้สาย	80

สารบัญรูป (ต่อ)

	หน้า
รูปที่4.37 ค่าต่างๆของมาตรวัดน้ำไร้สายที่ระบุตำแหน่งโดยอ่านจากอุปกรณ์เก็บข้อมูล	80
รูปที่4.38 การเคลียร์ (Clear) ข้อมูล	81



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

	หน้า
ตารางที่ 2.1 ชื่อขนาดมิติและเกณฑ์ความคลาดเคลื่อนของมาตรวัดน้ำ	3
ตารางที่ 2.2 เกลียว	3
ตารางที่ 2.3 วัสดุที่ใช้ทำส่วนประกอบหลักของมาตรวัดน้ำ	4
ตารางที่ 2.4 ส่วนประกอบทางเคมีของทองบรอนซ์ ทองเหลืองแท่งตีขึ้นรูปและเหล็กกล้าไร้สนิม	4
ตารางที่ 2.5 สมบัติทางกลของทองบรอนซ์และทองเหลืองแท่งตีขึ้นรูป	4
ตารางที่ 2.6 อัตราการไหลและความดันสูญเสีย	5
ตารางที่ 2.7 ตัวเลขเริ่มต้นและตัวเลขจำนวนเต็มที่บ้านทิกได้สูงสุด	5
ตารางที่ 2.8 โหมดการทำงานของพอร์ทสื่อสารอนุกรม	12
ตารางที่ 2.9 หน้าทีของขาใช้งาน แอลซีดี โมดูล	31
ตารางที่ 2.10 คำสั่งเคลียร์ตัวแสดงผล	31
ตารางที่ 2.11 คำสั่ง CURSOR AT HOME	32
ตารางที่ 2.12 คำสั่ง โหมดในการป้อนข้อมูล	32
ตารางที่ 2.13 คำสั่งควบคุมการแสดงผล	32
ตารางที่ 2.14 คำสั่งควบคุมการเลื่อนเคอร์เซอร์และตัวอักษร	33
ตารางที่ 2.15 การกำหนดบิต S/C และ R/L	33
ตารางที่ 2.16 คำสั่งการกำหนดฟังก์ชันการทำงาน	33
ตารางที่ 2.17 แสดงลักษณะการทำงานของขาของตัวโมดูลความถี่วิทยุ	42
ตารางที่ 2.18 การเชคค่าของแต่ละขาในโหมดต่างๆ	42
ตารางที่ 2.19 รายละเอียดทางไฟฟ้า	43
ตารางที่ 2.20 แสดงรายละเอียดของตำแหน่งบิตภายในตัว โมดูลความถี่วิทยุ	47
ตารางที่ 4.1 ตัวอย่างการเก็บผล ปริมาณน้ำตั้งแต่ 0.00 - 0.01 และจำนวนที่แม่เหล็กตัดผ่านตัวตรวจจับ สนามแม่เหล็ก	73

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

โลกเราทุกวันนี้มีน้ำนับเป็นปัจจัย 4 ในการดำเนินชีวิตของคนเราและกรรมวิธีในการจ่ายน้ำและคิดค่าน้ำยังต้องใช้บุคคลจำนวนมาก ในการเดินตรวจสอบหรือตรวจเช็คระดับปริมาณน้ำที่ใช้ไปจากมาตรวัดน้ำ เช่น บ้านพักอาศัย คอนโดฯ อพาร์ทเมนต์ และหอพัก ในปัจจุบันมีปัญหาในเรื่องของการคิดค่าน้ำผิดพลาด ทำให้เกิดความไม่พอใจแก่ผู้บริโภค

โครงการนี้จึงได้มีแนวคิดที่จะพัฒนาระบบมาตรวัดน้ำไร้สายขึ้นมาใช้งาน โดยมีตัวตรวจจับสนามแม่เหล็ก (Hall Effect Sensor) โดยอาศัยหลักการของปรากฏการณ์ฮอลล์ (Hall Effect) และการจัดเก็บข้อมูลของตัวมาตรวัดน้ำโดยอาศัยการประมวลผลของไมโครคอนโทรลเลอร์ และนำไปเก็บยังหน่วยความจำและแสดงผลทางหน้าจอคอมพิวเตอร์

1.1 วัตถุประสงค์ของโครงการ

- เพื่อที่จะศึกษาระบบและการทำงานของมาตรวัดน้ำ
- เพื่อที่จะศึกษาหลักการการทำงานของตัวตรวจจับสนามแม่เหล็ก
- เพื่อศึกษาการประยุกต์ใช้งาน ไมโครคอนโทรลเลอร์
- เพื่อสร้างระบบมาตรวัดน้ำไร้สาย
- เพื่อความสะดวกสบายและรวดเร็ว
- เพื่อศึกษาหลักการส่ง - รับ ข้อมูลแบบไร้สาย

1.2 ขอบเขตของโครงการ

จากการที่ได้ศึกษาและค้นคว้าเกี่ยวกับการทำงานของระบบมาตรวัดน้ำ และทำการสร้างระบบมาตรวัดน้ำไร้สายขึ้นมา โดยส่วนของข้อมูลที่ได้มาจากมาตรวัดน้ำ จะใช้ตัวตรวจจับสนามแม่เหล็ก ซึ่งเป็นตัวตรวจจับ (Sensor) ที่ใช้ในการตรวจจับสนามแม่เหล็ก โดยการติดแม่เหล็กขนาดเล็กลงบนใบพัดที่อยู่ในมาตรวัดน้ำ ในขณะที่น้ำไหลจะทำให้ใบพัดหมุนตัดผ่านตัวตรวจจับสนามแม่เหล็ก ซึ่งติดไว้ด้านข้างของมาตรวัดน้ำ ทำให้เกิดแรงดันเพิ่มขึ้นและลดลงเมื่อไม่มีแม่เหล็กมาตัดผ่านสลับกันไปเรื่อยๆ ทำให้ได้สัญญาณพัลส์ออกมาเพื่อเป็นสัญญาณให้ไมโครคอนโทรลเลอร์ตรวจสอบจำนวนรอบได้แล้วนำมาแสดงผลบนจอแสดงผล (LCD Module) ซึ่งในการแสดงผลนี้จะทำการเทียบกับหน้าปัดของมาตรวัดน้ำโดยใช้ทศนิยม 3 ตำแหน่ง นอกจากนี้มาตรวัดน้ำแต่ละตัวจะกำหนดตำแหน่งของมาตรวัดน้ำไร้สายไว้เพื่อป้องกันการส่งข้อมูลผิดพลาด ถ้าระบุตำแหน่งของมาตรวัดน้ำไร้สายผิดก็จะไม่ทำการส่งข้อมูลกลับมายังตัวรับ และในส่วนของภาครับสัญญาณและส่งสัญญาณไปยังตัวมาตรวัดน้ำนั้น

บทที่ 2 ทฤษฎีและหลักการ

2.1 ทฤษฎีเกี่ยวกับมาตรวัดน้ำ

มาตรวัดน้ำที่ใช้อยู่ในปัจจุบันมีอยู่ด้วยกันมากมายหลายชนิด แต่ละชนิดก็จะทำมาจากวัสดุที่ต่างชนิดกัน วัสดุที่เป็นส่วนประกอบหลัก โดยมีส่วนประกอบทางเคมีของทองบรอนซ์ ทองเหลืองแท่งตีขึ้นรูป และเหล็กกล้าไร้สนิม

2.1.1 ความหมายของค่าที่ใช้ในมาตรวัดน้ำ

อัตราการไหล หมายถึง ปริมาตรน้ำที่ไหลผ่านมาตรวัดน้ำในหนึ่งหน่วยเวลา

อัตราการไหลระบุ หมายถึง อัตราการไหลที่ออกแบบให้มาตรวัดน้ำทำงานได้อย่างต่อเนื่อง โดยไม่เกิดความเสียหาย ภายใต้การใช้งานปกติ มีค่าเป็น 0.5 เท่าของอัตราการไหลสูงสุด

อัตราการไหลสูงสุด หมายถึง อัตราการไหลมากที่สุดที่มาตรวัดน้ำยังทำงานได้ดีในระยะเวลาสั้นๆ โดยไม่เกิดความเสียหาย

อัตราการไหลต่ำสุด หมายถึง อัตราการไหลที่น้อยที่สุดที่มาตรวัดน้ำสามารถวัดปริมาณน้ำได้โดยมีเกณฑ์ความคลาดเคลื่อน \pm ร้อยละ 5 มีค่าเป็น 0.01 เท่าของอัตราการไหลระบุ

อัตราการไหลเปลี่ยนแปลง หมายถึง อัตราการไหลเมื่อเกณฑ์ความคลาดเคลื่อนในการวัดปริมาณน้ำไหลผ่านมาตรวัดน้ำเริ่มเปลี่ยนจากช่วงข้างต่ำไปเป็นช่วงข้างสูง มีค่าเป็น 0.015 เท่าของอัตราการไหลระบุ

พิสัยอัตราการไหล หมายถึง ช่วงระหว่างอัตราการไหลสูงสุดกับอัตราการไหลต่ำสุด ประกอบด้วยช่วงข้างต่ำและข้างสูง แยกกันที่อัตราการไหลเปลี่ยนแปลง

ช่วงข้างต่ำ หมายถึง ช่วงอัตราการไหลต่ำกว่าอัตราการไหลเปลี่ยนแปลง มีเกณฑ์ความคลาดเคลื่อนในการวัดปริมาณน้ำไหลผ่านมาตรวัดน้ำ \pm ร้อยละ 5

ช่วงข้างสูง หมายถึง ช่วงอัตราการไหลสูงกว่าอัตราการไหลเปลี่ยนแปลง มีเกณฑ์ความคลาดเคลื่อนในการวัดปริมาณน้ำไหลผ่านมาตรวัดน้ำ \pm ร้อยละ 2

2.1.2 ขนาดของมาตรวัดน้ำ

จะมีขนาดที่แตกต่างกันออกไปตามตารางที่ 2.1

ตารางที่ 2.1 ชื่อขนาดมิติและเกณฑ์ความคลาดเคลื่อนของมาตรวัดน้ำ

ชื่อขนาด	มิติของตัวเรือนหลัก					ความยาวระหว่างปลายของข้อต่อ ทั้ง 2 ข้าง (L4) ±2.5(mm)	
	H1 สูงสุด	H2 สูงสุด	L1		L2 สูงสุด		L3 สูงสุด
			ที่นิยมใช้	อื่นๆ			
15 (1/2)	60	220	170	110 130	65	65	248
20 (3/4)	60	240	190	165 -	65	65	311
25 (1)	65	260	260	- -	85	85	381
40 (1 1/2)	80	300	300	- -	105	105	-

2.1.3 เกลียว

เกลียวที่ใช้จะใช้ตามขนาดของมาตรวัดน้ำดังตามตารางที่ 2.2

ตารางที่ 2.2 เกลียว

ชื่อขนาด	เกลียวที่มาตรวัดน้ำ		เกลียวที่ปลายของข้อต่อ
	ชื่อขนาดของ เกลียวท่อ ตาม ISO 228/1	ความยาว เกลียวต่ำสุด มิลลิเมตร	ขนาดระบุของเกลียวท่อแบบ 55 ตาม มอก.281
15 (1/2)	G ¾ B	10	15
20 (3/4)	G 1 B	12	20
25 (1)	G 1 ¼ B	12	25
40 (1 1/2)	G 2 B	13	40

2.1.4 วัสดุและการทำ

1. วัสดุ

วัสดุที่ใช้ทำส่วนประกอบหลักของมาตรวัดน้ำให้เป็นไปตามตารางที่ 2.3 โดยมีส่วนประกอบทางเคมีของทองบรอนซ์ ทองเหลืองแท่งตีขึ้นรูปและเหล็กกล้าไร้สนิมเป็นไปตามตารางที่ 2.4 และสมบัติทางกลของทองบรอนซ์และทองเหลืองแท่งตีขึ้นรูปเป็นไปตามตารางที่ 2.5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 วัสดุที่ใช้ทำส่วนประกอบหลักของมาตรวัดน้ำ

ส่วนประกอบหลัก	วัสดุ
ตัวเรือนหลัก	ทองบรอนซ์ หรือทองเหลืองแท่งตีขึ้นรูป
ตัวกรองน้ำ	วัสดุสังเคราะห์
ลูกสูบ	วัสดุสังเคราะห์
เสื้อสูบ	วัสดุสังเคราะห์
แกนในของลูกสูบ	วัสดุสังเคราะห์

ตารางที่ 2.4 ส่วนประกอบทางเคมีของทองบรอนซ์ ทองเหลืองแท่งตีขึ้นรูป และเหล็กกล้าไร้สนิม

วัสดุ	ส่วนประกอบทางเคมี (ร้อยละ)				
	ดีบุก	ตะกั่ว	สังกะสี	ทองแดง	เหล็ก
ทองบรอนซ์	4.0 ถึง 6.0	4.0 ถึง 6.0	4.0 ถึง 7.0	82 ถึง 87	-
ทองเหลืองแท่งตีขึ้นรูป	-	1.0 ถึง 3.5	ส่วนที่เหลือ	56 ถึง 60	ไม่เกิน 0.3
เหล็กกล้าไร้สนิม	ส่วนประกอบทางเคมี (ร้อยละ)				
	ซิลิคอน	แมงกานีส	โครเมียม	นิกเกิล	
	ไม่เกิน 2.0	ไม่เกิน 2.0	17.0 ถึง 21	8.0 ถึง 12	

ตารางที่ 2.5 สมบัติทางกลของทองบรอนซ์และทองเหลืองแท่งตีขึ้นรูป

วัสดุ	ความต้านทานแรงดึง ต่ำสุด	ความต้านทานแรงดึง ที่จุดคราก ต่ำสุด	ความยืด ต่ำสุด (ร้อยละ)
	เมกะพาสคัล	เมกะพาสคัล	
ทองบรอนซ์	210	95	15
ทองเหลืองแท่งตีขึ้นรูป	340	-	12

2. การทำ

มาตรวัดน้ำต้องออกแบบให้มีอัตราการไหล และความดันสูญเสียเป็นไปตามตารางที่ 2.6

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.6 อัตราการไหลและความดันสูญเสีย

ชื่อขนาด	อัตราการไหลระบุลูกบาศก์เมตรต่อชั่วโมง	อัตราการไหลสูงสุดลูกบาศก์เมตรต่อชั่วโมง	พิสัยอัตราการไหลลูกบาศก์เมตรต่อชั่วโมง		ความดันสูญเสียที่อัตราการไหลสูงสุดไม่เกินเมกะพาสคัล
			ช่วงข้างต่ำ	ช่วงข้างสูง	
15 (1/2)	1.5	3.0	0.015 ถึง 0.022 5	0.0225 ถึง 3.0	0.10
20 (3/4)	2.5	5.0	0.025 ถึง 0.0375	0.0375 ถึง 5.0	
25 (1)	3.5	7.0	0.035 ถึง 0.0525	0.0525 ถึง 7.0	
40 (1 1/2)	10.0	20.0	0.100 ถึง 0.1500	0.1500 ถึง 20.0	

2.1.5 ส่วนประกอบ

1. ตัวเรือนหลัก

ตัวเรือนหลักเป็นตัวยุติที่สามารถแยกจากส่วนประกอบอื่นได้ แนวแกนของทางน้ำเข้าและทางน้ำออกต้องอยู่บนแกนเดียวกัน และปลายของทางน้ำเข้าและทางน้ำออกจะตั้งฉากกับแนวนอน

2. ตัวกรองน้ำ

มาตรวัดน้ำต้องติดตั้งตัวกรองน้ำไว้ที่ทางน้ำเข้า ตัวกรองน้ำต้องถอดออกเพื่อทำความสะอาดได้ง่ายและไม่เสียหายเมื่อถอดออกตามปกติ

3. เครื่องบันทึกปริมาตรน้ำ

เครื่องบันทึกปริมาตรน้ำจะต้องเป็นชนิดหน้าปิดแห้งหรือชนิดผนึกแน่น เพื่อป้องกันการเกิดฝ้าและสะดวกในการอ่านค่าตัวเลขที่บอกปริมาตรจำนวนเต็มของลูกบาศก์เมตร

ตัวเลขที่บอกปริมาตรจำนวนเต็มของลูกบาศก์เมตร ให้ใช้ตัวเลขสีค่าบนพื้นขาวหรือตัวเลขสีขาวบนพื้นดำ เศษของลูกบาศก์เมตรแสดงด้วยสีเพื่อให้เห็นแตกต่างอย่างเด่นชัดจากจำนวนเต็มของลูกบาศก์เมตร ความละเอียดของตัวเลขเริ่มต้นและตัวเลขจำนวนเต็มที่บันทึกได้สูงสุดให้เป็นไปตามตารางที่ 2.7

ตารางที่ 2.7 ตัวเลขเริ่มต้นและตัวเลขจำนวนเต็มที่บันทึกได้สูงสุด

ชื่อขนาด	ตัวเลขเริ่มต้น ละเอียด ไม่น้อยกว่า	ตัวเลขจำนวนเต็มที่บันทึกได้สูงสุด ไม่น้อยกว่า (ลูกบาศก์เมตร)
15 (1/2)	0.01	9.999
20 (3/4)	0.01	9.999

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.7 ตัวเลขเริ่มต้นและตัวเลขจำนวนเต็มที่บ้านทักได้สูงสุด (ต่อ)

25 (1)	0.01	9.999
40 (1 ¹ / 2)	0.10	99.999

2.1.6 คุณสมบัติ

1. ความทนความดัน

มาตรวัดน้ำต้องทนความดันน้ำ 1.72 เมกะพาสคัลได้เป็นเวลาไม่น้อยกว่า 1 นาที โดยไม่รั่วซึม

2. ความดันสูญเสียเนื่องจากมาตรวัดน้ำที่อัตราการไหลสูงสุด

ความดันสูญเสียที่อัตราการไหลสูงสุดต้องไม่เกิน 0.10 เมกะพาสคัล

3. ความเที่ยง

ปริมาณน้ำที่วัดได้สำหรับอัตราการไหลแต่ละค่าจะคลาดเคลื่อนได้ \pm ร้อยละ 5 สำหรับข้างต่ำ และ \pm ร้อยละ 2 สำหรับข้างสูง

4. การเร่งความสึกหรอ

ปริมาณน้ำที่วัดได้สำหรับอัตราการไหลแต่ละค่าจะคลาดเคลื่อนได้ \pm ร้อยละ 6 สำหรับข้างต่ำ และ \pm ร้อยละ 1.5 สำหรับข้างสูง

5. การป้องกันสนิม

มาตรวัดน้ำต้องมีอุปกรณ์ที่สามารถป้องกันสนามแม่เหล็กทุกด้าน ปริมาณน้ำที่วัดได้สำหรับอัตราการไหลแต่ละค่าจะคลาดเคลื่อนได้ ปริมาณน้ำที่วัดได้สำหรับอัตราการไหลแต่ละค่าจะคลาดเคลื่อนได้ \pm ร้อยละ 5 สำหรับข้างต่ำและ \pm ร้อยละ 2

2.1.7 การคำนวณค่าบริการ

วิธีคำนวณ

ปริมาณน้ำที่วัดจากมาตรวัดน้ำ ลูกบาศก์เมตร = ค่าสุดท้าย - ค่าเริ่มต้น

ปริมาณน้ำคำนวณได้จากสูตร

แบบต่อเนื่อง

ปริมาณน้ำที่คำนวณได้ ลูกบาศก์เมตร = อัตราการไหลที่ใช้ ลูกบาศก์เมตรต่อชั่วโมง \times ระยะเวลา ชั่วโมง

แบบไม่ต่อเนื่อง

ปริมาณน้ำที่คำนวณได้ ลูกบาศก์เมตร = 0.5 อัตราการไหลที่ใช้ ลูกบาศก์เมตรต่อชั่วโมง \times ระยะเวลา ชั่วโมง

2.1.8 การใช้งานมาตรวัดน้ำร่วมกับตัวตรวจจับสนามแม่เหล็ก

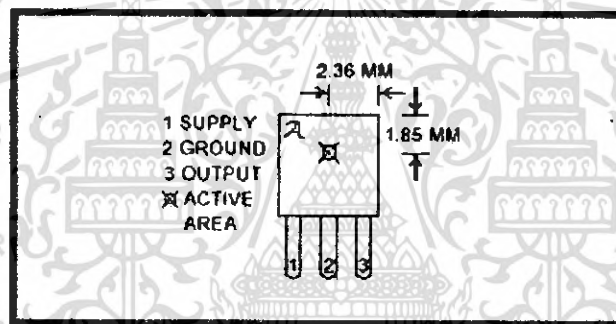
การนำตัวตรวจจับสนามแม่เหล็กมาใช้กับมาตรวัดน้ำ โดยการนำแผ่นแม่เหล็กขนาดเล็กมาติดกับไบพัดของมาตรวัดน้ำและติดตัวตรวจจับสนามแม่เหล็ก ไว้ด้านข้างของมาตรวัดน้ำ เวลาน้ำไหลไบพัดจะเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมุนทำให้แม่เหล็กที่ติดกับใบพัดหมุนมาติดกับตัวตรวจจับสนามแม่เหล็ก แล้วเกิดเป็นพัลส์ออกมา แล้วนำแรงดันที่ได้ไปประยุกต์ใช้กับงานอื่นต่อไป

2.2 ทฤษฎีเครื่องวัดสนามแม่เหล็ก

แม่เหล็ก (Magnet) เป็นสิ่งที่สามารถดูดวัสดุบางชนิดได้ เช่น เหล็ก นิกเกิล โคบอลต์ เป็นต้น การที่แม่เหล็กดูดสารบางอย่างได้ เนื่องจากมีสนามแม่เหล็ก (Magnetic Field) ในบริเวณโดยรอบแม่เหล็ก เราสามารถตรวจสอบว่าบริเวณใดมีสนามแม่เหล็กหรือไม่ โดยใช้เข็มทิศ แต่เราไม่สามารถทราบได้ว่ามีค่าเท่าใด นักวิทยาศาสตร์พยายามวัดสนามแม่เหล็กด้วยวิธีการต่างๆ แต่ในปัจจุบันเราสามารถวัดสนามแม่เหล็กได้สะดวกและรวดเร็วโดยใช้ตัวตรวจจับสนามแม่เหล็ก ซึ่งทำงานโดยอาศัยหลักการของปรากฏการณ์ฮอลล์ ตัวตรวจจับสนามแม่เหล็กเป็นวงจรรวมที่ทำให้เกิดความต่างศักย์ซึ่งเป็นสัดส่วนตรงกับความเข้มของสนามแม่เหล็กที่ผ่านในแนวตั้ง เมื่อนำตัวตรวจจับสนามแม่เหล็กไปต่อกับโวลต์มิเตอร์แล้วนำไปวางใกล้บริเวณที่มีสนามแม่เหล็กก็จะทำให้ทราบค่าความเข้มของสนามแม่เหล็กได้

2.2.1 ตัวตรวจจับสนามแม่เหล็ก (Hall Effect Sensor)



รูปที่ 2.1 ตัวตรวจจับสนามแม่เหล็ก

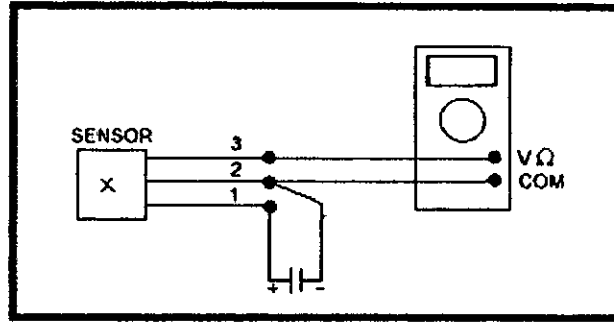
ตัวตรวจจับสนามแม่เหล็กเป็นวงจรรวม มีขนาดและลักษณะดังรูปที่ 2.1 และมีสมบัติดังนี้

Input Voltage 4.5 - 6V

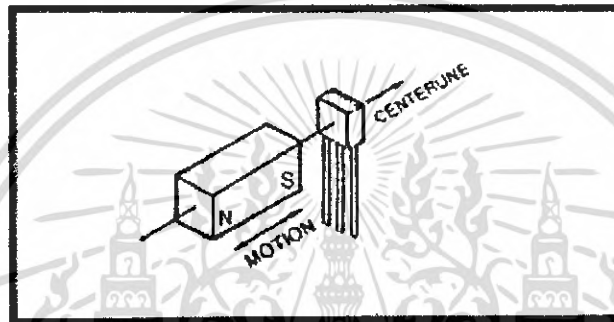
Offset Voltage 2.5 V (ประมาณ)

Sensitivity 13 V/T

เมื่อต่อแหล่งจ่ายไฟกระแสตรงหรือเซลล์ไฟฟ้า 4.5V - 6V เข้ากับขา 1 และขา 2 และต่อ โวลต์มิเตอร์เข้ากับขา 2 และขา 3 ดังรูปที่ 2.2 โวลต์มิเตอร์จะแสดงค่าประมาณ 2.5 V ค่านี้เป็นความต่างศักย์ขณะที่ไม่มีสนามแม่เหล็ก เรียกว่า Offset Voltage ค่านี้อาจเปลี่ยนแปลงได้เล็กน้อยขึ้นอยู่กับโวลเตจของแหล่งจ่ายไฟกระแสตรงที่ต่อกับขา 1 และขา 2 แต่จะมีค่าประมาณครึ่งหนึ่งของ โวลเตจของแหล่งจ่ายไฟกระแสตรง



รูปที่ 2.2 การต่อตัวตรวจจับสนามแม่เหล็กกับแหล่งจ่ายไฟกระแสตรงและโวลต์มิเตอร์



รูปที่ 2.3 การวัดความเข้มของสนามแม่เหล็ก

เมื่อนำแม่เหล็กเข้าใกล้ Active Area ของตัวตรวจจับสนามแม่เหล็ก ความต่างศักย์จะมีค่าเพิ่มขึ้นหรือลดลงขึ้นอยู่กับทิศของสนามแม่เหล็ก กล่าวคือถ้านำขั้วได้เข้าใกล้ ความต่างศักย์จะมีค่าเพิ่มขึ้น แต่ถ้านำขั้วเหนือเข้าใกล้ ความต่างศักย์จะมีค่าลดลง ความต่างศักย์ที่เปลี่ยนไปมีความสัมพันธ์กับความเข้มของสนามแม่เหล็กหรือความหนาแน่นฟลักซ์แม่เหล็ก (Magnetic Flux Density) ดังนี้

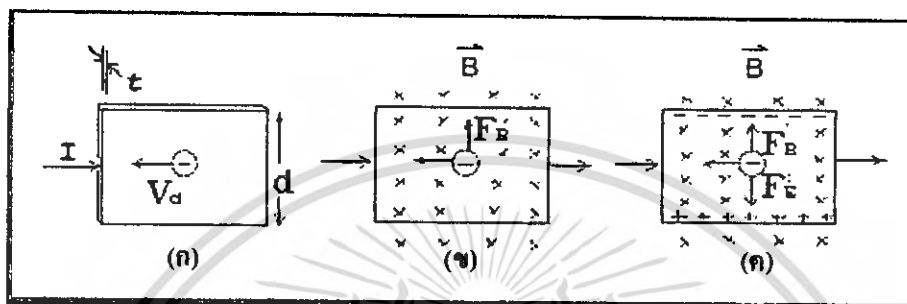
$$B = (V_{\text{out}(B)} - V_{\text{out}(O)}) S^{-1}$$

- เมื่อ
- $V_{\text{out}(O)}$ เป็นความต่างศักย์ขณะไม่มีสนามแม่เหล็ก
 - $V_{\text{out}(B)}$ เป็นความต่างศักย์ขณะมีสนามแม่เหล็ก
 - S เป็นสัมประสิทธิ์ความไว มีหน่วยเป็น โวลต์ต่อเทสลา (V/T)
สำหรับตัวตรวจจับสนามแม่เหล็กที่ใช้ในบทความนี้ $S = 13 \text{ V/T}$
 - B เป็นความเข้มของสนามแม่เหล็ก หรือความหนาแน่นฟลักซ์แม่เหล็ก
มีหน่วยเป็นเทสลา (T)

ตัวตรวจจับสนามแม่เหล็กสามารถวัดสนามแม่เหล็กในบริเวณใกล้แม่เหล็กถาวร สนามแม่เหล็กที่เกิดขึ้นบริเวณปลายโซเลนอยด์และสนามแม่เหล็กใกล้เส้นลวดตัวนำที่มีกระแสไฟฟ้าผ่านได้

2.2.2 ปฏิกิริยาฮอลล์ (Hall Effect)

ใน ค.ศ. 1879 เอ็ดวิน ฮอลล์ (Edwin Hall) นักศึกษามหาวิทยาลัยจอห์น ฮอปกินส์ ซึ่งในขณะนั้นเอ็ดวิน ฮอลล์มีอายุ 24 ปีได้พบว่าเมื่อนำแผ่นตัวนำบางที่มีกระแสไฟฟ้าผ่านไปวางไว้ในบริเวณที่มีสนามแม่เหล็ก ทาหะประจุ (Charge Carriers) ในตัวนำสามารถเบนไปจากแนวทางเดิมได้ และการเบนนี้มีผลทำให้เกิดสนามไฟฟ้าในตัวนำบางในทิศตั้งฉากกับทั้งกระแสไฟฟ้าและสนามแม่เหล็ก การค้นพบนี้เรียกว่า ปฏิกิริยาฮอลล์



รูปที่ 2.4 (ก - ค) แสดงการเกิดปรากฏการณ์ฮอลล์

การเกิดปรากฏการณ์ฮอลล์อธิบายได้โดยใช้รูปที่ 2.4 (ก - ค) ดังนี้

รูปที่ 2.4 (ก) แสดงแผ่นตัวนำบางที่มีความกว้าง d หนา t และมีกระแสไฟฟ้า (Conventional-Current) I ผ่านในทิศจากด้านซ้ายไปด้านขวาพาหะประจุคืออิเล็กตรอนเคลื่อนที่ (ด้วยอัตราเร็วลอยเลื่อน v_d) ในทิศตรงข้ามกับกระแสไฟฟ้า I จากด้านขวาไปด้านซ้าย

รูปที่ 2.4 (ข) เมื่อใส่สนามแม่เหล็ก B ในทิศพุ่งเข้าหาและตั้งฉากกับระนาบแผ่นตัวนำบางหรือกระดาษ จะเกิดแรงแม่เหล็ก F_B กระทำกับอิเล็กตรอน ทำให้อิเล็กตรอนเบนไปทางขอบด้านบนของแผ่นตัวนำบาง

รูปที่ 2.4 (ค) เมื่อเวลาผ่านไปจะมีอิเล็กตรอนถูกผลักไปที่ขอบด้านบนจำนวนมาก ส่วนขอบด้านล่างจะเกิดประจุไฟฟ้าบวกจำนวนมาก เช่นกันการที่มีประจุไฟฟ้าต่างชนิดกันที่ขอบทั้งสอง ทำให้เกิดสนามไฟฟ้า เรียกว่า สนามไฟฟ้าฮอลล์ (Hall Field) E_H ในแผ่นตัวนำบางมีทิศจากขอบด้านล่างไปขอบด้านบน สนามไฟฟ้าจะทำให้เกิดแรงไฟฟ้า F_E กระทำกับอิเล็กตรอน ซึ่งจะทำให้อิเล็กตรอนถูกผลักไปทางขอบด้านล่าง เมื่อแรงไฟฟ้าและแรงแม่เหล็กมีขนาดเท่ากัน อิเล็กตรอนจะเคลื่อนที่ในทิศไปทางซ้ายโดยไม่เบน

สนามไฟฟ้าที่เกิดในแผ่นตัวนำบางมีความสัมพันธ์กับความต่างศักย์หรือโวลเตจ V ดังนี้

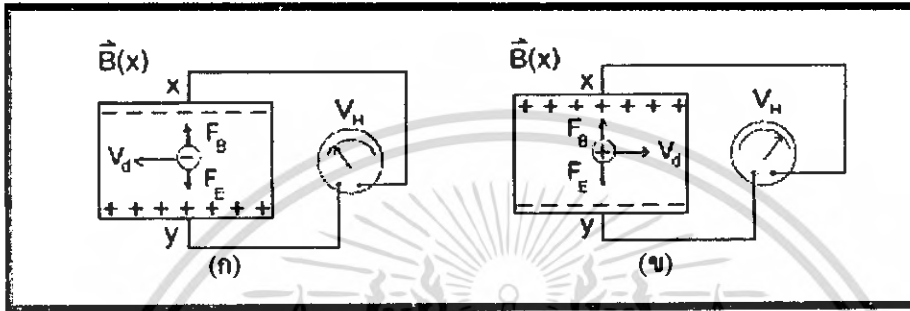
$$E_H = \frac{V}{d} \tag{2.1}$$

ความต่างศักย์หรือโวลเตจที่เกิดขึ้นนี้เรียกว่า ความต่างศักย์ฮอลล์ (Hall Potential Difference หรือ Hall Voltage) V_H พบว่า ความต่างศักย์ฮอลล์มีค่ามากที่สุด เมื่อแผ่นตัวนำบางทำจากสารกึ่งตัวนำ เช่น แออสสารนี้เป็นแออสสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซิลิกอน และเจอร์เมเนียม ส่วนตัวนำไฟฟ้าที่ดี ความต้านทานจะค่าน้อยกว่ามาก (เหตุผล พิจารณาได้จากสมการ (2.7) หรือสมการ (2.8) ในตอนท้าย)

จากสมการ (2.1) จะได้ว่า

$$V_H = E_H d \tag{2.2}$$



รูปที่ 2.5 การวัดความต่างศักย์ฮอลล์ V_H

เราสามารถวัด V_H โดยต่อ มิลลิโวลต์มิเตอร์เข้ากับจุด x และจุด y ดังรูปที่ 2.5 (ก) สภาพขั้วของ V_H ทราบได้จากเครื่องหมายที่อ่านได้จาก มิลลิโวลต์มิเตอร์ จากรูปที่ 2.5 (ก) พาหะประจุคืออิเล็กตรอนจึงมีประจุลบ ถ้าพาหะประจุมีประจุบวก ทิศของ V_d และ E_H จะตรงข้ามกับในรูปที่ 2.5 (ข) แต่ทิศของ F_B และ E_E ยังคงเดิม ดังแสดงในรูปที่ 2.5 (ข) ทำให้ประจุบวกถูกผลักไปที่ขอบด้านขวา ส่วนประจุลบถูกผลักไปที่ขอบด้านซ้าย และสภาพขั้วของ V_H จะตรงข้ามกับกรณีที่พาหะประจุมีประจุลบ

จากรูปที่ 2.4 (ค) ขณะที่แรงแม่เหล็กและแรงไฟฟ้ามีขนาดเท่ากัน เราจะได้

$$(-e)E_H = (-e)V_d B \tag{2.3}$$

$$E_H = V_d B$$

จากสมการ (2.2) จะได้

$$V_H = E_H d$$

$$V_H = V_d B d \tag{2.4}$$

เนื่องจากอัตราเร็วลอยเลื่อน V_d มีค่า

$$V_d = \frac{I}{neA} \tag{2.5}$$

เมื่อ n คือจำนวนพาหะประจุต่อลูกบาศก์เมตร (หรือความหนาแน่นของพาหะประจุ) และ A คือ พื้นที่หน้าตัดของแผ่นตัวนำบาง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แทนสมการ (2.5) ลงในสมการ (2.4) จะได้

$$V_H = \frac{IBd}{neA} \quad (2.6)$$

เนื่องจาก $t = \frac{A}{d}$ คือ ความหนาของแผ่นตัวนำบาง ดังนั้น

$$V_H = \frac{IB}{net} \quad (2.7)$$

สมการ (2.7) เขียนได้ใหม่เป็น

$$B = neti^{-1}V_H \quad (2.8)$$

ปริมาณ $V_H I$ และ t ในสมการ (2.8) หาได้จากการวัด ส่วนค่า n ขึ้นอยู่กับชนิดของวัสดุที่ใช้ทำหัววัด วัสดุที่เป็นสารกึ่งตัวนำจะมีจำนวนพาหะประจุน้อยกว่าตัวนำไฟฟ้าที่ดี แต่ก็ยังมีค่ามากพอที่จะทำให้เกิดกระแสไฟฟ้าที่สามารถวัดได้ ส่วนฉนวนมีจำนวนพาหะประจุน้อยมาก แต่ก็ยอมให้กระแสไฟฟ้าปริมาณเล็กน้อยผ่าน จากการศึกษาพบว่า สารกึ่งตัวนำที่เจือสิ่งเจือปนมีค่า $n \approx 10^{22} \text{ m}^{-3}$ และโลหะทั่วไปมีค่า $n \approx 10^{28} \text{ m}^{-3}$ ดังนั้น เราจึงสามารถหาความเข้มของสนามแม่เหล็กที่ไม่ทราบค่าจากสมการ(2.8) ได้

ความเข้มของสนามแม่เหล็กมีหน่วยในระบบเอสไอเป็นเทสลา (Tesla) แทนด้วยสัญลักษณ์ T หน่วยเดิมของความเข้มของสนามแม่เหล็กคือ เกาส์ (Gauss) แทนด้วยสัญลักษณ์ G โดยที่ $1\text{T} = 10^4 \text{ G}$

2.3 การสื่อสารข้อมูลอนุกรม

การสื่อสารข้อมูลอนุกรมเป็นการรับหรือส่งข้อมูลในลักษณะกลุ่มของบิต คราวละหนึ่งบิต เรียงลำดับเรื่อยไปจนถึงสิ้นสุด การสื่อสารแบบนี้จะมีข้อแตกต่างจากการสื่อสารแบบขนานเป็นอย่างมาก เนื่องจากการสื่อสารข้อมูลแบบขนานมีการ โอนย้ายมาพร้อมกัน จึงมีความจำเป็นต้องใช้จำนวนเส้นสัญญาณมากขึ้นตามจำนวนบิตของข้อมูลด้วย ในขณะที่การสื่อสารแบบอนุกรมนั้นต้องการเส้นสัญญาณเพียงสองหรือสามเส้นเท่านั้น ดังนั้นการสื่อสารแบบขนานจึงไม่เหมาะในการสื่อสารกับอุปกรณ์ภายนอกเป็นระยะทาง ไกลๆเพราะจะทำให้สิ้นเปลืองค่าใช้จ่ายมาก

2.3.1 ความเร็วของการสื่อสารข้อมูลอนุกรม

เนื่องจากการสื่อสารข้อมูลแบบอนุกรม เป็นการรับ/ส่งข้อมูลในลักษณะกลุ่มของบิตข้อมูล (Bit stream) ดังนั้น จึงต้องให้ความสนใจในการพิจารณาเรื่องอัตราเร็วในการรับ/ส่งบิตเหล่านี้เป็นอันดับแรกโดยทั่วไปมักจะระบุกันในหน่วยของจำนวนบิตข้อมูลภายในเวลาหนึ่งวินาที เรียกว่า อัตราบอด ตามค่ามาตรฐานเหล่านี้ ได้แก่ 110,150,300,1200,2400,4800,9600,19200 บอด ข้อมูลทั้งแปดบิตนี้หากว่าถูกส่งออกมาด้วยอัตรา 9600 บอด จะใช้เวลาในการส่งข้อมูลหนึ่งบิตมีค่าเท่ากับ $1/9600$ หรือ $104 \mu\text{s}$ และเวลาในการส่งข้อมูลทั้งแปดบิตมีค่าเท่ากับ 8×104 หรือ $832 \mu\text{s}$

2.3.2 รูปแบบของการส่งข้อมูลอนุกรม

การสื่อสารอนุกรมแบบอะซิงโครนัสจะใช้การแปลงข้อมูลขนานให้เป็นอนุกรมแล้วเพิ่มเคมิตบดบอย่างร่วมไปกับการส่งข้อมูลจริงซึ่งได้แก่

1. บิตเริ่มต้น (Start Bit)

บิตเริ่มต้นมีหน้าที่สำหรับการบ่งบอกให้ทราบถึงตำแหน่งเริ่มต้นก่อนบิตข้อมูล ตามปกติแล้วค่าของบิตเริ่มต้นจะเป็นระดับลอจิกต่ำ

2. บิตแสดงสถานะความเป็นเลขคู่หรือเลขคี่ (Parity Bit)

บิตนี้มีหน้าที่เพื่อการตรวจสอบความถูกต้องของข้อมูลโดยทั่วไปมักเรียกว่า บิตพาริตีและจะนำไปต่อท้ายบิตของข้อมูล ค่าของบิตนี้ขึ้นอยู่กับจำนวนค่าของบิตที่เป็น 1 ซึ่งจะเป็นได้สองลักษณะคือ พาริตีคู่ (Even Parity) หรือพาริตีคี่ (Odd Parity) ตัวอย่างเช่นระบบที่ติดต่อกันโดยระบุว่าจะใช้พาริตีคู่ทางด้านส่งจะนำค่าข้อมูลที่จะส่งมาพิจารณาจำนวนของบิตที่มีค่าเป็น 1 หากเป็นเลขจำนวนคู่อยู่แล้วค่าของพาริตีจะมีค่าเป็น 0 แต่หากว่าจำนวนของบิตที่มีค่าเป็น 1 เป็นเลขจำนวนคี่ ค่าของพาริตีก็จะมีค่าเป็น 1 การพิจารณาทางด้านรับเป็นการตรวจสอบจำนวนบิตที่มีค่าเป็น 1 ของข้อมูลที่ได้รับมาทั้งหมดรวมทั้งบิตพาริตี ถ้ามีค่าเป็นเลขจำนวนคู่ แสดงว่าข้อมูลที่ได้รับเข้ามาถูกต้องแต่หากไม่เป็นเลขจำนวนคู่แสดงว่าเกิดการผิดพลาดของข้อมูลขึ้น เป็นต้น

3. บิตสิ้นสุด (Stop Bit)

บิตสิ้นสุดเป็นบิตที่เพิ่มขึ้นเพื่อระบุถึงขอบเขตการสิ้นสุดของกลุ่มบิตข้อมูล บิตสิ้นสุดสามารถโปรแกรมได้คือ 1 บิต, $1\frac{1}{2}$ บิต และ 2 บิต ดังนั้นกรณีของการส่งข้อมูล 8 บิต หากข้อมูลถูกส่งออกไปด้วยอัตราเร็ว 9600 บิตต่อวินาทีในการส่งข้อมูลหนึ่งไบต์ จะมีค่าเป็น 12×10^4 หรือ 1.25 ms

2.4 ไมโครคอนโทรลเลอร์ MCS-51

2.4.1 การใช้งานพอร์ตสื่อสารอนุกรม

พอร์ตสื่อสารอนุกรมมีโครงสร้างการทำงานในแบบที่เรียกว่าฟูลดูเพล็กซ์ (Full Duplex) สามารถรับและส่งข้อมูลอนุกรมได้ในเวลาเดียวกัน

Serial Port Buffer (SBUF) ใช้เป็นบัฟเฟอร์สำหรับรับและส่งข้อมูลอนุกรมโดยมีอยู่ 2 ตัว พอร์ตสื่อสารอนุกรมสามารถโปรแกรมการทำงานได้หลายโหมดด้วยกัน โดยเลือกที่ละบิต SM0 และ SM1 ซึ่งอยู่ในรีจิสเตอร์ควบคุม SCON การทำงานของทั้ง 4 โหมด ของพอร์ตสื่อสารอนุกรม มีดังตารางที่ 2.8

ตารางที่ 2.8 โหมดการทำงานของพอร์ตสื่อสารอนุกรม

SM0	SM1	โหมด	การทำงาน
0	0	0	Shift Register ความเร็วในการรับหรือส่งข้อมูลเท่ากับ (1/2) ของ CPU Osc
0	1	1	8 Bit UART ความเร็วในการรับหรือส่งข้อมูลกำหนดได้จาก Timer 1,2
1	0	2	9 Bit UART ความเร็วในการรับหรือส่งข้อมูล = (1/32) หรือ (1/64) เท่าของ CPU Osc โดยขึ้นกับบิต SMOD ใน PCON
1	1	3	9 Bit UART ความเร็วในการรับหรือส่งข้อมูลกำหนดที่ Timer 1,2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โหมด 0 : พอร์ตสื่อสารอนุกรม 8 บิต โดยการส่งข้อมูลเลื่อนออกทีละบิตโดยส่งบิต D0 ออกไปก่อน ทางขา RxD เนื่องจากไม่มีการส่งบิตเริ่มต้น แต่จะส่ง Shift Clock ทางขา TxD (ความเร็ว 1/12 เท่าของ CPU Osc)

โหมด 1 : พอร์ตสื่อสารข้อมูลอนุกรม 10 บิต ข้อมูล 8 บิต 1 บิตเริ่มต้น และ 1 บิตสิ้นสุด และสามารถเปลี่ยนแปลงความเร็วในการส่งข้อมูลได้ โดยขึ้นกับบิต SMOD ใน PCON และอัตราโอเวอร์โพล์ของ ไทเมอร์ 1,2

$$\text{บอดเรท โหมด 1,3} = \frac{2^{\text{SMOD}} \times \text{CPU Osc}}{32 \times 12 \times [256 - (\text{TH1})]} \quad \text{โดยใช้ ไทเมอร์ 1} \quad (2.9)$$

$$\text{บอดเรท โหมด 1,3} = \frac{\text{CPU Osc}}{32 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]} \quad \text{โดยใช้ ไทเมอร์ 2} \quad (2.10)$$

โหมด 2 : พอร์ตสื่อสารอนุกรม 11 บิต ข้อมูล 9 บิต 1 บิตเริ่มต้น และ 1 บิตสิ้นสุด (TB8 นิยมนำมาใช้ส่งพาริตีบิต) ความเร็วในการรับส่งข้อมูลเท่ากับ 1/32 หรือ 1/64 เท่าของ CPU Osc โดยขึ้นกับบิต SMOD ใน PCON

$$\text{บอดเรท (โหมด 2)} = \frac{(2^{\text{SMOD}}) \text{CPU Osc}}{64} \quad (2.11)$$

- บอดเรท (โหมด 2) = 1/32 CPU Osc เมื่อ SMOD = 1
- บอดเรท (โหมด 2) = 1/64 CPU Osc เมื่อ SMOD = 0

โหมด 3 : พอร์ตสื่อสารอนุกรม 11 บิต ข้อมูล 9 บิต 1 บิตเริ่มต้น และ 1 บิตสิ้นสุด เหมือนโหมด 2 ยกเว้นอัตราความเร็วจะขึ้นกับบิต SMOD ใน PCON และอัตราโอเวอร์โพล์ของ ไทเมอร์ 2

$$\text{บอดเรท โหมด 1,3} = \frac{2^{\text{SMOD}} \times \text{CPU Osc}}{32 \times 12 \times [256 - (\text{TH1})]} \quad \text{โดยใช้ ไทเมอร์ 1} \quad (2.12)$$

$$\text{บอดเรท โหมด 1,3} = \frac{\text{CPU Osc}}{32 \times [65536 - (\text{RCAP2H}, \text{RCAP2L})]} \quad \text{โดยใช้ ไทเมอร์ 2} \quad (2.13)$$

2.4.2 กระบวนการรับและส่งข้อมูลอนุกรมของ MCS-51

การส่งข้อมูลออกทางพอร์ตอนุกรมของ MCS-51 จะเริ่มต้นขึ้นภายหลังเมื่อมีการเขียนข้อมูลลงไป ใน SBUF ข้อมูลนี้ จะถูกเลื่อนทีละบิต และส่งสัญญาณออกไปภายนอกโดยอัตโนมัติ เมื่อข้อมูลเหล่านี้ได้ส่งออกไปครบถ้วนแล้วจะทำให้ค่าแฟล็กซ์ T1 เป็น 1 เพื่อแจ้งให้ทราบว่าขณะนี้ SBUF ว่างและพร้อมที่

จะส่งข้อมูล 1 ไบต์ต่อไปแล้ว ในกรณีที่ผู้ใช้เขียนข้อมูลใหม่ลงในรีจิสเตอร์ SBUF โดยไม่รอให้แฟล็กซ์ T1 มีค่าเป็น 1 ก่อน จะมีผลทำให้ข้อมูลที่ส่งไปผิดพลาดได้

สำหรับการรับข้อมูลจากพอร์ทอนุกรมจะต้องเริ่มต้นโดยการกำหนดค่า REN (Receive Enable) ให้มีค่าเป็น 1 ก่อน หลังจากนั้นเมื่อมีข้อมูลภายนอกถูกส่งเข้ามายัง 8051 ที่ละบิตจนครบ และเมื่อบิตสุดท้ายเลื่อนเข้ามาเรียบร้อยแล้ว ข้อมูลนั้นจะถูกย้ายมาเก็บไว้ยังรีจิสเตอร์ SBUF และแฟล็กซ์ RI ก็จะมีค่าเป็น 1 (ถูกเซต)

1. พอร์ทอนุกรม (โหมด 0)

การทำงานของพอร์ทอนุกรม (โหมด 0) เป็นการรับและส่งข้อมูลอนุกรมจำนวน 8 บิต โดยใช้เพียงขาสัญญาณ Rx/D เท่านั้น (ขานี้ใช้งาน 2 หน้าที่ใช้ส่งและรับข้อมูล) ส่วนขาสัญญาณ Tx/D จะนำไปใช้เพื่อเป็นขาสัญญาณนาฬิกาในการให้จังหวะ การเลื่อนข้อมูลกับวงจร เลื่อนบิตภายนอก สำหรับอัตราเร็วจะถูกกำหนดไว้คงที่ที่ค่า 1/12 เท่าของ CPU Osc จากรูปที่ 2.6 แสดงให้เห็นถึงแผนภาพเวลาสัญญาณต่างๆ ในโหมด 0 เมื่อมีการรับหรือส่งข้อมูล 1 ไบต์ โดยสัญญาณนาฬิกาในการเลื่อนบิตนี้จะเกิดภายในตัว 8051 เอง เนื่องจากโหมดนี้ไม่มีการส่งบิตเริ่มต้นและบิตสิ้นสุด ดังนั้นจึงจำเป็นที่จะต้องส่งสัญญาณ Shift Clock ออกไป เพื่อใช้ Synchronize ระหว่างฝ่ายรับและฝ่ายส่ง โดยจะใช้ขา Tx/D ส่วนการรับข้อมูลจะรับข้อมูลเข้าทางขา Rx/D และรับ Shift Clock เข้าทางขา Tx/D ถ้า CPU Osc มีค่าเท่ากับ 12 MHz ก็จะส่งได้ถึง 1 ล้านบิต ซึ่งโหมด 0 เป็นโหมดที่ส่งข้อมูลได้เร็วที่สุด รายละเอียดผังเวลาในการรับส่งดังแสดงในรูปที่ 2.6

2. พอร์ทอนุกรม (โหมด 1)

การทำงานในโหมด 1 เป็นการสื่อสารข้อมูลอนุกรมจำนวน 10 บิต ประกอบด้วยบิตเริ่มต้น 1 บิต ข้อมูลจำนวน 8 บิต และบิตสิ้นสุดอีก 1 บิต ดังแสดงในรูปที่ 2.7 โดยข้อมูลจะถูกส่งออกทาง Tx/D และรับเข้ามาทางขาสัญญาณ Rx/D ในส่วนของข้อมูล 8 บิต ที่ได้รับหรือทำการส่งออก จะเป็นบิตนัยสำคัญต่ำเป็นลำดับแรก ทางฝ่ายรับค่าของบิตสิ้นสุดจะส่งเข้ามาจัดเก็บไว้ในบิต RB8 ภายในรีจิสเตอร์ SCON สำหรับอัตราเร็วในการส่งข้อมูลของโหมด 1 นั้น สามารถเลือกได้จากไทมเมอร์ 1 ผังเวลาการทำงานแสดงดังในรูปที่ 2.7

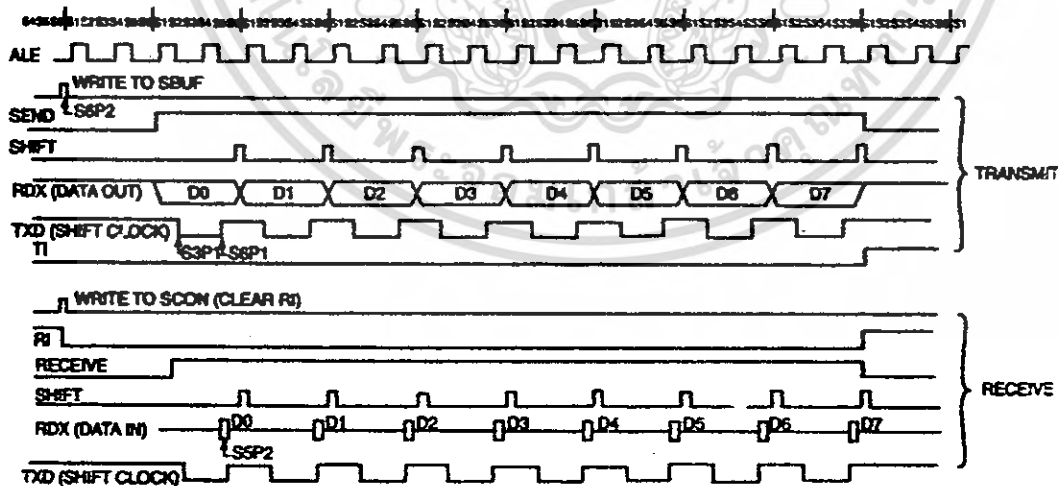
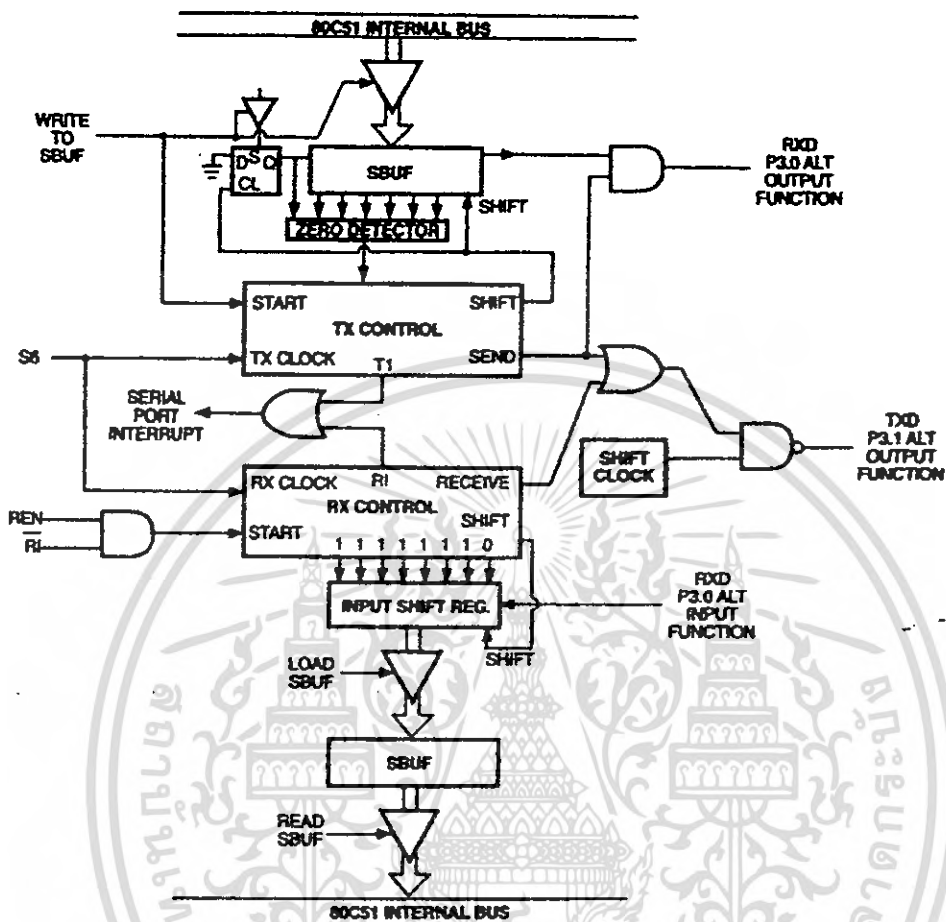
3. พอร์ทอนุกรม (โหมด 2)

โหมด 2 ใช้ทั้งหมด 11 บิต โดยแบ่งเป็น บิตเริ่มต้น 9 บิตข้อมูล และบิตสิ้นสุด โดยบิตที่ 9 ผู้ใช้สามารถกำหนดค่าเองได้ว่าจะส่งค่าอะไรออกไป โดยจะต้องนำไปใส่ไว้ในบิต TB8 ในรีจิสเตอร์ SCON ส่วนมากผู้ใช้นักจะนำบิตนี้มาใช้เป็นพาร์ตีบิต โดยรับค่ามาจากพาร์ตีแฟล็กซ์ใน PSW ส่วนทางด้านรับบิตที่ 9 จะถูกนำไปเก็บไว้ใน RB8 อัตราเร็วในการส่ง/รับข้อมูลกับ CPU Osc และค่า SMODซึ่งอยู่ในบิต 7 ใน SCON ผังเวลาการทำงานแสดงดังในรูปที่ 2.8

4. พอร์ทอนุกรม (โหมด 3)

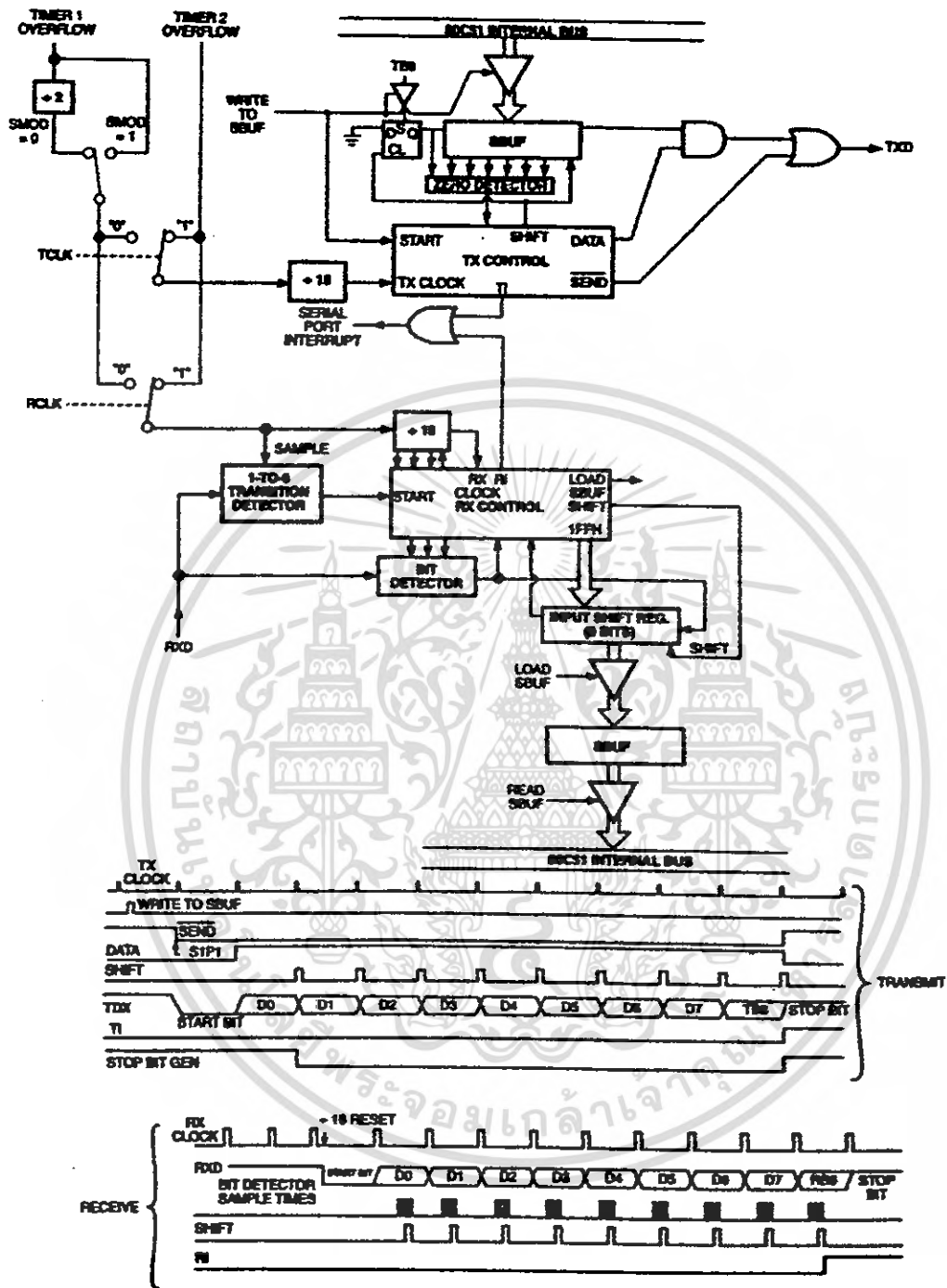
การทำงานเหมือนกับโหมด 2 ทุกอย่าง จะแตกต่างกันที่ความเร็วในการรับส่งข้อมูลจะขึ้นอยู่กับอัตราโอเวอร์โพล์ของไทมเมอร์ 1 หรือ ไทมเมอร์ 2 โดยมีผังการทำงานแสดงดังในรูปที่ 2.9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



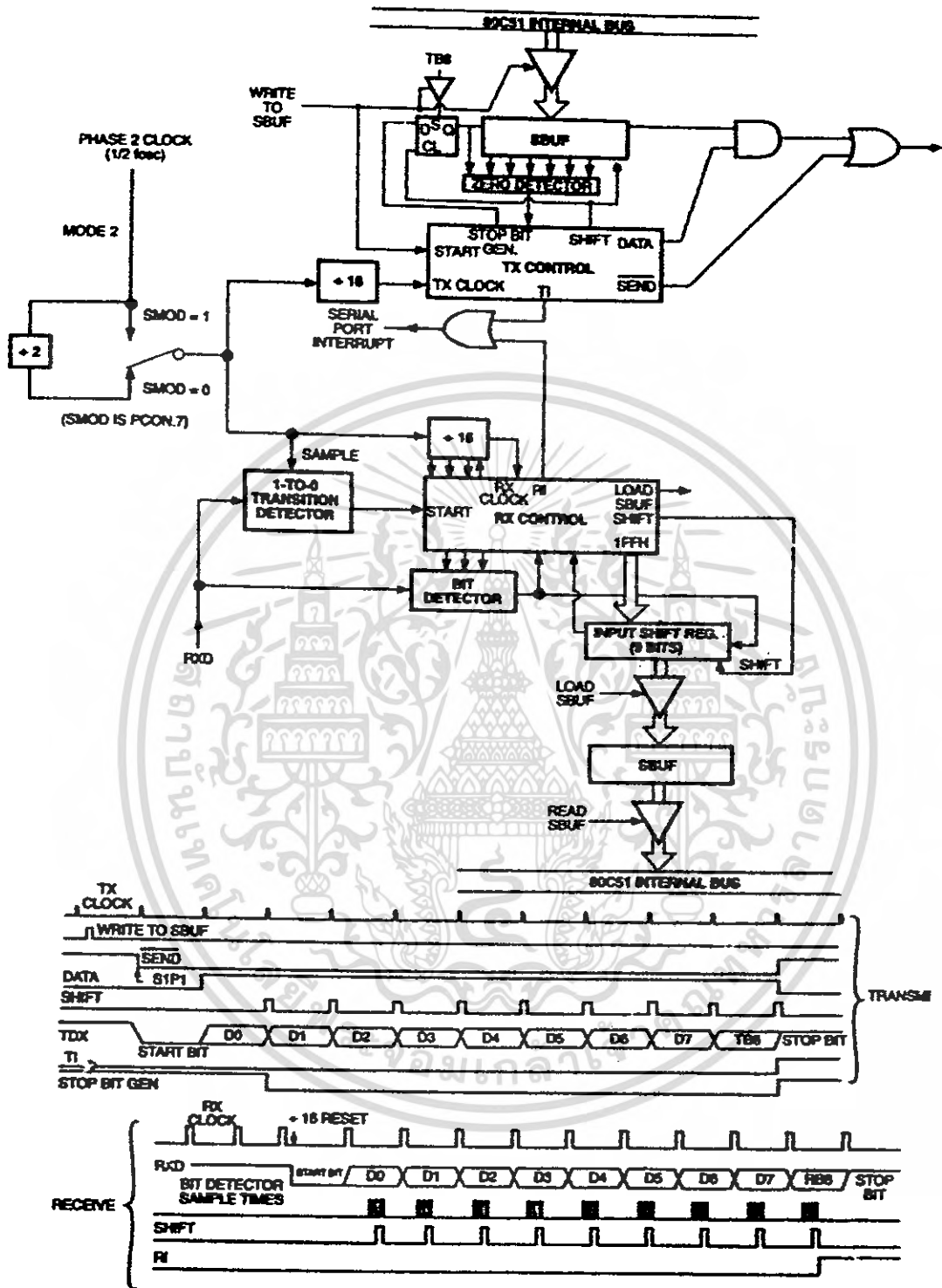
รูปที่ 2.6 ฟังก์ชันการทำงานโหมด 0

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.7 หังการทํางานโหมด 1

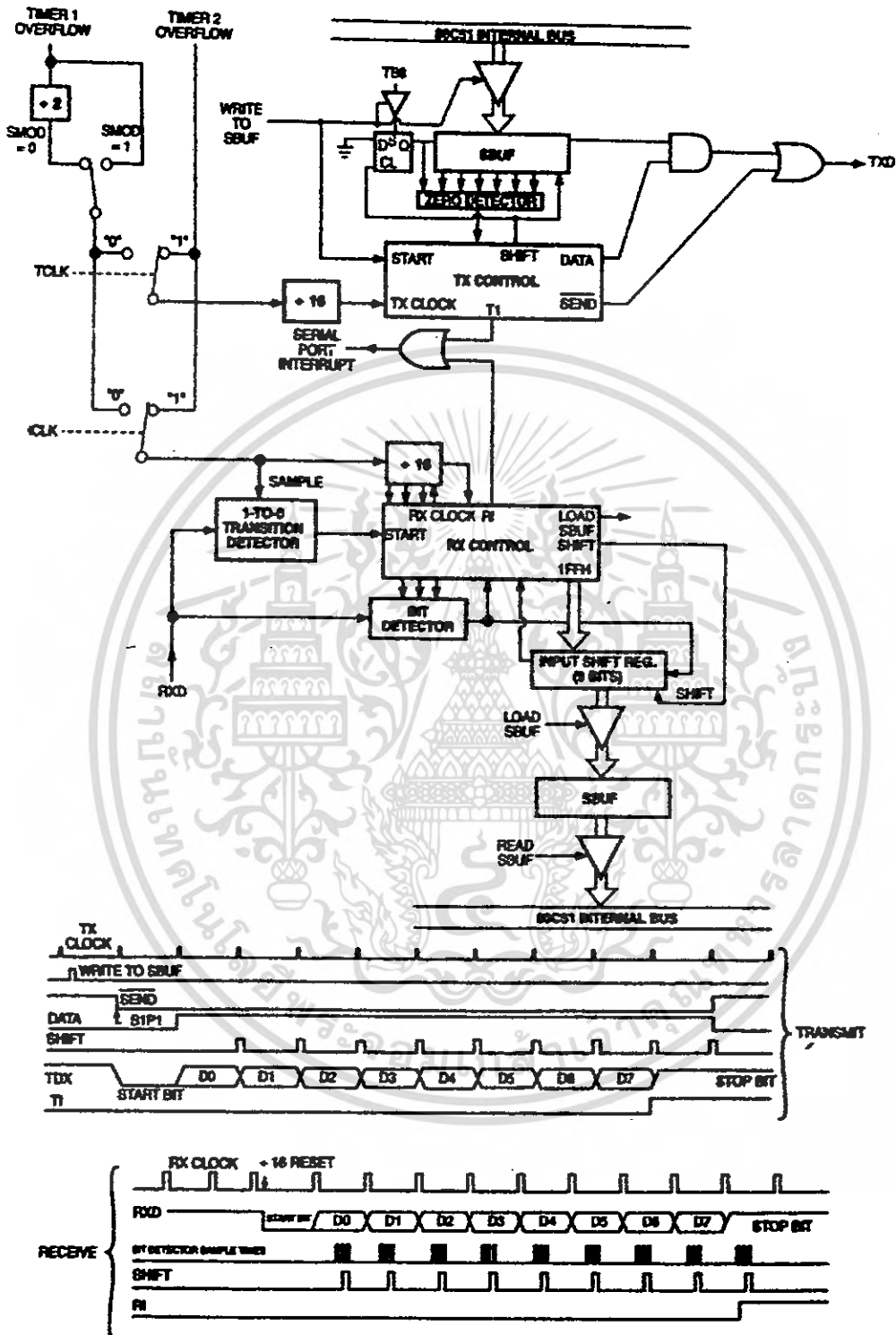
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.8 ฝั่งการทำงานโหมด 2

72917

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.9 หังการทำงานโมเด็ม 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 รีจิสเตอร์ที่ใช้ควบคุมการรับส่งข้อมูลอนุกรม SCON

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
-----	-----	-----	-----	-----	-----	----	----

SM0, SM1 : บิตเลือกโหมดการทำงาน

SM2 : บิตเลือกการทำงานแบบ Single Processor Environment หรือ Multiprocessor Environment

1 : เลือก Multiprocessor Environment ใช้ได้กับโหมด 2, 3

0 : เลือก Single Processor Environment ใช้ได้กับทุกโหมด

REN (Receive Enable): บิตควบคุมให้รับหรือไม่ให้รับข้อมูล

1 : ให้รับข้อมูล

0 : ห้ามรับข้อมูล

TB8 (Transmit bit D8): ข้อมูลบิตที่ 9 ที่จะส่งออกไปใน โหมด 2, 3 ให้ใส่ในบิตนี้

RB8 (Receive bit D8): ข้อมูลบิตที่ 9 ที่รับเข้ามาจะมากับในบิตนี้

TI: แฟล็กซ์ TI จะเป็น 1 เมื่อสิ้นสุดการส่งข้อมูล 1 ไบต์

RI: แฟล็กซ์ RI จะเป็น 1 เมื่อรับข้อมูลเสร็จ 1 ไบต์

2.4.4 หน่วยความจำและการเชื่อมต่อกับ MCS – 51

การสร้างระบบไมโครคอนโทรลเลอร์โดยการใช้ชิป MCS-51 จำเป็นต้องมีการขยายหน่วยความจำภายนอกสำหรับเก็บ โปรแกรมและเก็บข้อมูลจากการประมวลผล เพราะว่าชิป MCS-51 บางรุ่นไม่มีหน่วยความจำโปรแกรมภายใน บางเบอร์มีหน่วยความจำภายในขนาดเล็กซึ่งไม่พอต่อการใช้งานหน่วยความจำที่ใช้เก็บโปรแกรมเรียกว่าหน่วยความจำโปรแกรม (Program Memory)

หน่วยความจำโปรแกรม (Program Memory)

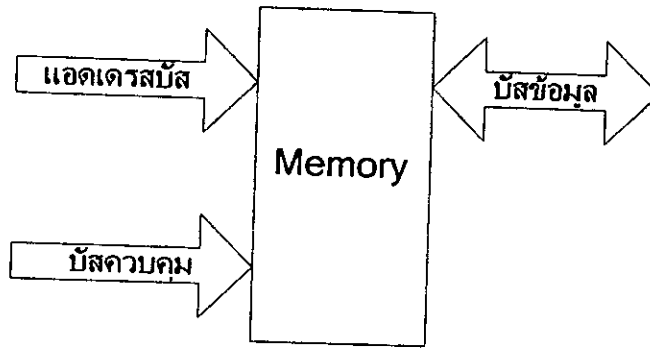
เป็นหน่วยความจำที่สามารถเก็บข้อมูลได้แม้ว่าไม่มีไฟเลี้ยง ข้อมูลก็จะยังคงอยู่ หน่วยความจำประเภทนี้มักใช้เก็บโปรแกรมระบบที่มีการเปลี่ยนแปลงอีก เช่น BIOS ในคอมพิวเตอร์ โปรแกรมมอนิเตอร์ในบอร์ดควบคุมต่างๆ หน่วยความจำประเภทนี้แบ่งได้หลายชนิดแต่เรียกรวมๆว่า ROM

1. EEPROM (Electrically Erasable PROM)

หน่วยความจำชนิดนี้จะคล้ายกับ EPROM แต่สามารถลบได้โดยใช้กระแสไฟฟ้าโดยไม่ต้องใช้แสง UV

การอ่านและเขียนข้อมูลกับหน่วยความจำ

ไอซีหน่วยความจำโดยทั่วไปจะประกอบด้วยกลุ่มสัญญาณ 3 กลุ่ม คือแอดเดรสบัส, บัสข้อมูล และบัสควบคุม โดยแอดเดรสบัสจะใช้ในการอ้างตำแหน่ง ถ้ามีแอดเดรสบัส 10 เส้น คือ A0 – A9 สามารถอ้างตำแหน่งได้ 2^{10} หรือ 1 กิโลไบต์ บัสข้อมูลจะเป็นตัวบอกว่าในแต่ละตำแหน่งจะเก็บข้อมูลได้อีกกี่บิต เช่น ถ้ามีข้อมูล 8 เส้น คือ D0 – D7 สามารถเก็บข้อมูลได้ตำแหน่งละ 1 ไบต์ ส่วนบัสควบคุมจะใช้ในการอ่านเขียนหน่วยความจำ ระบบบัสทั้ง 3 แสดงได้ดังรูปที่ 2.10



รูปที่ 2.10 แสดงกลุ่มสัญญาณที่ใช้ในการติดต่อกับหน่วยความจำ

โดยทั่วไปไอซีหน่วยความจำจะมีขา CE (Chip Enable) ไว้สำหรับเลือกให้ไอซีตัวที่ต้องการใช้ทำงาน ในกรณีที่มีไอซีหน่วยความจำหลายๆ ตัว ขา OE (Output Enable) จะใช้ต่อกับไอซีถอดรหัสสำหรับการอ่านข้อมูล ถ้าขานี้แอกทีฟ ข้อมูลที่อยู่ในไอซีหน่วยความจำจะส่งออกมาทางบัสข้อมูลได้ การอ่านข้อมูลจากหน่วยความจำ มีลำดับขั้นดังนี้

1. ส่งตำแหน่งที่จะอ่านไปก่อนทางแอดเดรสบัส
2. ส่งสัญญาณควบคุมว่าต้องการจะอ่าน
3. ข้อมูลในไอซีหน่วยความจำถูกส่งออกมาทางบัสข้อมูล

การอ่านข้อมูลจากหน่วยความจำสามารถเขียนเป็นไคอะแกรมเวลาได้ดังรูปที่ 2.11



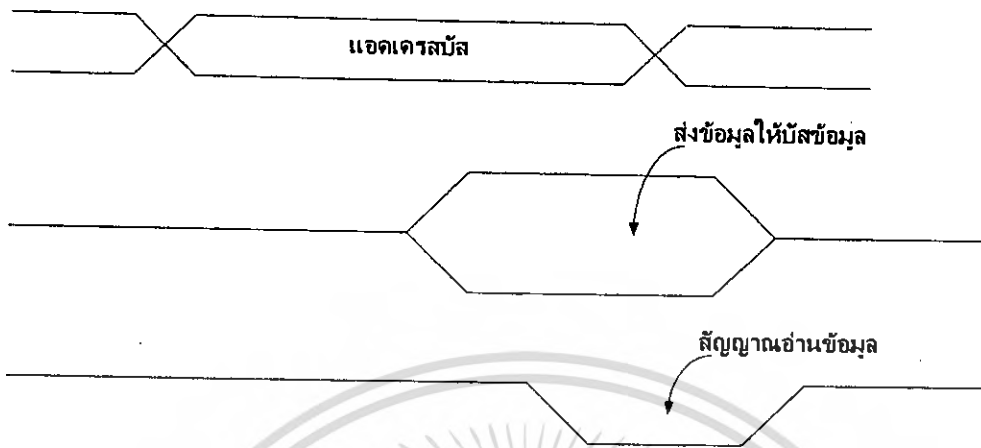
รูปที่ 2.11 แสดงสัญญาณในการอ่านข้อมูล

การเขียนข้อมูลลงหน่วยความจำ มีลำดับขั้นดังนี้

1. ส่งตำแหน่งที่จะเขียนข้อมูลออกไปก่อนทางแอดเดรสบัส
2. ส่งข้อมูลที่จะเขียนไปทางบัสข้อมูล
3. ส่งสัญญาณเขียนข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

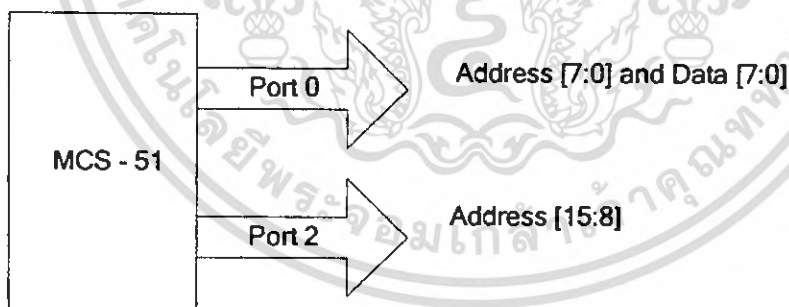
การเขียนข้อมูลให้กับหน่วยความจำสามารถเขียนเป็นไคอะแกรมเวลาได้ดังรูปที่ 2.12



รูปที่ 2.12 แสดงสัญญาณในการเขียนข้อมูล

การเชื่อมต่อหน่วยความจำกับ MCS – 51

สัญญาณที่ใช้ในการติดต่อกับหน่วยความจำประกอบด้วย แอดเดรสบัส บัสข้อมูล และบัสควบคุม ใน MCS – 51 สามารถติดต่อกับหน่วยความจำภายนอกได้ 64 กิโลไบต์ โดยจะใช้สายสัญญาณแอดเดรสบัส 16 เส้น ส่งออกมาทาง พอร์ต 0 และ พอร์ต 2 โดยพอร์ต 0 จะใช้ Multiplexed ระหว่างแอดเดรสบัส และบัสข้อมูล โดยแอดเดรสบัสจะเป็นบิตค่า A0 – A7 ดังรูปที่ 2.13

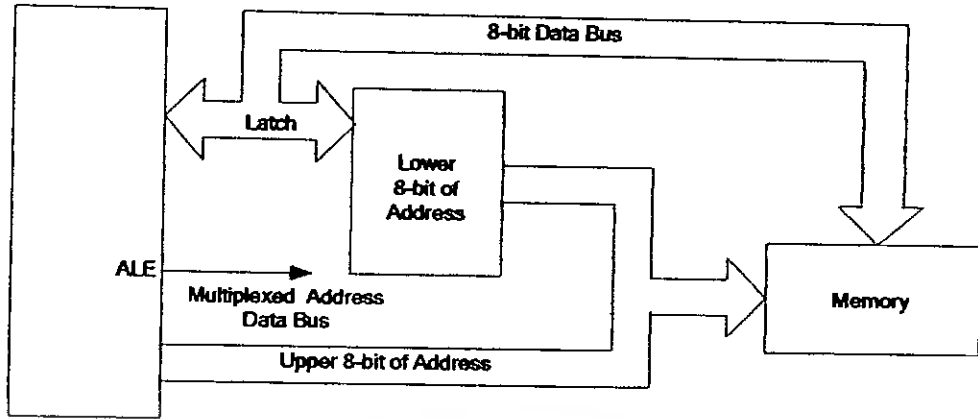


รูปที่ 2.13 แสดงสัญญาณของ MCS – 51 ที่ใช้ติดต่อกับตำแหน่งของหน่วยความจำ

ในการเชื่อมต่อกับหน่วยความจำจะต้องมีอุปกรณ์ภายนอกมา Latch สัญญาณแอดเดรสบัส ที่ได้จากพอร์ต 0 เพื่อที่จะใช้พอร์ต 0 เป็นบัสข้อมูลต่อไปนี้ ในรูปที่ 2.14 จะแสดงการต่อหน่วยความจำประเภท ROM และ RAM กับ MCS – 51 โดยมีอุปกรณ์ภายนอกมา Latch ค่าแอดเดรสไบต์ค่าเอาไว้เรียกว่า “Address Latching”

อุปกรณ์ที่นิยมใช้ได้แก่ ไอซี TTL โดยใช้สัญญาณที่ใช้ Latch คือสัญญาณ ALE จาก MCS – 51

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

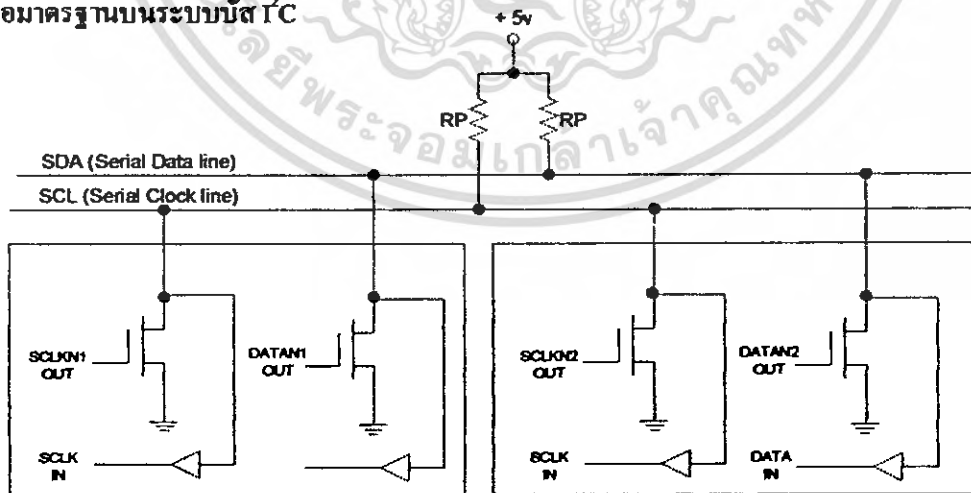


รูปที่ 2.14 แสดงการต่อหน่วยความจำประเภท ROM และ RAM กับ MCS - 51

2.4.5 ระบบบัสแบบ I²C และการเชื่อมต่อ

I²C ย่อมาจาก Inter-IC Communication หมายถึง การติดต่อสื่อสารระหว่างไอซี โดยบัส I²C ได้รับการพัฒนาขึ้นโดยฟิลิปส์ (Philips) ด้วยจุดมุ่งหมายหลักคือ ต้องการให้อไอซีหรือไมโครสามารถติดต่อถึงงาน และควบคุมภายใต้สายสัญญาณเพียง 2 เส้น เส้นหนึ่งคือ สายข้อมูลอนุกรม หรือ SDA (Serial Data Line) อีกเส้นหนึ่งคือ สัญญาณนาฬิกาอนุกรม หรือ SCL (Serial Clock Line) ใช้ในการกำหนดจังหวะการทำงาน การต่อร่วมกันของอุปกรณ์บนบัส I²C ทำได้ง่ายมาก เพียงต่อสายข้อมูลและสายสัญญาณนาฬิกาของอุปกรณ์แต่ละตัวขนานกันไป ส่วนการกำหนดแอดเดรสสำหรับติดต่ออุปกรณ์แต่ละตัวใช้รหัสข้อมูลร่วมกับการกำหนดสถานะลอจิกที่ขาแอดเดรส

อัตราการส่งข้อมูลบนบัส I²C สูงถึง 100 กิโลบิตต่อวินาทีในโหมดปกติ (Standard Mode) และสูงถึง 400 กิโลบิตต่อวินาทีในโหมดความเร็วสูง (Fast Mode) การเข้าถึงอุปกรณ์บนบัส I²C ใช้ข้อมูลสำหรับการเข้าถึง 2 คำคือ 7 บิต (7-bit addressing) หรือ 10 บิต (10-bit addressing) ในรูปที่ 2.15 แสดงการเชื่อมต่อมาตรฐานบนระบบบัส I²C



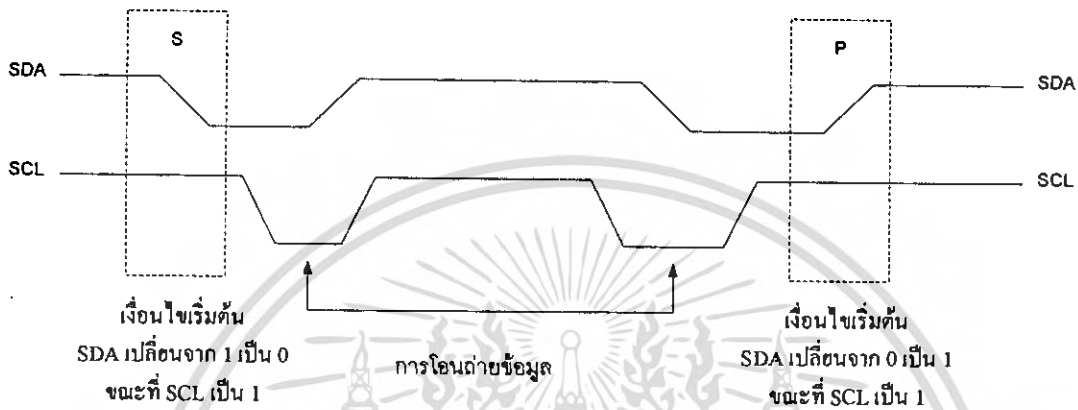
อุปกรณ์ I²C ตัวที่ 1

อุปกรณ์ I²C ตัวที่ 2

รูปที่ 2.15 รูปแบบการเชื่อมต่อมาตรฐานของระบบบัส I²C

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การติดต่อสื่อสารกับอุปกรณ์แต่ละตัวจะมีรูปแบบการรับส่งข้อมูลหรือ โพรโตคอล (Protocol) ที่อุปกรณ์ทุกตัวรู้จัก ถ้าหากต้องการให้อุปกรณ์ตัวใดเป็นตัวรับข้อมูล ตัวส่งจะส่งแอดเดรสของอุปกรณ์ตัวนั้นไปก่อนถ้าหากอุปกรณ์ตัวใดมีแอดเดรสตรงกันก็จะรับข้อมูลนั้นไป สำหรับตัวอุปกรณ์ที่ต้องการส่งข้อมูลจะเรียกว่ามาสเตอร์ (Master) ซึ่งเป็นตัวที่สร้างจังหวะสัญญาณต่างๆ บนระบบบัส ส่วนอุปกรณ์ที่ถูกควบคุมหรือเป็นตัวรับข้อมูลจะเรียกว่าสเลฟ (Slave)



รูปที่ 2.16 จังหวะสัญญาณเวทบนบัส

1. สถานะที่เกิดขึ้นบนบัส I²C

1. บัสว่าง (Bus Not Busy) สถานะนี้เกิดขึ้นเมื่อสถานะลอจิกบนสาย SDA และ SCL เป็นลอจิกสูงทั้งคู่ นั่นหมายความว่า การถ่ายโอนข้อมูลสามารถเริ่มต้นขึ้นได้

2. เริ่มต้นส่งข้อมูล (Start Data Transfer) สถานะนี้สาย SCL จะเป็นลอจิกสูง แต่สาย SDA จะเปลี่ยนจากลอจิกสูงไปเป็นลอจิกต่ำ เรียกว่าสถานะเริ่มต้น (Start)

3. สถานะหยุด (Stop) สถานะนี้สาย SCL จะเป็นลอจิกสูง แต่สาย SDA จะเปลี่ยนจากลอจิกต่ำไปเป็นลอจิกสูง

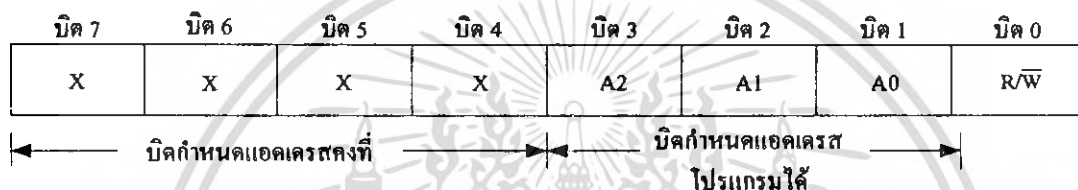
4. สถานะมีข้อมูล (Data Valid) สถานะนี้จะอยู่ระหว่างสถานะเริ่มต้นและสถานะหยุด โดยการรับส่งข้อมูลต่างๆ จะเกิดในสถานะนี้ การรับส่งข้อมูลแต่ละบิตจะใช้สัญญาณนาฬิกาหนึ่งลูก โดยข้อมูลบน SDA จะต้องคงที่ ขณะที่ SCL เป็นลอจิกสูง และบิตข้อมูลใน SDA จะเปลี่ยนแปลงได้ขณะที่ SCL เป็นลอจิกต่ำ ถ้าหากบิตบน SDA มีการเปลี่ยนแปลงขณะที่ SCL เป็นลอจิกสูง ระบบจะตีความว่าเป็นสถานะเริ่มต้นส่งข้อมูลหรือสถานะหยุดแทน

5. สถานะตอบรับ (Acknowledge) เมื่ออุปกรณ์มาสเตอร์ส่งข้อมูลมาครบหนึ่งไบต์แล้วในช่วงสัญญาณ SCL ลูกที่ 9 มาสเตอร์จะส่งข้อมูลลอจิกสูงออกมา และถ้าตัวรับได้รับข้อมูลครบแล้วมันจะส่งสัญญาณตอบรับ ACK โดยทำให้ระดับลอจิกสูงบนสัญญาณ SDA ให้กลับเป็นลอจิกต่ำ แต่ถ้าตัวรับได้รับข้อมูลไม่ถูกต้อง ตัวรับจะบังคับให้ตัวส่งหยุดในสถานะรอ

2. การอ้างถึงแบบ 7 บิต (7 – bit addressing)

ก่อนเริ่มส่งข้อมูลระหว่างอุปกรณ์ต่างๆ ที่ต่ออยู่บนบัส ต้องมีการอ้างถึงอุปกรณ์เสียก่อน โดย การอ้างถึงอุปกรณ์บนบัส IC นั้นจะใช้การอ้างถึงแบบ 7 บิตหรือ 10 บิต ในกรณีที่มีอุปกรณ์ต่ออยู่บนบัส ไม่มาก ใช้การอ้างถึงแบบ 7 บิตก็เพียงพอ แต่ถ้ามีอุปกรณ์ต่ออยู่บนบัสมากกว่า 127 แอดเดรส จำเป็นต้องใช้การอ้างถึงแบบ 10 บิต หลังจากที่ติดต่อกับอุปกรณ์แต่ละตัวได้เรียบร้อยแล้วก็จะเริ่มดำเนินการส่ง ข้อมูลต่อไป รูปแบบของข้อมูลมาตรฐานที่ใช้ในการเข้าถึงแสดงในรูปที่ 2.17

ข้อมูลไบต์แรกที่เกิดขึ้นหลังจากสถานะเริ่มต้นคือ ข้อมูลที่ใช้ในการอ้างถึงอุปกรณ์ที่ต้องการ ติดต่อกัน ใน 7 บิตรวมทั้งบิต MSB ด้วยจะเป็นข้อมูลแอดเดรสของอุปกรณ์สเลฟที่ต้องการติดต่อกัน โดย แบ่งเป็น



รูปที่ 2.17 รูปแบบข้อมูลมาตรฐานในการอ้างแอดเดรสของการเชื่อมต่อบนระบบบัส IC

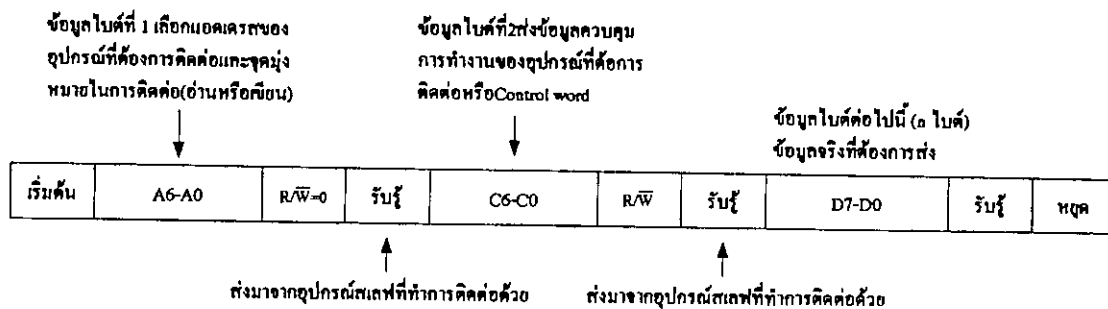
จากรูปที่ 2.17

1. บิตกำหนดแอดเดรสคงที่ (Fixed Address Bit) จำนวน 4 บิต ซึ่งข้อมูลนี้ อุปกรณ์แต่ละตัวจะถูกกำหนดจากผู้ผลิต ไม่สามารถเปลี่ยนแปลงแก้ไขได้
2. บิตกำหนดแอดเดรสที่สามารถโปรแกรมได้ (Programmable Address Bit) จำนวน 3 บิต (บิต 1 – 3) โดยกำหนดสถานะลอจิกให้แก่ขา A0 – A2 ของอุปกรณ์ที่เชื่อมต่อบัส IC
3. บิตกำหนดการอ่านหรือเขียนข้อมูลอุปกรณ์สเลฟ (เป็นบิต LSB)
 - เป็น “0” หมายถึง ต้องการเขียนข้อมูลไปยังอุปกรณ์สเลฟ
 - เป็น “1” หมายถึง ต้องการอ่านข้อมูลจากอุปกรณ์สเลฟ

- ข้อมูลใน ไบต์ต่อมาคือ ข้อมูลควบคุม (Control Byte) ข้อมูล ไบต์นี้จะขึ้นอยู่กับอุปกรณ์แต่ละประเภท เพื่อกำหนดการทำงานต่างๆ ของตัวมัน อุปกรณ์บางประเภทอาจจะไม่ต้องมีการเขียนไบต์นี้ก็ได้อีก

- ข้อมูลใน ไบต์ต่อมาคือ ข้อมูลที่ทำการส่งจริง (Data)

หลังจากที่มีการส่งข้อมูลในแต่ละ ไบต์ อุปกรณ์สเลฟที่ได้รับการติดต่อดังกล่าวจะส่งสัญญาณรับรู้ตอบกลับมาด้วยทุกครั้ง ในรูปที่ 2.18 แสดงรูปแบบข้อมูลอนุกรมที่เกิดขึ้นในการติดต่อบนบัส IC การอ้างถึงข้อมูล 7 บิต



รูปที่ 2.18 รูปแบบของข้อมูลในการอ้างแอดเดรสแบบ 7 บิตบนระบบบัส I²C

3. การอ้างถึงแบบ 10 บิต



รูปที่ 2.19 รูปแบบของข้อมูลในการอ้างแอดเดรสแบบ 10 บิตบนระบบบัส I²C

ในรูปที่ 2.19 แสดงรูปแบบข้อมูลอนุกรมของการอ้างถึงแบบ 10 บิตจะมีข้อมูลเพิ่มเติมขึ้นมาเล็กน้อย โดยในไบต์แรกหลังจากเกิดสถานะเริ่มต้น ต้องกำหนดได้ 5 บิตบนมีข้อมูลเป็น 11110 ส่วนอีก 2 บิตถัดมาเป็นบิตแอดเดรสของอุปกรณ์ที่ต้องการจะติดต่อ ในบิต LSB ของข้อมูลไบต์แรกยังคงเป็นการกำหนดว่า ต้องการอ่านหรือเขียนข้อมูลกับอุปกรณ์ตัวเลขที่ต้องการติดต่อด้วย ข้อมูลไบต์ต่อมาเป็นข้อมูลแอดเดรสในไบต์ที่ 2 ของอุปกรณ์ที่เราต้องการติดต่อด้วย ข้อมูลไบต์ถัดไปจึงเป็นข้อมูลควบคุมข้อมูลหลังจากนั้นก็จะจะเป็นข้อมูลจริงที่ใช้ในการติดต่อ

เช่นเดียวกันกับการอ้างถึงแบบ 7 บิต หลังจากส่งข้อมูลครบทุกไบต์ ต้องมีสถานะรับรู้เกิดขึ้น เพื่อให้กระบวนการส่งข้อมูลสามารถดำเนินต่อไปได้

2.4.5.1 ฟังก์ชันที่ช่วยในการเขียนโปรแกรมภาษา C เพื่อติดต่ออุปกรณ์บนระบบบัส I²C ดังนี้

1. i2c_delay ทำหน้าที่หน่วงเวลาเพื่อสร้างสถานะเงื่อนไขของระบบบัส I²C
2. i2c_clk ทำหน้าที่สร้างสัญญาณนาฬิกาที่ขา SCL เพื่อกำหนดจังหวะในการสื่อสารข้อมูลบนระบบบัส
3. i2c_start ทำหน้าที่สร้างสถานะเริ่มต้นในระบบบัส
4. i2c_stop ทำหน้าที่สร้างสถานะสิ้นสุดในระบบบัส

5. `i2c_wrddata` ทำหน้าที่ส่งข้อมูล 1 ไบต์ ไปยังอุปกรณ์สเลฟที่ต่อร่วมกับบัส โดยข้อมูลที่ ต้องการส่ง มีพารามิเตอร์ `dat` ทำหน้าที่รับค่า และคืนค่าออกมาเป็น "0" เมื่อการส่งข้อมูลไปยังอุปกรณ์ สเลฟเกิดขึ้นอย่างสมบูรณ์ แต่ถ้าไม่ จะคืนค่า "1" ออกมาแทน

6. `i2c_rddata` ทำหน้าที่อ่านข้อมูล 1 ไบต์จากอุปกรณ์สเลฟที่ต่อบนบัส โดยจะคืนค่าข้อมูล ขนาด 1 ไบต์ ออกมา

7. `i2c_ACK` ทำหน้าที่สร้างสถานะรับรู้ในการรับข้อมูลจากอุปกรณ์สเลฟ

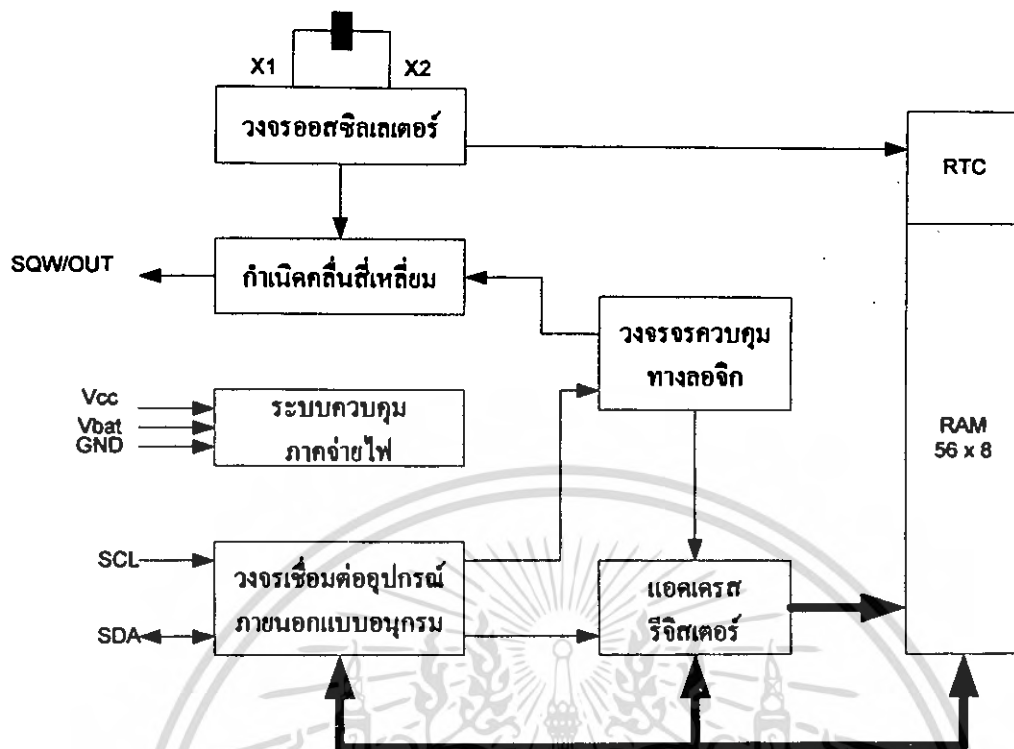
8. `i2c_NACK` ทำหน้าที่สร้างสถานะหยุดรับรู้ในการรับข้อมูลจากอุปกรณ์สเลฟ

2.4.6 การสร้างฐานเวลา

ระบบที่ใช้ไมโครคอนโทรลเลอร์ในการควบคุมบางระบบจะต้องมีเวลาเข้ามาเกี่ยวข้องกับคีย์ โดย จะมีการบอกเวลาเป็นชั่วโมง นาที และวินาที หรือมีการบอกวัน เดือน และปีให้กับระบบด้วย ดังนั้น การสร้างฐานเวลาจะต้องถูกต้องและแม่นยำ แม้ว่าระบบหยุดทำงานและเริ่มทำงานใหม่ค่าเวลาต่างๆ จะต้องถูกต้องด้วย ในปัจจุบันการทำงานประเภทนี้จะใช้ชิปไอซีที่ทำหน้าที่สร้างฐานเวลาจริงให้กับระบบ (RTC หรือ Real Time Clock) ไอซีประเภทนี้มีวงจรจัดการด้านเวลาจริงอยู่ภายใน การใช้งานเพียงต่อ คริสตอลให้กับมัน การทำงานด้านเวลาก็จะเป็นไปอย่างอัตโนมัติ เสมือนว่าตัวมันเป็นนาฬิกาและปฏิทิน ให้กับระบบ ถ้าหากเมื่อใดต้องการทราบเวลาเราสามารถอ่านจากหน่วยความจำภายในของมันได้โดยตรง

1. ไอซีฐานเวลาจริง DS1307

ชิปเบอร์ DS1307 ของบริษัท Dallas Semiconductor ซึ่งเป็นชิปที่มีความแม่นยำมาก โดยเป็น ไอซีแบบ 8 ขา การเชื่อมต่อกับไมโครคอนโทรลเลอร์จะใช้การอินเตอร์เฟสแบบอนุกรม 2 สาย หรือแบบ I²C ไอซีเบอร์นี้กินพลังงานต่ำมาก พร้อมทั้งมีปฏิทินเวลาแบบ BCD สามารถใช้ข้อมูลเกี่ยวกับเวลา เช่น วินาที นาที ชั่วโมง (ทั้งแบบ 24 ชั่วโมง / 12 ชั่วโมง พร้อมทั้งระบุค่า AM/PM) และบอกวัน เดือน ปี ได้ โดยจะมีการปรับวันที่โดยอัตโนมัติ ในแต่ละเดือนจะแสดงวันได้สูงสุดไม่เกิน 31 วัน และจะปรับวัน ต่างๆ อย่างถูกต้องเมื่อครบปี การใช้งานกับไมโครคอนโทรลเลอร์จะใช้การส่งข้อมูลและแอดเดรสของค่า ต่างๆ แบบอนุกรม โดยมีขาหนึ่งเป็นขาสัญญาณ อีกขาหนึ่งเป็นตัวกำหนดสัญญาณนาฬิกา โครงสร้าง ภายในของไอซี DS1307 แสดงได้ดังรูปที่ 2.20



รูปที่ 2.20 บล็อกไดอะแกรมภายในของไอซี DS1307

จากรูปที่ 2.20 ไอซี DS1307 ไอซีแบบ 8 ขา ขาต่างๆ ของ DS1307 เป็นดังต่อไปนี้

ขา SDA (Serial Data Input/Output) เป็นขารับส่งข้อมูลแบบอนุกรม ในการอินเตอร์เฟสจะต้องมีตัวต้านทานต่อพูลอัพภายนอกด้วย

ขา SCL (Serial Clock Input) เป็นขาอินพุตสัญญาณนาฬิกาอนุกรม เพื่อให้เกิดการซิงโครไนซ์ในการรับส่งข้อมูลแบบอนุกรม

ขา SQW/OUT เป็นขาส่งสัญญาณคลื่นสี่เหลี่ยมออกมาทางเอาต์พุต เมื่อเริ่มทำงานบิตนี้จะถูกเซตเป็นลอจิก "1" สัญญาณเอาต์พุตที่ได้รับจะมีอยู่ 4 ค่า คือ 1 Hz, 4 kHz, 8 kHz และ 32 kHz โดยเราสามารถเลือกได้ในการใช้งานจะต้องมีตัวต้านทานต่อพูลอัพภายนอกด้วย ในการใช้งานบางประเภทไม่จำเป็นต้องใช้ขานี้

ขา X1,X2 เป็นขาที่ใช้ต่อกับคริสตัลภายนอก โดยใช้ความถี่ 32.768 kHz วงจรสร้างสัญญาณนาฬิกาภายในออกแบให้ทำงานร่วมกับคริสตัลที่มีตัวประจุ 12.5 พิโกฟารัดต่อรวมอยู่ด้วย

ขา Vcc และ GND เป็นขาที่ใช้ต่อกับไฟเลี้ยง โดยทั่วไปแล้วจะต่อกับแรงดันไฟ 5 V ถ้าหากแรงดันไฟต่อลงต่ำกว่า $1.25 \times V_{BAT}$ การรักษาเวลาสำรองภายในจะทำงานต่อ โดยจะรับพลังงานจากแหล่งจ่ายไฟสำรอง

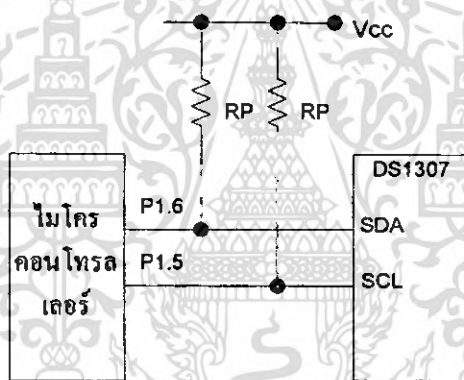
ขา V_{BAT} เป็นขาต่อกับแรงดันไฟเลี้ยงสำรอง โดยทั่วไปแล้วจะใช้แหล่งจ่ายไฟแรงดัน 3 V

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 2.22 ในรีจิสเตอร์ตำแหน่ง 00H ซึ่งเป็นรีจิสเตอร์วินาที ถ้าบิตที่ 7 ถูกเซตให้เป็นลอจิก “1” วงจรออสซิลเลเตอร์จะหยุดทำงาน ถ้าเป็น “0” วงจรออสซิลเลเตอร์จะทำงานต่อไป สำหรับรีจิสเตอร์ตำแหน่ง 01H เป็นรีจิสเตอร์นาฬิกา ซึ่งเวลาหลักสิบของวินาทีจะมีค่าตั้งแต่ 0 ถึง 5 ดังนั้นบิตที่ 7 จะไม่ใช่ สำหรับรีจิสเตอร์ตำแหน่ง 02H ซึ่งเป็นรีจิสเตอร์ชั่วโมง สามารถเซตได้ว่าจะให้เก็บข้อมูลแบบ 12 ชั่วโมง และยังบอกค่า AM/PM ได้อีกด้วย ส่วนรีจิสเตอร์ตำแหน่ง 07H จะเป็นรีจิสเตอร์ควบคุมเพื่อกำหนดว่าจะให้มีเอาต์พุตออกมาทางขา SQW/OUT หรือไม่ และเอาต์พุตที่ออกมาจะให้ความถี่เท่าใด รายละเอียดต่างๆ สามารถศึกษาได้จากคู่มือของไอซีได้โดยตรง

การจัดหน่วยความจำภายในของชิปไอซีฐานเวลาในลักษณะนี้ทำให้ผู้ใช้สามารถเลือกอ่านหรือเขียนข้อมูลใดๆ ในตำแหน่งที่กำหนดได้ เช่นถ้าหากอ่านค่าในตำแหน่ง 01H ออกมาได้เป็น 0011 0010 หรือ 32 ในระบบเลข BCD จะหมายความว่า เป็นเวลา 32 นาที ถ้าอ่านค่าในตำแหน่ง 02H ออกมาได้เป็น 0010 0010 หรือ 22 ในระบบเลข BCD หมายความว่า เป็นเวลา 22 ชั่วโมงเป็นต้น

2. การเชื่อมต่อ DS1307 กับไมโครคอนโทรลเลอร์ MCS - 51



รูปที่ 2.23 วงจรการเชื่อมต่อกับ MCS-51 อย่างง่าย

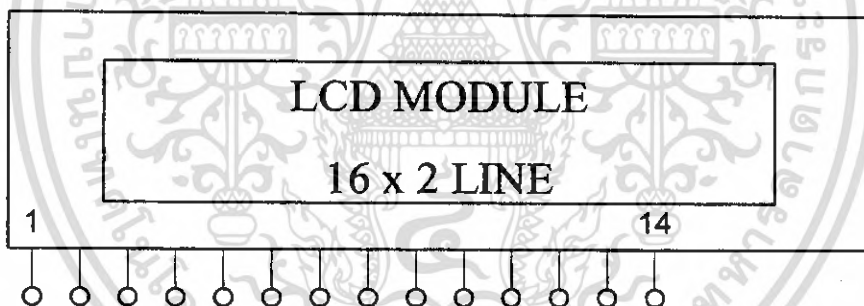
การเชื่อมต่ออย่างง่ายระหว่าง DS1307 กับไมโครคอนโทรลเลอร์แสดงไว้ดังรูปที่ 2.23 โดยใช้ P1.6 เป็นขารับส่งข้อมูลแบบอนุกรม ส่วน P1.5 ใช้ขาสัญญาณพิกาคควบคุม ตัว DS1307 สามารถรับส่งข้อมูลแบบสองทิศทางได้ และสามารถต่อกับอุปกรณ์อื่นๆ หลายตัวได้โดยมีอุปกรณ์ที่ทำหน้าที่ควบคุมการสื่อสารข้อมูล เรียกว่า มาสเตอร์ สำหรับตัวที่ถูกควบคุมเรียกว่า สเลฟ ในการทดลองนี้จะให้ไมโครคอนโทรลเลอร์เป็นตัวควบคุมการอ่านเขียนข้อมูล โดยบิต P1.5 จะเป็นตัวกำหนดสัญญาณพิกาคเพื่อควบคุมข้อมูลที่จะอ่านเขียนทาง P1.6 ที่เป็นขาข้อมูล

2.5 โครงสร้าง แอลซีดี โมดูล (LCD Module)

2.5.1 ส่วนประกอบหลักของแอลซีดี โมดูล

ใน แอลซีดี โมดูล จะมีส่วนประกอบหลักๆ 3 ส่วน ดังนี้

1. ตัวแสดงผล (Display) ภายในแอลซีดี โมดูล นี้ประกอบด้วยผลึกเหลว ดังนั้นการแสดงผลต้องใช้แสงจากภายนอก โดยจะต้องมีการปรับมุมมองเพื่อให้เห็นการแสดงผลของแอลซีดี โมดูล
2. ตัวควบคุม (Controller) เป็นตัวรับข้อมูลจากอุปกรณ์ภายนอกมาควบคุมการทำงานของแอลซีดี โมดูล เช่น ลบจอภาพ แสดงตัวอักษรหรือเลื่อนเคอร์เซอร์ เป็นต้น ตัวควบคุมนี้ใช้ชิปควบคุมโดยเฉพาะ ชิปที่นิยมใช้คือ เบอร์ HD44780 และ HD61830 โดย HD44780 จะใช้ควบคุม แอลซีดี แบบอักษร ส่วน HD44780 ใช้ควบคุมแอลซีดี แบบกราฟฟิก
3. ตัวขับ (Driver) เป็นตัวรับสัญญาณจากตัวควบคุมมาขับให้ตัวแสดงผลแสดงข้อมูลตามที่กำหนด ชิปที่ใช้ทำหน้าที่นี้ได้แก่ เบอร์ HD44100H และ MSM5259 เป็นต้น แอลซีดี โมดูล มีอยู่หลายรุ่น และคุณสมบัติแตกต่างกันไป ซึ่งแบ่งได้เป็น 2 แบบ คือ แบบ Dot matrix และ Graphic โดยแบบ Dot matrix จะแสดงผลเป็นแบบ 5x8 Dot หรือ 5x10 Dot มีตั้งแต่ 1 Line, 2 Line และ 4 Line ซึ่งการใช้งานแต่ละแบบจะใกล้เคียงกัน ลักษณะขาสัญญาณของ แอลซีดี โมดูล แบบ 1 Line ดังรูปที่ 2.24



รูปที่ 2.24 โครงสร้างแอลซีดีโมดูล

ตัวแอลซีดี โมดูล จะมีขาใช้งานทั้งหมด 14 ขาคู่กันหน้าที่ของแต่ละขามีดังนี้ คือ
ตารางที่ 2.9 หน้าที่ของขาใช้งาน แอลซีดี โมดูล

ขา 1 (GND)	เป็นกราวด์ ใช้ต่อกับระบบ กราวด์ ของไมโครคอนโทรลเลอร์
ขา 2 (VCC)	เป็นไฟเลี้ยงวงจรของ แอลซีดี มีขนาด +5 VDC
ขา 3 (Vee)	เป็นขาสำหรับปรับความเข้มของจอ แอลซีดี โดยที่เมื่อต่อกับ VCC จะมีความเข้มต่ำสุด และเมื่อต่อกับ กราวด์ จะมีความเข้มมากที่สุด โดยปกติจะต่อกับ กราวด์ เสมอเพื่อความสะดวกในการต่อ
ขา 4 (RS)	Register Select ใช้สำหรับบอก แอลซีดี ให้ทราบว่าข้อมูลที่ส่งให้มันเป็น Instruction หรือ Data โดยเมื่อนี้ เป็น "0" หมายถึง Instruction เป็น "1" หมายถึง Data
ขา 5 (R/W)	ใช้สำหรับกำหนดว่าเป็นการอ่านหรือเขียนข้อมูลให้กับ แอลซีดี โดยเมื่อนี้ เป็น "0" หมายถึงเป็นการเขียนข้อมูล เป็น "1" หมายถึงเป็นการอ่านข้อมูล
ขา 6 (E)	เป็นขา Enable เมื่อนี้ เป็น "1" ใช้สำหรับบอก แอลซีดี ว่าอุปกรณ์ภายนอกต้องการติดต่อกับ เป็น "0" ตัว แอลซีดี จะไม่สนใจในสัญญาณ RS, R/W และ (DB ₇ - DB ₀)
ขา 7 - 14 (DB ₇ - DB ₀)	เป็นขา Data Bus สำหรับอ่านหรือเขียนข้อมูลให้กับตัว แอลซีดี

2.5.2 การเชื่อมต่อ แอลซีดี โมดูล เข้ากับไมโครคอนโทรลเลอร์

การเชื่อมต่อ แอลซีดี โมดูล เข้ากับไมโครคอนโทรลเลอร์สามารถทำได้โดยตรงกับตัว MCS-51 หรือต่อผ่าน 8255 ก็ได้ ในที่นี้จะต่อโดยตรงกับตัว MCS-51 ดังนี้

- ขาสัญญาณข้อมูล D0-D7 (ขา 7-14) ต่อเข้ากับ MCS-51 พอร์ต 2
- ขา RS (ขา 14) ต่อเข้ากับ MCS-51 พอร์ต 0 บิต 0
- ขา R/W (ขา 5) ต่อเข้ากับ MCS-51 พอร์ต 0 บิต 1
- ขา E (ขา 6) ต่อเข้ากับ MCS-51 พอร์ต 0 บิต 2

2.5.3 ชุดคำสั่งของ แอลซีดี โมดูล

1. คำสั่งเคลียร์ตัวแสดงผล (CLEAR DISPLAY)

ตารางที่ 2.10 คำสั่งเคลียร์ตัวแสดงผล

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	0	1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่ง CLEAR DISPLAY เป็นคำสั่งที่ใช้เขียนข้อมูลหรือตัวอักษรว่าง (Space) ลงใน DDRAM ทั้งหมด และทำการกำหนดค่า DDRAM ADDRESS เป็น 0 และเคอร์เซอร์จะกลับไปอยู่ที่ตำแหน่งซ้ายสุดของจอแสดงผล

2. คำสั่ง CURSOR AT HOME

ตารางที่ 2.11 คำสั่ง CURSOR AT HOME

RS	R/ \bar{W}	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	0	1	*

คำสั่ง CURSOR AT HOME หรือ RETURN HOME เป็นคำสั่งที่ใช้ในการเลื่อนตำแหน่งของเคอร์เซอร์ไปอยู่ที่ตำแหน่งซ้ายสุดของจอแสดงผล โดยข้อมูลที่อยู่ใน DDRAM หรือที่หน้าจอแสดงผลจะไม่เปลี่ยนแปลง

3. คำสั่งโหมดในการป้อนข้อมูล (ENTRY MODE SET)

ตารางที่ 2.12 คำสั่งโหมดในการป้อนข้อมูล

RS	R/ \bar{W}	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	0	1	I/D	S

คำสั่งในการป้อนข้อมูล (ENTRY MODE SET) ใช้สำหรับกำหนดการเลื่อนของเคอร์เซอร์และตำแหน่งแอดเดรสของ DDRAM ดังนี้

I/D เป็นบิตที่ใช้ในการกำหนดการเลื่อนเคอร์เซอร์และตำแหน่งแอดเดรสของ DDRAM ว่า จะให้เพิ่มหรือลดลงเมื่อเขียนหรืออ่านข้อมูล

บิต I/D = 0 แอดเดรสของ DDRAM จะลดลง

บิต I/D = 1 แอดเดรสของ DDRAM จะเพิ่มขึ้นส่วนเคอร์เซอร์จะเลื่อนตามตำแหน่งแอดเดรสของ DDRAM

S เป็นบิตที่ใช้กำหนดลักษณะของการแสดงผลเมื่อมีการเขียนข้อมูล

บิต S = 1 เมื่อเขียนข้อมูลใหม่ลงไปแล้วตัวเคอร์เซอร์จะอยู่กับที่ แต่ตัวอักษรข้อมูลเดิมจะถูกผลักไปทางซ้าย

บิต S = 0 เมื่อเขียนข้อมูลใหม่ลงไปแล้วตัวเคอร์เซอร์จะเลื่อนไปทางขวา

4. คำสั่งควบคุมการแสดงผล (DISPLAY ON/OFF)

ตารางที่ 2.13 คำสั่งควบคุมการแสดงผล

RS	R/ \bar{W}	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	0	1	D	C	B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำสั่งควบคุมการแสดงผล เป็นคำสั่งที่ใช้ในการเปิดปิดจอแสดงผลและเคอร์เซอร์มีลักษณะดังนี้

- D = 0 กำหนดให้ปิดจอแสดงผล (Display OFF)
- D = 1 กำหนดให้เปิดจอแสดงผล (Display ON)
- C = 0 กำหนดให้ปิดเคอร์เซอร์ (Cursor OFF)
- C = 1 กำหนดให้เปิดเคอร์เซอร์ (Cursor ON)
- B = 0 กำหนดให้ไม่มีการกระพริบที่เคอร์เซอร์
- B = 1 กำหนดให้มีการกระพริบที่เคอร์เซอร์ (กระพริบเป็นรูปสี่เหลี่ยมทึบ)

5. คำสั่งควบคุมการเลื่อนเคอร์เซอร์และตัวอักษร (DISPLAY SHIFT)

ตารางที่ 2.14 คำสั่งควบคุมการเลื่อนเคอร์เซอร์และตัวอักษร

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	0	1	S/C	R/L	*	*

คำสั่งควบคุมการเลื่อนเคอร์เซอร์และตัวอักษร เป็นการควบคุมการเลื่อนเคอร์เซอร์และตัวอักษรบนจอแสดงผล โดยขึ้นอยู่กับกำหนัดบิต S/C และ R/L โดยมีลักษณะดังนี้

ตารางที่ 2.15 การกำหนดบิต S/C และ R/L

S/C	R/L	ลักษณะการเลื่อน
0	0	เลื่อนเคอร์เซอร์ไปทางซ้าย
0	1	เลื่อนเคอร์เซอร์ไปทางขวา
1	0	เลื่อนตัวอักษรตัวใหม่ไปทางซ้าย
1	1	เลื่อนตัวอักษรตัวใหม่ไปทางขวา

6. คำสั่งการกำหนดฟังก์ชันการทำงาน (FUNCTION SET)

ตารางที่ 2.16 คำสั่งการกำหนดฟังก์ชันการทำงาน

RS	R/W	DB7	DB6	DB5	DB4	DB3	DB2	DB1	DB0
0	0	0	0	1	DL	N	F	*	*

- DL = 0 กำหนดให้ติดต่อกับ แอลซีดี โมดูลเป็นแบบ 4 บิต
- DL = 1 กำหนดติดต่อกับ แอลซีดี โมดูลเป็นแบบ 8 บิต
- N = 0 กำหนดการแสดงผลแบบ 1 บรรทัด
- N = 1 กำหนดการแสดงผลตั้งแต่ 2 บรรทัดขึ้นไป
- F = 0 กำหนดความละเอียดของการแสดงผลเป็น 5 x 7 Dot
- F = 1 กำหนดความละเอียดของการแสดงผลเป็น 5 x 10 Dot

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์หรือการสงวนสิทธิ์ในชื่อการค้าเท่านั้น มิอนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.5.4 ฟังก์ชันการเขียนโปรแกรมภาษา C เพื่อติดต่อกับ แอลซีดี โมดูล

- `lcd_init`

เป็นฟังก์ชันอินิเชียลแอลซีดี โมดูลกำหนดการติดต่อกับแอลซีดี โมดูลแบบ 4 บิต แสดงผล 2 บรรทัด ที่ความละเอียด 5 x 7 จุด ควรเรียกใช้ฟังก์ชันนี้ในตอนต้นของ โปรแกรมหลักก่อนใช้งานฟังก์ชันใดๆ

รูปแบบ

```
void lcd_init ()
```

การคืนค่า

ไม่มีการคืนค่า

- `lcd_delay` และ `delay_ms`

เป็นฟังก์ชันหน่วงเวลาสำหรับสร้างสัญญาณพัลส์เอ็นเอเบิลป้อนให้กับขา E ของแอลซีดี โมดูล

รูปแบบ

```
void lcd_delay (unsigned int tick)
```

พารามิเตอร์

tick ใช้กำหนดค่าหน่วงเวลาในหน่วยมิลลิวินาที กำหนดค่าได้ตั้งแต่ 0 ถึง 65,535

การคืนค่า

ไม่มีการคืนค่า

- `lcd_command`

เป็นฟังก์ชันส่งคำสั่งควบคุมการแสดงผลที่แอลซีดี โมดูล

รูปแบบ

```
void lcd_command (unsigned char com)
```

พารามิเตอร์

com สำหรับกำหนดรหัสคำสั่งควบคุมการแสดงผลที่แอลซีดี โมดูล

การคืนค่า

ไม่มีการคืนค่า

- `lcd_text`

เป็นฟังก์ชันส่งข้อมูลอักขระแสดงผลที่แอลซีดี โมดูล

รูปแบบ

```
void lcd_text (unsigned char text)
```

พารามิเตอร์

text สำหรับกำหนดข้อมูลอักขระที่ต้องการแสดงผลที่แอลซีดี โมดูล

การคืนค่า

ไม่มีการคืนค่า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- lcd_puts

เป็นฟังก์ชันทำหน้าที่ส่งสายอักขระแสดงผลที่แอลซีดี โมดูล
รูปแบบ

```
void lcd_puts (unsigned char line , char *p)
```

พารามิเตอร์

line ใช้กำหนดแอดเดรสแรกของสายอักขระที่จะเริ่มการแสดงผลที่แอลซีดี โมดูล

p ใช้กำหนดการเข้าถึงแอดเดรสของสายอักขระที่จะส่งไปแสดงยังแอลซีดี โมดูล
การคืนค่า

ไม่มีการคืนค่า

- inttolcd

เป็นฟังก์ชันทำหน้าที่แสดงค่าข้อมูลตัวเลขจำนวนเต็มที่แอลซีดี โมดูล
รูปแบบ

```
void inttolcd (unsigned char addr , unsigned int value , unsigned char base)
```

พารามิเตอร์

addr สำหรับกำหนดแอดเดรสแรกของสายอักขระที่จะเริ่มการแสดงผลที่ แอลซีดี โมดูล

value ใช้กำหนดจำนวนเต็มที่ต้องการนำไปแสดงที่แอลซีดี โมดูล โดยค่าที่แสดงทำได้เฉพาะค่า
บวก ตั้งแต่ 0 ถึง +65,535

base ใช้กำหนดเลขฐานที่ต้องการแสดงที่แอลซีดี โมดูล ควรกำหนดเป็น 10 (สำหรับแสดงใน
แบบเลขฐานสิบ) หรือ 16 (สำหรับแสดงในแบบเลขฐานสิบหก)

การคืนค่า

ไม่มีการคืนค่า

2.6 การใช้งานคีย์แพด (Keypad)

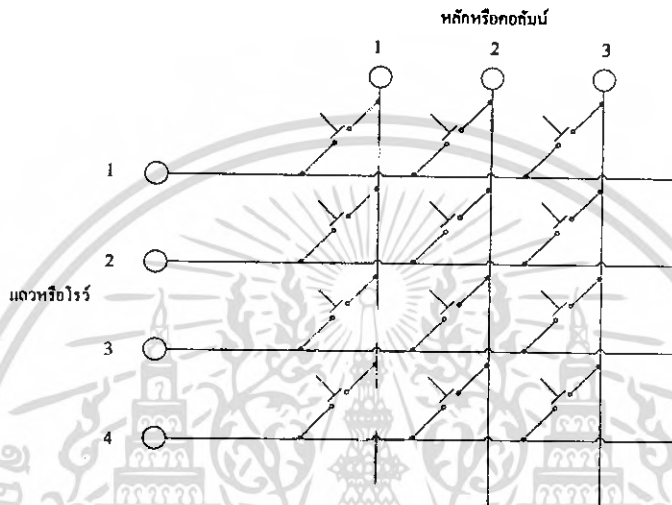
วิธีการอ่านค่าหรือรับค่าจากสวิตช์นั้นมีด้วยกัน 2 ลักษณะใหญ่ๆคือ แบบต่อเข้ากับไฟเลี้ยงหรือ
กราวด์โดยตรง และต่อแบบวงจรเมทริกซ์ (Matrix Switch) ในรูปที่ 2.25 จะเห็นว่าสวิตช์จะถูกต่อใน
แนวแกนตั้งและแนวแกนนอน จะเรียกแนวแกนตั้งว่าหลักหรือคอลัมน์ (Column) และเรียกแนวแกนนอน
ว่าแถวหรือโรว์ (Row) ดังนั้นค่าของสวิตช์ในแต่ละตำแหน่งจะต้องประกอบด้วยหลักและแถว

กระบวนการในการอ่านค่าของสวิตช์แบบนี้จะใช้วิธีการเขียนโปรแกรมในไมโครคอนโทรลเลอร์
โดยจะต้องใช้สายทั้งหมด 7 เส้น ซึ่งเป็นสายของหลัก 3 เส้นและเป็นสายของแถว 4 เส้น สำหรับ
สวิตช์แบบ 4 x 3 ต่อเข้ากับไมโครคอนโทรลเลอร์จะทำการส่งข้อมูล "0" ไปยังพอร์ทที่ต่อกับด้านแถวทีละ
เส้นตามลำดับ ในทุกครั้งที่มีการส่งข้อมูลไปยังสายด้านแถวของคีย์แพดไมโครคอนโทรลเลอร์จะทำการ
อ่านค่าที่ด้านหลักเข้ามาด้วย หากไม่มีการกดค่าด้านหลักก็จะเป็น "1" ทั้งหมด ถ้าหากมีการกดคีย์ค่าของ
ด้านหลักก็จะเป็น "111" อีกต่อไป เป็นการแจ้งให้ทราบว่ามีอาการคีย์แพดเกิดขึ้นแล้ว จากนั้น
ไมโครคอนโทรลเลอร์ก็จะทำการค้นหาตำแหน่งต่อไป โดยการค้นหาตำแหน่งนั้นสิ่งที่ได้มาอย่างแรกคือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ค่าตำแหน่งของคีย์นั้น จากนั้นก็จะนำค่าเหล่านั้นไปเปิดตารางข้อมูล เพื่อที่จะได้ค่าที่ต้องการนำไปแสดงผลที่แท้จริง

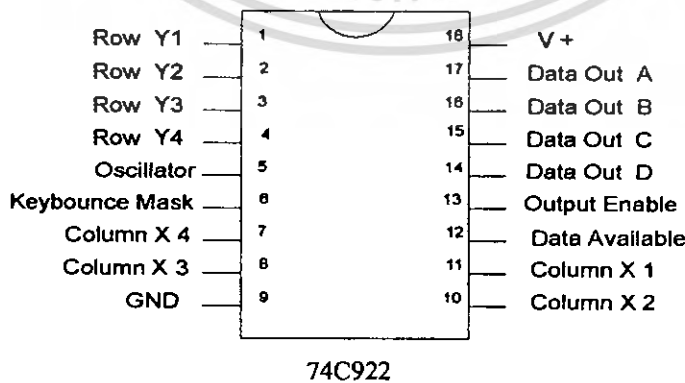
ข้อดีของสวิตช์แบบนี้ก็คือสามารถรองรับการเพิ่มของสวิตช์ได้อย่างสะดวก เพียงแค่แก้ไขซอฟต์แวร์เพียงเล็กน้อยเท่านั้น ทำให้วงจรสวิตช์แบบแมทริกซ์นี้เป็นที่นิยมกันมากในระบบควบคุมอัตโนมัติหรือกึ่งอัตโนมัติ โดยในการใช้งานทั่วไปจะเรียกสวิตช์แบบแมทริกซ์นี้ว่า “คีย์แพด” ดังแสดงในรูปที่ 2.25



รูปที่ 2.25 วงจรสวิตช์แบบแมทริกซ์หรือคีย์แพด

2.6.1 การใช้ไอซีสแกนคีย์สวิตช์

ในปัจจุบันได้มีการออกแบบ ไอซีสแกนคีย์สวิตช์แบบแมทริกซ์ โดยเอาต์พุตที่ออกจากไอซีจะเป็นเลขไบนารี จากนั้นนำไปต่อกับพอร์ทของระบบไมโครคอนโทรลเลอร์ทำให้ระบบไมโครคอนโทรลเลอร์ไม่ต้องเสียเวลาในการทำโปรแกรมสแกนคีย์สวิตช์ ในที่นี้ใช้ไอซีเบอร์ 74C922

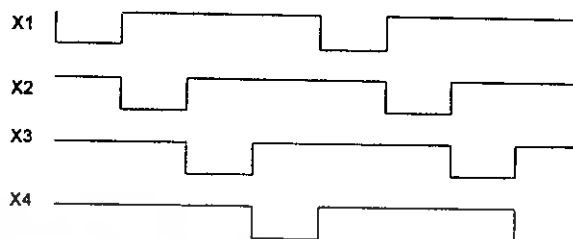


16 – Key Encoder

รูปที่ 2.26 แสดง ไอซีสแกนคีย์สวิตช์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Switch	Data Output			
	D	C	B	A
Y1, X1	0	0	0	0
Y1, X2	0	0	0	1
Y1, X3	0	0	1	0
Y1, X4	0	0	1	1
Y2, X1	0	1	0	0
Y2, X2	0	1	0	1
Y2, X3	0	1	1	0
Y2, X4	0	1	1	1
Y3, X1	1	0	0	0
Y3, X2	1	0	0	1
Y3, X3	1	0	1	0
Y3, X4	1	0	1	1
Y4, X1	1	1	0	0
Y4, X2	1	1	0	1
Y4, X3	1	1	1	0
Y4, X4	1	1	1	1



รูปที่ 2.26 แสดงไอซีสแกนกีบสวิทช์ (ต่อ)

2.7 มาตรฐานพอร์ทอนุกรมแบบ RS – 232

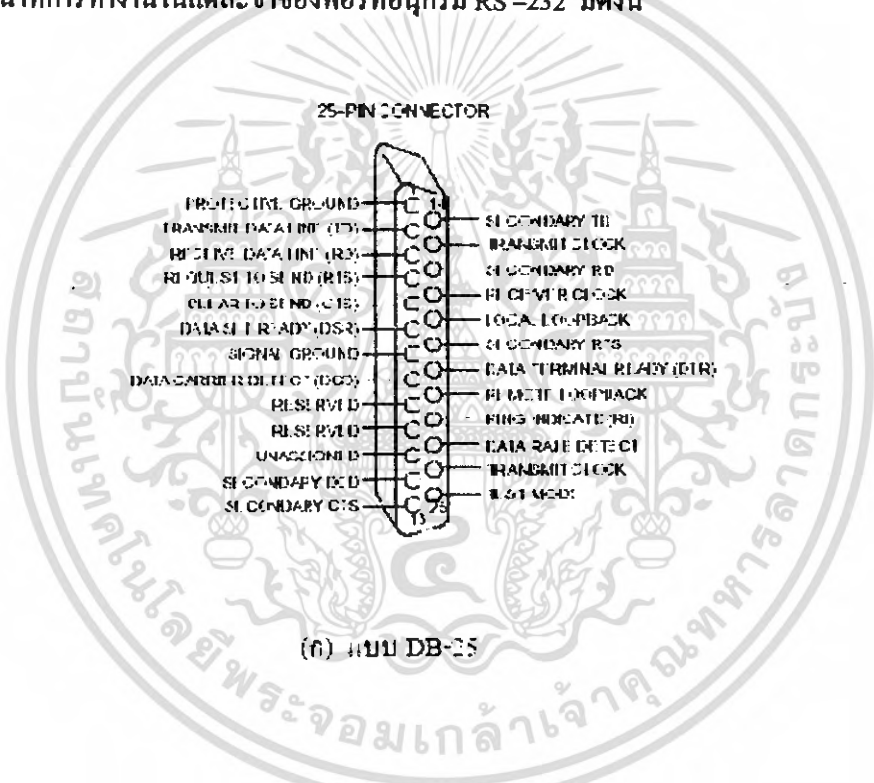
มาตรฐานการเชื่อมต่อแบบอนุกรม RS –232 เป็นมาตรฐานอุตสาหกรรมที่ออกแบบมาเพื่อใช้ในการส่งข้อมูลอนุกรมแบบอะซิงโครนัส 2 ทิศทาง ด้วยมาตรฐาน RS – 232 ในอดีตนั้นถูกออกแบบมาเพื่อการส่งผ่านข้อมูลจากคอมพิวเตอร์ ไปยังโมเด็มเพียงอย่างเดียว เพื่อที่จะนำข้อมูลจากโมเด็มนี้สื่อสารผ่านสายโทรศัพท์ไปยังคอมพิวเตอร์ อีกชุดหนึ่งซึ่งอยู่ห่างไกลกัน โดยคณะกรรมการที่เรียกว่า สมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronic Industries Association : EIA) ได้วางมาตรฐานที่มีชื่อเรียกว่า EIA RS-232 มาตรฐานนี้ในช่วงแรกจะใช้คอนเน็คเตอร์ เป็นแบบ DB – 25 โดยกำหนดความยาวสูงสุดของสายสัญญาณไว้ที่ 50 ฟุต มีระดับสัญญาณตั้งแต่ -3 V ถึง -12V แสดงว่ามีข้อมูล (Mark) และ +3 ถึง +12 V แสดงว่าเป็นช่องว่าง (Space) มาตรฐาน RS-232 ได้กำหนดรูปแบบของอุปกรณ์ เชื่อมต่อข้อมูล (Data Terminal Equipment : DTE) กับวงจรข้อมูลปลายทาง (Data Circuit Terminating :DCE) ไว้ว่า อุปกรณ์ DTE จะต้องเป็นอุปกรณ์ที่มีการประมวลผลในตัว เช่น ไมโครคอนโทรลเลอร์ หรือ ไมโครคอมพิวเตอร์ซึ่งมีความสามารถในการสร้างบิตข้อมูลแบบอนุกรมได้ ส่วนอุปกรณ์ DCE จะทำหน้าที่เป็นเพียงตัวรับข้อมูลที่ส่งมาจาก DTE เท่านั้น โดยการรับส่งข้อมูลระหว่างอุปกรณ์ทั้งสองจะผ่านมาตรฐาน RS –232 ข้อแตกต่างของอุปกรณ์ DTE และอุปกรณ์ DCE อย่างหนึ่งให้เห็นได้ชัดคือคอนเน็คเตอร์ของ DTE เป็นตัวผู้ส่วน คอนเน็คเตอร์ของ DCE จะเป็นตัวเมียซึ่งพอร์ทอนุกรมของคอมพิวเตอร์ ที่ใช้กันอยู่ทั่วไปจะเป็นแบบ DTE ส่วนคอนเน็คเตอร์ ที่อยู่ที่โมเด็มจะเป็นแบบ DCE สำหรับ การใช้งานบนคอมพิวเตอร์ พอร์ทอนุกรม RS – 232 มักถูกใช้เชื่อมต่อกับโมเด็มหรือเมาส์ โดยสามารถรับส่งข้อมูลได้ที่มีความยาวของสายสัญญาณสูงสุดถึง 20 เมตร

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

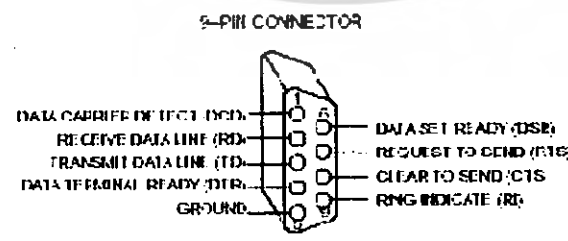
2.7.1 กอนเน็คเตอร์สำหรับพอร์ท RS – 232 และการเชื่อมต่อ

มาตรฐานการเชื่อมต่อแบบ RS –232 จะใช้คอนเน็คเตอร์แบบ DB-25 ตัวผู้ หรือ DB-9 ตัวผู้ คอนเน็คเตอร์ DB-25 จะมีขาต่อใช้งานเพียง 9 เส้นเช่นเดียวกับคอนเน็คเตอร์แบบ DB-9 เนื่องจากขาอื่น ๆ ที่เคยใช้งานในอดีตเพราะปัจจุบันมีการใช้งานไม่มากนัก จึงถูกยกเลิกไปโดยแสดงรูปร่างและตำแหน่งขาในรูปที่ 2.27

สำหรับการเชื่อมต่อคอมพิวเตอร์กับอุปกรณ์ ภายนอกที่แสดงในรูปที่ 2.28 ถูกสรในรูปแสดงถึงทิศทางในข้อมูลในรูปที่ 2.28 (ก) เป็นการเชื่อมต่อแบบ Null Modem หรือการเชื่อมต่อแบบโดยตรง โดยไม่ต้องผ่านโมเด็ม ส่วนในรูป 2.28 (ข) เป็นการเชื่อมต่อแบบ Null Modem ในลักษณะที่ใช้สายสัญญาณเพียง 3 เส้นโดยเส้นหนึ่ง สำหรับส่งข้อมูล อีกเส้นสำหรับรับข้อมูลและเส้นสุดท้ายเป็นกราวด์ สำหรับรายละเอียดหน้าที่การทำงานในแต่ละขาของพอร์ทอนุกรม RS –232 มีดังนี้



(ก) แบบ DB-25



(ข) แบบ DB-9

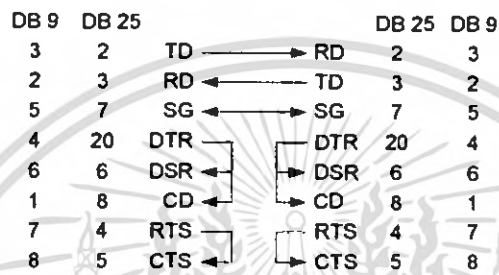
รูปที่ 2.27 แสดงการจัดขาของคอนเน็คเตอร์พอร์ทอนุกรมตามมาตรฐาน RS-232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

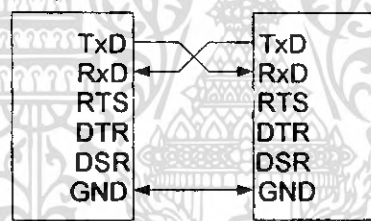
Data Carrier Detect: DCD หรืออาจเรียกว่า Carrier Detect: CD ขานี้จะแอกทีฟเมื่อมีการส่งสัญญาณพาห์จากอุปกรณ์สื่อสารข้อมูลเช่น โมเด็มสำหรับการใช้งานปกติ ขานี้จะไม่ได้ถูกใช้งานมากนัก

Receive Data: RD หรือ RxD ขานี้ใช้เพื่อรับสัญญาณอนุกรมเข้ามายังคอมพิวเตอร์โดยนำข้อมูลทีอ่านได้เก็บไว้ในรีจิสเตอร์ บัฟเฟอร์

Transmitted Data: TD หรือ TxD ขานี้ ใช้ เพื่อส่งข้อมูลออกจากคอมพิวเตอร์ โดยนำข้อมูลที่เก็บอยู่ในบัฟเฟอร์สำหรับส่งข้อมูลออกไป



(ก) การเชื่อมต่อแบบ Null Modem



คอมพิวเตอร์ อุปกรณ์ภายนอก

(ข) การต่ออุปกรณ์ภายนอกเข้ากับคอมพิวเตอร์แบบ RS 232 โดยใช้สายสัญญาณเพียง 3 เส้น

รูปที่ 2.28 แสดงการต่ออุปกรณ์ภายนอกกับพอร์ตอนุกรมของคอมพิวเตอร์ในลักษณะต่าง ๆ

Data Terminal Ready : DTR เป็นขาสัญญาณที่ส่งออกจากคอมพิวเตอร์ให้อุปกรณ์ปลายทางรับรู้ ว่า ต้องการติดต่อด้วย โดยขา DTR นี้จะต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทางและขา DTR ของอุปกรณ์ปลายทางจะต้องเชื่อมต่อกับขา DSR ของคอมพิวเตอร์ ถ้าใช้การเชื่อมต่อแบบ Null Modem ซึ่งใช้สายในการเชื่อมต่อเพียง 3 เส้น จะต้องต่อขา DTR และ DSR ของตัวมันเองเข้าด้วยกันและต้องต่อกับขา DCD ด้วยในกรณีที่ใช้โปรแกรมสื่อสารที่ใช้ในการตรวจสอบจับสัญญาณพาห์ Signal Ground : GND ขากราวด์ของระบบ

Data Set Ready: DSR ขานี้จะใช้คู่กับขา DTR เพื่อตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง ซึ่งขา DSR นี้จะเป็นขาสำหรับรับข้อมูลจากภายนอกซึ่งถูกส่งมาจากขา DTR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Request to Send: RTS เป็นขาสำหรับส่งสัญญาณร้องขอให้ทางอุปกรณ์ปลายทางส่งข้อมูลกลับมายังคอมพิวเตอร์ โดยขาที่รับสัญญาณ RTS ก็คือขา CTS ในกรณีที่ใช้การเชื่อมต่อแบบ Null Modem 3 สาย จะต้องเชื่อมต่อ RTS และ CTS ของตัวมันเองเข้าด้วยกันเพื่อจะให้การรับ และส่งข้อมูลสามารถเกิดขึ้นได้ตลอดเวลา Clear To Send: CTS ขานี้จะคอยรับสัญญาณจากขา RTS เมื่อรับสัญญาณได้ ข้อมูลที่ขา TxD จะถูกส่งออกไป ดังนั้นขานี้จึงถูกใช้เพื่อตรวจสอบอุปกรณ์ต่อพ่วงว่าพร้อมที่จะรับข้อมูลหรือไม่

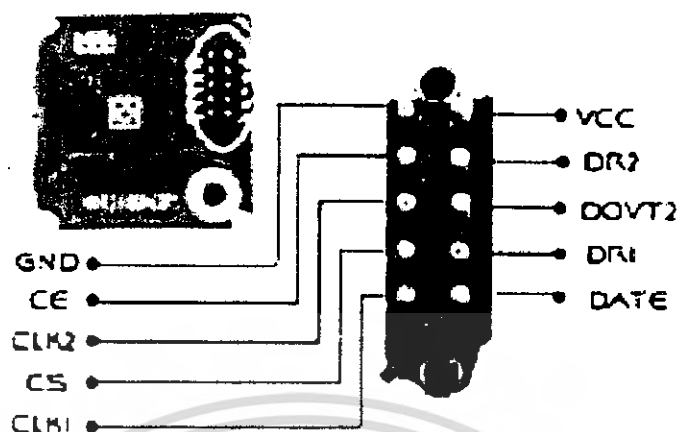
Ring Indicator: RI ใช้แสดงสถานะสัญญาณเรียกจากสายโทรศัพท์ ปกติในการสื่อสารโดยทั่วไป สายนี้จะไม่ถูกใช้งาน จะใช้งานก็ต่อเมื่อมีการเชื่อมต่อกับ โมเด็มและโปรแกรมมีการตรวจสอบสัญญาณนี้เท่านั้น

2.8 โมดูลความถี่วิทยุ 2.4 GHz

เป็นโมดูลสำเร็จรูปที่ใช้รับ - ส่ง ข้อมูล ในแบบอนุกรม ใช้กับความถี่ 2.4 GHz ปรับแต่งสำเร็จรูป พร้อมมีสายอากาศในตัว ซึ่งสามารถใช้งานได้ในระยะไกล 280 m (ความเร็วข้อมูล 250 Kbps) หรือระยะ 150 m (ความเร็ว 1 Mbps) ในพื้นที่โล่งแจ้ง

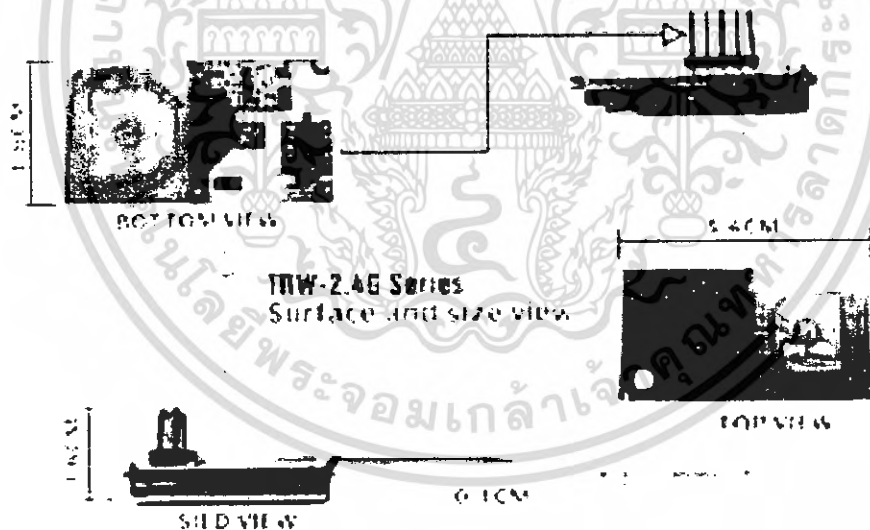
- ความถี่ในการใช้งาน 2.4 ~ 2.524 GHz
- มีรูปแบบและความเร็วในการรับ - ส่งข้อมูลโดยส่งแบบGFSK(Gaussian Frequency Shift Keying)
- ทำงานที่ความต่างศักย์ทางไฟฟ้า 3 V
- กำลังงานเอาต์พุต + 4 dBm
- ความเร็วในการรับส่งข้อมูลถึง 1 Mbps, 250 Kbps
- ขนาด 20.0 * 36.7 * 2.4 mm
- ทำงานที่อุณหภูมิ -40 ~ +85 Centigrade
- ระยะทางในการรับส่งสัญญาณ 280 m (250 Kbps), 150 m (1 Mbps)
- มีสายอากาศรับส่งสัญญาณในตัวโมดูลความถี่วิทยุ

Wiring Diagram



รูปที่ 2.29 แสดงลักษณะของโมดูลความถี่วิทยุ (TRW – 2.4 GHz)

Surface AND Size View



รูปที่ 2.30 แสดงรายละเอียดทางด้านบน, ด้านข้างและด้านหน้าของตัวโมดูลความถี่วิทยุ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.81 การจัดขาของโมดูลความถี่วิทยุ (TRW - 2.4 GHz)

ตารางที่ 2.17 แสดงลักษณะการทำงานของขาของตัวโมดูลความถี่วิทยุ

ขา	ชื่อ	ลักษณะการทำงานของขา	รายละเอียด
1	GND	Power	Ground (0 V)
2	CE	Input	Chip Enable activates Rx or Tx mode
3	CLK2	I/O	Clock output/Input for Rx data channel 2
4	CS	Input	Chip Select activates Configuration mode
5	CLK1	I/O	Clock Input (Tx) & I/O(Rx) for data channel 1 3 - wire interface
6	DATA	I/O	Rx data channel 1/Tx data input /3 – wire interface
7	DR1	Output	Rx data ready at data channel 1 (Shock Burst Only)
8	DOUT2	Output	Rx data channel 2
9	DR2	Output	Rx data ready at data channel 2 (Shock Burst only)
10	VCC	Power	Power Supply (+ 3V DC)

การเซตค่าของแต่ละขาในโหมดต่างๆสามารถดูได้จากตารางที่ 2.18

ตารางที่ 2.18 การเซตค่าของแต่ละขาในโหมดต่างๆ

โหมด	PWR_UP	CE	CS
Active (RX/TX)	1	1	0
Configuration	1	0	1
Stand by	1	0	0
Power down	0	X	X

2.8.2 โหมดการทำงานของ (TRW - 2.4 GHz)

โหมดในการใช้งานของ (TRW - 2.4 GHz) มีอยู่ 2 โหมด

1. Shock Burst Mode
2. Direct Mode

โดยในโครงงานนี้ได้กำหนดให้โมดูลความถี่วิทยุมีการทำงานในโหมด Shock Burst ซึ่งมีรายละเอียดดังต่อไปนี้

โหมด Shock Burst

โหมด Shock Burst เป็นการใช้เทคโนโลยีรับ/ส่งข้อมูลบนชิปแบบเข้าก่อน – ออกก่อน (First in – First out) โดยในการส่งข้อมูลมีทั้งระดับอัตราในการส่งบิตข้อมูลมีทั้งความเร็วต่ำและระดับความเร็วสูง เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อโมดูลความถี่วิทยุทำงานในโหมด Shock Burst สามารถเพิ่มการเข้าถึงระดับข้อมูลได้สูง (1 Mbps) โดยใช้ย่านความถี่ 2.4 GHz และต้องใช้ไมโครคอนโทรลเลอร์ความเร็วสูงในการประมวลผล โดยการจัดการกระบวนการประมวลผลให้เหมาะสมกับโปรโตคอลบนชิปจะทำให้ได้รับประโยชน์จากโมดูลความถี่วิทยุที่ตามมาดังนี้

- ประหยัดกระแส
- ระบบมีราคาต่ำ (เนื่องจากไมโครคอนโทรลเลอร์มีราคาถูก)
- ลดการชนกันของข้อมูลเมื่อใช้เวลาในการส่งระยะสั้นๆ

รายละเอียดทางไฟฟ้า

ตารางที่ 2.19 รายละเอียดทางไฟฟ้า

ชื่อ	ความหมาย	ค่าต่ำสุด	ค่าสูงสุด	หน่วย
เงื่อนไข				
VDD	แรงดัน	1.9	3.6	V
TEMP	อุณหภูมิ	-40	85	C
อินพุตดิจิทัล				
V _{IH}	ระดับแรงดันอินพุต HIGH	VDD - 0.3	VDD	V
V _{IL}	ระดับแรงดันอินพุต LOW	V _{SS}	0.3	V
เอาต์พุตดิจิทัล				
V _{OH}	ระดับแรงดันเอาต์พุต HIGH (I _{OH} =0.5 mA)	VDD - 0.3	VDD	V
V _{OL}	ระดับแรงดันเอาต์พุต LOW (I _{OH} =0.5 mA)	V _{SS}	0.3	V
เงื่อนไขต่างๆเกี่ยวกับความถี่				
f _{op}	ช่วงความถี่วิทยุ	2400	2524	MHz
R _{GFSK}	อัตราการส่งข้อมูลในโหมด Shock Burst	> 0	1000	kbps
กระแสในการใช้งานที่อัตราส่งต่างกันของ 1 ช่องสัญญาณ				
I _{VDD}	250 kbps		18	mA
I _{VDD}	1000 kbps		19	mA

หลักการทำงานในโหมด Shock Burst

เมื่อทำการกำหนดค่าให้โมดูลทำงานในโหมด Shock Burst แล้ว การทำงานของโมดูลในการรับ/ส่ง ข้อมูลมีหลักการทำงานดังนี้

การส่งข้อมูลในโหมด Shock Burst

โดยการเชื่อมต่อไมโครคอนโทรลเลอร์กับขา CE, CLK1, DATA ของตัวโมดูล

- เมื่อไมโครคอนโทรลเลอร์ต้องการส่งข้อมูลให้กับโมดูลต้องทำการเซตขา CE ให้อยู่ในสถานะ "High" เพื่อกระตุ้นให้โมดูลทำการประมวลผลข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมออนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ทำการเซตไมโครคอนโทรลเลอร์ให้อยู่ในสถานะ “Low” เพื่อกระตุ้นให้ไมโครทำการส่งข้อมูล

การรับข้อมูลในโหมด Shock Burst

โดยไมโครคอนโทรลเลอร์ทำการเชื่อมต่อกับขา CE, CLK1, DR1 และ DATA (กรณีที่ใช้ช่องสัญญาณเพียงช่องเดียว)

- เมื่อ RF Package มีแอดเดรสที่ถูกต้องและขนาดของข้อมูลที่เข้ามา ตัวไมโครจะทำการเซตค่าให้ขา CE อยู่ในสถานะ “High”
- เมื่อข้อมูลที่รับเข้ามาถูกต้อง (แอดเดรสและ CRC ถูกต้อง) ไมโครจะทำการย้าย Preamble, Address, และ CRC โดยแจ้งไปยังไมโครคอนโทรลเลอร์ให้ทำการเซตค่าให้ขา DR1 อยู่ในสถานะ “High” และเซตค่าให้ขา CE อยู่ในสถานะ “Low” เพื่อบอกว่าขณะนี้ทำการรับข้อมูลอยู่
- ไมโครคอนโทรลเลอร์จะทำการเซตค่าขาต่างๆเพื่อรับข้อมูลได้เหมาะสมและเมื่อทำการรับข้อมูล เสร็จเรียบร้อยแล้วจะทำการเซตค่าให้ขา DR1 ให้อยู่ในสถานะ “Low” เพื่อเตรียมพร้อมที่จะรับข้อมูลที่เข้ามาใหม่

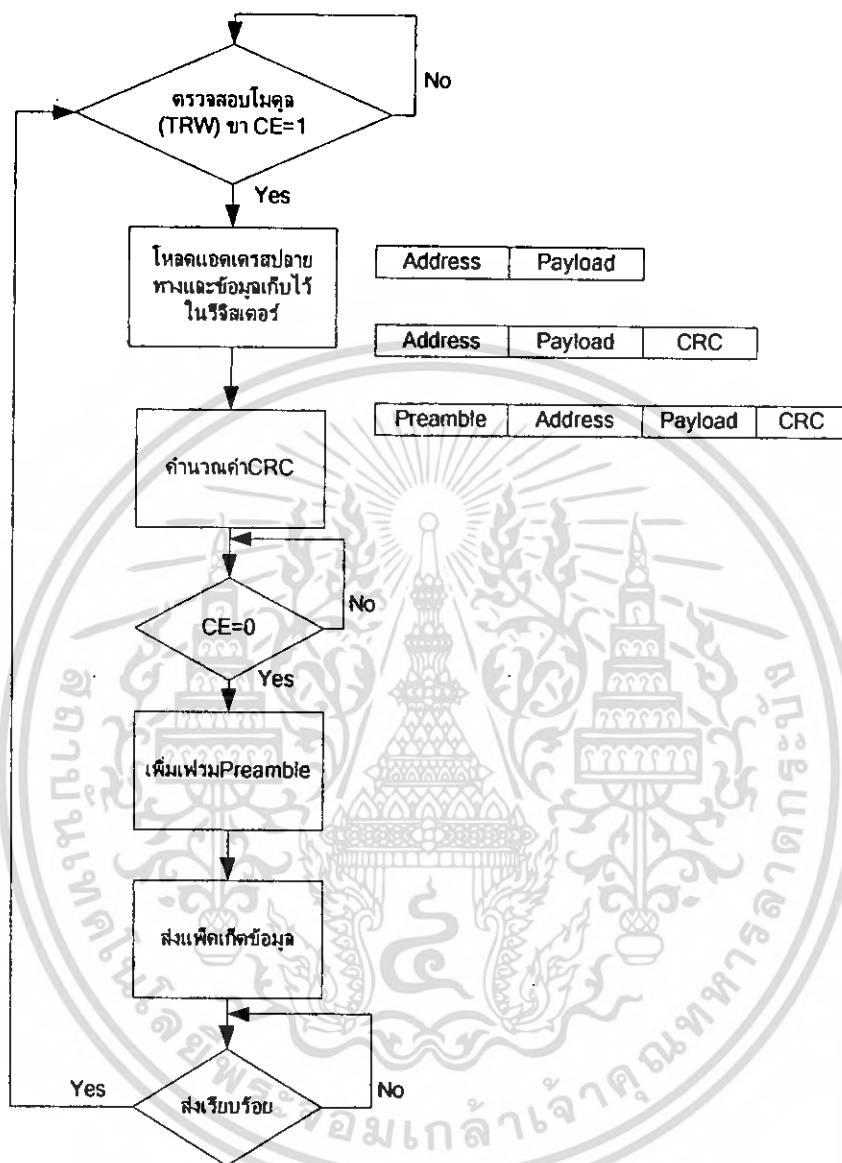
2.8.3 ส่วนประกอบของชุดข้อมูล

Preamble	Address	Payload	CRC
----------	---------	---------	-----

ส่วนประกอบของชุดข้อมูลแบ่งได้เป็น 4 ส่วน คือ

- Preamble เป็นส่วนแรกของชุดข้อมูล เพื่อแสดงจุดเริ่มต้นของเฟรมข้อมูล
- Address เป็นส่วนที่ระบุแอดเดรสของตัวรับข้อมูล
- Payload เป็นส่วนที่เก็บข้อมูล
- CRC เป็นส่วนที่ใช้ตรวจสอบความผิดพลาดของชุดข้อมูล

หลักการทํางานของโมดูลความถี่วิทยุ (TRW - 2.4 GHz) ในการส่งและรับข้อมูล

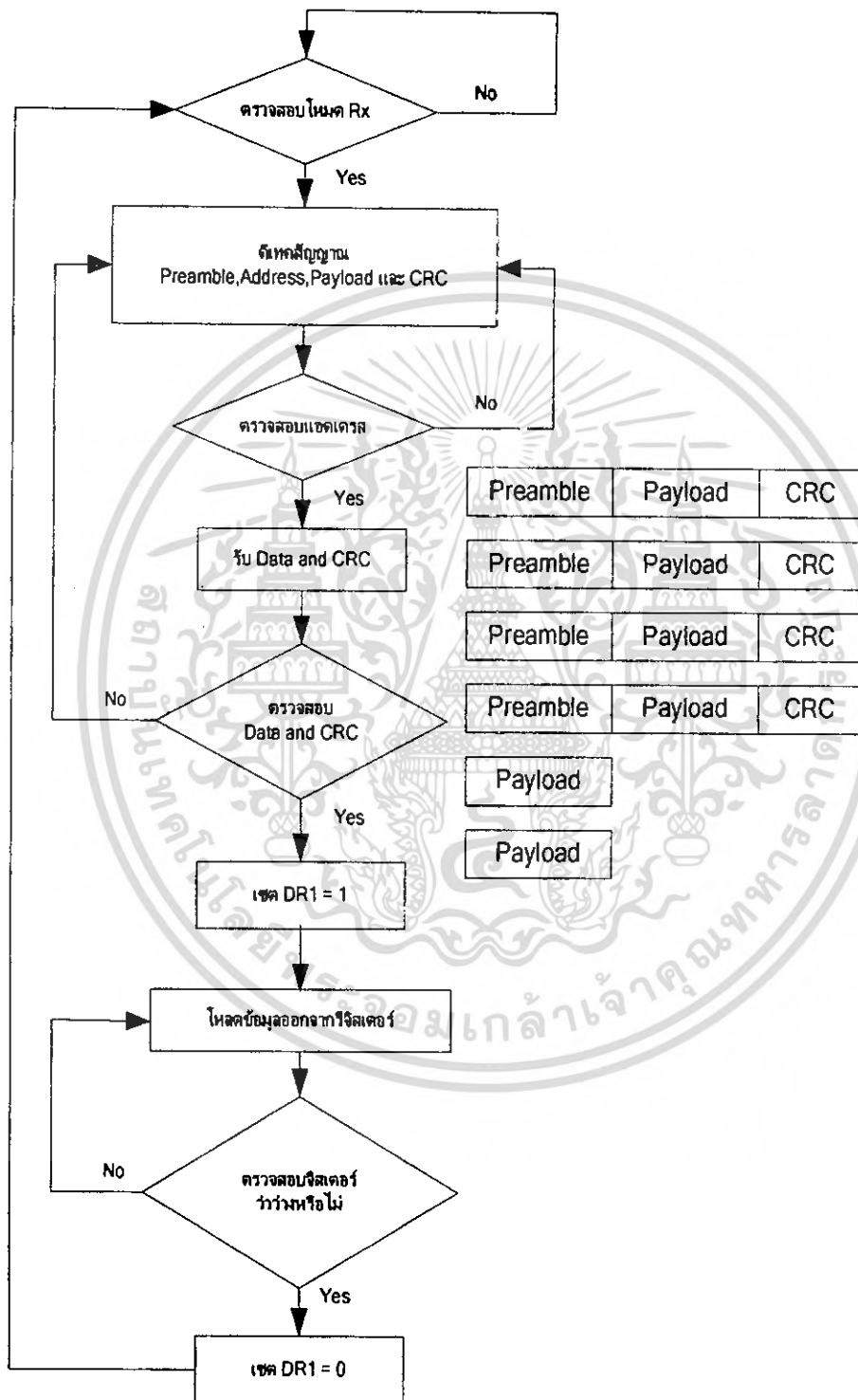


รูปที่ 2.31 โฟลว์ชาร์ต Shock Burst Transmit ของโมดูลความถี่วิทยุ (TRW - 2.4 GHz)

อธิบายการทํางานของโฟลว์ชาร์ต ตัวส่ง

หลังจากทำการกำหนดรูปแบบ (Configuration) ให้เป็นโหมด Tx แล้ว จากนั้นหลักการทํางานของโมดูลจะเป็นดังโฟลว์ชาร์ตรูปที่ 2.31 โดยโมดูลจะตรวจสอบว่าเป็น CE เป็น High หรือไม่ ถ้าใช่โมดูลจะรับแอดเดรสปลายทางและข้อมูล จากนั้นจะทำการคำนวณหา CRC เพื่อเป็นเฟรมที่ใช้ในการตรวจสอบข้อมูล จากนั้นจะตรวจสอบว่าเป็น CE ถ้าเป็น Low แสดงว่าเฟรมข้อมูลและ CRC พร้อมทั้งจะส่ง

แล้ว จากนั้น โมดูลจะทำการเพิ่มเฟรม Preamble เข้าไปแล้วทำการส่งแพ็คเกจเกิดของข้อมูลทั้งหมดและเมื่อส่งข้อมูลเสร็จเรียบร้อยแล้ว จะทำการวนกลับไปยังจุดเริ่มต้นของโฟลว์ชาร์ตใหม่อีกครั้ง



รูปที่ 2.32 โฟลว์ชาร์ต Shock Burst Receive ของโมดูลความถี่วิทยุ (TRW - 2.4 GHz)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อธิบายการทำงานของโฟลว์ชาร์ต ตัวรับ

จากรูปที่ 2.32 หลังจากทำการกำหนดรูปแบบ ให้เป็นโหมด Rx แล้วไมโครจะทำการตีเทคสัญญาณจาก ไมโครตัวส่ง หลังจากตีเทคสัญญาณพบแล้วจะทำการตรวจสอบแอดเดรสที่ส่งมาตรงกับแอดเดรสประจำตัวของไมโครตัวรับที่ทำการตั้งไว้หรือไม่ ถ้าใช่จะทำการรับข้อมูลกับ CRC จากนั้นทำการตรวจสอบข้อมูลว่าถูกต้องหรือไม่ ถ้าถูกต้องจะทำการเซตค่า DR1 ให้เป็นสถานะ “High” จากนั้นจะทำการโหลดข้อมูลออกจากรีจิสเตอร์ของตัวไมโคร โดยจะตรวจสอบว่ารีจิสเตอร์ของไมโครว่างหรือไม่ เมื่อรีจิสเตอร์ว่างแล้ว จะทำการเซตค่า DR1 ให้เป็นสถานะ “Low” และวนกลับไปยังจุดเริ่มของโฟลว์ชาร์ต

2.8.4 ตำแหน่งบิตข้อมูลของไมโครความถี่วิทยุ

ในการที่จะกำหนดค่าให้กับตัวไมโครความถี่วิทยุเพื่อให้ไมโครความถี่วิทยุทำงานในโหมดที่เราต้องการทราบก่อนว่าต้องกำหนดค่าอะไรบ้างลงในตำแหน่งบิตที่เท่าไร ดังนั้นตารางข้างล่างนี้แสดงตำแหน่งของบิตภายในตัวไมโครความถี่วิทยุที่ใช้ในการกำหนดค่าให้กับตัวไมโครความถี่วิทยุ

ตารางที่ 2.20 แสดงรายละเอียดของตำแหน่งบิตภายในตัวไมโครความถี่วิทยุ

ตำแหน่งบิต	จำนวนบิต	ชื่อ	ความหมาย
143 : 120	24	Test	จองไว้สำหรับทดสอบข้อมูล
119 : 112	8	DATA2_W	จำนวนความกว้างของบิตข้อมูล Channel 2
111 : 104	8	DATA1_W	จำนวนความกว้างของบิตข้อมูล Channel 1
103 : 64	40	ADDR2	แอดเดรสของตัวรับ Channel 2
63 : 24	40	ADDR1	แอดเดรสของตัวรับ Channel 1
23 : 18	6	ADDR_W	จำนวนบิตที่จองไว้สำหรับแอดเดรสตัวรับ
17	1	CRC_L	จำนวนความกว้างของบิต CRC
16	1	CRC_EN	ขอมให้สร้าง CRC สำหรับ Tx และมีการตรวจสอบบิต CRC สำหรับ Rx
15	1	RX2_EN	กำหนดจำนวนสัญญาณ Rx
14	1	CM	เลือกโหมด (Direct หรือ Shock Burst)
13	1	REDR_SB	เลือกอัตราการส่งผ่านข้อมูล (1 Mbps และ 250 Kbps)
12 : 10	3	XO_F	ความถี่คริสตอล
9 : 8	2	RF_PWR	เลือกค่ากำลังของตัวส่ง
7 : 1	7	RF_CH#	เลือกความถี่
0	1	RXEN	เลือกโหมดรับ / ส่ง

บทที่ 3

การสร้างและการออกแบบ

ในบทที่ 2 ได้กล่าวถึงในส่วนที่เป็นรายละเอียดของทฤษฎีต่างๆ ที่เกี่ยวข้องกับการสร้างโครงการมาแล้ว มาในบทนี้จะได้กล่าวถึงขั้นตอนในการออกแบบ การคำนวณ การสร้างและอธิบายหลักการทำงานของวงจรในแต่ละส่วนที่ประกอบกันขึ้นเป็นโครงการระบบมาตรวัดน้ำไร้สาย

3.1 บล็อกโคอะแกรมการทำงานของโครงการระบบมาตรวัดน้ำไร้สายและคำอธิบาย

โครงการระบบมาตรวัดน้ำไร้สายนี้ ประกอบไปด้วย 2 ส่วน คือ ส่วนอุปกรณ์เก็บข้อมูลและส่วนของมาตรวัดน้ำไร้สาย

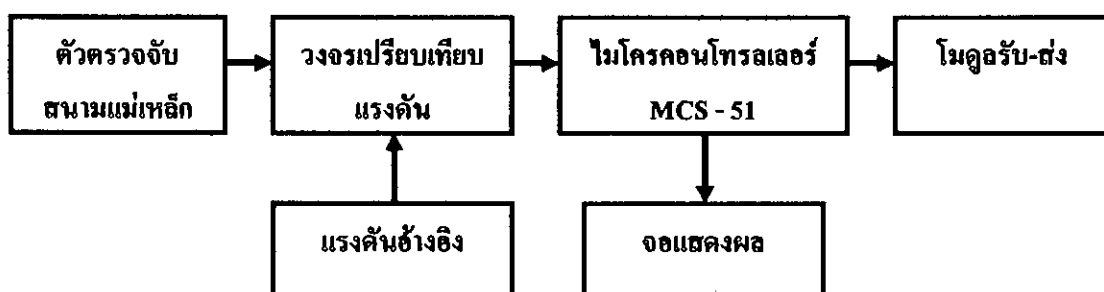
1. อุปกรณ์เก็บข้อมูล

- จอแสดงผล
- ไมโครคอนโทรลเลอร์ตระกูล MCS – 51
- สร้างฐานเวลา
- หน่วยความจำ
- คีย์แพด
- โมดูล เครื่องรับและเครื่องส่ง แบบ Half Duplex
- อินเทอร์เฟส

2. มาตรวัดน้ำไร้สาย

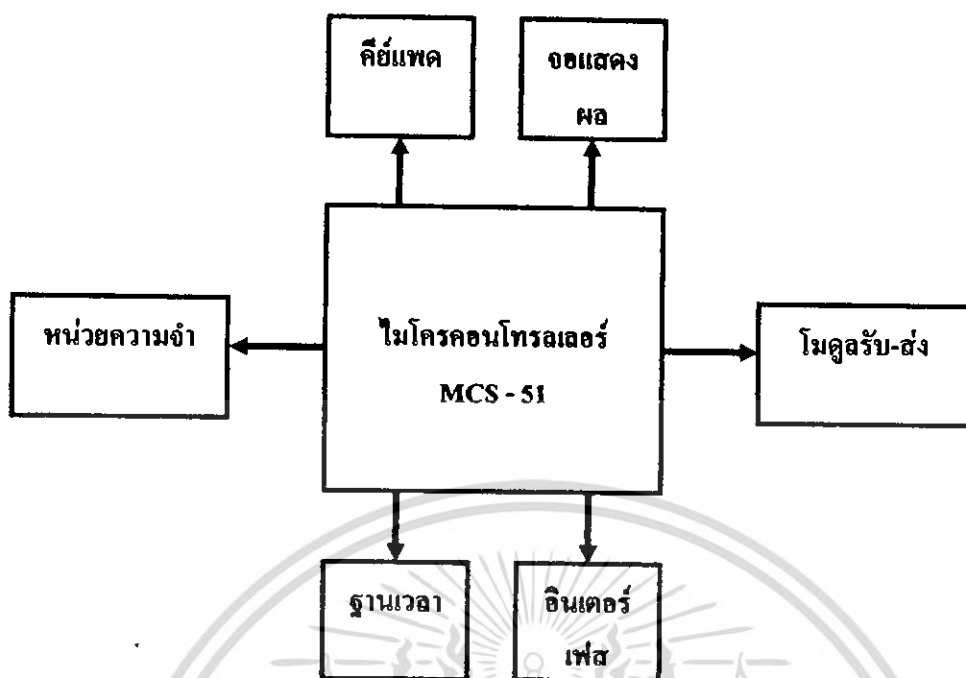
- โมดูล เครื่องรับและเครื่องส่ง แบบ Half Duplex
- จอแสดงผล
- ไมโครคอนโทรลเลอร์ตระกูล MCS – 51
- ตัวตรวจจับสนามแม่เหล็ก
- วงจรเปรียบเทียบแรงดัน (Comparator)

บล็อกโคอะแกรม (Block Diagram) แสดงการทำงานสามารถแบ่งออกเป็น 2 บล็อก คือ มาตรวัดน้ำไร้สาย และ อุปกรณ์เก็บข้อมูล แสดงดังรูปที่ 3.1 และรูปที่ 3.2



รูปที่ 3.1 บล็อกโคอะแกรมการทำงานของมาตรวัดน้ำไร้สาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไมอนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 บล็อกไดอะแกรมการทำงานของอุปกรณ์เก็บข้อมูล

หลักการทำงานสามารถอธิบายได้ดังนี้

สัญญาณควบคุมการสั่งงานมาตรวัดน้ำไร้สายโดยจะถูกส่งจากอุปกรณ์เก็บข้อมูลผ่านตัวโมดูลไปยังมาตรวัดน้ำไร้สาย มาตรวัดน้ำไร้สายก็จะทำงานตามคำสั่งที่ได้รับ จากนั้นก็จะส่งสัญญาณตอบสนองกลับมายังอุปกรณ์เก็บข้อมูล เพื่อให้อุปกรณ์เก็บข้อมูลรับทราบว่ามีมาตรวัดน้ำไร้สายได้รับสัญญาณควบคุมการสั่งงาน

รายละเอียดของการทำงานในส่วนของมาตรวัดน้ำไร้สาย อธิบายได้ดังนี้

จากรูปที่ 3.1

- จอแสดงผล ทำหน้าที่ในการแสดงรายละเอียดปริมาณน้ำที่ใช้ไปและชื่อของมาตรวัดน้ำไร้สาย
- ไมโครคอนโทรลเลอร์ จะเป็นหัวใจหลักของมาตรวัดน้ำไร้สาย โดยจะทำงานใน 2 ลักษณะคือ
 - ลักษณะที่ 1 จะทำหน้าที่ในการประมวลผลสัญญาณที่รับมาจากตัวตรวจจับสนามแม่เหล็ก
 - ลักษณะที่ 2 จะทำหน้าที่ในการรับ - ส่งสัญญาณจากอุปกรณ์เก็บข้อมูล
- วงจรเปรียบเทียบแรงดัน ทำหน้าที่ในการเปรียบเทียบระดับแรงดันจากตัวตรวจจับสนามแม่เหล็กและระดับแรงดันอ้างอิง
- ตัวตรวจจับสนามแม่เหล็ก จะทำหน้าที่ในการตรวจจับสนามแม่เหล็กที่ติดอยู่กับใบพัดของมาตรวัดน้ำเพื่อที่จะนำมาแสดงค่าปริมาณที่ใช้ไป โดยเทียบกับระบบบันทึกผลของมาตรวัดน้ำจริง

รายละเอียดของการทำงานในส่วนของอุปกรณ์เก็บข้อมูล อธิบายได้ดังนี้

จากรูปที่ 3.2

- คีย์แพด จะเป็นตัวในการรับคำสั่งจากผู้ควบคุมว่าจะเลือกทำงานเมนูใด
- จอแสดงผล จะเป็นตัวแสดงผลเมนูการทำงานและรายละเอียด

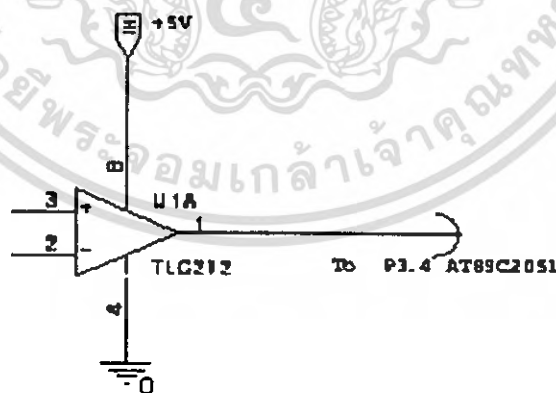
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ไมโครคอนโทรลเลอร์ เป็นหัวใจหลักของอุปกรณ์เก็บข้อมูล โดยจะทำงานใน 2 ลักษณะคือ ลักษณะที่ 1 จะทำหน้าที่รับคำสั่งจากคีย์แพคแล้วสร้างสัญญาณควบคุมซึ่งเป็นสัญญาณดิจิทัลออกมาเพื่อส่งไปยัง โมดูลตัวส่ง ลักษณะที่ 2 จะทำหน้าที่รับสัญญาณตอบรับซึ่งเป็นสัญญาณดิจิทัลจากตัวส่ง เพื่อนำมาแสดงสถานะการทำงานที่จอแสดงผล
- ฐานเวลา จะทำหน้าที่ในการสร้างสัญญาณเวลาและวันที่เพื่อเป็นตัวเปรียบเทียบเวลาในการทำงาน
- หน่วยความจำ จะทำหน้าที่ในการบันทึกข้อมูลและเก็บข้อมูลจากมาตรวัดน้ำไร้สายแต่ละตัว
- อินเตอร์เฟส จะทำหน้าที่ในการเชื่อมต่ออุปกรณ์เก็บข้อมูล ไปยังเครื่องคอมพิวเตอร์
- โมดูลรับ - ส่ง จะทำหน้าที่ในการรับ - ส่งข้อมูล

3.2 การสร้างและการออกแบบวงจร

3.2.1 วงจรเปรียบเทียบแรงดันและตัวตรวจจับสนามแม่เหล็ก

วงจรเปรียบเทียบแรงดันจะทำหน้าที่ในการเปรียบเทียบแรงดันระหว่างอินพุตที่ขา 3 ซึ่งเป็นแรงดันไฟ +5V กับแรงดันอินพุตที่ขา 2 ซึ่งเป็นแรงดันไฟฟ้าที่ได้จากตัวตรวจจับสนามแม่เหล็ก โดยในสถานะปกติแล้วเมื่อทำการป้อนแรงดันไฟฟ้า +5V ให้กับตัวตรวจจับสนามแม่เหล็ก จะมีแรงดันเอาต์พุตออกมาประมาณ +2.5V และในสถานะที่มีสนามแม่เหล็กมาตัดผ่านตัวตรวจจับสนามแม่เหล็ก จะมีแรงดันเอาต์พุตออกมาประมาณ +5V เป็นพัลส์สี่เหลี่ยมที่มีระดับแรงดัน +5V ในสถานะที่สนามแม่เหล็กมาตัดกับตัวตรวจจับสนามแม่เหล็ก จะทำให้แรงดันอินพุตที่ขา 2, 3 มีค่าเท่ากัน จึงต้องใช้ตัวต้านทานปรับค่าได้ 10 K Ω (VR 10 K Ω) ปรับระดับแรงดันที่อินพุตขา 3 ให้มีค่าน้อยกว่าระดับแรงดันอินพุตที่ขา 2 ทำให้ได้ระดับแรงดันเอาต์พุตที่ออกมามีค่าประมาณ +5V ซึ่งเป็นสัญญาณพัลส์ส่งต่อไปยังขา 8 ของไมโครคอนโทรลเลอร์เพื่อประมวลผลก่อนที่จะแสดงผลออกทางจอแสดงผลเทียบกับเครื่องบันทึกที่ตัวมาตรวัดน้ำไร้สาย



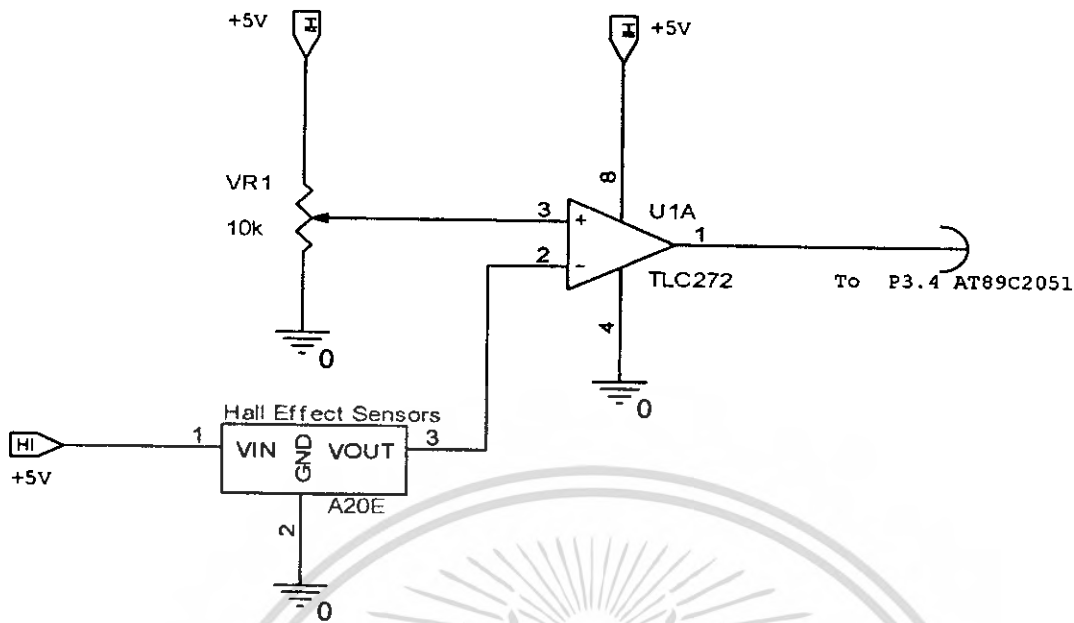
รูปที่ 3.3 วงจรเปรียบเทียบแรงดัน

โดยสมการ $V_{out} = G(V_2 - V_3)$

$G =$ อัตราการขยาย

การป้อน R-Feedback ออกไป จึงทำให้มีอัตราการขยายเป็นอนันต์ โดย V_{out} สูงสุดเท่ากับ 5 V และ V_{out} ต่ำสุดเท่ากับ 0 V

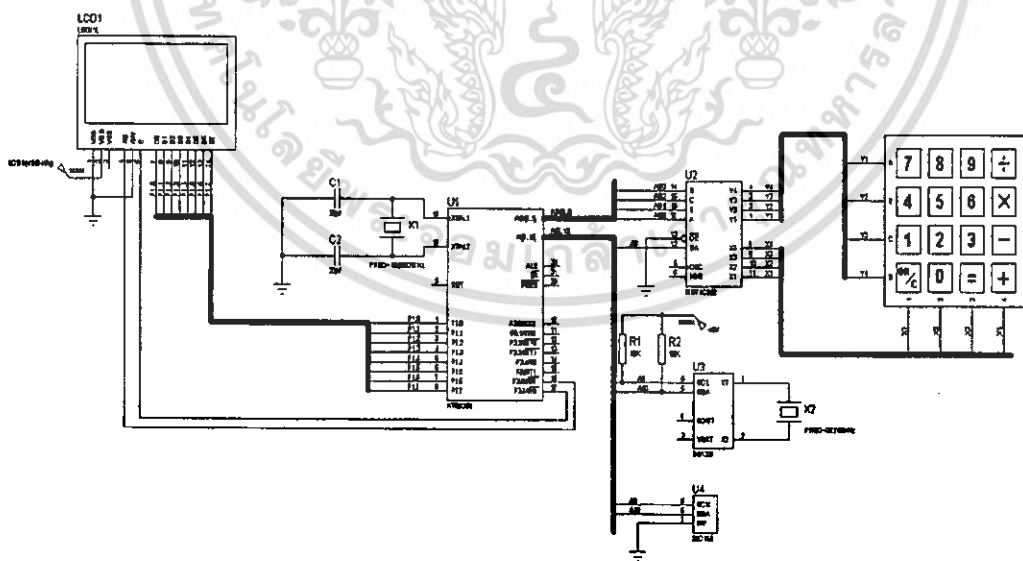
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดลอกเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.4 วงจรเปรียบเทียบแรงดันและตัวตรวจจับสนามแม่เหล็ก

3.2.2 วงจรไมโครคอนโทรลเลอร์ในอุปกรณ์เก็บข้อมูล

วงจรไมโครคอนโทรลเลอร์ในอุปกรณ์เก็บข้อมูลจะทำหน้าที่ในการรับค่าจากคีย์แพด ส่งสัญญาณควบคุมไปยังมาตรวัดน้ำไร้สายและรับสัญญาณจากมาตรวัดน้ำไร้สาย รายละเอียดการต่อวงจรในส่วนนี้แสดงได้ดังรูปที่ 3.5

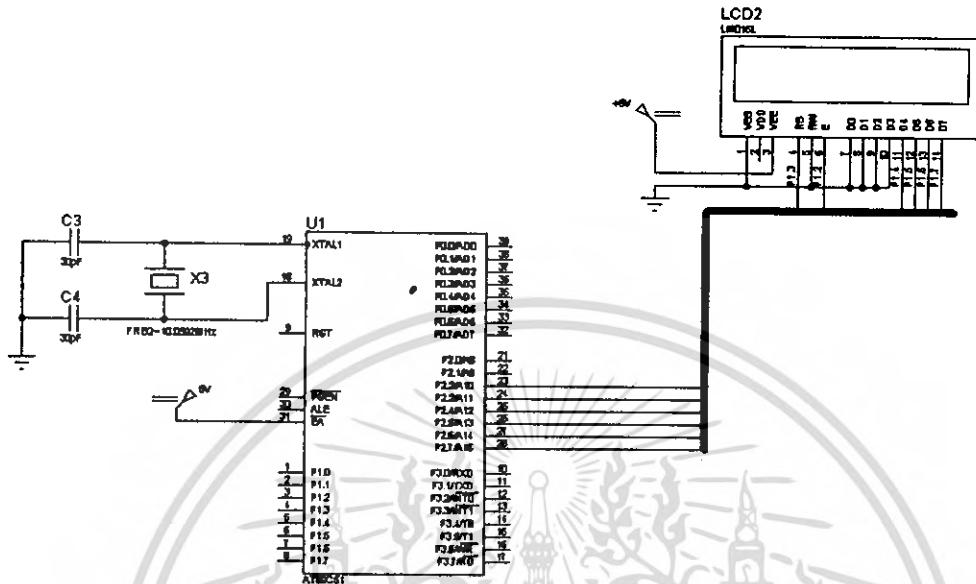


รูปที่ 3.5 วงจรไมโครคอนโทรลเลอร์ในอุปกรณ์เก็บข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

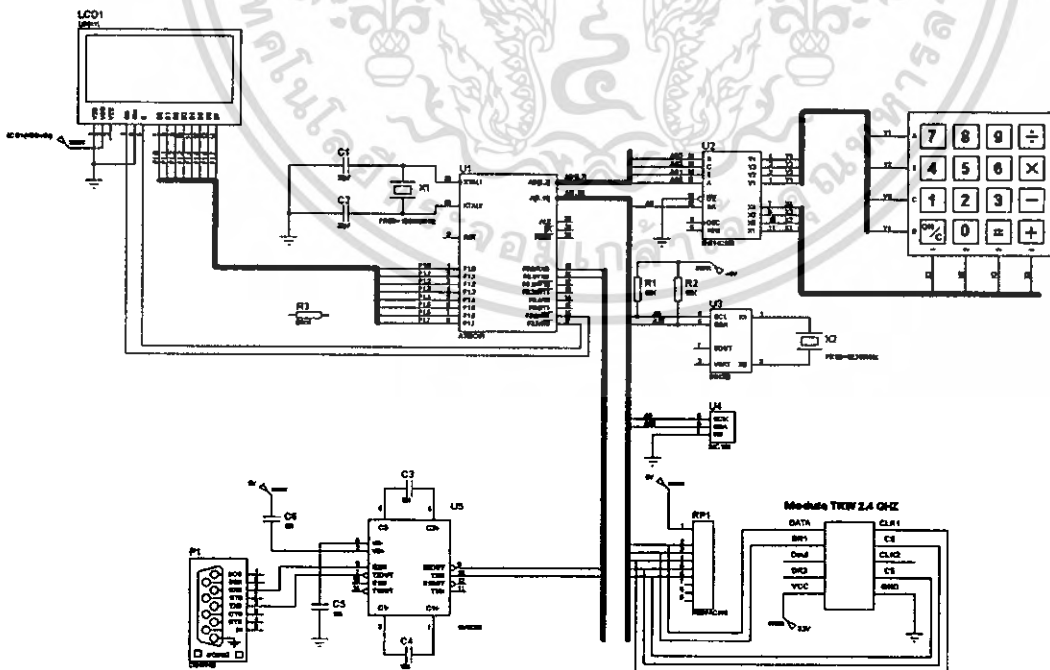
3.2.3 วงจรไมโครคอนโทรลเลอร์ในมาตรวัดน้ำไร้สาย

วงจรไมโครคอนโทรลเลอร์ในมาตรวัดน้ำไร้สายจะทำหน้าที่ในการรับสัญญาณจากอุปกรณ์เก็บข้อมูลและส่งสัญญาณกลับไปยังอุปกรณ์เก็บข้อมูล รายละเอียดการต่อวงจรในส่วนนี้แสดงผังรูปที่ 3.6



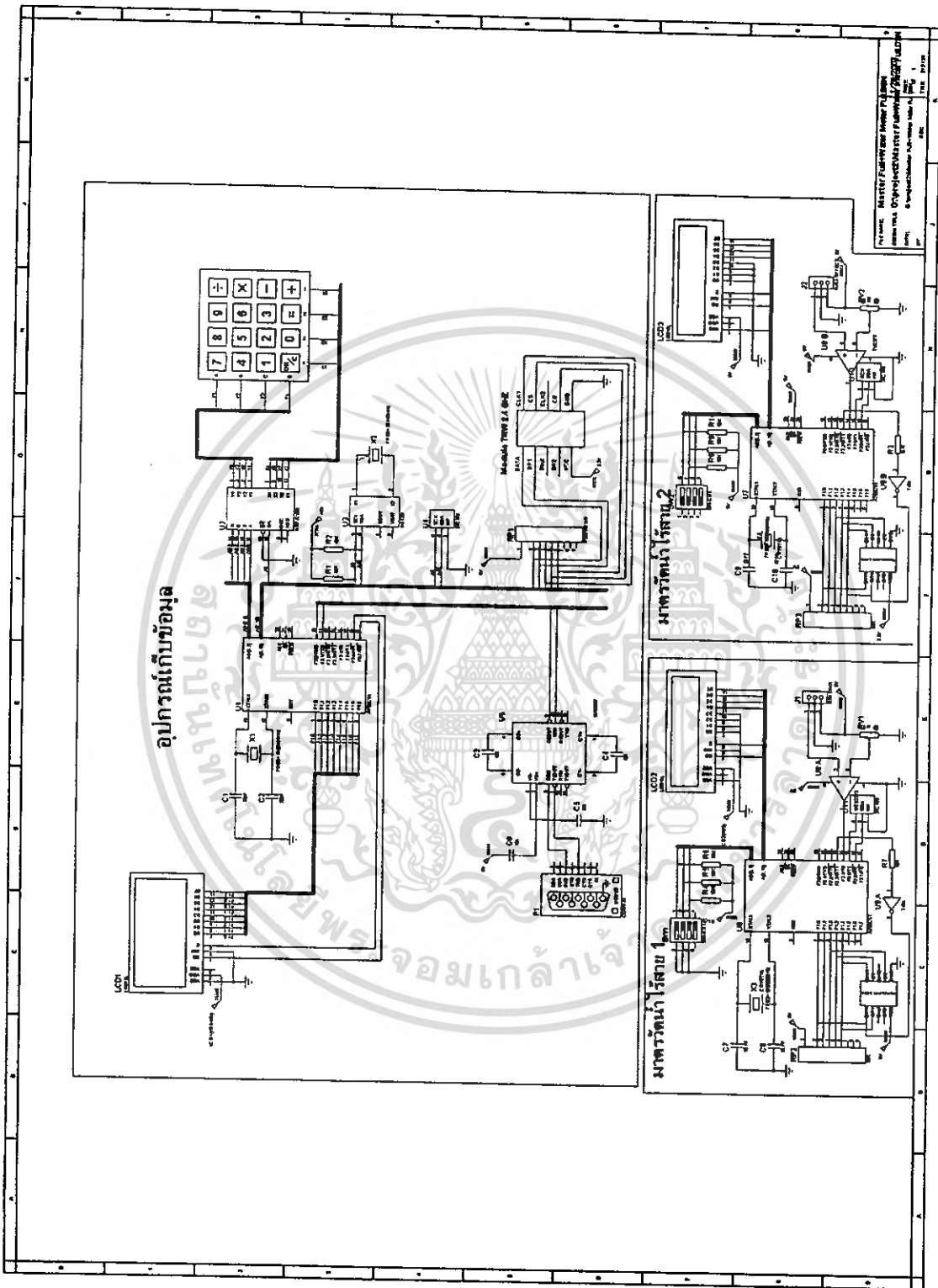
รูปที่ 3.6 วงจรไมโครคอนโทรลเลอร์ในมาตรวัดน้ำไร้สาย

วงจรรวมที่ใช้ในโครงการนี้แบ่งออกเป็น 2 ส่วน ได้แก่ วงจรรวมของอุปกรณ์เก็บข้อมูลและวงจรรวมของมาตรวัดน้ำไร้สาย แสดงผังรูปที่ 3.7



รูปที่ 3.7 วงจรรวมของอุปกรณ์เก็บข้อมูล

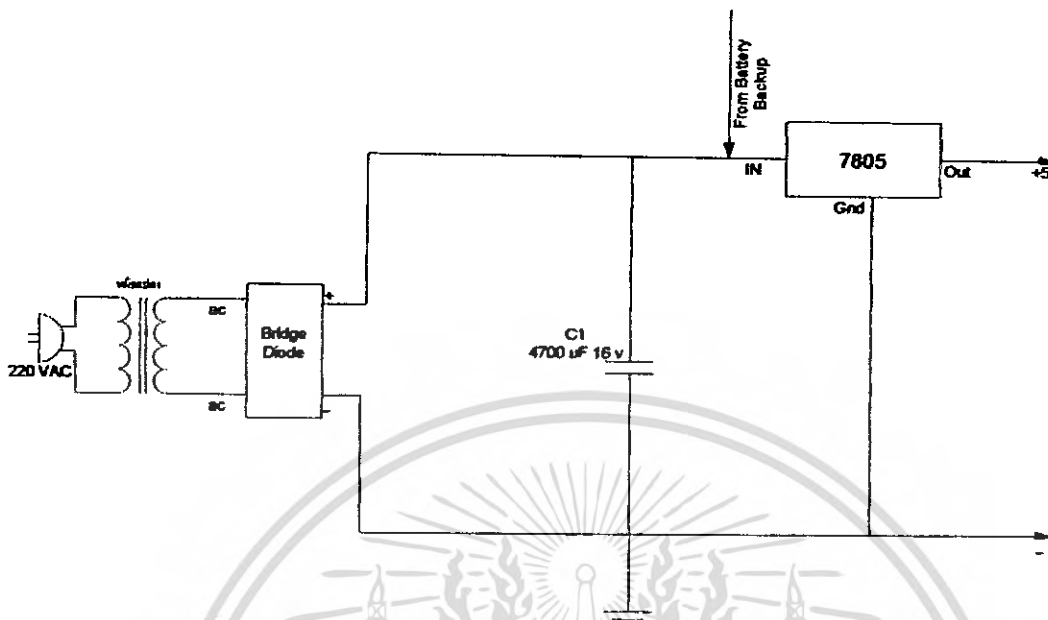
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.9 วงจรการต่อใช้งานจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.3.4 วงจรทาวเวอร์จ่ายหลาย (Power Supply)



รูปที่ 3.10 วงจรทาวเวอร์จ่ายหลาย (Power Supply)

ในส่วนของวงจรทั้งหมดจะใช้แหล่งจ่ายไฟคงที่เพียงชุดเดียว คือ แหล่งจ่ายไฟ +5V เป็นตัวจ่ายไฟให้กับวงจร สามารถอธิบายหลักการการทำงานได้คือ

วงจรทาวเวอร์จ่ายหลายแสดงดังรูปที่ 3.10 ซึ่งเป็นวงจรแหล่งจ่ายไฟอย่างง่ายจากไฟบ้าน 220 V ผ่านหม้อแปลงเพื่อแปลงแรงดันจาก 220 V ให้เหลือ 9 V แล้วนำมาต่อผ่านวงจรบริดจ์จะได้แรงดันไฟออกมามีประมาณ +12V ผ่าน C1 ค่า 4700 ไมโครฟารัด 16 V เพื่อกรองกระแสให้เรียบก่อนที่จะนำมาต่อเข้ากับไอซีเรกูเลเตอร์เบอร์ 7805 ก็จะได้แรงดันไฟที่กระแสตรงออกมา +5V เป็นตัวจ่ายไฟให้กับวงจร ซึ่งค่า C1 ที่ต้องเลือกใช้ค่ามากๆ ก็เพราะว่าเวลาเกิด ไฟดับจะได้มีไฟเลี้ยงวงจรก่อนที่วงจรแหล่งจ่ายไฟสำรอง (Battery Backup) จะทำงาน

3.3.5 วงจรแหล่งจ่ายไฟสำรอง (Battery Backup)

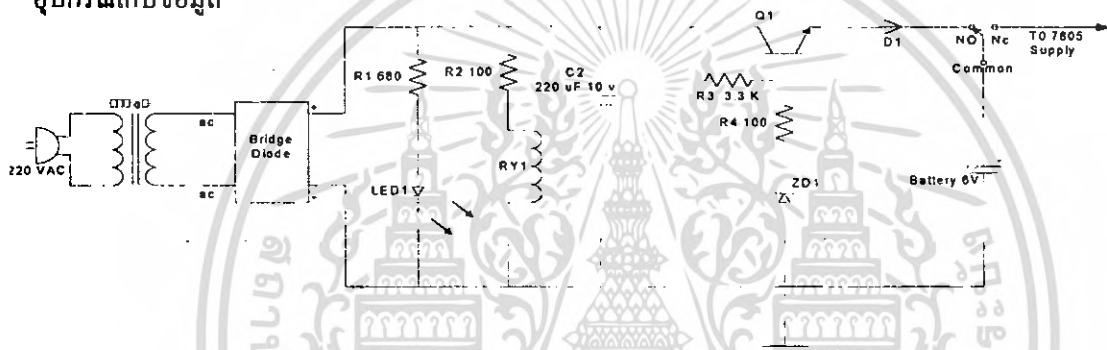
ในส่วนของวงจรจะใช้วงจรแหล่งจ่ายไฟสำรองเพียงชุดเดียวคือ ไฟที่กระแสตรง +5V เพื่อเป็นตัวจ่ายไฟให้กับวงจรในช่วงที่เกิดไฟดับ นอกจากนี้วงจรที่ไว้ยังสามารถชาร์จไฟได้อีกด้วยดังแสดงดังรูปที่ 3.11 ซึ่งในส่วนนี้ของวงจรชาร์จไฟสามารถอธิบายได้คือ

วงจรชาร์จไฟจะใช้วงจรอย่างง่ายจากไฟบ้าน 220 V ผ่านหม้อแปลงเพื่อแปลงแรงดันจาก 220 V ให้เหลือ 9V แล้วนำมาต่อผ่านวงจรบริดจ์จะได้แรงดันไฟออกมามีประมาณ +12V ผ่าน C2 ค่า 220 ไมโครฟารัด เพื่อกรองกระแสให้เรียบ LED1 จะสว่างเมื่อมีไฟเข้าวงจร โดยมี R1 เป็นตัวจำกัดกระแสให้พอเหมาะแก่ LED1 ไฟที่ผ่านการแปลงเป็นไฟกระแสตรงแล้วจะถูกส่งต่อมาซึ่งทรานซิสเตอร์ Q1 ซึ่งเป็นวงจรเรกูเลเตอร์ควบคุมทั้งแรงดันและกระแสเพื่อให้เหมาะสม ก่อนที่จะป้อนเข้าวงจรชาร์จไฟ มีตัวไดโอด D1 เป็นตัวกั้นกระแสไม่ให้ไหลย้อนกลับจากแหล่งจ่ายไฟเข้าสู่วงจร แรงดันเรกูเลเตอร์จะถูก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น. ไม่อนุญาตให้นำไปใช้ประโยชน์ทางการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ควบคุมให้คงที่โดยเซ็นเซอร์ไดโอด ZD1 ค่า 6.8 V โดยแรงดันเอาต์พุตที่ออกมาจะมีค่า 6V (แรงดันเอาต์พุตจะมีค่าต่ำกว่าแรงดันของไดโอดอยู่ 0.8V) กระแสสูงสุดของวงจรจะจำกัดอยู่ที่ค่าของ R3 ในวงจรจะจำกัดอยู่ประมาณ 150 มิลลิแอมป์ เต็มที่ขณะแหล่งจ่ายไฟมีแรงดัน 0V หรือขณะไฟจากจากแหล่งจ่ายไฟหมดวงจรจะชาร์จกระแสเข้าไปในแหล่งจ่ายไฟเรื่อยๆ เมื่อแรงดันเพิ่มขึ้นกระแสที่ชาร์จเข้าไปในแหล่งจ่ายไฟก็จะมีค่าลดลงในขณะเดียวกันแรงดันจะมีค่าเพิ่มสูงขึ้นจนถึง 7.8 V หรือชาร์จไฟเต็ม กระแสที่ชาร์จจะไม่หยุดทันทีแต่จะลดเหลือประมาณ 50 มิลลิแอมป์ เพื่อเลี้ยงให้แหล่งจ่ายไฟยังคงมีไฟ เต็มอยู่ตลอดเวลาชดเชยกับความสูญเสียภายในแหล่งจ่ายไฟ

เนื่องจากวงจรคล้ายกับไฟตลอดเวลากระแสที่ชาร์จจึงกำหนดไว้ไม่สูงนัก เมื่อกระแสไฟฟ้า คับริเลย์ซึ่งปกติหน้าสัมผัสจะทำหน้าที่ต่อแหล่งจ่ายไฟเข้ากับวงจรชาร์จไฟ เมื่อเกิดไฟคับริเลย์จะสวิตช์ให้แหล่งจ่ายไฟต่อเข้ากับวงจรที่ต่อใช้งาน โดยวงจรที่ต่อใช้งานนี้จะเป็นวงจร ไมโครคอนโทรลเลอร์ของ อุปกรณ์เก็บข้อมูล



รูปที่ 3.11 วงจรแหล่งจ่ายไฟสำรอง

3.3 วิธีการใช้งานและตัวอย่างการใช้งานระบบมาตรวัดน้ำไร้สาย

ความสามารถของโครงการระบบมาตรวัดน้ำไร้สาย ได้แก่ การแสดงปริมาณน้ำที่ใช้ไปในหน่วยลูกบาศก์เมตรทางหน้าจอแสดงผลเทียบกับเครื่องบันทึกของมาตรวัดน้ำจริง, การบันทึกข้อมูล, การแสดงเวลาและวันที่และการถ่ายโอนข้อมูลไปยังเครื่องคอมพิวเตอร์

ต่อไปนี้จะเป็นการยกตัวอย่างวิธีการใช้งานระบบมาตรวัดน้ำไร้สายในกรณีต่างๆ ดังต่อไปนี้

1. การตั้งเวลาและวันที่

ให้ทำตามขั้นตอนดังนี้

- กดปุ่ม T/D ที่คีย์แพดจะปรากฏเมนูในการตั้งเวลาและวันที่
- กดปุ่ม T/D เพื่อตั้งเวลา จากนั้นกดปุ่ม Send
- กดปุ่ม R/Met เพื่อทำการตั้งวันที่ จากนั้นกดปุ่ม Send
- จะแสดงเวลาและวันที่ที่ได้ทำการตั้งไป

2. การสั่งให้มาตรวัดน้ำไร้สายส่งค่าปริมาณน้ำที่แสดงที่ตัวมาตรวัดน้ำไร้สายมายังอุปกรณ์เก็บข้อมูล

- กดปุ่ม R/Met ที่คีย์แพดจะปรากฏเมนูให้ทำการระบุตำแหน่งของมาตรวัดน้ำไร้สายที่ต้องการ

ดูค่าปริมาณน้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ระบุตำแหน่งของมาตรวัดน้ำไร้สายสองหลัก เช่น 03, 01
- กดปุ่ม Send
- ที่หน้าจอของอุปกรณ์เก็บข้อมูลจะแสดงปริมาณน้ำที่อ่านได้จากมาตรวัดน้ำไร้สายและจะแสดงเวลาและวันที่
- กดปุ่ม Clear เพื่อเข้าสู่เมนูหลัก

ตัวอย่างข้างบนเป็นเพียงบางตัวอย่างของการใช้งานของโครงการระบบมาตรวัดน้ำไร้สายเท่านั้น ในการใช้งานจริงที่อุปกรณ์เก็บข้อมูลจะมีจอแสดงผล ทำการแสดงขั้นตอนในการใช้งานให้ทำตามอย่างเป็นขั้นตอนไม่ยุ่งยาก

3.4 ส่วนของซอฟต์แวร์

ส่วนของซอฟต์แวร์จะแบ่งเป็น 3 ส่วนโดยจะแสดงการทำงานของซอฟต์แวร์ต่างๆ เป็นลักษณะของโฟลว์ชาร์ต

1. ส่วนของอุปกรณ์เก็บข้อมูล

- โปรแกรมหลักของอุปกรณ์เก็บข้อมูล
- การตั้งเวลาและวันที่
- การอ่านค่าจากค่าปริมาณน้ำที่บันทึกไว้ในหน่วยความจำ
- การอ่านค่าปริมาณน้ำจากมาตรวัดน้ำไร้สาย
- การแก้ไขค่าปริมาณน้ำที่มาตรวัดน้ำไร้สาย
- การขัดจังหวะ (Interrupt) ที่ขา Rx ใช้ในการติดต่อไมโครคอนโทรลเลอร์กับคอมพิวเตอร์

2. ส่วนของมาตรวัดน้ำไร้สาย

- การวัดปริมาณน้ำโดยอาศัยการตัดผ่านของสนามแม่เหล็กที่ตัวตรวจจับสนามแม่เหล็ก
- การขัดจังหวะ (Interrupt) เพื่ออ่านหรือแก้ไขปริมาณน้ำที่มาตรวัดน้ำไร้สาย

3. ส่วนของคอมพิวเตอร์

- การแสดงปริมาณน้ำโดยรับข้อมูลจากอุปกรณ์เก็บข้อมูล

3.4.1 ส่วนของอุปกรณ์เก็บข้อมูล

- การตั้งเวลาและวันที่

โดยจะกดปุ่ม T/D จากนั้นเลือกว่าต้องการตั้งเวลาหรือวันที่ หรือเลือกที่จะออกโดย กดปุ่ม Clear เพื่อเข้าสู่เมนูหลัก เมื่อเลือกการตั้งเวลาหรือวันที่ ทำการตั้งค่าโดยใช้คีย์แพดเมื่อป้อนเสร็จแล้ว กดปุ่ม Send

- การอ่านค่าจากค่าปริมาณน้ำที่บันทึกไว้ในหน่วยความจำ

โดยจะกดปุ่ม R/Mem จากนั้นทำการใส่ตำแหน่งของมาตรวัดน้ำไร้สายแล้วกดปุ่ม Send จากนั้นหน้าจอจะแสดงปริมาณน้ำที่ใช้ไปและวันที่ ที่ทำการบันทึกจากนั้น กดปุ่ม Clear เพื่อเข้าสู่เมนูหลัก

- การอ่านค่าปริมาณน้ำจากมาตรวัดน้ำไร้สาย

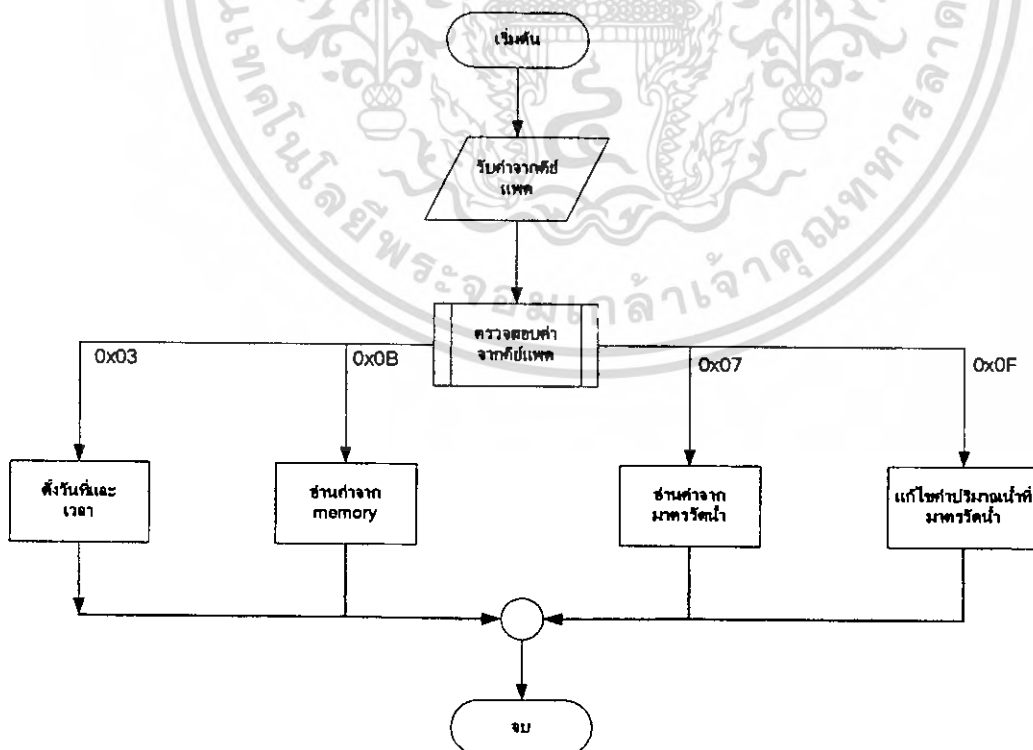
โดยจะกดปุ่ม R/Met จากนั้นทำการใส่ตำแหน่งของมาตรวัดน้ำไร้สายแล้วกดปุ่ม Send จากนั้นหน้าจอจะแสดงปริมาณน้ำที่และวันที่ แต่ถ้ามาตรวัดน้ำไร้สายไม่ตอบสนองหน้าจอจะแสดง Can't find จากนั้น กดปุ่ม Clear เพื่อเข้าสู่เมนูหลัก

- การแก้ไขค่าปริมาณน้ำที่มาตรวัดน้ำไร้สาย

โดยจะกดปุ่ม Set Met จากนั้นทำการใส่ตำแหน่งของมาตรวัดน้ำไร้สายแล้วกดปุ่ม Send แล้วทำการใส่ค่าปริมาณน้ำที่ต้องการแก้ไขลงไปแล้วกดปุ่ม Send อีกครั้ง หน้าจอจะแสดง Edit complete ถ้ามาตรวัดน้ำไร้สายไม่มีการตอบสนองหรือไม่ได้รับข้อมูลที่ส่งไปหน้าจอจะแสดง Can't find จากนั้น กดปุ่ม Clear เพื่อเข้าสู่เมนูหลัก

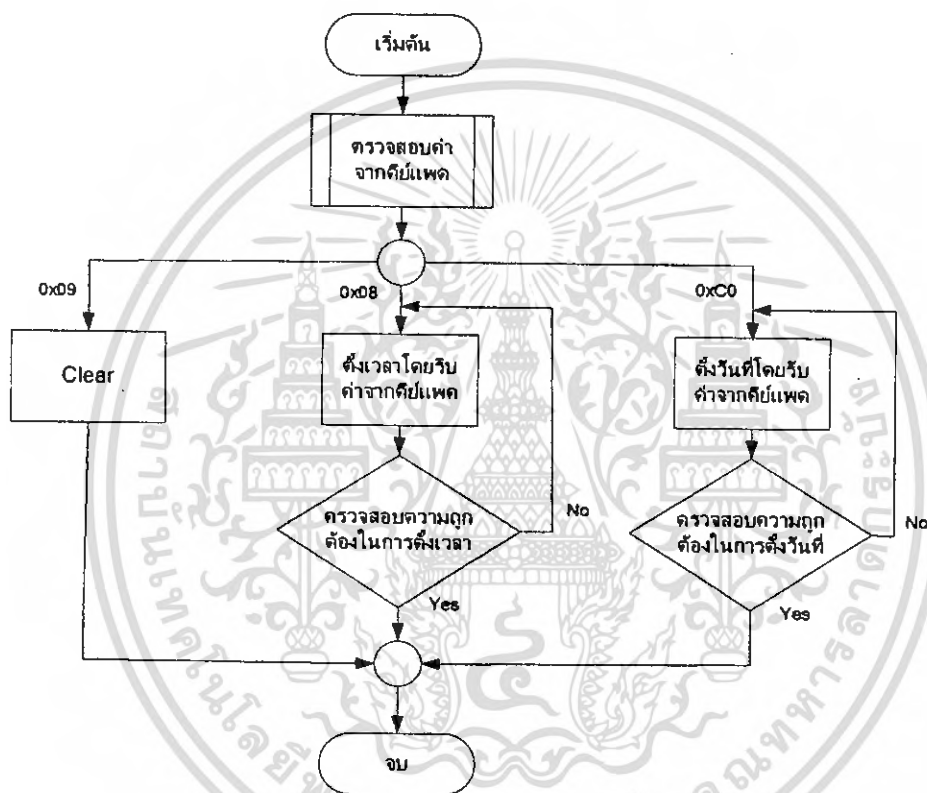
- การขัดจังหวะ (Interrupt) ที่ขา Rx ใช้ในการติดต่อไมโครคอนโทรลเลอร์กับคอมพิวเตอร์

เมื่อมีการขัดจังหวะที่ขา Rx ไมโครคอนโทรลเลอร์ทำการตรวจสอบว่าค่าที่ส่งมาคือค่าอะไร ถ้าเป็นค่า 0x41 หมายถึงคอมพิวเตอร์ ต้องการอ่านข้อมูลที่ได้นั้นทีในหน่วยความจำ จากนั้น ไมโครคอนโทรลเลอร์จะทำการส่ง 0x45 เพื่อบอกคอมพิวเตอร์ว่าพร้อมให้บริการแล้ว จากนั้นคอมพิวเตอร์จะทำการส่งตำแหน่งของมาตรวัดน้ำไร้สาย ที่ต้องการทราบข้อมูลให้กับไมโครคอนโทรลเลอร์ แล้วไมโครคอนโทรลเลอร์ก็จะส่งค่าปริมาณน้ำที่บันทึกและเวลาของมาตรวัดน้ำไร้สายที่คอมพิวเตอร์ต้องการออกไปให้กับคอมพิวเตอร์



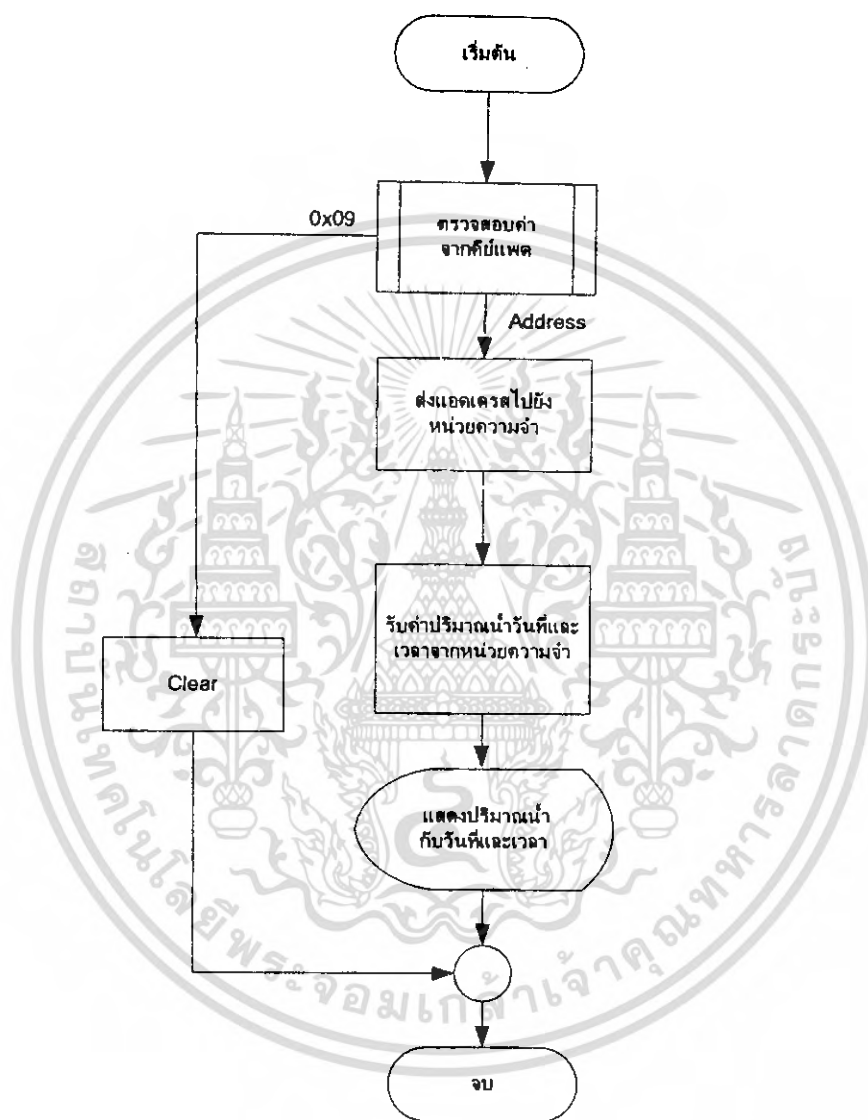
รูปที่ 3.12 โฟลว์ชาร์ตโปรแกรมหลักของอุปกรณ์เก็บข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น เมื่อผู้ผู้ใดนำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



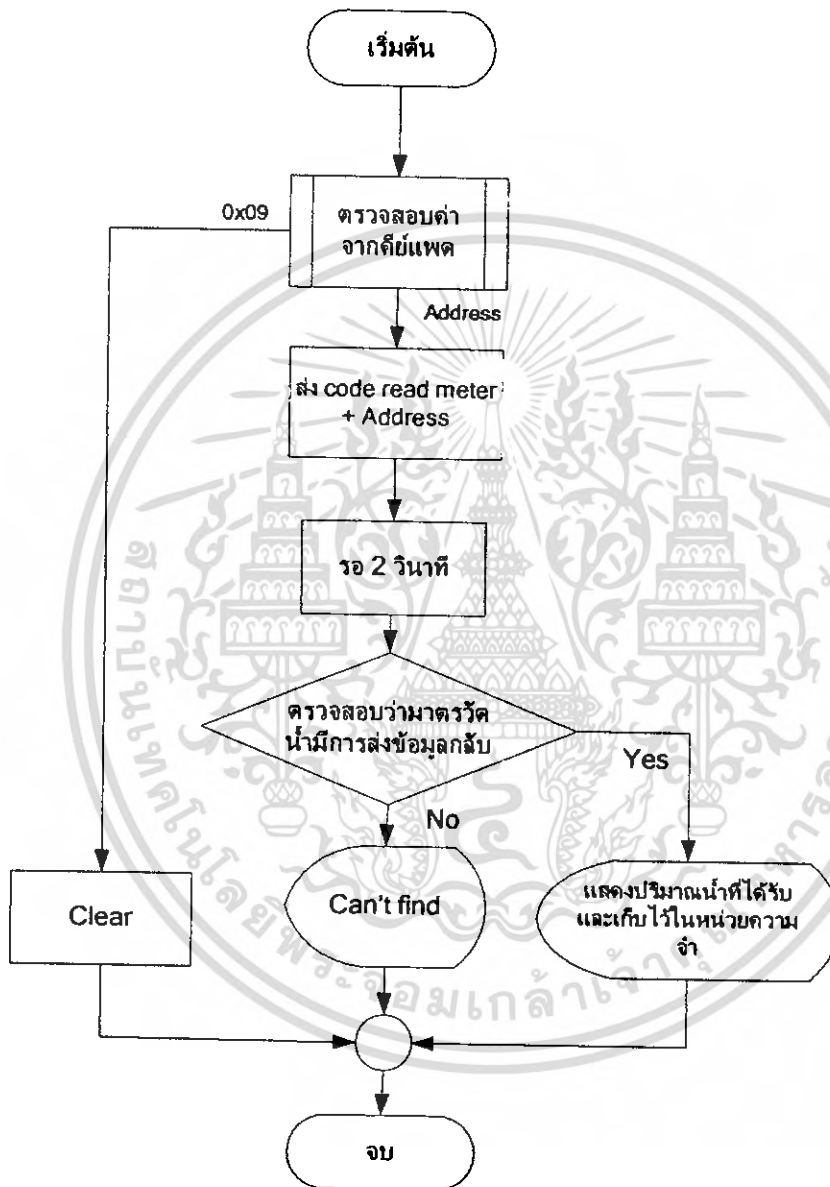
รูปที่ 3.13 โฟลว์ชาร์ตการตั้งเวลาและวันที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



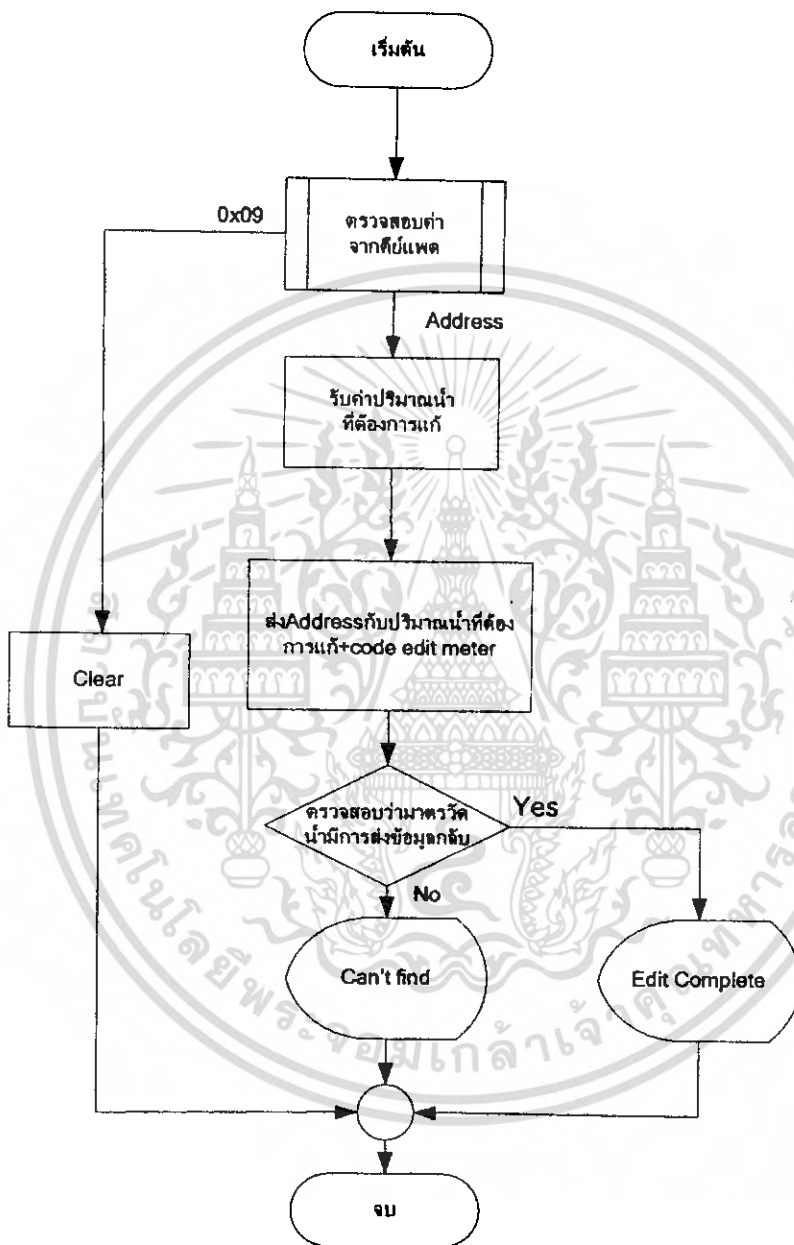
รูปที่ 3.14 โฟลว์ชาร์ตการอ่านค่าจากค่าปริมาณน้ำที่บันทึกไว้ในหน่วยความจำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



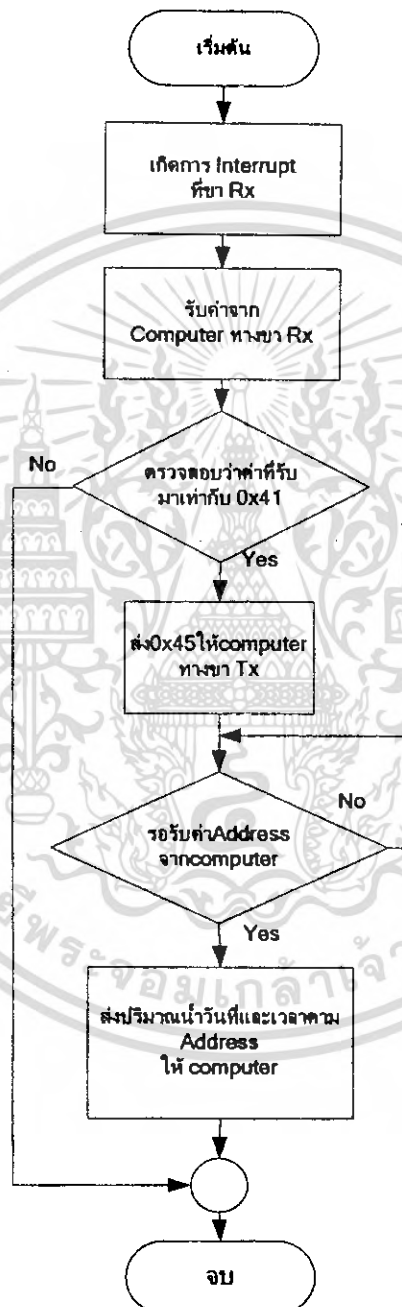
รูปที่ 3.15 โฟลว์ชาร์ตการอ่านค่าปริมาณน้ำจากมาตรวัดน้ำไร้สาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.16 โปรแกรมการแก้ไขค่าปริมาณน้ำที่มาตรวัดน้ำไร้สาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.17 โปรแกรมการขัดจังหวะ (Interrupt) ที่ขา Rx ใช้ในการติดต่อกับไมโครคอนโทรลเลอร์กับคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

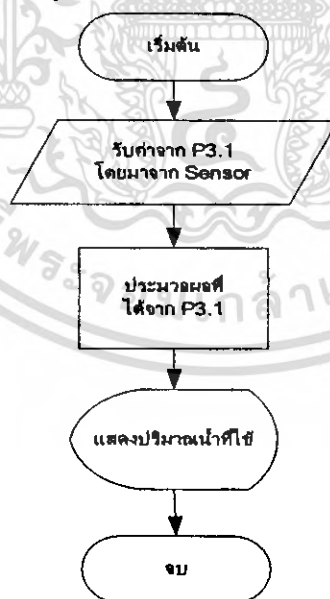
3.4.2 ส่วนของมาตรวัดน้ำไร้สาย

- การวัดปริมาณน้ำโดยอาศัยการตัดผ่านของสนามแม่เหล็กที่ตัวตรวจจับสนามแม่เหล็ก

การวัดปริมาณน้ำ โดยอาศัยการตัดผ่านของสนามแม่เหล็กที่ตัวจับสนามแม่เหล็ก โดยจะรับสัญญาณเข้าทางขา P3.1 โดยที่ขา P3.1 จะต่อกับเอาต์พุตของออปแอมป์ในวงจรเปรียบเทียบแรงดัน ซึ่งเอาต์พุตของออปแอมป์ในวงจรเปรียบเทียบแรงดัน จะ ได้จากการเปรียบเทียบผลต่างของตัวตรวจจับสนามแม่เหล็ก กับแรงดันที่ได้จากการปรับแรงดันของตัวต้านทานปรับค่าได้ (ขา 3 ของออปแอมป์) โดยโปรแกรมจะทำการนับจำนวนครั้งที่สนามแม่เหล็กตัดผ่านโดยจำนวนสนามแม่เหล็กที่ตัดผ่านตัวตรวจจับสนามแม่เหล็ก 400 ครั้ง เท่ากับปริมาณน้ำ 0.001 หน่วยแล้วนำค่าที่ได้ไปประมวลผล แล้วส่งผลลัพธ์ไปยังจอแสดงผล

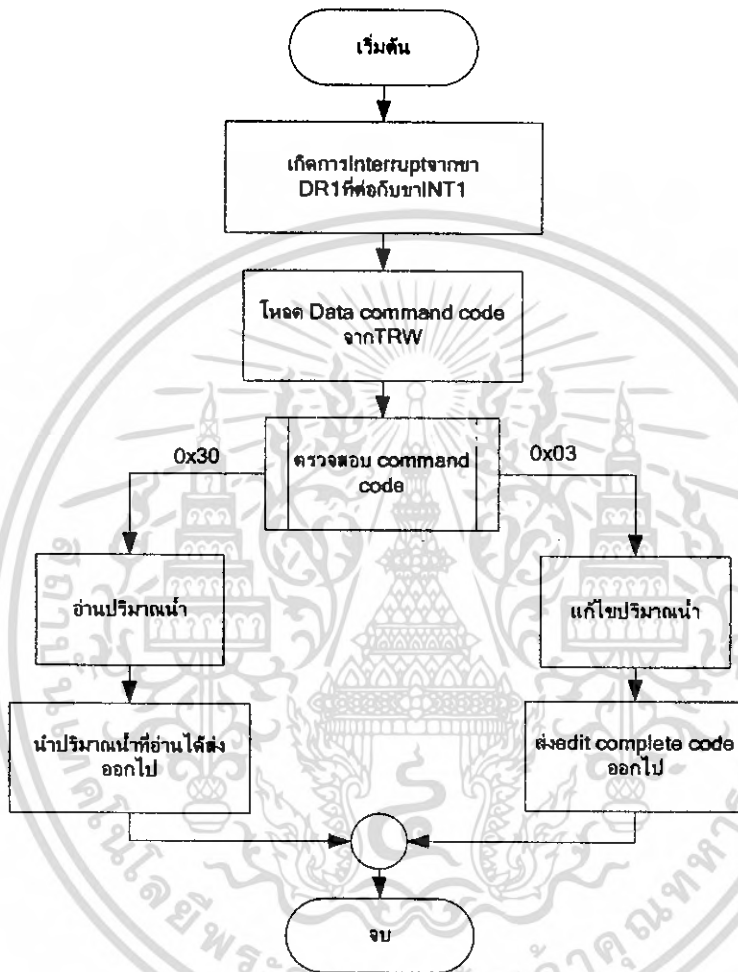
- การขัดจังหวะ (Interrupt) เพื่ออ่านหรือแก้ไขปริมาณน้ำที่มาตรวัดน้ำไร้สาย

การขัดจังหวะนี้จะใช้ขา DR1 ของไมโครคอนโทรลเลอร์เป็นตัวขัดจังหวะต่อเข้ากับขา INT1 ของไมโครคอนโทรลเลอร์โดยผ่าน NOT GATE ก่อน เมื่อไมโครคอนโทรลเลอร์ได้รับข้อมูลขา DR1 จะเซตค่าเป็น "High" ซึ่งจะทำให้ที่ขา INT1 เป็น "Low" จึงเกิดการขัดจังหวะของโปรแกรมขึ้น จากนั้นจะให้ไมโครคอนโทรลเลอร์โหลดรหัสคำสั่ง (Command Code) เพื่อมาพิจารณาว่าเป็นรหัสคำสั่งอะไร ถ้ารหัสคำสั่งคือ 0x03 จะเป็นการแก้ไขค่าปริมาณน้ำที่ตัวมาตรวัดน้ำไร้สาย โดยไมโครคอนโทรลเลอร์จะทำการแก้ไขค่าแล้วส่งรหัสแก้ไขเสร็จ (Complete Code) ให้กับอุปกรณ์เก็บข้อมูล แต่ถ้ารหัสคำสั่งคือ 0x30 จะเป็นการอ่านค่าปริมาณน้ำที่ตัวมาตรวัดน้ำไร้สาย โดยไมโครคอนโทรลเลอร์จะทำการส่งค่าที่อ่านได้ขณะนั้นส่งออกไปให้กับอุปกรณ์เก็บข้อมูล



รูปที่ 3.18 ไฟล์ซาร์ตการวัดปริมาณน้ำโดยอาศัยการตัดผ่านของสนามแม่เหล็กที่ตัวตรวจจับสนามแม่เหล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



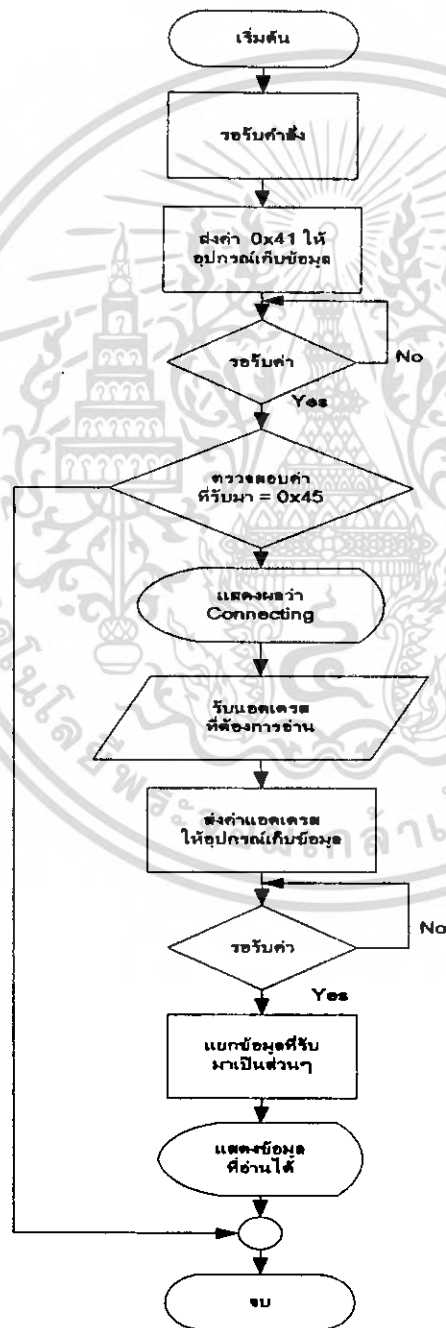
รูปที่ 3.19 โฟลว์ชาร์ตการขัดจังหวะ(Interrupt) เพื่ออ่านหรือแก้ไขปริมาณน้ำที่มาตรวัดน้ำไร้สาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.3 ส่วนของคอมพิวเตอร์

- การแสดงปริมาณน้ำโดยรับข้อมูลจากอุปกรณ์เก็บข้อมูล

โดยโปรแกรมจะรอรับคำสั่งเมื่อกดปุ่ม Connect to address คอมพิวเตอร์จะทำการส่งค่า 0x41 ให้กับอุปกรณ์เก็บข้อมูลแล้วรอรับค่าจากนั้นจะทำการตรวจสอบค่าที่อุปกรณ์เก็บข้อมูลส่งมาว่าเท่ากับ 0x45 ถ้าใช่จะแสดงผลว่า Connecting แล้วรับค่าตำแหน่งมาครวดน้ำไว้สายที่ต้องการอ่านข้อมูลจากนั้นจะส่งค่าตำแหน่งมาครวดน้ำไว้สายที่ต้องการอ่านข้อมูลออกไปให้กับอุปกรณ์เก็บข้อมูลแล้วรอรับข้อมูลที่อุปกรณ์เก็บข้อมูลส่งมาเมื่อได้รับข้อมูลที่อุปกรณ์เก็บข้อมูลส่งมาจะทำการแยกข้อมูลเป็นส่วนๆแล้วแสดงผลออกสู่หน้าจอ



รูปที่ 3.20 ไฟล์ชาร์ตการแสดงผลปริมาณน้ำโดยรับข้อมูลจากอุปกรณ์เก็บข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

การทดลองและผลการทดลอง

4.1 อุปกรณ์เก็บข้อมูล

อุปกรณ์เก็บข้อมูลจะรับคำสั่งจากคีย์แพด โดยเข้าทางขา P0.0, P0.1, P0.2, P0.3 โดยตัวไมโครคอนโทรลเลอร์ จะทำการตรวจสอบอินพุตที่เข้าทางขา P0.0, P0.1, P0.2, P0.3 ที่ได้จากการกดคีย์แพดแล้วส่งไปยังจอแสดงผลหรือส่งข้อมูลออกทางขา Tx ของตัวไมโครคอนโทรลเลอร์ ขึ้นอยู่กับการกดคีย์แพด โดยจะมีคำสั่งการตั้งเวลาและวันที่ การส่งข้อมูลไปยังมาตรวัดน้ำไร้สาย

เมื่อเข้าไปยังคำสั่งการตั้งเวลาและวันที่แล้ว ไมโครคอนโทรลเลอร์จะรอรับคำสั่งอีกครั้งเพื่อเลือกที่จะตั้งเวลาหรือวันที่ โดยรอรับคำสั่งจากคีย์แพดอีกครั้ง เมื่อตั้งเวลาและวันที่เรียบร้อยแล้ว ข้อมูลจะถูกส่งไปยัง IC DS1307 เพื่อเป็นฐานข้อมูลวันและเวลาให้กับตัวไมโครคอนโทรลเลอร์

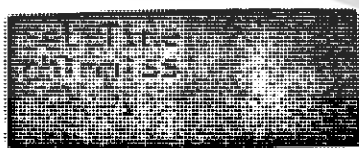
1. การตั้งเวลาเวลาและวันที่

เมื่อกดปุ่ม T/D ที่คีย์แพดจะปรากฏเมนูในการตั้งเวลาและวันที่ดังรูปที่ 4.1



รูปที่ 4.1 เป็นการแสดงคำสั่งเพื่อเลือกการตั้งเวลาหรือการตั้งวันที่

เมื่อกดปุ่ม T/D ซ้ำ เป็นการตั้งเวลาดังรูปที่ 4.2 โดยที่สามารถตั้งเวลาโดยกดหมายเลขจากคีย์แพด ถ้าตั้งเวลาผิดพลาดจะแสดงผลว่า Time Error ดังรูปที่ 4.3 และเมื่อกดปุ่ม R/Met จะเป็นการตั้งวันที่ดังรูปที่ 4.4 ถ้าตั้งเวลาผิดพลาดจะแสดงผลว่า Time Error รูปที่ 4.5 จะตั้งทำการตั้งค่าใหม่



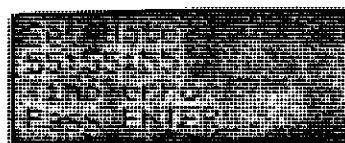
รูปที่ 4.2 คำสั่งการตั้งเวลา



รูปที่ 4.3 แสดงผลของการตั้งเวลาผิด

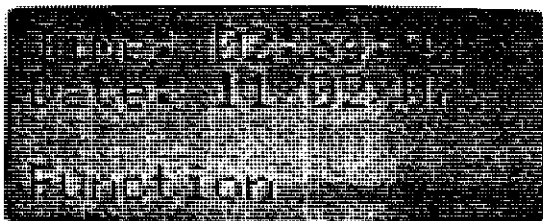


รูปที่ 4.4 คำสั่งการตั้งวันที่



รูปที่ 4.5 แสดงผลของการตั้งวันที่ผิด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.6 แสดงเวลาและวันที่

2. การตั้งค่าตำแหน่งของมาตรวัดน้ำไร้สาย

เมื่อกดปุ่ม R/Met ที่คีย์แพคเกจจะปรากฏเมนูการตั้งค่าตำแหน่งของมาตรวัดน้ำไร้สายดังรูปที่ 4.7 กำหนด ตำแหน่งของมาตรวัดน้ำไร้สายดังรูปที่ 4.8 เมื่อกด Send ตำแหน่งของมาตรวัดน้ำไร้สายที่กำหนดจะแสดงที่ตัวมาตรวัดน้ำไร้สาย ดังรูปที่ 4.9



รูปที่ 4.7 คำสั่งในการตั้งค่าตำแหน่งของมาตรวัดน้ำไร้สาย



รูปที่ 4.8 แสดงตำแหน่งของมาตรวัดน้ำไร้สายที่กำหนด



รูปที่ 4.9 แสดงตำแหน่งที่มาตรวัดน้ำไร้สาย

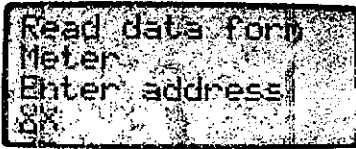
จากรูปที่ 4.7 เป็นคำสั่งในการตั้งค่าตำแหน่งของมาตรวัดน้ำไร้สาย จากรูปที่ 4.8 แสดงตำแหน่งของมาตรวัดน้ำไร้สาย จากรูปที่ 4.9 จะแสดงตำแหน่งของมาตรวัดน้ำไร้สายที่ตัวมาตรวัดน้ำไร้สาย

3. การอ่านข้อมูลจากมาตรวัดน้ำไร้สาย

เมื่อเข้าไปยังคำสั่งการติดต่อระหว่างอุปกรณ์เก็บข้อมูลกับมาตรวัดน้ำไร้สายเพื่อขอข้อมูลในการวัดปริมาณน้ำที่ตัวมาตรวัดน้ำไร้สายอ่านได้ โดยคำสั่งต่างๆ เมื่อกดปุ่ม R/Met จะเป็นคำสั่งที่ต้องการดูค่าปริมาณน้ำที่ใช้ไปมาตรวัดน้ำไร้สายระบุตำแหน่งของมาตรวัดน้ำไร้สายสองหลัก เช่น 03, 01 ดังรูปที่ 4.10 โดยจะระบุตำแหน่งของมาตรวัดน้ำไร้สายที่ต้องการติดต่อแล้วกดปุ่ม Send หลังจากนั้นอุปกรณ์เก็บข้อมูลจะทำการรอการตอบสนองจากมาตรวัดน้ำไร้สาย 25 ms (เวลา 25 millisecond เป็นเวลาที่ใช้ประมวลผล (Process) ของอุปกรณ์เก็บข้อมูลกับมาตรวัดน้ำไร้สายในการส่งข้อมูล) ถ้ายังไม่มีการตอบสนองกลับมายังมาตรวัดน้ำไร้สาย หรืออุปกรณ์เก็บข้อมูล ไม่สามารถรับการตอบสนองจากมาตรวัดน้ำไร้สายได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

วัดน้ำไร้สายได้ อุปกรณ์เก็บข้อมูลจะทำการแสดงผลว่า Can't find address ดังรูปที่ 4.11 ทำการกำหนดตำแหน่งของมาตรวัดน้ำไร้สาย ดังรูปที่ 4.12 และเมื่ออุปกรณ์เก็บข้อมูลได้รับการตอบสนองจากมาตรวัดน้ำจะแสดงผลเป็นจำนวนปริมาณการใช้น้ำของมาตรวัดน้ำไร้สายดังรูปที่ 4.13



รูปที่ 4.10 คำสั่งการอ่านข้อมูลจากมาตรวัดน้ำไร้สายที่ระบุตำแหน่ง

รูปที่ 4.11 เป็นการแสดงผลเมื่อไม่ได้รับการตอบสนองของมาตรวัดน้ำไร้สายตำแหน่ง 07



รูปที่ 4.12 กำหนดตำแหน่งของมาตรวัดน้ำไร้สายที่ต้องการ

รูปที่ 4.13 เป็นการแสดงผลเมื่อได้รับการตอบสนองและปริมาณการใช้น้ำที่มาตรวัดน้ำไร้สายที่ระบุตำแหน่ง 00 ที่วัดได้

4. การอ่านข้อมูลจากหน่วยความจำ

เมื่อกดปุ่ม R/Mem ที่คีย์แพดจะปรากฏเมนูในการตั้งค่าตำแหน่งของมาตรวัดน้ำไร้สายดังรูปที่ 4.14 เพื่อทำการระบุค่าตำแหน่งของมาตรวัดน้ำไร้สายที่ต้องการจะคูค่านั้นกดซ้ำที่ปุ่ม R/Mem ก็จะแสดงค่าที่บันทึกไว้ในหน่วยความจำออกมาดังแสดงดังรูปที่ 4.15

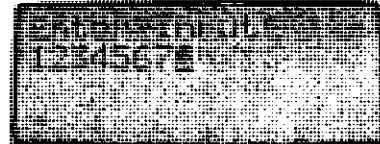
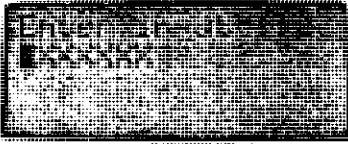


รูปที่ 4.14 คำสั่งในการอ่านข้อมูลจากหน่วยความจำ

รูปที่ 4.15 แสดงตำแหน่งของมาตรวัดน้ำไร้สายและปริมาณน้ำที่ใช้ไป

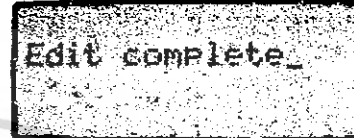
5. การแก้ไขค่าที่มาตรวัดน้ำไร้สาย

เมื่อกดปุ่ม Set /Met ที่คีย์แพดจะปรากฏเมนูในการแก้ไขค่าแสดงดังรูปที่ 4.16 เมื่อทำการระบุตัวเลขในการแก้ไขค่าเสร็จก็จะแสดงดังรูปที่ 4.17 จากนั้นกดปุ่ม Send ก็จะแสดงค่าตัวเลขที่ทำการแก้ไขค่าที่ตัวมาตรวัดน้ำไร้สายแสดงดังรูปที่ 4.18 เมื่อกระบวนการทำงานสิ้นสุดลงก็จะแสดงผลว่าการทำงานสมบูรณ์แล้วดังรูปที่ 4.19



รูปที่ 4.16 คำสั่งในการแก้ไขค่าที่มาตรวัดน้ำไร้สาย

รูปที่ 4.17 แสดงการระบุตัวเลข

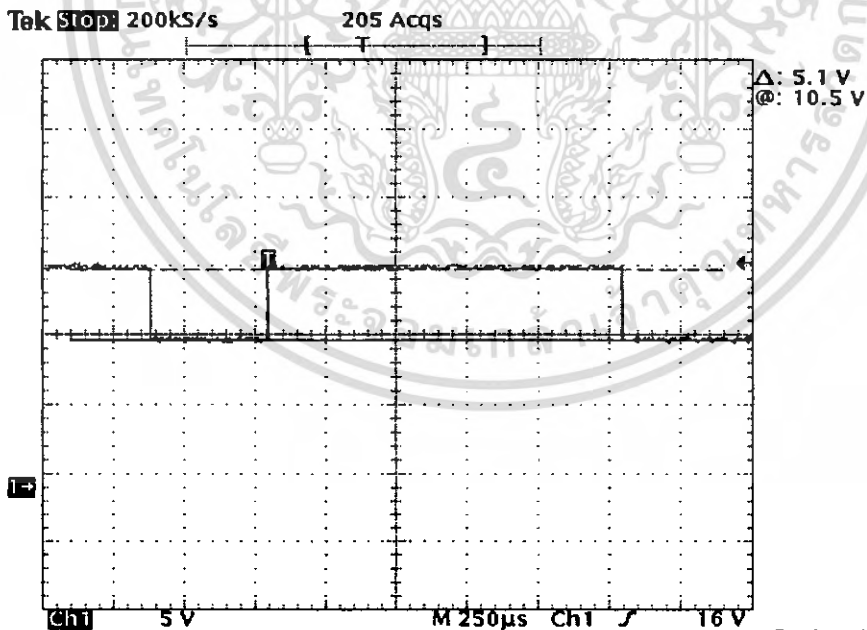


รูปที่ 4.18 แสดงค่าตัวเลขที่แก้ไขที่มาตรวัดน้ำไร้สาย

รูปที่ 4.19 แสดงการทำงานเสร็จสิ้นสมบูรณ์

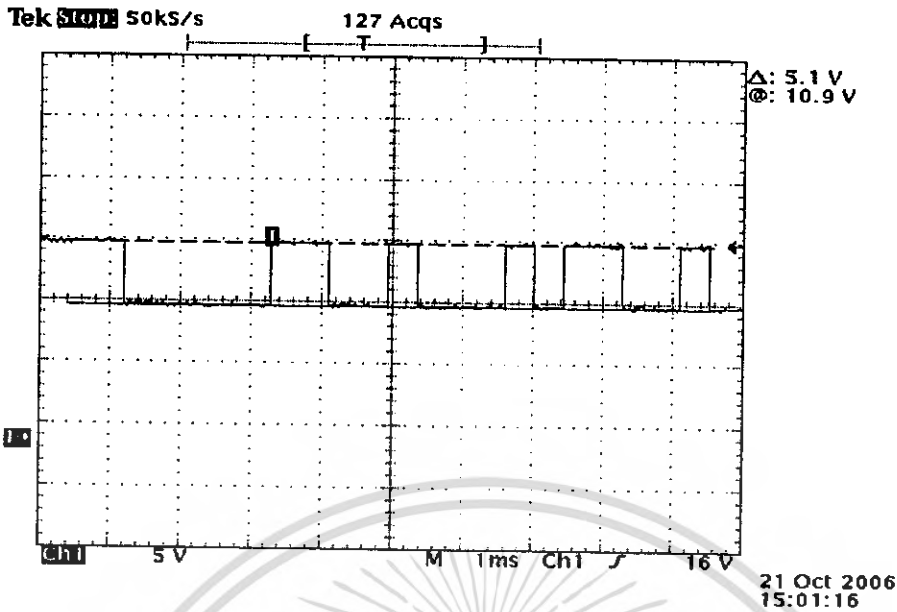
4.2 มาตรวัดน้ำไร้สาย

ในการทดลองนี้จะเก็บผลการติดต่อแบบมีสายผ่านขา Tx และ Rx ของตัวไมโครคอนโทรเลอร์ โดยสัญญาณจะแบ่งเป็น 2 ช่วงคือ ช่วงส่งสัญญาณออกจากอุปกรณ์เก็บข้อมูลไปยังมาตรวัดน้ำไร้สาย และสัญญาณออกจากมาตรวัดน้ำไร้สายไปยังอุปกรณ์เก็บข้อมูล ดังรูปที่ 4.20 และ รูปที่ 4.21



21 Oct 2006
14:49:33

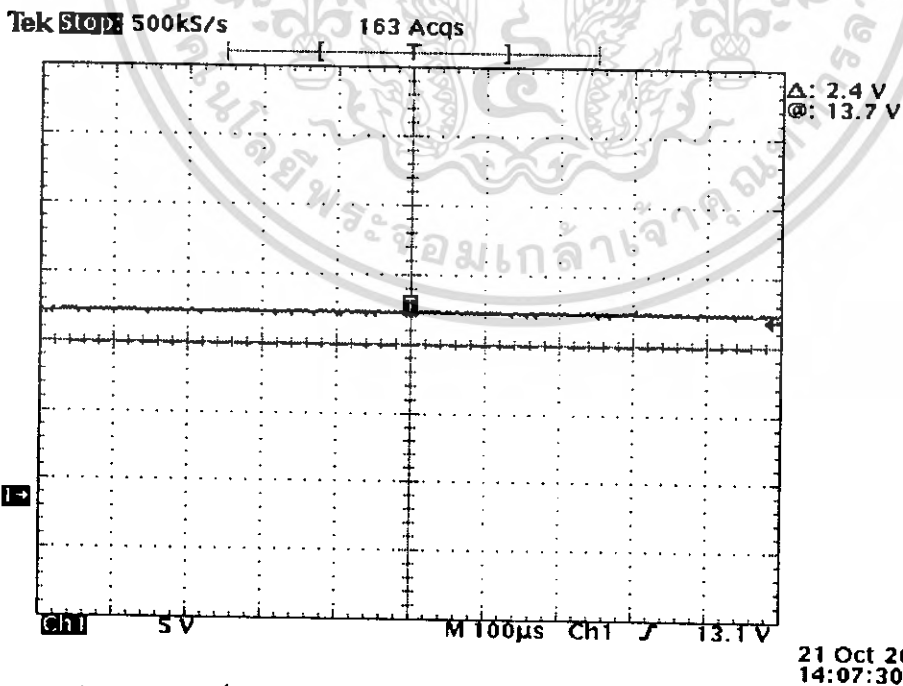
รูปที่ 4.20 เป็นรูปสัญญาณที่อุปกรณ์เก็บข้อมูลเรียกมาตรวัดน้ำไร้สายที่ต้องการติดต่อโดยเรียกตามรหัสเครื่อง



รูปที่ 4.21 เป็นรูปสัญญาณที่มาตรวัดนำไว้สายตามรหัสที่อุปกรณ์เก็บข้อมูลส่งสัญญาณกลับมา

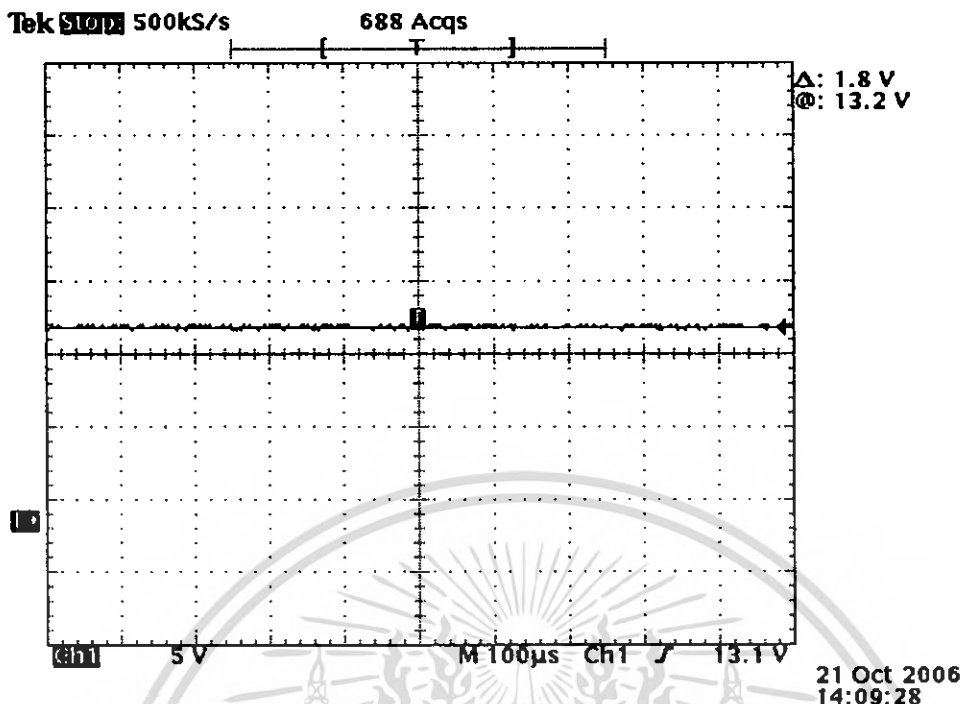
4.2.1 การเปรียบเทียบสัญญาณต่างๆของวงจรเปรียบเทียบแรงดัน

ในส่วนนี้จะทำการวัดความต่างกันของสัญญาณที่ได้จากตัวตรวจจับสนามแม่เหล็กและออปปี้ เมื่อไม่มีสนามแม่เหล็กมาตัดผ่านตัวตรวจจับสนามแม่เหล็กและขณะที่มีสนามแม่เหล็กมาตัดผ่านตัวตรวจจับสนามแม่เหล็กดังแสดงในรูปที่ 4.22 และ รูปที่ 4.23

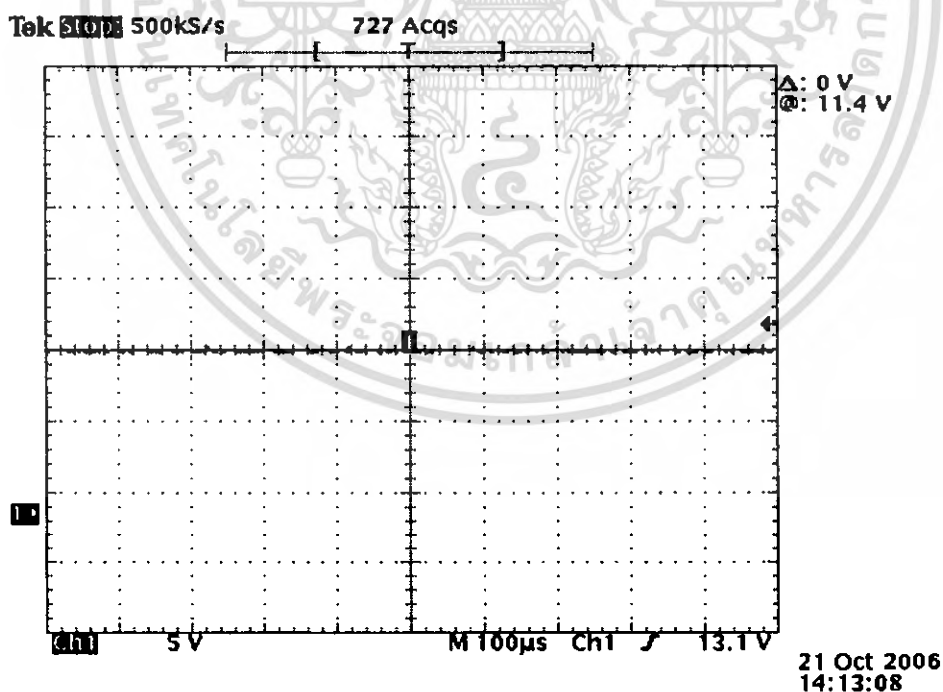


รูปที่ 4.22 รูปสัญญาณเอาต์พุตของตัวตรวจจับสนามแม่เหล็กขณะที่ไม่มีสนามแม่เหล็กมาตัดผ่านตัวตรวจจับสนามแม่เหล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

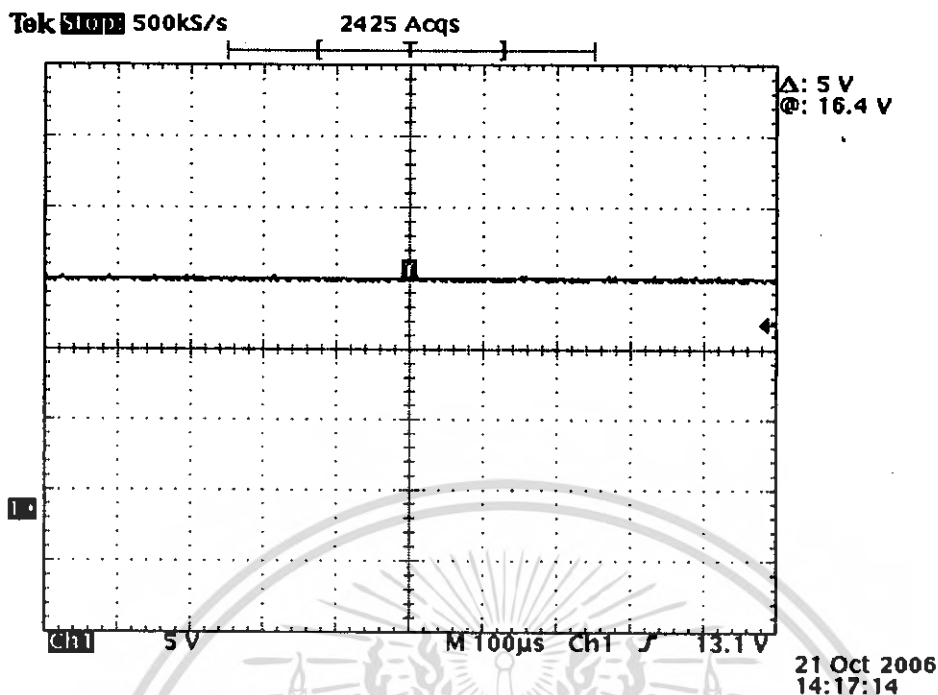


รูปที่ 4.23 รูปสัญญาณเอาต์พุตของตัวตรวจจับสนามแม่เหล็ก ขณะที่แม่เหล็กมาตัดผ่านตัวตรวจจับสนามแม่เหล็ก



รูปที่ 4.24 รูปสัญญาณเอาต์พุตของอปแอมป์ เมื่อแรงคันทึษา 2 (เอาต์พุตของตัวตรวจจับสนามแม่เหล็ก) มากกว่าแรงคันทึษา 3 (แรงคันทึษาของความค้ำานทานปรับค่าได้)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.25 รูปสัญญาณเอาต์พุตของออปแอมป์ เมื่อแรงดันที่ขา 2 (เอาต์พุตของตัวตรวจจับสนามแม่เหล็ก) น้อยกว่าแรงดันขา 3 (แรงดันของความต้านทานปรับค่าได้)

4.2.2 การเปรียบเทียบจำนวนครั้งที่แม่เหล็กตัดผ่านตัวตรวจจับสนามแม่เหล็ก จะเปรียบเทียบ กับ ปริมาณน้ำ 0.01 ลูกบาศก์เมตร โดยจะแสดงในตารางที่ 4.1

ตารางที่ 4.1 ตัวอย่างการเก็บผล ปริมาณน้ำตั้งแต่ 0.00 - 0.01 และจำนวนที่แม่เหล็กตัดผ่านตัวตรวจจับ สนามแม่เหล็ก

ครั้งที่	จำนวนแม่เหล็กตัดผ่านตัวตรวจจับสนามแม่เหล็ก (ครั้ง)	ปริมาณน้ำ (0.01 ลูกบาศก์เมตร)	ปริมาณน้ำ (ลิตร)	เวลา (วินาที)
1	000 - 402	0.00 - 0.01	0 - 10	42
2	402 - 800	0.01 - 0.02	10 - 20	44
3	800 - 1200	0.02 - 0.03	20 - 30	42
4	1200 - 1598	0.03 - 0.04	30 - 40	39
5	1598 - 2000	0.04 - 0.05	40 - 50	39
เฉลี่ย	400	0.01	10	42

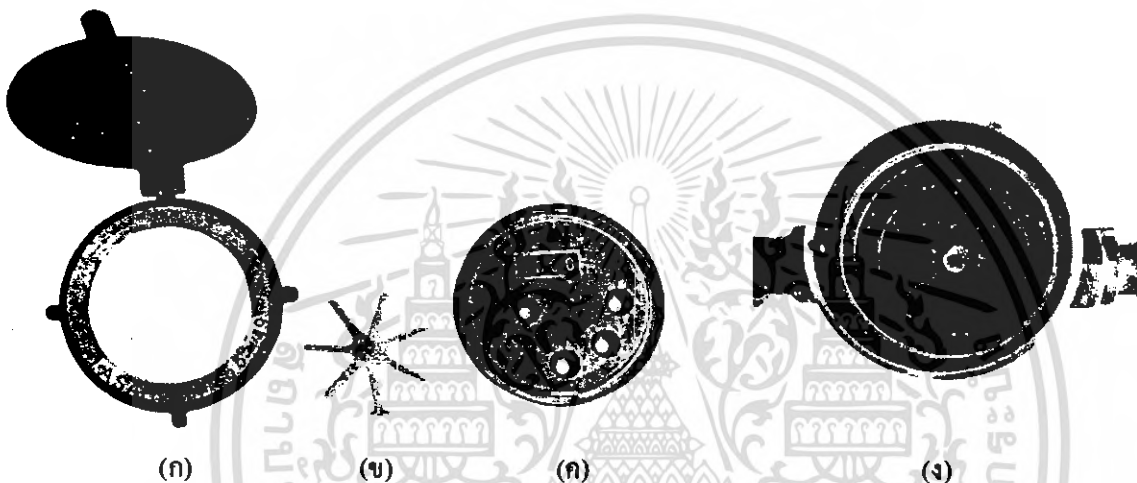
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ดังนั้นจำนวนที่แม่เหล็กตัดผ่านตัวตรวจจับสนามแม่เหล็กในปริมาณน้ำ 1 ลิตร

$$= \frac{\text{จำนวนเฉลี่ยที่แม่เหล็กตัดผ่านตัวตรวจจับสนามแม่เหล็ก}}{\text{ปริมาณน้ำเฉลี่ย}}$$

จะได้จำนวนครั้งแม่เหล็กตัดผ่านตัวตรวจจับสนามแม่เหล็กในปริมาณน้ำ 1 ลิตร เท่ากับ 40 ครั้ง จากนั้นจะนำค่าที่ได้ไปใช้ทศในการนับของโปรแกรมเพื่อให้จำนวนหลักของมาตรวัดน้ำ ไร้สายและมาตรวัดน้ำของการประปามีค่าเท่ากัน

4.3 ส่วนประกอบของมาตรวัดน้ำ

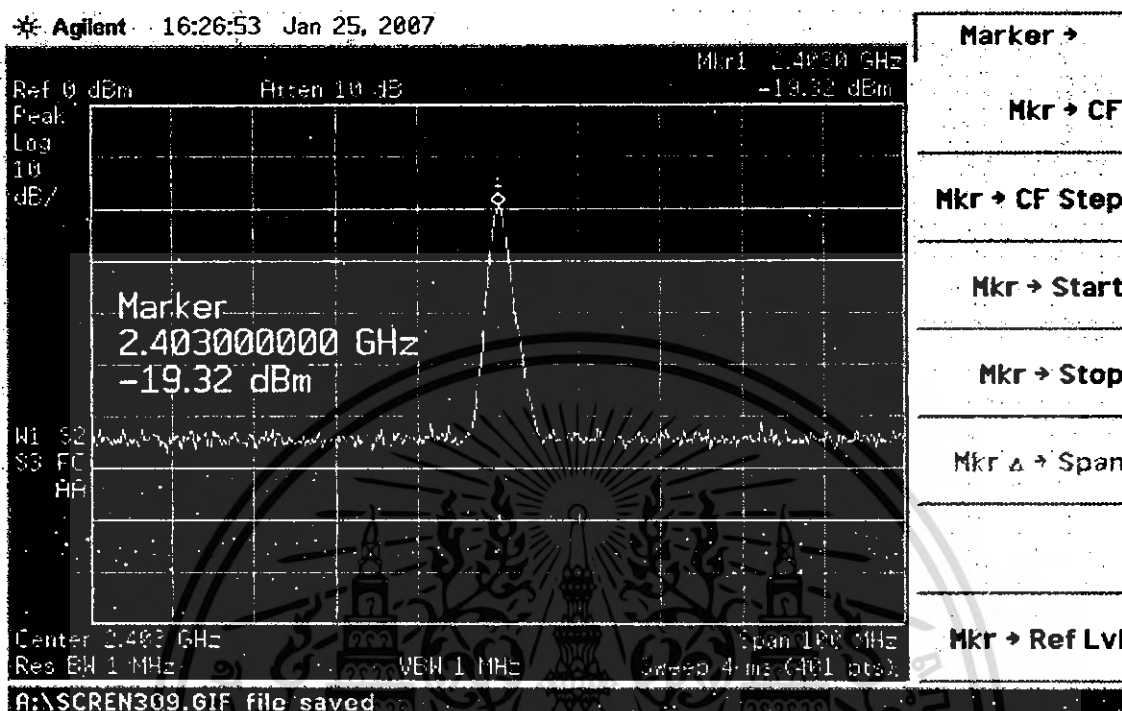


รูปที่ 4.26 ส่วนประกอบของมาตรวัดน้ำ

- (ก) ฝาปิด
- (ข) ใบพัด
- (ค) อุปกรณ์แสดงปริมาณน้ำที่ใช้
- (ง) ตัวถังของมาตรวัดน้ำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.4 การรับส่งข้อมูลแบบไร้สาย

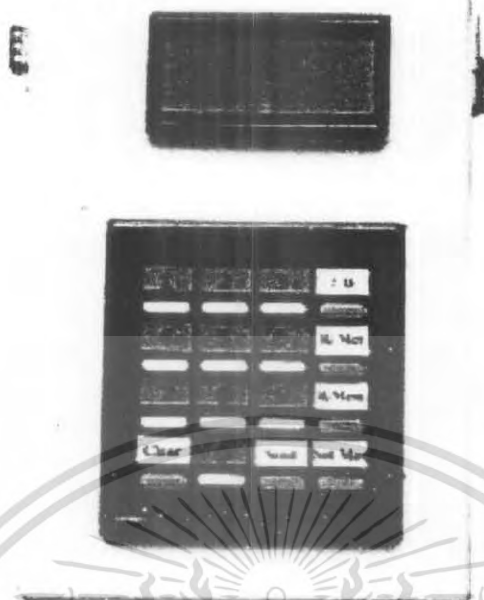


รูปที่ 4.27 สัญญาณสเปกตรัมของตัวโมดูลรับ-ส่ง (TRW-2.4 GHz)

4.5 การต่อใช้งานจริง

จากรูปทั้งหมดจะเป็นการติดตั้งเพื่อต่อใช้งานจริงในระบบมาตรวัดน้ำไร้สาย ซึ่งประกอบไปด้วย อุปกรณ์เก็บข้อมูล มาตรวัดน้ำไร้สาย สายนำสัญญาณ และการติดตั้ง Hall Effect Sensor เพื่อใช้ในการตรวจจับสนามแม่เหล็กที่ติดไว้กับตัวใบพัดของมาตรวัดน้ำ

รูปถ่ายเก็บข้อมูล

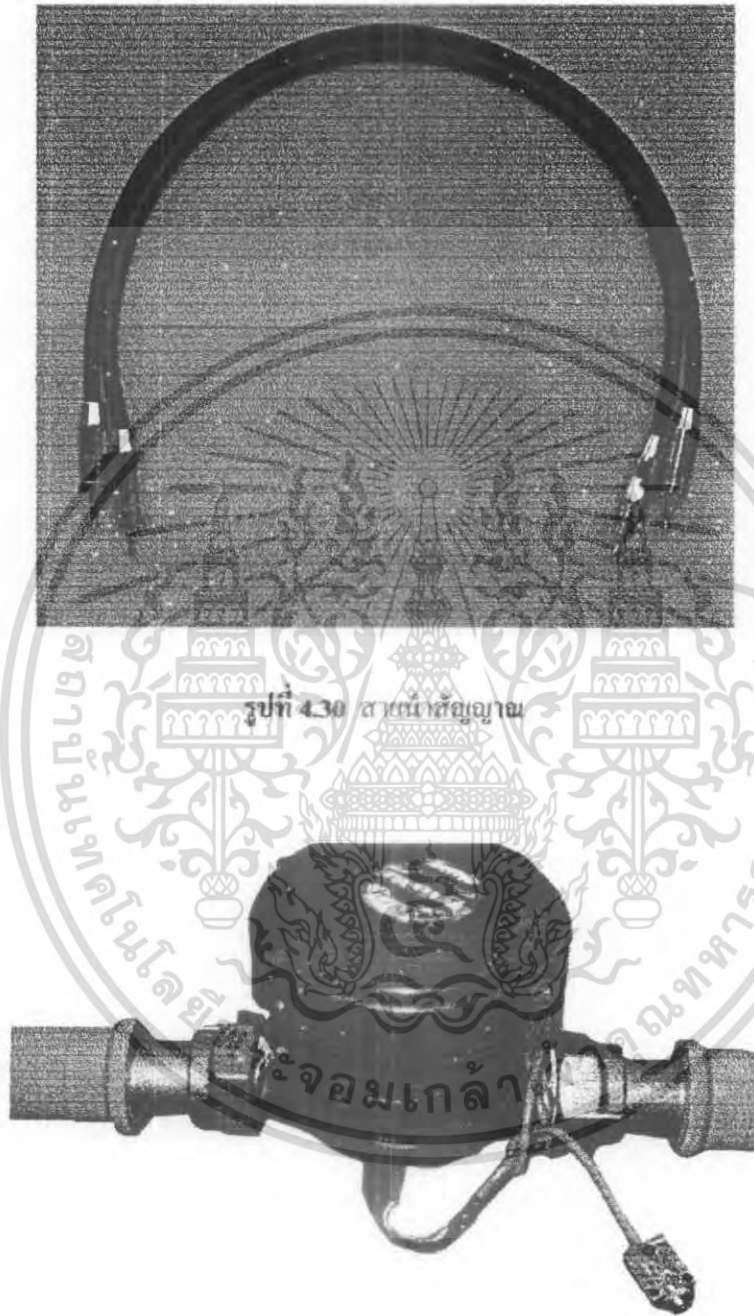


รูปที่ 4.28 อุปกรณ์เก็บข้อมูล



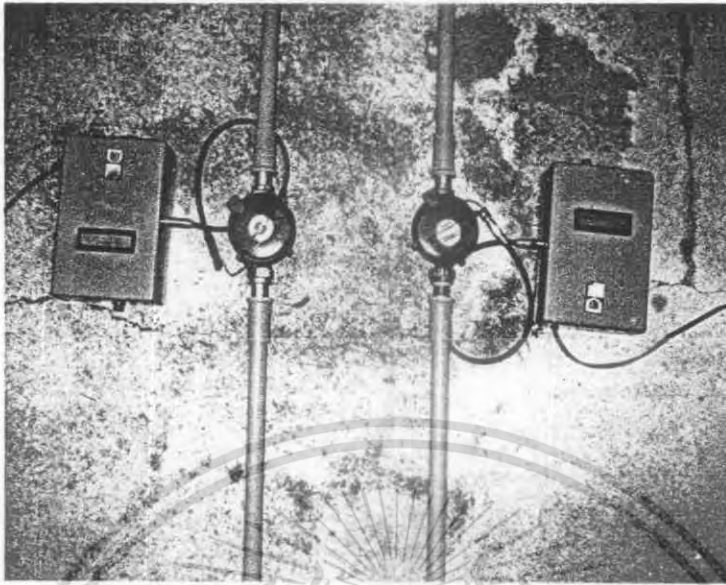
รูปที่ 4.29 มาตรฐานไร้สาย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.31 การติดตั้ง Hall Effect Sensor เข้ากับมาตรวัดน้ำจริง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.32 การต่อใช้งานมาตรวัดน้ำไร้สายกับมาตรวัดน้ำจริง

4.6 การแสดงผลทางหน้าจอคอมพิวเตอร์

Water meter

Address

Volume Unit

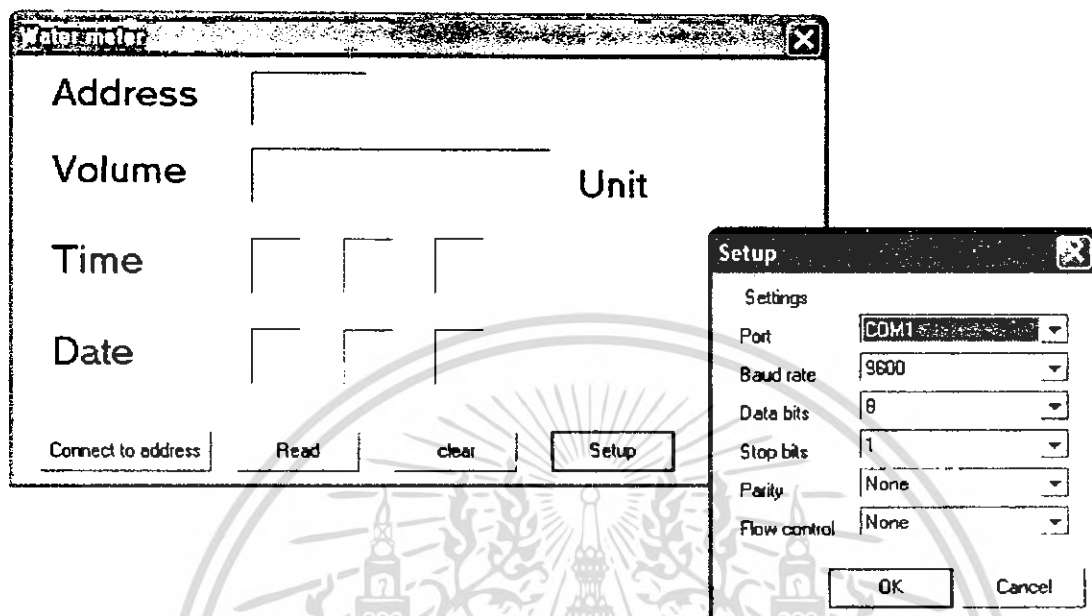
Time

Date

รูปที่ 4.33 การแสดงผลหน้าจอหลัก

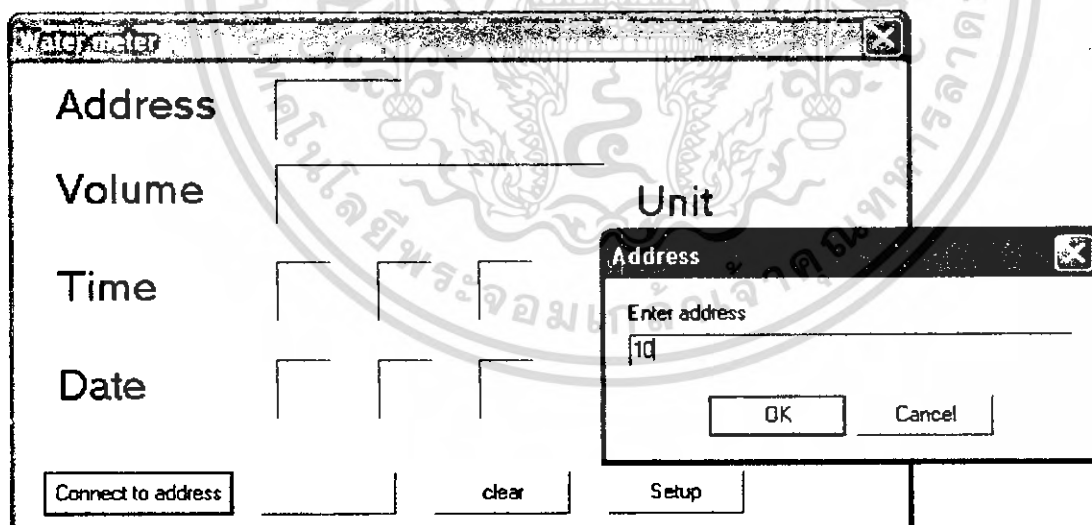
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 4.33 เมื่อทำการเปิด โปรแกรมที่จะรับบนหน้าจอคอมพิวเตอร์ซึ่งเชื่อมต่อกับอุปกรณ์ เก็บข้อมูลจะแสดงผลหน้าจอหลักออกมา



รูปที่ 4.34 การ Setup

จากรูปที่ 4.34 คลิกที่ปุ่ม Setup เพื่อที่จะเซตค่าต่างๆ



รูปที่ 4.35 แสดงการระบุตำแหน่งของมาตรวัดน้ำไร้สาย

จากรูปที่ 4.35 จากนั้นกดปุ่ม Connect to address จะขึ้นหน้าจอให้ทำการระบุตำแหน่งของ มาตรวัดน้ำไร้สายลงไปจากนั้นกดปุ่ม OK

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Water meter

Address 10

Volume _____ Unit

Time _____ Connecting

Date _____

Connect to address Read clear Setup

รูปที่ 4.36 แสดงค่าตำแหน่งของมาตรวัดน้ำไร้สาย

จากรูปที่ 4.36 จะแสดงค่าที่ทำการระบุตำแหน่งของมาตรวัดน้ำไร้สาย

Water meter

Address 10

Volume 0000.000 Unit

Time 14 03 55 Connecting

Date 14 02 07

Connect to address Read clear Setup

รูปที่ 4.37 ค่าต่างๆของมาตรวัดน้ำไร้สายที่ระบุตำแหน่งโดยอ่านจากอุปกรณ์เก็บข้อมูล

จากรูปที่ 4.37 แสดงตำแหน่งของมาตรวัดน้ำไร้สาย จำนวนปริมาณน้ำที่ใช้ไปและแสดงเวลากับ

วันที่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Water meter
✕

Address

Volume Unit

Time Connecting

Date

รูปที่ 4.38 การเคลียร์ (Clear) ข้อมูล

จากรูปที่ 4.38 เป็นการเคลียร์ข้อมูลที่ทำให้การอ่านค่าได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5 บทวิจารณ์และสรุป

ในการสร้างระบบมาตรวัดน้ำไร้สายต้องอาศัยความรู้ในหลายด้าน ไม่ว่าจะเป็นการเขียนโปรแกรมคอมพิวเตอร์โดยใช้ภาษาซี เกี่ยวกับมาตรวัดน้ำ ตัวตรวจจับสนามแม่เหล็ก การรับส่งข้อมูลแบบไร้สาย การแสดงผลทางหน้าจอแสดงผล

จากการทดลองจะพบว่าในส่วนของตัวตรวจจับสนามแม่เหล็ก จะทำงานได้อย่างถูกต้องหากแม่เหล็กมีสภาพสมบูรณ์และวางอยู่ในตำแหน่งตัวตรวจจับสนามแม่เหล็กตรวจจับได้ ในส่วนของโปรแกรม สามารถทำงานได้อย่างถูกต้อง และในส่วนการแสดงผลจะสามารถแสดงผลได้อย่างมีประสิทธิภาพพอสมควร โดยการเทียบกับตัวมาตรวัดน้ำจริง นอกจากนี้ในส่วนการเก็บบันทึกข้อมูลจะสามารถเก็บบันทึกข้อมูลได้อย่างถูกต้องหากข้อมูลที่ได้รับมาถูกต้อง สุดท้ายในส่วนของการรับส่งข้อมูลแบบไร้สายสามารถทำงานได้ได้อย่างถูกต้องสมบูรณ์

ปัญหาที่พบในส่วนของตัวตรวจจับสนามแม่เหล็ก ถ้าวางตำแหน่งแม่เหล็กไม่อยู่ในช่วงที่สามารถจะตรวจจับได้จะไม่สามารถทำงานได้ ในส่วนของตัวมาตรวัดน้ำเทียบกับมาตรวัดน้ำไร้สายที่สร้างขึ้น อาจเกิดความคลาดเคลื่อนหรือผิดพลาดจากตัวมาตรวัดน้ำจริงเพราะว่าไม่มีมาตรฐานมารับรองว่ามาตรวัดน้ำไร้สายที่สร้างขึ้นทำงานได้อย่างถูกต้องหรือไม่เพราะว่าในส่วนนี้ได้ทำการสร้างขึ้นเองโดยเทียบกับมาตรวัดน้ำจริง นอกจากนี้ในตัวมาตรวัดน้ำจะต้องทำการแกะอุปกรณ์ที่อยู่ข้างในออกมาเพื่อมาติดแผ่นแม่เหล็กอาจทำให้สภาพมาตรวัดน้ำที่แกะมีสภาพไม่สมบูรณ์เหมือนกับของเดิมเลยทำให้การแสดงผลอาจเกิดความคลาดเคลื่อนและผิดพลาดขึ้นได้ ส่วนของการรับ-ส่งข้อมูลแบบไร้สายนั้นจะสามารถรับ-ส่งข้อมูลได้ไกลสุดประมาณ 25 เมตรถ้าไกลกว่านี้จะไม่สามารถรับส่งข้อมูลได้

หนังสืออ้างอิง

- [1] สำนักงานมาตรฐานผลิตภัณฑ์อุตสาหกรรม. “มาตรฐานน้ำดื่มด้วยเกลือวานิลลากลูบูล.” กรุงเทพฯ : กระทรวงอุตสาหกรรม. 2538.
- [2] รังสรรค์ ศรีสาร. “เครื่องมือวัดสนามแม่เหล็ก.” [Online]. เข้าถึงได้จาก : <http://www.ku.ac.th/schoolnet/socet3/saowalak/hall/hall.htm>. 2549.
- [3] นคร ภักดีชาติ และชัยวัฒน์ ลิ้มพจรวิไล. “การทดลองและใช้งานไมโครคอนโทรลเลอร์ MCS-51 ด้วยโปรแกรมภาษา C ฉบับ AT89Cx051.” กรุงเทพฯ : อินโนเวตีฟ เอ็กเซอร์เมนส์.
- [4] ชีรวัดน์ ประกอบผล. “ภาษาแอสเซมบลี สำหรับ MCS – 51.” กรุงเทพฯ : สมาคมส่งเสริมเทคโนโลยี (ไทย – ญี่ปุ่น). 2546.
- [5] ชีรวัดน์ ประกอบผล. “การประยุกต์ใช้งานไมโครคอนโทรลเลอร์.” พิมพ์ครั้งที่ 8. กรุงเทพฯ : สมาคมส่งเสริมเทคโนโลยี (ไทย – ญี่ปุ่น). 2547.
- [6] สมยศ จุณณะปิติ. “การใช้งานไมโครคอนโทรลเลอร์ตระกูล MCS-51.” พิมพ์ครั้งที่ 5. กรุงเทพฯ : คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าลาดกระบัง. 2546.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมในส่วนของตัวเก็บบันทึกข้อมูล

```
#include<reg52.h>
#include<LCD_1.h>
#include<Time.h>
#include<trw24.h>
#include<stdio.h>
#define AT24C16_ID 0xA0

unsigned char AT24C16_rd(unsigned char num)
{
    unsigned char dat = 0;
    unsigned char addr_h,addr_l;
    addr_h = 0;
    addr_h = addr_h<<1;
    addr_h = AT24C16_ID | addr_h;
    addr_l = num;
    i2c_start();
    i2c_write(addr_h);
    i2c_write(addr_l);
    i2c_start();
    i2c_write(addr_h+1);
    dat = i2c_read();
    i2c_NACK();
    i2c_stop();
    return(dat);
}
/***** Function convert ascii to hex and send to serial port *****/
void AT24C16_wr(unsigned char dat,unsigned char num)
{
    unsigned char addr_h,addr_l;
    addr_h = 0;
    ascii to hex
    addr_h = addr_h<<1;
    addr_h = AT24C16_ID | addr_h;
    addr_l = num;
    i2c_start();
    i2c_write(addr_h);
    i2c_write(dat);
    i2c_stop();
    delay(3);
}
void wrdata_eeporm(unsigned char addr)
{
    int i;
    unsigned char init;
    init=((addr*0x0D)-0x0D);
    for(i=0;i<=12;i++)
    {
        AT24C16_wr(array[i],init);
        init++;
    }
}

void display_init()
{
    Lcd_puts(0xD0,"Function");
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void set_data(void)
{
    int i;
    char addr[2]={0};
    char addr1;
    lcd_clear();
    lcd_command(0x0F);
    lcd_puts(0x80,"Set meter");
    lcd_puts(0xC0,"Enter address");
    lcd_puts(0x90,"XX");
    lcd_command(0x90);
    while(!sw);
    while(sw);
    if(P0!=0x0C)
    {
        for(i=0;i<=1;i++)
        {
            while(!sw);
            while(sw);
            while(convert(P0)==0x0F);
            lcd_text(convert(P0)+'0');
            addr[i]=(convert(P0));
        }
        while(!sw);
        while(sw);
        addr1=((addr[0]<<4)|addr[1]);
        lcd_clear();
        lcd_puts(0x80,"Enter input");
        lcd_puts(0xC0,"XXXXXXXX");
        lcd_command(0xC0);
        for(i=6;i<=12;i++)
        {
            while(!sw);
            while(sw);
            while(convert(P0)==0x0F);
            lcd_text(convert(P0)+'0');
            array[i]=convert(P0);
        }
        while(!sw);
        while(sw);
        lcd_clear();
        SetMode_TRW24(1,0xA0);
        send_command(0x03,addr1);
        SetMode_TRW24(0,0xA0);
        lcd_puts(0xC0," Please Wait");
        dmsec(1000);
        if(DR1==1)
        {
            lcd_clear();
            lcd_puts(0xC0,"Edit
complete");
            while(!sw);
            while(sw);
        }
        else
        {
            lcd_clear();
            lcd_puts(0xC0,"Can't find");
            while(!sw);
            while(sw);
        }
    }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

}
void Read_TRW24(void)
{
    unsigned char i,j,Temp;
    bit Out;
    Data = 1;
    for(i=6;i<=12;i++)
    {
        Temp=0x00;
        for (j=0;j<=7;j++)
        {
            Temp = Temp << 1;
            CLK1 = 1;
            Wait(50);
            Out = Data;
            if (Out){Temp = Temp + 0x01;}
            CLK1 = 0;
            Wait(50);
        }
        array[i]=Temp;
    }
}
void read_data(void)
{
    int i;
    char posi;
    char rx_addr[2];
    char rx_addr1;
    TI=0;
    lcd_clear();
    lcd_puts(0x80,"Read data form");
    lcd_puts(0xC0,"Meter");
    lcd_puts(0x90,"Enter address");
    lcd_puts(0xD0,"XX");
    lcd_command(0x0F);
    lcd_command(0xD0);
    while(!sw);
    while(sw);
    if(P0!=0x0C)
    {
        for(i=0;i<=1;i++)
        {
            while(!sw);
            while(sw);
            lcd_text((convert(P0)+'0'));
            rx_addr[i]=(convert(P0));
        }
        rx_addr1=((rx_addr[0]<<4)|rx_addr[1]);
        while(!sw);
        while(sw);
        SetMode_TRW24(1,0xA0);
        send_command(0x30,rx_addr1);
        SetMode_TRW24(0,0xA0);
        lcd_clear();
        lcd_puts(0xC0," Please Wait");
        dmsec(1000);
        if(DR1==1)
        {
            Read_TRW24();
            lcd_clear();
            lcd_puts(0x80,"Adress ");

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        lcd_text(rx_addr[0]+'0');
lcd_text(rx_addr[1]+'0');
        lcd_puts(0xC9, "Unit");
        lcd_puts(0x90, "Time");
        lcd_puts(0xD0, "Date");
        array[0] = DS1307_rd(0x00);
array[1] = DS1307_rd(0x01);
array[2] = DS1307_rd(0x02);
        array[3] = DS1307_rd(0x04);

array[4] = DS1307_rd(0x05);
array[5] = DS1307_rd(0x06);

        /*-----Show Time-----
*/
        lcd_origin();
        lcd_command(0x96);
        send_to_lcd(array[2]);
        lcd_text(':');
        send_to_lcd(array[1]);
        lcd_text(':');
        send_to_lcd(array[0]);
        /*-----Show Date-----
*/
        lcd_origin();
        lcd_command(0xD6);
        send_to_lcd(array[3]);
        lcd_text('/');
        send_to_lcd(array[4]);
        lcd_text('/');
        send_to_lcd(array[5]);

        posi=0xCF;
for(i=6;i<=8;i++)
    {
        lcd_command(posi);
        if(array[i]==0x00){array[i]='0';}
        lcd_text(array[i]);
        posi--;
    }
lcd_command(0xCC);
lcd_text('.');
posi=0xCB;
for(i=9;i<=12;i++)
    {
        lcd_command(posi);
        if(array[i]==0x00){array[i]='0';}
        lcd_text(array[i]);
        posi--;
    }

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    }
    wrdata_eeporm(rx_addr1);
} // end if

else
{
    lcd_clear();
    lcd_puts(0x80,"Can't find");
    lcd_puts(0xC0,"address ");
}
while(!sw);
while(sw);
}

lcd_clear();
}
}

```

```
void rddata_eeporm()
```

```

{
    int i;
    unsigned char init=0x00;
    char dat[13];
    char rx_addr[2];
    char rx_addr1;

    lcd_clear();
    lcd_puts(0x80,"Read data form");
    lcd_puts(0xC0,"Memory");
    lcd_puts(0x90,"Enter address");
    lcd_puts(0xD0,"XX");
    lcd_command(0x0F);
    lcd_command(0xD0);
    while(!sw);
    while(sw);

if (P0!=0x0C)
{
    for(i=0;i<=1;i++)
    {
        while(!sw);
        while(sw);
        lcd_text((convert(P0)+'0'));
        rx_addr[i]=(convert(P0));
    }
    rx_addr1=((rx_addr[0]<<4){rx_addr[1]);
    init=((rx_addr1*0x0D)-0x0D);
    for(i=0;i<=12;i++)
    {
        dat[i]= AT24C16_rd(init);
        init++;
    }
    lcd_clear();
    lcd_puts(0x80,"Adress ");
    lcd_text(rx_addr[0]+'0'); lcd_text(rx_addr[1]+'0');
    lcd_puts(0xC9,"Unit");
    lcd_puts(0x90,"Time");
    lcd_puts(0xD0,"Date");

    /*-----Show Time-----
*/

    lcd_origin();
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        lcd_command(0x96);

        send_to_lcd(dat[2]);

        lcd_text(':');
        send_to_lcd(dat[1]);

        lcd_text(':');

        send_to_lcd(dat[0]);

        /*-----Show Date-----

*/

        lcd_origin();

        lcd_command(0xD6);

        send_to_lcd(dat[3]);

        lcd_text('/');

        send_to_lcd(dat[4]);

        lcd_text('/');
        send_to_lcd(dat[5]);

        lcd_command(0xC0);
        for(i=12;i>=9;i--)
        {   if(dat[i]==0)
            {dat[i]='0';}
            lcd_text(dat[i]);
        }
        lcd_text('.');
        lcd_command(0xC5);
        for(i=8;i>=6;i--)
        {   if(dat[i]==0)

            lcd_text(dat[i]); }
            while(!sw);
            while(sw);

            lcd_clear();

}

void edit_time(void)
{
    lcd_clear();
    lcd_puts(0x80,"Set time");
    lcd_puts(0xC0,"Set date");
    lcd_puts(0x90,"Exit");
    while(!sw);
    while(sw);
    switch(P0)
    {
        case 0x03 : set_time(); break;
        case 0x07 : set_date(); break;
        case 0x0C : break;
    }

    lcd_clear();
}
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

void send_to_port(unsigned char value)
{
    unsigned char buf = 0;
    buf = value & 0xF0;
    buf = (buf>>4)|(0x30);

    SBUF= buf ;
    while(~TI);
    TI=0;
    buf = value & 0x0F;
    buf = buf | 0x30;
    SBUF= buf ;
    while(~TI);
    TI=0;
}
char convert2( char num)
{
    switch(num)
    {
        case 0x30 : return(0x00); break;
        case 0x31 : return(0x01); break;
        case 0x32 : return(0x02); break;
        case 0x33 : return(0x03); break;
        case 0x34 : return(0x04); break;
        case 0x35 : return(0x05); break;
        case 0x36 : return(0x06); break;
        case 0x37 : return(0x07); break;
        case 0x38 : return(0x08); break;
        case 0x39 : return(0x09); break;
    }
}
void sevice(void) interrupt 4
{
    unsigned char command;
    char addr,addr1;
    char init=0x00;
    char buf,value;
    int i,j;
    TI=0;
    if(RI)
    {
        command=SBUF;
        RI=0;
        lcd_clear();
        lcd_puts(0x80,"Connecting");

        if(command==0x41)
        {
            SBUF=0x45;
            while(~TI);
            TI=0;
            addr=_getkey();
            addr=convert2(addr);
            addr=(addr<<4);
            addr1=_getkey();
            addr1=convert2(addr1);
            addr=addr|addr1;
            init=((addr*0x0D)-0x0D);
            for(i=6;i<=18;i++)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับใช้ภายในเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        {
            array[i]= AT24C16_rd(init);
            init++;
        }
        j=0;

    for(i=6;i<=11;i++)
    {
        value=array[i];
        buf = value & 0xF0;
            buf = (buf>>4)|(0x30);
                array[j]= buf ;

        j++;
        buf = value & 0x0F;
            buf = buf | 0x30;

            array[j]= buf ;

        j++;
    }
    for( i=18;i>=12;i--)
    {
        SBUF= array[i];
        while(~TI);
        TI=0;
    }
    for( i=0;i<=11;i++)
    {
        SBUF= array[i];
        while(~TI);
        TI=0;
    }
}
if(TI)
{ TI=0;}
}
while(_getkey()!=0x30);
lcd_clear();
}

```

```

void main(void)
{
    TMOD=0x21;
    SCON=0x50;
    TH1=0xFD;
    TL1=0xFD;
    RI=0;
    TI=0;
    EA=1;
    ES=1;
    TR1=1;
    P2=1;
    sw=1;
    DR1=1;
    lcd_init();
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        lcd_command (0x80);
while(1)
{
    display_init();
    switch(P0)
    {
        case 0x07 :   read_data(); break;
        case 0x0F :   set_data(); break;
        case 0x03 :   edit_time(); break;
        case 0x0B :   rddata_eeporm(); break;
        default      :   display_datetime(); break;
    }
}
//end main program
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมในส่วนของตัวควบคุมไมโครคอนโทรลเลอร์

```
#pragma code
#include <reg51.h>
#include<lcd.h>
#include <intrins.h>
/*****
/* Define I/O Port */
/*****
char addr=0x10;
unsigned long int count=7654321;
#define Port_Test P3
sbit Data = P1^0; // Port send data to module 2.4G
chanel 1
sbit CLK1 = P1^3; // Port Clock before send data
chanel 1
sbit CS = P1^2; // Port ch
sbit CE = P1^1;
sbit TX_LED = P3^2;
sbit in = P3^3;

//char show[8]={0x30,0x31,0x32,0x33,0x34,0x35,0x36,0x37};
/*****
/* Subroutine for Delay time */
/*****
void dmsec (unsigned int count) { // mSec Delay 11.0592 Mhz
    unsigned int i; // Keil v7.08
    while (count) {
        i = 225; while (i>0) i--;
        count--;
    }
}
//=====//
void Wait(unsigned int x)
{
    unsigned int i;
    for (i=0;i<x;i++)
    {}
}
//=====//
void CLK_TRW24(void)
{
    CLK1 = 0;
    dmsec(1);
    CLK1 = 1;
    dmsec(1);
}
//=====//
void Write_TRW24(unsigned char Dat)
{
    unsigned char i;
    bit Out;
    for (i=0;i<=7;i++)
    {
        Out = Dat & 0x80;
        Data = Out;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        CLK_TRW24();
        Dat = Dat << 1;
    }
}
char Read_command(void)
{
    unsigned char i,j,Temp;
    bit Out;
    char num[7]={0};
    Data = 1;
    for(i=0;i<=7;i++)
    {
        Temp=0x00;
        for (j=0;j<=7;j++)
        {
            Temp = Temp << 1;
            CLK1 = 1;
            Wait(50);
            Out = Data;
            if (Out){ Temp = Temp + 0x01;}
            CLK1 = 0;
            Wait(50);
        }
        num[i]=Temp;
    }
    return(num[7]);
}
//=====//
void SetMode_TRW24( unsigned char Mode, unsigned char tx_addr)
{
    Wait(500);
    CE = 0;
    CS = 1;
    Write_TRW24(0x8E); /* Reserved for testing */
    Write_TRW24(0x08); /* Reserved for testing */
    Write_TRW24(0x1C); /* Reserved for testing */

    Write_TRW24(0x08); /* Length of Bit Ch 2 */
    Write_TRW24(0x40); /* Length of Bit Ch 1 */

    Write_TRW24(0xC0); /* Address 5 Byte Ch 2 */
    Write_TRW24(0xAA);
    Write_TRW24(0x55);
    Write_TRW24(0xAA);
    Write_TRW24(0x55);

    Write_TRW24(0xAA); /* Address 5 Byte Ch 1 */
    Write_TRW24(0x55);
    Write_TRW24(0xAA);
    Write_TRW24(0x55);
    Write_TRW24(tx_addr);

    Write_TRW24(0xA3); /* Number of Address bit + CRC */
    Write_TRW24(0x6F); /* RF Programming */

    if (Mode == 1) /* Tx Mode */
    {

```

เอกสารนี้เป็นเอกสารที่สงวนลิขสิทธิ์ © 2013 โดย บริษัท เทคโนโลยีสารสนเทศและการสื่อสาร สำนักงานปลัดกระทรวงเทคโนโลยีสารสนเทศและการสื่อสาร
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

//      Data=1; DR1=1; CE=1;
}
else /* Rx_Mode */
{
    Write_TRW24(0x15);
    Data=1; CE=1;
}

CS = 0;
Wait(200);
}
//=====

```

```

void send_TRW24(unsigned long int val,unsigned char addr2)
{

```

```

    int i=0;
    char num[8]={0};
    do
    {
        num[i]={(val%10)+'0'};
        val=val/10;
        i++;
    }while(val!=0);
    Wait(500);
    CS = 0;
    CE = 1;
    Write_TRW24(0xAA); // address 5 byte send
    Write_TRW24(0x55); // address 5 byte send
    Write_TRW24(0xAA); // address 5 byte send
    Write_TRW24(0x55); // address 5 byte send
    Write_TRW24(addr2); // address 5 byte send
    Write_TRW24(num[0]); // data
    Write_TRW24(num[1]); // data
    Write_TRW24(num[2]); // data
    Write_TRW24(num[3]); // data
    Write_TRW24(num[4]); // data
    Write_TRW24(num[5]); // data
    Write_TRW24(num[6]); // data
    Write_TRW24(0x0A); // data
    Wait(250);
    CLK1 = 0;
    CE = 0;
    Wait(250);

```

```

}

void command(void) interrupt 2
{

```

```

    P0=0xFF;
    TX_LED = 0;
    dmsec(200);
    TX_LED = 1;
    dmsec(200);
    SetMode_TRW24(1, addr);
    TX_LED = 0;
    dmsec(200);
    send_TRW24(count, 0xA0);
    TX_LED = 1;

```

เอกสารนี้เป็นเอกสารลิขสิทธิ์สงวนลิขสิทธิ์ไว้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        dmsec(200);
        SetMode_TRW24(0,addr);
        TX_LED = 0;
            dmsec(200);
        TX_LED = 1;
            dmsec(200);
    }

/*****
/* Main Program
*****/

void main (void)
{
    P0=0xFF;
    dmsec(750);
    EX1=1;
    IT1=1;
    EA=1;
    TX_LED = 0;
    dmsec(750);
    TX_LED = 1;
    dmsec(750);
    TX_LED = 0;
    dmsec(750);
    TX_LED = 1;
    Data = 0;
    CLK1= 0;
    CE = 0;
    CS = 0;
    dmsec(1000);
    SetMode_TRW24(0,addr);
    dmsec(200);
/*****STE TRW24G*/
    TMOD = 0x26;
    TH0 = 0xFF;
    TLO = 0xFF;
    TFO = 0;
    TR0 = 1;
/*****set count*****/
    lcd_init();
    lcd_clear();
    lcd_put(lin1,"Water Meter");
        while(1)
        {

                if(TF0)
                {
                    TFO = 0;
                    count++;
                }
            dotdisplay((count%1000),0xCF);
            lcd_command(0xCC);
            lcd_text(0x2E);
            intdisplay((count/1000),0xCB);
            //send_TRW24(count,0xA0);
        }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมในส่วนของตัวโมดูล TRW

```
sbit Data = P2^3; // Port send data to module 2.4G
chanel 1
sbit CLK1 = P2^4; // Port Clock befor send data
chanel 1
sbit DR1 = P2^5; // port select data to chanel1
sbit CS = P2^6; // Port chip select
sbit CE = P2^7; // Port Enable Shift
sbit TX_LED = P3^2;

void dmsec (unsigned int count) { // mSec Delay 11.0592 Mhz
    unsigned int i; // Keil v7.08
    while (count) {
        i = 225; while (i>0) i--;
        count--;
    }
}

//=====//
void Wait(unsigned int x)
{
    unsigned int i;
    for (i=0;i<x;i++)
    {}
}
//=====//

//=====//
void CLK_TRW24(void)
{
    CLK1 = 0;
    dmsec(1);
    CLK1 = 1;
    dmsec(1);
}
//=====//

//=====//
void Write_TRW24(unsigned char Dat)
{
    unsigned char i;
    bit Out;
    for (i=0;i<=7;i++)
    {
        Out = Dat & 0x80;
        Data = Out;
        CLK_TRW24 ();
        Dat = Dat << 1;
    }
}
//=====//

//=====//
void SetMode_TRW24( unsigned char Mode,unsigned char addr)
{
    Wait(500);
    CE = 0;
    CS = 1;
    Write_TRW24(0x8E); /* Reserved for testing */
    Write_TRW24(0x08); /* Reserved for testing */
    Write_TRW24(0x1C); /* Reserved for testing */
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นิยมนำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

Write_TRW24(0x08); /* Length of Bit Ch 2 */
Write_TRW24(0x40); /* Length of Bit Ch 1 */

Write_TRW24(0xC0); /* Address 5 Byte Ch 2 */
Write_TRW24(0xAA);
Write_TRW24(0x55);
Write_TRW24(0xAA);
Write_TRW24(0x55);

Write_TRW24(0xAA); /* Address 5 Byte Ch 1 */
Write_TRW24(0x55);
Write_TRW24(0xAA);
Write_TRW24(0x55);
Write_TRW24(addr);

Write_TRW24(0xA3); /* Number of Address bit + CRC */
Write_TRW24(0x6F); /* RF Programming */

if (Mode == 1) /* Tx Mode */
{
    Write_TRW24(0x14);
//    Data=1; DR1=1; CE=1;
}
else /* Rx_Mode */
{
    Write_TRW24(0x15);
    Data=1; CE=1;
}

CS = 0;
Wait(200);
}
//=====
=====

```

```

void send_TRW24(unsigned long int val , unsigned char addr)
{
    int i=0;
    char num[8]={0};
    do
    {
        num[i]=((val%10)+'0');
        val=val/10;
        i++;
    }while(val!=0);

    Wait(500);
    CS = 0;
    CE = 1;
    Write_TRW24(addr); // address 5 byte send
    Write_TRW24(addr); // address 5 byte send
    Write_TRW24(addr); // address 5 byte send
    Write_TRW24(addr); // address 5 byte send
    Write_TRW24(addr); // address 5 byte send
    Write_TRW24(num[0]); // data
    Write_TRW24(num[1]); // data
    Write_TRW24(num[2]); // data
    Write_TRW24(num[3]); // data
    Write_TRW24(num[4]); // data
    Write_TRW24(num[5]); // data
    Write_TRW24(num[6]); // data
}

```

เอกสารนี้เป็นเอกสารที่จัดทำขึ้นเพื่อใช้ในการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    Write_TRW24(0x0A);    // data
    Wait(250);
    CLK1 = 0;
    CE = 0;
    Wait(250);
}

void send_command(unsigned char command, unsigned char addr)
{
    Wait(500);
    CS = 0;
    CE = 1;
    Write_TRW24(0xAA);    // address 5 byte send
    Write_TRW24(0x55);    // address 5 byte send
    Write_TRW24(0xAA);    // address 5 byte send
    Write_TRW24(0x55);    // address 5 byte send
    Write_TRW24(addr);    // address 5 byte send
    Write_TRW24(command); // data
    Write_TRW24(command); // data
    Write_TRW24(command); // data
    Write_TRW24(command); // data
    Write_TRW24(command); // data
    Write_TRW24(command); // data
    Write_TRW24(command); // data
    Write_TRW24(command); // data
    Wait(250);
    CLK1 = 0;
    CE = 0;
    Wait(250);
}

```



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมแก้ไขค่าเวลา

```
/*-----*/
// Program : I2C DS1307 Real Time Clock
// Description : I2C DS1307 Read and Write time ,address 000
// : Setting by keypad
// Filename : 11401.c
// C complier : RIDE 51 V6.4.35
/*-----*/

sbit sw=P2^0;
#include<I2C_1.h>
#define DS1307_ID 0xD0 // Define constant DS1307_ID
unsigned char control=0; // For keep Date and Time
unsigned char array[13];
/***** Function convert to ascii and send to LCD
*****/
char convert(char word)
{
    switch(word){
        case 0x00 : return(0x01); break;
        case 0x01 : return(0x02); break;
        case 0x02 : return(0x03); break;
        case 0x04 : return(0x04); break;
        case 0x05 : return(0x05); break;
        case 0x06 : return(0x06); break;
        case 0x08 : return(0x07); break;
        case 0x09 : return(0x08); break;
        case 0x0A : return(0x09); break;
        case 0x0D : return(0x00); break;
        default : return(0x0F); break;
    }
}

void send_to_lcd(unsigned char value)
{
    unsigned char buf = 0; // For keep value convert
    buf = value & 0xF0; // Filter for
high byte
    buf = (buf>>4) | (0x30); // Shift to 4
times(sw) OR 0x30 for convert
// to ascii code
    lcd_text(buf); // Send
to show LCD
    buf = value & 0x0F; // Filter for low
byte
    buf = buf | 0x30; // OR
0x30 for convert to ascii code
    lcd_text(buf); // Send to show
LCD
}

/***** Function read data from DS1307
*****/
unsigned char DS1307_rd(unsigned char addr)
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

while (convert(P0) == 0x0F);
edit_date[i] = convert(P0);
lcd_text(convert(P0) + '0');

                                while(!sw);
                                while(sw);
                                }

L1 = ((edit_date[0] << 4) | edit_date[1]);
L2 = ((edit_date[2] << 4) | edit_date[3]);
L3 = ((edit_date[4] << 4) | edit_date[5]);

                                if((L1 >= 0x32) || (L2 >= 0x13))
                                {
error");
                                lcd_puts(0x90, "Time
ENTER");
                                lcd_puts(0xD0, "Prass
                                while(P0 != 0x0B);
                                while(sw);
                                lcd_puts(0xD0, "
                                goto loop0;
                                }
");
                                else
                                {
                                DS1307_wrrdate(L1, L2, L3);
                                while(P0 != 0x0F);
                                while(sw);
                                }
}

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

error");
ENTER");
");
lcd_puts(0x90, "Time
lcd_puts(0xD0, "Pass
while(P0!=0x0F);
while(sw);
lcd_puts(0xD0, "
goto loop0;
}
DS1307_wrtime(L1,L2,L3);
}

```

```
void set_date(void)
```

```

{
int i,L1,L2,L3;
char edit_time[5]=0;
char edit_date[5]=0;
lcd_clear();
lcd_puts(0x80, "Set date");
lcd_puts(0xC0, "dd:mm:yy");
lcd_command(0x0F);
loop0: lcd_command(0xC0);
while(!sw);
while(sw);
while(!sw);
while(sw);
for(i=0;i<=1;i++)
{
while(convert(P0)==0x0F);
edit_date[i]=convert(P0);
lcd_text(convert(P0)+'0');
while(!sw);
while(sw);
}
lcd_command(0xC3);
for(i=2;i<=3;i++)
{
while(convert(P0)==0x0F);
edit_date[i]=convert(P0);
lcd_text(convert(P0)+'0');
while(!sw);
while(sw);
}
lcd_command(0xC6);
for(i=4;i<=5;i++)
{

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

{

int i,L1,L2,L3;
char edit_time[5]=0;
char edit_date[5]=0;
lcd_clear();
lcd_puts(0x80,"Set Time");
lcd_puts(0xC0,"hh:mm:ss");
lcd_command(0x0F);
lcd_command(0xC0);
loop0:
while(!sw);
while(sw);

for(i=0;i<=1;i++)
{
while(convert(P0)==0x0F);
edit_time[i]=convert(P0);
lcd_text(convert(P0)+'0');

while(!sw);
while(sw);
}
lcd_command(0xC3);
for(i=2;i<=3;i++)
{
while(convert(P0)==0x0F);
edit_time[i]=convert(P0);
lcd_text(convert(P0)+'0');

while(!sw);
while(sw);
}
lcd_command(0xC6);
for(i=4;i<=5;i++)
{
while(convert(P0)==0x0F);
edit_time[i]=convert(P0);
lcd_text(convert(P0)+'0');

while(!sw);
while(sw);
}

L1=((edit_time[0]<<4)|edit_time[1]);
L2=((edit_time[2]<<4)|edit_time[3]);
L3=((edit_time[4]<<4)|edit_time[5]);
if((L1>=0x24)|| (L2>=0x60)|| (L3>=0x60))

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

        lcd_command(0x0C);
bring                                     // None cursor and
        lcd_puts(0x80,"Time:");           // Display screen for Date
        lcd_puts(0xC0,"Date:");          // Display screen for Time
    }
/***** Display Date and Time
*****/
//_____ Date dd/mm/yy_____//
//_____ Time hh/mm/ss_____//
/*****
*****/
void display_datetime(void)
{
    array[0] = DS1307_rd(0x00);           // Read sec before
show on LCD
    array[1] = DS1307_rd(0x01);           // Read min before
show on LCD
    array[2] = DS1307_rd(0x02);           // Read hour before
show on LCD
    array[3] = DS1307_rd(0x04);           // Read date before
show on LCD
    array[4] = DS1307_rd(0x05);           // Read month before show
on LCD
    array[5] = DS1307_rd(0x06);           // Read year before
show on LCD

    scr_datetime();
    /*-----Show Time-----*/
    lcd_origin();                         // Go to origin address
    lcd_command(0x86);                     // Set LCD address 46H
    send_to_lcd(array[2]);                 // Write hour show on LCD
    lcd_text(':');                          // Write ":" show on LCD
    send_to_lcd(array[1]);                 // Write min show on
LCD
    lcd_text(':');                          // Write ":" show on
LCD
    send_to_lcd(array[0]);                 // Write sec show on
LCD
    /*-----Show Date-----*/
    lcd_origin();                         // Go to origin
address
    lcd_command(0xC6);                     // Set LCD address
06H
    send_to_lcd(array[3]);                 // Write date show on
LCD
    lcd_text('/');                         // Write "/" show on LCD
    send_to_lcd(array[4]);                 // Write month show
on LCD
    lcd_text('/');                         // Write "/"
show on LCD
    send_to_lcd(array[5]);                 // Write year show on
LCD
}
void set_time(void)

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

    i2c_write(addr);
on DS1307 for connect // Write RAM address

    i2c_start(); // i2c start
condition
    i2c_write(DS1307_ID+1); // Write DS1307 ID for Read
Mode connect
    ret = i2c_read(); // Read
data and keep to ret
    i2c_stop(); // i2c stop
condition
    return(ret); // return value
}

```

```

/***** Function write hour/min/sec on chip DS1307
*****/

```

```

void DS1307_wrtime(unsigned char hh,unsigned char mm,unsigned char
ss)
{
    i2c_start(); // i2c start condition
    i2c_write(DS1307_ID); // Write DS1307 ID for
connect
    i2c_write(0x00); // Write control byte
to access RAM address 00H
    i2c_write(ss); // Write sec on
RAM address 00H
    i2c_write(mm); // Write min on
RAM address 01H
    i2c_write(hh); // Write hour
on RAM address 02H
    i2c_stop(); //
i2c stop condition
}

```

```

/***** Function write date/month/year on chip DS1307
*****/

```

```

void DS1307_wrdate(unsigned char dd,unsigned char mm,unsigned char
yy)
{
    i2c_start(); // i2c
start condition
    i2c_write(DS1307_ID); // Write DS1307 ID for
connect
    i2c_write(0x04); // Write control byte
to access RAM address 04H
    i2c_write(dd); // Write date
on RAM address 04H
    i2c_write(mm); // Write month
on RAM address 05H
    i2c_write(yy); // Write year
on RAM address 06H
    i2c_stop(); // i2c stop
condition
}

```

```

/***** Function display screen on
LCD*****/

```

```

void scr_datetime(void)
{
    // Clear LCD display

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่นับญาติให้มาไปไซ้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมอ่านค่าจากอุปกรณ์เก็บข้อมูลโดยใช้ Delphi

```
unit Unit1;

interface

uses
  Windows, Messages, SysUtils, Variants, Classes, Graphics, Controls,
  Forms,
  Dialogs, StdCtrls, CPort;

type
  TForm1 = class(TForm)
    connect: TButton;
    vol: TEdit;
    Label2: TLabel;
    Label3: TLabel;
    Label4: TLabel;
    min: TEdit;
    sec: TEdit;
    Label5: TLabel;
    day: TEdit;
    month: TEdit;
    year: TEdit;
    ComPort1: TComPort;
    hour: TEdit;
    Label1: TLabel;
    Button2: TButton;
    read: TButton;
    id: TEdit;
    Button1: TButton;
    Label6: TLabel;
    procedure connectClick(Sender: TObject);
    procedure Button2Click(Sender: TObject);
    procedure ComPort1RxChar(Sender: TObject; Count: Integer);
    procedure readClick(Sender: TObject);
    procedure Button1Click(Sender: TObject);

  private
    { Private declarations }
  public
    { Public declarations }
  end;

var
  Form1: TForm1;
  tr: integer;
implementation

{$R *.dfm}

procedure TForm1.connectClick(Sender: TObject);
var buff:string;

    addr,addr2,addr3 :integer;
    dat :integer;
    buffer :int64;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

count : integer;
str,blackup : string;

begin
    read.Enabled :=True;
    comport1.Connected :=true;
    dat:=65;
    comport1.Write(dat,1);
    inputQuery('Address','Enter address',buff);

```

```

    if(buff<>'') then
    begin
        addr:=strtoint(buff);
        addr2:=addr div 10;
        addr3:= addr mod 10;
        comport1.Write(addr2,1);
        comport1.Write(addr3,1);
        comport1.Readstr(str,19);

```

```

        if str='E'then
        begin
            label6.Caption :='Connecting';
            id.Text:=inttostr(addr);
        end

```

```

    else
    begin
        label6.Caption :='No Hardware';
        read.Enabled :=false;
    end;

```

```

    vol.Text :='';
    sec.Text :='';
    min.Text :='';
    hour.Text :='';
    day.Text :='';
    mount.Text :='';
    year.Text :='';
end;

```

```
end;
```

```
procedure TForm1.Button2Click(Sender: TObject);
```

```

    var dat :integer;
begin
    id.Text:='';
    vol.Text :='';
    sec.Text :='';
    min.Text :='';
    hour.Text :='';
    day.Text :='';
    mount.Text :='';
    year.Text :='';
end;

```

```
procedure TForm1.ComPort1RxChar(Sender: TObject; Count: Integer);
```

```

    var str:string;
begin
    TR:=0;
end;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```

procedure TForm1.readClick(Sender: TObject);

var buff:string;

    addr,addr2,addr3 :integer;
    dat :integer;

    count : integer;
    str,blackup,blackup2 : string;

begin

    comport1.Readstr(str,19);
    { vol.Text :=str; }
    blackup:=str;
    delete(blackup,8,12);
    blackup2:=blackup;
    delete(blackup,5,3);
    delete(blackup2,1,4);

    vol.Text :=blackup+'.'+ blackup2;
    blackup:=str;
    delete(blackup,1,7);
    delete(blackup,3,10);
    sec.Text:=blackup;
    blackup:=str;
    delete(blackup,1,9);
    delete(blackup,12,8);
    min.Text:=blackup;
    blackup:=str;
    delete(blackup,1,11);
    delete(blackup,14,6);
    hour.Text:=blackup;
    blackup:=str;
    delete(blackup,1,13);
    delete(blackup,16,4);
    day.Text:=blackup;
    blackup:=str;
    delete(blackup,1,15);
    delete(blackup,18,2);
    mount.Text:=blackup;
    blackup:=str;
    delete(blackup,1,17);

    year.Text:=blackup;
    dat:=48;
    comport1.Write(dat,1);
end;

procedure TForm1.Button1Click(Sender: TObject);
begin
    comport1.ShowSetupDialog ;
end;

end.

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้คัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้