สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ระบบมอนิเตอร์อุณหภูมิแบบไร้สาย

Wireless Temperature Monitoring System

โดย
นายภูมินทร์ มูลสถาน
นางสาวมยุรี นามสิน

ปพ.
ว668ร
2549

เลขหมู่.....
เลขทะเบียน 72114
วัน,เดือน,ปี - 8 มิ.ย. 2550

.b 11763553
.i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต
สาขาวิชาอิเล็กทรอนิกส์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบมอนิเตอร์อุณหภูมิแบบไร้สาย

Wireless Temperature Monitoring System

โดย

นายภูมินทร์ มูลสถาน รหัส 46010589

นางสาวมยุรี นามสิน รหัส 46010600

อาจารย์ที่ปรึกษา

รศ.ดร. ชูชาติ ปิณฑวิรุจน์

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตร์บัณฑิต

สาขาวิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2549

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ปริญญานิพนธ์ปีการศึกษา 2549

ภาควิชาอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าลาดกระบัง

เรื่อง ระบบมอนิเตอร์อุณหภูมิแบบไร้สาย (Wireless Temperature Monitoring System)

ผู้จัดทำ

1. นายภูมินทร์ มูลสถาน รหัส 46010589
2. นางสาวมยุรี นามสิน รหัส 46010600

.....อาจารย์ที่ปรึกษา

(รศ.ดร. ชูชาติ ปิณฑวิรุจน์)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ระบบมอนิเตอร์อุณหภูมิแบบไร้สาย

นายภูมินทร์ มูลสถาน รหัส 46010589
นางสาวมยุรี นามสิน รหัส 46010600
รศ.ดร. ชูชาติ ปิณฑวิรุจน์ อาจารย์ที่ปรึกษา
ปีการศึกษา 2549

บทคัดย่อ

โครงงานนี้เกี่ยวกับการตรวจสอบอุณหภูมิภายในห้องซึ่งสามารถตรวจสอบได้โดยไม่ต้องอยู่ที่บริเวณห้อง โดยจะใช้ไอซี DS18S20 เป็นเซนเซอร์อ่านอุณหภูมิและทำการแสดงผลทาง 7-Segment แล้วทำการส่งผ่านข้อมูลแบบไร้สายโดยใช้ RF โมดูล ทำการเชื่อมต่อกันเป็นโครงข่ายและส่งค่ามายังตัวรับซึ่งทำการเชื่อมต่อเข้ากับเครื่องคอมพิวเตอร์แล้วทำการประมวลผลโดยใช้ Visual Basic เพื่อแสดงค่าอุณหภูมิที่อ่านค่าได้ นำมาแสดงบนหน้าจอและแสดงผลออกมาเป็นรูปกราฟ และเก็บค่าที่รับได้ไว้ในระบบฐานข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Wireless Temperature Monitoring System

Mr. Poomin Mulsatan ID 46010589

Miss Mayuri Namsin ID 46010600

Assoc. Prof. Dr. Chuchart Pintavirooj Advisor

Educational Year 2006

Abstract

This project is about how to "Monitor Temperature". It can check temperature in a room by using IC "DS1820". It shows temperature by 7-segment then create network between transmitter and receiver and then send wireless signal (radio frequency) to transfer data to receiver by RF module. The receiver is connect with computer by serial port. Computer will show temperature as graph in time and temperature by using the written in programm Visual Basic.NET and then save data to database (Microsoft Access)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ

โครงงานระบบมอนิเตอร์อุณหภูมิแบบไร้สายนี้สามารถเสร็จสิ้นลงได้ ผู้จัดทำรายงานขอขอบพระคุณอาจารย์ที่ปรึกษาที่ได้ให้คำปรึกษา และคำแนะนำในการทำโครงงานชิ้นนี้ ขอขอบคุณพ่อแม่ผู้ปกครองที่ให้การสนับสนุนทั้งด้านการเงินและคอยเป็นกำลังใจในการทำงานเสมอมา และขอบคุณเพื่อน ๆ ที่คอยให้ความช่วยเหลือเป็นอย่างดี แม้บางคราวจะรบกวนเพื่อนไปหน่อย

นายภูมินทร์ มูลสถาน รหัส 46010589
นางสาวมยุรี นามสิน รหัส 46010600
ผู้จัดทำ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ (ต่อ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป

หน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป (ต่อ)

หน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญตาราง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1
บทนำ

ระบบไร้สายเป็นระบบควบคุมที่ไม่ต้องมีอุปกรณ์ใด ๆ เป็นตัวนำสัญญาณ โดยสัญญาณจะเดินทางผ่านไปในอากาศ ชนิดของสัญญาณที่เดินทางผ่านไปในอากาศได้ อาจอยู่ในรูปแบบของสัญญาณเสียง สัญญาณแสง และคลื่นวิทยุ สำหรับคลื่นวิทยุนั้นสามารถทะลุผ่านสิ่งกีดขวางได้ สัญญาณของระบบวิทยุจะถูกนำมามอดดูเลตกับคลื่นวิทยุที่ภาคส่งก่อนซึ่งเป็นหลักการทั่วไปของการสื่อสาร

ในโครงงานนี้เป็นระบบมอนิเตอร์อุณหภูมิแบบไร้สาย ซึ่งต้องการอ่านค่าอุณหภูมิในจุดต่าง ๆ โดยที่เราสามารถอ่านค่าได้โดยไม่ต้องเดินทางไปยังจุดนั้น ทั้งยังไม่ต้องยุ่งยากในการเดินสาย เนื่องจากเป็นแบบไร้สายโดยอาศัยคลื่นวิทยุในการส่งข้อมูลอุณหภูมิ และได้นำอุปกรณ์ทางอิเล็กทรอนิกส์มาช่วยในการอ่านอุณหภูมิและช่วยในการส่งข้อมูล โดยนำไมโครคอนโทรลเลอร์ตระกูล MCS-51 เบอร์ AT89S52 เข้ามาเป็นตัวควบคุม ส่วนตัววัดอุณหภูมิใช้ไอซีเบอร์ DS1820 มาเป็นตัวตรวจจับอุณหภูมิ ค่าที่ได้ออกมาเป็นสัญญาณดิจิตอลส่งไปยังไมโครคอนโทรลเลอร์พร้อมกับแสดงผลออกมา ส่วนสัญญาณไร้สายได้นำ Module RF TRW มาใช้ในการส่งสัญญาณ ในส่วนของการแสดงผลจะใช้โปรแกรม Visual Basic.Net โดยจะแสดงผลเป็นกราฟระหว่างอุณหภูมิกับเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2
ทฤษฎีและหลักการ

2.1 ทฤษฎีไมโครคอนโทรลเลอร์ MCS-51

2.1.1 คุณลักษณะพื้นฐานของ MCS-51

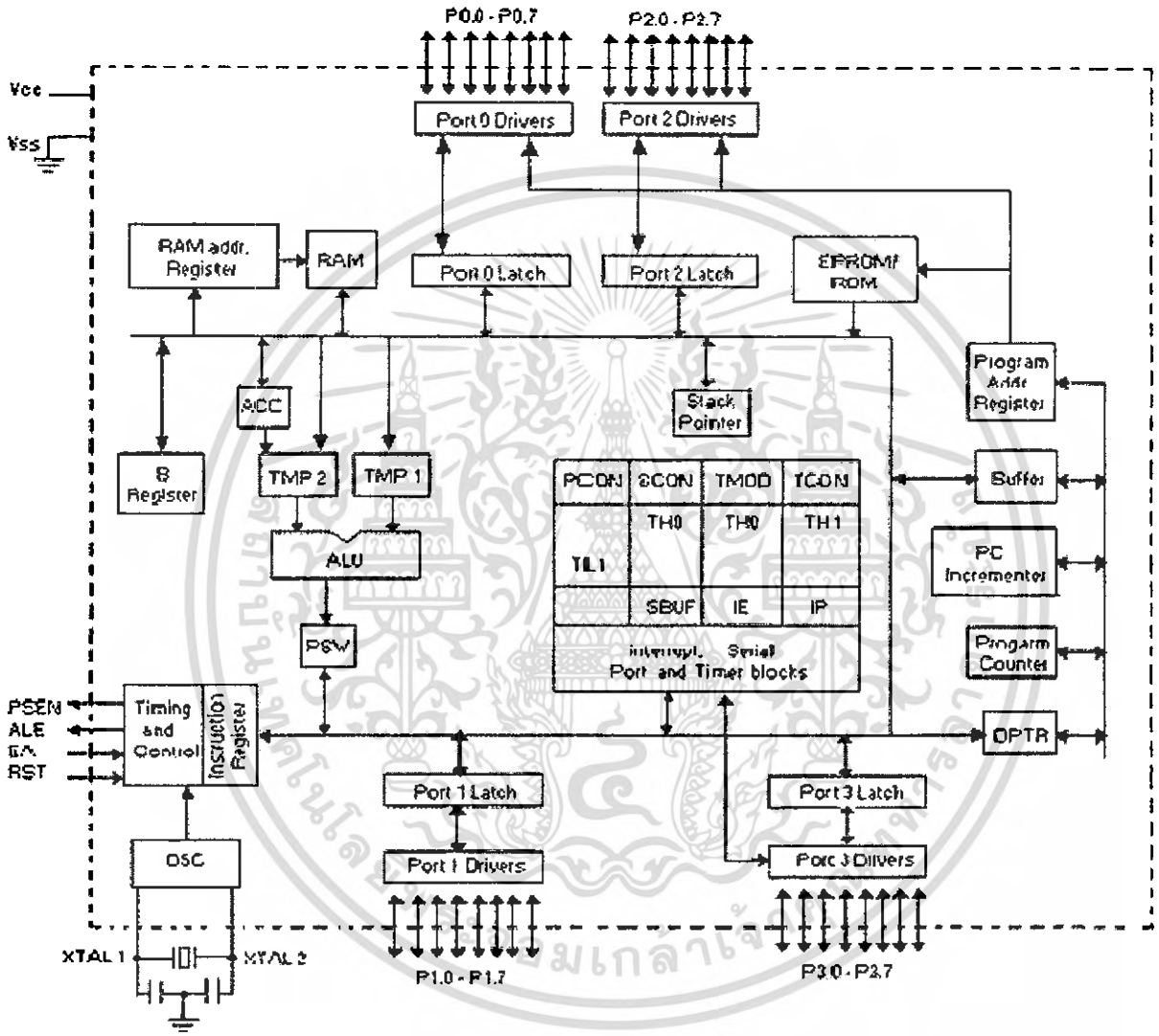
คุณสมบัติของไมโครคอนโทรลเลอร์ ตระกูล MCS-51 อนุกรม AT89xx

1. เป็นไมโครคอนโทรลเลอร์ที่ใช้ซีพียูขนาด 8 บิต
2. ภายในมีหน่วยความจำโปรแกรมเป็นแบบแฟลช สามารถลบและเขียนใหม่ได้พันครั้ง
3. หน่วยความจำข้อมูลพื้นฐานเป็นหน่วยความจำแบบแรม ในบางเบอร์จะมีหน่วยความจำแบบอีอีพรอมเพิ่มเติม
4. ขาพอร์ตเป็นแบบสองทิศทาง สามารถใช้งานเป็นได้ทั้งอินพุทและเอาท์พุท
5. มีวงจรสื่อสารอนุกรมแบบฟูลดูเพล็กซ์
6. ไทเมอร์/เคาน์เตอร์ขนาด 16 บิต อย่างน้อย 2 ตัว
7. สามารถรองรับแหล่งกำเนิดอินเตอร์รัปต์ได้ 6 ประเภท
8. สามารถขยายหน่วยความจำภายนอกเพิ่มเติมได้สูงสุด 64 กิโลไบต์
9. มีวงจรกำเนิดสัญญาณนาฬิกาอยู่ภายในชิป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

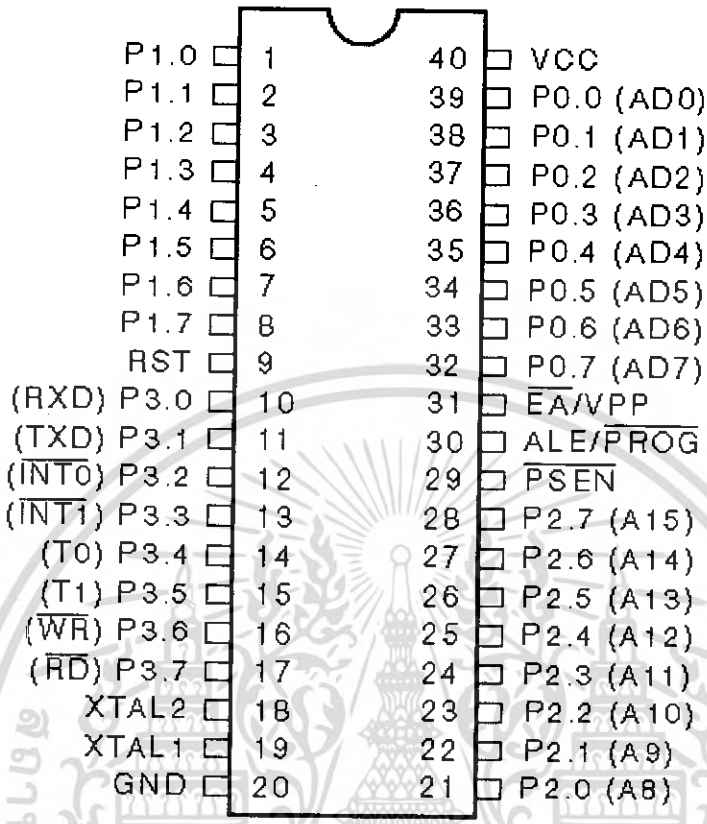
2.1.2 ลักษณะการจัดขาของ MCS-51

ไมโครคอนโทรลเลอร์ MCS-51 ทุกเบอร์จะมีสถาปัตยกรรมและขาใช้งานพื้นฐานเหมือนกัน ดังแสดงในรูปที่ 2.1 และ 2.2 โดยมีรายละเอียดขั้นต้นดังนี้



รูปที่ 2.1 แสดงรายละเอียดโครงสร้างหลักของไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลช ของ ATMEL

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่2.2 การจัดขามาตรฐานของไมโครคอนโทรลเลอร์ MCS-51 ในอนุกรม AT89C5x

ขา Vcc ใช้สำหรับต่อไฟเลี้ยง +5V

ขา GND เป็นขากราวด์ สำหรับต่อกับกราวด์ของระบบ

ขาพอร์ต 0 (P0.0-P0.7) มีขา 8 ขา สามารถกำหนดให้เป็นได้ทั้งอินพุทและเอาท์พุท สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ต 0 ขาใดขาหนึ่งเป็นอินพุท สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุทอิมพิแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุทได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์ต่ำของหน่วยความจำภายนอก (A0-A7) และขาข้อมูล (D0-D7) โดยใช้กระบวนการมัลติเพล็กซ์เข้าช่วย เพื่อสลับการทำงานเป็นได้ทั้งขาติดต่อแอดเดรส และขาข้อมูล

ขาพอร์ต 1 (P1.0-P1.7) มีขา 8 ขา แต่ละขาสามารถกำหนดให้เป็นทั้งอินพุทและเอาท์พุท สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุท สามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขาพอร์ต 2 (P2.0-P2.7) มีขา 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุทและเอาท์พุท สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุทสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุทอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุทได้ นอกจากนั้นขาพอร์ตนี้ยังถูกใช้งานในการติดต่อกับขาแอดเดรสไบต์สูงของหน่วยความจำภายนอก (A8-A15)

ขาพอร์ต 3 (P3.0-P3.7) มี 8 ขา แต่ละขาสามารถกำหนดให้เป็นได้ทั้งอินพุทและเอาท์พุท สำหรับใช้งานทั่วไป ถ้าหากต้องการกำหนดให้ขาพอร์ตใดเป็นอินพุทสามารถทำได้โดยการเขียนข้อมูล "1" ไปยังแต่ละบิตของพอร์ตที่ต้องการติดต่อด้วย ส่งผลให้ขาพอร์ตนั้นมีสถานะปล่อยลอย (float) จึงมีอินพุทอิมพีแดนซ์สูง สามารถใช้งานเป็นขาพอร์ตอินพุทได้ นอกจากนั้นขาพอร์ต 3 ยังเป็นขาที่มีหน้าที่การใช้งานพิเศษ ดังมีรายละเอียดขั้นตอนต่อไปนี้

P3.0 ใช้เป็นขาอินพุทสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา RxD

P3.1 ใช้เป็นขาอินพุทสำหรับรับข้อมูลจากการสื่อสารแบบอนุกรม หรือขา TxD

P3.2 ใช้เป็นขาอินพุทรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 0 หรือขา $\overline{\text{INT0}}$

P3.3 ใช้เป็นขาอินพุทรับสัญญาณอินเตอร์รัปต์จากภายนอกช่อง 1 หรือขา $\overline{\text{INT1}}$

P3.4 ใช้เป็นขาอินพุทสำหรับรับสัญญาณไทเมอร์จากภายนอกช่อง 0 หรือขา T0

P3.5 ใช้เป็นขาอินพุทสำหรับรับสัญญาณไทเมอร์จากภายนอกช่อง 1 หรือขา T1

P3.6 ใช้เป็นขาสัญญาณ $\overline{\text{WR}}$ ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

P3.7 ใช้เป็นขาสัญญาณ $\overline{\text{RD}}$ ในกรณีที่ใช้เชื่อมต่อกับหน่วยความจำภายนอก

ขารีเซต (Reset) ใช้ในการรีเซตการทำงานของไมโครคอนโทรลเลอร์ โดยใช้การป้อนสัญญาณเพื่อรีเซตสถานะที่ขานี้ต้องอยู่ในระดับรีเซตอย่างน้อย 2 แมชชีนไซเคิล โดยที่วงจรกำเนิดสัญญาณนาฬิกายังคงทำงานต่อเนื่องไปอย่างเป็นปกติ

ขา $\overline{\text{ALE}}$ /PROG (Address Latch Enable/Program pulse input) เป็นขาที่ใช้ในการควบคุมการแลตช์ของขาพอร์ต 0 เมื่อมีการใช้งานหน่วยความจำภายนอก นอกจากนั้นขานี้ยังใช้เป็นขาสำหรับรับพัลส์ของการโปรแกรมสำหรับโปรแกรมข้อมูลลงในไมโครคอนโทรลเลอร์ MCS-51 ในรุ่นที่มีหน่วยความจำโปรแกรมเป็นแบบอีพรอม

ขา $\overline{\text{PSEN}}$ (Program Store Enable) ขานี้ใช้ในการส่งสัญญาณเพื่อร้องขอติดต่อกับหน่วยความจำโปรแกรมภายนอก เมื่อไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอก ตัวไมโครคอนโทรลเลอร์ต้องการอ่านข้อมูลจากหน่วยความจำโปรแกรมภายนอกตัวไมโครคอนโทรลเลอร์จะส่งสัญญาณออกมาที่ขานี้ 2 ครั้ง ในแต่ละแมชชีนไซเคิล แต่ถ้าหากติดต่อกับหน่วยความจำข้อมูลภายนอกขานี้จะไม่มีการส่งสัญญาณใดๆออกมา

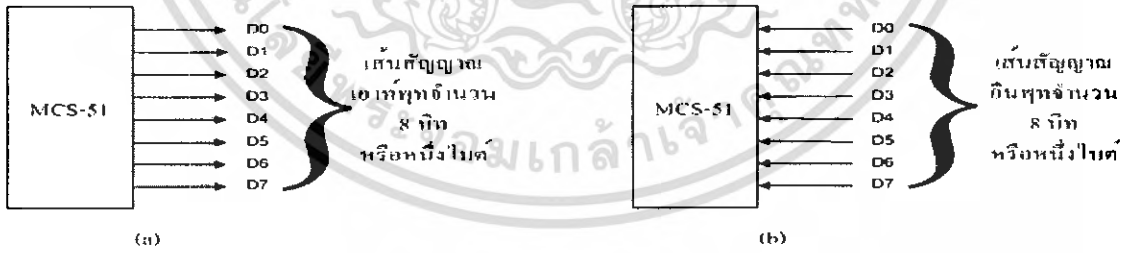
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ขา $\overline{\text{EA}}$ /Vpp (External Access enable/Programming voltage input) ใช้สำหรับเลือกการติดต่อหน่วยความจำโปรแกรมจากภายนอกหรือภายในตัวไมโครคอนโทรลเลอร์ ถ้าหาขานี้เป็น "0" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำโปรแกรมภายนอก แต่ถ้าหากขานี้เป็น "1" เป็นการเลือกให้ไมโครคอนโทรลเลอร์ติดต่อกับหน่วยความจำภายในตัวไมโครคอนโทรลเลอร์ นอกจากนี้ที่ขานี้ยังใช้เป็นขาอินพุทสำหรับรับแรงดันไฟสูงสำหรับการโปรแกรมหน่วยความจำภายในไมโครคอนโทรลเลอร์ สำหรับไมโครคอนโทรลเลอร์ MCS-51 แบบแฟลชต้องการแรงดันสำหรับการโปรแกรม +5V

ขา XTAL1 และ XTAL2 เป็นขาสำหรับต่อคริสตัลเพื่อสร้างสัญญาณนาฬิกาในการกำหนดจังหวะการทำงานของไมโครคอนโทรลเลอร์

2.1.3 พอร์ตอินพุทและพอร์ตเอาท์พุท

พอร์ต คือ แอดเดรสหนึ่งที่ได้รับการกำหนดไว้เพื่อการโอนย้ายข้อมูลระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์ภายนอก การกำหนดประเภทของการติดต่อขึ้นอยู่กับทิศทางการไหลของข้อมูล เมื่อพิจารณาจากไมโครคอนโทรลเลอร์เป็นหลัก จากรูปที่ 2.3(a) เป็นการใช้ไมโครคอนโทรลเลอร์เป็นเอาท์พุทพอร์ต และจากรูป 2.3(b) เป็นการใช้ไมโครคอนโทรลเลอร์เป็นอินพุทพอร์ต



รูปที่ 2.3 การใช้ไมโครคอนโทรลเลอร์เป็นอินพุทและเอาท์พุทพอร์ต

2.1.3.1 การใช้งานพอร์ตเป็นอินพุท

การใช้งานพอร์ตเป็นการอินพุทข้อมูลจะต้องเริ่มต้นด้วยการส่งข้อมูลที่มีค่าเป็น 1 ออกมาทางบิตของพอร์ตสั้นก่อนเป็นอันดับแรก เพื่อหยุดการทำงานของทรานซิสเตอร์ที่ทำหน้าที่ขับสัญญาณ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอาท์พุทของบิตนั้น ทำให้ขาสัญญาณของบิตถูกต่อเข้ากับตัวต้านทานซึ่งทำหน้าที่ Pull-up ภายใน ซึ่งมีผลทำให้บิตนั้นของพอร์ต 1 , 2 และ 3 เป็นสภาวะลอจิกสูง ตัวต้านทานนี้มี

ค่าประมาณ 50 kohm ซึ่งเป็นค่าที่สูงมาก และทำให้อุปกรณ์ภายนอกสามารถขับสัญญาณของพอร์ตเหล่านี้เป็นลอจิกต่ำได้ง่าย สำหรับบิตของพอร์ต 0 นั้นแม้ว่าจะมีหลักการทำงานที่คล้ายคลึงกันกับบิตของพอร์ตอื่นๆ แต่เนื่องจากไม่มีตัวต้านทานซึ่งทำหน้าที่ Pull-up ภายในไว้ ทำให้เมื่อทรานซิสเตอร์ที่ทำหน้าที่ขับสัญญาณเอาท์พุทนั้นหยุดการทำงาน ก็จะเป็นผลให้สัญญาณนี้อยู่ในสภาวะอิมพีแดนซ์สูงแทน

2.1.3.2 การใช้งานพอร์ตเป็นเอาท์พุท

เมื่อมีการส่งข้อมูลที่มีค่าเป็น 0 ให้กับแต่ละบิตของพอร์ตทุกพอร์ต ข้อมูลนี้จะถูกส่งให้กับฟลิปฟลอป ซึ่งจะค้างค่านี้ไว้ และมีผลทำให้ทรานซิสเตอร์ที่ทำหน้าที่ขับสัญญาณเอาท์พุทนั้นทำงาน ดังนั้นขาสัญญาณก็จะมีสภาวะลอจิกเป็นลอจิกต่ำด้วย

ส่วนการส่งข้อมูลที่มีค่าเป็น 1 ออกมานั้น ในกรณีที่เป็นการทำงานในแต่ละบิตของพอร์ต 1 , 2 หรือ 3 จะทำให้ทรานซิสเตอร์ที่ทำหน้าที่ขับสัญญาณเอาท์พุทนั้นหยุดทำงาน มีผลทำให้ขาของสัญญาณเป็นลอจิกสูงด้วยตัวต้านทานที่ Pull-up อยู่ภายในนั้น แต่สำหรับการใช้งานในแต่ละบิตทางพอร์ต 0 นั้นจะมีผลแตกต่างออกไป โดยขาสัญญาณจะมีสภาวะอิมพีแดนซ์สูงแทน เนื่องจากไม่มีตัวต้านทานภายในเชื่อมต่ออยู่นั้นเอง ดังนั้นการใช้งานพอร์ต 0 เป็นการนำข้อมูลออกทางเอาท์พุทจึงจำเป็นต้องใช้ตัวต้านทานภายนอก Pull-up สัญญาณไว้กับลอจิกสูงแทน

2.1.4 หน่วยความจำโปรแกรมของ MCS-51

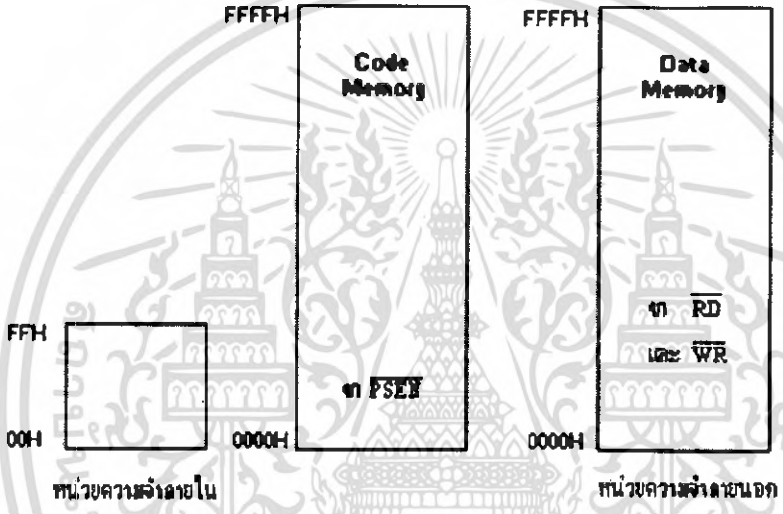
หน่วยความจำโปรแกรม(Program Memory) มีไว้เพื่อบรรจุคำสั่งหรือโปรแกรมที่ผู้ใช้พัฒนาขึ้น จัดเก็บไว้ในหน่วยความจำ โดยอาจจะประกอบอยู่ภายในตัวของไอซี 89C51 เอง หรือเป็นไอซีหน่วยความจำ EPROM หรือ ROM แยกออกต่างหากได้ ในกรณีหลังจำเป็นต้องมีการใช้พอร์ตอินพุทเอาท์พุท ทำหน้าที่เป็นบัสแอดเดรส และบัสข้อมูลเพื่อให้สามารถติดต่อกับหน่วยความจำมาตรฐานทั่วไปได้

หน่วยความจำโปรแกรมของ 89C51 เป็นบริเวณหน่วยความจำสำหรับเก็บข้อมูลและคำสั่งใช้งานต่างๆ ซึ่งแม้ว่าจะไม่มีการจ่ายกระแสไฟฟ้าให้ระบบข้อมูลเหล่านี้ก็ยังไม่สูญหาย โครงสร้างของหน่วยความจำโปรแกรมมีลักษณะเช่นเดียวกับหน่วยความจำที่บรรจุอยู่ในไอซีหน่วยความจำประเภทต่างๆเช่น หน่วยความจำแบบ ROM (Read Only Memory) หรือ EPROM (Erasable Programmable Read Only Memory)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การจัดพื้นที่ของหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ 89C51 จะมีการจัดพื้นที่ดังรูปที่ 2.4 โดยที่ไมโครคอนโทรลเลอร์ 89C51 สามารถอ่านข้อมูลหน่วยความจำโปรแกรมสูงสุดได้ไม่เกิน 64 กิโลไบต์ และแยกประเภทของหน่วยความจำโปรแกรมเป็น 2 ลักษณะตาม

ตำแหน่งของหน่วยความจำนั้น คือ หน่วยความจำโปรแกรมภายใน (Internal Program Memory) ซึ่งเป็นหน่วยความจำ ROM หรือ EPROM ที่อยู่ภายในตัวไอซี ไมโครคอนโทรลเลอร์เอง และหน่วยความจำภายนอก (External Program Memory) ซึ่งเป็นการใช้ไอซีหน่วยความจำมาทำหน้าที่เป็นหน่วยความจำโปรแกรมของระบบ

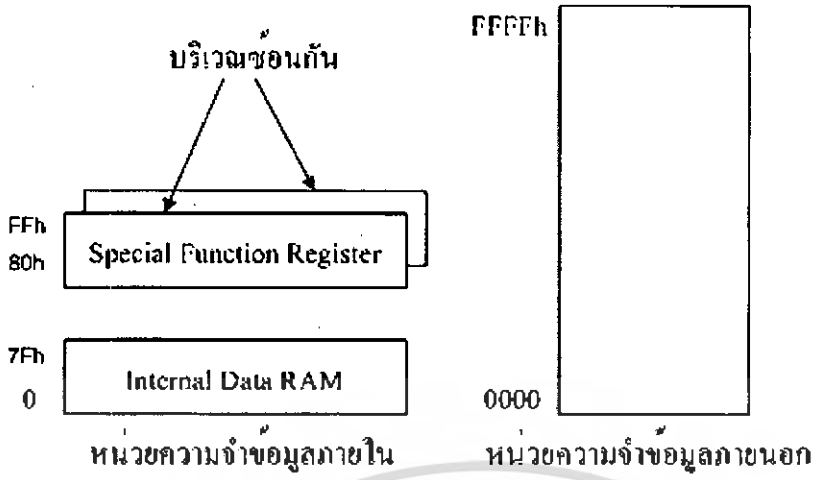


รูปที่ 2.4 แสดงการจัดพื้นที่ของหน่วยความจำโปรแกรมของไมโครคอนโทรลเลอร์ MCS-51

2.1.5 หน่วยความจำข้อมูลของ MCS-51

หน่วยความจำข้อมูลมีหน้าที่สำหรับเก็บข้อมูลหรือตัวแปรที่เกิดขึ้นในขณะที่กำลังประมวลผลโปรแกรมไว้เป็นการชั่วคราว โดยที่หน่วยความจำข้อมูลจะมีลักษณะเป็นหน่วยความจำ RAM แบบสแตติก (Static) ดังนั้นเมื่อไม่มีการจ่ายไฟให้กับระบบ ก็จะมีผลทำให้ข้อมูลที่จัดเก็บไว้ในหน่วยความจำนี้สูญหายไป พื้นที่ของหน่วยความจำข้อมูลของ 89C51 สามารถมีได้ไม่เกิน 64 กิโลไบต์ และแยกประเภทออกเป็นสองลักษณะตำแหน่งที่ตั้งของหน่วยความจำนั้นดังลักษณะในรูปที่ 2.5 คือ หน่วยความจำโปรแกรมภายใน (Internal Datc Memory) หรือ RAM ที่อยู่ภายในตัวไมโครคอนโทรลเลอร์เอง และหน่วยความจำข้อมูลภายนอก (External Data Memory) ซึ่งเป็นการใช้ไอซีหน่วยความจำ RAM มาเพิ่มเติมเข้าไปในวงจร ลักษณะเดียวกับการนำไอซี EPROM มาใช้งานเป็นหน่วยความจำโปรแกรมนั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.5 แสดงลักษณะตำแหน่งที่ตั้งของหน่วยความจำข้อมูล

2.1.5.1 หน่วยความจำข้อมูลภายใน

หน่วยความจำข้อมูลภายใน 89C51 มีจำนวนทั้งหมด 256 ไบต์ โดยจำแนกออกได้เป็นสองลักษณะคือ พื้นที่เฉพาะสำหรับตัวประมวลผลกลางใช้งานเท่านั้นซึ่งเรียกว่ารีจิสเตอร์ และพื้นที่ใช้งานทั่วไปสำหรับโปรแกรมใช้งานที่ผู้ใช้สร้างขึ้นมา จากรูปที่ 2.6 แสดงถึงการจัดพื้นที่ของหน่วยความจำข้อมูลภายในของ 89C51 ซึ่งแบ่งเป็นสองส่วน คือ หน่วยความจำขนาด 128 ไบต์แรก และหน่วยความจำขนาด 128 ไบต์ถัดไป

2.1.5.2 หน่วยความจำขนาด 128 ไบต์แรก

บริเวณนี้จะมีตำแหน่งแอดเดรสอยู่ในช่วง 00H ซึ่งแบ่งได้เป็นอีกสามส่วนดังนี้

* บริเวณแอดเดรส 00H-1FH จำนวน 32 ไบต์ จำแนกออกเป็นกลุ่มหรือแบงก์ข้อมูลจำนวน 8 ไบต์ รวมทั้งหมดสี่กลุ่ม พื้นที่ข้อมูลในแต่ละกลุ่มจะถูกใช้งานในฐานะของรีจิสเตอร์ทั่วไป เรียกว่า รีจิสเตอร์ R0-R7

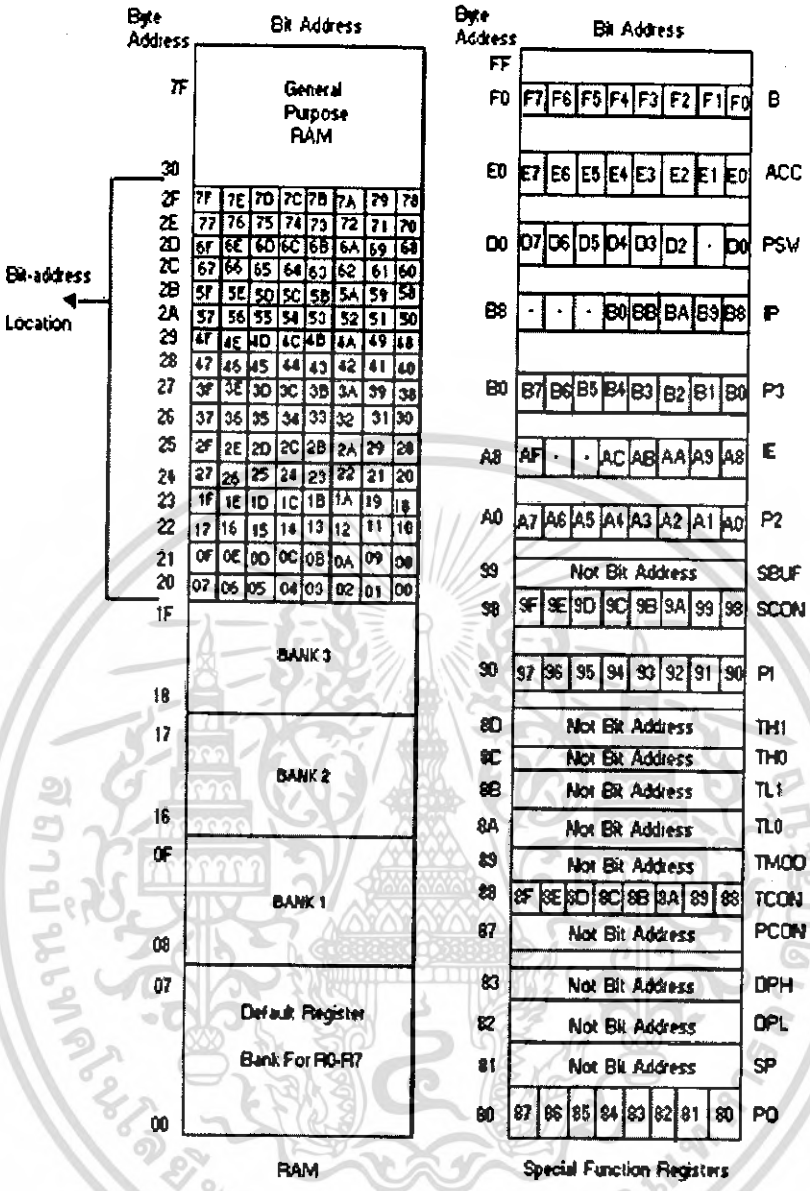
* บริเวณแอดเดรส 20H-2FH จำนวน 16 ไบต์ จะเป็นพื้นที่ส่วนสำหรับผู้ใช้ที่สามารถอ้างถึงข้อมูลได้ทั้งแบบไบต์และแบบบิต

* บริเวณแอดเดรส 30H-7FH เป็นบริเวณที่สามารถนำไปใช้งานได้อย่างอิสระ โดยสามารถอ้างอิงได้เฉพาะในลักษณะของไบต์ข้อมูลตามปกติเท่านั้น

2.1.5.3 หน่วยความจำขนาด 128 ไบต์ถัดไป

พื้นที่ตั้งแต่แอดเดรส 80H-FFH เป็นหน่วยความจำที่นำมาใช้งานเป็นรีจิสเตอร์หน้าที่พิเศษ แต่ก็ยังมีบริเวณของหน่วยความจำที่อยู่บริเวณนี้ที่ผู้ใช้สามารถเก็บข้อมูลได้ แต่การเรียกใช้งานจะต้องมีการเข้าถึงข้อมูลแบบโดยอ้อมเท่านั้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.6 แสดงตำแหน่งของหน่วยความจำแบบไบต์และแบบบิต

2.1.6 รีจิสเตอร์หน้าที่พิเศษ

เป็นรีจิสเตอร์ที่ทำหน้าที่ควบคุมการทำงานของอุปกรณ์หรือพอร์ตของ 89C51 ทั้งหมด โดยมีตำแหน่งอยู่ในบริเวณแอดเดรส 80H-FFH การใช้งานรีจิสเตอร์หน้าที่พิเศษเหล่านี้สามารถทำได้ทั้งการระบุถึงชื่อรีจิสเตอร์หรือตำแหน่งแอดเดรสที่เป็นของรีจิสเตอร์นั้นก็ได้

2.1.6.1 แอกคิวมูเลเตอร์ (Accumulator)

เป็นรีจิสเตอร์ขนาด 8 บิต ทำหน้าที่เก็บข้อมูลที่ส่งให้หน่วยทำงานในซีพียูและเก็บผลลัพธ์ที่ได้จากการทำงานนั้น การใช้งานในโปรแกรมจะเรียกว่า รีจิสเตอร์ A

2.1.6.2 รีจิสเตอร์ B

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เป็นรีจิสเตอร์ที่ใช้สำหรับการทำคำสั่งการคูณหารตัวเลข ในกรณีที่ไม่ใช้การคำนวณทางด้านคณิตศาสตร์ ก็สามารถนำไปใช้งานเช่นเดียวกับรีจิสเตอร์ทั่วไปได้

2.1.6.3 โปรแกรมเคาน์เตอร์(Program Counter)

เป็นรีจิสเตอร์ที่ใช้สำหรับการชี้ตำแหน่งแอดเดรสของหน่วยความจำโปรแกรม ซึ่งจะต้องไปทำงานในลำดับถัดไป การใช้งานในโปรแกรมจะเรียกว่า รีจิสเตอร์ PC

2.1.6.4 สแต็กพอยน์เตอร์(Stack Pointer)

เป็นรีจิสเตอร์ขนาด 8 บิต ทำหน้าที่เก็บตำแหน่งของตัวชี้หรือพอยน์เตอร์ของบริเวณสแต็กสำหรับเก็บข้อมูลแอกคิวมูเลเตอร์ รีจิสเตอร์ต่างๆ รวมทั้งข้อมูลจากโปรแกรม ค่าเริ่มต้นของสแต็กจะอยู่ที่ตำแหน่ง 07H การใช้งานในโปรแกรมจะเรียกว่ารีจิสเตอร์ SP

2.1.6.5 ตัวชี้ข้อมูลหรือดาต้าพอยน์เตอร์(Data Pointer)

เป็นรีจิสเตอร์ขนาด 16 บิต ซึ่งเรียกว่า รีจิสเตอร์ DPTR และสามารถใช้งานแยกออกเป็นรีจิสเตอร์ขนาด 8 บิต สองตัว คือ รีจิสเตอร์ DPH และ DPL เพื่อเก็บค่าแอดเดรสของหน่วยความจำที่จะต้องใช้งานภายในโปรแกรม หรืออาจเป็นแอดเดรสของอุปกรณ์ภายนอก

2.1.6.6 โปรแกรมสเตตัสเวิร์ด(PSW)

รีจิสเตอร์นี้ทำหน้าที่บอกถึงแฟลกสภาวะการทำงานต่างๆ รวมทั้งบิตสำหรับการกำหนดเลือกแบงก์(Bank) ของรีจิสเตอร์ที่ใช้งานด้วย

2.1.6.7 รีจิสเตอร์ที่เกี่ยวข้องกับพอร์ต(Port Register)

รีจิสเตอร์เหล่านี้จะมีความเกี่ยวข้องกับการทำงานของพอร์ตอินพุทเอาท์พุท โดยตรงซึ่งจะเป็นรีจิสเตอร์ขนาด 8 บิต สามารถใช้งานได้ทั้งในลักษณะการอินพุทหรือการเอาท์พุทข้อมูลได้

2.1.6.8 รีจิสเตอร์ SBUF

เป็นบัฟเฟอร์ขนาด 8 บิต สำหรับการสื่อสารข้อมูลแบบอนุกรมทั้งการรับและการส่งข้อมูล

2.1.6.9 รีจิสเตอร์ PCON

เป็นรีจิสเตอร์ควบคุมการทำงานในสามลักษณะ ซึ่งได้แก่ การควบคุมการทำงานของโปรแกรมเซ็นเซอร์ การกำหนดอัตราทวีคูณของอัตราเร็วในการสื่อสารข้อมูลอนุกรมและแฟลกสภาวะการทำงานทั่วไป

2.1.6.10 รีจิสเตอร์ IP,IE,TMOD,SCON

เป็นกลุ่มรีจิสเตอร์ที่ทำหน้าที่ควบคุมการทำงานของอินเตอร์รัปต์ต่างๆ

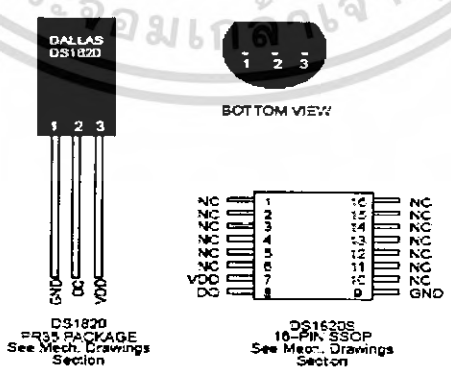
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2 DS 1820

2.2.1 ทฤษฎี DS 1820

อุปกรณ์ที่ทำหน้าที่เซ็นเซอร์ขนาดเล็กให้สัญญาณเอาท์พุทออกมาจากสายสัญญาณเส้นเดียวมีอยู่มากมายหลายเบอร์และหลายรูปแบบแต่ส่วนมากแล้วอุปกรณ์เซนเซอร์อุณหภูมิเหล่านั้นมักจะให้สัญญาณเอาต์พุตออกมาเป็นแบบอะนาลอก และอาจเป็นผลดีในแง่การใช้งานที่ไม่ต้องการควบคุมการทำงานมากนัก คือให้เอาท์พุตออกมาเพื่อการแสดงผลเป็นหลัก และให้วงจรอะนาลอกภายนอกต่อใช้งานร่วม แต่ถ้าอุปกรณ์ตังกล่าวให้สัญญาณเซนเซอร์อุณหภูมิออกมาเป็นแบบดิจิตอลแล้วการใช้งานจะสามารถกระทำได้มากกว่าการแสดงผล เพราะสัญญาณดิจิตอลที่ออกมาจะถูกประมวลผลได้ด้วยไมโครโปรเซสเซอร์ทำให้สามารถที่จะกำหนดค่าการทำงานและควบคุมจุดตรวจจับอุณหภูมิของแต่ละตัวเซนเซอร์ได้ รวมไปถึงการเซตตัวแสดงผลของอุณหภูมิในหลาย ๆ จุดได้พร้อม ๆ กันและอื่น ๆ อีกมากมายที่ไมโครโปรเซสเซอร์จะทำงานได้

DS 1820 สามารถทำงานได้มากกว่านั้น เพราะนอกจากจะให้สัญญาณเอาต์พุตออกมาเป็นแบบดิจิตอลแล้วยังสามารถที่จะทำการโปรแกรมเข้าไปยังส่วนหน่วยความจำและควบคุมฟังก์ชันภายในตัวไอซีได้อีกด้วย ซึ่งมีหน่วยความจำ ROM ขนาด 64 บิตแบบเลเซอร์รอม ดังนั้นจึงสามารถที่จะทำการอ่านและเขียนข้อมูลต่าง ๆ ที่เกี่ยวกับหน้าที่ในการทำงานเกี่ยวกับการตรวจจับอุณหภูมิได้อย่างมากมายตามการประมวลผลของไมโครโปรเซสเซอร์ นอกจากนั้นแล้วยังสามารถติดตั้ง DS 1820 เพื่อการตรวจวัดอุณหภูมิได้ในหลายลักษณะและหลายสถานที่ ตำแหน่งการติดตั้งที่มีความแตกต่างอย่างมากมายกับอุปกรณ์ทั่วไป ไม่ว่าจะเป็นการติดตั้งภายในอาคาร อุปกรณ์เครื่องใช้ต่าง ๆ หรือภายในเครื่องจักรก็สามารถติดตั้งได้ และเอาต์พุตที่เป็นตัวอนุกรมตัวเลขของ DS 1820 นี้จึงสามารถต่อเอาต์พุตบนสายสัญญาณเพียงเส้นเดียวได้หลาย ๆ ชุดโดยไม่สับสนข้อมูลซึ่งกันและกัน

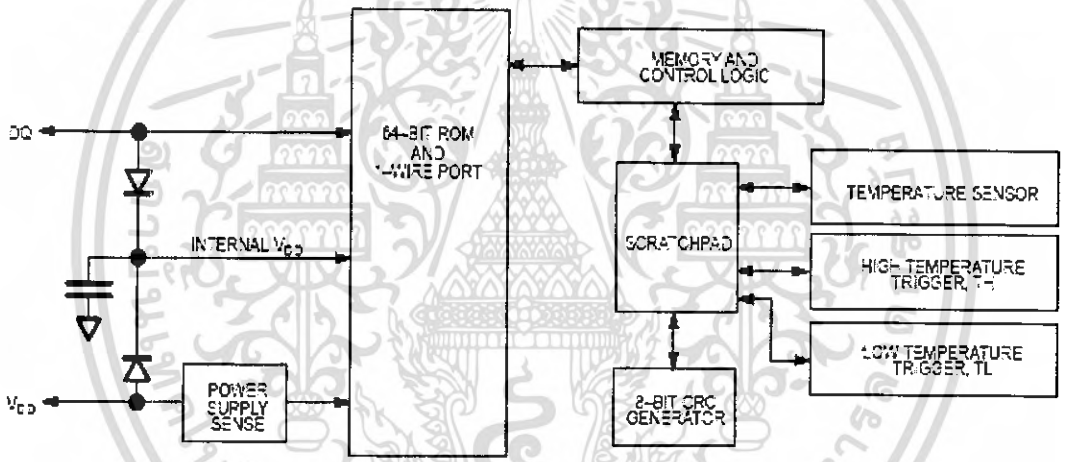


รูปที่ 2.7 ลักษณะตัวถังและการจัดขาของ DS1820

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.2 บล๊อกไดอะแกรมภายใน

จากรูป2.8 แสดงบล๊อกไดอะแกรมส่วนประกอบของการทำงานต่าง ๆ ภายในตัว DS 1820 จะเห็นว่ามีส่วนประกอบหลัก ๆ อยู่ 3 ส่วนด้วยกันคือ หน่วนความจำเลเซอร์รอมขนาด 64 บิต ส่วนเซนเซอร์อุณหภูมิและส่วนกระตุ้นเตือนอุณหภูมิแบบ non-voletile (TH and TL) โดยอุปกรณ์ตรวจวัดอุณหภูมินี้จะถูกควบคุมสภาวะการเพาเวอร์ออนและเพาเวอร์ออฟจากไลน์ข้อมูลเพียง 1 สายข้อมูลจากการเก็บรักษากำลังงานสำรองไว้ในตัวเก็บประภายใน ในช่วงระหว่างคาบเวลาเมื่อสัญญาณภายในไลน์มีสถานะเป็น HIGH และจะทำงานต่อเนื่องไปเรื่อย ๆ และการหยุดการทำงานก็จะเกิดขึ้นจากการหยุดจ่ายแหล่งจ่ายในช่วงระหว่างคาบเวลานั้นเป็น LOW ของไลน์ข้อมูลและจะหยุดอยู่เช่นนั้นจนกว่าขาไลน์ข้อมูลจะกลับมาเป็น HIGH อีกครั้งจึงจะเกิดการทำงานที่ DS 1820 และแหล่งจ่ายไฟหลักให้กับไอซีก็จะได้จากแหล่งจ่ายไฟ +5 โวลต์ภายนอก



รูปที่ 2.8 บล๊อกไดอะแกรมภายใน DS1820

การติดต่อข้อมูลกับDS 1820 จะติดต่อผ่านพอร์ตเพียงพอร์ตเดียวคือ 1-Wire port ภายในพอร์ต 1-Wire นี้ในส่วนของหน่วยความจำและควบคุมฟังก์ชันจะยังไม่รับรู้ข้อมูลใด ๆ ทั้งสิ้นก่อนที่ฟังก์ชันโปรโตคอลของ ROM จะถูกทำการเซตเสียก่อน ในส่วนสำคัญของการทำงานฟังก์ชันอันดับแรกซึ่งเป็นหนึ่งในห้าอันดับของการสั่งการฟังก์ชันใน ROM ก็คือ

1. อ่านหน่วนความจำ ROM
2. ทำการแมตซ์ ROM
3. ค้นหา ROM
4. กระโดดข้าม ROM เตือนการค้นหา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งการทำงานของระบบสั่งการนี้จะทำงานบนพื้นที่หน่วนความจำเลเซอร์รอมขนาด 64 บิต ผ่านพอร์ตของไอซีแต่ละตัวและสามารถให้เอาต์พุตเดียวเพื่อกำหนดคุณสมบัติของอุปกรณ์ตรวจจับอุณหภูมิหลาย ๆ ตัว โดยสั่งการผ่านไลน์ข้อมูล 1-Wire นี้ หลังจากฟังก์ชันใน ROM ถูกลำดับการทำงานแล้วก็พร้อมที่จะถูกใช้งานหรือเริ่มต้นการทำงานได้แล้ว และสามารถที่จะเข้าถึงการทำงานภายในตัวไอซีได้ทั้งหมด หน่วยความจำและส่วนควบคุมฟังก์ชันก็จะถูกเข้าถึงการทำงานได้และส่วนจัดเก็บค่าที่เซ็ตไว้สามารถหรืออาจจะถูกเก็บไว้ในพื้นที่ 1 ส่วนจากทั้งหมด 6 ส่วนของหน่วยความจำส่วนควบคุมฟังก์ชันการสั่งการ

ตารางที่ 2.1 แสดงคุณสมบัติทางไฟฟ้าของ DS 1820

พารามิเตอร์	สัญลักษณ์	ค่า	หน่วย
แรงดันไฟเลี้ยง	VDD	2.2 ถึง 5.5	V
แรงดันขาข้อมูล	I/O	(-0.5) ถึง (+5.5)	V
ความผิดพลาดของการวัดอุณหภูมิ	TERR	-0.5	^{o}C
กระแสซิงค์	IL	-0.4	mA
กระแสขณะแสตนด์บาย	IQ	200-300	nA
กระแสขณะทำงาน	IDD	1-1.5	mA
กระแสโหลดทางอินพุต	IL	5	uA
ค่าเวลาการแปลงอุณหภูมิ	TCONV	200-5000	mS
ค่าเวลาไทม์สล๊อต	TSLOT	60-120	uS
ค่าความจุ I/O	C iN/OUT	25	pF
เวลาอ่านข้อมูล	TRDV	15	uS
ย่านอุณหภูมิทำงาน	TO	(-55) ถึง(+125)	^{o}C

ส่วนควบคุมฟังก์ชันการสั่งการหนึ่งส่วนจะถูกกำหนดคุณสมบัติของ DS 1820 ให้อยู่ในรูปแบบของการวัดค่าของอุณหภูมิซึ่งผลของการวัดนี้จะถูกบันทึกไว้ใน DS 1820 ในส่วนของหน่วยความจำส่วนหนึ่ง (scratchpad) และบางครั้งก็จะอ่านออกมาได้จากตารางสารบัญของหน่วยความจำฟังก์ชันการสั่งการซึ่งเป็นการสั่งการเฉพาะหัวข้อที่ถูกบันทึกไว้ในหน่วยความจำ scratchpad สัญญาณกระตุ้นเตือนค่าอุณหภูมิสูงเกินและต่ำเกิน (THและ TL) จะประกอบด้วย 1 ไบต์ EEPROM ถ้าสัญญาณเตือนการค้นหาไม่ถูกจ่ายเข้าไปยัง DS 1820 รีจิสเตอร์เหล่านี้บางครั้ง จะถูกใช้ได้อย่างทั่ว ๆ ไปจากหน่วยความจำที่ผู้ใช้งานกำหนดได้และการเขียนเข้าไปในส่วนของการเตือน TH และ TL จะไม่ใช่หน่วยความจำฟังก์ชันสั่งงานและการอ่านเข้าไปถึงรีจิสเตอร์นี้จะอ่าน

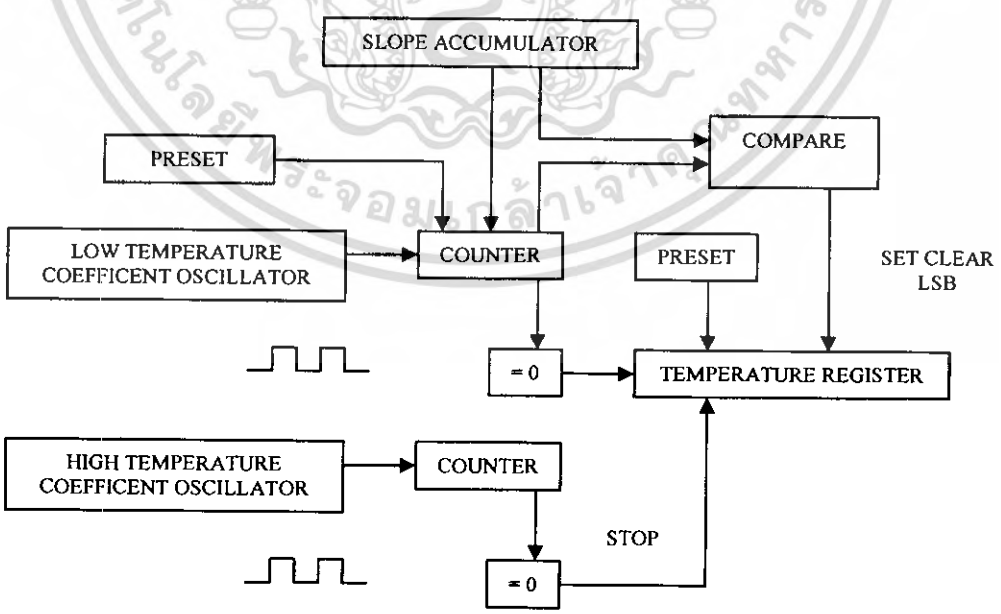
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ผ่านหน่วยความจำ scratchpad และข้อมูลอื่น ๆ ที่ต้องการอ่านและเขียนจะกระทำได้ในบิตแรกของ LSB

2.2.3 การทำงานในการวัดอุณหภูมิ

DS 1820 จะทำการวัดค่าอุณหภูมิโดยอาศัยเทคนิคการวัดแบบออนบอร์ดพิเศษ ซึ่งเป็นเทคนิคการวัดอุณหภูมิโดยเฉพาะของอุปกรณ์ชนิดนี้ ในรูปที่ 2.9 แสดงบล๊อกไดอะแกรมการวัดค่าอุณหภูมิของ DS 1820 ซึ่งอาศัยการวัดอุณหภูมิโดยการวัดจำนวนวงรอบของสัญญาณนาฬิกาที่ออสซิลเลเตอร์ผลิตขึ้นมา ช่วงเวลาเกตของสัญญาณนาฬิกาที่ออสซิลเลตขึ้นมาจะเป็นการกำหนดได้จากช่วงเวลาที่ค่าสัมประสิทธิ์อุณหภูมิต่ำ ไปจนถึงสัมประสิทธิ์อุณหภูมิสูง ซึ่งจะมีค่าความถี่สัญญาณนาฬิกาที่ไม่เท่ากัน โดยที่ค่าการนับตัวเลขจะเริ่มนับที่อุณหภูมิต่ำสุดพื้นฐาน คือ -55 องศาเซลเซียส ถ้าการนับสัญญาณนาฬิกามาถึงค่าศูนย์ก่อนที่ค่าเวลาเกตจะเกินมา รีจิสเตอร์อุณหภูมิก็จะแสดงผลที่ค่า -55 องศาเซลเซียส ถ้าหากค่าอุณหภูมิเพิ่มขึ้น การแสดงผลของอุณหภูมิขณะนั้นก็จะสูงกว่า -55 องศาเซลเซียส

ในทำนองเดียวกันนี้ การตั้งค่าของการนับจะกำหนดได้จากการเพิ่มความลาดลงของวงจรนับซึ่งวงจรนี้ต้องการการชดเชยสำหรับการแสดงคุณสมบัติของส่วนโค้งของออสซิลเลเตอร์ที่อุณหภูมิมีค่าเกินมา วงจรนับนั้นจะนับสัญญาณนาฬิกาอีกครั้งจนกว่ามันจะได้ค่าเป็นศูนย์ ถ้าคาบเวลาเกตอยู่ในสภาวะสงบนิ่งไม่มีการปรับแต่งก็จะเกิดการประมวลผลใหม่อีกครั้งหนึ่ง



รูปที่ 2.9 บล๊อกไดอะแกรมการวัดค่าอุณหภูมิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การคำนวณค่าภายใน DS 1820 จะให้ค่าความละเอียด 0.5 องศาเซลเซียสต่อสเต็ปของการเปลี่ยนแปลงอุณหภูมิ การอ่านค่าของอุณหภูมิจะถูกกำหนดไว้ภายใน 16 บิต โดยมีนัยสำคัญของตัวเลขสองส่วนประกอบการอ่าน ในตารางแสดงคุณลักษณะรายละเอียดความสัมพันธ์ของข้อมูลทางเอาต์พุตกับการจัดอุณหภูมิ ข้อมูลจะถูกส่งออกมาเป็นแบบอนุกรมบนการอินเตอร์เฟสกับสายข้อมูล 1-Wire ซึ่ง DS 1820 สามารถทำการวัดค่าอุณหภูมิได้เกินย่านตั้งแต่ -55 ถึง +125 องศาเซลเซียสที่ 0.5 องศาต่อสเต็ป ค่าอุณหภูมิที่ถูกทำการปรับตั้งไว้ใน DS 1820 ในเทอมของ 0.5 องศาเซลเซียส LSB ซึ่งเป็นไปตามแบบของข้อมูล 9 บิต

ที่ MSB บิตเป็นคู่เปรียบเทียบกับทุกบิตใน MSB สูงสุดของรีจิสเตอร์อุณหภูมิขนาด 2 ไบต์ในหน่วยความจำซึ่งการอ่านค่าอุณหภูมิแบบ 16 บิต ในลักษณะสำคัญต่าง ๆก็แสดงไว้ในตาราง

ตารางที่ 2.2 ความสัมพันธ์ของค่าอุณหภูมิกับข้อมูลดิจิตอลเอาต์พุต

ค่าอุณหภูมิ	ดิจิตอลเอาต์พุต (Binary)	ดิจิตอลเอาต์พุต(Hex)
+125 °C	00000000 11111010	00FAH
+25 °C	00000000 00110010	0032H
+1/2 °C	00000000 00000001	0001H
0 °C	00000000 00000000	0000H
-1/2 °C	11111111 11111111	FFFFH
-25 °C	11111111 11001110	FFCEH
-125 °C	11111111 10010010	FF92H

2.2.4 การทำงานของสัญญาณเตือน

หลังจากที่ DS 1820 มีการตรวจจับอุณหภูมิเกิดขึ้นแล้วค่าของอุณหภูมิก็จะทำการเปรียบเทียบเพื่อทำเป็นสัญญาณกระตุ้น การเปรียบเทียบค่าอุณหภูมิจะเปรียบเทียบกับค่าที่ถูกบันทึกหรือกำหนดไว้ของค่าอุณหภูมิสูงสุด (TH) และค่าของอุณหภูมิต่ำสุด (TL) ตลอดย่านอุณหภูมิที่วัดไว้โดยที่จะใช้พื้นที่รีจิสเตอร์ 8 บิตสำหรับการทำงานนี้ใน MSB ของ TH หรือ TL ที่ตรงกันก็จะถูกส่งไปยัง SB ของรีจิสเตอร์อุณหภูมิขนาด 16 บิต ถ้าผลของการวัดอุณหภูมิมีค่าสูงเกินกว่า TH หรือต่ำกว่า TL ลำดับสัญญาณเตือนภายในก็จะถูกเซต ซึ่งลำดับของสัญญาณเตือนนี้จะถูกอัพเดตทุกครั้งที่มีการวัดค่าอุณหภูมิ เมื่อลำดับช่องสัญญาณเตือนถูกเซต DS 1820 จะมีการตอบสนองนำไปสู่การ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

ค้นหาสัญญาณเตือนการสั่งการและจะยอมให้ทำการต่อ DS 1820 ในลักษณะขนานกันหลายตัวได้ เพื่อทำการจำลองการวัดค่าอุณหภูมิแล้วนำมาเฉลี่ยค่าของการวัดในครั้งนี้อีกขั้นตอนหนึ่ง

2.2.5 บิตเอเซอร์รอม

ใน DS 1820 นั้นจะประกอบด้วยส่วนของรหัสหน่วยความจำรอมที่มีความยาวถึง 64 บิต โดยใน 8 บิตแรกจะเป็นรหัสระบุตระกูล (family code) 1-Wireของ DS 1820 (DS 1820 มีรหัสเป็น 10H) และอีก 48 บิตต่อมาเป็นส่วนระบุอนุกรมตัวเลข (serial number) และอีก 8 บิต สุดท้ายคือส่วนบันทึก CRC ของ 56 บิตแรก ดังแสดงการแบ่งส่วนไว้ในรูปที่ 2.10 หน่วยความจำรอมขนาก 64 บิต และส่วนควบคุมฟังก์ชันรอมนี้จะยอมให้ DS1820 สามารถทำเป็นอุปกรณ์อินเตอร์เฟสแบบ 1-Wire ได้ และมีรายละเอียดตามโปรโตคอลของระบบบัส 1-Wire ซึ่งฟังก์ชันและส่วนควบคุมต่าง ๆ ใน DS 1820 จะยังไม่สามารถทำงานหรือเข้าถึงได้จนกว่าจะมีการเซตอัพโปรโตคอลฟังก์ชันในหน่วยความจำรอมเสียก่อน โดยในการอินเตอร์เฟสในส่วนหลักของฟังก์ชันการสั่งการในหน่วยความจำรอมจะต้องมีการลำดับฟังก์ชันดังนี้คือ 1. อ่านรอม 2. แมตซ์รอม 3. ค้นหารอม 4. กระโดดข้ามรอม 5. เตือนการค้นหา หลังจากที่มีการเซตกับฟังก์ชันรอมดังข้างต้นเสร็จแล้วฟังก์ชันต่าง ๆ ของ DS 1820 ก็จะสามารถเข้าถึงได้

8-BIT CRC CODE	48-BIT SERIAL NUMBER	8-BIT FAMILY CODE(10H)

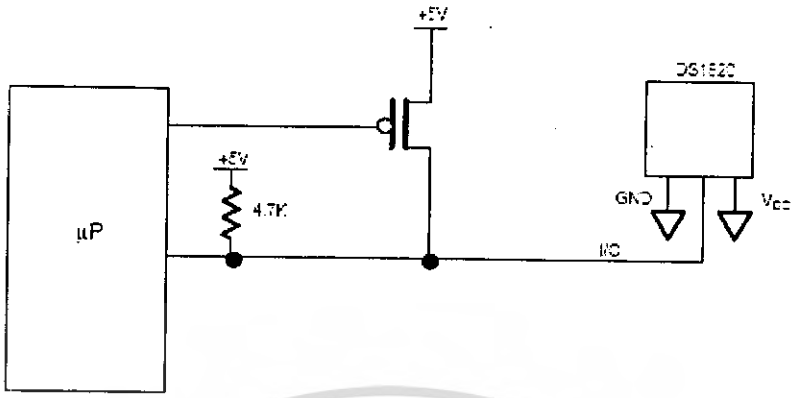
รูปที่ 2.10 การแบ่งส่วนในหน่วยความจำรอม 64 บิต

2.2.6 การจัดหาแหล่งจ่ายไฟ

ในรูปที่ 2.11 เป็นการต่อ DS 1820 ร่วมกับไมโครโปรเซสเซอร์ เพื่อการควบคุมจากระยะไกล จะสังเกตว่าที่ขารับไฟเลี้ยงของ VDD ของ DS 1820 นั้นจะถูกต่อไว้ดับกราวด์ แต่จะได้รับไฟเลี้ยงมาจากขา I/O ข้อมูลในแบบ 1-Wire จากระบบควบคุมหลักไมโครโปรเซสเซอร์แทน ซึ่งวิธีนี้จะใช้สำหรับการควบคุมระยะไกลและไม่ต้องจัดหาแหลงจ่ายไฟภายนอกให้กับ DS 1820 ให้ยุ่งยากเพราะในการอินเตอร์เฟสแบบ 1-Wire นี้จะสามารถไปกำหนดการทำงาน (Power on) ของไอซีได้จากคำสั่งของไมโครโปรเซสเซอร์จึงไม่สิ้นเปลืองกำลังและจะเป็นในลักษณะการกระตุ้นแหล่งจ่ายไฟสำรองที่เป็นตัวเก็บประจุภายในไอซีให้จ่ายไปเลี้ยงวงจรในตัวไอซีแทน

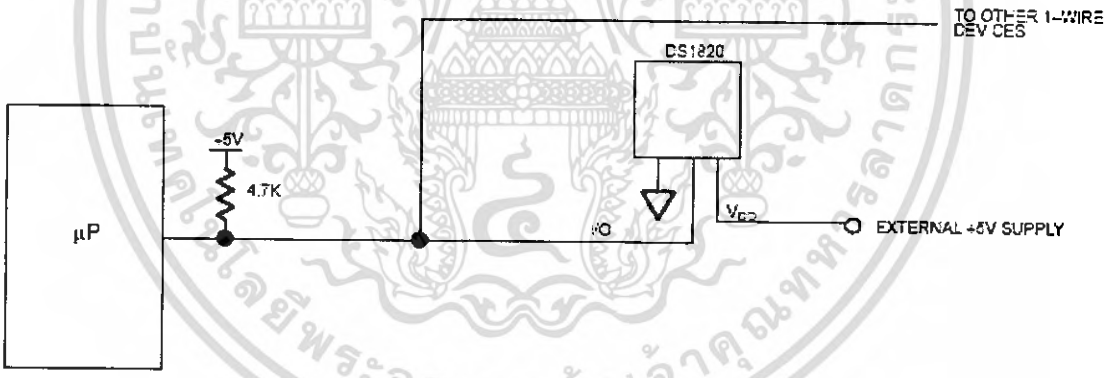
72114

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.11 การจัดหาแหล่งจ่ายไฟเลี้ยงให้กับ DS1820 ในบัส 1-Wire

ส่วนในรูป 2.12 เป็นการจัดแหล่งจ่ายไฟเลี้ยงภายนอกให้กับ DS 1820 ณ จุดที่ติดตั้งใช้งาน และการสั่งการจากไมโครโปรเซสเซอร์ก็ยังทำงานได้เหมือนเดิมและสามารถที่จะติดต่อ DS 1820 ขนานกันโดยใช้การอินเตอร์เฟสผ่านบัส 1-Wire เดียวกันได้หลายตัวเลยทีเดียว จึงเหมาะกับเป็นการตรวจวัดอุณหภูมิที่ควบคุมได้จากระยะไกลผ่านระบบไมโครโปรเซสเซอร์



รูปที่ 2.12 การจัดแหล่งจ่ายไฟภายนอกและการอินเตอร์เฟสร่วม DS1820 หลายตัวบนบัส 1-Wire

2.3 การอินเตอร์เฟสผ่านสายเส้นเดียว

การเชื่อมต่อหรือการอินเตอร์เฟสระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์ภายนอก โดยใช้จำนวนสายสัญญาณให้น้อยที่สุดได้มีการพัฒนามาอย่างต่อเนื่องจากหลายบริษัทผู้ผลิต เช่น การเชื่อมต่ออุปกรณ์ต่อพ่วงแบบอนุกรม (Serial Peripheral Interface , SPI) ในไมโครคอนโทรลเลอร์ 68HC11 ของ Motorolla การเชื่อมต่อแบบ SPI นี้ช่วยให้ไมโครคอนโทรลเลอร์แลกเปลี่ยนข้อมูลกับอุปกรณ์ต่อพ่วงได้ด้วยความเร็วถึง 1 ล้านบิตต่อวินาที โดยใช้สายรับส่งสัญญาณเพียง 3 หรือ 4 เส้น รวมกับสายกราวนด์อีกเส้น

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

นี่ไม่ใช่ระบบบัสที่ใช้จำนวนสายน้อยที่สุด ที่สามารถใช้ติดต่อกับอุปกรณ์อื่น ๆ เพราะระบบบัสแบบ I^2C bus ที่นิยมใช้ในไมโครคอนโทรลเลอร์ตระกูล 8051 สามารถส่งผ่านข้อมูลในแบบ 2 ทิศทาง (bidirectional data flow) โดยใช้สายสัญญาณ 2 เส้นกับกราวนด์อีกเส้นหนึ่ง

ปัจจุบันได้มีระบบบัสข้อมูลแบบสายเส้นเดียว (1-Wrie bus) ที่สนับสนุนการขนถ่ายข้อมูลแบบ 2 ทิศทางโดยใช้สายสัญญาณหนึ่งเส้นสำหรับเป็นสายสัญญาณนาฬิกาและสัญญาณข้อมูลและสายกราวด์อีก 1 เส้นเท่านั้น

อุปกรณ์เชื่อมต่อแบบ 1-Wire bus นี้ จะเน้นเรื่องการประยุกต์ใช้งานในด้านการรักษาความปลอดภัยของข้อมูล (data security) และการจำแนกอุปกรณ์ชนิดต่าง ๆ (identification) ยกตัวอย่างเช่น ชิป DS 1990A ซึ่งเป็นอุปกรณ์บรรจุหมายเลขประจำตัวไว้ข้างใน (Touch Serial Number device) โดยจะมีรูปร่างคล้าย ๆ กระดุมโลหะเล็ก ๆ ซึ่งมีหมายเลขประจำตัว (Serial Number) ขนาด 48 บิต ที่ไม่ซ้ำกับใครไว้ภายใน หมายเลขเหล่านี้สามารถแยกจากกันทางไฟฟ้าออกเป็น 2 ส่วน แต่ละส่วนจะทำหน้าที่เป็นสายกราวนด์และสายข้อมูลตามลำดับ

2.3.1 หลักการพื้นฐาน

อุปกรณ์ที่สนับสนุนระบบบัสเส้นเดียวจะมีสายเพียง 2 เส้นเท่านั้นคือสายกราวด์และสายสัญญาณ ซึ่งเรียกอีกอย่างว่าสาย DATA สายนี้จะจัดการเกี่ยวกับทั้งสัญญาณข้อมูลและสัญญาณนาฬิกาที่ใช้ทำการแลกเปลี่ยนข้อมูล สาย DATA นี้จะเป็นชนิด open-drain ดังนั้นในการออกแบบวงจร จะต้องออกแบบให้มีตัวต้านทานมาพูลอัพสาย DATA นี้ด้วย สามารถต่ออุปกรณ์ 1-Wire divice เข้ากันกับบัสนี้ได้มากกว่าเท่าที่ต้องการขา DATA ของไมโครคอนโทรลเลอร์ต้องเป็นแบบ 2 ทิศทาง (bidirectional) และมีเอาต์พุตแบบ open-drain

การแลกเปลี่ยนข้อมูลระหว่างไมโครคอนโทรลเลอร์กับอุปกรณ์ที่ใช้ 1-Wire bus นี้ไม่ได้ง่าย ๆ เหมือนการรับส่งข้อมูลผ่านทางบัส SPI เพราะในระบบ 1-Wire bus นั้นสาย DATA จะต้องจัดการเกี่ยวกับจังหวะเวลา (timing), ระดับสัญญาณ (level), และทิศทาง (direction) ของข้อมูลทั้งหมด ทำให้การเขียนซอฟแวร์ของไมโครคอนโทรลเลอร์ที่ติดต่อกับอุปกรณ์พวกนี้ต้องมีความซับซ้อนมากขึ้น อย่างไรก็ตามถ้ามีการออกแบบวงจรอย่างระมัดระวังแล้ว เราสามารถทำงานได้สำเร็จโดยไม่ต้องเขียนโปรแกรมขนาดใหญ่จนเกินไป

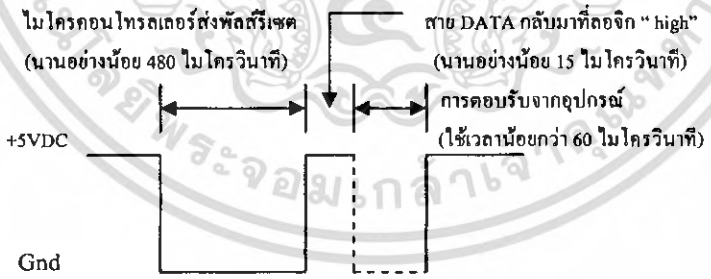
สิ่งที่น่าสนใจอย่างหนึ่งก็คือ อุปกรณ์ที่ใช้บัสแบบ 1-Wire bus ส่วนใหญ่จะไม่มีขาสัญญาณ DATA ซึ่งปกติจะต้องพูลอัพกับขาไฟเลี้ยง (Vcc) อยู่แล้ว สามารถจ่ายกระแส ให้กับอุปกรณ์เหล่านี้ได้อย่างเพียงพอระหว่างการทำงานตามปกติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ในสภาวะพัก (quiescent state) อุปกรณ์ที่ใช้บัสแบบ 1-Wire bus จะให้ขา DATA อยู่ในภาวะลอย (float) ทำให้ขานี้มีแรงดันเท่ากับแรงดันพูลอัพซึ่งปกติก็คือ 5 โวลต์นั่นเอง ส่วนไมโครคอนโทรลเลอร์ก็จะปล่อยให้ขาเอาต์พุตที่ใช้ติดต่อกับขา DATA นี้ในสภาวะความต้านทานสูง (hight-impedance state) เช่นกัน เมื่ออุปกรณ์ทั้งสองชนิดนี้ปล่อยให้ขา DATA อยู่ในสภาวะลอย นี้ความต้านทานพูลอัพก็จะช่วยรักษาระดับแรงดันไฟเลี้ยงที่จ่ายให้กับอุปกรณ์ที่ใช้บัสแบบ 1-Wire bus นี้ ได้อย่างสม่ำเสมอเพราะอุปกรณ์ที่ใช้บัสแบบ 1-Wire bus นี้จะใช้พลังงานในการทำงานน้อยมาก

ในการแลกเปลี่ยนข้อมูลกับไมโครคอนโทรลเลอร์และอุปกรณ์ที่ใช้บัสแบบ 1-Wire bus จะต้องดำเนินการอย่างระมัดระวัง ตามลำดับขั้นตอนในการทำให้สาย DATA มีลอจิกเป็น LOW ปล่อยให้สาย DATA กลับมามีลอจิกเป็น HIGH และการตรวจจับการตอบรับจากอุปกรณ์อีกด้านหนึ่ง ช่วงจังหวะเวลาที่ใช้ในกระบวนการนี้จะถูกกำหนดโดยข้อกำหนดโดยข้อกำหนดเฉพาะของระบบ 1-Wire bus นี้ ซึ่งต้องตรวจสอบความสามารถของระบบให้ดีเสียก่อนที่จะใช้ระบบบัสแบบ 1-Wire bus

จากสภาวะพักข้างต้นการแลกเปลี่ยนข้อมูลจะเริ่มขึ้นด้วยการที่ไมโครคอนโทรลเลอร์กระทำกระบวนการรีเซต (RESET sequence) ซึ่งทำได้โดยการทำให้สาย DATA มีลอจิก LOW เป็นเวลาอย่างน้อย 480 ไมโครวินาที แล้วจึงปล่อยให้กลับมาอยู่ในสภาวะ HIGH อีกครั้งหนึ่ง ให้ดูแผนภูมิเวลาในรูปที่ 2.13 ประกอบด้วย



รูปที่ 2.13 จังหวะเวลาในการทำกระบวนการตรวจสอบว่ามีอุปกรณ์อยู่บนบัส

เมื่ออุปกรณ์ที่ใช้บัส 1-Wire bus นี้ตรวจพบสภาวะ RESET นี้ก็จะตอบสนองด้วยการส่งพัลส์กลับไปเพื่อบอกให้ไมโครคอนโทรลเลอร์รู้ว่าบนสายบัสนี้มีอุปกรณ์แบบ 1-Wire device กำลังทำงานอยู่โดยอุปกรณ์ 1-Wire device จะปล่อยให้สาย DATA อยู่ในลอจิก HIGH อย่างน้อย 15 ไมโครวินาที แต่ไม่เกิน 60 ไมโครวินาที จากนั้นมันก็จะทำให้สาย DATA ลงมามีลอจิก LOW ในช่วงนานประมาณ 60-240 ไมโครวินาท แล้วจึงปล่อยให้กลับไปมีลอจิด HIGH เช่นเดิม ช่วงเวลา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

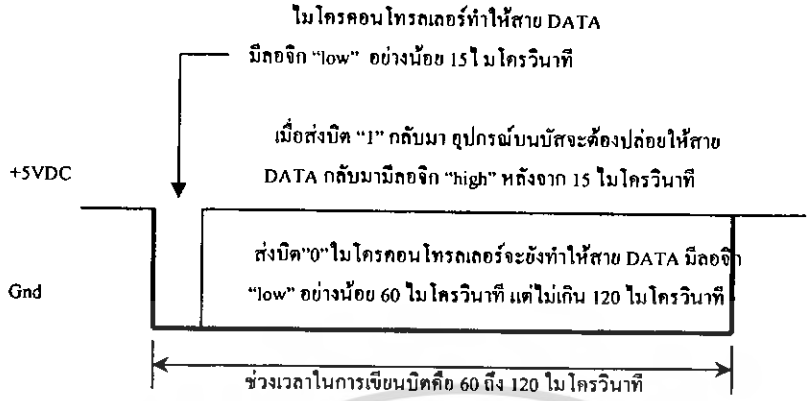
นี้เรียกกันหลายชื่อเช่น ช่วงเวลาเริ่มติดต่อ(Initalization) หรือเรียกว่าช่วงเวลาตรวจสอบว่ามีอุปกรณ์ต่อยู่บนบัสนี้หรือไม่ (presence detect)

หลังจากอุปกรณ์ 1-Wire device ปล่อยให้สาย DATA กลับมาอยู่ในลอจิก HIGHแล้ว ไมโครคอนโทรลเลอร์จะต้องปล่อยสาย DATA อยู่ในลอจิกนี้นานอย่างน้อย 240 ไมโครวินาที ก่อนที่จะเริ่มกระบวนการแลกเปลี่ยนข้อมูล (data exchange) ต่อไป จากนั้นไมโครคอนโทรลเลอร์ก็จะส่งคำสั่งเริ่มต้น (initial command) ขนาด 1 ไบต์ไปยังอุปกรณ์ 1-Wire bus ซึ่งคำสั่งนี้อาจจะเป็นคำสั่งอะไรก็ได้ขึ้นอยู่ว่าต้องกานให้อุปกรณ์ 1-Wire device ทำอะไร อุปกรณ์ 1-Wire device นี้จะรองรับชุดคำสั่งนั้นออกไป โดยการเลี่ยนสถานะของสาย DATA กลับกลับมาโดยตอนแรกจะให้เป็นลอจิก LOW แล้วจึงกลับมาเป็น HIGH ตามช่วงจังหวะเวลาที่เหมาะสม ช่วงเวลานานที่ไมโครคอนโทรลเลอร์ทำให้สาย DATA มีลอจิก LOW จะเป็นตัวแยกแยะว่าค่าบิตไหนที่มีลอจิกเป็น 1 บิตไหนมีลอจิกเป็น 0 ให้ดูแผนภูมิเวลาที่แสดงจังหวะเวลาในการเขียนบิต 1 หรือเขียนบิต 0 ในรูปที่ 2.14 ประกอบด้วย

ในทั้งสองกรณีไมโครคอนโทรลเลอร์จะเริ่มด้อนโดยการทำให้สาย DATA มีลอจิก LOW เสียก่อน เมื่อส่งบิต 1 ไมโครคอนโทรลเลอร์ก็จะปล่อยให้สาย DATA กลับมามีลอจิก HIGH ภายใน 1-15 ไมโครวินาทีหลังจากทำให้สาย DATA มีลอจิกLOW แต่ถ้าต้องการจะส่งบิต 0 ไมโครคอนโทรลเลอร์ก็จะยังคงปล่อยให้สาย DATA คงมีลอจิก LOW ต่อไปเป็นเวลานานอย่างน้อย 60-120 ไมโครวินาที

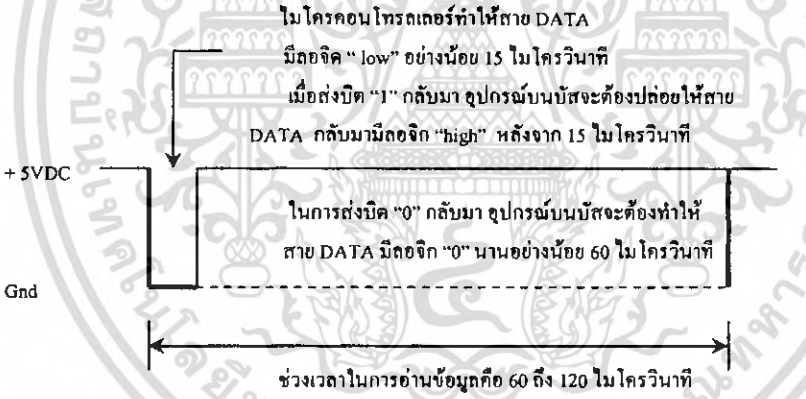
ช่วงเวลาระหว่าง 15-120 ไมโครวินาทีหลังจากไมโครคอนโทรลเลอร์ทำให้สาย DATA มีลอจิก LOW ในตอนแรกนี้ จะเรียกว่าช่วงการอ่านสถานะบิตข้อมูล (sampling window) ไมโครคอนโทรลเลอร์จะต้องรักษาระดับลอจิกของสาย DATA ให้คงที่ตลอดช่วงเวลานี้เพราะระดับลอจิกในช่วงเวลานี้จะเป็นตัวบอกว่าบิตคำสั่งนี้เป็นบิต 1 หรือบิต 0 ให้สังเกตว่าเมื่อจะเขียนบิต 1 ลงบนสาย DATA นี้ ไมโครคอนโทรลเลอร์จะให้สาย DATA กลับมามีลอจิก HIGH ในช่วงเวลาใดก็ได้ระหว่าง 1-15 ไมโครวินาทีหลังจากทำให้สาย DATA มีลอจิก LOW แล้วแต่ต้องไม่เกิน 15 ไมโครวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.14 จังหวะเวลาที่ไมโครคอนโทรลเลอร์ใช้เขียนบิตข้อมูล 0 หรือ 1 ไปยังอุปกรณ์บนบัส

หลังจากส่งคำสั่งไปยังอุปกรณ์ 1-Wire device แล้ว ไมโครคอนโทรลเลอร์ก็จะต้องรอการตอบรับ โดยการอ่านค่าบิตข้อมูลทีละบิตในช่วงเวลาที่แน่นอนค่าหนึ่ง ตามข้อกำหนดของบัสแบบ 1-Wire bus โดยให้ดูแผนภูมิเวลาในการอ่านบิต 0หรือ 1 จากรูปที่ 2.15



รูปที่ 2.15 จังหวะเวลาสำหรับไมโครคอนโทรลเลอร์ในการอ่านบิตข้อมูล 0 หรือ 1 จากอุปกรณ์บนบัส

ในการอ่านบิตข้อมูลแบบบัส 1-Wire bus ตอนแรกไมโครคอนโทรลเลอร์จะต้องทำให้สาย DATA มีลอจิก LOQ เป็นเวลานานไม่เกิน 15 ไมโครวินาที แล้วจึงปล่อยให้สาย DATA กลับมามีลอจิก HIGH เช่นเดิม จากนั้นอุปกรณ์ 1-Wire device ก็จะเข้าควบคุมสาย DATA แทน โดยจะส่งบิต 0 โดยจะทำให้สาย DATA มีลอจิกเป็น LOW และส่งบิด 1 โดยปล่อยให้สาย DATA กลับมามีลอจิก HIGH ตามเดิม

เมื่อส่งข้อมูลออกไปยังไมโครคอนโทรลเลอร์เรียบร้อยแล้ว จะมีการพักอยู่ชั่วขณะหนึ่ง จากนั้นอุปกรณ์ 1-Wire device จะปล่อยจากรควบคุมจากสาย DATA ให้เป็นอิสระ และรอรับคำสั่ง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การอ่านข้อมูลครั้งต่อไป ถ้าอุปกรณ์ 1-Wire device ส่งบิต 0 ออกไป ไมโครคอนโทรลเลอร์ก็จะสามารถตรวจสอบจุดสิ้นสุดของบิต 0 นี้ได้ง่าย ๆ เพราะสาย DATA จะกลับมาอยู่ที่ลอจิก HIGH ตามเดิม แต่ถ้าการตรวจสอบจุดสิ้นสุดของการส่งบิต 1 ของอุปกรณ์ 1-Wire device จะต้องใช้เทคนิคมากกว่านี้ เพราะสาย DATA นี้จะอยู่ที่ลอจิก HIGH อยู่แล้ว

นี้เป็นเหตุผลว่าทำไมจังหวะเวลาในการอ่านเขียนข้อมูลจึงเป็นเรื่องที่สำคัญมาก เมื่อจะอ่านบิต 1 จากอุปกรณ์ 1-Wire device ไมโครคอนโทรลเลอร์จะต้องทำตามช่วงเวลาที่แสดงในแผนภูมิเวลาอย่างเคร่งครัด และต้องไม่ทำการอ่านลอจิกถัดมา จนกว่าเวลาจะผ่านไปแล้วอย่างน้อย 60 ไมโครวินาที

2.4 TRW

Mode of operation (โหมดดำเนินการ)

2.4.1 Overview

ระบบย่อยของ nRF2401 สามารถจัดอยู่ในโหมดหลักได้ดังต่อไปนี้ โดยจะขึ้นกับ 3 pins ควบคุม

Mode	PWR_UP	CE	CS
Active	1	1	0
Configuration	1	0	1
Stand by	1	0	0
Power down	0	X	x

ตารางที่ 2.3 nRF2401 subsystem main modes

2.4.2 Active modes

ระบบย่อย nRF2401 มี 2 Active โหมด (TX / Rx) :

- Shock Burst
- Direct Mode (not supported by nRF24E1)

ฟังก์ชันการทำงานของอุปกรณ์ในโหมดนี้เลือกโดย เนื้อหาของ Configuration word โดย Configuration word จะแสดงในส่วนของ Configuration

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3 Shock Burst

เทคโนโลยี Shock Burst ใช้บนชิป FIFO เพื่อจับเวลาข้อมูลที่อัตราข้อมูลต่ำและส่งต่อที่อัตราข้อมูลสูงมาก ด้วยเหตุนี้จะสามารถทำให้เกิดการลดทอนของกำลังสูงสุด เมื่อดำเนินการระบบย่อย nRF2401 ใน Shock Burst เราจะได้อัตราข้อมูลสูงสุด (1 Mbps) โดย band 2.4 GHz โดยไม่ต้องการไมโครคอนโทรลเลอร์ความเร็วสูงและราคาสูง (MCU) สำหรับการดำเนินการทางข้อมูล

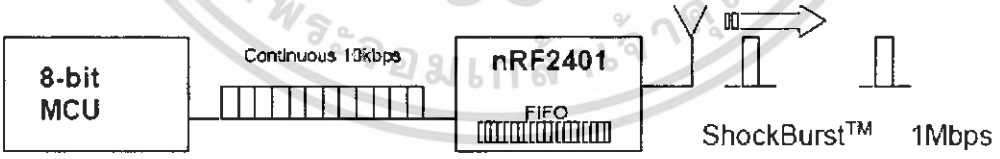
โดยการใส่การประมวลผลข้อมูลความเร็วสูง สัมพันธ์กับ RF Protocol บนชิป nRF24E1 มีประโยชน์ ดังนี้

- ลดการใช้กระแส
- ระบบที่มีราคาถูก (ใช้งานได้ง่ายสำหรับ microcontroller ราคาไม่แพง)
- ลดอัตราการเสี่ยงของการชนกันในอากาศ เนื่องจากการส่งสัญญาณในช่วงเวลาสั้นๆ ได้ดีมาก

ระบบย่อย nRF 2401 สามารถโปรแกรมโดยใช้ สาย interface พื้นฐาน 3 เส้น ซึ่งอัตราของข้อมูลจะเลือกโดยความเร็ว CPU โดยยอมรับส่วนดิจิตอลของ application เพื่อจะ run ที่ความเร็วต่ำ ขณะที่อัตราข้อมูลบน RF Link มีค่าเพิ่มสูงสุด โหมด Shock Burst จะลดการใช้กระแสเฉลี่ยใน application

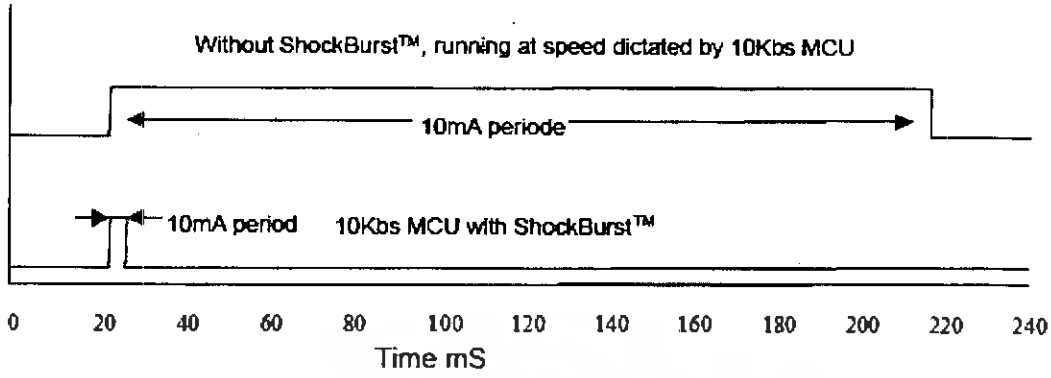
2.4.3.1 หลักการของ Shock Burst

เมื่อระบบย่อย nRF 2401 มีโครงร่างใน Shock Burst กระบวนการ Tx จะดำเนินการดังต่อไปนี้ ตัวอย่าง 10 kbps

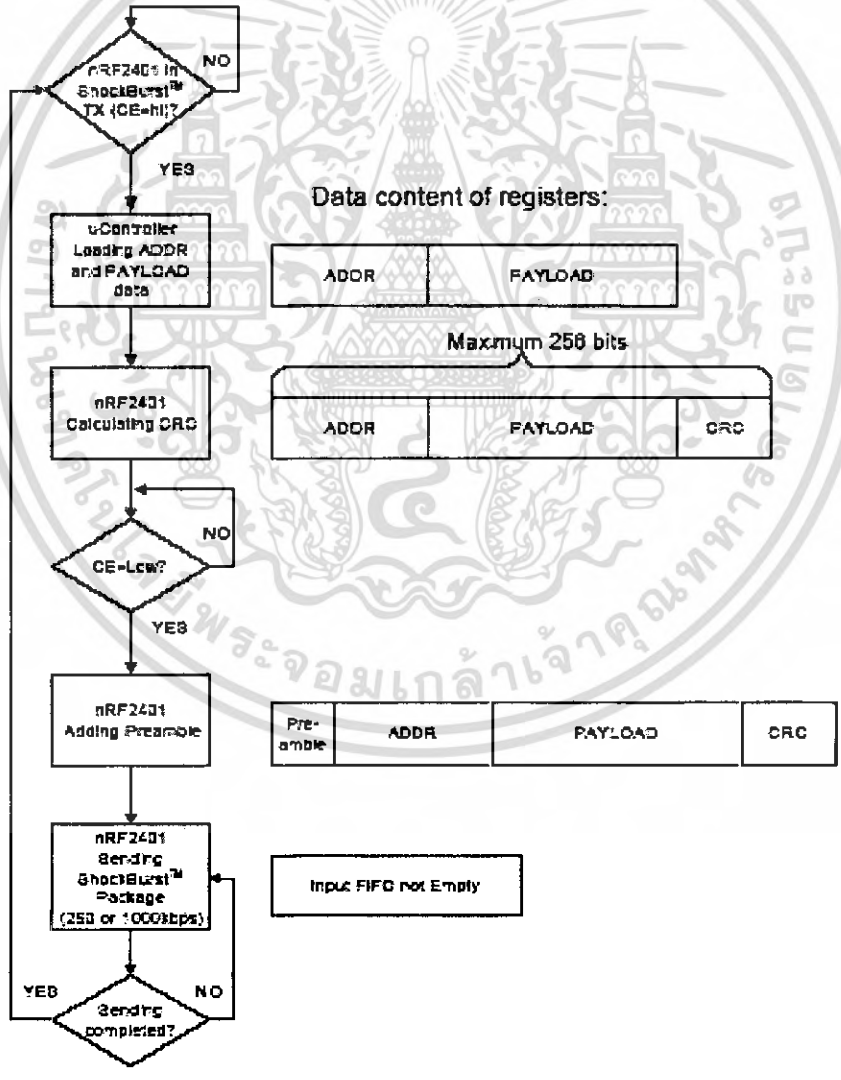


รูป 2.16 จับเวลาข้อมูลโดย CPU ส่งโดยเทคโนโลยี Shock Burst

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.17 การใช้กระแส RF โดยใช้และไม่ใช้เทคโนโลยี Shock Burst



รูปที่ 2.18 Flow Chart ShockBurst การส่งของระบบย่อย nRF2401

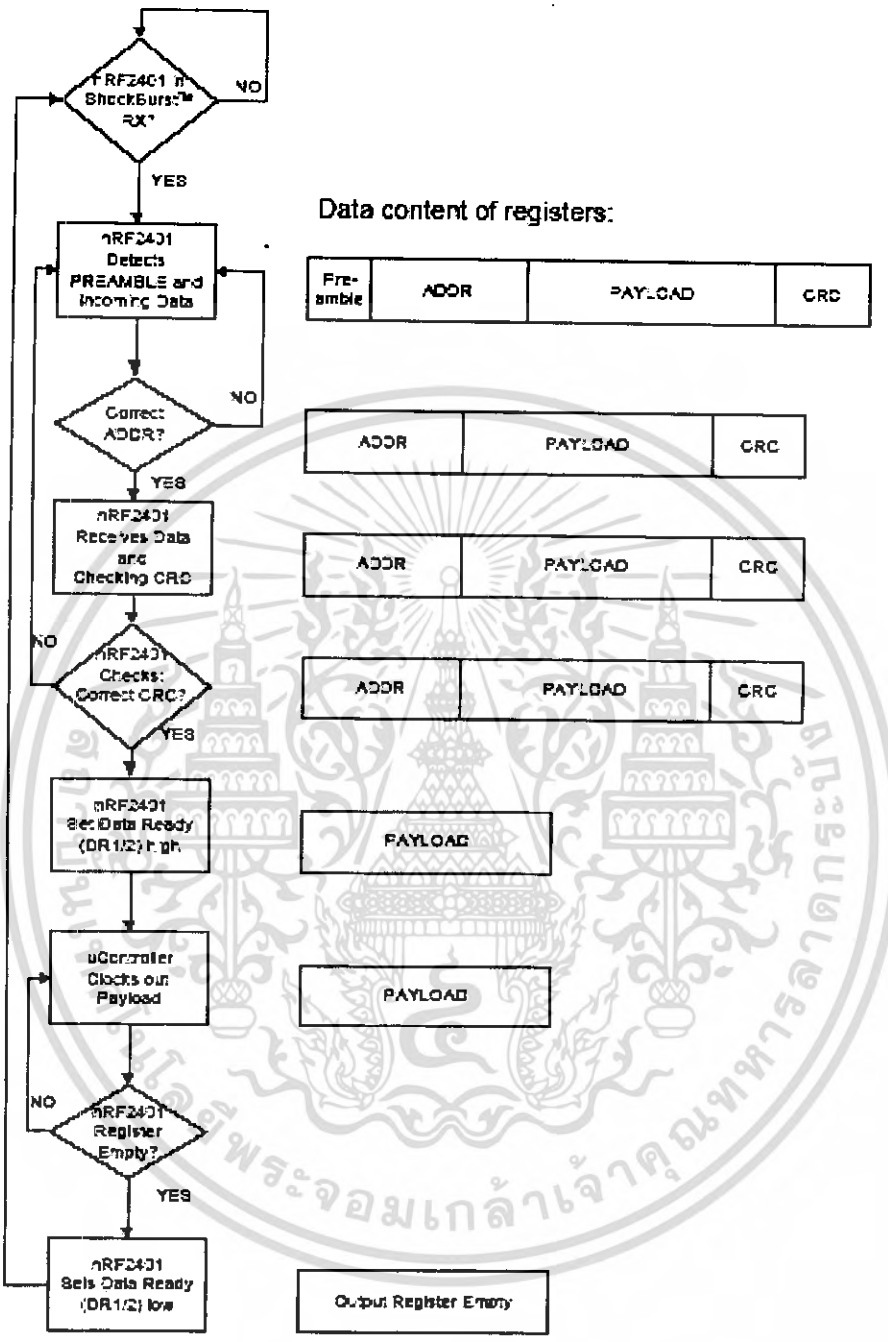
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.3.2 การส่ง Shock Burst

CPU interface pins : CE, CLK1, DATA

1. เมื่อ CPU มีข้อมูลที่จะส่ง set CE high คือ กระตุ้น nRF2401 ให้ประมวลผลบนบอร์ด
2. แอดเดรสของโหมดรับ (Rx) และ payload ข้อมูลจะจับเวลาเข้าสู่ช่วงเวลาระบบย่อย nRF2401 protocol หรือ CPU ตั้งความเร็วน้อยกว่า 1 Mbps
3. CPU ตั้ง CE low คือ กระตุ้นการส่ง Shock Burst
4. Shock Burst
    - RF front end จะมีกำลังเพิ่มขึ้น
    - RF Package จะสมบูรณ์
    - ข้อมูลจะถูกส่งที่ความเร็วสูง (250 bps หรือ 1Mbps แล้วแต่ผู้ใช้)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.19 Flow Cart ShockBurst การรับระบบย่อย nRF2401

2.4.3.3 <u>การรับ Shock Burst</u>

CPU interface pins : CE, DRT, CLK1, DATA (1 ช่องรับ Rx)

1. แอดเดรสถูกต้องและขนาดของ payload ของ RF package ที่เข้ามาจะ set เมื่อระบบย่อย nRF 2401 เป็นโครงร่างของ Shock Burst Rx

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. เมื่อกระตุ้น Rx ; set CE high

3. หลังจากตั้ง 200us nRF 2401 จะตรวจสอบอากาศสำหรับการสื่อสารที่จะเข้ามา

4. เมื่อ package ที่ถูกต้องถูกเรียกมา (แอดเดรสถูกต้องและพบ CRC) nRF 2401 จะทำการย้ายบทนำแอดเดรสและบิต CRC

5. จากนั้นระบบย่อย nRF 2401 จะ interrupt CPU โดยการตั้ง DR1 high

6. CPU จะ set CE low เพื่อป้องกัน RF front end (โหมดที่กระแสต่ำ)

7. CPU จะจับเวลาที่หยุด payload data ที่อัตราที่เหมาะสม

8. เมื่อ payload data ทั้งหมดถูกเรียกมา nRF 2401 set DR1 low อีกครั้ง และพร้อมสำหรับข้อมูลที่จะเข้ามา ถ้า CE ยัง high ระหว่างที่ download ข้อมูล ถ้า CE set low การลำดับ start up ใหม่จะสามารถเริ่มได้ ดูตารางที่ 2.4

RF_CH#							RXEN
7	6	5	4	3	2	1	0

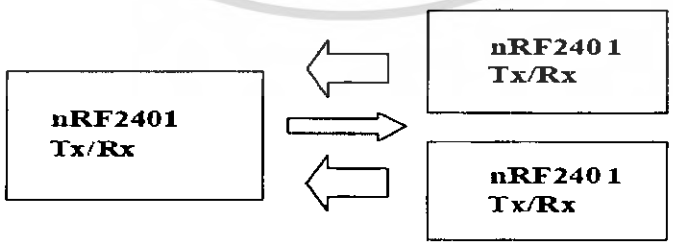
ตารางที่ 2.4 ช่องความถี่แลการ set Rx

2.4.4 DUOCiever Simultaneous two channel Receive mode

ในโหมด Shock Burst ของ nRF 24E1 สามารถรองรับความถี่ 2channel ที่ขนานกันละเป็นอิระต่อกัน ที่อัตราข้อมูลสูงสุด ได้สะดวกในเวลาเดียวกัน ซึ่งหมายความว่า

- RF 2401 สามารถรับข้อมูลจากเครื่องส่ง 2 เครื่อง ที่1 Mbps และ 8 MHz ผ่าน 1 เสาอากาศ
- Output จาก 2 ช่องสัญญาณข้อมูล จะป้อนให้กับ 2 interface pins ที่แยกจากกัน
- Data Channel 1: CLK1, DATA, DR1
- Data Channel 2: CLK2, DATA2, DR2

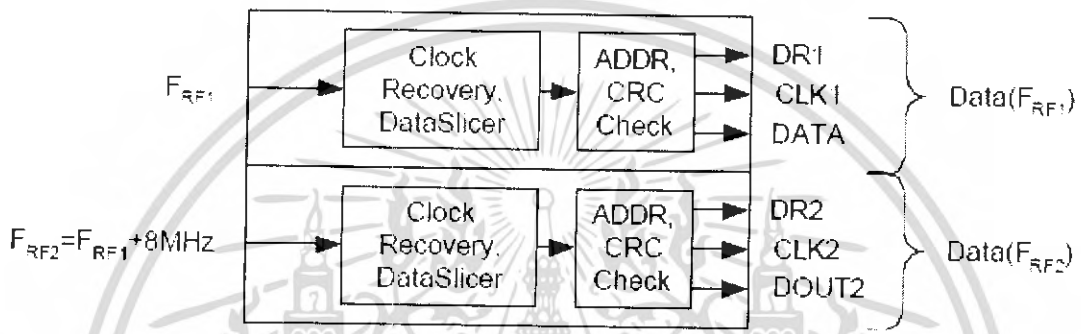
เทคโนโลยี DUOCiever ให้ 2 ช่องสัญญาณข้อมูลที่แยกกันอย่างละเอียดสำหรับ Rx และตอบสนองความต้องการต่อ 2 channel โดยมีระบบรับเพียง 1



รูปที่ 2.20 Simultaneous 2 channel receiver on nRF24E1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำหรับ nRF24E1 จะสามารถรับที่ช่องสัญญาณข้อมูลช่องที่ 2 ได้นั้น ความถี่ช่องสัญญาณต้องอยู่สูงกว่าอยู่ 8 MHz ความถี่ของช่องสัญญาณข้อมูล nRF2401 ต้อง program เพื่อรับข้อมูลที่ความถี่ของช่องสัญญาณที่1 ไม่มีการใช้เวลา multiplexing เพื่อทำฟังก์ชันนี้ให้สำเร็จ ใน Direct Mode ถ้ามันไม่รับส่งได้พร้อมกันระหว่าง 2 ช่องสัญญาณข้อมูล CPU ต้องสามารถจัดการข้อมูลที่เข้ามาพร้อมกันได้ ส่วนใน Shock Burst สามารถทำให้ CPU หยุดจับเวลาข้อมูลช่วงนึง ขณะที่ไม่มีเกิดการสูญเสีย package ข้อมูลและไม่ลดประสิทธิภาพของ CPU



รูปที่ 2.21 DUOCiever โดยมี 2 ช่องสัญญาณรับข้อมูลที่เป็นอิสระต่อกัน

2.4.5 Device configuration

Configuration ทั้งหมดของระบบย่อย nRF2401 จะสามารถทำผ่านทาง 3-wire interface ของการติดต่อสื่อสาร (CS, CLK1, DATA) กับ register เดี่ยว โดย configuration word สามารถมีได้ถึง 18 bytes ส่วนconfiguration bit (DATA) จะต้องจับเวลาโดย clocked (CLK1) โดย msb มาก่อนขณะที่ CS = 1 ไม่เกิน 18 bytes อาจถูกดาวน์โหลด

2.4.5.1 Configuration for ShockBurst operation (Configuration สำหรับ การดำเนินการ ShockBurst)

Configuration word ใน Shock Burst ทำให้ระบบย่อย nRF2401 สามารถจัดการกับ RF protocol ได้ หาก protocol สมบูรณ์ และโหลดไประบบย่อย nRF2401 เพียง 1 byte :bit[7:0] จำเป็นที่จะต้อง update ตลอดช่วงที่ดำเนินการ

Configuration block โดยละเอียด สำหรับ Shock Burst ดังนี้

- ความกว้างของ payload: ระบุจำนวนของบิต payload ใน RF package ซึ่งมันทำให้ระบบย่อย nRF2401 สามารถแบ่งแยกระหว่างข้อมูล payload และ CRC bytes ในตัวรับ package ได้
- ความกว้าง address: set จำนวนบิตที่แอดเดรสใช้ใน RF package ซึ่งมันทำให้ระบบย่อย nRF2401 สามารถแบ่งแยกระหว่างแอดเดรสและข้อมูล package ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- แอดเดรส (Rx ช่อง 1 และ 2) : กำหนดแอดเดรสสำหรับข้อมูล

- CRC : สามารถทำให้ on-chip CRC generate และถอดรหัสได้

Configuration Block นี้ โดยการยกเว้น CRC จะละเอียดสำหรับ package ซึ่งระบบ nRF2401 ที่จะรับในโหมด Tx CPU ต้อง generate แอดเดรส และส่วน payload ให้เหมาะกับระบบย่อย nRF2401 เพื่อจะรับข้อมูลเมื่อใช้ระบบย่อย nRF2401 on-chip รูปแบบบ CRC ต้องแน่ใจว่า CRC นั้นใช้ได้และใช้ที่ความยาวเดียวกันทั้งอุปกรณ์ Tx และ Rx

PRE-AMBLE	ADDRESS	PAYLOAD	CRC

รูปที่ 2.22 DATA package set-up

2.4.5.2 Configuration for Direct Mode operation (Configuration สำหรับการดำเนินการใน Direct Mode)

สำหรับใน Direct mode จะดำเนินการเพียงแค่ใน 2 ไบต์แรกของ Configuration word ที่เกี่ยวข้อง

	Bit position	Number of bits	Name	Function
ShockBurst™ configuration	143 120	24	TEST	Reserved for testing
	119 112	8	DATA2_W	Length of data payload section RX channel 2
	111 104	8	DATA1_W	Length of data payload section RX channel 1
	103:64	40	ADDR2	Up to 5 byte address for RX channel 2
	63 24	40	ADDR1	Up to 5 byte address for RX channel 1
	23 18	6	ADDR_W	Number of address bits (both RX channels)
	17	1	CRC_L	8 or 16 bit CRC
	16	1	CRC_EN	Enable on-chip CRC generation/checking
General device configuration	15	1	RX2_EN	Enable two channel receive mode
	14	1	CM	Communication mode (Direct or ShockBurst™)
	13	1	RFDR_SB	RF data rate (1Mbps requires 16MHz crystal)
	12 10	3	XO_F	Crystal frequency
	9 8	2	RF_PWR	RF output power
	7 1	7	RF_CH	Frequency channel
	0	1	RXEN	RX or TX operation

ตารางที่ 2.5 ตารางของ Configuration word

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Configuration word จะ shift MSB ก่อนที่ clock ขอบขาขึ้น configuration ใหม่จะเกิดที่ขอบขาลงของCS

2.4.6 Configuration Word Detailed Description (อธิบายรายละเอียดของ configuration word)

อธิบายฟังก์ชันของ 144 บิต (บิต 143 = MSB) ซึ่งใช้ configure ระบบ nRF2401

Configuration อุปกรณ์ทั่วไป: bit [15:0]

Configuration Shock Burst: bit [119:0]

Test Configuration: bit [143:120]

MSB	TEST						
D143	D142	D141	D140	D139	D138	D137	D136
Reserved for testing							
1	0	0	0	1	1	1	0	Default

MSB	TEST														
D135	D134	D133	D132	D131	D130	D129	D128	D127	D126	D125	D124	D123	D122	D121	D120
Reserved for testing														Close PLL in TX	
0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0	Default

DATA2_W							
D119	D118	D117	D116	D115	D114	D113	D112
Data width channel=2 in # of bits excluding addr crc							
0	0	1	0	0	0	0	0	Default

DATA1_W							
D111	D110	D109	D108	D107	D106	D105	D104
Data width channel=1 in # of bits excluding addr crc							
0	0	1	0	0	0	0	0	Default

ADDR2												
D103	D102	D101	....	D71	D70	D69	D68	D67	D66	D65	D64	
Channel=2 Address RX (up to 40bit)												
0	0	0	...	1	1	1	0	0	1	1	1	Default

ADDR1												
D63	D62	D61	....	D31	D30	D29	D28	D27	D26	D25	D24	
Channel=1 Address RX (up to 40bit)												
0	0	0	...	1	1	1	0	0	1	1	1	Default

ADDR_W						
D23	D22	D21	D20	D19	D18	
Address width in # of bits (both channels)						
0	0	1	0	0	0	Default

CRC		
D17	D16	
CRC Mode 1 = 16bit, 0 = 8bit	CRC 1 = enable, 0 = disable	
0	1	Default

RF-Programming															LSB	
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	
Two Ch	BUF	OD	NO Frequency			RF Power		Channel selection							RXEN	
0	0	0	0	1	1	1	1	0	0	0	0	0	1	0	0	Default

ตารางที่ 2.6 Configuration data word

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดย MSB bit ควรจะถูกโหลดสู่ configuration register เป็นอันดับแรก

Default Configuration word: h8E08.1C20.2000.0000.00E7.0000.0000.E721.0F04.

2.4.6.1 Shock Burst configuration:

ช่วงบิต [119:16] ประกอบด้วย segments อย่างละเอียดของ configuration register ไปยัง protocol การทำงานของ Shock Burst บิตรวมของ 120 บิตต้อง shift เข้าเพื่อจะ switch ระหว่าง Rx และ Tx

2.4.6.1.1 PLL_CTRL

PLL_CTRL		
D121	D120	PLL
0	0	Open TX Closed RX
0	1	Open TX Open RX
1	0	Closed TX Closed RX
1	1	Closed TX Open RX

ตารางที่ 2.7 PLL setting

Bit 121-120:

PLL_CTRL ควบคุมการ set ของ PLL ใน Tx จะไม่เกิดการเบี่ยงเบน สำหรับโหมดดำเนินการปกติ 2 บิตนี้ต้องเป็น bit low

2.6.6.1.2 DATAx_W

DATA2_W							
119	118	117	116	115	114	113	112

DATA1_W							
111	110	109	108	107	106	105	104

ตารางที่ 2.8 จำนวนของบิตใน payload

Bit 119-112:

DATA2_W ความยาว package ของ RF payload สำหรับสัญญาณช่อง 2

Bit 111-104:

DATA1_W ความยาว package ของ RF payload สำหรับสัญญาณช่อง 1

NOTE:

จำนวนบิตทั้งหมดใน Shock Burst RF package อาจไม่เกิน 256! ความยาวสูงสุดของส่วน payload หาได้จาก DATAx_W (bits) = 256 – ADDR_W – CRC

โดย ADD_W: ความยาวแอดเดรสของ Rx ที่ set ใน configuration word Bit[23:18]

CRC: Check sum 8 หรือ 16 บิต ที่เซ็ตใน configuration word Bit[17]

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PRE: Preamble 8 บิตซึ่งรวมอัตโนมัติ

แอดเดรสและCRC ที่สั้นขึ้น จะทำให้มีที่สำหรับ payload data มากขึ้น

2.4.6.1.3 ADDRx

ADDR2											
103	102	101	....	71	70	69	68	67	66	65	64

ADDR1											
63	62	61	....	31	30	29	28	27	26	25	24

ตารางที่ 2.9 แอดเดรสของตัวรับ 1 และ 2

Bit 103-64:

ตัวรับ ADDR2 แอดเดรส channel2 มีได้ถึง 40 บิต

Bit 63-24:

ตัวรับ ADDR1 แอดเดรส channel1 มีได้ถึง 40 บิต

NOTE:

บิตใน ADDRx มากกว่าความกว้างที่ set ใน ADDR_W ซึ่งยืดยาวเกินไปและสามารถ set เป็น logic 0 ได้

2.4.6.1.4 ADDR_W & CRC

						ADDR_W		CRC_L	CRC_EN
23	22	21	20	19	18	17	16		

ตารางที่ 2.10 จำนวนบิตที่ต้องคงไว้สำหรับ RX address + CRC setting

Bit 23-18:

ADDR_W :จำนวนบิตที่ต้องสงวนไว้สำหรับแอดเดรส Rx ใน Shock Burst

NOTE: จำนวนบิตแอดเดรสสูงสุดคือ 40 บิต (5 ไบต์) ค่าที่มากกว่า 40 บิต ADDR_W จะไม่สามารถใช้ได้

Bit 17:

CRC_L: ความยาว CRC ที่จะคำนวณใน Shock Burst

Logic 0: 8 bit CRC

Logic 1: 16 bit CRC

Bit 16:

CRC_EN: ทำให้ CRC on-chip Tx และ Rx ใช้ได้

Logic 0: on-chip CRC generation / checking disable

Logic 1: on-chip CRC generation / checking enable

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

NOTE: บิต 8 CRC อาจเพิ่มจำนวน payload ใน Shock Burst ได้แต่จะลดความสมบูรณ์ของระบบลง

2.4.6.2 General RF configuration

General RF configuration ของ configuration word จะควบคุม RF และ พารามิเตอร์ที่เกี่ยวกับอุปกรณ์

2.4.6.2.1 Modes

RX2_EN	CM	RFDR_SB	XO_F			RF_PWR	
15	14	13	12	11	10	9	8

ตารางที่ 2.11 RF operational setting

Bit 15: Rx2_EN

Logic 0: Channel 1 รับ

Logic 1: Channel 2 รับ

NOTE: ในตัวรับทั้ง 2 channel nRF 24E1 รับทั้ง 2 channel แยกความถี่ของช่องสัญญาณทันที ความถี่ของตัวรับ channel 1 จะ set ในบิต [7-1] ของ configuration word ตัวรับ channel 2 จะเหนือตัวรับ channel 1 อยู่ 8 channel

(8 MHz)

Bit 14:

Communication Mode:

Logic 0: ระบบย่อย nRF 2401 ดำเนินการใน Direct Mode

Logic 1: ระบบย่อย nRF 2401 ดำเนินการใน Shock Burst Mode

Bit 13:

RF Data Rate:

Logic 0: 250 kbps

Logic 1: 1 Mbps

NOTE: ใช้ 250 kbps แทน 1 Mbps จะช่วยปรับปรุง sensitivity ของตัวรับ 10 db 1Mbps ต้องใช้ crystal 16 MHz

Bit 12-10:

XO_F: เลือกความถี่ crystal ที่จะใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

XO FREQUENCY SELECTION			
D12	D11	D10	Crystal Frequency [MHz]
0	0	0	4
0	0	1	8
0	1	0	12
0	1	1	16
1	0	0	20

ตารางที่ 2.12 crystal frequency setting

Bit 9-8:

RF_PWR set nRF 24E1 RF กำลังของเอาท์พุทในโหมดการส่ง

RF OUTPUT POWER		
D9	D8	P [dBm]
0	0	-20
0	1	-10
1	0	-5
1	1	0

ตารางที่ 2.13 RF output power setting

2.6.6.2.2 RF channel & direction

RF_CH#							RXEN
7	6	5	4	3	2	1	0

ตาราง 2.14 ช่องความถี่ และ การ set Rx / Tx

Bit 7-1:

RF_CH# ตั้งความถี่ช่องสัญญาณของ nRF 24E1 ให้ดำเนินการความถี่ของช่องสัญญาณในการส่งได้โดย

Channel RF = 2400 MHz + RF_CH#*1.0 MHz

RF_CH#: อาจตั้งไว้ระหว่าง 2400 MHz และ 2527 MHz

NOTE: ช่วงที่มากกว่า 83 สามารถใช้ได้เพียงแค่ในอาณาเขตที่แน่ใจได้

ความถี่ช่วงสัญญาณในช่องข้อมูล2 หาได้โดย

Channel RF = 2400 MHz + RF_CH#*1.0 MHz + 8 MHz (รับที่ PIN#4)

RF_CH#: อาจตั้งไว้ระหว่าง 2400 MHz และ 2527 MHz

Bit 0:

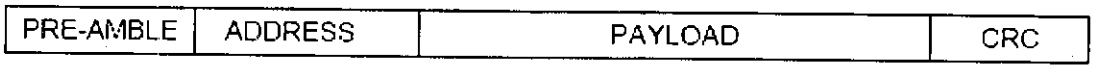
Active Mode

Logic 0: transmit mode โหมดส่ง

Logic 1: receive mode โหมดรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.7 Data package Description



รูปที่ 2.23 Data package Diagram

ชุดข้อมูลสำหรับการสื่อสารใน Shock Burst mode และ Direct mode จะแบ่งเป็น 4 ส่วน ดังนี้

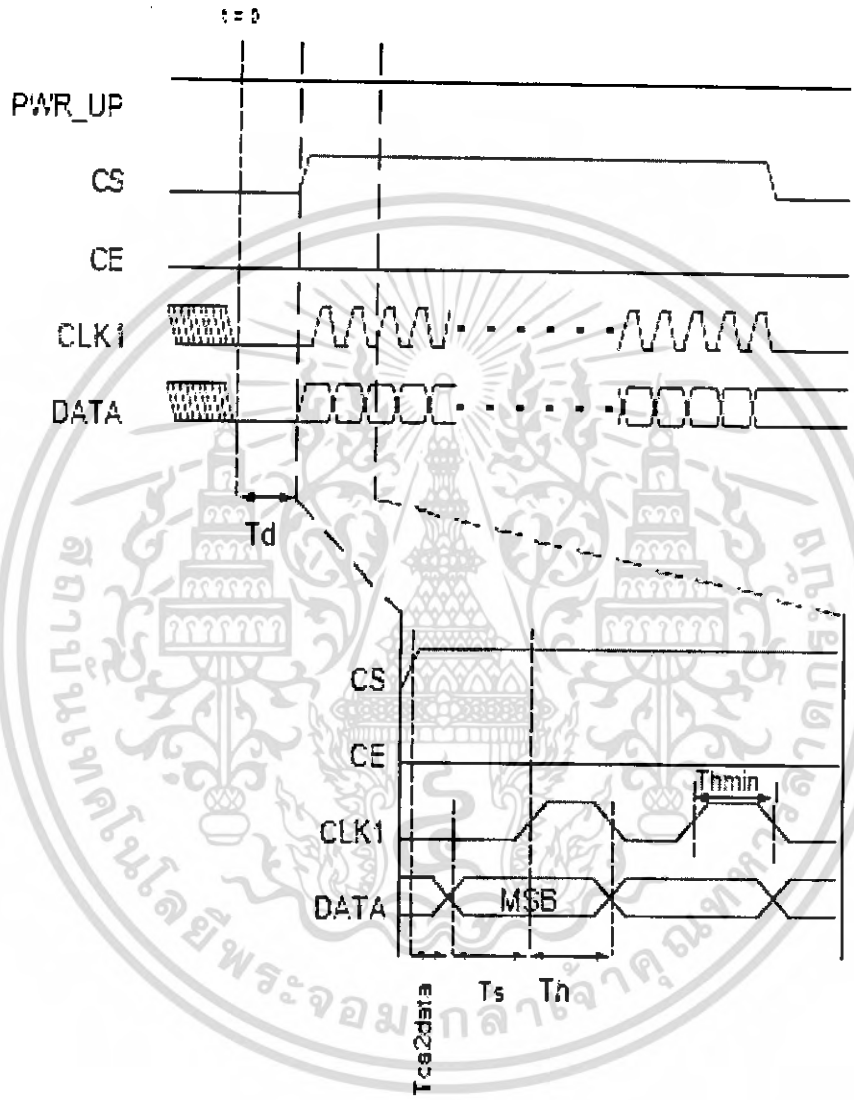
1. PREMBLE	- ส่วน preamble ต้องมีใน Shock Burst mode และ Direct mode - Preamble มีความยาว 8 บิตและ ขึ้นกับบิตแรกของแอดเดรส PREAMBLE 1st ADDR_BIT 01010101 0 10101010 1 - Preamble จะใส่ไปในชุดข้อมูลโดยอัตโนมัติและดังนั้นจึงมีที่ว่างพิเศษ สำหรับ payload ใน Direct mode MCU ต้องควบคุม preamble ใน ShockBurst mode Rx preamble จะถูกย้ายจากรับข้อมูลเอาท์พุท ใน Direct mode preamble จะเห็นได้ชัดเจนสำหรับข้อมูลเอาท์พุท
2. ADDRESS	- ส่วนของแอดเดรสต้องมีใน ShockBurst โหมด และ Direct โหมด - ความยาว 80 – 40 บิต - แอดเดรสจะถูกย้ายโดยอัตโนมัติจากชุดรับใน ShockBurst mode ใน Direct mode MCU ต้องควบคุมแอดเดรส
3. PAYLOAD	- ข้อมูลถูกส่ง - ใน ShockBurst โหมด payload มีขนาดน้อยที่สุด 256 บิต (แอดเดรส 8-40 บิต +CRC 8หรือ16 บิต - ใน Direct mode ความยาวมากสุดสำหรับ 1 Mbps คือ 4000 บิต
4. CRC	- CRC จะเลือกได้ใน ShockBurst โหมด และไม่ใช้ใน Direct โหมด - ความยาว 8 หรือ 16 บิต - ShockBurst Rx CRC จะถูกย้ายจากรับข้อมูลเอาท์พุท

ตารางที่ 2.15 รายละเอียดของ Data package

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.8 Configuration mode timing

เมื่อ 1 บิต หรือ มากกว่า 1 บิต ใน configuration word จำเป็นที่จะต้องเปลี่ยนแปลง เราจะไปดูได้จาก timing

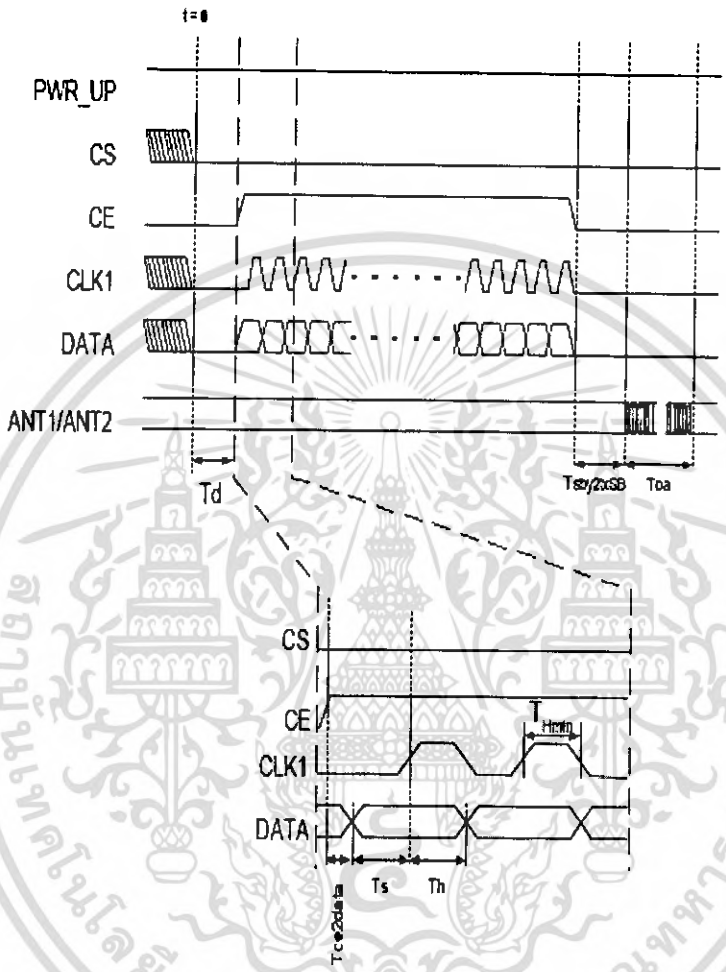


รูปที่ 2.24 Timing Diagram สำหรับ configuration ระบบย่อย nRF2401ถ้า configuration mode เข้ามาจาก power down CS สามารถ set high หลังจาก Tpd2sby

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.4.9 ShockBurst Mode Timing

ShockBurst TX:



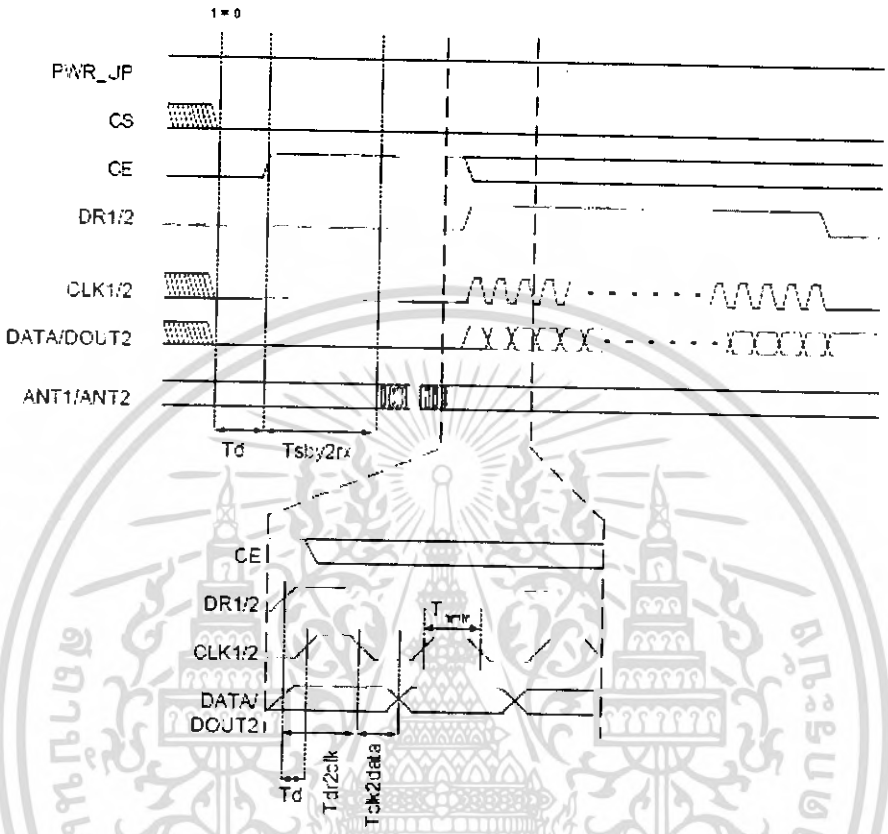
รูปที่ 2.25 Timing ของ ShockBurst ใน TX

ความยาวชุดข้อมูลและอัตราข้อมูลทำให้ delay Toa (Time on air : เวลาในอากาศ) ซึ่งแสดงในสมการ บิตข้อมูลจะเป็นผลรวมของบิตที่รวมทุกๆบิต CRC และบิต preamble ซึ่งอาจเพิ่มขึ้น

Toa = 1/data rate*(#databits+1)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ShockBurst Rx:



รูปที่ 2.26 Timing ของ ShockBurst ใน Rx

CE อาจยังคงสถานะ high ช่วงที่ดาวน์โหลดข้อมูล แต่จะทำให้ค่าการใช้กระแสสูงกว่า (19mA) และผลประโยชน์คือ เวลา start-up น้อย (200us) เมื่อ DR1 ขาลง

2.5 การเขียนโปรแกรมติดต่อ Serial Port

พื้นฐานการสื่อสารแบบอนุกรมในเครื่องคอมพิวเตอร์นั้นจะมีความเร็วในการสื่อสารช้ากว่าแบบขนาน ทั้งนี้ก็เพราะว่าการเคลื่อนย้ายข้อมูลแบบอนุกรมเป็นการส่งข้อมูลครั้งละ 1 บิต แต่พอร์ตขนานนั้นสามารถส่งข้อมูลได้ครั้งละหลาย ๆ บิตพร้อมกัน ส่งผลให้การสื่อสารข้อมูลแบบอนุกรมมีความเร็วต่ำกว่าแบบขนาน

แต่ว่าการส่งข้อมูลแบบอนุกรมนั้นมีข้อที่เหนือกว่าการส่งข้อมูลแบบขนาคือ การสามารถส่งข้อมูลได้ในระยะทางที่ไกลกว่าแบบขนาน อีกทั้งสายสัญญาณที่ใช้ยังมีน้อยกว่าการส่งข้อมูลแบบขนานอีกด้วย การสื่อสารแบบอนุกรมสามารถแบ่งออกได้เป็น 3 รูปแบบดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1. Simplex สามารถส่งข้อมูลได้อย่างเดียว เป็นการสื่อสารแบบทางเดียว
2. Half-Duplex สามารถส่งข้อมูลไปยังปลายทางและสามารถรับข้อมูลจากปลายทางได้ แต่ไม่ สามารถทำการส่งและรับข้อมูลได้ในเวลาเดียวกัน
3. Full-Duplex สามารถรับและส่งข้อมูลได้ในเวลาเดียวกัน

2.5.1 มาตราฐาน RS-232C

มาตราฐาน RS-232C เป็นมาตราฐานที่ได้รับการออกแบบมาเพื่อที่จะทำให้อุปกณ์ต่อพ่วงจากผู้ผลิตต่างกันสามารถทำงานร่วมกันได้ มาตราฐาน RS-232C ประกาศโดยสมาคมอุตสาหกรรมอิเล็กทรอนิกส์ (Electronic Industries Association : EIA)

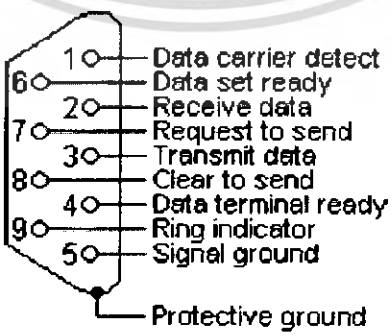
มาตราฐาน RS-232C ได้แบ่งอุปกรณ์ออกเป็น 2 ประเภทคือ

1. อุปกรณ์ DTE (Data Terminal Equipment) เป็นอุปกรณ์สำหรับส่งข้อมูล (เอาต์พุท)
2. อุปกรณ์ DCE (Data Communication Equipment) เป็นอุปกรณ์สำหรับรับข้อมูล (อินพุท)

ตามมาตราฐาน RS-232C แล้วคอนเน็กเตอร์ของ DTE จะเป็นตัวผู้ ส่วนคอนเน็กเตอร์ของ DCE จะเป็นตัวเมีย ที่นิยมใช้กันอยู่จะเป็นชนิด D-Type แบบ 9 ขา และแบบ 25 ขา โดยจะติดตั้งอยู่ด้านหลังเครื่องคอมพิวเตอร์ ระดับแรงดันจะมีค่าระหว่าง -3 ถึง -15 V สำหรับลอจิก High และ Low จะมีระดับแรงดันระหว่าง +3 ถึง +15 V สามารถรับส่งข้อมูลได้ที่ความยาวของสายสัญญาณสูงสุด 50 ฟุต หรือ 150 เมตร แต่ถ้าเราต้องการการสื่อสารกับอุปกรณ์อื่นที่อยู่ห่างกันมากๆ เราจำเป็นต้องใช้อุปกรณ์อื่น ๆ ช่วยเช่น การใช้โมเด็ม เป็นต้น

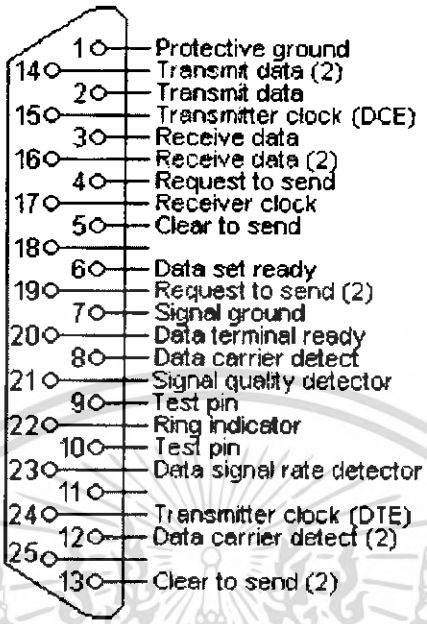
2.5.2 ลักษณะของคอนเน็กเตอร์แบบ D-Type

หัวต่อแบบ D-Type ที่ใช้ในการสื่อสารแบบอนุกรมของเครื่องคอมพิวเตอร์นั้นมีอยู่ 2 ลักษณะคือ แบบ 9 ขาและ 25 ขา บางครั้งเรียกว่า DB9 และ DB25 ซึ่งหัวต่อทั้งสองชนิดจะมีลักษณะการทำงานของสัญญาณเหมือนกัน แต่การจัดเรียงไม่เหมือนกัน



รูปที่ 2.27 DB9

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.28 DB25

D-Type 25 Pin	D-Type 9 Pin	สัญลักษณ์	ชื่อสัญลักษณ์
Pin 3	Pin 3	TD	Transmit Data
Pin 3	Pin 2	RD	Receive Data
Pin 4	Pin 7	RTS	Request To Send
Pin 5	Pin 8	CTS	Clear To Send
Pin 6	Pin 6	DSR	Data Set Ready
Pin 7	Pin 5	SG	Signal Ground
Pin 8	Pin 1	CD	Carrier Detect
Pin 20	Pin 4	DTR	Data Terminal Ready
Pin 22	Pin 9	RI	Ring Indicator

ตารางที่ 2.16 แสดงสัญลักษณ์และชื่อสัญญาณของ Connector แบบ C-Type

รายละเอียดของสายสัญญาณ

Transmit Data: TD ใช้สำหรับส่งข้อมูลอนุกรมออกจากคอมพิวเตอร์

Receive Data: RD ใช้สำหรับรับข้อมูลอนุกรมเข้ามายังคอมพิวเตอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Request To Send: RTS	ใช้สำหรับส่งข้อมูลไปยังอุปกรณ์ปลายทาง เพื่อร้องขอให้ปลายทางส่งข้อมูลกลับมา
Clear To Send: CTS	ใช้สำหรับตรวจสอบว่าอุปกรณ์ที่เชื่อมต่อด้วยพร้อมจะรับข้อมูลหรือไม่ โดยจะคอยรับสัญญาณ RTS เมื่อทุกอย่างพร้อมก็จะทำการส่งข้อมูลออกทางขา TD
Data Set Ready: DSR	ใช้สำหรับตรวจสอบการเชื่อมต่อกันระหว่างคอมพิวเตอร์กับอุปกรณ์ปลายทาง จะใช้คู่กับขา DTR
Signal Ground: SG	เป็นกราวด์ของระบบ
Carrier Detect: CD	ขานี้จะ Active เมื่อมีการส่งสัญญาณ Carrier จากโมเด็ม
Data Terminal Ready: DTR	ใช้สำหรับบอกให้อุปกรณ์ปลายทางรับรู้ว่าต้องการติดต่อด้วยโดยขา DTR นี้ต้องเชื่อมต่อกับขา DSR ของอุปกรณ์ปลายทาง
Ring Indicator: RI	ขานี้จะ Active เมื่อโมเด็มได้รับสัญญาณเรียกเข้าจากสายโทรศัพท์

2.5.3 องค์ประกอบของการรับส่งข้อมูลแบบอนุกรม

การสื่อสารแบบอนุกรมที่นิยมใช้กับคอมพิวเตอร์นั้น เป็นการสื่อสรข้อมูลแบบอะซิงโครนัสนั่นคือ ต้องใช้สายสัญญาณเส้นเดียวทำหน้าที่ทั้งส่งส่วนที่เป็นข้อมูล และส่วนที่ใช้ควบคุมการส่งข้อมูล ดังนั้นข้อมูลที่อ่านได้แต่ละบิตจากการส่งแบบอนุกรม จึงต้องถูกแยกแยะว่าใช้สำหรับวัตถุประสงค์ใด โดยเราสามารถแบ่งได้ 4 ส่วนคือ

1. Start Bit หรือบิตเริ่มต้น จะมีขนาด 1 บิต จะใส่ที่จุดเริ่มต้นเพื่อบอกอุปกรณ์ฝ่ายรับว่าข้อมูลกำลังจะมาถึง
2. Data Character หรือบิตข้อมูล ขนาด 7 หรือ 8 บิต การส่งบิตข้อมูลจะส่งเป็นกลุ่ม ๆ
3. Parity Bit หรือบิตพาริตี้ ขนาด 1 บิต ใช้ในการตรวจสอบความถูกต้องของข้อมูลที่ส่ง เราจะใส่บิตพาริตี้เข้าไป แต่ทั้งตัวรับและตัวส่งจะต้องรู้ตรงกันว่าใช้พาริตี้แบบไหนในการส่งข้อมูล ซึ่งหลักการในการกำหนดบิตพาริตี้มีหลายแบบดังนี้
   - พาริตี้คู่ (Even Parity) ค่าของพาริตี้นี้เมื่อรวมกับทุก ๆ บิตของข้อมูลแล้วจะต้องมีจำนวนบิตที่เป็นเลข1 เป็นเลขคู่
   - พาริตี้คี่ (Odd Parity) ค่าของพาริตี้นี้เมื่อรวมกับทุก ๆ บิตของข้อมูลแล้วจะต้องมีจำนวนบิตที่เป็นเลข 1 เป็นเลขคี่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- ไม่มีพาริตี้ (None) ถ้าตั้งพารีตี้เป็น None ทั้งภาครับและภาคส่งจะไม่มีการตรวจสอบบิตพาริตี้

4. Stop Bit หรือบิตจบ ขนาด 1 หรือ 2 บิต เป็นบิตปิดท้ายข้อมูล

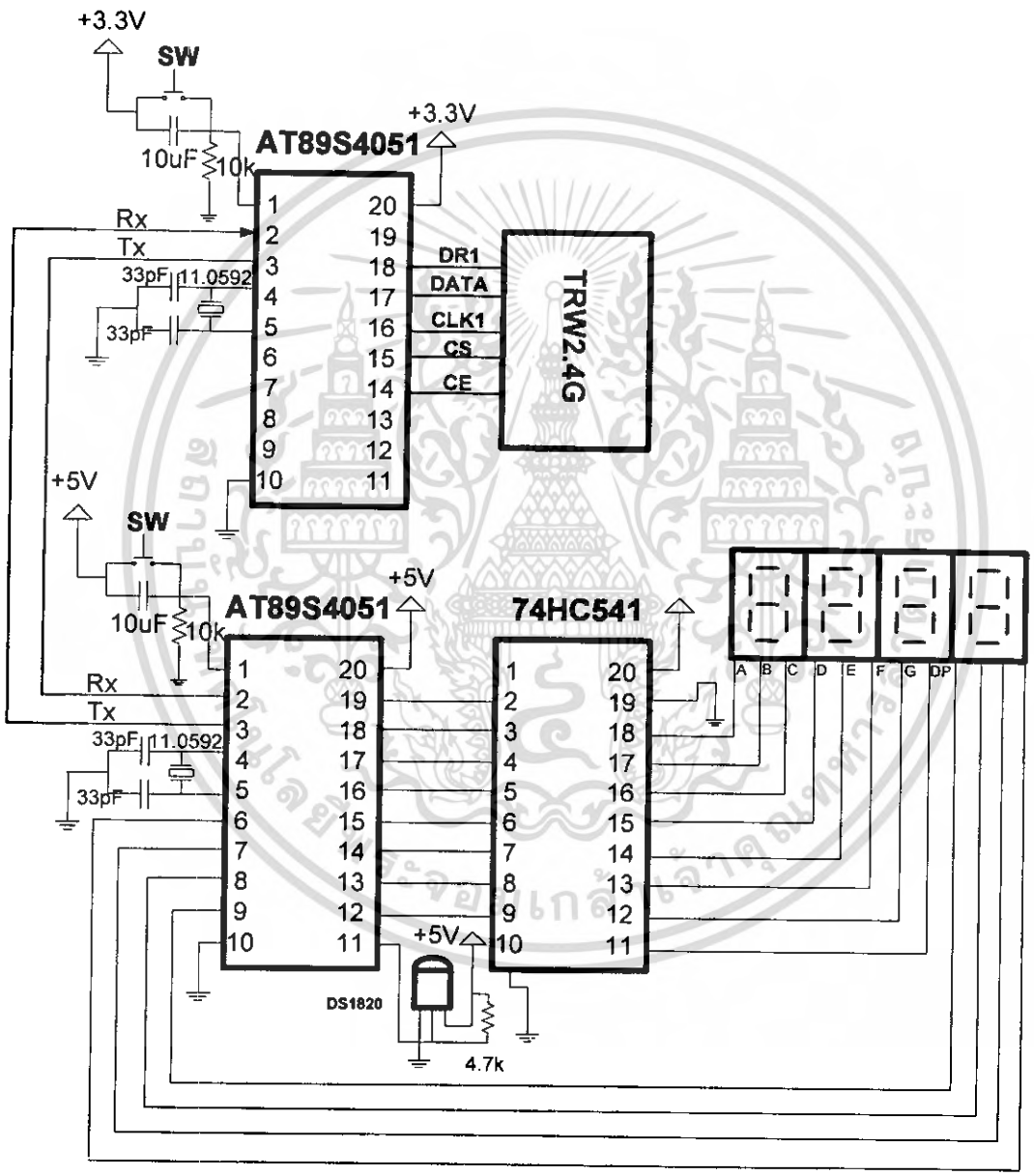
2.5.4 อัตราเร็วในการรับส่งข้อมูลแบบอนุกรม

การที่อุปกรณ์ 2 อย่างจะติดต่อสื่อสารกันได้นั้น จะต้องทำงานด้วยอัตราเร็วเท่ากัน ซึ่งอัตราเร็วในการสื่อสารแบบอะซิงโครนัสคือ ค่าบอดเรต (Baud Rate) มีหน่วยเป็นบิตต่อวินาที ซึ่งอัตราเร็วในการสื่อสารแบบอนุกรมนั้นมีใช้ดังนี้ 110, 150, 300, 600, 1200, 2400, 4800, 9600 และ 19200 บิตต่อวินาที

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3
การออกแบบและการสร้างเครื่องรับ – เครื่องส่ง RF

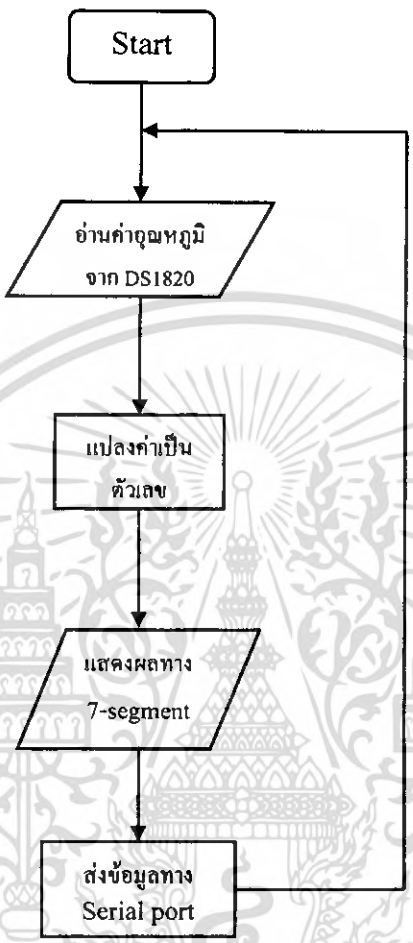
3.1 การออกแบบและการสร้างเครื่องรับ-เครื่องส่ง



รูปที่ 3.1 วงจรตัวส่ง SENSOR DS 1820

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.1 การทำงานของวงจร Thermal Sensor

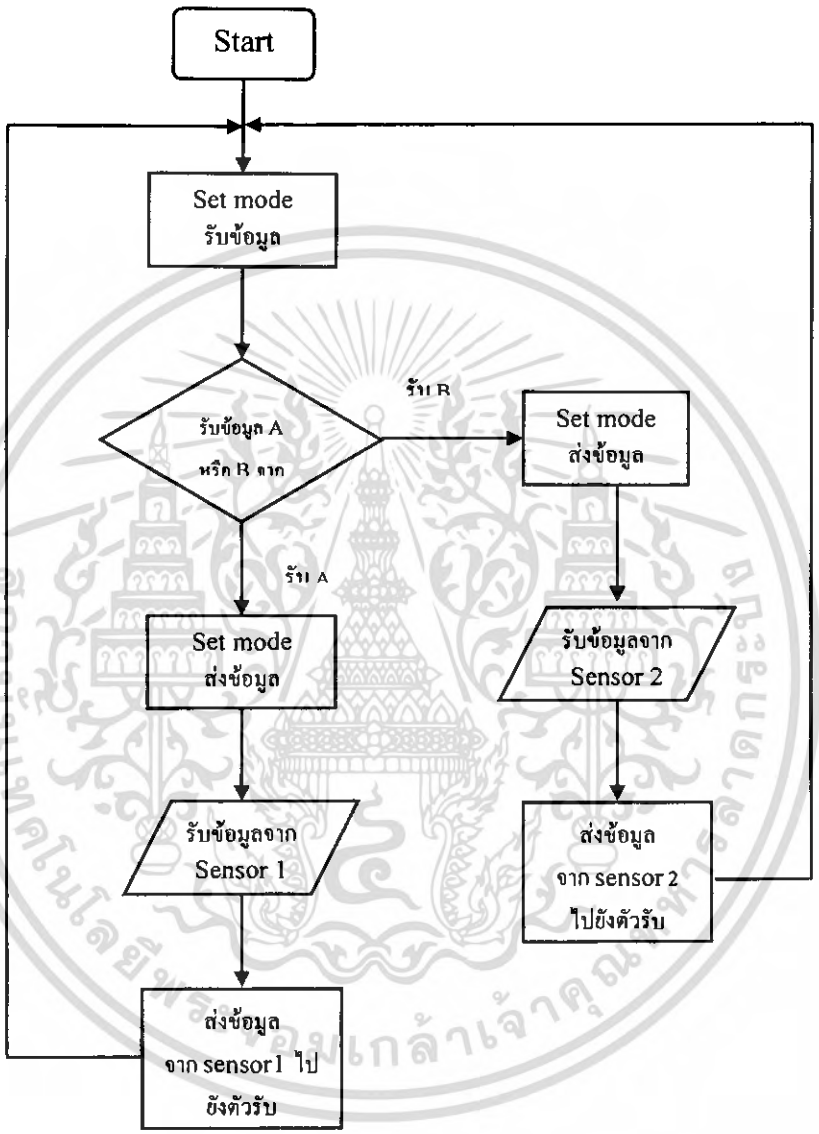


รูปที่ 3.2 Flow Chart แสดงขั้นตอนการทำงานของ Thermal Sensor

เริ่มจากทำการติดต่อและอ่านค่าอุณหภูมิจาก Thermal Sensor DS 1820 โดยใช้ไมโครคอนโทรลเลอร์เป็นตัวติดต่อกับเซนเซอร์ เมื่อทำการเชื่อมต่อกันได้แล้วจะนำค่าที่อุณหภูมิที่ DS 1820 อ่านได้มาแปลงค่าให้กลายเป็นตัวเลขอุณหภูมิที่ถูกต้องตามที่กำหนดไว้ จากนั้นนำค่าอุณหภูมิที่ถูกต้องแล้วนั้นมาแสดงผลผ่านทาง 7-segment แล้วจึงส่งข้อมูลออกแแบบอนุกรมไปยังตัวส่ง TRW 2.4 GHz

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.1.2 การทำงานของ Transmitter



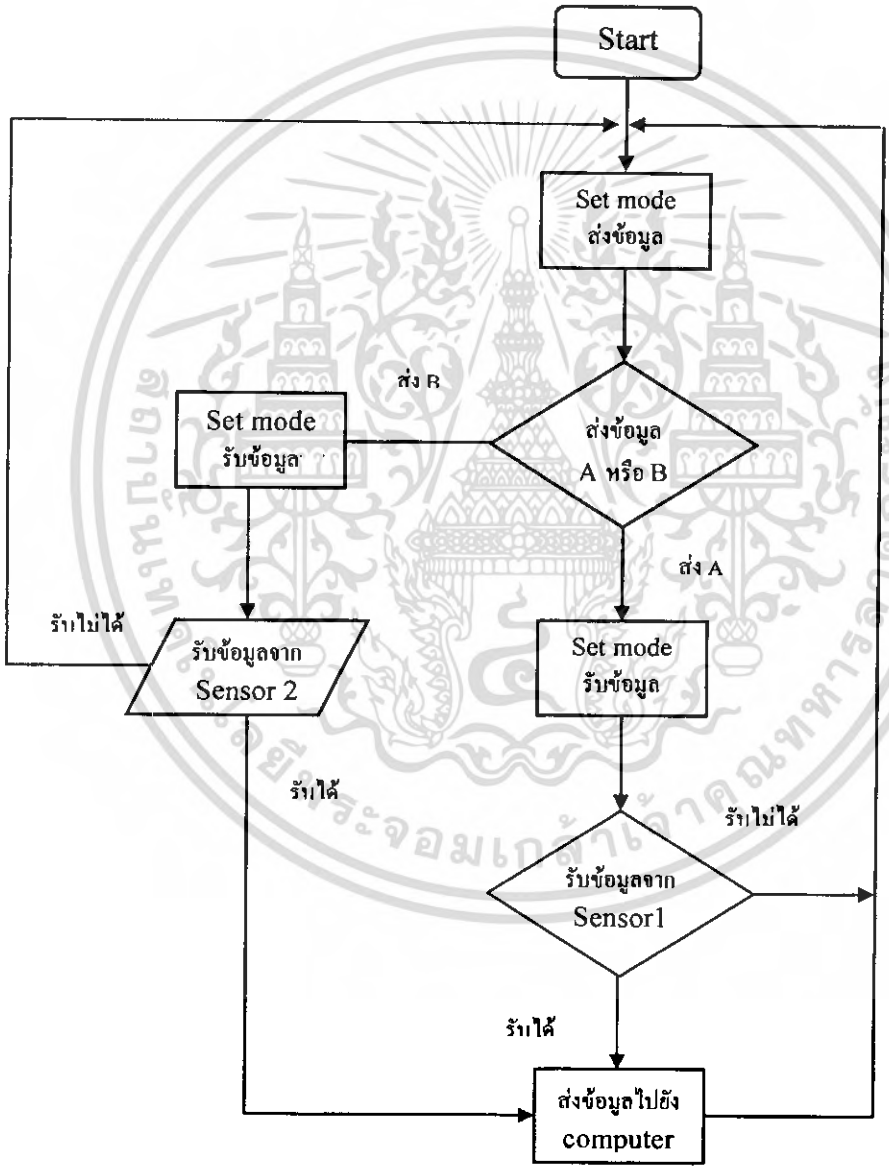
รูปที่ 3.3 Flow Chart แสดงขั้นตอนการทำงานของ Transmitter

เมื่อเริ่มต้นการทำงาน ตัวส่งข้อมูลทั้ง 2 ตัว จะต้องทำการตั้งค่าสถานะให้กลายเป็นตัวรับข้อมูลเสียก่อน เพื่อรอรับสัญญาณอนุญาต (A หรือ B) จากตัวรับที่ส่งสัญญาณมาให้ จากนั้นตัวส่งข้อมูลทั้ง 2 จะต้องทำการรับสัญญาณนั้น ๆ (A หรือ B) มาตรวจสอบว่าตรงกับตัวส่งนั้นหรือไม่ (โดยในที่นี้ ตัวส่งตัวที่ 1 สามารถส่งข้อมูลได้ก็ต่อเมื่อ ได้รับสัญญาณ A และตัวส่งที่ 2 จะสามารถ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่งข้อมูลได้ก็ต่อเมื่อได้รับสัญญาณ B) เมื่อทำการตรวจสอบที่ได้รับแล้ว จากนั้นเมื่อสัญญาณที่ได้รับตรงตามที่กำหนดแล้ว จึงทำการเปลี่ยนสถานะจากสถานะรับ กลายเป็นสถานะส่ง เพื่อทำการส่งข้อมูลจากตัว sensor นั้น ๆ กลับไปยังตัวรับอีกครั้ง และแปลงกลับไปเป็นสถานะรับดังเดิม เพื่อรอรับสัญญาณอนุญาตต่อไป

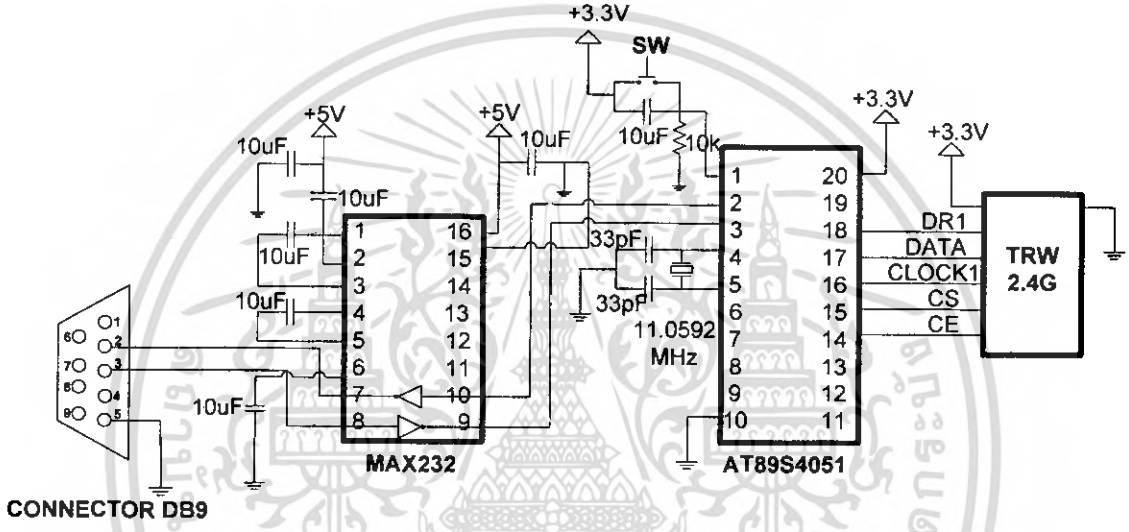
3.1.3 การทำงานของ Receiver



รูปที่ 3.4 Flow Chart แสดงขั้นตอนการทำงานของ Receiver

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของตัวรับข้อมูลเริ่มจากการตั้งค่าสถานะให้เป็นตัวส่งข้อมูลเพื่อทำการส่งสัญญาณอนุญาต (A หรือ B) ไปยังตัวส่ง เพื่อให้ตัวส่งสามารถตรวจสอบได้ว่า ตัวใดควรจะทำหน้าที่ส่งข้อมูลกลับเข้ามายังตัวรับ ซึ่งเมื่อทำการส่งสัญญาณออกไปแล้ว จะทำการเปลี่ยนสถานะเพื่อทำการรอรับค่าข้อมูลที่จะถูกส่งมาจากตัวส่งข้อมูลที่ทำการเลือกไว้แล้ว หลังจากรับข้อมูลได้แล้วจะนำข้อมูลนั้นส่งไปยังคอมพิวเตอร์ แล้วจึงเปลี่ยนสถานะเป็นตัวส่งอีกครั้ง เพื่อส่งสัญญาณให้ตัวส่งตัวต่อ ๆ ไปส่งข้อมูลเข้ามา

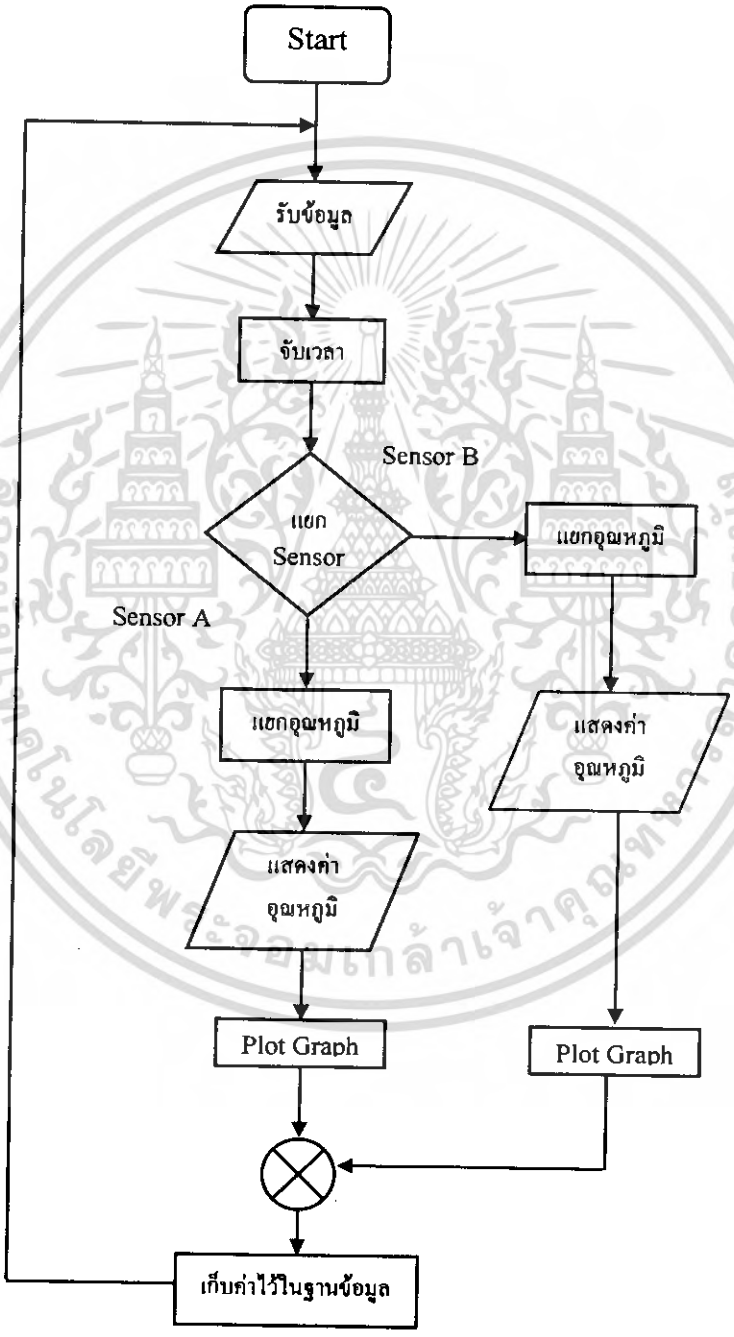


รูปที่ 3.5 วงจรตัวรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ภาคแสดงผล

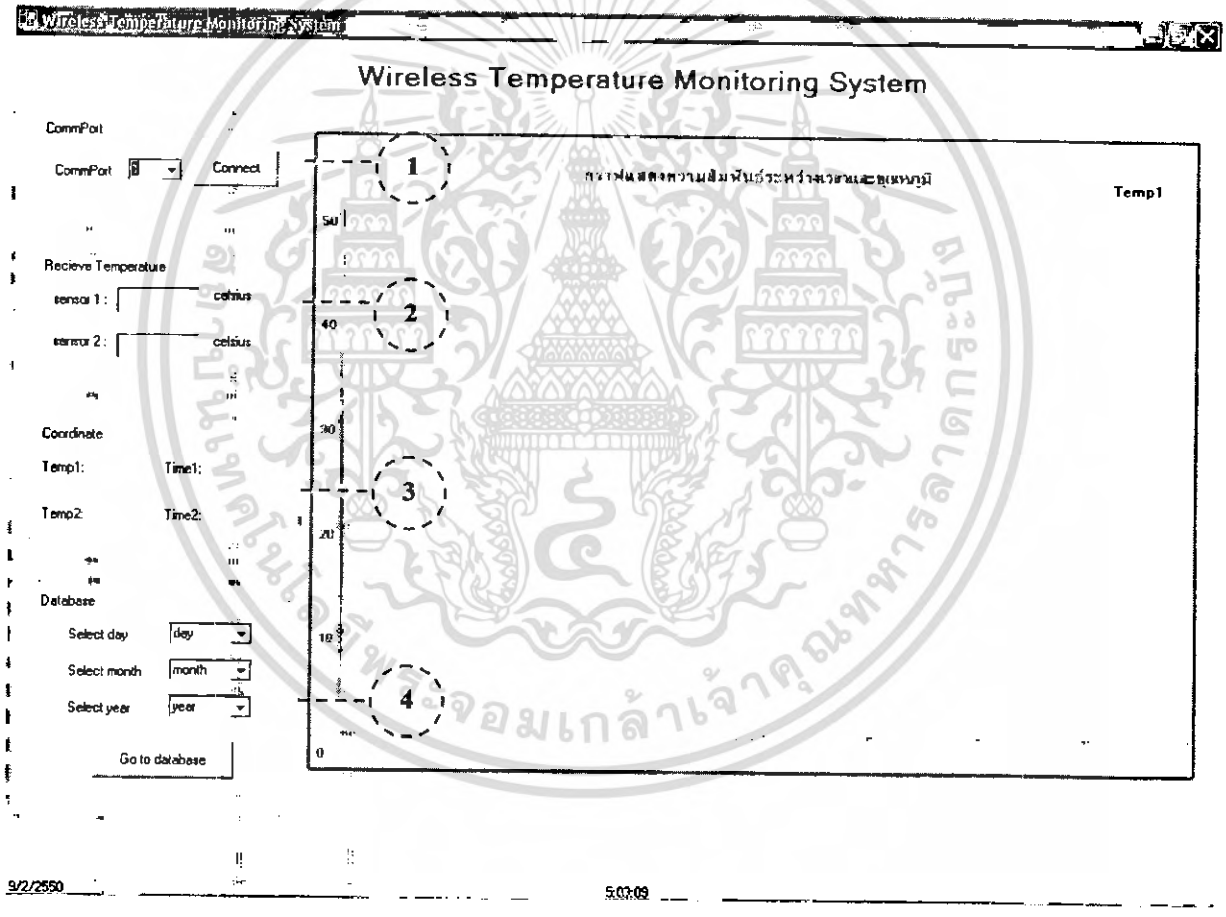
ส่วนของภาคแสดงผลนี้ใช้การแสดงผลผ่านคอมพิวเตอร์ โดยใช้โปรแกรม Visual Basic.NET โดยจะติดต่อผ่านพอร์ตอนุกรม (RS232) แล้วนำค่าข้อมูลที่ส่งเข้ามาแสดงผลออกมาเป็นกราฟระหว่างค่าอุณหภูมิกับเวลาของ sensor แต่ละตัว



รูปที่ 3.6 Flow Chart แสดงการทำงานของภาคแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การทำงานของภาคแสดงผลเริ่มจากเลือก ComPort ที่จะใช้รับข้อมูลจากพอร์ตอนุกรม(1) เมื่อเชื่อมต่อแล้วจึงเริ่มรับข้อมูล จากนั้นเริ่มจับเวลา นำค่าข้อมูลที่ได้มานี้มาแยกว่าเป็นข้อมูลที่ส่งมาจาก Sensor ตัวใด โดย ถ้าข้อมูลเริ่มต้นด้วยตัวอักษร "A" จะเป็นข้อมูลที่ส่งมาจาก Sensor A หากข้อมูลเริ่มต้นด้วยตัวอักษร "B" จะเป็นข้อมูลที่ส่งมาจาก Sensor B เมื่อแยก Sensor ได้แล้ว จะทำการแยกข้อมูลค่าอุณหภูมิออกมาแล้วนำมาแสดงที่ Receive Temperature (2) จากนั้นเมื่อครบตามเวลาที่เราจับไว้ จะนำค่า Coordinate ของอุณหภูมิกับเวลามาแสดง(3) แล้วนำค่า Coordinate นั้น ๆ มาวาดกราฟ จากนั้นเก็บค่าอุณหภูมิกับเวลาเข้าสู่ฐานข้อมูล โดยสามารถเรียกดูข้อมูลในฐานข้อมูลได้โดยเลือก วันที่ เวลา และปี ใน Database (4)



รูปที่ 3.7 ฟอร์มของโปรแกรมแสดงผล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

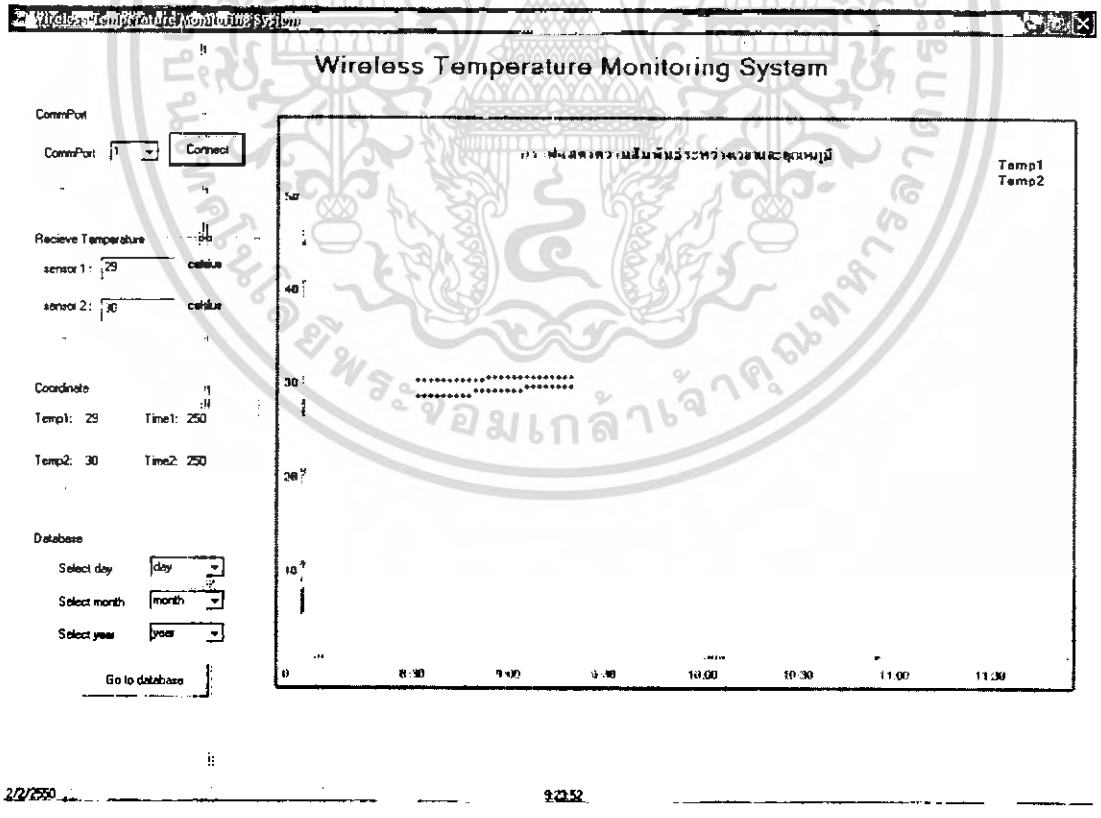
บทที่ 4

การทดลองและผลการทดลอง

ทดลองวัดค่าอุณหภูมิเปรียบเทียบกัน โดย เปรียบเทียบจากการวัดด้วย Thermometer , Multimeter และ ค่าอุณหภูมิที่วัดได้จาก Thermal Sensor DS 1820 ได้ผลการทดลองดังนี้

Thermometer (Celsius)	Multimeter (Celsius)	DS 1820 (Celsius)
23	24	25
24	24	25.5
25	26	27
26	26	28
27	28	29.5

ตารางที่ 4.1 ผลการทดลองวัดค่าอุณหภูมิโดยใช้ Thermometer, Multimeter และ DS 1820



รูปที่ 4.1 แสดงผลโดยการ Plot Graph

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5
สรุปผลการทดลอง

จากการทดลองจะเห็นว่าค่าอุณหภูมิที่เราวัดได้จาก Thermometer, Multimeter และ Thermal Sensor DS 1820 นั้นมีค่าใกล้เคียงกัน แตกต่างกันเล็กน้อยซึ่งค่าอุณหภูมที่ DS 1820 อ่านได้นั้นมีค่ามากกว่าที่อ่านได้จาก Thermometer และ Multimeter ซึ่งความร้อนส่วนที่เกินมานี้อาจเกิดจากอุณหภูมิของวงจรซึ่งอาจส่งผลให้ค่าอุณหภูมิที่อ่านได้อาจจะมีค่าสูงขึ้น ระยะในการส่งสัญญาณนั้นจากการวัดโดยประมาณอยู่ที่ 40 เมตรซึ่งเป็นระยะที่เครื่องรับ-เครื่องส่งยังทำงานได้ดี หากที่ระยะไกลเกินจากนี้จะรับสัญญาณไม่ได้ และส่งสัญญาณโดยใช้ช่องสัญญาณเพียง 1 ช่องสัญญาณโดยเป็นการติดต่อสื่อสารแบบ Half-Duplex

ส่วนของภาคแสดงผลสามารถรับค่าจากพอร์ตอนุกรมของคอมพิวเตอร์ได้โดยการที่เลือก comport ให้ถูกต้องและสามารถ plot กราฟของค่าอุณหภูมิและเวลาที่รับค่าอุณหภูมิเหล่านั้นได้ ส่วนฐานข้อมูลจะบันทึกค่าอุณหภูมิที่แสดง ค่าเวลาที่รับค่ามา และบันทึกวันที่ที่มีการรับค่าอุณหภูมิ โดยสามารถเรียกดูฐานข้อมูลตามวันที่ที่รับค่าอุณหภูมิ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารอ้างอิง

1. อภิชาติ ภู่พลับ "เริ่มต้นการเขียนโปรแกรมติดต่อและควบคุมฮาร์แวร์ ด้วย VisuaBasic",Infopress Developer Book,2546
2. ธาริน สิทธิธรรมชาตรี " คู่มือการเขียนโปรแกรม Microsoft Visual Basic.NET",บริษัท ซัคเซส มีเดียจำกัด
3. วรพจน์ กรแก้ววัฒนกุล, ชัยวัฒน์ ลิ้มพรจิตรวิไล, "เรียนรู้และปฏิบัติการไมโครคอนโทรเลอร์ MCS-51 แบบแฟลช",อินโนเวทิฟ เอ็กเพอริเมนต์ จำกัด
4.. ผศ. ธีรวัฒน์ ประกอบผล, "การประยุกต์ใช้งานไมโครคอนโทรลเลอร์",สำนักพิมพ์ส.ส.ท.,2544
5. รศ. สมยศ อุณณะปียะ, " การประยุกต์ใช้งานไมโครคอนโทรลเลอร์ตระกูล MCS-51 ", ภาควิชาวิศวกรรมโทรคมนาคม สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาคผนวก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DS 1820

```
DPC             EQU   P3.2
DP              EQU   P3.3
DG1             EQU   P3.4
DG10            EQU   P3.5
ONEWIRE         EQU   P3.7

DG1_BUF         EQU   030H
DG10_BUF        EQU   031H
ONEWIRE_DATA    EQU   032H
ONEWIRE_CNT     EQU   033H
TEMP            EQU   034H

                ORG   0000H
                JMP   INITIAL

INITIAL:        SETB  P3.2
                SETB  P3.3
                SETB  P3.4
                SETB  P3.5

                MOV       SCON,#50H
                MOV       TMOD,#21H
                MOV       TH1,#0FDH
                MOV       TL1,#0FDH
                SETB  TR1
MAIN:           ACALLDS1820_TEMP_RD
                MOV       R1,#250
                JMP       MAIN
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DISPLAY:        DEC     R1
                MOV     R2,#100

DISPLAY_1:      DEC     R2
                MOV     A,TEMP
                MOV     B,#2
                DIV     AB
                MOV     R0,B
                CJNE    R0,#0,SHOW_DP
                JMP     SHOW_0

SHOW_DP:        ACALL LOOP
                MOV     P1,#6DH
                CLR             DP
                SETB    DP
                JMP     DG_LOOP

SHOW_0:         ACALL LOOP
                MOV     P1,#3FH
                CLR     DP
                SETB    DP
                JMP     DG_LOOP

DG_LOOP:        MOV     A,DG1_BUF
                MOV     DPTR,#SEG
                MOVC    A,@A+DPTR
                ORL     A,#10000000B
                MOV     P1,A
                CLR     DG1
                SETB    DG1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    MOV   A,DG10_BUF
                    MOV   DPTR,#SEG
                    MOVC  A,@A+DPTR
                    MOV   P1,A
                    CLR   DG10
                    SETB  DG10

                    MOV   P1,#00111001B
                    CLR   DPC
                    SETB  DPC

                    CJNE  R2,#0,DISPLAY_1
                    CJNE  R1,#0,DISPLAY
                    RET

LOOP:               MOV   B,#10
                    DIV   AB
                    MOV   DG1_BUF,B
                    MOV   DG10_BUF,A
                    RET
;-------------------------------------------------------------
;DS1820        TEMP READ
;-------------------------------------------------------------
DS1820_TEMP_RD:     ACALLDS1820_RST
                    ACALLDS1820_PRES

                    MOV   ONEWIRE_DATA,#0CCH
                    ACALLDS1820_WR
                    MOV   ONEWIRE_DATA,#044H
                    ACALLDS1820_WR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    SETB   ONEWIRE
                    ACALLDISPLAY

                    ACALLDS1820_RST
                    ACALLDS1820_PRES

                    MOV    ONEWIRE_DATA,#0CCH
                    ACALLDS1820_WR
                    MOV    ONEWIRE_DATA,#0BEH
                    ACALLDS1820_WR

                    ACALLDS1820_RD
                    TEMP,ONEWIRE_DATA

                    MOV    SBUF,TEMP
                    JNB    TI,$
                    CLR    TI

DS1820_TEMP_CLR:    ACALLDS1820_RST
                    ACALLDS1820_PRES
                    RET
;-------------------------------------------------------------
;DS1820 DATA READ
;-------------------------------------------------------------
DS1820_RD:          MOV    ONEWIRE_CNT,#8
                    CLR    A

DS1820_RD_LOOP:     CLR    ONEWIRE
                    NOP
                    SETB   ONEWIRE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                NOP
                NOP
                MOV   C,ONEWIRE
                ACALLONEWIRE_DELAY
                RRC   A
                SETB  ONEWIRE
                DJNZ  ONEWIRE_CNT,DS1820_RD_LOOP
                MOV   ONEWIRE_DATA,A
                RET
;---------------------------------------------------------
;DS1820 DATA WRITE
;---------------------------------------------------------
DS1820_WR:      MOV   ONEWIRE_CNT,#8
                MOV   A,ONEWIRE_DATA

DS1820_WR_LOOP: RRC   A
                JNC   DS1820_WR_L
                CLR   ONEWIRE
                NOP
                NOP
                SETB  ONEWIRE
                ACALLONEWIRE_DELAY
                JMP   DS1820_WR_NX

DS1820_WR_L:    CLR   ONEWIRE
                ACALLONEWIRE_DELAY
                SETB  ONEWIRE
                NOP
                NOP
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
DS1820_WR_NX:       DJNZ   ONEWIRE_CNT,DS1820_WR_LOOP
                    RET
;-------------------------------------------------------------
;DS1820 RESET
;-------------------------------------------------------------
DS1820_RST:         CLR    ONEWIRE
                    ACALL  DELAY_750US
                    SETB   ONEWIRE
                    MOV    ONEWIRE_CNT,#22
                    DJNZ   ONEWIRE_CNT,$
                    RET
;-------------------------------------------------------------
;DS1820 RECIEVE PRESENCE PULSE
;-------------------------------------------------------------
DS1820_PRES:    MOV    ONEWIRE_CNT,#110

DS1820_PRES_1:      JNB    ONEWIRE,DS1820_PRES_2
                    DJNZ   ONEWIRE_CNT,DS1820_PRES_1
                    RET

DS1820_PRES_2:      MOV    ONEWIRE_CNT,#110

DS1820_PRES_3:      JB     ONEWIRE_CNT,DS1820_PRES_4
                    DJNZ   ONEWIRE_CNT,DS1820_PRES_3

DS1820_PRES_4:      SETB   ONEWIRE
                    ACALL  ONEWIRE_DELAY
                    RET
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
;-------------------------------------------------------------
;DELAY TIME
;-------------------------------------------------------------
ONEWIRE_DELAY:      MOV   R4,#012H
ONEWIRE_DELAY_1:    NOP
                    NOP
                    DJNZ  R4,ONEWIRE_DELAY_1
                    RET
DELAY_1S:           MOV   R4,#100
DELAY_1S_1:         ACALL DELAY_10MS
                    DJNZ  R4,DELAY_1S_1
                    RET
DELAY_10MS:         MOV   R5,#10
DELAY_10MS_1:       MOV   R6,#0E6H
DELAY_10MS_2:       NOP
                    NOP
                    DJNZ  R6,DELAY_10MS_2
                    DJNZ  R5,DELAY_10MS_1
                    RET
DELAY_750US:        MOV   R4,#250
DELAY_750US_1:      NOP
                    DJNZ  R4,DELAY_750US_1
                    RET
;-------------------------------------------------------------
;SEGMENT
;-------------------------------------------------------------
SEG:                DB    3FH
                    DB    06H
                    DB    5BH
                    DB    4FH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        DB      66H
        DB      6DH
        DB      7DH
        DB      07H
        DB      7FH
        DB      6FH
END
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SLAVE 1

```
CE              BIT P1.2
CS              BIT P1.3
DAT             BIT P1.5
CLK             BIT P1.4
DR1             BIT P1.6
                ORG 0000H
;*****************************************************************
INIT:           MOV   TMOD,#021H
                MOV   TH1,#0FDH
                MOV   TL1,#0FDH
                MOV   SCON,#050H
                SETB  TR1
;*****************************************************************
                CLR   CE
                CLR   CS
                CLR   DAT
                CLR   CLK
;*****************************************************************
MAIN:           CALL  SETMODE_RX
                JNB   DR1,$
                CALL  READ_TRW24

CHECK:          CJNE  A,#01000001B,MAIN   ;CHECK RECIEVE
"A"
                CALL  SETMODE_TX
RX:             JNB   RI,$
                CLR   RI
                MOV   A,SBUF
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SEND:                   CALL  SEND_TRW
                        JMP   MAIN
;*****************************************************************
;SETMODE FOR RECIVER
;*****************************************************************
SETMODE_RX:                CLR   CE
                        SETB  CS
                        CLR   A
                        MOV   R1,#18
SETMODE_0_RX:           MOV   DPTR,#CONFIG_TEST_RX
                        PUSH  ACC
                        MOVC  A,@A+DPTR
                        CALL  WRITE_TRW24
                        POP   ACC
                        INC   A
                        DJNZ  R1,SETMODE_0_RX
                        SETB  DAT
                        SETB  DR1
                        SETB  CE
                        CLR   CS
                        RET
;*****************************************************************
;SETMODE FOR TRANCEIVER
;*****************************************************************
SETMODE_TX:                CLR   CE
                        SETB  CS
                        CLR   A
                        MOV   R1,#18
SETMODE_0_TX:           MOV   DPTR,#CONFIG_TEST_TX
                        PUSH  ACC
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    MOVC A,@A+DPTR
                    CALL  WRITE_TRW24
                    POP   ACC
                    INC   A
                    DJNZ  R1,SETMODE_0_TX
                    SETB  DAT
                    SETB  DR1
                    SETB  CE
                    CLR   CS
                    RET
;*****************************************************************
;SEND DATA FROM TRW
;*****************************************************************
SEND_TRW:           CALL  DELAY_1ms
                    CLR   CS
                    SETB  CE
                    PUSH  ACC
                    CLR   A
                    MOV   R1,#5
SEND_TRW_0:         MOV   DPTR,#CONFIG_ADDR1_TX
                    PUSH  ACC
                    MOVC A,@A+DPTR
                    CALL  WRITE_TRW24
                    POP   ACC
                    INC   A
                    DJNZ  R1,SEND_TRW_0
                    POP   ACC
                    CALL  WRITE_TRW24
                    CLR   CLK
                    CLR   CE
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        CLR    DAT
                        RET
;*****************************************************************
:CLOCK
:*****************************************************************
CLK_TRW:                CLR    CLK
                        NOP
                        SETB   CLK
                        NOP
                        RET
;*****************************************************************
;WRITE DATA FROM TRW
:*****************************************************************
WRITE_TRW24:            MOV    R0,#8
WRITE_TRW24_0:          JB     ACC.7,WRITE1
                        CLR    DAT
                        JMP    WRITE_TRW2
WRITE1:                 SETB   DAT
WRITE_TRW2:             CALL   CLK_TRW
                        RL     A
                        DJNZ   R0,WRITE_TRW24_0
                        RET
;*****************************************************************
;READ DATA FROM TRW
:*****************************************************************
READ_TRW24:             CLR    A
                        MOV    R0,#8
READ_TRW24_0:           RL     A
                        SETB   CLK
                        NOP
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        JB      DAT,READ_1
                        CLR     ACC.0
                        JMP     READ_TRW24_1
READ_1:                 SETB    ACC.0
READ_TRW24_1:           CLR     CLK
                        NOP
                        DJNZ    R0,READ_TRW24_0
                        RET
;---------------------------------------------------------
;DELAY TIME
;---------------------------------------------------------
DELAY_1ms:              MOV     R6,#0E6H ; Each loop = 1 ms
DELAY_1ms_1:            NOP
                        NOP
                        DJNZ    R6,DELAY_1ms_1
                        RET

DELAY_100ms:            MOV     R7,#100 ; Do 100 times
DELAY_100ms_1:          MOV     R6,#0E6H ; Each loop = 1 ms
DELAY_100ms_2:          NOP
                        NOP
                        DJNZ    R6,DELAY_100ms_2
                        DJNZ    R7,DELAY_100ms_1
                        RET
;*************************************************************
;CONFIG MODE TRW
;*************************************************************
CONFIG_TEST_RX:         DB 8EH,08H,1CH
CONFIG_LEN2_RX:         DB 08H
CONFIG_LEN1_RX:         DB 08H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CONFIG_ADDR2_RX:      DB 0C0H,0AAH,55H,0AAH,55H
CONFIG_ADDR1_RX:      DB 0AAH,55H,0AAH,55H,0AAH
CONFIG_NUMADDR_RX:    DB 0A3H
CONFIG_RF_RX:         DB 6FH
;CONFIG_CH_TX:        DB 0AH ;TX
CONFIG_CH_RX:         DB 0BH ;RX

CONFIG_TEST_TX:       DB 8EH,08H,1CH
CONFIG_LEN2_TX:       DB 08H
CONFIG_LEN1_TX:       DB 08H
CONFIG_ADDR2_TX:      DB 0C0H,0AAH,55H,0AAH,55H
CONFIG_ADDR1_TX:      DB 0AAH,55H,0AAH,55H,0AAH
CONFIG_NUMADDR_TX:    DB 0A3H
CONFIG_RF_TX:         DB 6FH
CONFIG_CH_TX:         DB 0AH ;TX
;CONFIG_CH_RX:        DB 0BH ;RX

END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

SLAVE 2

```
CE              BIT P1.2
CS              BIT P1.3
DAT             BIT P1.5
CLK             BIT P1.4
DR1             BIT P1.6

                ORG 0000H
;*************************************************************
INIT:           MOV   TMOD,#021H
                MOV   TH1,#0FDH
                MOV   TL1,#0FDH
                MOV   SCON,#050H
                SETB  TR1
;*************************************************************
                CLR   CE
                CLR   CS
                CLR   DAT
                CLR   CLK
;*************************************************************
MAIN:           CALL  SETMODE_RX
                JNB   DR1,$
                CALL  READ_TRW24

CHECK:          CJNE  A,#01000010B,MAIN   ;CHECK RECIEVE
"A"
                CALL  SETMODE_TX
RX:             JNB   RI,$
                CLR   RI
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    MOV   A,SBUF

SEND:               CALL  SEND_TRW
                    JMP   MAIN
;*****************************************************************
;SETMODE FOR RECIVER
;*****************************************************************
SETMODE_RX:             CLR   CE
                    SETB  CS
                    CLR   A
                    MOV   R1,#18
SETMODE_0_RX:       MOV   DPTR,#CONFIG_TEST_RX
                    PUSH  ACC
                    MOVC  A,@A+DPTR
                    CALL  WRITE_TRW24
                    POP   ACC
                    INC   A
                    DJNZ  R1,SETMODE_0_RX
                    SETB  DAT
                    SETB  DR1
                    SETB  CE
                    CLR   CS
                    RET
;*****************************************************************
;SETMODE FOR TRANCEIVER
;*****************************************************************
SETMODE_TX:             CLR   CE
                    SETB  CS
                    CLR   A
                    MOV   R1,#18
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SETMODE_0_TX:       MOV   DPTR,#CONFIG_TEST_TX
                    PUSH  ACC
                    MOVC  A,@A+DPTR
                    CALL  WRITE_TRW24
                    POP   ACC
                    INC   A
                    DJNZ  R1,SETMODE_0_TX
                    SETB  DAT
                    SETB  DR1
                    SETB  CE
                    CLR   CS
                    RET
;*************************************************************
;SEND DATA FROM TRW
;*************************************************************
SEND_TRW:           CALL  DELAY_1ms
                    CLR   CS
                    SETB  CE
                    PUSH  ACC
                    CLR   A
                    MOV   R1,#5
SEND_TRW_0:         MOV   DPTR,#CONFIG_ADDR1_TX
                    PUSH  ACC
                    MOVC  A,@A+DPTR
                    CALL  WRITE_TRW24
                    POP   ACC
                    INC   A
                    DJNZ  R1,SEND_TRW_0
                    POP   ACC
                    CALL  WRITE_TRW24
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        CLR   CLK
                        CLR   CE
                        CLR   DAT
                        RET
;*************************************************************
:CLOCK
:*************************************************************
CLK_TRW:                CLR   CLK
                        NOP
                        SETB  CLK
                        NOP
                        RET
;*************************************************************
;WRITE DATA FROM TRW
:*************************************************************
WRITE_TRW24:            MOV   R0,#8
WRITE_TRW24_0:          JB    ACC.7,WRITE1
                        CLR   DAT
                        JMP   WRITE_TRW2
WRITE1:                 SETB  DAT
WRITE_TRW2:             CALL  CLK_TRW
                        RL    A
                        DJNZ  R0,WRITE_TRW24_0
                        RET
;*************************************************************
;READ DATA FROM TRW
:*************************************************************
READ_TRW24:             CLR   A
                        MOV   R0,#8
READ_TRW24_0:           RL    A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        SETB   CLK
                        NOP
                        JB     DAT,READ_1
                        CLR    ACC.0
                        JMP    READ_TRW24_1
READ_1:                 SETB   ACC.0
READ_TRW24_1:           CLR    CLK
                        NOP
                        DJNZ   R0,READ_TRW24_0
                        RET
;---------------------------------------------------------------
;DELAY TIME
;---------------------------------------------------------------
DELAY_1ms:              MOV    R6,#0E6H ; Each loop = 1 ms
DELAY_1ms_1:            NOP
                        NOP
                        DJNZ   R6,DELAY_1ms_1
                        RET

DELAY_100ms:            MOV    R7,#100 ; Do 100 times
DELAY_100ms_1:          MOV    R6,#0E6H ; Each loop = 1 ms
DELAY_100ms_2:          NOP
                        NOP
                        DJNZ   R6,DELAY_100ms_2
                        DJNZ   R7,DELAY_100ms_1
                        RET
;*********************************************************************
;CONFIG MODE TRW
;*********************************************************************
CONFIG_TEST_RX:         DB 8EH,08H,1CH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CONFIG_LEN2_RX:        DB 08H
CONFIG_LEN1_RX:        DB 08H
CONFIG_ADDR2_RX:       DB 0C0H,0AAH,55H,0AAH,55H
CONFIG_ADDR1_RX:       DB 0AAH,55H,0AAH,55H,0AAH
CONFIG_NUMADDR_RX:     DB 0A3H
CONFIG_RF_RX:          DB 6FH
;CONFIG_CH_TX:         DB 0AH ;TX
CONFIG_CH_RX:          DB 0BH ;RX

CONFIG_TEST_TX:        DB 8EH,08H,1CH
CONFIG_LEN2_TX:        DB 08H
CONFIG_LEN1_TX:        DB 08H
CONFIG_ADDR2_TX:       DB 0C0H,0AAH,55H,0AAH,55H
CONFIG_ADDR1_TX:       DB 0AAH,55H,0AAH,55H,0AAH
CONFIG_NUMADDR_TX:     DB 0A3H
CONFIG_RF_TX:          DB 6FH
CONFIG_CH_TX:          DB 0AH ;TX
;CONFIG_CH_RX:         DB 0BH ;RX

END
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MASTER

```
CE          BIT P1.2
CS          BIT P1.3
DAT         BIT P1.5
CLK         BIT P1.4
DR1         BIT P1.6
BITTIM      EQU 45 ;(((11059200/9600)/12) - 5) / 2
SUBTIMER    EQU 30H
BUFFER      EQU 31H
CHK         EQU 32H
STRING10    EQU 33H
STRING1     EQU 34H

            ORG 0000H
            JMP   INIT

            ORG   000BH
            AJMP  TF0_SUB
;*****************************************************************
INIT:       MOV   IE,#10010010B
            MOV   TMOD,#00100001B
            MOV   PCON,#00H
            MOV   SCON,#50H
            MOV   TH1,#0FDH
            MOV   TH0,#176          ;9.9    ;175  10.05
            MOV   TL0,#176          ;9.9
            MOV   SUBTIMER,#0
            SETB  TR0
;*****************************************************************
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    CLR   CE
                    CLR   CS
                    CLR   DAT
                    CLR   CLK
                    CLR   DR1
;*****************************************************************
                    MOV   CHK,#1

MAIN:               SETB  TR0
                    JNB   DR1,$
                    CLR   TR0
                    CALL  READ_TRW24
                    MOV   BUFFER,A

CHK_R0_1:           CJNE  R0,#1,CHK_R0_2
                    MOV   A,BUFFER
                    CALL  DATA_TO_STRING
                    MOV   A,#41H
                    CALL  TX_TO_COM
                    MOV   A,R2
                    CALL  TX_TO_COM
                    MOV   A,R3
                    CALL  TX_TO_COM
                    MOV   CHK,#2
                    JMP   MAIN

CHK_R0_2:           CJNE  R0,#2,MAIN
                    MOV   A,BUFFER
                    CALL  DATA TO_STRING
                    MOV   A,#42H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                CALL   TX_TO_COM
                MOV    A,R2
                CALL   TX_TO_COM
                MOV    A,R3
                CALL   TX_TO_COM
                MOV    CHK,#1
                JMP    MAIN

TX_TO_COM:      CLR    P3.1          ;Drop line for start bit
                MOV    R0,#BITTIM    ;Wait full bit-time
                DJNZ   R0,$    ;For START bit
                MOV    R1,#8         ;Send 8 bits
PUTC1:          RRC    A             ;Move next bit into carry
                MOV    P3.1,C  ;Write next bit
                MOV    R0,#BITTIM    ;Wait full bit-time
                DJNZ   R0,$          ;For DATA bit
                DJNZ   R1,PUTC1      ;write 8 bits
                SETB   P3.1          ;Set line high
                RRC    A             ;Restore ACC contents
                MOV    R0,#BITTIM    ;Wait full bit-time
                DJNZ   R0,$          ;For STOP bit
                RET

DATA_TO_STRING: MOV    B,#10
                DIV    AB
                MOV    STRING10,A
                MOV    STRING1,B
                MOV    A,STRING10
                MOV    DPTR,#STRING_0
                MOVC   A,@A+DPTR
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    MOV   R2,A
                    MOV   A,STRING1
                    MOV   DPTR,#STRING_0
                    MOVC  A,@A+DPTR
                    MOV   R3,A
                    RET
;*********************************************************************
;SETMODE RECIEVER
;*********************************************************************
SETMODE_RX:         CLR   CE
                    SETB  CS
                    CLR   A
                    MOV   R1,#18
SETMODE_0:          MOV   DPTR,#CONFIG_TEST_RX
                    PUSH  ACC
                    MOVC  A,@A+DPTR
                    CALL  WRITE_TRW24
                    POP   ACC
                    INC   A
                    DJNZ  R1,SETMODE_0
                    SETB  DAT
                    SETB  DR1
                    SETB  CE
                    CLR   CS
                    RET
;*********************************************************************
;SETMODE TRANCIEVER
;*********************************************************************
SETMODE_TX:         CLR   CE
                    SETB  CS
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    CLR   A
                    MOV   R1,#18
SETMODE_1:          MOV   DPTR,#CONFIG_TEST_TX
                    PUSH  ACC
                    MOVC  A,@A+DPTR
                    CALL  WRITE_TRW24
                    POP   ACC
                    INC   A
                    DJNZ  R1,SETMODE_1
                    SETB  DAT
                    SETB  DR1
                    SETB  CE
                    CS
                    RET
;*************************************************************
;SEND DATA FROM TRW
;*************************************************************
SEND_TRW:           CALL  DELAY_1ms
                    CLR   CS
                    SETB  CE
                    PUSH  ACC
                    CLR   A
                    MOV   R1,#5
SEND_TRW_0:         MOV   DPTR,#CONFIG_ADDR1_TX
                    PUSH  ACC
                    MOVC  A,@A+DPTR
                    CALL  WRITE_TRW24
                    POP   ACC
                    INC   A
                    DJNZ  R1,SEND_TRW_0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    POP     ACC
                    CALL    WRITE_TRW24
                    CLR     CLK
                    CLR     CE
                    CLR     DAT
                    RET
;*****************************************************************
;CLOCK
;*****************************************************************
CLK_TRW:            CLR     CLK
                    NOP
                    SETB    CLK
                    NOP
                    RET
;*****************************************************************
;WRITE DATA FROM TRW
;*****************************************************************
WRITE_TRW24:        MOV     R4,#8
WRITE_TRW24_0:      JB      ACC.7,WRITE1
                    CLR     DAT
                    JMP     WRITE_TRW2
WRITE1:             SETB    DAT
WRITE_TRW2:         CALL    CLK_TRW
                    RL      A
                    DJNZ    R4,WRITE_TRW24_0
                    RET
;*****************************************************************
;READ DATA FROM TRW
;*****************************************************************
READ_TRW24:         CLR     A
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                        MOV   R5,#8
READ_TRW24_0:           RL    A
                        SETB  CLK
                        NOP
                        JB    DAT,READ_1
                        CLR   ACC.0
                        JMP   READ_TRW24_1
READ_1:                 SETB  ACC.0
READ_TRW24_1:           CLR   CLK
                        NOP
                        DJNZ  R5,READ_TRW24_0
                        RET
;*****************************************************
;TIMMER 0 INTERRUPT TR0
;*****************************************************
TF0_SUB:                PUSH  PSW
                        PUSH  ACC
                        INC   SUBTIMER
                        MOV   A,#20
                        CJNE  A,SUBTIMER,TF0_EXIT
                        MOV   SUBTIMER,#0
                        MOV   R0,CHK
                        CJNE  R0,#1,SEND_2

SEND_1:                 CALL  SETMODE_TX
                        MOV   A,#01000001B
                        CALL  SEND_TRW
                        CALL  SETMODE_RX
                        MOV   CHK,#2
                        JMP   TF0_EXIT
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
SEND_2:         CALL  SETMODE_TX
                MOV   A,#01000010B
                CALL  SEND_TRW
                CALL  SETMODE_RX
                MOV   CHK,#1
                JMP   TF0_EXIT

TF0_EXIT:       POP   ACC
                POP   PSW
                RETI
;*********************************************************************
;DELAY TIME
;*********************************************************************
DELAY_1ms:      MOV   R6,#0E6H ; Each loop = 1 ms
DELAY_1ms_1:    NOP
                NOP
                DJNZ  R6,DELAY_1ms_1
                RET

DELAY_100ms:    MOV   R7,#100 ; Do 100 times
DELAY_100ms_1:  MOV   R6,#0E6H ; Each loop = 1 ms
DELAY_100ms_2:  NOP
                NOP
                DJNZ  R6,DELAY_100ms_2
                DJNZ  R7,DELAY_100ms_1
                RET
;*********************************************************************
;CONFIG MODE TRW
;*********************************************************************
CONFIG_TEST_TX: DB 8EH,08H,1CH
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
CONFIG_LEN2_TX:         DB 08H
CONFIG_LEN1_TX:         DB 08H
CONFIG_ADDR2_TX:        DB 0C0H,0AAH,55H,0AAH,55H
CONFIG_ADDR1_TX:        DB 0AAH,55H,0AAH,55H,0AAH
CONFIG_NUMADDR_TX:      DB 0A3H
CONFIG_RF_TX:           DB 6FH
CONFIG_CH_TX:           DB 0AH      ;TX
;CONFIG_CH:             DB 0BH      ;RX


CONFIG_TEST_RX:         DB 8EH,08H,1CH
CONFIG_LEN2_RX:         DB 08H
CONFIG_LEN1_RX:         DB 08H
CONFIG_ADDR2_RX:        DB 0C0H,0AAH,55H,0AAH,55H
CONFIG_ADDR1_RX:        DB 0AAH,55H,0AAH,55H,0AAH
CONFIG_NUMADDR_RX:      DB 0A3H
CONFIG_RF_RX:           DB 6FH
;CONFIG_CH_TX:          DB 0AH ;TX
CONFIG_CH_RX:           DB 0BH ;RX
;*********************************************************************
;STRING 0-9
;*********************************************************************
STRING_0:               DB 30H
STRING_1:               DB 31H
STRING_2:               DB 32H
STRING_3:               DB 33H
STRING_4:               DB 34H
STRING_5:               DB 35H
STRING_6:               DB 36H
STRING_7:               DB 37H
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
STRING_8:       DB 38H
STRING_9:       DB 39H

END
```


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

VB.NET

```
Option Explicit On
Imports System.Data
Imports System.Data.OleDb

Public Class Form1
Inherits System.Windows.Forms.Form

  Public G As Graphics
  Public Obrush As New SolidBrush(Color.DarkOrange)
  Public Bbrush As New SolidBrush(Color.White)
  Public Rbrush As New SolidBrush(Color.Red)
  Public Gbrush As New SolidBrush(Color.Green)
  Public YP As New Pen(Color.Yellow, 3)
  Public OP As New Pen(Color.Orange, 3)
  Public RP As New Pen(Color.Red, 2)
  Public GP As New Pen(Color.Green, 1)

  Public fm As New FontFamily("Microsoft Sans Serif")
  Public ft As New Font(fm, 10, FontStyle.Bold, GraphicsUnit.Pixel)
  Public ft2 As New Font(fm, 15, FontStyle.Bold, GraphicsUnit.Pixel)
  Public ft3 As New Font(fm, 12, FontStyle.Bold, GraphicsUnit.Pixel)

  Public x As Single
  Public x0 As Single
  Public x1 As Single
  Public count(2) As Integer
  Public y1 As Single
  Public y2 As Single
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    Public Rxd As String
    Public tem1 As Single
    Public tem2 As Single

    Public DT1 As DataTable
    Public DT2 As DataTable
    Public DR1 As DataRow
    Public DR2 As DataRow

    Private Sub Btnconnect_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Btnconnect.Click
        If AxMSComm1.PortOpen = False Then
            AxMSComm1.CommPort = Val(Combo1.Text)
            AxMSComm1.Settings = "9600,n,8,1"
            AxMSComm1.PortOpen = True
            Btnconnect.BackColor = Color.LightGreen
        End If
    End Sub

Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer1.Tick
        lbltt.Text = Now.ToLongTimeString
        Dim myDateTime As DateTime = DateTime.Now
        Dim a As Single
        Dim b As Single
        Dim c As Single
        a = myDateTime.Hour * 60 * 60
        b = myDateTime.Minute * 60
        c = myDateTime.Second
    x = MyFunction1(a, b, c)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Select Case x

    Case 0 To 14399
        G.DrawString("0:30", ft, Obrush, 90, 10)
        G.DrawString("1:00", ft, Obrush, 180, 10)
        G.DrawString("1:30", ft, Obrush, 270, 10)
        G.DrawString("2:00", ft, Obrush, 360, 10)
        G.DrawString("2:30", ft, Obrush, 450, 10)
        G.DrawString("3:00", ft, Obrush, 540, 10)
        G.DrawString("3:30", ft, Obrush, 630, 10)

    Case 14400 To 28799
        G.DrawString("4:30", ft, Obrush, 90, 10)
        G.DrawString("5:00", ft, Obrush, 180, 10)
        G.DrawString("5:30", ft, Obrush, 270, 10)
        G.DrawString("6:00", ft, Obrush, 360, 10)
        G.DrawString("6:30", ft, Obrush, 450, 10)
        G.DrawString("7:00", ft, Obrush, 540, 10)
        G.DrawString("7:30", ft, Obrush, 630, 10)

    Case 28800 To 43199
        G.DrawString("8:30", ft, Obrush, 90, 10)
        G.DrawString("9:00", ft, Obrush, 180, 10)
        G.DrawString("9:30", ft, Obrush, 270, 10)
        G.DrawString("10:00", ft, Obrush, 360, 10)
        G.DrawString("10:30", ft, Obrush, 450, 10)
        G.DrawString("11:00", ft, Obrush, 540, 10)
        G.DrawString("11:30", ft, Obrush, 630, 10)
    Case 43200 To 57599
        G.DrawString("12:30", ft, Obrush, 90, 10)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            G.DrawString("13:00", ft, Obrush, 180, 10)
            G.DrawString("13:30", ft, Obrush, 270, 10)
            G.DrawString("14:00", ft, Obrush, 360, 10)
            G.DrawString("14:30", ft, Obrush, 450, 10)
            G.DrawString("15:00", ft, Obrush, 540, 10)
            G.DrawString("15:30", ft, Obrush, 630, 10)

        Case 57600 To 71999
            G.DrawString("16:30", ft, Obrush, 90, 10)
            G.DrawString("17:00", ft, Obrush, 180, 10)
            G.DrawString("17:30", ft, Obrush, 270, 10)
            G.DrawString("18:00", ft, Obrush, 360, 10)
            G.DrawString("18:30", ft, Obrush, 450, 10)
            G.DrawString("19:00", ft, Obrush, 540, 10)
            G.DrawString("19:30", ft, Obrush, 630, 10)

        Case 72000 To 86399
            G.DrawString("20:30", ft, Obrush, 90, 10)
            G.DrawString("21:00", ft, Obrush, 180, 10)
            G.DrawString("21:30", ft, Obrush, 270, 10)
            G.DrawString("22:00", ft, Obrush, 360, 10)
            G.DrawString("22:30", ft, Obrush, 450, 10)
            G.DrawString("23:00", ft, Obrush, 540, 10)
            G.DrawString("23:30", ft, Obrush, 630, 10)

        End Select
    End Sub
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        lbldate.Text = Now.ToLongDateString
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        txtbox1.Text = ""
        txtbox2.Text = ""
    End Sub

Private Sub Form1_Paint(ByVal sender As Object, ByVal e As
System.Windows.Forms.PaintEventArgs) Handles MyBase.Paint
        G = Me.CreateGraphics
        G.TranslateTransform(280, 600)
        G.FillRectangle(Bbrush, -25, -520, 750, 550)
        G.DrawLine(OP, -25, -520, 725, -520)
        G.DrawLine(OP, -25, -520, -25, 30)
        G.DrawLine(OP, -25, 30, 725, 30)
        G.DrawLine(OP, 725, -520, 725, 30)

        G.DrawLine(YP, 0, 0, 0, -450)
        G.DrawLine(YP, 0, 0, 720, 0)
        Dim intX As Integer
        Dim intY As Integer

        Do
            intX = intX + 90
            G.DrawLine(YP, intX, -4, intX, 4)
        Loop Until intX = 720

        Do
            intY = intY - 90
            G.DrawLine(YP, 4, intY, -4, intY)
        Loop Until intY = -450

        For intY = 10 To 50 Step 10
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        G.DrawString(intY, ft, Obrush, -20, -intY * 9)
    Next intY

    G.DrawString("0", ft, Obrush, -20, 10)
    G.DrawString("กราฟแสดงความสัมพันธ์ระหว่างเวลาและอุณหภูมิ", ft2, Obrush, 200,
-495)
    G.DrawString("Sensor 1", ft3, Rbrush, 650, -490)
    G.DrawString("Sensor 2", ft3, Gbrush, 650, -470)
  End Sub

Private Sub AxMSComm1_OnComm(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles AxMSComm1.OnComm

    Static Dim index As Byte = 1
    Static Dim buff(3) As Byte
    Dim i As Byte
    Dim size As Byte
    Dim ch As Char

    If AxMSComm1.CommEvent = 2 Then
      Rxd = AxMSComm1.Input
      txtbox3.AppendText(Rxd)
      size = Len(Rxd)
      For i = 1 To size
        ch = Mid(Rxd, i, 1)
        If ch = "A" Or ch = "B" Then
          index = 1
          buff(index) = Asc(ch)
        Else
          buff(index) = Mid(Rxd, i, 1)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            End If
            If index = 3 Then
              If buff(1) = Asc("A") Then
                txtbox1.Text = (buff(2) * 10 + buff(3)) / 2

                tem1 = txtbox1.Text
                y1 = MyFunction2(tem1)

                count(1) = count(1) + 1

              ElseIf buff(1) = Asc("B") Then
                txtbox2.Text = (buff(2) * 10 + buff(3)) / 2

                tem2 = txtbox2.Text
                y2 = MyFunction3(tem2)

                count(2) = count(2) + 1

              End If
            End If
            index = (index Mod 3) + 1

        Next i
      End If
    End Sub

Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer2.Tick
      If count(1) > 0 Then
        count(1) = 0
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Dim myDateTime As DateTime = DateTime.Now
Dim h As Single
Dim m As Single
Dim s As Single

h = myDateTime.Hour * 60 * 60
m = myDateTime.Minute * 60
s = myDateTime.Second

x0 = MyFunction1(h, m, s)

Select Case x0

  Case 0 To 14399
    x1 = Myfunction4(x0)

  Case 14400 To 28799
    x1 = MyFunction5(x0)

  Case 28800 To 43199
    x1 = MyFunction6(x0)

  Case 43200 To 57599
    x1 = MyFunction7(x0)

  Case 57600 To 71999
    x1 = MyFunction8(x0)

  Case 72000 To 86399
    x1 = MyFunction9(x0)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
End Select
        lbltemp1.Text = tem1
        lbltime1.Text = x1
        G.DrawEllipse(RP, x1, y1, 2, 2)

        OleDbDataAdapter1.Fill(DataSet11, "Table1")

        DR1 = DataSet11.Tables("Table1").NewRow()
        DR1("Temperature") = tem1
        DR1("day") = DateTime.Now.Day
        DR1("month") = DateTime.Now.Month
        DR1("year") = DateTime.Now.Year
        DR1("Hour") = DateTime.Now.Hour
        DR1("Minute") = DateTime.Now.Minute
        DataSet11.Tables("Table1").Rows.Add(DR1)

        Dim commandbuild As New OleDbCommandBuilder(OleDbDataAdapter1)
        commandbuild = New OleDbCommandBuilder(OleDbDataAdapter1)

        OleDbDataAdapter1.Update(DataSet11, "Table1")
        DataSet11.AcceptChanges()

    End If
End Sub

Private Sub btngo_Click(ByVal sender As System.Objeet, ByVal e As System.EventArgs)
Handles btngo.Click
    Dim newform As New Form2
    newform.Show()
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    End Sub
Private Sub Timer3_Tick(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles Timer3.Tick

    If count(2) > 0 Then
        count(2) = 0

        Dim myDateTime As DateTime = DateTime.Now

        Dim h As Single
        Dim m As Single
        Dim s As Single

        h = myDateTime.Hour * 60 * 60
        m = myDateTime.Minute * 60
        s = myDateTime.Second

        x0 = MyFunction1(h, m, s)

        Select Case x0
Case 0 To 14399
            x1 = Myfunction4(x0)

        Case 14400 To 28799
            x1 = MyFunction5(x0)

        Case 28800 To 43199
            x1 = MyFunction6(x0)

        Case 43200 To 57599
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            x1 = MyFunction7(x0)
        Case 57600 To 71999
            x1 = MyFunction8(x0)

        Case 72000 To 86399
            x1 = MyFunction9(x0)

    End Select
    lbltemp2.Text = tem2
    lbltime2.Text = x1
    G.DrawEllipse(GP, x1, y2, 2, 2)

    OleDbDataAdapter2.Fill(DataSet11, "Table2")
    DR2 = DataSet11.Tables("Table2").NewRow()
    DR2("Temperature") = tem2
    DR2("day") = DateTime.Now.Day
    DR2("month") = DateTime.Now.Month
    DR2("year") = DateTime.Now.Year
    DR2("Hour") = DateTime.Now.Hour
    DR2("Minute") = DateTime.Now.Minute
    DataSet11.Tables("Table2").Rows.Add(DR2)

    Dim commandbuild As New OleDbCommandBuilder(OleDbDataAdapter2)
    commandbuild = New OleDbCommandBuilder(OleDbDataAdapter2)
    OleDbDataAdapter2.Update(DataSet11, "Table2")
    DataSet11.AcceptChanges()

  End If
 End Sub
End Class
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Module Module1
    Public Function MyFunction1(ByVal X As Single, ByVal Y As Single, ByVal Z As
Single)
        MyFunction1 = (X + Y + Z)
    End Function
    Public Function MyFunction2(ByVal A As Single)
        MyFunction2 = A * -9
    End Function
    Public Function MyFunction3(ByVal B As Single)
        MyFunction3 = B * -9
    End Function
    Public Function Myfunction4(ByVal X1)
        Myfunction4 = X1 / 20
    End Function
    Public Function MyFunction5(ByVal X2)
    MyFunction5 = (X2 - 14400) / 20
    End Function
    Public Function MyFunction6(ByVal X3)
        MyFunction6 = (X3 - 28800) / 2
    End Function
    Public Function MyFunction7(ByVal X4)
        MyFunction7 = (X4 - 43200) / 20
    End Function
    Public Function MyFunction8(ByVal X5)
        MyFunction8 = (X5 - 57600) / 20
    End Function
    Public Function MyFunction9(ByVal X6)
        MyFunction9 = (X6 - 72000) / 20
    End Function
End Module
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Public Class Form2
    Inherits System.Windows.Forms.Form

Private Sub btnsearch_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles btnsearch.Click

        Dim myRowView As DataRowView
        Dim totalRec As Integer
        Dim j As Date
        Dim dvm As New DataViewManager

        DataView1 = DataSet14.Tables("Table1").DefaultView
        dvm.DataSet = DataSet14
        Dim dvs As DataViewSetting
        dvs = dvm.DataViewSettings("Table1")
        j = Convert.ToString(Convert.ToDateTime(comboday.Text + " " & combomonth.Text
& " " & comboyear.Text))
        Label6.Text = j
        DataView1.RowFilter = "day=" & comboday.Text And "month=" &
combomonth.Text And "year=" & comboyear.Text

        With DataGrid1
            .AllowNavigation = False
            .ReadOnly = True
            .DataSource = dvm
            .DataMember = "Table1"
        End With
        totalRec = DataView1.Count()
        DataGrid1.CaptionText = totalRec & " " & "Records"
End Sub
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Private Sub Form2_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
    adaptertable1.Fill(DataSet14, "Table1")
    DataGrid1.CaptionText = DataView1.Count & " " & "Records"
    adaptertable2.Fill(DataSet14, "Table3")
    DataGrid2.CaptionText = DataView2.Count & " " & "Records"

  End Sub
End Class
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

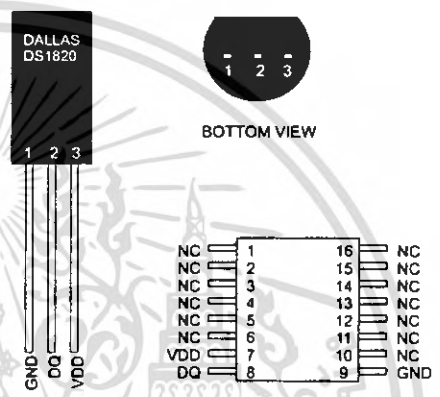
DALLAS SEMICONDUCTOR

DS1820
1–Wire™ Digital Thermometer

FEATURES

- Unique 1–Wire™ interface requires only one port pin for communication
- Multidrop capability simplifies distributed temperature sensing applications
- Requires no external components
- Can be powered from data line
- Zero standby power required
- Measures temperatures from –55°C to +125°C in 0.5°C increments. Fahrenheit equivalent is –67°F to +257°F in 0.9°F increments
- Temperature is read as a 9–bit digital value.
- Converts temperature to digital word in 200 ms (typ.)
- User–definable, nonvolatile temperature alarm settings
- Alarm search command identifies and addresses devices whose temperature is outside of programmed limits (temperature alarm condition)
- Applications include thermostatic controls, industrial systems, consumer products, thermometers, or any thermally sensitive system

PIN ASSIGNMENT



DS1820
PR35 PACKAGE
See Mech. Drawings Section

DS1820S
16–PIN SSOP
See Mech. Drawings Section

PIN DESCRIPTION

GND	–	Ground
DQ	–	Data In/Out
V_{DD}	–	Optional V_{DD}
NC	–	No Connect

DESCRIPTION

The DS1820 Digital Thermometer provides 9–bit temperature readings which indicate the temperature of the device.

Information is sent to/from the DS1820 over a 1–Wire interface, so that only one wire (and ground) needs to be connected from a central microprocessor to a DS1820. Power for reading, writing, and performing temperature conversions can be derived from the data line itself with no need for an external power source.

Because each DS1820 contains a unique silicon serial number, multiple DS1820s can exist on the same 1–Wire bus. This allows for placing temperature sensors in many different places. Applications where this feature is useful include HVAC environmental controls, sensing temperatures inside buildings, equipment or machinery, and in process monitoring and control.

©Copyright 1995 by Dallas Semiconductor Corporation.
All Rights Reserved. For important information regarding patents and other intellectual property rights, please refer to Dallas Semiconductor data books.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

DETAILED PIN DESCRIPTION

PIN 16–PIN SSOP	PIN PR35	SYMBOL	DESCRIPTION
9	1	GND	**Ground.**
8	2	DQ	**Data Input/Output pin.** For 1–Wire operation: Open drain. (See "Parasite Power" section.)
7	3	V_{DD}	**Optional V_{DD} pin.** See "Parasite Power" section for details of connection.

DS1820S (16–pin SSOP): All pins not specified in this table are not to be connected.

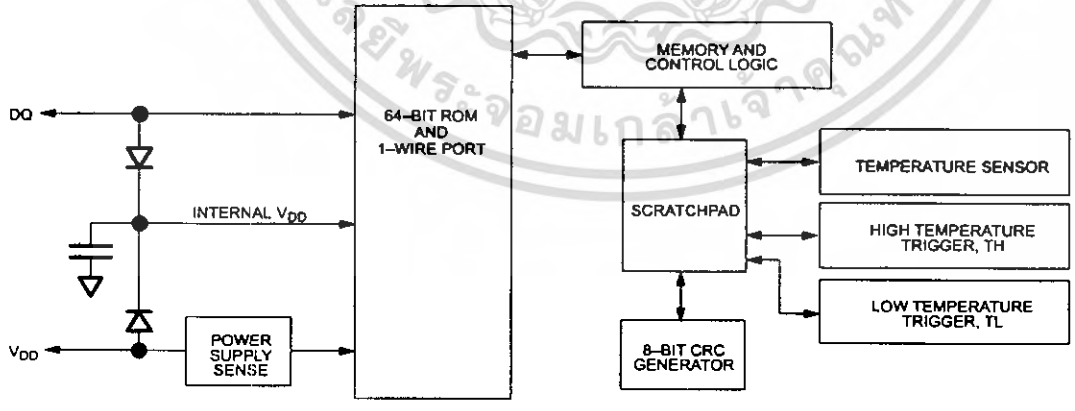
OVERVIEW

The block diagram of Figure 1 shows the major components of the DS1820. The DS1820 has three main data components: 1) 64–bit lasered ROM, 2) temperature sensor, and 3) nonvolatile temperature alarm triggers TH and TL. The device derives its power from the 1–Wire communication line by storing energy on an internal capacitor during periods of time when the signal line is high and continues to operate off this power source during the low times of the 1–Wire line until it returns high to replenish the parasite (capacitor) supply. As an alternative, the DS1820 may also be powered from an external 5 volts supply.

Communication to the DS1820 is via a 1–Wire port. With the 1–Wire port, the memory and control functions will not be available before the ROM function protocol has been established. The master must first provide one of five ROM function commands: 1) Read ROM, 2) Match ROM, 3) Search ROM, 4) Skip ROM, or 5) Alarm Search. These commands operate on the 64–bit lasered ROM portion of each device and can single out a specific device if many are present on the 1–Wire line as well as indicate to the Bus Master how many and what types of devices are present. After a ROM function sequence has been successfully executed, the memory and control functions are accessible and the master may then provide any one of the six memory and control function commands.

One control function command instructs the DS1820 to perform a temperature measurement. The result of this measurement will be placed in the DS1820's scratchpad memory, and may be read by issuing a memory function command which reads the contents of the scratchpad memory. The temperature alarm triggers TH and TL consist of one byte EEPROM each. If the alarm search command is not applied to the DS1820, these registers may be used as general purpose user memory. Writing TH and TL is done using a memory function command. Read access to these registers is through the scratchpad. All data is read and written least significant bit first.

DS1820 BLOCK DIAGRAM Figure 1



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PARASITE POWER

The block diagram (Figure 1) shows the parasite powered circuitry. This circuitry "steals" power whenever the I/O or V_{DD} pins are high. I/O will provide sufficient power as long as the specified timing and voltage requirements are met (see the section titled "1–Wire Bus System"). The advantages of parasite power are two–fold: 1) by parasiting off this pin, no local power source is needed for remote sensing of temperature, and 2) the ROM may be read in absence of normal power.

In order for the DS1820 to be able to perform accurate temperature conversions, sufficient power must be provided over the I/O line when a temperature conversion is taking place. Since the operating current of the DS1820 is up to 1 mA, the I/O line will not have sufficient drive due to the 5K pull–up resistor. This problem is particularly acute if several DS1820's are on the same I/O and attempting to convert simultaneously.

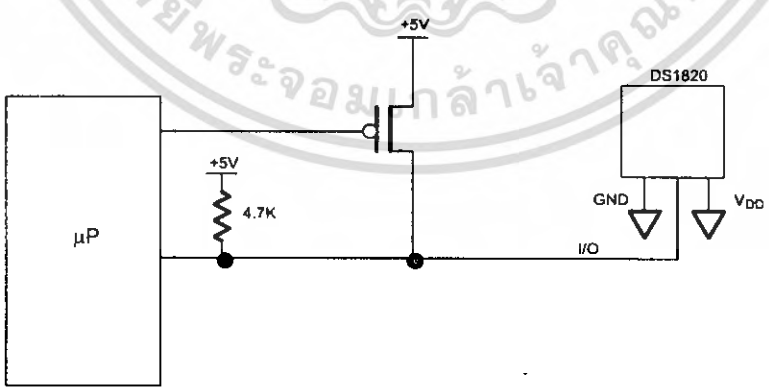
There are two ways to assure that the DS1820 has sufficient supply current during its active conversion cycle. The first is to provide a strong pull–up on the I/O line whenever temperature conversions or copies to the E^2 memory are taking place. This may be accomplished by using a MOSFET to pull the I/O line directly to the power supply as shown in Figure 2. The I/O line must be switched over to the strong pull–up within 10 µs maximum after issuing any protocol that involves copying to the E^2 memory or initiates temperature conversions. When using the parasite power mode, the V_{DD} pin must be tied to ground.

Another method of supplying current to the DS1820 is through the use of an external power supply tied to the V_{DD} pin, as shown in Figure 3. The advantage to this is that the strong pull–up is not required on the I/O line, and the bus master need not be tied up holding that line high during temperature conversions. This allows other data traffic on the 1–Wire bus during the conversion time. In addition, any number of DS1820's may be placed on the 1–Wire bus, and if they all use external power, they may all simultaneously perform temperature conversions by issuing the Skip ROM command and then issuing the Convert T command. Note that as long as the external power supply is active, the GND pin may not be floating.

The use of parasite power is not recommended above 100°C, since it may not be able to sustain communications given the higher leakage currents the DS1820 exhibits at these temperatures. For applications in which such temperatures are likely, it is strongly recommended that V_{DD} be applied to the DS1820.

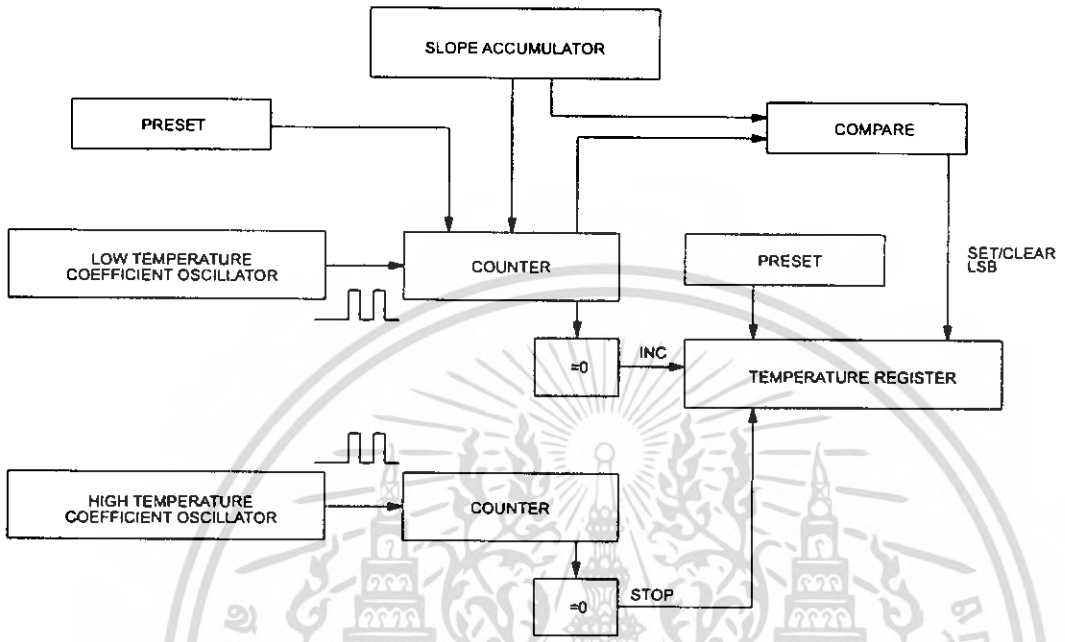
For situations where the bus master does not know whether the DS1820's on the bus are parasite powered or supplied with external V_{DD}, a provision is made in the DS1820 to signal the power supply scheme used. The bus master can determine if any DS1820's are on the bus which require the strong pull–up by sending a Skip ROM protocol, then issuing the read power supply command. After this command is issued, the master then issues read time slots. The DS1820 will send back "0" on the 1–Wire bus if it is parasite powered; it will send back a "1" if it is powered from the V_{DD} pin. If the master receives a "0", it knows that it must supply the strong pull–up on the I/O line during temperature conversions: See "Memory Command Functions" section for more detail on this command protocol.

STRONG PULL–UP FOR SUPPLYING DS1820 DURING TEMPERATURE CONVERSION Figure 2



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TEMPERATURE MEASURING CIRCUITRY Figure 4



TEMPERATURE/DATA RELATIONSHIPS Table 1

TEMPERATURE	DIGITAL OUTPUT (Binary)	DIGITAL OUTPUT (Hex)
+125°C	00000000 11111010	00FA
+25°C	00000000 00110010	0032h
$+^1/_2$°C	00000000 00000001	0001h
+0°C	00000000 00000000	0000h
$-^1/_2$°C	11111111 11111111	FFFFh
–25°C	11111111 11001110	FFCEh
–55°C	11111111 10010010	FF92h

OPERATION – ALARM SIGNALING

After the DS1820 has performed a temperature conversion, the temperature value is compared to the trigger values stored in TH and TL. Since these registers are 8–bit only, the 0.5°C bit is ignored for comparison. The most significant bit of TH or TL directly corresponds to the sign bit of the 16–bit temperature register. If the result of a temperature measurement is higher than TH or lower than TL, an alarm flag inside the device is set. This flag is updated with every temperature measurement. As long as the alarm flag is set, the DS1820 will respond to the alarm search command. This allows many DS1820s to be connected in parallel doing simultaneous temperature measurements. If somewhere the temperature exceeds the limits, the alarming device(s) can be identified and read immediately without having to read non–alarming devices.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

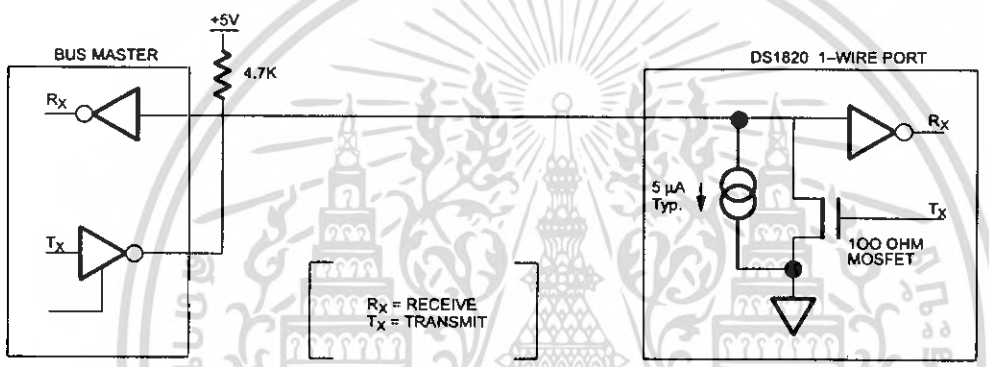
1–WIRE BUS SYSTEM

The 1–Wire bus is a system which has a single bus master and one or more slaves. The DS1820 behaves as a slave. The discussion of this bus system is broken down into three topics: hardware configuration, transaction sequence, and 1–Wire signaling (signal types and timing).

HARDWARE CONFIGURATION

The 1–Wire bus has only a single line by definition; it is important that each device on the bus be able to drive it at the appropriate time. To facilitate this, each device attached to the 1–Wire bus must have open drain or 3–state outputs. The 1–Wire port of the DS1820 (I/O pin) is open drain with an internal circuit equivalent to that shown in Figure 9. A multidrop bus consists of a 1–Wire bus with multiple slaves attached. The 1–Wire bus requires a pullup resistor of approximately 5KΩ.

HARDWARE CONFIGURATION Figure 9



The idle state for the 1–Wire bus is high. If for any reason a transaction needs to be suspended, the bus MUST be left in the idle state if the transaction is to resume. Infinite recovery time can occur between bits so long as the 1–Wire bus is in the inactive (high) state during the recovery period. If this does not occur and the bus is left low for more than 480 µs, all components on the bus will be reset.

TRANSACTION SEQUENCE

The protocol for accessing the DS1820 via the 1–Wire port is as follows:

- Initialization
- ROM Function Command
- Memory Function Command
- Transaction/Data

INITIALIZATION

All transactions on the 1–Wire bus begin with an initialization sequence. The initialization sequence consists of a reset pulse transmitted by the bus master followed by presence pulse(s) transmitted by the slave(s).

The presence pulse lets the bus master know that the DS1820 is on the bus and is ready to operate. For more details, see the "1–Wire Signaling" section.

ROM FUNCTION COMMANDS

Once the bus master has detected a presence, it can issue one of the five ROM function commands. All ROM function commands are 8–bits long. A list of these commands follows (refer to flowchart in Figure 6):

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Read ROM [33h]

This command allows the bus master to read the DS1820's 8–bit family code, unique 48–bit serial number, and 8–bit CRC. This command can only be used if there is a single DS1820 on the bus. If more than one slave is present on the bus, a data collision will occur when all slaves try to transmit at the same time (open drain will produce a wired AND result).

Match ROM [55h]

The match ROM command, followed by a 64–bit ROM sequence, allows the bus master to address a specific DS1820 on a multidrop bus. Only the DS1820 that exactly matches the 64–bit ROM sequence will respond to the following memory function command. All slaves that do not match the 64–bit ROM sequence will wait for a reset pulse. This command can be used with a single or multiple devices on the bus.

Skip ROM [CCh]

This command can save time in a single drop bus system by allowing the bus master to access the memory functions without providing the 64–bit ROM code. If more than one slave is present on the bus and a read command is issued following the Skip ROM command, data collision will occur on the bus as multiple slaves transmit simultaneously (open drain pulldowns will produce a wired AND result).

Search ROM [F0h]

When a system is initially brought up, the bus master might not know the number of devices on the 1–Wire bus or their 64–bit ROM codes. The search ROM command allows the bus master to use a process of elimination to identify the 64–bit ROM codes of all slave devices on the bus.

Alarm Search [ECh]

The flowchart of this command is identical to the Search ROM command. However, the DS1820 will respond to this command only if an alarm condition has been encountered at the last temperature measurement. An alarm condition is defined as a temperature higher than TH or lower than TL. The alarm condition remains set as long as the DS1820 is powered up, or until another temperature measurement reveals a non–alarming value. For alarming, the trigger values stored in EEPROM are taken into account. If an alarm condition exists and the TH or TL settings are changed, another temperature conversion should be done to validate any alarm conditions.

Example of a ROM Search

The ROM search process is the repetition of a simple 3–step routine: read a bit, read the complement of the bit, then write the desired value of that bit. The bus master performs this simple, 3–step routine on each bit of the ROM. After one complete pass, the bus master knows the contents of the ROM in one device. The remaining number of devices and their ROM codes may be identified by additional passes.

The following example of the ROM search process assumes four different devices are connected to the same 1–Wire bus. The ROM data of the four devices is as shown:

ROM1 00110101...
ROM2 10101010...
ROM3 11110101...
ROM4 00010001...

The search process is as follows:

1. The bus master begins the initialization sequence by issuing a reset pulse. The slave devices respond by issuing simultaneous presence pulses.
2. The bus master will then issue the Search ROM command on the 1–Wire bus.
3. The bus master reads a bit from the 1–Wire bus. Each device will respond by placing the value of the first bit of their respective ROM data onto the 1–Wire bus. ROM1 and ROM4 will place a 0 onto the 1–Wire bus, i.e., pull it low. ROM2 and ROM3 will place a 1 onto the 1–Wire bus by allowing the line to stay high. The result is the logical AND of all devices on the line, therefore the bus master sees a 0. The bus master reads another bit. Since the Search ROM data command is being executed, all of the devices on the 1–Wire bus respond to this second read by placing the complement of the first bit of their respective ROM data onto the 1–Wire bus. ROM1 and ROM4 will place a 1 onto the 1–Wire, allowing the line to stay high. ROM2 and ROM3 will place a 0 onto the 1–Wire, thus it will be pulled low. The bus master again observes a 0 for the complement of the first ROM data bit. The bus master has determined that there are some devices on the 1–Wire bus that have a 0 in the first position and others that have a 1.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

The data obtained from the two reads of the 3–step routine have the following interpretations:

00 There are still devices attached which have conflicting bits in this position.

01 All devices still coupled have a 0–bit in this bit position.

10 All devices still coupled have a 1–bit in this bit position.

11 There are no devices attached to the 1–Wire bus.

4. The bus master writes a 0. This deselects ROM2 and ROM3 for the remainder of this search pass, leaving only ROM1 and ROM4 connected to the 1–Wire bus.
5. The bus master performs two more reads and receives a 0–bit followed by a 1–bit. This indicates that all devices still coupled to the bus have 0's as their second ROM data bit.
6. The bus master then writes a 0 to keep both ROM1 and ROM4 coupled.
7. The bus master executes two reads and receives two 0–bits. This indicates that both 1–bits and 0–bits exist as the third bit of the ROM data of the attached devices.
8. The bus master writes a 0–bit. This deselects ROM1 leaving ROM4 as the only device still connected.
9. The bus master reads the remainder of the ROM bits for ROM4 and continues to access the part if desired. This completes the first pass and uniquely identifies one part on the 1–Wire bus.
10. The bus master starts a new ROM search sequence by repeating steps 1 through 7.
11. The bus master writes a 1–bit. This decouples ROM4, leaving only ROM1 still coupled.
12. The bus master reads the remainder of the ROM bits for ROM1 and communicates to the underlying logic if desired. This completes the second ROM search pass, in which another of the ROMs was found.
13. The bus master starts a new ROM search by repeating steps 1 through 3.
14. The bus master writes a 1–bit. This deselects ROM1 and ROM4 for the remainder of this search pass, leaving only ROM2 and ROM3 coupled to the system.
15. The bus master executes two read time slots and receives two zeros.
16. The bus master writes a 0–bit. This decouples ROM3, and leaving only ROM2.
17. The bus master reads the remainder of the ROM bits for ROM2 and communicates to the underlying logic if desired. This completes the third ROM search pass, in which another of the ROMs was found.
18. The bus master starts a new ROM search by repeating steps 13 through 15.
19. The bus master writes a 1–bit. This decouples ROM2, leaving only ROM3.
20. The bus master reads the remainder of the ROM bits for ROM3 and communicates to the underlying logic if desired. This completes the fourth ROM search pass, in which another of the ROMs was found.

Note the following:
The bus master learns the unique ID number (ROM data pattern) of one 1–Wire device on each ROM Search operation. The time required to derive the part's unique ROM code is:

$$960\ \mu s + (8 + 3 \times 64)\ 61\ \mu s = 13.16\ ms$$

The bus master is therefore capable of identifying 75 different 1–Wire devices per second.

I/O SIGNALING

The DS1820 requires strict protocols to insure data integrity. The protocol consists of several types of signaling on one line: reset pulse, presence pulse, write 0, write 1, read 0, and read 1. All of these signals, with the exception of the presence pulse, are initiated by the bus master.

The initialization sequence required to begin any communication with the DS1820 is shown in Figure 11. A reset pulse followed by a presence pulse indicates the DS1820 is ready to send or receive data given the correct ROM command and memory function command.

The bus master transmits (TX) a reset pulse (a low signal for a minimum of 480 µs). The bus master then releases the line and goes into a receive mode (RX). The 1–Wire bus is pulled to a high state via the 5K pull–up resistor . After detecting the rising edge on the

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

I/O pin, the DS1820 waits 15–60 µs and then transmits the presence pulse (a low signal for 60–240 µs).

MEMORY COMMAND FUNCTIONS

The following command protocols are summarized in Table 2, and by the flowchart of Figure 10.

Write Scratchpad [4Eh]

This command writes to the scratchpad of the DS1820, starting at address 2. The next two bytes written will be saved in scratchpad memory, at address locations 2 and 3. Writing may be terminated at any point by issuing a reset.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

MODES OF OPERATION

Overview

The nRF2401 can be set in the following main modes depending on three control pins:

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

Table 6 nRF2401 main modes

For a complete overview of the nRF2401 I/O pins in the different modes please refer to Table 8.

Active modes

The nRF2401 has two active (RX/TX) modes:

- ShockBurst™
- Direct Mode

The device functionality in these modes is decided by the content of a configuration word. This configuration word is presented in configuration section.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ShockBurst™

The ShockBurst™ technology uses on-chip FIFO to clock in data at a low data rate and transmit at a very high rate thus enabling extremely power reduction.
When operating the nRF2401 in ShockBurst™, you gain access to the high data rates (1 Mbps) offered by the 2.4 GHz band without the need of a costly, high-speed micro controller (MCU) for data processing.
By putting all high speed signal processing related to RF protocol on-chip, the nRF2401 offers the following benefits:

- Highly reduced current consumption
- Lower system cost (facilitates use of less expensive micro controller)
- Greatly reduced risk of 'on-air' collisions due to short transmission time

The nRF2401 can be programmed using a simple 3-wire interface where the data rate is decided by the speed of the micro controller.
By allowing the digital part of the application to run at low speed while maximizing the data rate on the RF link, the nRF ShockBurst™ mode reduces the average current consumption in applications considerably.

ShockBurst™ principle

When the nRF2401 is configured in ShockBurst™, TX or RX operation is conducted in the following way (10 kbps for the example only).

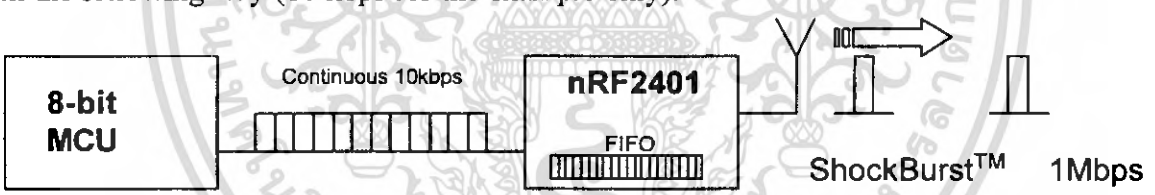


Figure 5 Clocking in data with MCU and sending with ShockBurst™ technology

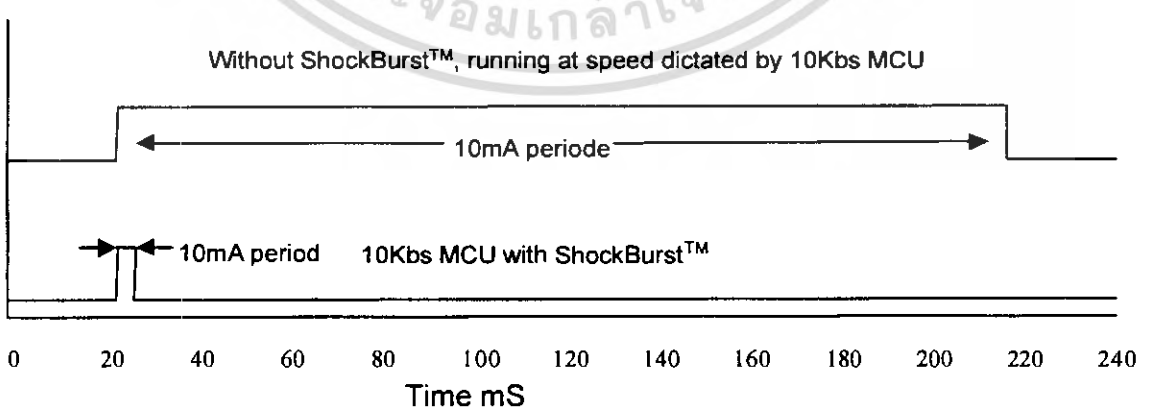


Figure 6 Current consumption with & without ShockBurst™ technology

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

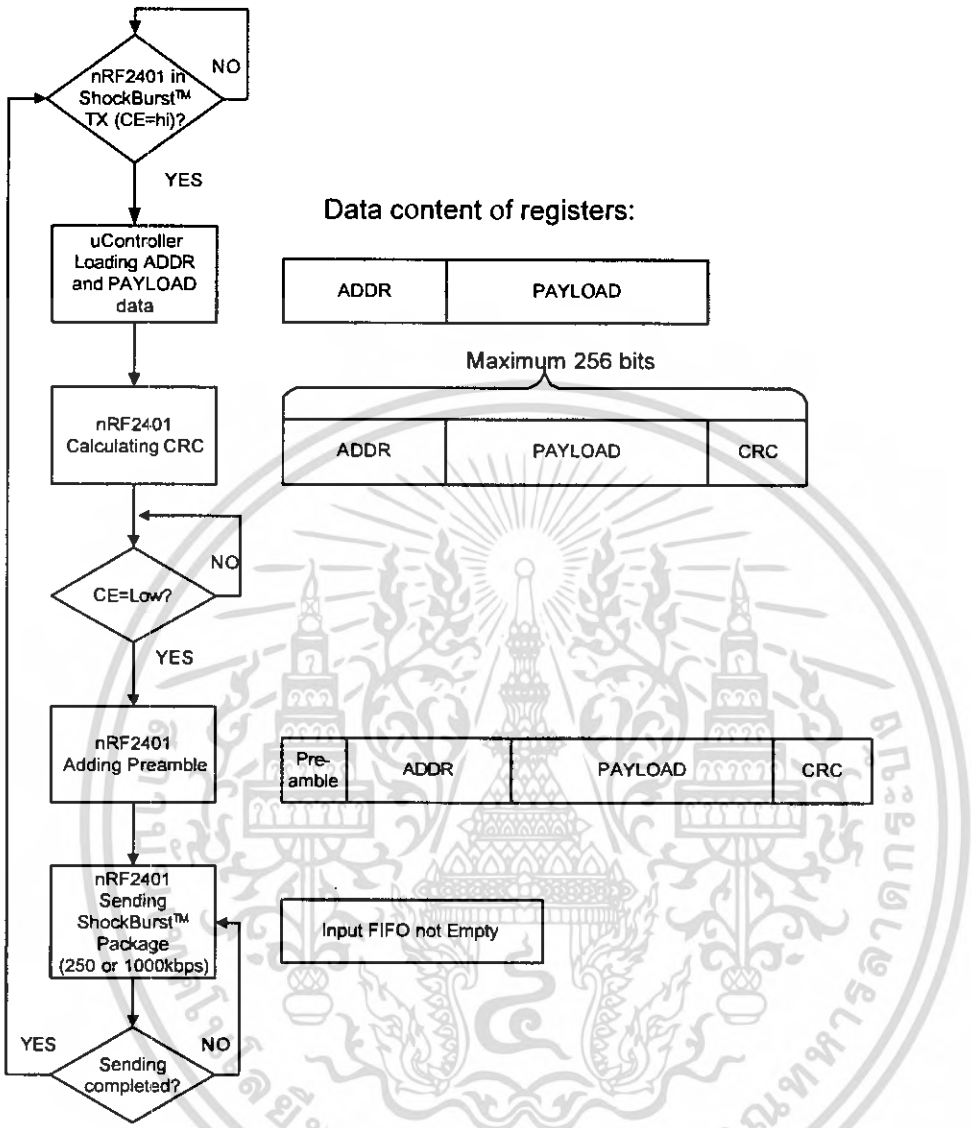


Figure 7 Flow Chart ShockBurst™ Transmit of nRF2401

nRF2401 ShockBurst™ Transmit:

MCU interface pins: CE, CLK1, DATA

1. When the application MCU has data to send, set CE high. This activates nRF2401 on-board data processing.
2. The address of the receiving node (RX address) and payload data is clocked into the nRF2401. The application protocol or MCU sets the speed <1Mbps (ex: 10kbps).
3. MCU sets CE low, this activates a nRF2401 ShockBurst™ transmission.
4. nRF2401 ShockBurst™:
   - RF front end is powered up
   - RF package is completed (preamble added, CRC calculated)
   - Data is transmitted at high speed (250 kbps or 1 Mbps configured by user).
   - nRF2401 return to stand-by when finished

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

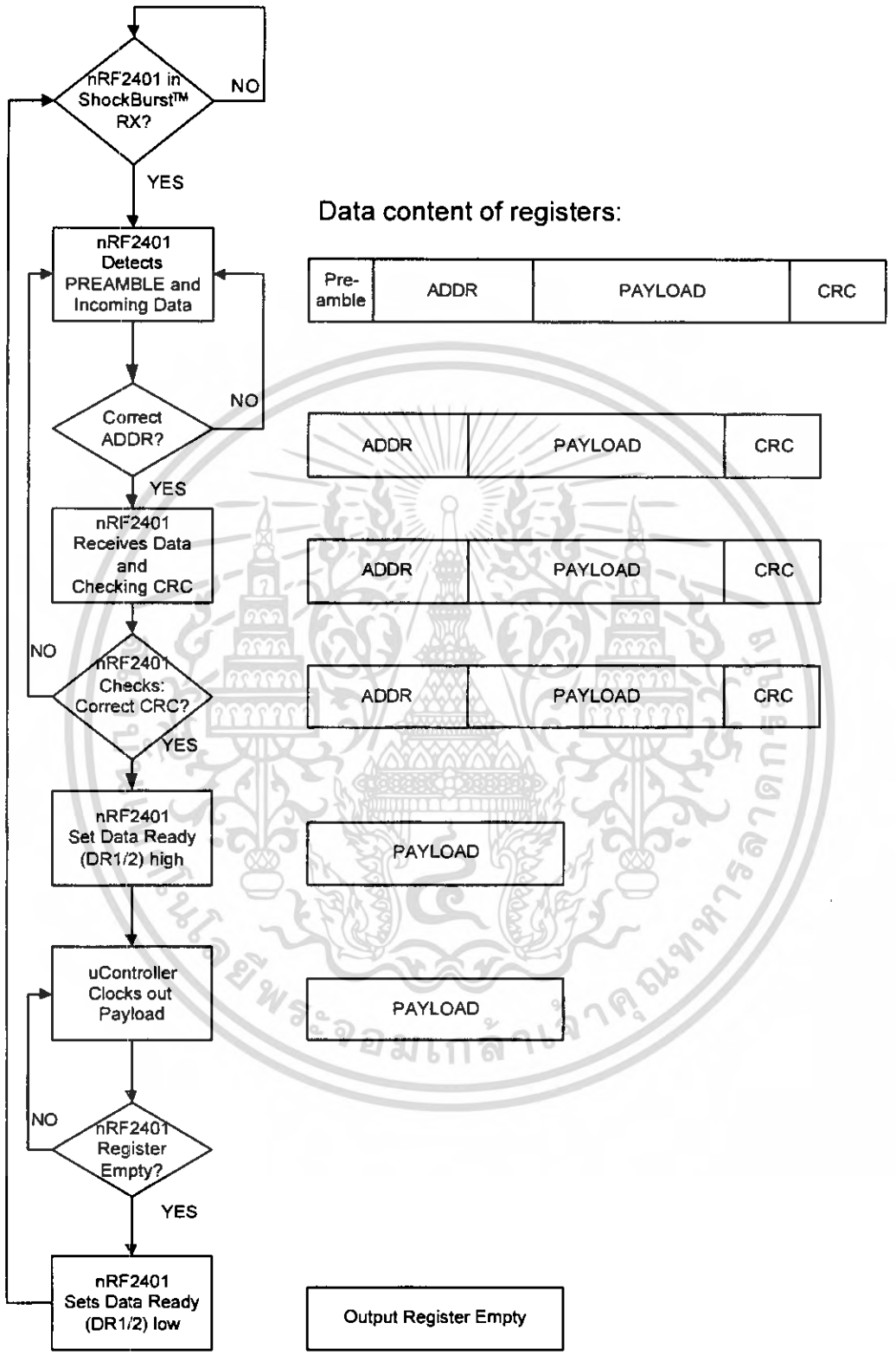


Figure 8 Flow Chart ShockBurst™ Receive of nRF2401

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

nRF2401 ShockBurst™ Receive:

MCU interface pins: CE, DR1, CLK1 and DATA (one RX channel receive)

1. Correct address and size of payload of incoming RF packages are set when nRF2401 is configured to ShockBurst™ RX.
2. To activate RX, set CE high.
3. After 200 µs settling, nRF2401 is monitoring the air for incoming communication.
4. When a valid package has been received (correct address and CRC found), nRF2401 removes the preamble, address and CRC bits.
5. nRF2401 then notifies (interrupts) the MCU by setting the DR1 pin high.
6. MCU may (or may not) set the CE low to disable the RF front end (low current mode).
7. The MCU will clock out just the payload data at a suitable rate (ex. 10 kbps).
8. When all payload data is retrieved nRF2401 sets DR1 low again, and is ready for new incoming data package if CE is kept high during data download. If the CE was set low, a new start up sequence can begin, see Figure 17.

Direct Mode

In direct mode the nRF2401 works like a traditional RF device. Data must be at 1Mbps ±200ppm, or 250kbps ±200ppm at low data rate setting, for the receiver to detect the signals.

Direct Mode Transmit:

MCU interface pins: CE, DATA

1. When application MCU has data to send, set CE high
2. The nRF2401 RF front end is now immediately activated, and after 200 µs settling time, data will modulate the carrier directly.
3. All RF protocol parts must hence be implemented in MCU firmware (preamble, address and CRC).

Direct Mode Receive:

MCU interface pins: CE, CLK1, and DATA

1. Once the nRF2401 is configured and powered up (CE high) in direct RX mode, DATA will start to toggle due to noise present on the air.
2. CLK1 will also start to toggle as nRF2401 is trying to lock on to the incoming data stream.
3. Once a valid preamble arrives, CLK1 and DATA will lock on to the incoming signal and the RF package will appear at the DATA pin with the same speed as it is transmitted.
4. To enable the demodulator to re-generate the clock, the preamble must be 8 bits toggling hi-low, starting with low if the first data bit is low.
5. In this mode no data ready (DR) signals is available. Address and checksum verification must also be done in the receiving MCU.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

DuoCeiver™ Simultaneous Two Channel Receive Mode

In both ShockBurst™ & direct modes the nRF2401 can facilitate simultaneous reception of two parallel independent frequency channels at the maximum data rate. This means:

- nRF2401 can receive data from two 1 Mbps transmitters (ex: nRF2401 or nRF2402) 8 MHz (8 frequency channels) apart through one antenna interface.
- The output from the two data channels is fed to two separate MCU interfaces.
  - Data channel 1: CLK1, DATA, and DR1
  - Data channel 2: CLK2, DOUT2, and DR2
  - DR1 and DR2 are available only in ShockBurst™.

The nRF2401 DuoCeiver™ technology provides 2 separate dedicated data channels for RX and replaces the need for two, stand alone receiver systems.

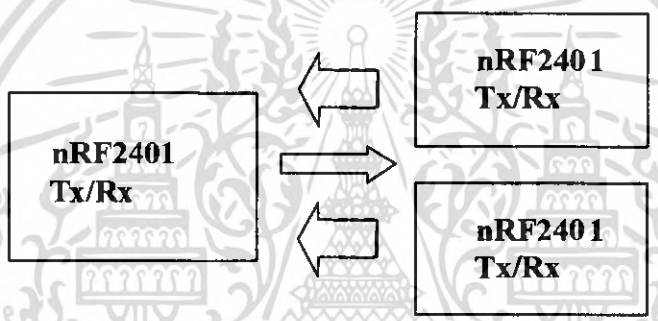


Figure 9 Simultaneous 2 channel receive on nRF2401

There is one absolute requirement for using the second data channel. For the nRF2401 to be able to receive at the second data channel the frequency channel must be 8MHz higher than the frequency of data channel 1. The nRF2401 must be programmed to receive at the frequency of data channel 1. No time multiplexing is used in nRF2401 to fulfil this function. In direct mode the MCU must be able to handle two simultaneously incoming data packets if it is not multiplexing between the two data channels. In ShockBurst™ it is possible for the MCU to clock out one data channel at a time while data on the other data channel waits for MCU availability, without any lost data packets, and by doing so reduce the needed performance of the MCU.

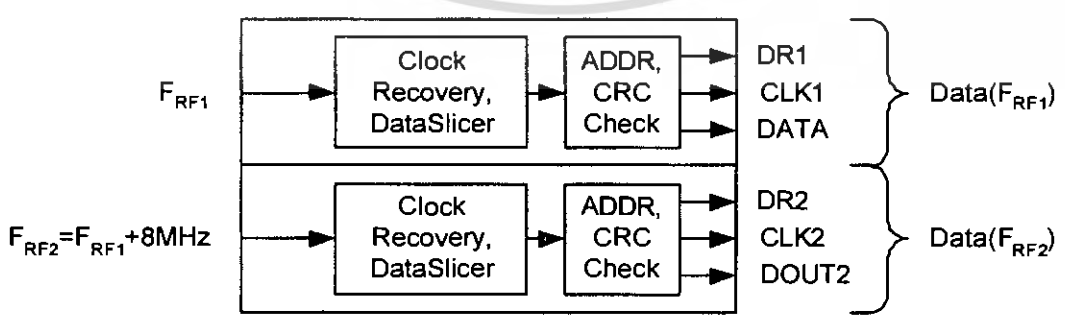


Figure 10 DuoCeiver™ with two simultaneously independent receive channels.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Configuration Mode

In configuration mode a configuration word of up to 15 bytes is downloaded to nRF2401. This is done through a simple 3-wire interface (CS, CLK1 and DATA). For more information on configuration please refer to the nRF2401 Device configuration chapter on page17.

Stand-By Mode

Stand by mode is used to minimize average current consumption while maintaining short start up times. In this mode, part of the crystal oscillator is active. Current consumption is dependent on crystal frequency (Ex: 12 µA @ 4 MHz, 32 µA @ 16 MHz). The configuration word content is maintained during stand by.

Power Down Mode

In power down the nRF2401 is disabled with minimal current consumption, typically less than 1µA. Entering this mode when the device is not active minimizes average current consumption, maximizing battery lifetime. The configuration word content is maintained during power down.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Pin configuration for the different modes of nRF2401

nRF2401 MODES	MODE SWITCHES		INPUT PINS			BIDIR PINS			OUTPUT PINS		
	RXMODE	ShockBurst	PWR_UP	CE	CS	direction CLK1	direction DATA	direction CLK2	DR1	DR2	DOUT2
Power down	X	X	0	X[1]	1	In X	In X	In X	0	0	0
Power down	0	1	0	X	0	In X	In X	In X	0	0	0
Power down	1	1	0	X	0	In CLK	In X	In X	0	0	0
Stand by	0	X	1	0	0	In X	In X	In X	0	0	0
Stand by	1	0	1	0	0	In X	In X	In X	0	0	0
Stand by	1	1	1	0	0	In CLK	In[2] DATA	In X	0	DR2	0
Stand by	1	1	1	0	0	In CLK	Out[3] DATA	In X	1	DR2	0
Configuration	X	X	1	0	1	In CLK	In CONFIG DATA	In X	0	0	0
TX ShockBurst™	0	1	1	1	0	In CLK	In DATA	In X	0	0	0
TX Direct	0	0	1	1	0	In X	In DATA	In X	0	0	0
RX ShockBurst™ in one channel	1	1	1	1	0	In CLK	Out DATA	In X	DR1	0	0
RX ShockBurst™ in two channels	1	1	1	1	0	In CLK	Out DATA	In CLK	DR1	DR2	DATA
RX Direct in one channel	1	0	1	1	0	Out CLK	Out DATA	Out 0	0	0	0
RX Direct in two channels	1	0	1	1	0	Out CLK	Out DATA	Out CLK	0	0	DATA

Table 8 Pin configuration of nRF2401.

1 In = X means the input should be set to either "low" or "high".
2 Input if DR1 is "low".
3 Output if DR1 is "high"

DEVICE CONFIGURATION

All configuration of the nRF2401 is done via a 3-wire interface to a single configuration register. The configuration word can be up to 15 bytes long for ShockBurst™ use and up to 2 bytes long for direct mode.

Configuration for ShockBurst™ operation

The configuration word in ShockBurst™ enables the nRF2401 to handle the RF protocol. Once the protocol is completed and loaded into nRF2401 only one byte, bit[7:0], needs to be updated during actual operation.

The configuration blocks dedicated to ShockBurst™ is as follows:

- Payload section width: Specifies the number of payload bits in a RF package. This enables the nRF2401 to distinguish between payload data and the CRC bytes in a received package.
- Address width: Sets the number of bits used for address in the RF package. This enables the nRF2401 to distinguish between address and payload data.
- Address (RX Channel 1 and 2): Destination address for received data.
- CRC: Enables nRF2401 on-chip CRC generation and de-coding.

NOTE:
These configuration blocks, with the exception of the CRC, are dedicated for the packages that a nRF2401 is to receive.
In TX mode, the MCU must generate an address and a payload section that fits the configuration of the nRF2401 that is to receive the data.
When using the nRF2401 on-chip CRC feature ensure that CRC is enabled and uses the same length for both the TX and RX devices.

Figure 11 Data packet set-up

Configuration for Direct Mode operation

For direct mode operation only the two first bytes (bit[15:0]) of the configuring word are relevant.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Configuration Word overview

	Bit position	Number of bits	Name	Function
ShockBurst™ configuration	143:120	24	TEST	Reserved for testing
	119:112	8	DATA2_W	Length of data payload section RX channel 2
	111:104	8	DATA1_W	Length of data payload section RX channel 1
	103:64	40	ADDR2	Up to 5 byte address for RX channel 2
	63:24	40	ADDR1	Up to 5 byte address for RX channel 1
	23:18	6	ADDR_W	Number of address bits (both RX channels).
	17	1	CRC_L	8 or 16 bit CRC
	16	1	CRC_EN	Enable on-chip CRC generation/checking.
General device configuration	15	1	RX2_EN	Enable two channel receive mode
	14	1	CM	Communication mode (Direct or ShockBurst™)
	13	1	RFDR_SB	RF data rate (1Mbps requires 16MHz crystal)
	12:10	3	XO_F	Crystal frequency
	9:8	2	RF_PWR	RF output power
	7:1	7	RF_CH#	Frequency channel
	0	1	RXEN	RX or TX operation

Table 9 Table of configuration words.

The configuration word is shifted in MSB first on positive CLK1 edges. New configuration is enabled on the falling edge of CS.

NOTE.
On the falling edge of CS, the nRF2401 updates the number of bits actually shifted in during the last configuration.
Ex:
If the nRF2401 is to be configured for 2 channel RX in ShockBurst™, a total of 120 bits must be shifted in during the first configuration after VDD is applied.
Once the wanted protocol, modus and RF channel are set, only one bit (RXEN) is shifted in to switch between RX and TX.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Configuration Word Detailed Description

The following describes the function of the 144 bits (bit 143 = MSB) that is used to configure the nRF2401.
General Device Configuration: bit[15:0]
ShockBurst™ Configuration: bit[119:16]
Test Configuration: bit[143:120]

MSB	TEST						
D143	D142	D141	D140	D139	D138	D137	D136
Reserved for testing							
1	0	0	0	1	1	1	0 Default

MSB	TEST														
D135	D134	D133	D132	D131	D130	D129	D128	D127	D126	D125	D124	D123	D122	D121	D120
Reserved for testing														Close PLL in TX	
0	0	0	0	1	0	0	0	0	0	0	1	1	1	0	0 Default

DATA2_W							
D119	D118	D117	D116	D115	D114	D113	D112
Data width channel#2 in # of bits excluding addr/crc							
0	0	1	0	0	0	0	0 Default

DATA1_W							
D111	D110	D109	D108	D107	D106	D105	D104
Data width channel#1 in # of bits excluding addr/crc							
0	0	1	0	0	0	0	0 Default

ADDR2												
D103	D102	D101	....	D71	D70	D69	D68	D67	D66	D65	D64	
Channel#2 Address RX (up to 40bit)												
0	0	0	...	1	1	1	0	0	1	1	1	Default

ADDR1												
D63	D62	D61	....	D31	D30	D29	D28	D27	D26	D25	D24	
Channel#1 Address RX (up to 40bit)												
0	0	0	...	1	1	1	0	0	1	1	1	Default

ADDR_W						
D23	D22	D21	D20	D19	D18	
Address width in # of bits (both channels)						
0	0	1	0	0	0	Default

CRC		
D17	D16	
CRC Mode 1 = 16bit, 0 = 8bit	CRC 1 = enable; 0 = disable	
0	1	Default

RF-Programming															LSB	
D15	D14	D13	D12	D11	D10	D9	D8	D7	D6	D5	D4	D3	D2	D1	D0	
Two Ch.	BUF	OD	XO Frequency			RF Power		Channel selection							RXEN	
0	0	0	0	1	1	1	1	0	0	0	0	0	1	0	0	Default

Table 10 Configuration data word

The MSB bit should be loaded first into the configuration register.
Default configuration word: h8E08.1C20.2000.0000.00E7.0000.0000.E721.0F04.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ShockBurst™ configuration:

The section bit[119:16] contains the segments of the configuration register dedicated to ShockBurst™ operational protocol. After VDD is turned on ShockBurst™ configuration is done once and remains set whilst VDD is present. During operation only the first byte for frequency channel and RX/TX switching need to be changed.

PLL_CTRL

PLL_CTRL		
D121	**D120**	**PLL**
0	0	Open TX/Closed RX
0	1	Open TX/Open RX
1	0	Closed TX/Closed RX
1	1	Closed TX/Open RX

Table 11 PLL setting.

Bit 121-120:

PLL_CTRL: Controls the setting of the PLL for test purposes. With closed PLL in TX no deviation will be present. For normal operational mode these two bits must both be low.

DATAx_W

DATA2_W							
119	118	117	116	115	114	113	112

DATA1_W							
111	110	109	108	107	106	105	104

Table 12 Number of bits in payload.

Bit 119 – 112:

DATA2_W: Length of RF package payload section for receive-channel 2.

Bit 111 – 104:

DATA1_W: Length of RF package payload section for receive-channel 1.

NOTE:

The total number of bits in a ShockBurst™ RF package may not exceed 256! Maximum length of payload section is hence given by:

$$DATAx\_W(bits) = 256 - ADDR\_W - CRC$$

Where:

ADDR_W: length of RX address set in configuration word B[23:18]
CRC: check sum, 8 or 16 bits set in configuration word B[17]
PRE: preamble, 8 bits are automatically included

Shorter address and CRC leaves more room for payload data in each package.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ADDRx

ADDR2											
103	102	101	....	71	70	69	68	67	66	65	64

ADDR1											
63	62	61	....	31	30	29	28	27	26	25	24

Table 13 Address of receiver #2 and receiver #1.

Bit 103 – 64:

ADDR2: Receiver address channel 2, up to 40 bit.

Bit 63 – 24: ADDR1

ADDR1: Receiver address channel 1, up to 40 bit.

NOTE!

Bits in ADDRx exceeding the address width set in ADDR_W are redundant and can be set to logic 0.

ADDR_W & CRC

ADDR_W						CRC_L	CRC_EN
23	22	21	20	19	18	17	16

Table 14 Number of bits reserved for RX address + CRC setting.

Bit 23 – 18:

ADDR_W: Number of bits reserved for RX address in ShockBurst™ packages.

NOTE:

Maximum number of address bits is 40 (5 bytes). Values over 40 in ADDR_W are not valid.

Bit 17:

CRC_L: CRC length to be calculated by nRF2401 in ShockBurst™.
Logic 0: 8 bit CRC
Logic 1: 16 bit CRC

Bit: 16:

CRC_EN: Enables on-chip CRC generation (TX) and verification (RX).
Logic 0: On-chip CRC generation/checking disabled
Logic 1: On-chip CRC generation/checking enabled

NOTE:

An 8 bit CRC will increase the number of payload bits possible in each ShockBurst™ data packet, but will also reduce the system integrity.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

General device configuration:

This section of the configuration word handles RF and device related parameters.

Modes:

RX2_EN	CM	RFDR_SB	XO_F			RF_PWR	
15	14	13	12	11	10	9	8

Table 15 RF operational settings.

Bit 15:

RX2_EN:

Logic 0: One channel receive
Logic 1: Two channels receive

NOTE:

In two channels receive, the nRF2401 receives on two, separate frequency channels simultaneously. The frequency of receive channel 1 is set in the configuration word bit[7-1], receive channel 2 is always 8 channels (8 MHz) above receive channel 1.

Bit 14:

Communication Mode:

Logic 0: nRF2401 operates in direct mode.
Logic 1: nRF2401 operates in ShockBurst™ mode

Bit 13:

RF Data Rate:

Logic 0: 250 kbps
Logic 1: 1 Mbps

NOTE:

Utilizing 250 kbps instead of 1Mbps will improve the receiver sensitivity by 10 dB. 1Mbps requires 16MHz crystal.

Bit 12-10:

XO_F: Selects the nRF2401 crystal frequency to be used:

XO FREQUENCY SELECTION			
D12	D11	D10	Crystal Frequency [MHz]
0	0	0	4
0	0	1	8
0	1	0	12
0	1	1	16
1	0	0	20

Table 16 Crystal frequency setting.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Bit 9-8:

RF_PWR: Sets nRF2401 RF output power in transmit mode:

RF OUTPUT POWER		
D9	D8	P [dBm]
0	0	-20
0	1	-10
1	0	-5
1	1	0

Table 17 RF output power setting.

RF channel & direction

RF_CH#							RXEN
7	6	5	4	3	2	1	0

Table 18 Frequency channel and RX / TX setting.

Bit 7 – 1:

RF_CH#: Sets the frequency channel the nRF2401 operates on.

The channel frequency in ***transmit*** is given by:

$$Channel_{RF} = 2400\ MHz + RF\_CH\# \cdot 1.0\ MHz$$

RF_CH #: between 2400MHz and 2527MHz may be set.

The channel frequency in ***data channel 1*** is given by:

$$Channel_{RF} = 2400\ MHz + RF\_CH\# \cdot 1.0\ MHz$$ (Receive at PIN#8)

RF_CH #: between 2400MHz and 2524MHz may be set.

NOTE:
The channels above 83 can only be utilized in certain territories (ex: Japan)

The channel frequency in ***data channel 2*** is given by:

$$Channel_{RF} = 2400\ MHz + RF\_CH\# \cdot 1.0\ MHz$$ +8MHz (Receive at PIN#4)

RF_CH #: between 2408MHz and 2524MHz may be set.

Bit 0:

Set active mode:

Logic 0: transmit mode
Logic 1: receive mode

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

DATA PACKAGE DESCRIPTION

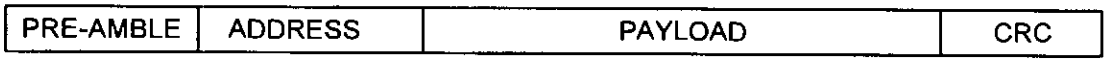


Figure 12 Data Package Diagram

The data packet for both ShockBurst™ mode and direct mode communication is divided into 4 sections. These are:

1. PREAMBLE	• The preamble field is required in ShockBurst™ and Direct modes. • Preamble is 8 bits in length and is dependent of the first data bit in direct mode. PREAMBLE — 1st ADDR-BIT 01010101 — 0 10101010 — 1 • Preamble is automatically added to the data packet in ShockBurst™ and thereby gives extra space for payload. In Direct mode MCU must handle preamble. In ShockBurst™ mode RX, the preamble is removed from the received output data, in direct mode the preamble is transparent to the output data.
2 ADDRESS	• The address field is required in ShockBurst™ mode.[1] • 8 to 40 bits length. • Address automatically removed from received packet in ShockBurst™ mode. In Direct mode MCU must handle address.
3 PAYLOAD	• The data to be transmitted • In ShockBurst™ mode payload size is 256 bits minus the following: (Address: 8 to 40 bits. + CRC 8 or 16 bits). • In Direct mode the maximum packet size (length) is for 1Mbps 4000 bits (4ms).
4 CRC	• The CRC is optional in ShockBurst™ mode, and is not used in Direct mode. • 8 or 16 bits length • The CRC is removed from the received output data in ShockBurst™ RX.

Table 19 Data package description

[1] Suggestions for the use of addresses in ShockBurst™: In general more bits in the address gives less false detection, which in the end may give lower data packet loss.

A. The address made by (5, 4, 3, or 2) equal bytes are not recommended because it in general will make the packet-error-rate increase.

B. Addresses where the level shift only one time (i.e. 000FFFFFFF) could often be detected in noise that may give a false detection, which again may give raised packet-error-rate.

Direct mode will be dependent on the software used in the MCU, but it is recommended to have the same restrictions on addresses for this mode.

เอกสารนี้ใช้เพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น ห้ามมิให้ทำซ้ำ ดัดแปลง เผยแพร่ หรือนำไปใช้

IMPORTANT TIMING DATA

The following timing applies for operation of nRF2401.

nRF2401 Timing Information

nRF2401 timing	Max.	Min.	Name
PWR_DWN → Configuration mode	3ms		Tpd2cfgm
PWR_DWN→ Active mode (RX/TX)	3ms		Tpd2a
ST_BY → TX ShockBurst™	195µs		Tsby2txSB
ST_BY → TX Direct Mode	202µs		Tsby2txDM
ST_BY → RX mode	202µs		Tsby2rx
Minimum delay from CS to data.		5µs	Tcs2data
Minimum delay from CE to data.		5µs	Tce2data
Minimum delay from DR1/2 to clk.		50ns	Tdr2clk
Maximum delay from clk to data.	50ns		Tclk2data
Delay between edges		50ns	Td
Setup time		500ns	Ts
Hold time		500ns	Th
Delay to finish internal GFSK data		1/data rate	Tfd
Minimum input clock high		500ns	Thmin
Set-up of data in Direct Mode	50ns		Tsdm
Minimum clock high in Direct Mode		300ns	Thdm
Minimum clock low in Direct Mode		230ns	Tldm
Time on air, TX Direct mode	4ms		ToaDM

Table 20 Operational timing of nRF2401

When the nRF2401 is in power down it must always settle in stand by for 3ms before it can enter configuration or one of the active modes.

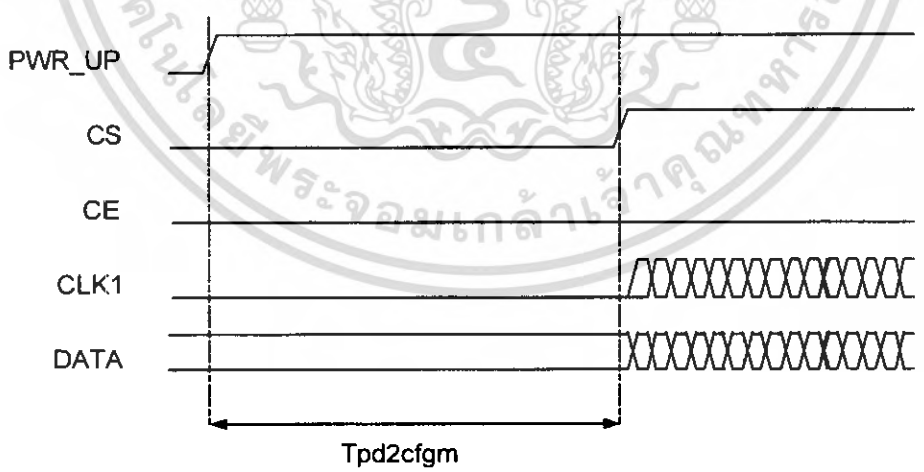


Figure 13 Timing diagram for power down (or VDD off) to configuration mode for nRF2401.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

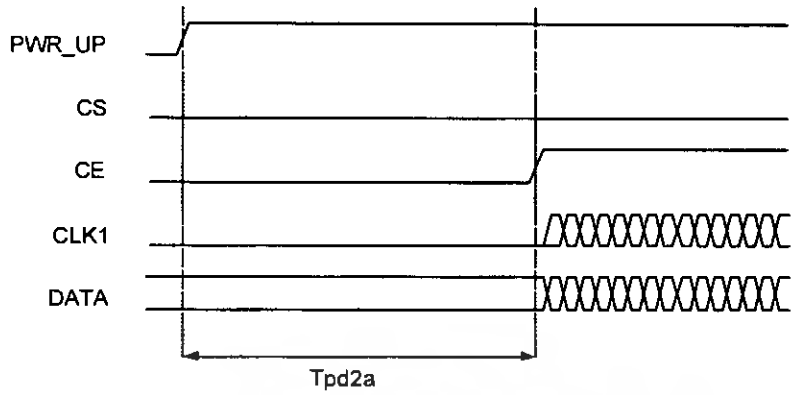


Figure 14 Power down (or VDD off) to active mode

Note that the configuration word will be lost when VDD is turned off and that the device then must be configured before going to one of the active modes. If the device is configured one can go directly from power down to the wanted active mode.

Note:

CE and CS may not be high at the same time. Setting one or the other decides whether configuration or active mode is entered.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Configuration mode timing

When one or more of the bits in the configuration word needs to be changed the following timing apply.

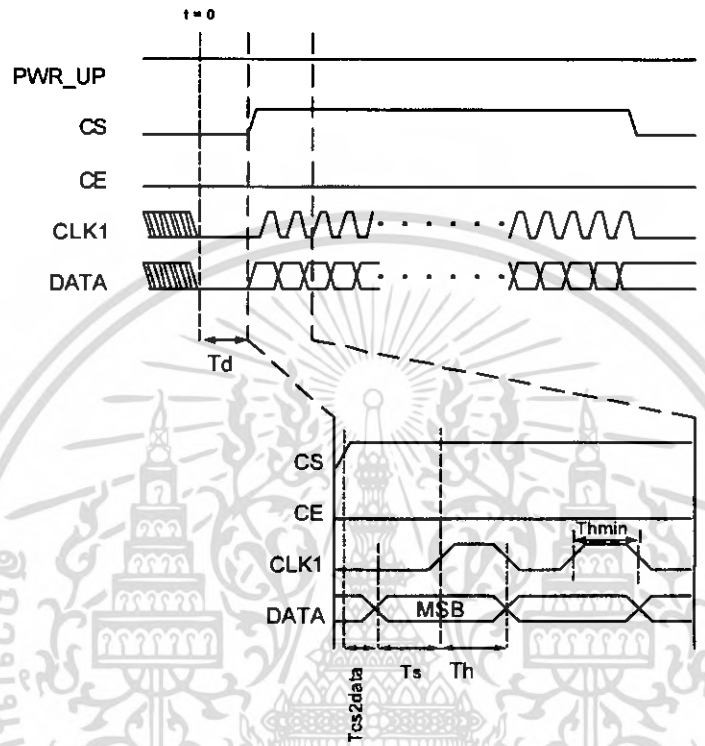


Figure 15 Timing diagram for configuration of nRF2401

If configuration mode is entered from power down, CS can be set high after Tpd2sby as shown in Figure 13.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

ShockBurst™ Mode timing

ShockBurst™ TX:

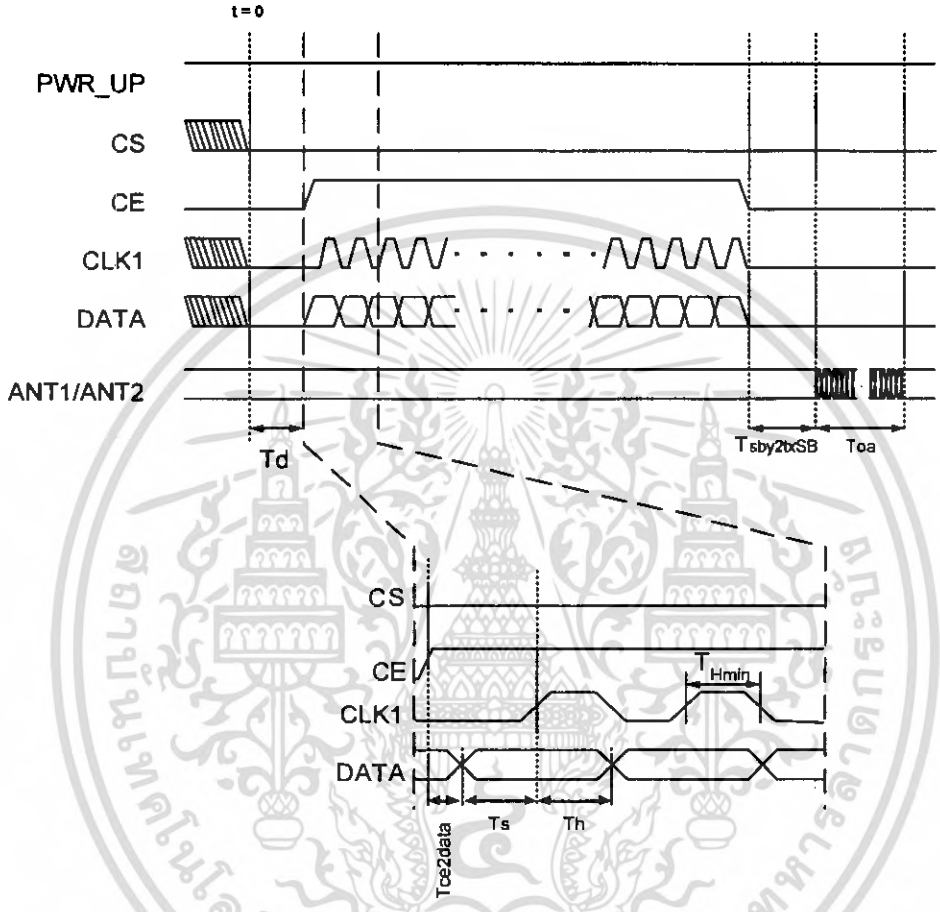


Figure 16 Timing of ShockBurst™ in TX

The package length and the data rate give the delay Toa (time on air), as shown in the equation.

$$T_{OA} = 1/datarate \cdot (\#databits + 1)$$

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

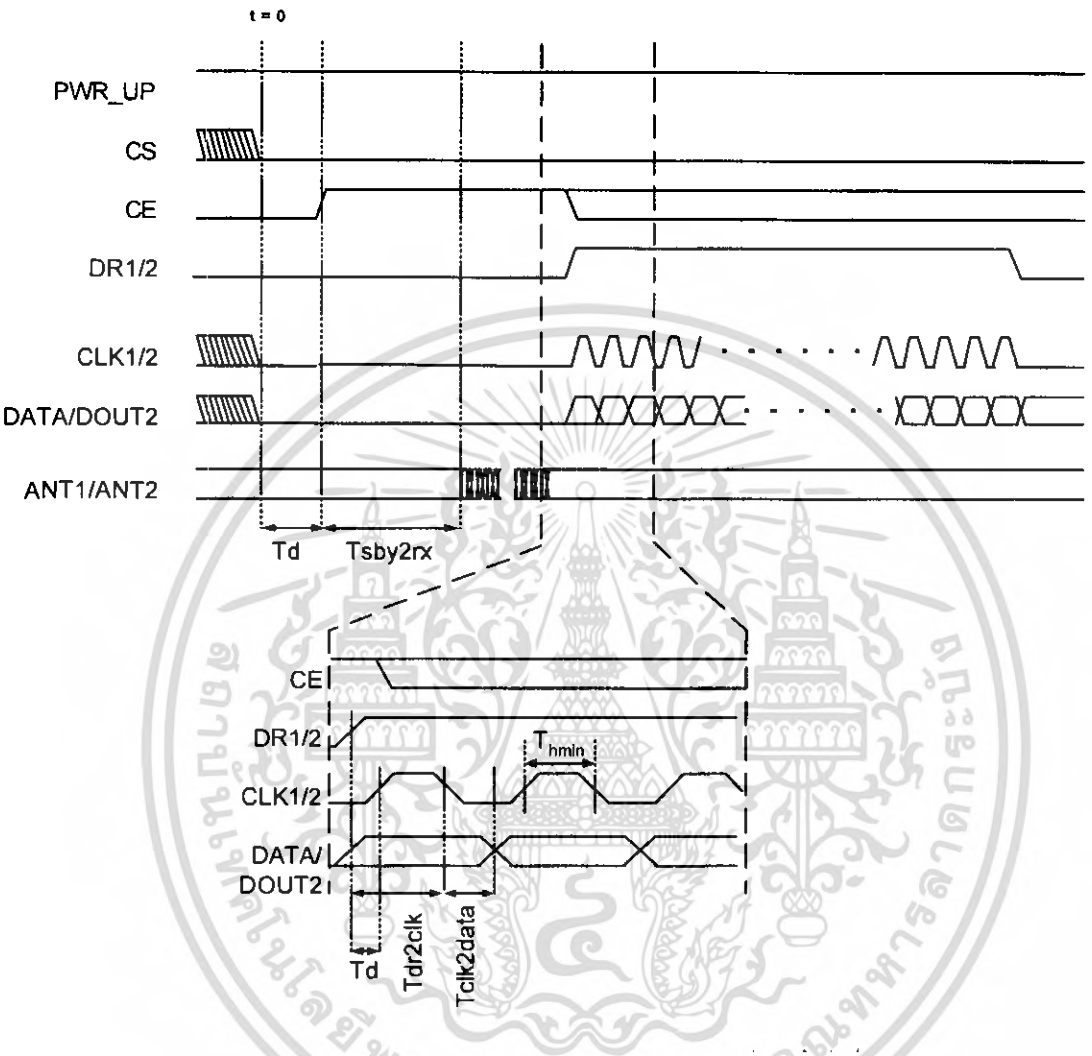


Figure 17 Timing of ShockBurst™ in RX

The CE may be kept high during downloading of data, but the cost is higher current consumption (18mA) and the benefit is short start-up time (200µs) when DR1 goes low.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Direct Mode

Direct Mode TX:

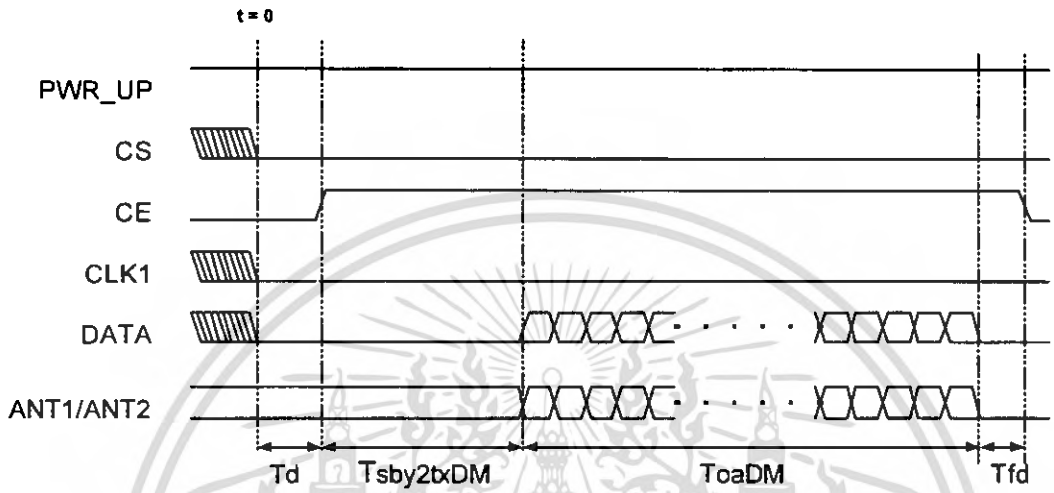


Figure 18 Timing of direct mode TX

In TX direct mode the input data will be sampled by nRF2401 and therefore no clock is needed. The clock must be stable at low level during transmission due to noise considerations. The exact delay $T_{sby2txDM}$ is given by the equation:

$$T_{sby2txDM} = 194us + 1/F_{XO} \cdot 14 + 2.25us$$

The maximum length of a package (ToaDM) over all voltages and temperatures is 4ms. This is limited by frequency drift in the transmitter and is independent of data rate and frequency channel.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า

Direct Mode RX:

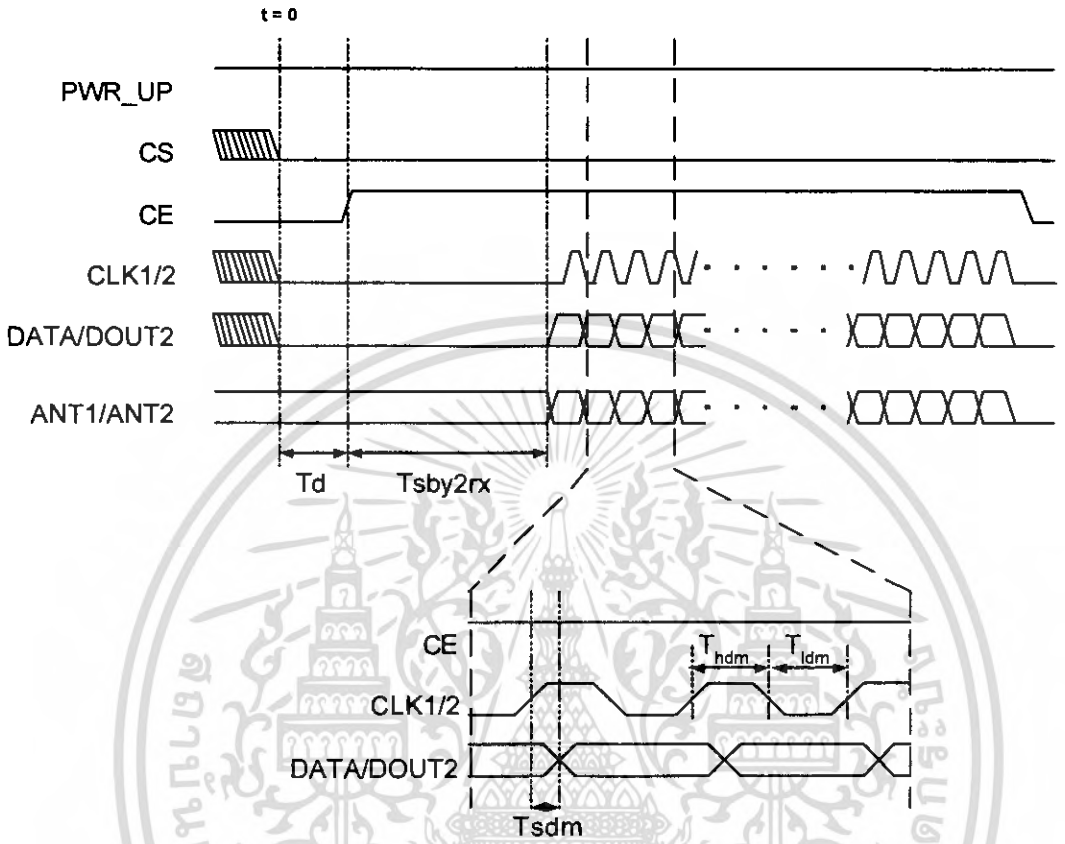


Figure 19 Timing of direct mode RX

Tsby2rx describes the delay from the positive edge of CE to start detection of (demodulating) incoming data.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า