สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

# การเก็บข้อมูลไร้สายด้วยเทคโนโลยี MIWI

# MIWI Technology Data Logging

T119219

นายวัชร แซ่โค้ว
นายสิทธิพร ปลื้มยุทธ์
นายเสกสรร ธูปเทียนทอง

เลขหมู่.....
เลขทะเบียน 119219
วัน,เดือน,ปี 6 ธ.ค. 2554

b.....
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิศวกรรมระบบควบคุม

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2553

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# MIWI Technology Data Logging

Mr. Watchara Sae-Khow

Mr. Sittiporn Plumyut

Mr. Seksun Thubtientong

THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT

OF THE REQUIREMENTS FOR THE DEGREE OF

BACHELOR OF ENGNEERING IN CONTROL ENGINEERING

FACULTY OF ENGINEERING

KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG

ACADEMIC YEAR 2010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**ปริญญานิพนธ์ปีการศึกษา 2553**

สาขาวิชาวิศวกรรมการวัดและควบคุม คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การเก็บข้อมูลไร้สายด้วยเทคโนโลยี MIWI

MIWI Technology Data Logging


ผู้จัดทำ นายวัชร แซ่โค้ว 50011425

นายสิทธิพร ปลื้มยุทธ์ 50011678

นายเสกสรร ธูปเทียนทอง 50011788

.....สังวาล บกสุวรรณ.....อาจารย์ที่ปรึกษา

(อาจารย์ สังวาล บกสุวรรณ)


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# การเก็บข้อมูลไร้สายด้วยเทคโนโลยี MIWI

โดย

นายวัชร แซ่โค้ว 50011425

นายสิทธิพร ปลื้มยุทธ์ 50011678

นายเสกสรร ธูปเทียนทอง 50011788

อาจารย์ที่ปรึกษา

อาจารย์สังวาล บกสุวรรณ

ปีการศึกษา 2553

## บทคัดย่อ

ปริญญานิพนธ์ฉบับนี้ เป็นการนำเสนอการรับส่งผ่านข้อมูลไร้สายระยะใกล้แบบ MIWI ตามแบบมาตรฐานการรับส่งข้อมูลแบบ IEEE 802.15.4 ซึ่งสามารถนำมาประยุกต์ใช้งานร่วมกับไมโครคอนโทรลเลอร์ให้สามารถรับส่งผ่านข้อมูลไร้สายได้ ซึ่งเป้าหมายที่สำคัญของโครงงานนี้เพื่อนำความรู้ที่ได้นี้ไปประยุกต์ใช้ในการออกแบบการทดลองต่างๆที่สนใจให้สามารถรับส่งผ่านข้อมูลไร้สายได้ภายในห้องทดลอง

ขั้นตอนการดำเนินการ เริ่มจากการเรียนรู้หลักการทำงานในการส่งผ่านข้อมูลชองชิพ MIWI เบื้องต้นว่าเป็นการรับส่งข้อมูลแบบไหน ต่อมาทำการศึกษา สถาปัตยกรรมโครงสร้างภายในและการทำงานภายในตัวไมโครคอนโทรลเลอร์เพื่อสามารถเลือกใช้ไมโครคอนโทรลเลอร์มาควบคุมการทำงานชิพMIWI ได้อย่างมีประสิทธิภาพ ต่อมาเป็นการศึกษาและออกแบบวงจรอิเล็กทรอนิกส์เพื่อสร้างบอร์ดทดลองไมโครคอนโทรลเลอร์มาใช้งานในการควบคุมการทำงานชิพMIWI แล้วจึงเขียนโปรแกรมคอมพิวเตอร์โดยใช้ภาษาซี เพื่อควบคุมการทำงานของชิพMIWI ให้สามารถรับส่งผ่านข้อมูลไร้สายแบบ MIWI ได้ และทำการทดลองการส่งผ่านค่าอุณหภูมิ จากการทดลองพบว่าสามารถส่งผ่านค่าอุณหภูมิด้วยเทคโนโลยี MIWI ไปแสดงผลบนคอมพิวเตอร์ได้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# MIWI Technology Data Logging

By

Mr. Watchara Sae-Khow 50011425

Mr. Sittiporn Plumyut 50011678

Mr. Seksun Thubtientong 50011788

Advisor

Mr. Sungwan Boksuwan

Acedemic Year 2010

## ABSTRACT

This thesis presents the implementation of wireless data logging using MIWI technology, which is the IEEE 802.15.4 standard. The MIWI module can communicate with a microcontroller through the SPI interface which allows to send and to receive data wirelessly. The aim of making this project is to study the wireless data logging using MIWI technology in order to apply the gained knowledge for setting up laboratories.

This project has been conducted as in the following steps. First, the principle of MIWI technology is focused including Application Program Interface (API). Second, the architecture of microcontrollers and programming techniques are deeply studied, especially multi-tasking programming. After that, the transmitter and receiver board consisting mainly of the microcontroller and MIWI module are designed and the operating programs for running the project are composed. Finally, the first version of the wireless data logging boards is responsible for sensing the surrounding temperature and broadcasting the information to the computer wirelessly.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# กิตติกรรมประกาศ

การจัดทำปริญญานิพนธ์ฉบับนี้ สามารถสำเร็จลุล่วงไปได้ด้วยดี เพราะได้รับความช่วยเหลือเป็นอย่างดี จาก อาจารย์สังวาล บกสุวรรณ ที่ได้กรุณาให้คำปรึกษาแนะนำที่ดีมาโดยตลอดตั้งแต่ต้น รวมทั้งเอื้อเฟื้ออุปกรณ์ที่จำเป็น และความช่วยเหลืออื่นๆที่เป็นประโยชน์ต่อโครงงาน ผู้จัดทำรู้สึกซาบซึ้งและขอกราบขอบพระคุณอย่างสูง

ขอขอบพระคุณ ผศ.ดร. ถาวร เบญจนราสุทธิ์ ที่คอยถามถึงความคืบหน้าตลอดเวลา และ นายอมรเทพ กาพย์แก้ว ที่ให้ความช่วยเหลือด้านอุปกรณ์การทดลอง

ขอบคุณเพื่อนๆทุกคนที่ให้กำลังใจ สนับสนุนอุปกรณ์ที่ขาดเหลือ กระตุ้นเตือน รวมทั้งคอยถามไถ่ความคืบหน้าของโครงงานอยู่เสมอ

สุดท้ายนี้ผู้จัดทำขอกราบขอบพระคุณบิดา มารดา และครอบครัว ที่คอยเป็นกำลังใจที่ดีตลอดมา รวมถึงการสนับสนุนในเรื่องของงบประมาณที่ขาดเหลือ ตลอดจนเป็นแรงบันดาลใจที่ดีที่สุดที่ทำให้โครงงานนี้สำเร็จสมบูรณ์ลงได้

คณะผู้จัดทำ

นายวัชร แซ่โค้ว

นายสิทธิพร ปลื้มยุทธ์

นายเสกสรร ธูปเทียนทอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ


| | หน้า |
|---|---|
| **บทคัดย่อ** | **I** |
| **บทคัดย่อภาษาอังกฤษ** | **II** |
| **กิตติกรรมประกาศ** | **III** |
| **สารบัญ** | **IV** |
| **สารบัญภาพ** | **VII** |
| **สารบัญตาราง** | **IX** |
| **บทที่ 1 บทนำ** | **1** |
| 1.1 กล่าวนำ | 1 |
| 1.2 วัตถุประสงค์ในการทำปริญญานิพนธ์ | 2 |
| 1.3 ขั้นตอนการศึกษาและขอบเขตของโครงงาน | 2 |
| 1.4 รายละเอียดของปริญญานิพนธ์ | 3 |
| **บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้อง** | **4** |
| 2.1 MIWI | 4 |
| 2.1.1 โครงสร้างตำแหน่งขาของชิพ MIWI | 5 |
| 2.1.2 หลักการทำงานและการเชื่อมต่อใช้งานชิพMIWI | 5 |
| 2.2 ไมโครคอนโทรลเลอร์ตระกูล PIC24FJ128GA010 | 6 |
| 2.2.1 คุณสมบัติไมโครคอนโทรลเลอร์ PIC24F128GA010 | 7 |
| 2.2.2 สถาปัตยกรรมโครงสร้างภายใน PIC24FJ128GA010 | 7 |
| 2.2.3 การใช้งานโมดูลในไมโครคอนโทรลเลอร์ PIC24FJ128GA010 | 8 |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ(ต่อ)


| | หน้า |
|---|---|
| 2.2.3.1 การใช้งานพอร์ต I/O | 8 |
| 2.2.3.2 Timer Module | 9 |
| 2.2.3.3 SPI Module | 12 |
| 2.2.3.4 UART Module | 16 |
| 2.3 การเขียนโปรแกรมควบคุมการทำงานชิพ MIWI | 19 |
| 2.3.1 MIWI Connectivity Made Simple | 19 |
| 2.3.2 MIWI P2P Wireless Protocol Stack | 20 |
| **บทที่ 3 การคำนวณและการสร้าง** | **22** |
| 3.1 บอร์ดรับส่งข้อมูลไร้สายโดยผ่านโมดูล MIWI | 23 |
| 3.1.1 สิ่งที่จำเป็นในการรับส่งข้อมูลไร้สายแบบ MIWI | 23 |
| 3.1.2 คุณสมบัติของโมดูลรับส่งข้อมูลแบบ MIWI และไมโครคอนโทรลเลอร์ | 23 |
| 3.2 จอ LCD แสดงสถานะการทำงาน | 24 |
| 3.3 โครงสร้างทางด้านฮาร์ดแวร์ | 24 |
| **บทที่ 4 การทดลองและผลการทดลอง** | **27** |
| 4.1 การทดลองรับส่งข้อมูลจากบอร์ดทดลองที่ 1 ไปยังบอร์ดทดลองที่ 2 | 27 |
| 4.1.1 ทดสอบระยะในการรับส่งข้อมูลของบอร์ดทดลอง | 27 |
| 4.1.2 ทดสอบการแสดงผลของข้อมูลผ่านทางจอคอมพิวเตอร์ | 28 |
| 4.1.3 ทดสอบการอ่านค่าอุณหภูมิ | 30 |


เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญ(ต่อ)


| | หน้า |
|---|---|
| **บทที่ 5 บทวิจารณ์และสรุปผล** | **32** |
| 5.1 สรุปผลการทดลอง | 32 |
| 5.2 ปัญหาและแนวทางแก้ไข | 32 |
| 5.3 ข้อเสนอแนะและแนวทางในการค้นคว้าพัฒนา | 33 |
| **ภาคผนวก ก โปรแกรมการประมวลผล** | **34** |
| ก.1โปรแกรมส่วนของการส่งข้อมูลอุณหภูมิ | 34 |
| ก.2โปรแกรมส่วนของการรับข้อมูลอุณหภูมิ | 46 |
| **ภาคผนวก ข เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์** | **55** |
| ข.1 เอกสารคู่มือการใช้งาน โมดูล MIWI | 55 |
| ข.2 เอกสารคู่มือการใช้งาน ไมโครคอนโทรลเลอร์ | 58 |
| **เอกสารอ้างอิง** | **68** |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญภาพ


| รูปที่ | หน้า |
|---|---|
| 1.1 บอร์ดรับส่งข้อมูลไร้สายแบบ MIWI | 1 |
| 2.1 MRF24J40MA | 4 |
| 2.2 Pin Diagram MRF24J40MA | 5 |
| 2.3 แสดงการเชื่อมต่อใช้งาน MRF24J40MA | 5 |
| 2.4 แสดงการส่งและรับข้อมูลระหว่างบอร์ด MIWI | 6 |
| 2.5 PIC24FJ128GA010 | 6 |
| 2.6 Pin Diagram PIC24FJ128GA010 | 7 |
| 2.7 สถาปัตยกรรมโครงสร้างภายในของ PIC24FJ128GA010 | 8 |
| 2.8 บล็อกไดอะแกรมการทำงานโมดูลTimer 1 | 10 |
| 2.9 Timer1 Control Register | 10 |
| 2.10 บล็อกไดอะแกรมการทำงานโมดูล SPI | 12 |
| 2.11 SPIx Control Register1 | 13 |
| 2.12 SPIx Status and Control Register | 13 |
| 2.13 การเชื่อมต่อสัญญาณในโหมดมาสเตอร์และสลาฟแบบ 3 สาย | 14 |
| 2.14 การเชื่อมต่อสัญญาณในโหมดมาสเตอร์และสลาฟแบบ 4 สาย | 14 |
| 2.15 UART Simplified Block Diagram | 16 |
| 2.16 UARTx Mode Register | 17 |
| 2.17 UARTx Status and Control Register | 17 |
| 2.18 MIWI DE | 19 |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญภาพ(ต่อ)


| รูปที่ | หน้า |
|---|---|
| 2.19 ConfigApp.h | 20 |
| 2.20 HardwareProfile.h | 21 |
| 3.1 โครงสร้างโดยรวมของบอร์ดรับส่งข้อมูลไร้สายแบบ MIWI | 22 |
| 3.2 โมดูลรับส่งข้อมูลไร้สาย MIWI และไมโครคอนโทรลเลอร์ | 23 |
| 3. 3 วงจรในการออกแบบส่วนที่ 1 | 25 |
| 3. 4 วงจรในการออกแบบส่วนที่ 2 | 26 |
| 4.1 บอร์ดที่ 1 แสดงสถานะว่าได้มีการเชื่อมต่อแล้ว | 28 |
| 4.2 แสดงการรับข้อมูลจากบอร์ดทดลองที่ 2 | 29 |
| 4.3 บอร์ดที่ 2 แสดงสถานะว่าได้มีการเชื่อมต่อแล้ว | 29 |
| 4.4 แสดงการรับข้อมูลจากบอร์ดทดลองที่ 1 ที่ส่งมา | 30 |
| 4.5 แสดงการรับค่าอุณหภูมิจากบอร์ดทดลองที่ 1 | 30 |
| 4.6 กราฟแสดงช่วงเวลาในการอ่านค่าอุณหภูมิ | 31 |



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# สารบัญตาราง


| ตารางที่ | หน้า |
|---|---|
| 4.1 ผลการรับส่งข้อมูลของบอร์ดรับส่งข้อมูลไร้สายโดยผ่านโมดูล MIWI | 27 |





เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

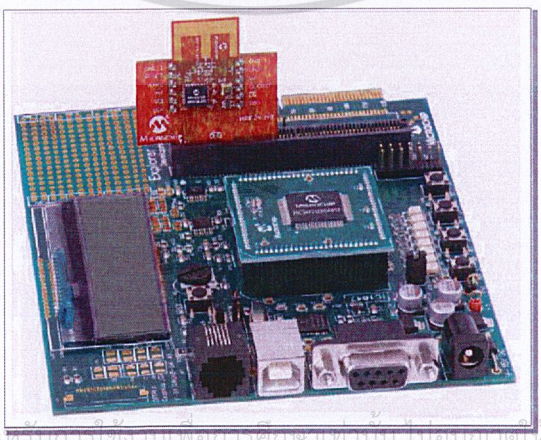
# บทที่ 1

# บทนำ

## 1.1 กล่าวนำ

การศึกษาในสาขาวิชาวิศวกรรมระบบควบคุม เป็นการศึกษาและประยุกต์ใช้ทฤษฎีในด้านต่างๆ เพื่อการออกแบบและควบคุมระบบให้มีเสถียรภาพ ซึ่งทำให้ระบบมีสมรรถนะตามต้องการหรือเป็นไปตามข้อกำหนดของงาน ดังนั้นเพื่อเป็นไปตามวัตถุประสงค์ที่ต้องการจำเป็นต้องมีการศึกษาไมโครคอนโทรลเลอร์ การศึกษาการออกแบบวงจรไมโครคอนโทรลเลอร์ การศึกษาการเขียนโปรแกรมคอมพิวเตอร์ (ภาษาซี) เพื่อควบคุมการทำงานไมโครคอนโทรลเลอร์ และการศึกษาการเลือกอุปกรณ์ต่างๆเพื่อสร้างบอร์ดทดลอง ตลอดจนบูรณาการเรื่องที่ศึกษาเหล่านี้มาประยุกต์ใช้กับโครงงานจริงทางกายภาพ ซึ่งนับเป็นหัวใจสำคัญในการศึกษา

ในโครงงานนี้เป็นการศึกษาการส่งผ่านข้อมูลแบบไร้สายระยะใกล้ MIWI ตามแบบมาตรฐานการรับส่งข้อมูล IEEE 802.15.4 ซึ่งนำมาประยุกต์ใช้งานร่วมกับไมโครคอนโทรลเลอร์ในการควบคุมการทำงานของชิพ MRF24J40MA ซึ่งเป็นชิพMIWI ที่ควบคุมการส่งผ่านข้อมูลแบบไร้สายระยะใกล้ให้สามารถส่งผ่านข้อมูลแบบไร้สายได้ เป้าหมายที่สำคัญของโครงงานนี้ เพื่อนำความรู้ข้อมูลนี้ไปประยุกต์ใช้ในออกแบบการทดลองต่างๆให้สามารถส่งผ่านข้อมูลแบบไร้สายได้ ภายในห้องทดลอง เพื่อประโยชน์ในการศึกษาเรียนรู้ในสาขาวิชาวิศวกรรมระบบควบคุมให้ก้าวหน้าทันสมัยยิ่งขึ้น



**รูปที่ 1.1** บอร์ดรับส่งข้อมูลไร้สายแบบ MIWI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.2 วัตถุประสงค์ในการทำปริญญานิพนธ์

1. เพื่อศึกษาหลักการส่งผ่านข้อมูลไร้สายระยะใกล้แบบ MIWI ตามแบบมาตรฐานการรับส่งข้อมูล IEEE 802.15.4
2. เพื่อศึกษาสถาปัตยกรรมและการทำงานภายใน Microcontroller และสามารถเลือกใช้ Microcontroller มาประมวลผลและควบคุมชิพMIWI ให้ทำงานได้อย่างมีประสิทธิภาพ
3. เพื่อฝึกทักษะการใช้งานโปรแกรมต่างๆเช่น Altium Designer, MPLAB IDE (C30 Compiler ให้สามารถออกแบบสร้างบอร์ดทดลองและเขียนโปรแกรมลงบน Microcontroller เพื่อใช้ควบคุมการส่งผ่านข้อมูลไร้สายระยะใกล้แบบ MIWI ให้สามารถใช้งานได้
4. เพื่อเรียนรู้การทำงานเป็นทีมเป็นระบบ และรู้จักการวิเคราะห์แก้ปัญหาได้ตรงจุด

## 1.3 ขั้นตอนการศึกษาและขอบเขตของโครงงาน

โครงงานการส่งผ่านข้อมูลไร้สายระยะใกล้แบบ MIWI จำเป็นต้องเรียนรู้หลักการทำงานในการส่งผ่านข้อมูลของชิพ MIWI เบื้องต้นว่าเป็นการรับส่งข้อมูลแบบไหน ต่อมาเราก็ทำการศึกษาสถาปัตยกรรมโครงสร้างและการทำงานภายในตัวไมโครคอนโทรลเลอร์เพื่อสามารถเลือกใช้ไมโครคอนโทรลเลอร์มาควบคุมการทำงานชิพMIWI (MRF24F40MA) และการทำงานในด้านอื่นๆได้อย่างมีประสิทธิภาพ จากนั้นทำการศึกษาโปรแกรมในด้านต่างๆเช่น Altium Designer และ MPLAB IDE ของบริษัท Microchip ซึ่งสามารถ Download ได้ฟรีจาก www.microchip.com ในรูปแบบ Student Versions ซึ่งเป็นเครื่องมือช่วยในการออกแบบสร้างวงจรและเขียนโปรแกรมภายในโครงงานให้สามารถใช้งานได้

หลังจากที่ทำการออกแบบวงจรได้แล้ว ก็ได้สร้างบอร์ดทดลองจริงและเขียนโปรแกรมสำหรับควบคุมการทำงานชิพMIWI (MRF24F40MA) ให้สามารถส่งผ่านข้อมูลไร้สายระยะใกล้แบบ MIWI ได้ และสามารถนำไปประยุกต์ใช้งานร่วมกับการทดลองต่างๆภายในห้องทดลองให้สามารถส่งผ่านข้อมูลแบบไร้สายได้ ในการกำหนดการทำงานของชิพMIWI (MRF24F40MA) เบื้องต้นและการทำงานในส่วนต่างๆของโครงงาน รายละเอียดจะกล่าวในบทต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.4 รายละเอียดของปริญญานิพนธ์

เนื้อหาที่จะกล่าวในปริญญานิพนธ์ฉบับนี้ประกอบด้วย

บทที่1 บทนำ กล่าวนำถึงวัตถุประสงค์ในการทำปริญญานิพนธ์ ขั้นตอนการศึกษาและการจัดทำโครงงาน พร้อมทั้งรายละเอียดของปริญญานิพนธ์ของแต่ละบท

บทที่ 2 ทฤษฎีและความรู้ที่เกี่ยวข้อง กล่าวถึงหลักการและทฤษฎีที่เกี่ยวข้องที่ใช้ในการศึกษาออกแบบบอร์ดทดลองรวมถึงวงจรอิเล็กทรอนิกส์ที่เกี่ยวข้อง การใช้งานพอร์ตขนานในการส่งข้อมูลต่างๆ และสามารถนำเอาความรู้ที่ศึกษาไปประยุกต์ใช้ในการจัดทำโครงงาน

บทที่ 3 หลักการออกแบบ เป็นการนำเสนอหลักการออกแบบและสร้างบอร์ดทดลองให้สามารถนำไปใช้งานได้จริงในโครงงาน

บทที่ 4 การทดลอง เป็นส่วนของการทดสอบองค์ประกอบต่างๆภายในโครงงาน ตลอดจนการทดลองระยะในการรับส่งข้อมูลไร้สายระยะใกล้แบบ MIWI

บทที่ 5 บทวิจารณ์และสรุป เป็นการสรุปผลการดำเนินงานภายในโครงงาน ปัญหาที่เกิดขึ้นและแนวทางการปรับปรุงพัฒนาโครงงานนี้ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 2

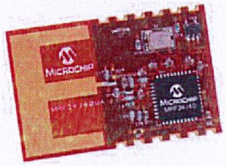
# ทฤษฎีและความรู้ที่เกี่ยวข้อง

จากที่ได้กล่าวมาในบทที่ 1 แล้วว่า ก่อนขั้นตอนการออกแบบบอร์ดและสร้างบอร์ดทดลองจริง ตลอดจนการเขียนโปรแกรมควบคุมการทำงานของไมโครคอนโทรลเลอร์ จำเป็นต้องศึกษาความรู้เพิ่มเติมในด้านต่างๆเพื่อเป็นองค์ความรู้พื้นฐานสำหรับการนำมาประยุกต์ใช้ในโครงงานให้ได้ผลตามเป้าหมาย ดังนั้น ในบทนี้จะศึกษาทฤษฎีและหลักการตลอดจนการอธิบายรายละเอียดขององค์ประกอบต่างๆที่เป็นความรู้พื้นฐานและนำมาประยุกต์ใช้ในโครงงานได้

## 2.1 MIWI

MRF24J40MA นับว่าเป็นอุปกรณ์ที่มีความสำคัญมากในโครงงานนี้ เพราะเป็นโมดูลที่สามารถทำให้เกิดการสื่อสารส่งผ่านข้อมูลไร้สายระยะใกล้แบบ MIWI ได้ และแสดงคุณสมบัติที่สำคัญเบื้องต้นของชิพMIWI (MRF24J40MA) ดังนี้

1. ทำงานที่แรงดัน 3.3 Volt
2. รับส่งข้อมูลตามแบบมาตรฐาน IEEE 802.15.4 แบบ Wireless PAN ความถี่ 2.4 GHz
3. อัตราการรับส่งข้อมูล 250 kbps ระยะทางในการรับส่งข้อมูล 400 ft.
4. ใช้งานได้กับ PIC16F, PIC18F, PIC24F/H, dsPIC33 และ PIC32
5. เชื่อมต่อการใช้งานแบบ SPI โมดูล

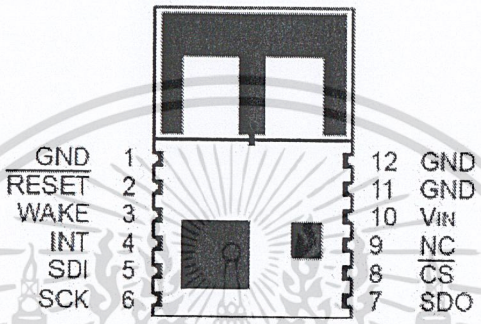


**รูปที่ 2.1** MRF24J40MA

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

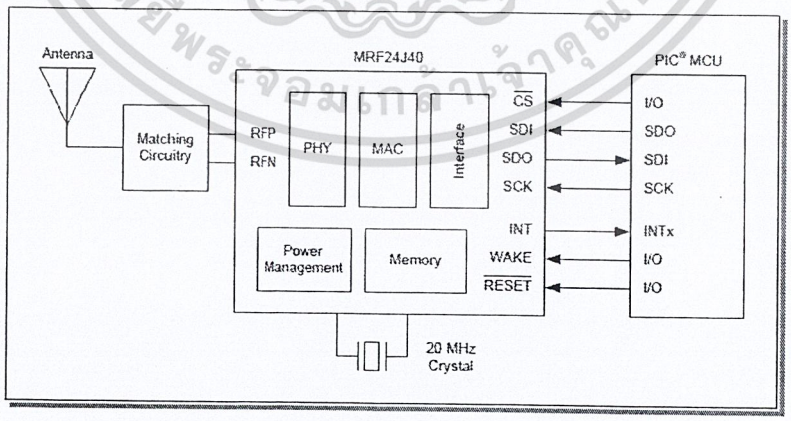
### 2.1.1 โครงสร้างตำแหน่งขาของชิพ MIWI

ชิพMIWI (MRF24J40MA) มีขาที่สามารถต่อใช้งานกับไมโครคอนโทรลเลอร์ได้ทั้งหมด 12 ขา และได้แสดงหน้าที่การทำงานของขาอุปกรณ์แต่ละขาบนชิพMIWI (MRF24J40MA) ซึ่งแสดงได้ดังรูป




**รูปที่ 2.2** Pin Diagram MRF24J40MA

### 2.1.2 หลักการทำงานและการเชื่อมต่อใช้งานชิพMIWI

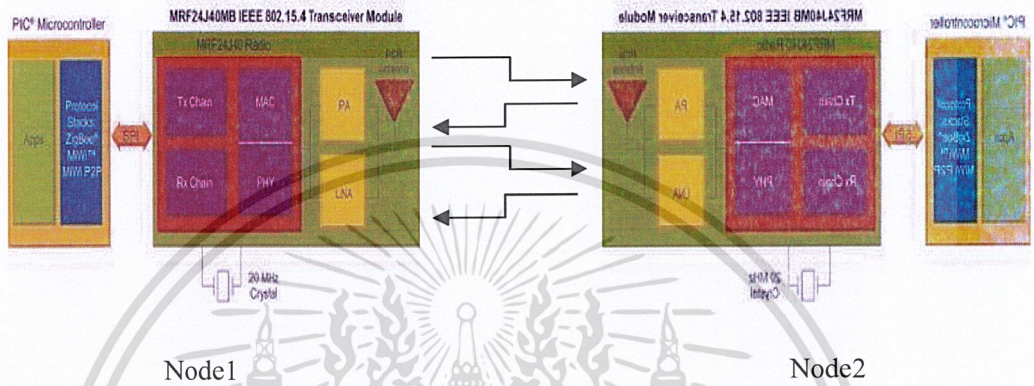



**รูปที่ 2.3** แสดงการเชื่อมต่อใช้งาน MRF24J40MA

จากรูปที่ 2.3 เป็นการเชื่อมต่อ MRF24J40MA ใช้งานร่วมกับไมโครคอนโทรลเลอร์ในโหมด SPI ซึ่งการทำงานของไมโครคอนโทรลเลอร์จะใช้โหมด SPI ในการควบคุมและสื่อสารกับชิพ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

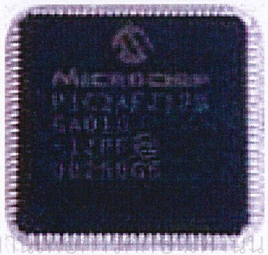
MIWI ให้ทำการส่งข้อมูลผ่านออกไปยังเสาอากาศ (Antenna) ของชิพMIWI (node 1) ไปยังอีกบอร์ด (node 2) ที่มีโครงสร้างการเชื่อมต่อของบอร์ดที่คล้ายกัน และในขณะเดียวกันบอร์ดที่ทำการส่งข้อมูลออกไปก็สามารถรับข้อมูลได้เช่นเดียวกัน ซึ่งแสดงได้ดังรูป 2.4




**รูปที่ 2.4** แสดงการส่งและรับข้อมูลระหว่างบอร์ด MIWI

## 2.2 ไมโครคอนโทรลเลอร์ตระกูล PIC24FJ128GA010

เมื่อทราบคุณสมบัติเบื้องต้นของชิพMIWI (MRF24J40MA) เช่น ใช้แรงดันไฟฟ้าเท่าไรในการทำงาน สามารถรับส่งข้อมูลไร้สายระยะใกล้แบบ MIWI ได้อย่างไรและเชื่อมต่อใช้งานกับไมโครคอนโทรลเลอร์ในโหมดต่างๆอย่างไร ในขั้นตอนนี้ก็จะทำการเลือกไมโครคอนโทรลเลอร์ใช้งานให้เหมาะสมกับชิพMIWI (MRF24J40MA) และตรงตามจุดประสงค์ของงานในด้านอื่นๆ ตามต้องการ และในตอนนี้ได้เลือกใช้ PIC24FJ128GA010 ซึ่งมีคุณสมบัติต่างๆตรงตามจุดประสงค์ของงาน โดยรายละเอียดจะกล่าวในหัวข้อย่อยถัดไป



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูปที่ 2.5** PIC24FJ128GA010

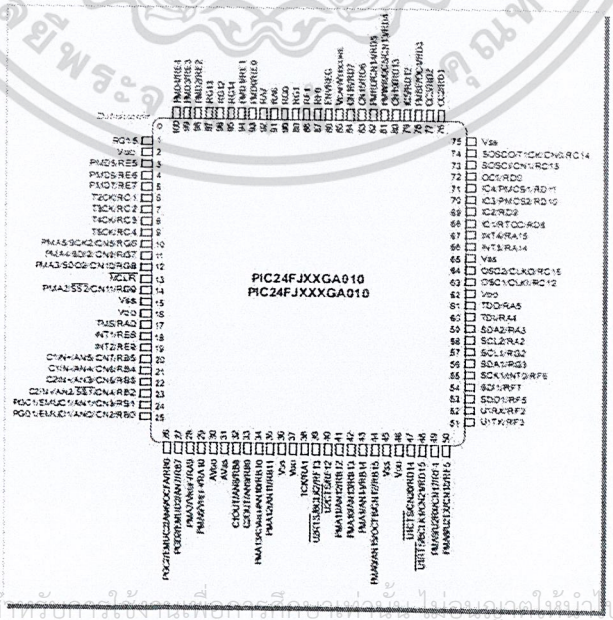
### 2.2.1 คุณสมบัติไมโครคอนโทรลเลอร์ PIC24F128GA010

แสดงคุณสมบัติที่สำคัญเบื้องต้นของ PIC24FJ128GA010 ดังนี้

1. เป็นไมโครคอนโทรลเลอร์ที่ใช้ CPU ขนาด 16 บิต
2. ทำงานสูงสุด 16 ล้านคำสั่งต่อวินาที ที่ความถี่ 32 MHz
3. ภายในมีหน่วยความจำแบบFlash สามารถลบแล้วเขียนใหม่ได้ 10,000 ครั้ง
4. เก็บข้อมูลได้นานสูงสุด 20 ปี
5. มีขาพอร์ตสำหรับติดต่อกับอุปกรณ์ภายนอก เช่น โมดูล SPI, UART, $I^2C$
6. มี Timers 16 bit ให้ใช้งานถึง 5 ตัว

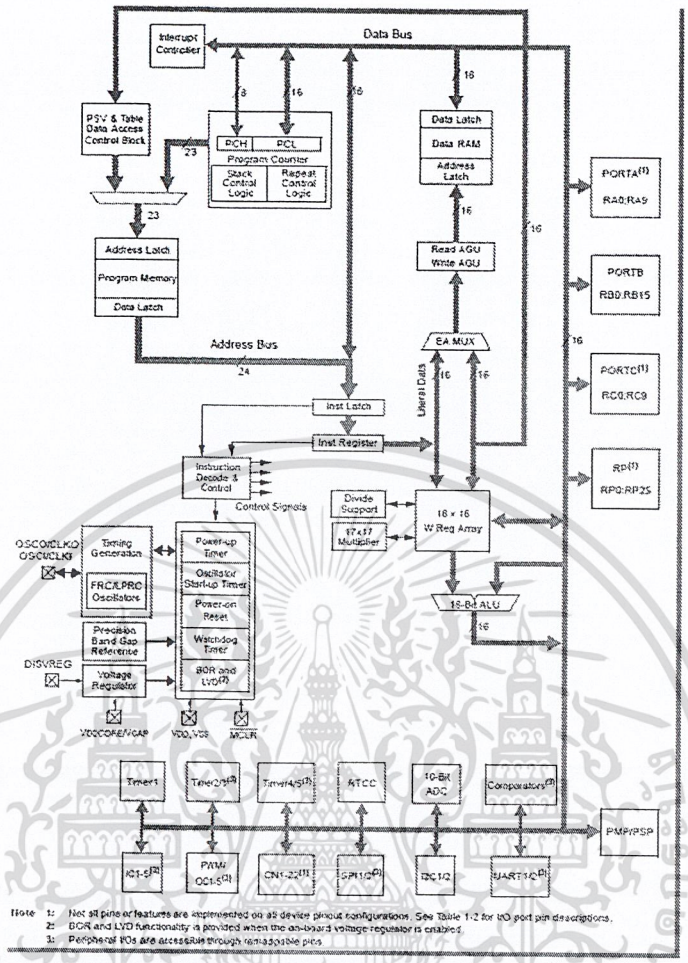
### 2.2.2 สถาปัตยกรรมโครงสร้างภายใน PIC24FJ128GA010

เป็นการแสดงสถาปัตยกรรมโครงสร้างภายในและขาใช้งานพื้นฐานไมโครคอนโทรลเลอร์ PIC24F128GA010 ดังรูปที่ 2.6 และ รูปที่ 2.7




**รูปที่ 2.6** Pin Diagram PIC24FJ128GA010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 2.7** สถาปัตยกรรมโครงสร้างภายในของ PIC24FJ128GA010

### 2.2.3 การใช้งานโมดูลในไมโครคอนโทรลเลอร์ PIC24FJ128GA010

#### 2.2.3.1 การใช้งานพอร์ต I/O

การใช้งานพอร์ต I/O ภายในขาพอร์ตประกอบด้วยรีจิสเตอร์ 3 ตัวที่ควบคุมการทำงานของขาพอร์ต คือ

**1. รีจิสเตอร์ TRIS (Data Direction register)**

รีจิสเตอร์ TRISx ใช้ในการกำหนดทิศทางการทำงานของขาพอร์ตให้เป็นอินพุตหรือเอาต์พุตพอร์ต หากกำหนดเป็น 1 ในบิตใดของรีจีสเตอร์ TRISx จะทำให้บิตที่ตำแหน่งเดียวกันของรีจิสเตอร์ PORTx และ LATx เป็นอินพุต และหากเป็น 0 จะเป็นเอาต์พุต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น และห้ามมิให้ดัดแปลงเนื้อหา หรือนำไปเผยแพร่ต่อ ทั้งนี้ผู้ใช้งานจะต้องอ้างอิงถึงเอกสารทุกครั้งที่มีการนำไปใช้

**2. รีจีสเตอร์ PORT (I/O Port register)**

รีจีสเตอร์ PORTx ให้ในการอ่านเขียนข้อมูลกับขาพอร์ต การอ่านข้อมูลด้วยรีจีสเตอร์ PORTx จะเป็นการอ่านข้อมูลจากพอร์ตโดยตรง (I/O pin) และหากเป็นการเขียนข้อมูลด้วยรีจิสเตอร์ PORTx จะเป็นการเขียนข้อมูลไปที่แลตซ์พอร์ตข้อมูล (port data latch) รีจีสเตอร์ PORTx เช่น PORTA, PORTB, PORTC, PORTD

**3. รีจีสเตอร์ LAT (I/O Latch register)**

รีจีสเตอร์ LATx ถูกนำมาใช้แก้ไขปัญหาของรีจีสเตอร์ PORTx ในกระบวนการ อ่าน – แก้ไข – เขียนข้อมูลแบบต่อเนื่องกระชั้นชิด โดยการอ่านข้อมูลด้วยรีจีสเตอร์ LATx จะเป็นการอ่านข้อมูลที่ค้างอยู่ในแลตซ์พอร์ตเอาต์พุต (Port Output Latches) และการเขียนข้อมูลจะเป็นไปในรูปแบบเดียวกับรีจีสเตอร์ PORTx คือ เขียนข้อมูลไปที่แลตซ์พอร์ตข้อมูล (Port Data Latch) ด้วยวิธีนี้จะไม่ทำให้เกิดความผิดพลาด

สรุปความแตกต่างระหว่างรีจีสเตอร์ PORT และ LAT มีรายละเอียดดังนี้

1. การเขียนข้อมูลไปที่รีจีสเตอร์ PORTx จะเขียนข้อมูลไปที่แลตซ์พอร์ต
2. การเขียนข้อมูลไปที่รีจีสเตอร์ LATx จะเขียนข้อมูลไปที่แลตซ์พอร์ต
3. การอ่านข้อมูลจากรีจีสเตอร์ PORTx จะเป็นการอ่านข้อมูลจาก I/O พอร์ตโดยตรง
4. การอ่านข้อมูลจากรีจีสเตอร์ LATx จะเป็นการอ่านข้อมูลที่ค้างอยู่ในแลตซ์พอร์ต

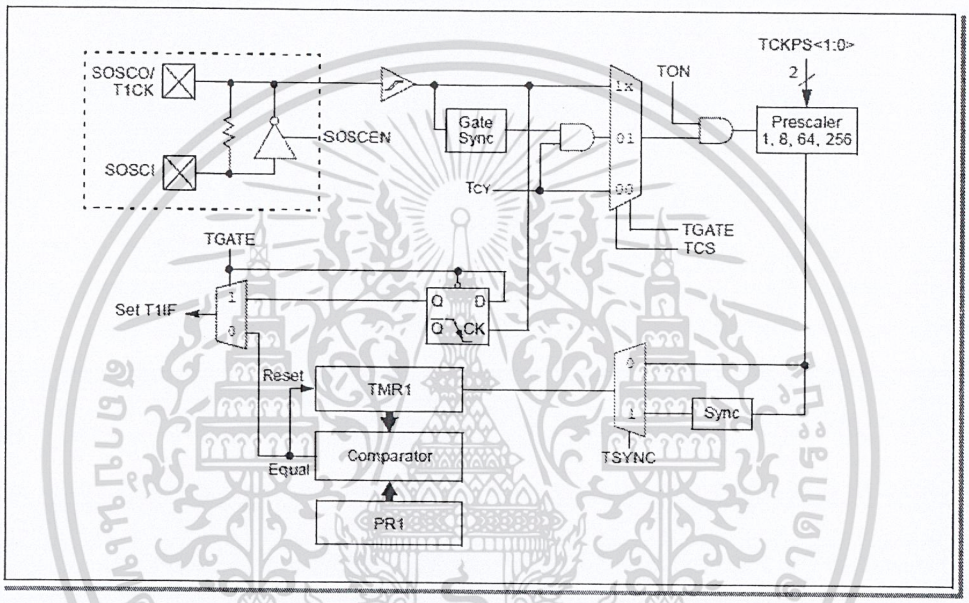
**2.2.3.2 Timer Module**

การทำงานของTimer จะเป็นตัวนับจับเวลา Timer หรือตัวนับ Counter ขึ้นอยู่กับการนับสัญญาณ หากเป็นการนับสัญญาณนาฬิกาจากภายในตัวไมโครคอนโทรลเลอร์ เราจะเรียกว่าตัวจับเวลา “Timer” และหากเป็นการนับสัญญาณนาฬิกาจากภายนอกจะเรียกว่าตัวนับ “Counter” ซึ่งโมดูล Timer ใน PIC16FJ128GA010 มีทั้งหมด 5 โมดูลสามารถแบ่งการทำงานของTimerในรูปแบบของฐานเวลาได้ 3 รูปแบบในตอนนี้จะขอกล่าวถึงเฉพาะ Timer1 ส่วน Timer เบอร์อื่นๆ สามารถศึกษาใช้งานเพิ่มเติมได้จาก Datasheet PIC24FJ128GA010

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**1. การใช้งานโมดูล Timer1**

Timer1 เป็น Timer ฐานเวลาแบบ A ขนาด 16 Bit สามารถใช้งานเป็นฐานเวลานาฬิกาจริงหรือ Real-Time Clock ได้และสามารถทำงานได้ถึง 3 Mode คือ 16-Bit Timer, 16-Bit Synchronous Counter และ 16-Bit Asynchronous Counter และได้แสดงบล็อกไดอะแกรมการทำงานของTimer ฐานเวลาแบบ A ดังรูป




**รูปที่ 2.8** บล็อกไดอะแกรมการทำงานโมดูลTimer 1

**2. รีจีสเตอร์ควบคุมการทำงานของ Timer1**

เป็นการแสดงรายละเอียดรีจีสเตอร์ที่ควบคุมการทำงานของ Timer ฐานเวลาแบบ A หรือ Timer1 ซึ่งมีรีจิสเตอร์ T1CON ทำงานอยู่และสามารถอ้างถึงการทำงานใน Bit ต่างๆได้ดังรูป

| R/W-0 | U-0 | R/W-0 | U-0 | U-0 | U-0 | U-0 | U-0 |
|---|---|---|---|---|---|---|---|
| TON | — | TSIDL | — | — | — | — | — |
| bit 15 | | | | | | | bit 8 |

| U-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 | U-0 |
|---|---|---|---|---|---|---|---|
| — | TGATE | TCKPS1 | TCKPS0 | — | TSYNC | TCS | — |
| bit 7 | | | | | | | bit 0 |

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูปที่ 2.9** Timer1 Control Register

**3. การคำนวณค่าเวลานับของ Timer**

การใช้งาน Timer จะเกี่ยวข้องกับการคำนวณเวลาในการนับแต่ละครั้งของ Timer หรือการเพิ่มค่ารีจีสเตอร์ TMRx ที่เกี่ยวข้องกับ Timer เช่น Timer1 จะเพิ่มค่าการนับในรีจีสเตอร์ TMR1 เป็นต้น นอกจากนี้ยังมีตัวหารหรือปรีสเกลเลอร์ (Prescaler) และตัวคูณหรือ PLL (Phase Locked Loop) ที่เกี่ยวข้องกับ Timer เพิ่มเติม สามารถหาค่าการนับแต่ละครั้งได้ด้วยสมการต่อไปนี้

จาก $T = 1 / f$ โดย $f = Fosc / 4$

$= 1 / (Fosc / 4)$

$T = 4 / Fosc$

โดยที่ T คือ เวลาในการนับแต่ละครั้งหรือการเพิ่มค่ารีจีสเตอร์ TMRx ทีละหนึ่ง

Fosc คือ ความถี่ออสซิเลเตอร์ที่ใช้

เมื่อมีการใช้ตัวหารและตัวคูณร่วมด้วยจะได้สมการดังนี้

$$\mathbf{T = (4 \times Prescaler) / (Fosc \times PLLx)}$$

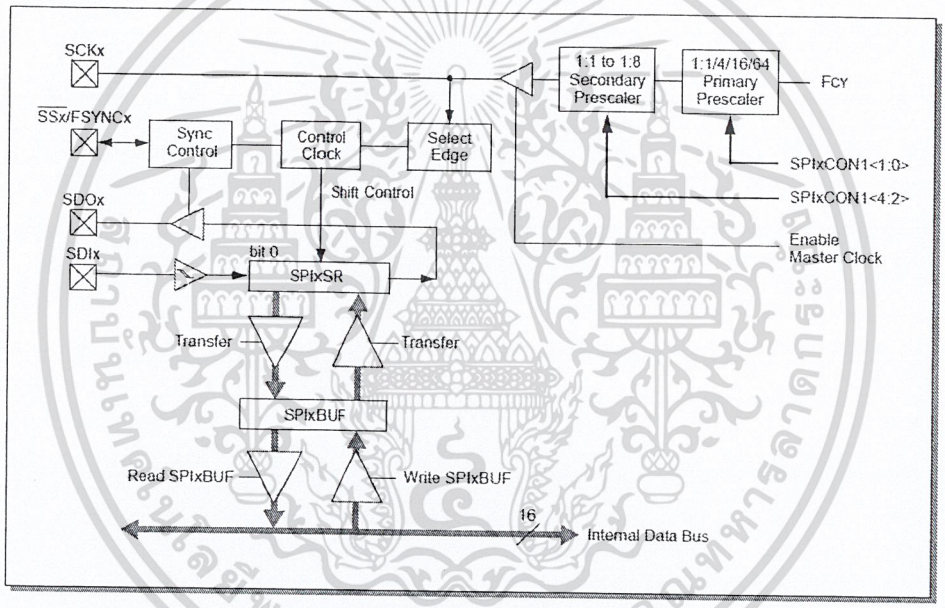
**2.2.3.3 SPI Module**

โมดูลเชื่อมต่ออุปกรณ์อนุกรม (Serial Peripheral Interface) หรือ SPI มีทั้งหมด 2 โมดูล เป็นโมดูลเชื่อมต่ออุปกรณ์ภายนอกแบบซิงโครนัส เพื่อใช้ในการสื่อสารกับอุปกรณ์ต่างๆหรือกับไมโครคอนโทรเลอร์ด้วยกัน เช่น การเชื่อมต่อกับหน่วยความจำอีอีพรอมแบบอนุกรมไอซีชิพรีจิสเตอร์ อุปกรณ์แสดงผล อุปกรณ์แปลงสัญญาณอนาลอกเป็นดิจิตอล โดยอุปกรณ์เหล่านี้ต้องสามารถทำงานเข้ากันได้กับ SPI และการเชื่อมต่อแบบ SIOP (Simple Synchronous Serial I/O Port)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**1. การใช้งานโมดูล SPI**

การทำงานของโมดูล SPI จะอาศัยรีจิสเตอร์ SPIxSR (x คือ 1 หรือ 2) โดยเป็นชิพรีจิสเตอร์ขนาด 16 บิต ทั้งในการรับและการส่งข้อมูล พร้อมกับบัฟเฟอร์รีจิสเตอร์ SPIxBUF (SPIxBuffer) เป็นพื้นที่แอดเดรสที่ใช้งานในการเก็บข้อมูลระหว่างการรับและการส่งข้อมูล โดยแอดเดรสดังกล่าวจะใช้งานร่วมกับรีจิสเตอร์ SPIxTXB และ SPIxRXB และได้แสดงบล็อกไดอะแกรมการทำงานโมดูล SPI ดังรูป




**รูปที่ 2.10** บล็อกไดอะแกรมการทำงานโมดูล SPI

การต่อการใช้งานโมดูล SPI จะใช้ขาพอร์ต 4 ขาในการเชื่อมต่อกับอุปกรณ์ได้ดังนี้

1. **SDIx** (Serial data input) ขาสำหรับรับข้อมูลอนุกรม
2. **SDOx** (Serial data output) ขาสำหรับส่งข้อมูลอนุกรม
3. **SCKx** (Shift clock input or output) ขาสำหรับสัญญาณนาฬิกาอนุกรม
4. **SSx** (Slave select) ขาสำหรับเลือกอุปกรณ์สเลฟ ทำงานที่ลอจิก 0 หรือเป็นขาอินพุตเอาต์พุตสัญญาณพัลส์แบบซิงโครนัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**2. รีจีสเตอร์ควบคุมการทำงานโมดูล SPI**

ในการกำหนดคุณสมบัติการทำงานของโมดูล SPI จะถูกกำหนดผ่านรีจิสเตอร์ SPIxCON (SPIx Control Register) และสถานะ การทำงานจะถูกกำหนดผ่านรีจิสเตอร์ SPIxSTAT (SPIs Status Register) ซึ่งแสดงBits ต่างๆที่ใช้กำหนดสถานะ การทำงานของทั้ง 2 Register ดังรูป

| U-0 | U-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| — | — | — | DISSCK | DISSDO | MODE16 | SMP | CKE(1) |
| bit 15 | | | | | | | bit 8 |

| R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| SSEN | CKP | MSTEN | SPRE2 | SPRE1 | SPRE0 | PPRE1 | PPRE0 |
| bit 7 | | | | | | | bit 0 |

| Legend: | | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

**รูปที่ 2.11** SPIx Control Register1

| R/W-0 | U-0 | R/W-0 | U-0 | U-0 | R-0 | R-0 | R-0 |
|---|---|---|---|---|---|---|---|
| SPIEN | — | SPISIDL | — | — | SPIBEC2 | SPIBEC1 | SPIBEC0 |
| bit 15 | | | | | | | bit 8 |

| R/W-0 | R/C-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R-0 | R-0 |
|---|---|---|---|---|---|---|---|
| SRMPT | SPIROV | SRXMPT | SISEL2 | SISEL1 | SISEL0 | SPITBF | SPIREF |
| bit 7 | | | | | | | bit 0 |

| Legend: | C = Clearable bit | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

**รูปที่ 2.12** SPIx Status and Control Register

โหมดการทำงานภายในโมดูล SPI ใน PIC สามารถที่จะกำหนดได้หลายโหมดด้วยเช่นกัน

1. รับส่งข้อมูลขนาด 8 และ 16 บิต
2. โหมดมาสเตอร์และสเลฟ
3. โหมด SPI เฟรม

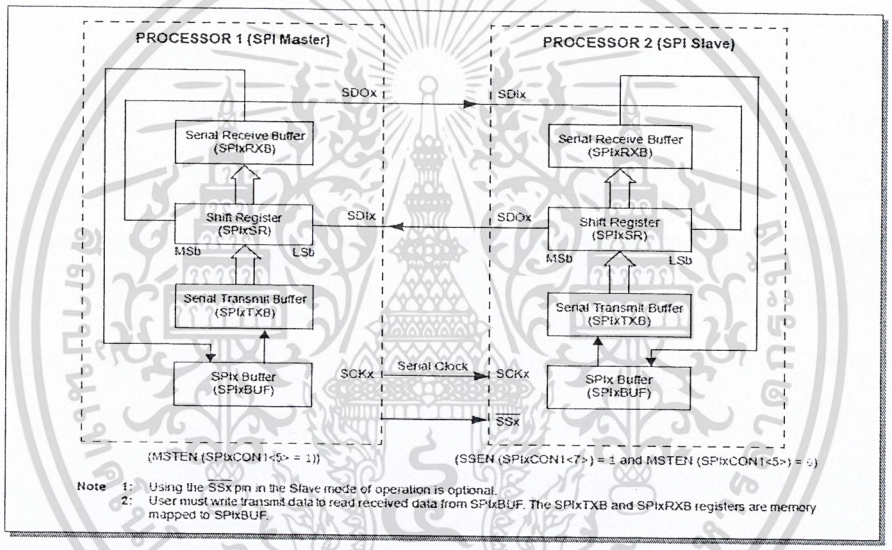
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**3. โหมดการรับส่งข้อมูลขนาด 8 และ 16 บิต**

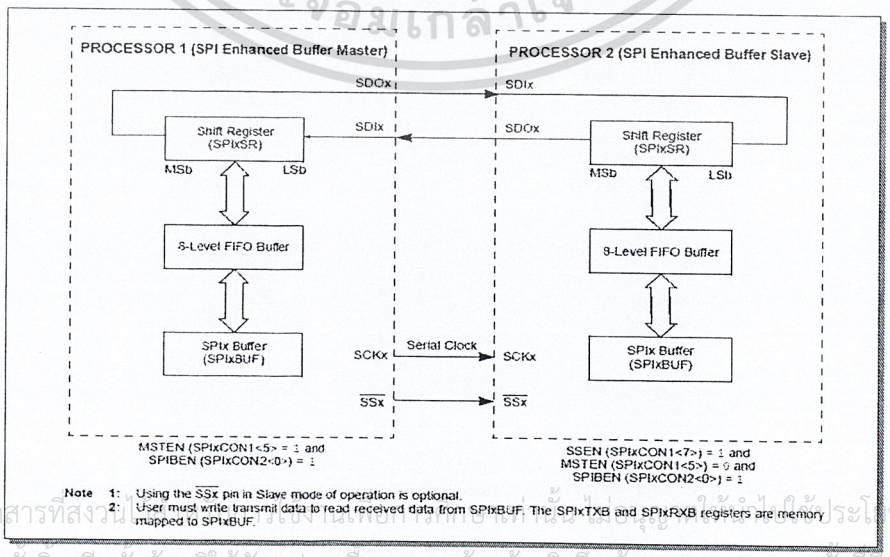
การรับส่งข้อมูลขนาด 8 หรือ 16 บิตโดยควบคุมผ่านบิต MODE16 (SPIxCON)

**4. โหมดมาสเตอร์และสเลฟ**

การทำงานในโหมดมาสเตอร์และสเลฟ จะต้องกำหนดความสัมพันธ์ของสัญญาณนาฬิกาในบิต SPRE (SPIxCON <2:0>) และบิต PPRE (SPIxCON <1:0>) การรับส่งข้อมูลระหว่างมาสเตอร์และสเลฟในบิต CKE (SPI Clock Edge Select bit) และ CKP (Clock Polarity Select bit) การเชื่อมต่อโหมดมาสเตอร์และสเลฟแบบใช้สาย 3 เส้น และ 4 เส้น แสดงดังรูป




**รูปที่ 2.13** การเชื่อมต่อสัญญาณในโหมดมาสเตอร์และสเลฟแบบ 3 สาย




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูปที่ 2.14** การเชื่อมต่อสัญญาณในโหมดมาสเตอร์และสเลฟแบบ 4 สาย

อุปกรณ์สเลฟบางตัวจะกำหนดความถี่สัญญาณนาฬิกาสูงสุดในการติดต่อไว้ การปรับความถี่เพื่อให้เหมาะสมสามารถกำหนดด้วยค่าปรีสเกลเลอร์ทั้งในส่วนของ Primary และ Secondary โดยคำนวณไว้ดังนี้

$$FSCK = \frac{FCY}{\text{Primary Prescaler} * \text{Secondary Prescaler}}$$

**Note 1:** Based on FCY = FOSC/2, Doze mode and PLL are disabled.

**โดย** **FSCK** : ความถี่สัญญาณนาฬิกาอนุกรม SCK

**FCY** : ความถี่สัญญาณนาฬิกาหลัก Fosc/4 หรือ 1/4 ของสัญญาณนาฬิกาหลัก

**Primary Prescaler** : ค่าปรีสเกลเลอร์ในโหมดมาสเตอร์ตัวที่ 1 มีค่าดังนี้ 1:1, 2:1, 3:1,..., 8:1

**Secondary Prescaler** : ค่าปรีสเกลเลอร์ในโหมดมาสเตอร์ตัวที่ 2 มีค่าดังนี้ 1:1, 4:1, 16:1, 64:1

**ตารางแสดงความถี่ของสัญญาณนาฬิกาอนุกรม SCKx**

| FCY = 16 MHz | | Secondary Prescaler Settings | | | | |
|---|---|---|---|---|---|---|
| | | 1:1 | 2:1 | 4:1 | 6:1 | 8:1 |
| Primary Prescaler Settings | 1:1 | Invalid | 8000 | 4000 | 2667 | 2000 |
| | 4:1 | 4000 | 2000 | 1000 | 667 | 500 |
| | 16:1 | 1000 | 500 | 250 | 167 | 125 |
| | 64:1 | 250 | 125 | 63 | 42 | 31 |
| FCY = 5 MHz | | | | | | |
| Primary Prescaler Settings | 1:1 | 5000 | 2500 | 1250 | 833 | 625 |
| | 4:1 | 1250 | 625 | 313 | 208 | 156 |
| | 16:1 | 313 | 156 | 78 | 52 | 39 |
| | 64:1 | 78 | 39 | 20 | 13 | 10 |

Note 1: Based on FCY = FOSC/2, Doze mode and PLL are disabled.
2: SCKx frequencies shown in kHz.

### 5. โหมด SPI เฟรม

โปรโตรคอลโหมดเฟรมในโมดูล SPI จะถูกใช้กำหนดโหมดการทำงานของเฟรมโหมดมาสเตอร์หรือโหมดเฟรมสเลฟ ควบคุมผ่านบิต FRMEN (SPIxCON <14>) และบิต SPIFSD (SPIxCON <13>) เพื่อควบคุมการทำงานของขาพอร์ต SSx สำหรับการสร้างเฟรมซิงโครไนส์พัลส์

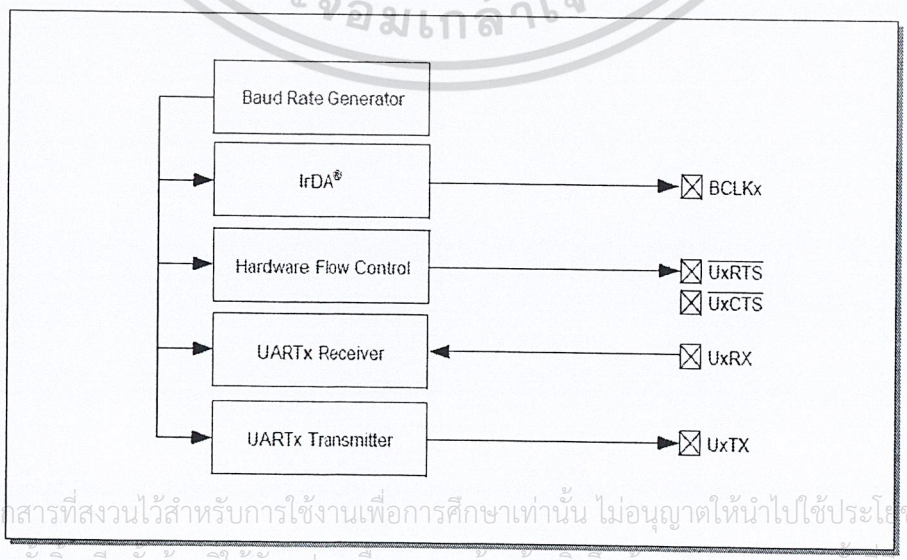
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ลบข้อความนี้ออกจากเอกสารทุกครั้งที่มีการนำไปใช้

**หมายเหตุ** : รายละเอียดการใช้งานโมดูล SPI ศึกษาเพิ่มเติมได้ที่ Datasheet PIC24FJ128GA010

**2.2.3.4 UART Module**

โมดูล UART หรือ Universal Asynchronous Receiver Transmitter เป็นการสื่อสารข้อมูลในรูปแบบอะซิงโคนัสแบบฟลูดูเพล็กซ์ (Full-Duplex) ภายใน PIC24FJ128GA010 มีโมดูล UART 2 โมดูลซึ่งเป็นการรับส่งข้อมูลได้พร้อมกันสองทิศทาง สามารถนำมาประยุกต์ใช้งานกับการเชื่อมต่อกับอุปกรณ์ภายนอก เช่น Computer, RS-232, RS-485 และ สามารถใช้งานร่วมกับ Hardware Flow Control โดยควบคุมการทำงานผ่านขา UxCTS และ UxRTS รวมทั้ง IrDA encoder และ decoder พร้อมทั้งแสดงคุณสมบัติเบื้องต้นของโมดูล UART ดังนี้

1. รับส่งข้อมูลแบบ Full- Duplex
2. มีคุณสมบัติการกำหนดข้อมูลการรับส่ง ไม่ว่าจะเป็นจำนวนบิตข้อมูล (8 หรือ 9 บิต), บิตพาริตี้ (คู่, คี่, ไม่มี), บิตหยุดข้อมูล (1 หรือ 2 บิต)
3. กำหนดอัตรา Baud Rate พร้อมกับปรีสเกลเลอร์ 16 บิต
4. มีบัฟเฟอร์รับส่งข้อมูล 4 ระดับ
5. กำหนดอัตรารับส่งข้อมูลได้ตั้งแต่ 15 bps ถึง 1Mbps ที่ 16 MIPS
6. อินเตอร์รัปต์รับส่งข้อมูล
7. มีโหมด Loop Back เพื่อตรวจสอบข้อมูล
8. มีส่วนตรวจสอบข้อผิดพลาดพาริตี้, เฟรมข้อมูลและการซ้อนทับของบัฟเฟอร์ข้อมูล
9. สนับสนุนการทำงานในโหมด 9 บิต พร้อมการตรวจจับแอดเดรส




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**รูปที่ 2.15** UART Simplified Block Diagram

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง



### 1. รีจีสเตอร์ควบคุมการทำงานโมดูล UART

ในการกำหนดคุณสมบัติการทำงานของโมดูล UART จะถูกกำหนดผ่านรีจิสเตอร์ที่ควบคุมการทำงาน คือ UxMODE และ UxSTA ซึ่งแสดง Bits ต่างๆที่ใช้กำหนดสถานะ การทำงานของทั้ง 2 Register ดังรูป

| R/W-0 | U-0 | R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| UARTEN | — | USIDL | IREN(1) | RTSMD | — | UEN1 | UEN0 |
| bit 15 | | | | | | | bit 8 |

| R/W-0, HC | R/W-0 | R/W-0, HC | R/W-0 | R/W-0 | R/W-0 | R/W-0 | R/W-0 |
|---|---|---|---|---|---|---|---|
| WAKE | LPBACK | ABAUD | RXINV | BRGH | PDSEL1 | PDSEL0 | STSEL |
| bit 7 | | | | | | | bit 0 |

| Legend: | HC = Hardware Clearable bit | | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

**รูปที่ 2.16** UARTx Mode Register

| R/W-0 | R/W-0 | R/W-0 | U-0 | R/W-0, HC | R/W-0 | R-0 | R-1 |
|---|---|---|---|---|---|---|---|
| UTXISEL1 | TXINV | UTXISEL0 | — | UTXBRK | UTXEN | UTXBF | TRMT |
| bit 15 | | | | | | | bit 8 |

| R/W-0 | R/W-0 | R/W-0 | R-1 | R-0 | R-0 | R/C-0 | R-0 |
|---|---|---|---|---|---|---|---|
| URXISEL1 | URXISEL0 | ADDEN | RIDLE | PERR | FERR | OERR | URXDA |
| bit 7 | | | | | | | bit 0 |

| Legend: | C = Clearable bit | HC = Hardware Clearable bit | |
|---|---|---|---|
| R = Readable bit | W = Writable bit | U = Unimplemented bit, read as '0' | |
| -n = Value at POR | '1' = Bit is set | '0' = Bit is cleared | x = Bit is unknown |

**รูปที่ 2.17** UARTx Status and Control Register

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

119219

**2. การคำนวณหาอัตราบอด (Baud Rate)**

เป็นการแสดงสูตรการคำนวณอัตราบอดพร้อมตัวอย่างการคำนวณ

$$\text{Baud Rate} = \frac{F_{CY}}{16 \bullet (BRGx + 1)}$$

$$BRGx = \frac{F_{CY}}{16 \bullet \text{Baud Rate}} - 1$$

โดยที่ $F_{CY}$ = ความถี่สัญญาณนาฬิกาหลักหารด้วย 4 (Fosc/4)

**Ex.** Fosc = 7.3728MHz, PLLx = 4, Baud Rate = 9600 bps คำนวณหาค่า BRG เพื่อนำไปกำหนดให้กับรีจิสเตอร์ UxBRG ได้ดังนี้

$F_{CY}$ = (Fosc x PLLx) / 4

= (7.3728MHz x 4) / 4

= 7.3728MHz

Baud Rate = $F_{CY}$ / [16 x (UxBRG + 1)]

9600 = 7.3728MHz / [16 x (UxBRG + 1)]

UxBRG = [7372800 / (16 x 9600)] – 1

= 47

Calc Baud Rate = 7372800 / (16(47 + 1))

= 9600

Error = (9600 – 9600) / 9600

= 0%

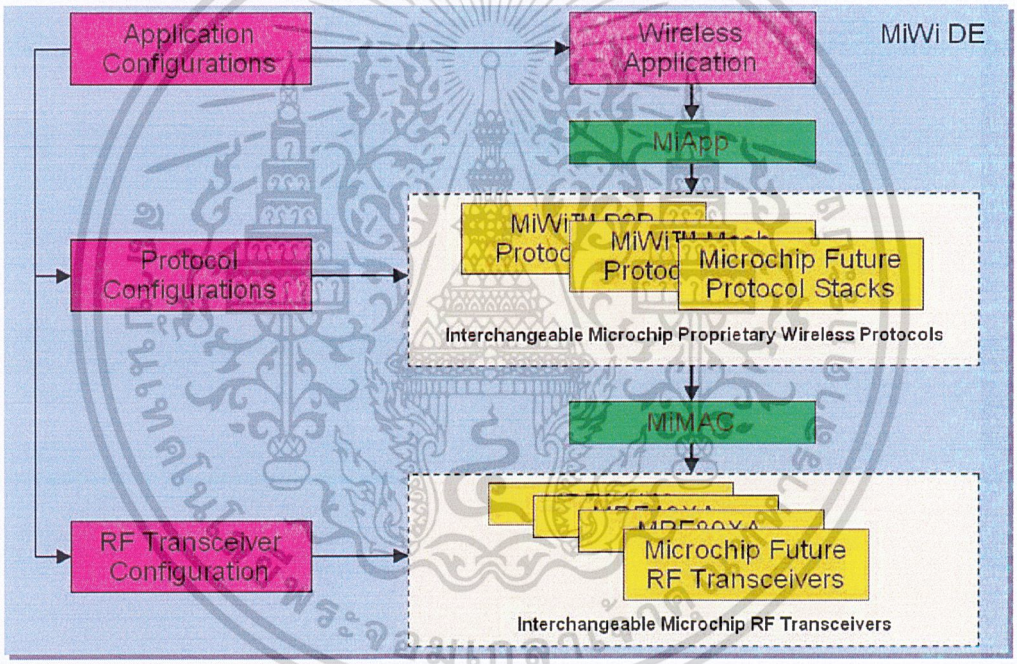
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น ทั้งนี้ห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารนี้ทุกครั้งที่มีการนำไปใช้

**หมายเหตุ** : รายละเอียดการใช้งานโมดูล UART ศึกษาเพิ่มเติมได้ที่ Datasheet PIC24FJ128GA010

## 2.3 การเขียนโปรแกรมควบคุมการทำงานชิพ MIWI

เมื่อทำการออกแบบและสร้างบอร์ดทดลองได้แล้ว ขั้นตอนต่อมาเป็นการทดสอบบอร์ดทดลองว่าสามารถใช้งานได้ตามปกติแล้ว หัวข้อนี้จะเป็นส่วนของการเขียนโปรแกรม โดยโครงงานนี้จะทำการเขียนโปรแกรมด้วยภาษาซี (C Programming) ซึ่งโปรแกรมที่ใช้ในการเขียนภาษาซีคือโปรแกรม MPLAB IDE และใช้ C30 Compiler เป็นตัวคอมไพล์ และในส่วนของโปรแกรมในเบื้องต้นจะทำการใช้ MIWI P2P Wireless Protocol Stack ซึ่งรายละเอียดจะกล่าวในหัวข้อถัดไป

### 2.3.1 MIWI Connectivity Made Simple




**รูปที่ 2.18** MIWI DE

จากรูปที่ 2.8 MIWI DE เป็นสถาปัตยกรรมภายในของชิพMIWI (MRF24J40MA) ซึ่งภายในจะประกอบด้วย 3 Layer คือ Application Layer, Protocol Layer และRF Transceiver Layer และภายใน Application Layer จะใช้ MiApp เป็นตัวสื่อสารกับ Protocol Layer ส่วนภายใน Protocol Layer จะมีฟังก์ชัน MIWI P2P และ MIWI Wireless Communication ที่สามารถถูกเรียกใช้งานส่วนภายใน RF Transceiver Layer จะมี MRF24J40, MRF49XA และMRF89XA ซึ่งถูกเรียกใช้โดย MiMAC ที่ได้กล่าวมาข้างต้น เป็นลำดับขั้นตอนการกำหนดการทำงานของฟังก์ชันในแต่ละ Layer เบื้องต้นของชิพMIWI (MRF24J40MA)

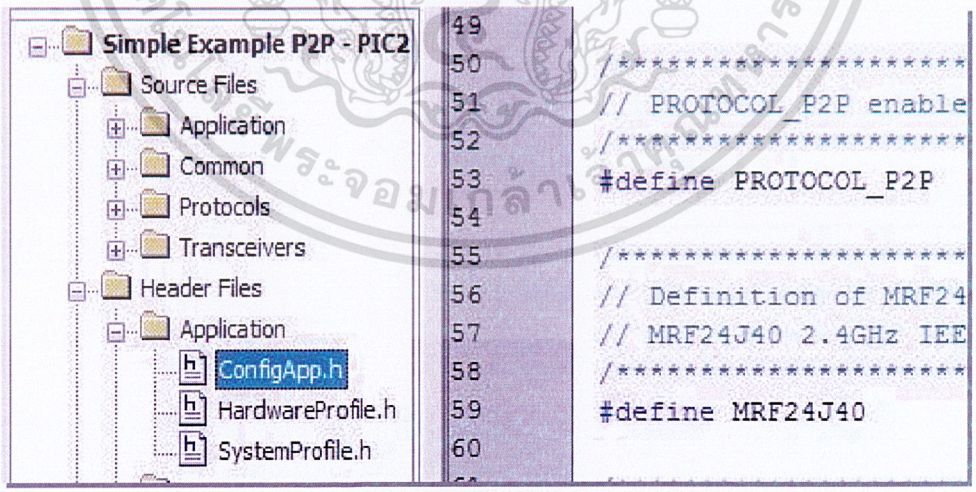
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

### 2.3.2 MIWI P2P Wireless Protocol Stack

เป็นที่ทราบกันว่าในคอมพิวเตอร์มี Windows, Linux เป็นOS (Operating System) ในทำนองเดียวกัน Stack ก็เปรียบเสมือนเป็นOS (Operating System) ในไมโครคอนโทรลเลอร์ซึ่งทำหน้าที่ควบคุมการทำงานในส่วนต่างๆของ MIWI ซึ่งถือเป็นจุดเด่นของบริษัท Microchip ที่สร้าง Stack ออกมาเพราะทำให้ผู้ใช้สามารถใช้งานได้สะดวกยิ่งขึ้น เมื่อนำ Stack MIWI มาใช้งานเราจะต้องทำการกำหนดฟังก์ชันการทำงานเบื้องต้นภายในโปรแกรมตรง Application Configuration ซึ่งจะเป็นกำหนดในส่วน Application Layer ตามรูปที่ 2.8 MIWI DE ซึ่งจะมีรายละเอียดขั้นตอนดังต่อไปนี้

**1.** เมื่อทำการเปิดโปรแกรม MIWI Stack จะต้องทำการเลือกบน Menu Configure -> Select Device -> PIC24FJ128GA010 -> OK

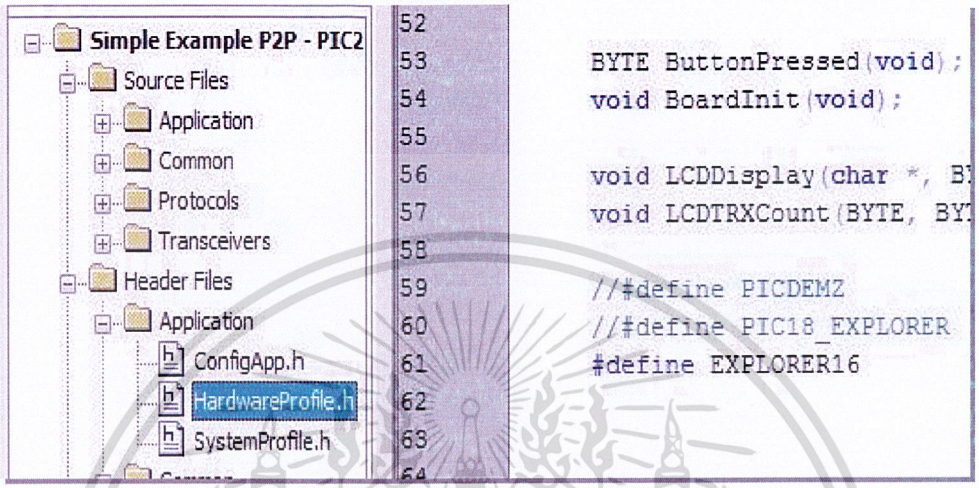
**2.** ดับเบิ้ลคลิกที่ Header Files -> Application -> ConfigApp.h ทำการเลือกใช้ #define PROTOCOL_P2P และ #define MRF24J40 ดังรูป




**รูปที่ 2.19** ConfigApp.h

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**3.** ดับเบิ้ลคลิกที่ Header Files -> Application -> HardwareProfile.h ทำการเลือกใช้บอร์ด Explorer 16 ตรง #define EXPLORER16 และปิดการใช้งาน PICDEMZ และ PIC18_EXPLORER ดังรูป




**รูปที่ 2.20** HardwareProfile.h

เมื่อทำการกำหนดฟังก์ชันการทำงานได้ตามต้องการแล้วขั้นตอนต่อไปคือการ Compile ก็จะได้ file สกุล.hex ไปโปรแกรมลงบนไมโครคอนโทลเลอร์ โดยขั้นตอนการกำหนดค่าเบื้องต้นที่ได้กล่าวมานี้เป็นส่วนของ Node 1 และในขั้นตอนต่อไปก็ทำการกำหนดค่ากับ Node 2 ซึ่งขั้นตอนต่างๆเหมือนกับที่ได้กล่าวไว้ตามขั้นตอนข้างต้น เพียงแค่เลือกใช้ MIWI Stack ของ Node 2 และเมื่อเสร็จทั้ง 2 ขั้นตอนก็จะสามารถส่งผ่านข้อมูลไร้สายระยะใกล้แบบ MIWI ในแบบพื้นฐานได้

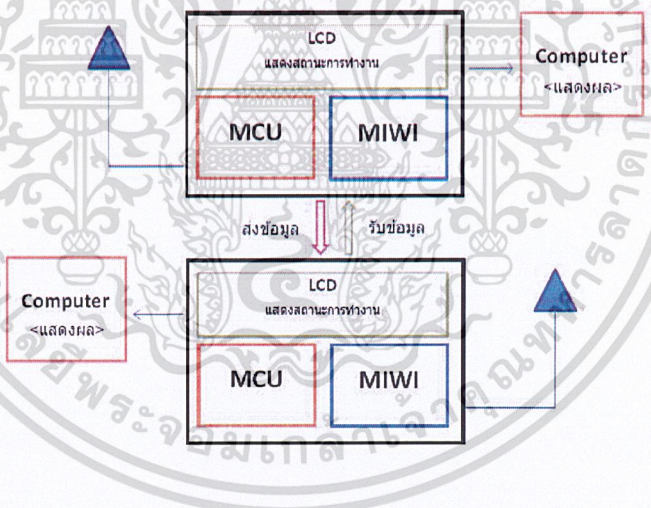
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 3

# การคำนวณและการสร้าง

การออกแบบบอร์ดรับส่งข้อมูลไร้สายโดยผ่านโมดูล MIWI ในเบื้องต้นนั้นต้องเข้าใจหลักการทำงานในภาพรวมของระบบก่อนว่ามีหลักการทำงานอย่างไร แล้วจึงพิจารณาส่วนย่อยต่างๆ อันประกอบไปด้วย ส่วนส่งข้อมูลไร้สายแบบ MIWI ส่วนรับข้อมูลไร้สายแบบ MIWI ส่วนแสดงสถานการณ์ทำงาน และโครงสร้างทางฮาร์ดแวร์

หลักการทำงานในภาพรวมของบอร์ดรับส่งข้อมูลไร้สายโดยผ่านโมดูล MIWI สามารถแสดงดังแผนภาพในรูปที่ 3.1




**รูปที่ 3.1** โครงสร้างโดยรวมของบอร์ดรับส่งข้อมูลไร้สายแบบ MIWI

จากรูปที่ 3.1 เห็นว่าบอร์ดรับส่งข้อมูลไร้สายโดยผ่านโมดูล MIWI ประกอบด้วยส่วนหลักๆ ดังนี้คือ

- บอร์ดส่งข้อมูล MIWI
- บอร์ดรับข้อมูล MIWI
- จอ LCD แสดงสถานะการทำงาน

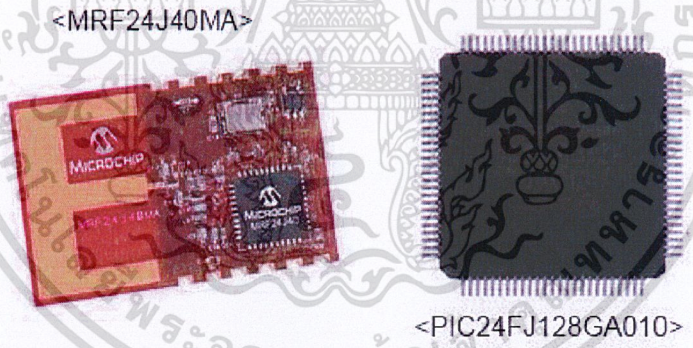
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 3.1 บอร์ดรับส่งข้อมูลไร้สายโดยผ่านโมดูล MIWI

บอร์ดรับและบอร์ดส่งข้อมูลแบบ MIWI เป็นอุปกรณ์ที่สำคัญสำหรับโครงงานนี้ เนื่องจากเป็นอุปกรณ์หลักในการรับส่งข้อมูลไร้สาย จะเห็นได้ว่าในโครงงานนี้ได้สร้างบอร์ดรับส่งข้อมูลไร้สาย เหมือนกันเพื่อที่จะสะดวกในการใช้งาน เพราะในบอร์ด 1 บอร์ดสามารถเป็นได้ทั้งตัวรับและตัวส่งข้อมูล

ในการเลือกใช้โมดูลไร้สายแบบ MIWI และไมโคคอนโทรลเลอร์ในการรับส่งข้อมูลต้องพิจารณาในเรื่องต่างๆ ดังนี้

- ส่งที่จำเป็นในการรับส่งข้อมูลไร้สายแบบ MIWI
- คุณสมบัติของโมดูล MIWI และไมโครคอนโทรลเลอร์




**รูปที่ 3.2** โมดูลรับส่งข้อมูลไร้สาย MIWI และไมโครคอนโทรลเลอร์

### 3.1.1 สิ่งที่จำเป็นในการรับส่งข้อมูลไร้สายแบบ MIWI

- มาตรฐานในการรับส่งข้อมูลต้องไม่ต่ำกว่า IEEE 802.15.4
- ระยะในการรับส่งต้องไม่น้อยกว่า 300 Ft
- สามารถส่งข้อมูลในรูปแบบ P2P

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**3.1.2** คุณสมบัติของโมดูลรับส่งข้อมูลแบบ MIWI และไมโครคอนโทรลเลอร์

- โมดูลรับส่งข้อมูลแบบ MIWI

  - สามารถส่งด้วยความเร็ว 250 Kps
  - สามารถส่งได้ระยะสูงสุดที่ 400 Ft
  - สามารถรับส่งได้แบบ ZigBee, MIWI, MIWI P2P และ Wireless Networking

- ไมโครคอนโทรลเลอร์

  - สามารถเชื่อมต่อกับโมดูล MIWI (MRF24J40MA) ได้
  - มีความเร็วในการประมวลผลที่ 16 MIPS
  - มี Port UART และหน่วยความจำ 128 Kbytes

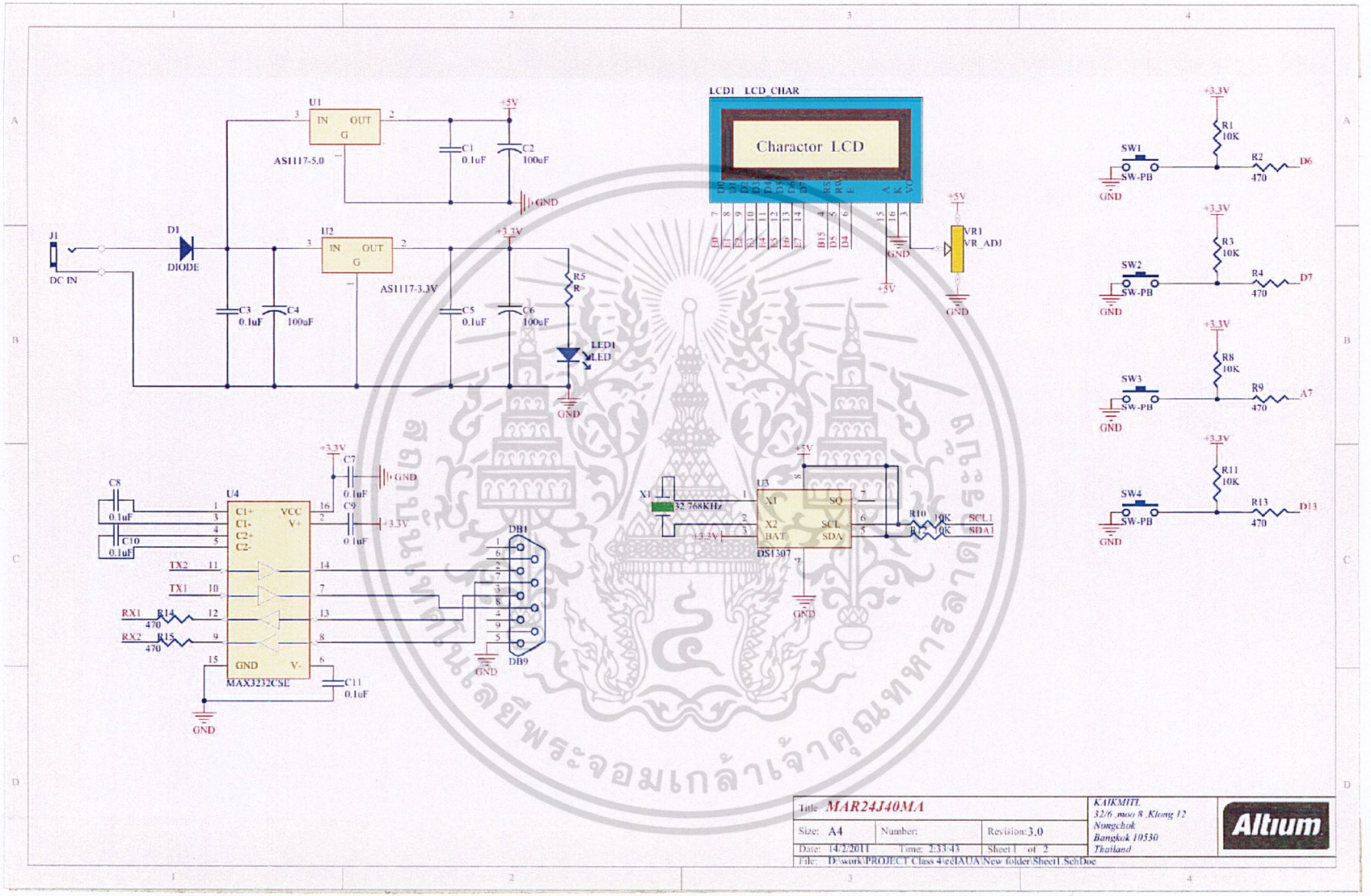
## 3.2 จอ LCD แสดงสถานะการทำงาน

จอ LCD ในโครงงานนี้ที่ติดตั้งอยู่กับบอร์ดรับส่งข้อมูลไร้สายแบบ MIWI เพื่อที่จะทำการแสดงสถานการทำงานของบอร์ดรับส่งข้อมูล ว่าได้มีการเชื่อมต่อกันอย่างถูกต้องตามที่ได้สั่งการไปหรือไม่

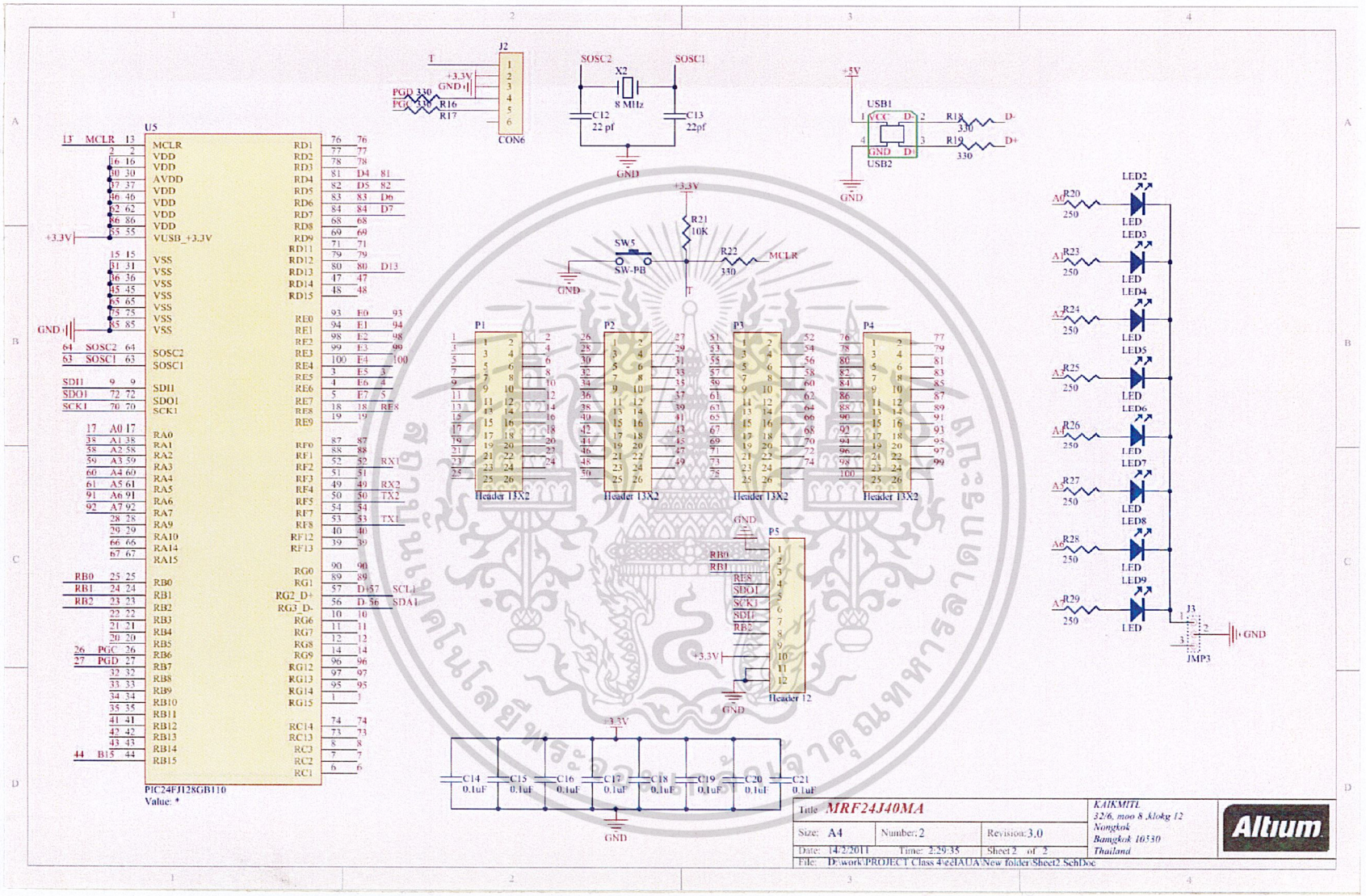
## 3.3 โครงสร้างทางด้านฮาร์ดแวร์

โครงงานนี้ได้ทำบอร์ดรับส่งข้อมูลไร้สายแบบ MIWI โดยใช้โครงสร้างทางด้านฮาร์ดแวร์ดังรูปที่ 3.3 และรูปที่ 3.4 โดยใช้โปรแกรม Altium Design Winter 09 ในการออกแบบวงจร Schematic ในส่วนต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




รูปที่ 3. 3 วงจรในการออกแบบส่วนที่ 1




รูปที่ 3. 4 วงจรในการออกแบบส่วนที่ 2

# บทที่ 4

# การทดลองและผลการทดลอง

ในบทนี้ จะกล่าวถึงการทดลองและผลการทดลอง ของบอร์ดรับส่งข้อมูลไร้สายโดยผ่านโมดูล MIWI โดยมีรายละเอียดของการทดลองดังนี้

## 4.1 การทดลองรับส่งข้อมูลจากบอร์ดทดลองที่ 1 ไปยังบอร์ดทดลองที่ 2

### 4.1.1 ทดสอบระยะในการรับส่งข้อมูลของบอร์ดทดลอง

ในส่วนที่ 1 นี้จะเป็นการทดสอบระยะทางในการรับส่งข้อมูล บริเวณพื้นที่โล่งแจ้งเพื่อทดสอบประสิทธิภาพในการรับส่งข้อมูลของบอร์ดทดลอง โดยจะทำการส่งข้อมูลจากบอร์ดทดลองที่ 1 ไปสั่งการให้บอร์ดทดลองที่ 2 ทำงาน และดูผลจากสถานะของหลอดไฟ LED

**ตารางที่ 4.1** ผลการรับส่งข้อมูลของบอร์ดรับส่งข้อมูลไร้สายโดยผ่านโมดูล MIWI

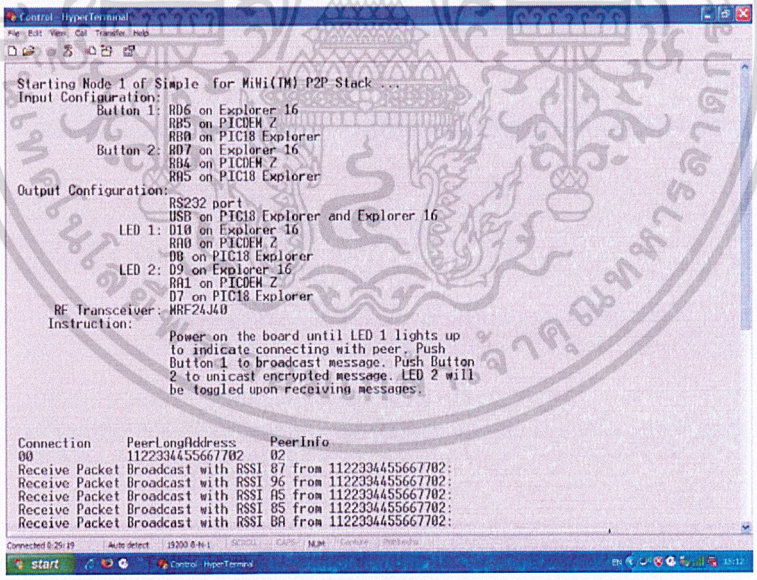
| ลำดับที่ | ระยะทาง (เมตร) | บอร์ดส่งข้อมูล | บอร์ดรับข้อมูล | สถานะของ LED | ผลการรับส่งข้อมูล | |
|---|---|---|---|---|---|---|
| | | | | | ได้ | ไม่ได้ |
| 1 | 20 | High | High | ติด | / | - |
| 2 | 40 | Low | Low | ดับ | / | - |
| 3 | 60 | High | High | ติด | / | - |
| 4 | 80 | Low | Low | ดับ | / | - |
| 5 | 100 | High | High | ติด | / | - |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากการทดลองจะเห็นได้ว่า เมื่อบอร์ดทดลองบอร์ดที่ 1 สั่งให้ข้อมูลออกเป็น High เพื่อไปขับ LED ที่บอร์ดที่ 2 บอร์ดที่ 2 ก็ทำการส่งข้อมูลไปขับ LED ที่ติดอยู่ที่บอร์ด LED จึงแสดงสถานะสว่าง จากนั้นได้ทำการสั่ง Low ไปยังบอร์ดที่ 2 เพื่อให้ LED ดับ และจากการทดสอบระยะในการรับส่งข้อมูลนั้น สามารถวัดระยะการส่งข้อมูลในบริเวณพื้นที่โล่งแจ้งและไม่มีสิ่งกีดขวางได้ 100 เมตร ซึ่งเพียงพอแล้วสำหรับการรับส่งข้อมูลภายในห้อง อย่างไรก็ตามโมดูล MIWI (MRF24J40MA) สามารถรับส่งข้อมูลได้ระยะสูงสุดอยู่ที่ 120 เมตร

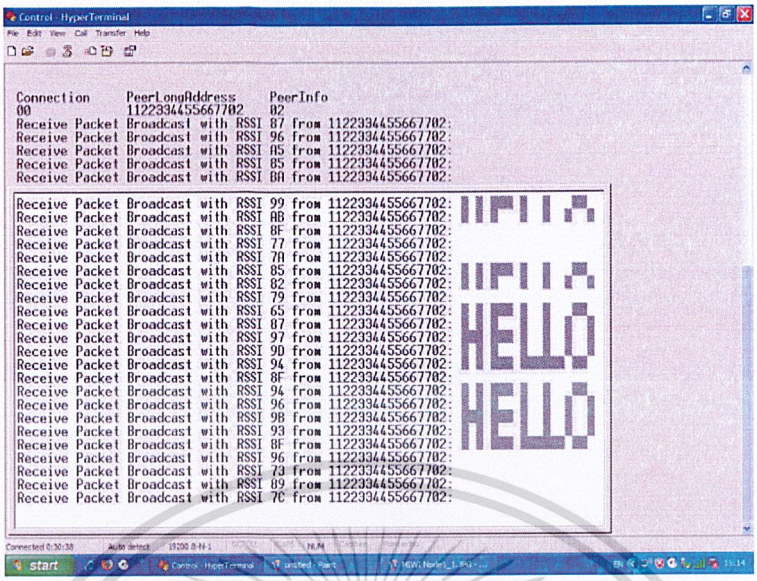
**4.1.2** ทดสอบการแสดงผลของข้อมูลผ่านทางจอคอมพิวเตอร์

ในส่วนนี้จะทำการเชื่อมต่อบอร์ดทดลองทั้ง 2 บอร์ดเข้ากับคอมพิวเตอร์ เพื่อดูสถานะการเชื่อมต่อและข้อมูลในการรับส่งว่าเป็นไปตามที่ต้องการหรือไม่ ผลการทดลองดูได้จากรูปที่ 4.1, 4.2, 4.3 และรูปที่ 4.4



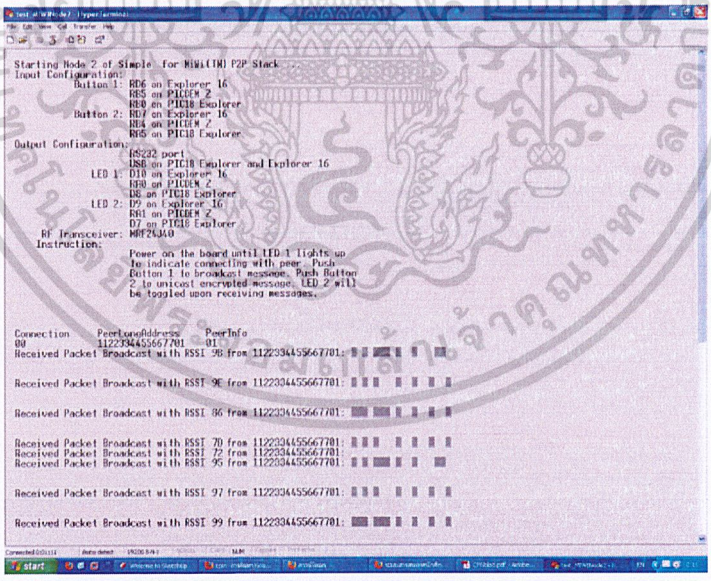

**รูปที่ 4.1** บอร์ดที่ 1 แสดงสถานะว่าได้มีการเชื่อมต่อแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



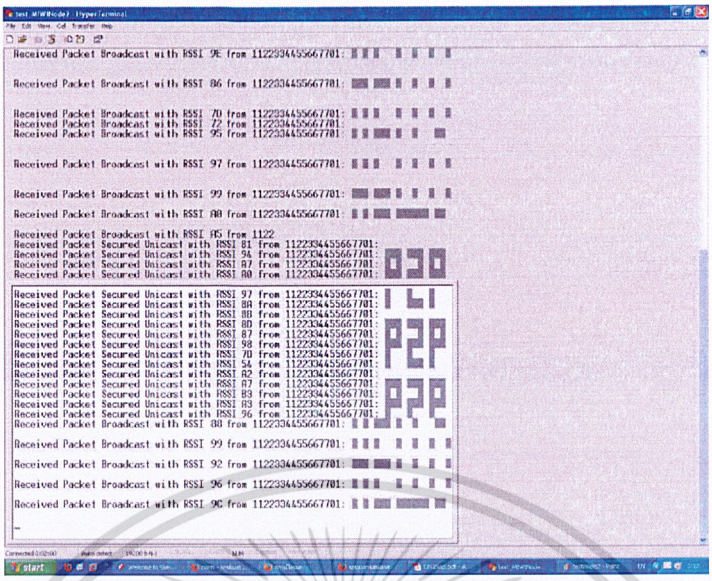

**รูปที่ 4.2** แสดงการรับข้อมูลจากบอร์ดทดลองที่ 2

จากรูปที่ 4 .1 และรูปที่ 4.2 เป็นการแสดงว่าบอร์ดทดลองบอร์ดที่ 1 สามารถที่จะเชื่อมต่อกับบอร์ดทดลองบอร์ดที่ 2 ได้แล้วและสามารถรับข้อมูลที่บอร์ดที่ 2 ส่งมาได้ด้วยเช่นกัน




**รูปที่ 4.3** บอร์ดที่ 2 แสดงสถานะว่าได้มีการเชื่อมต่อแล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

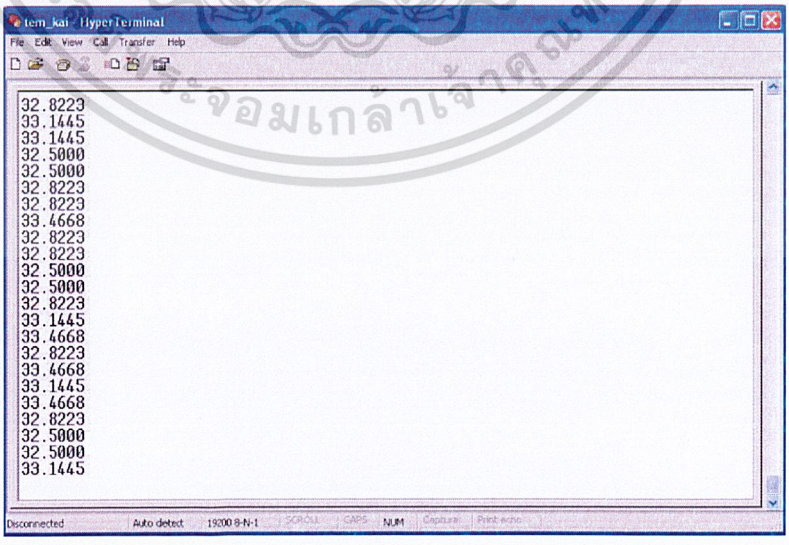



**รูปที่ 4.4** แสดงการรับข้อมูลจากบอร์ดทดลองที่ 1 ที่ส่งมา

จากรูปที่ 4.3 และรูปที่ 4.4 แสดงให้เห็นว่าบอร์ดทดลองทั้ง 2 บอร์ดสามารถเชื่อมต่อกันและสามารถรับส่งข้อมูลให้กันได้

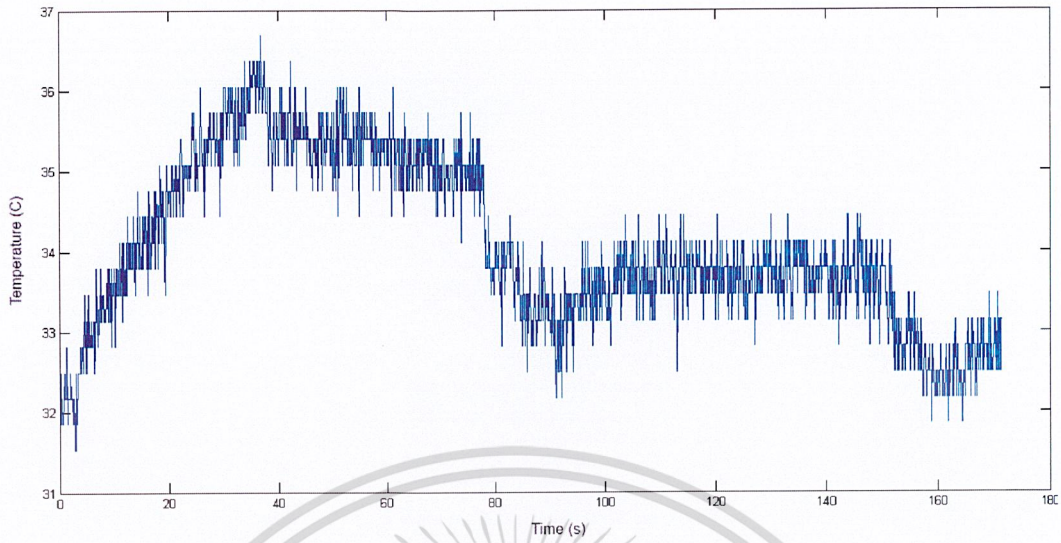
**4.1.3** ทดสอบการอ่านค่าอุณหภูมิ

ในส่วนนี้จะทำการรับส่งค่าอุณหภูมิที่อ่านได้จากบอร์ดทดลองที่ 1 แล้วส่งข้อมูลที่ได้ไปยังบอร์ดทดลองที่ 2 เพื่อแสดงค่าของอุณหภูมิที่บอร์ดที่ 1 ส่งมาดังนี้




**รูปที่ 4.5** แสดงการรับค่าอุณหภูมิจากบอร์ดทดลองที่ 1

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้




**รูปที่ 4.6** กราฟแสดงช่วงเวลาในการอ่านค่าอุณหภูมิ

จากการทดลองทั้ง 3 กรณีข้างต้นพบว่า บอร์ดทดลองสามารถรับส่งข้อมูลได้โดยดูจากสถานะของ LED หรือดูจากการแสดงผลผ่านหน้าจอคอมพิวเตอร์ และจากการทดสอบระยะการรับส่งข้อมูลสามารถรับส่งข้อมูลตามระยะทางที่เรายอมรับได้เป็นอย่างดี ในส่วนของการอ่านค่าอุณหภูมิจากบอร์ดทดลองที่ 1 แล้วให้บอร์ดทดลองที่ 2 แสดงผลก็สามารถรับส่งข้อมูลได้อย่างถูกต้องตามต้องการ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# บทที่ 5

# บทวิจารณ์และสรุปผล

## 5.1 สรุปผลการทดลอง

จากการทดลองบอร์ดรับส่งข้อมูลไร้สายผ่านโมดูล MIWI โดยทำการส่งข้อมูลเพื่อควบคุมหลอดไฟ LED โดยบอร์ดที่ 1 เป็นตัวส่งข้อมูลและบอร์ดที่ 2 เป็นตัวรับข้อมูล พบว่าบอร์ดทดลองทั้ง 2 บอร์ดสามารถเชื่อต่อและรับส่งข้อมูลกันได้โดยดูจากสถานะของ LED ที่มีการเปลี่ยนแปลงเมื่อทำการสั่งงานจากบอร์ดทดลองที่ 1 จากการทดลองสามารถรับส่งข้อมูลได้ในระยะ 100 เมตร ซึ่งเพียงพอแล้วกับการใช้งาน แต่พบข้อที่น่าสังเกตว่าเมื่อระยะในการรับส่งข้อมูลเพิ่มขึ้นข้อมูลที่ถูกส่งไปนั้นเริ่มมีการ Delay หรือรับส่งข้อมูลไม่ได้เนื่องจากระดับสัญญาณในการเชื่อมต่อมีระดับที่ต่ำ สาเหตุน่าจะมาจากหลายปัจจัย เช่น มีสิ่งกีดขวางทางผ่านของสัญญาณ สภาพอากาศ เป็นต้น เนื่องจากในการส่งสัญญาณนั้นไปทำการกีดขวางทางผ่านของสัญญาณที่ส่งมา และที่สำคัญโมดูลที่เลือกใช้เป็นโมดูลในการรับส่งข้อมูลระยะใกล้เท่านั้นและการรับส่งผ่านพื้นที่ที่มีสิ่งกีดขวางจะทำได้ไม่ดีเท่าไหร่

## 5.2 ปัญหาและแนวทางแก้ไข

จากการศึกษาและทำโครงงานนี้ในช่วงแรกเกิดปัญหาทางด้านการออกแบบฮาร์ดแวร์และการเขียนโปรแกรมเชื่อมต่อโมดูล MIWI กับไมโครคอนโทรลเลอร์ เนื่องจากไม่มีประสบการณ์ในการออกกแบบฮาร์ดแวร์ร่วมทั้งการเขียนโปรแกรมไมโครคอนโทรลเลอร์มาก่อน นอกจากนั้นเมื่อได้ทำการทดลองแล้วพบว่า การออกแบบฮาร์ดแวร์มีปัญหาจึงต้องทำการออกแบบฮาร์ดแวร์ขึ้นมาใหม่ เพื่อทำให้การรับส่งข้อมูลของบอร์ดทดลองเชื่อมต่อกันได้และสามารถรับส่งข้อมูลได้ตามต้องการ

ในส่วนของปัญหาที่ยังไม่ได้ทำการปรับปรุงคือในส่วนของการตรวจสอบการ Delay ของการรับส่งข้อมูลว่าข้อมูลที่ทำการส่งไปนั้นในแต่ละระยะทางสามารถส่งได้ความเร็วเท่าไหร่ถึงจะไปถึงบอร์ดทดลองรับข้อมูล ดังนั้นควรติดตั้งอุปกรณ์เสริมในการวัดเวลาแบบ Real Time Clock เพื่อตรวจสอบเวลาในการรับส่งข้อมูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 5.3 ข้อเสนอแนะและแนวทางในการค้นคว้าพัฒนา

จากการที่ได้ทำโครงงานชิ้นนี้จะเห็นได้ว่า การรับส่งข้อมูลนี้สามารถที่จะนำไปประยุกต์ใช้งานในส่วนของการควบคุมอุปกรณ์ต่างๆ ในบริเวณห้องขนาดเล็ก ตัวอย่างเช่น การควบคุมหลอดไฟภายในห้อง การควบคุมโทรทัศน์ เครื่องเสียง หรือการนำไปเชื่อมต่อกับอุปกรณ์ที่ต้องทำการรับส่งข้อมูลแต่อยู่คนละตำแหน่งกัน เพื่อให้การรับส่งข้อมูลเป็นไปได้สะดวกและรวดเร็วขึ้น เพื่อลดความยุ่งยากเรื่องการเดินสายเชื่อมต่อข้อมูล

นอกจากนี้เราสามารถทำการรับส่งข้อมูลเป็นระบบ Network ซึ่งจะทำให้เราสามารถควบคุมอุปกรณ์หลายๆอย่างได้พร้อมๆกัน ทำให้มีความสะดวกมากขึ้นในการควบคุมอุปกรณ์หลายส่วน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก ก

# โปรแกรมการประมวลผล

โครงงานนี้ทำการเขียนโปรแกรมการประมวลผลด้วยภาษาซี (C Programming) ซึ่งโปรแกรมที่ใช้ในการเขียนภาษาซี คือ โปรแกรม MPLAB IDE และใช้ C30 Compiler เป็นตัวคอมไพล์ และในส่วนของโปรแกรมเบื้องต้นจะทำการใช้ MIWI P2P Wireless Protocol Stack ซึ่งรายละเอียดสามารถ Download ได้จาก www.microchip.com

## ก.1 โปรแกรมส่วนของการส่งข้อมูลอุณหภูมิ

- **Node 1**

```
----------HEADERS----------

#include "Common\Console.h"

#include "Transceivers\Transceivers.h"

#include "Common\SymbolTime.h"

#include "WirelessProtocols\MCHP_API.h"

#include "Common\LCDBlocking.h"

----------VARIABLES----------

#define LIGHT   0x01

#define SWITCH  0x02
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
------------------------------------------------------------------------------------------------------------

// AdditionalNodeID variable array defines the additional

// information to identify a device on a PAN. This array

// will be transmitted when initiate the connection between

// the two devices. This  variable array will be stored in

// the Connection Entry structure of the partner device. The

// size of this array is ADDITIONAL_NODE_ID_SIZE, defined in

// ConfigApp.h.

// In this demo, this variable array is set to be empty.

------------------------------------------------------------------------------------------------------------

#if ADDITIONAL_NODE_ID_SIZE > 0

BYTE AdditionalNodeID[ADDITIONAL_NODE_ID_SIZE] = {SWITCH};

#endif

------------------------------------------------------------------------------------------------------------

// The variable myChannel defines the channel that the P2P connection

// is operate on. This variable will be only effective if energy scan

// (ENABLE_ED_SCAN) is not turned on. Once the energy scan is turned

// on, the operating channel will be one of the channels available with

// least amount of energy (or noise).
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
-----------------------------------------------------------------------------------------------------------

BYTE myChannel = 25;

extern WORD_VAL myNetworkAddress;

------------------------------------------ Enable ADC 10 Bit----------------------------------------------

void InitADC(void)

{

        AD1PCFG = 0xFFEF; // all PORTB = Digital; RB4 = analog

        AD1CON1 = 0x00E0; // SSRC<2:0> = 111 implies internal counter ends sampling

        // and starts converting.

        AD1CHS = 0x0004; // Connect AN4 as S/H input.

        // in this example AN12 is the input

        AD1CSSL = 0;

        AD1CON3 = 0x1F02; // Sample time = 31Tad, Tad = 3Tcy

        AD1CON2 = 0;

        AD1CON1bits.ADON = 1; // turn ADC ON

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
----------------------------------------------MAIN----------------------------------------------------------

int main(void)

{

        BYTE i;

        BYTE TxNum = 0;

        BYTE RxNum = 0;

        // Initialize the system

        BoardInit();

        ConsoleInit();

        LED_1 = 0;

        LED_2 = 0;

        // Following block display demo information on LCD of Explore 16 or

        // PIC18 Explorer demo board.

        #if defined(MRF49XA)

                LCDDisplay((char *)"Simple P2P Demo  MRF49XA Node 2", 0, TRUE);

        #elif defined(MRF24J40)

                LCDDisplay((char *)"Simple P2P Demo MRF24J40 Node 2", 0, TRUE);

        #endif

        // Initialize Microchip proprietary protocol. Which protocol to use
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// depends on the configuration in ConfigApp.h

MiApp_ProtocolInit();

#if defined(PICDEMZ) || defined(PIC18_EXPLORER)

        INTCONbits.GIEH = 1;

#elif defined(EXPLORER16)

#else

        #error "Unknown board.  Please initialize board as required."

#endif



Printf("\r\nStarting Node 2 of Simple  for MiWi(TM) P2P Stack ...");

Printf("\r\nInput Configuration:");

Printf("\r\n        Button 1: RD6 on Explorer 16");

Printf("\r\n                  RB5 on PICDEM Z");

Printf("\r\n                  RB0 on PIC18 Explorer");

Printf("\r\n        Button 2: RD7 on Explorer 16");

Printf("\r\n                  RB4 on PICDEM Z");

Printf("\r\n                  RA5 on PIC18 Explorer");

Printf("\r\nOutput Configuration:");

Printf("\r\n                  RS232 port");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Printf("\r\n            USB on PIC18 Explorer and Explorer 16");

Printf("\r\n        LED 1: D10 on Explorer 16");

Printf("\r\n            RA0 on PICDEM Z");

Printf("\r\n            D8 on PIC18 Explorer");

Printf("\r\n        LED 2: D9 on Explorer 16");

Printf("\r\n            RA1 on PICDEM Z");

Printf("\r\n            D7 on PIC18 Explorer");

#if defined(MRF24J40)

Printf("\r\n  RF Transceiver: MRF24J40");

#elif defined(MRF49XA)

Printf("\r\n  RF Transceiver: MRF49XA");

#endif

Printf("\r\n  Instruction:");

Printf("\r\n            Power on the board until LED 1 lights up");

Printf("\r\n            to indicate connecting with peer. Push");

Printf("\r\n            Button 1 to broadcast message. Push Button");

Printf("\r\n            2 to unicast encrypted message. LED 2 will");

Printf("\r\n            be toggled upon receiving messages. ");

Printf("\r\n\r\n");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Set default channel

if( MiApp_SetChannel(myChannel) == FALSE )

{

        Printf("\r\nSelection of channel ");

        PrintDec(myChannel);

        Printf(" is not supported in current condition.\r\n");

        #if defined(__18CXX)

                return;

        #else

                return 0;

        #endif

  }

// Function MiApp_ConnectionMode defines the connection mode. The

// possible connection modes are:

//  ENABLE_ALL_CONN:    Enable all kinds of connection

//  ENABLE_PREV_CONN:  Only allow connection already exists in

//  Connection table

//  ENABL_ACTIVE_SCAN_RSP:  Allow response to Active scan

//  DISABLE_ALL_CONN:  Disable all connections.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
MiApp_ConnectionMode(ENABLE_ALL_CONN);

// Display current opertion on LCD of demo board, if applicable

LCDDisplay((char *)"Connecting Peer  on Channel %d", myChannel, FALSE);

// Function MiApp_EstablishConnection try to establish a new connection

// with peer device.

// The first parameter is the index to the active scan result, which is

// acquired by discovery process (active scan). If the value of the

// index is 0xFF, try to establish a connection with any peer.

// The second parameter is the mode to establish connection, either direct

// or indirect. Direct mode means connection within the radio range;

//  Indirect mode means connection may or may not in the radio range.

#ifdef ENABLE_HAND_SHAKE

while( (i = MiApp_EstablishConnection(0xFF, CONN_MODE_DIRECT)) == 0xFF );

#endif

// Display current opertion on LCD of demo board, if applicable

LCDDisplay((char *)" Connected Peer  on Channel %d", myChannel, TRUE);

// Function DumpConnection is used to print out the content of the

// P2P Connection Entry on the hyperterminal. It may be useful in

// the debugging phase.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// The only parameter of this function is the index of the P2P
// Connection Entry. The value of 0xFF means to print out all
// valid P2P Connection Entry; otherwise, the P2P Connection Entry
// of the input index will be printed out.
#ifdef ENABLE_DUMP
        DumpConnection(i);
#endif
// Turn on LED 1 to indicate P2P connection established
LED_1 = 1;
// Following block display demo instructions on LCD based on the
// demo board used.
#if defined(EXPLORER16)
        LCDDisplay((char *)"RD6: Broadcast  RD7: Unicast", 0, FALSE);
#elif defined(PIC18_EXPLORER)
        LCDDisplay((char *)"RB0: Broadcast  RA5: Unicast", 0, FALSE);
#endif
        while(1)
        {
                WORD_VAL ADCValue;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
int i;

unsigned char ADC[2];

static char buf[15];

float T,VoltageOut;

static char state = 1;

TICK myt, myt1, myt2;

if( MiApp_MessageAvailable() )

{

        // Toggle LED2 to indicate receiving a packet.

          LED_2 ^= 1;

          MiApp_DiscardMessage();

          LCDTRXCount(TxNum, ++RxNum);

}

else

{

        BYTE PressedButton = ButtonPressed();

        switch(state)

        {

        case 1:
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
if(PressedButton == 1)

{

        state = 2;

        myt1 = TickGet();

        MiApp_WriteData("a");

        }

break;

case 2:

myt2 = TickGet();

myt.Val = TickGetDiff(myt2, myt1);

if(myt.Val > HUNDRED_MILI_SECOND)

{

        myt1.Val =      myt2.Val;

        MiApp_FlushTx();

         AD1CON1bits.SAMP = 1; // start sampling...

        AD1CON1bits.SAMP = 0; // start converting

        while(!AD1CON1bits.DONE){}; // conversion done?
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                ADCValue.Val = ADC1BUF0;

                sprintf(buf,"\r%d\n",ADCValue.Val);

                ConsolePutROMString(buf);

                MiApp_WriteData(ADCValue.byte.LB);
                MiApp_WriteData(ADCValue.byte.HB);

                MiApp_BroadcastPacket(FALSE);

                LCDTRXCount(++TxNum, RxNum);
            }
            break;
        default:
            break;
    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ก.2 โปรแกรมส่วนของการรับข้อมูลอุณหภูมิ

- **Node 2**

```
----------------------------------------------------HEADERS----------------------------------------------------

#include "Common\Console.h"

#include "Transceivers\Transceivers.h"

#include "Common\SymbolTime.h"

#include "WirelessProtocols\MCHP_API.h"

#include "Common\LCDBlocking.h"

-----------------------------------------------VARIABLES----------------------------------------------

#define LIGHT   0x01

#define SWITCH  0x02

-------------------------------------------------------------------------------------------------------

// AdditionalNodeID variable array defines the additional

// information to identify a device on a PAN. This array

// will be transmitted when initiate the connection between

// the two devices. This  variable array will be stored in

// the Connection Entry structure of the partner device. The

// size of this array is ADDITIONAL_NODE_ID_SIZE, defined in

// ConfigApp.h.
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// In this demo, this variable array is set to be empty.

------------------------------------------------------------------------------------------------------------

#if ADDITIONAL_NODE_ID_SIZE > 0

BYTE AdditionalNodeID[ADDITIONAL_NODE_ID_SIZE] = {SWITCH};

#endif

------------------------------------------------------------------------------------------------------------

// The variable myChannel defines the channel that the P2P connection

// is operate on. This variable will be only effective if energy scan

// (ENABLE_ED_SCAN) is not turned on. Once the energy scan is turned

// on, the operating channel will be one of the channels available with

// least amount of energy (or noise).



------------------------------------------------------------------------------------------------------------

BYTE myChannel = 25;

-----------------------------------------MAIN-----------------------------------------------------

int main(void)

{

        BYTE i;

        BYTE TxNum = 0;

        BYTE RxNum = 0;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// Initialize the system

BoardInit();

ConsoleInit();

LED_1 = 0;

LED_2 = 0;

// Following block display demo information on LCD of Explore 16 or

// PIC18 Explorer demo board.

#if defined(MRF49XA)

        LCDDisplay((char *)"Simple P2P Demo  MRF49XA Node 2", 0, TRUE);

#elif defined(MRF24J40)

        LCDDisplay((char *)"Simple P2P Demo MRF24J40 Node 2", 0, TRUE);

#endif

// Initialize Microchip proprietary protocol. Which protocol to use

// depends on the configuration in ConfigApp.h

MiApp_ProtocolInit();

#if defined(PICDEMZ) || defined(PIC18_EXPLORER)

        INTCONbits.GIEH = 1;

#elif defined(EXPLORER16)

#else
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        #error "Unknown board.  Please initialize board as required."
#endif

Printf("\r\nStarting Node 2 of Simple  for MiWi(TM) P2P Stack ...");
Printf("\r\nInput Configuration:");
Printf("\r\n        Button 1: RD6 on Explorer 16");
Printf("\r\n                  RB5 on PICDEM Z");
Printf("\r\n                  RB0 on PIC18 Explorer");
Printf("\r\n        Button 2: RD7 on Explorer 16");
Printf("\r\n                  RB4 on PICDEM Z");
Printf("\r\n                  RA5 on PIC18 Explorer");
Printf("\r\nOutput Configuration:");
Printf("\r\n                  RS232 port");
Printf("\r\n                  USB on PIC18 Explorer and Explorer 16");
Printf("\r\n          LED 1: D10 on Explorer 16");
Printf("\r\n                  RA0 on PICDEM Z");
Printf("\r\n                  D8 on PIC18 Explorer");
Printf("\r\n          LED 2: D9 on Explorer 16");
Printf("\r\n                  RA1 on PICDEM Z");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
Printf("\r\n                D7 on PIC18 Explorer");

#if defined(MRF24J40)

Printf("\r\n    RF Transceiver: MRF24J40");

#elif defined(MRF49XA)

Printf("\r\n    RF Transceiver: MRF49XA");

#endif

Printf("\r\n   Instruction:");

Printf("\r\n                Power on the board until LED 1 lights up");

Printf("\r\n                to indicate connecting with peer. Push");

Printf("\r\n                Button 1 to broadcast message. Push Button");

Printf("\r\n                2 to unicast encrypted message. LED 2 will");

Printf("\r\n                be toggled upon receiving messages. ");

Printf("\r\n\r\n");

// Set default channel

if( MiApp_SetChannel(myChannel) == FALSE )

{

        Printf("\r\nSelection of channel ");

        PrintDec(myChannel);

        Printf(" is not supported in current condition.\r\n");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        #if defined(__18CXX)
            return;
        #else
            return 0;
        #endif
    }
// Function MiApp_ConnectionMode defines the connection mode. The
// possible connection modes are:
// ENABLE_ALL_CONN:   Enable all kinds of connection
// ENABLE_PREV_CONN:  Only allow connection already exists in
// Connection table
// ENABL_ACTIVE_SCAN_RSP:  Allow response to Active scan
// DISABLE_ALL_CONN:  Disable all connections.
MiApp_ConnectionMode(ENABLE_ALL_CONN);
// Display current opertion on LCD of demo board, if applicable
LCDDisplay((char *)"Connecting Peer  on Channel %d", myChannel, FALSE);
// Function MiApp_EstablishConnection try to establish a new connection
// with peer device.
// The first parameter is the index to the active scan result, which is
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
// acquired by discovery process (active scan). If the value of the

// index is 0xFF, try to establish a connection with any peer.

// The second parameter is the mode to establish connection, either direct

// or indirect. Direct mode means connection within the radio range;

//  Indirect mode means connection may or may not in the radio range.

while( (i = MiApp_EstablishConnection(0xFF, CONN_MODE_DIRECT)) == 0xFF );

// Display current opertion on LCD of demo board, if applicable

LCDDisplay((char *)" Connected Peer  on Channel %d", myChannel, TRUE);

// Function DumpConnection is used to print out the content of the

// P2P Connection Entry on the hyperterminal. It may be useful in

// the debugging phase.

// The only parameter of this function is the index of the P2P

// Connection Entry. The value of 0xFF means to print out all

// valid P2P Connection Entry; otherwise, the P2P Connection Entry

// of the input index will be printed out.

#ifdef ENABLE_DUMP

        DumpConnection(i);

#endif

// Turn on LED 1 to indicate P2P connection established
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
LED_1 = 1;

// Following block display demo instructions on LCD based on the

// demo board used.

#if defined(EXPLORER16)

        LCDDisplay((char *)"RD6: Broadcast  RD7: Unicast", 0, FALSE);

#elif defined(PIC18_EXPLORER)

        LCDDisplay((char *)"RB0: Broadcast  RA5: Unicast", 0, FALSE);

#endif

        while(1)

        {

            WORD_VAL ReceiveData;

            float VoltageOut,T;

            int test;

            char buf[15];

            static char bufs[15];

            if( MiApp_MessageAvailable() )

            {

                ReceiveData.byte.LB = rxMessage.Payload[0];

                ReceiveData.byte.HB = rxMessage.Payload[1];
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            VoltageOut = (3.3/1024.0)*ReceiveData.Val;

            T = (VoltageOut - 0.5)/0.01;

            sprintf(bufs,"\r%.4f\n",T);

            ConsolePutROMString(bufs);

            // Toggle LED2 to indicate receiving a packet.

            LED_2 ^= 1;

            MiApp_DiscardMessage();

            LCDTRXCount(TxNum, ++RxNum);

        }

    }

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# ภาคผนวก ข

# เอกสารคู่มืออุปกรณ์อิเล็กทรอนิกส์

## ข.1 เอกสารคู่มือการใช้งานโมดูล MIWI

ไอซี MRF24J40MA เป็นอุปกรณ์ส่งข้อมูลไร้สายแบบ MIWI โดยมีรายละเอียดสำคัญต่างๆ ดังนี้

MICROCHIP

# MRF24J40MA

## 2.4 GHz IEEE Std. 802.15.4™ RF Transceiver Module

**Features:**

- IEEE Std. 802.15.4™ Compliant RF Transceiver
- Supports ZigBee®, MiWi™, MiWi™ P2P and Proprietary Wireless Networking Protocols
- Small Size: 0.7" x 1.1" (17.8 mm x 27.9 mm), Surface Mountable
- Integrated Crystal, Internal Voltage Regulator, Matching Circuitry and PCB Antenna
- Easy Integration into Final Product – Minimize Product Development, Quicker Time to Market
- Radio Regulation Certification for United States (FCC), Canada (IC) and Europe (ETSI)
- Compatible with Microchip Microcontroller Families (PIC16F, PIC18F, PIC24F/H, dsPIC33 and PIC32)
- Up to 400 ft. Range

**Operational:**

- Operating Voltage: 2.4-3.6V (3.3V typical)
- Temperature Range: -40°C to +85°C Industrial
- Simple, Four-Wire SPI Interface
- Low-Current Consumption:
  - RX mode: 19 mA (typical)
  - TX mode: 23 mA (typical)
  - Sleep: 2 µA (typical)

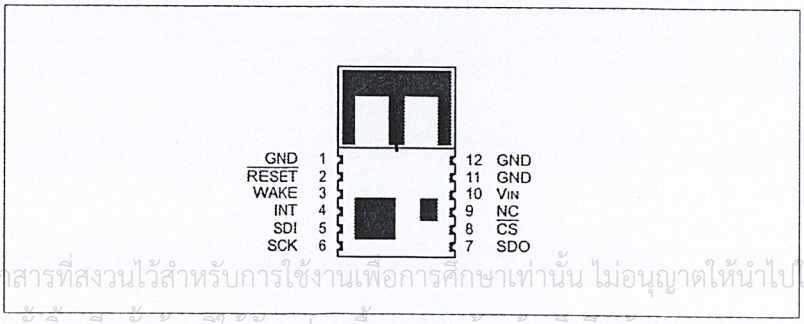
**RF/Analog Features:**

- ISM Band 2.405-2.48 GHz Operation
- Data Rate: 250 kbps
- -94 dBm Typical Sensitivity with +5 dBm Maximum Input Level
- +0 dBm Typical Output Power with 36 dB TX Power Control Range
- Integrated Low Phase Noise VCO, Frequency Synthesizer and PLL Loop Filter
- Digital VCO and Filter Calibration
- Integrated RSSI ADC and I/Q DACs
- Integrated LDO
- High Receiver and RSSI Dynamic Range

**MAC/Baseband Features:**

- Hardware CSMA-CA Mechanism, Automatic ACK Response and FCS Check
- Independent Beacon, Transmit and GTS FIFO
- Supports all CCA modes and RSS/LQI
- Automatic Packet Retransmit Capable
- Hardware Security Engine (AES-128) with CTR, CCM and CBC-MAC modes
- Supports Encryption and Decryption for MAC Sublayer and Upper Layer

**FIGURE 1: PIN DIAGRAM**




เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## 1.0 DEVICE OVERVIEW

The MRF24J40MA is a 2.4 GHz IEEE Std. 802.15.4™ compliant, surface mount module with integrated crystal, internal voltage regulator, matching circuitry and PCB antenna. The MRF24J40MA module operates in the non-licensed 2.4 GHz frequency band and is FCC, IC and ETSI compliant. The integrated module design frees the integrator from extensive RF and antenna design, and regulatory compliance testing, allowing quicker time to market.

The MRF24J40MA module is compatible with Microchip's ZigBee®, MiWi™ and MiWi P2P software stacks. Each software stack is available as a free download, including source code, from the Microchip web site http://www.microchip.com/wireless.

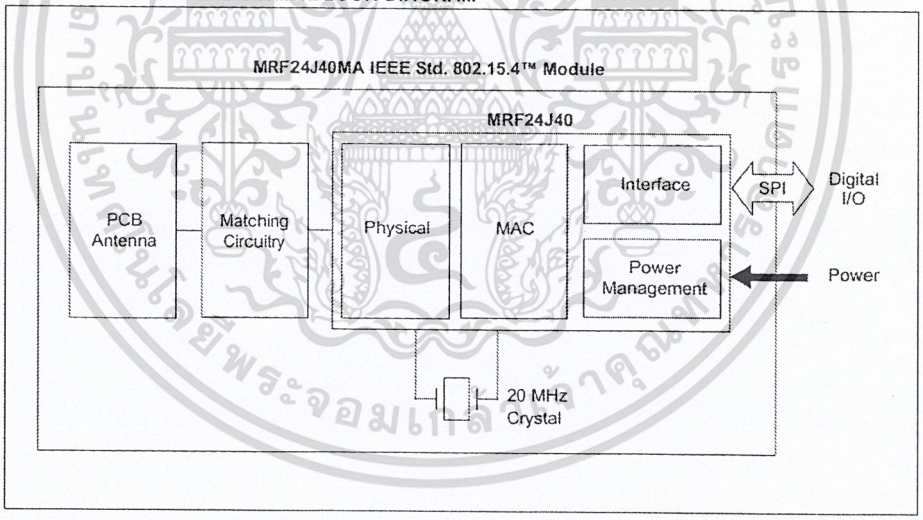
The MRF24J40MA module has received regulatory approvals for modular devices in the United States (FCC), Canada (IC) and Europe (ETSI). Modular approval removes the need for expensive RF and antenna design and allows the end user to place the MRF24J40MA module inside a finished product and not require regulatory testing for an intentional radiator (RF transmitter). See **Section 3.0 "Regulatory Approval"** for specific requirements to be followed by the integrator.

### 1.1 Interface Description

Figure 1-1 shows a simplified block diagram of the MRF24J40MA module. The module is based on the Microchip Technology MRF24J40 IEEE 802.15.4™ 2.4 GHz RF Transceiver IC. The module interfaces to many popular Microchip PIC® microcontrollers via a 4-wire serial SPI interface, interrupt, wake, Reset, power and ground, as shown in Figure 1-2. Table 1-1 provides the pin descriptions.

Data communications with the MRF24J40MA module are documented in the *"MRF24J40 IEEE 802.15.4™ 2.4 GHz RF Transceiver Data Sheet"* (DS39776). Refer to the MRF24J40 Data Sheet for specific serial interface protocol and register definitions.

**FIGURE 1-1: MRF24J40MA BLOCK DIAGRAM**





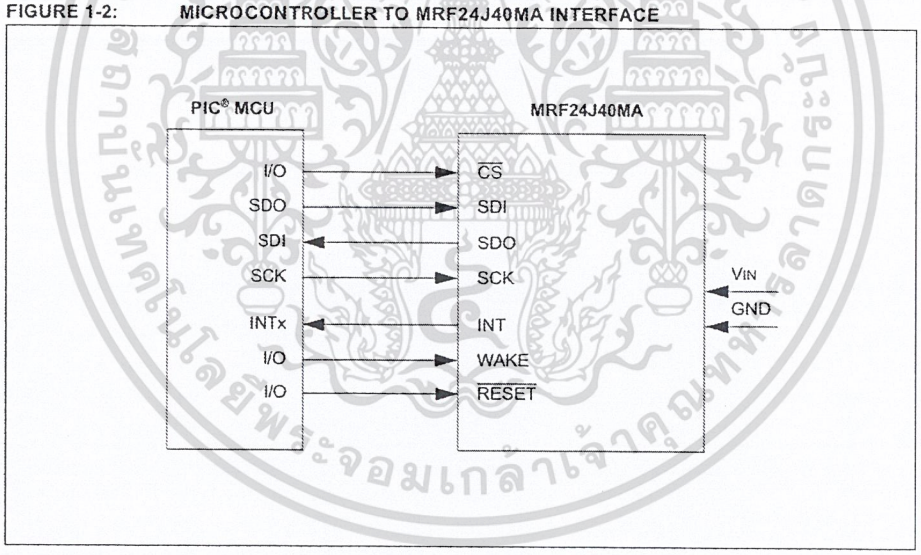

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**TABLE 1-1: PIN DESCRIPTION**

| Pin | Symbol | Type | Description |
|---|---|---|---|
| 1 | GND | Power | Ground |
| 2 | $\overline{\text{RESET}}$ | DI | Global hardware Reset pin |
| 3 | WAKE | DI | External wake-up trigger |
| 4 | INT | DO | Interrupt pin to microcontroller |
| 5 | SDI | DI | Serial interface data input |
| 6 | SCK | DI | Serial interface clock |
| 7 | SDO | DO | Serial interface data output from MRF24J40 |
| 8 | $\overline{\text{CS}}$ | DI | Serial interface enable |
| 9 | NC | — | No connection (allow pin to float; do not connect signal) |
| 10 | VIN | Power | Power supply |
| 11 | GND | Ground | Ground |
| 12 | GND | Ground | Ground |

**Legend:** Pin type abbreviation: D = Digital, I = Input, O = Output

**FIGURE 1-2: MICROCONTROLLER TO MRF24J40MA INTERFACE**






© 2008 Microchip Technology Inc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

## ข.2 เอกสารคู่มือการใช้งานไมโครคอนโทรลเลอร์

เป็นไอซีที่ใช้เขียนโปรแกรมควบคุมการส่งข้อมูล มีรายละเอียดและคุณสมบัติในการใช้งานที่สำคัญต่างๆ ดังนี้

MICROCHIP

# PIC24FJ128GA010 FAMILY

## 64/80/100-Pin General Purpose, 16-Bit Flash Microcontrollers

### High-Performance CPU:

- Modified Harvard Architecture
- Up to 16 MIPS Operation @ 32 MHz
- 8 MHz Internal Oscillator with 4x PLL Option and Multiple Divide Options
- 17-Bit x 17-Bit Single-Cycle Hardware Multiplier
- 32-Bit by 16-Bit Hardware Divider
- 16 x 16-Bit Working Register Array
- C Compiler Optimized Instruction Set Architecture:
  - 76 base instructions
  - Flexible addressing modes
- Two Address Generation Units for Separate Read and Write Addressing of Data Memory

### Special Microcontroller Features:

- Operating Voltage Range of 2.0V to 3.6V
- Flash Program Memory:
  - 1000 erase/write cycles
  - 20-year data retention minimum
- Self-Reprogrammable under Software Control
- Selectable Power Management modes:
  - Sleep, Idle and Alternate Clock modes
- Fail-Safe Clock Monitor Operation:
  - Detects clock failure and switches to on-chip, low-power RC oscillator
- On-Chip 2.5V Regulator
- JTAG Boundary Scan and Programming Support
- Power-on Reset (POR), Power-up Timer (PWRT) and Oscillator Start-up Timer (OST)
- Flexible Watchdog Timer (WDT) with On-Chip, Low-Power RC Oscillator for Reliable Operation
- In-Circuit Serial Programming™ (ICSP™) and In-Circuit Emulation (ICE) via 2 Pins

### Analog Features:

- 10-Bit, Up to 16-Channel Analog-to-Digital Converter
  - 500 ksps conversion rate
  - Conversion available during Sleep and Idle
- Dual Analog Comparators with Programmable Input/Output Configuration

### Peripheral Features:

- Two 3-Wire/4-Wire SPI modules, Supporting 4 Frame modes with 8-Level FIFO Buffer
- Two $I^2C$™ modules Support Multi-Master/Slave mode and 7-Bit/10-Bit Addressing
- Two UART modules:
  - Supports RS-232, RS-485 and LIN 1.2
  - On-chip hardware encoder/decoder for IrDA®
  - Auto-wake-up on Start bit
  - Auto-Baud Detect
  - 4-level FIFO buffer
- Parallel Master Slave Port (PMP/PSP):
  - Supports 8-bit or 16-bit data
  - Supports 16 address lines
- Hardware Real-Time Clock/Calendar (RTCC):
  - Provides clock, calendar and alarm functions
- Programmable Cyclic Redundancy Check (CRC)
  - User-programmable polynomial
  - 8/16-level FIFO buffer
- Five 16-Bit Timers/Counters with Programmable Prescaler
- Five 16-Bit Capture Inputs
- Five 16-Bit Compare/PWM Outputs
- High-Current Sink/Source (18 mA/18 mA) on All I/O Pins
- Configurable, Open-Drain Output on Digital I/O Pins
- Up to 5 External Interrupt Sources
- 5.5V Tolerant Input (digital pins only)

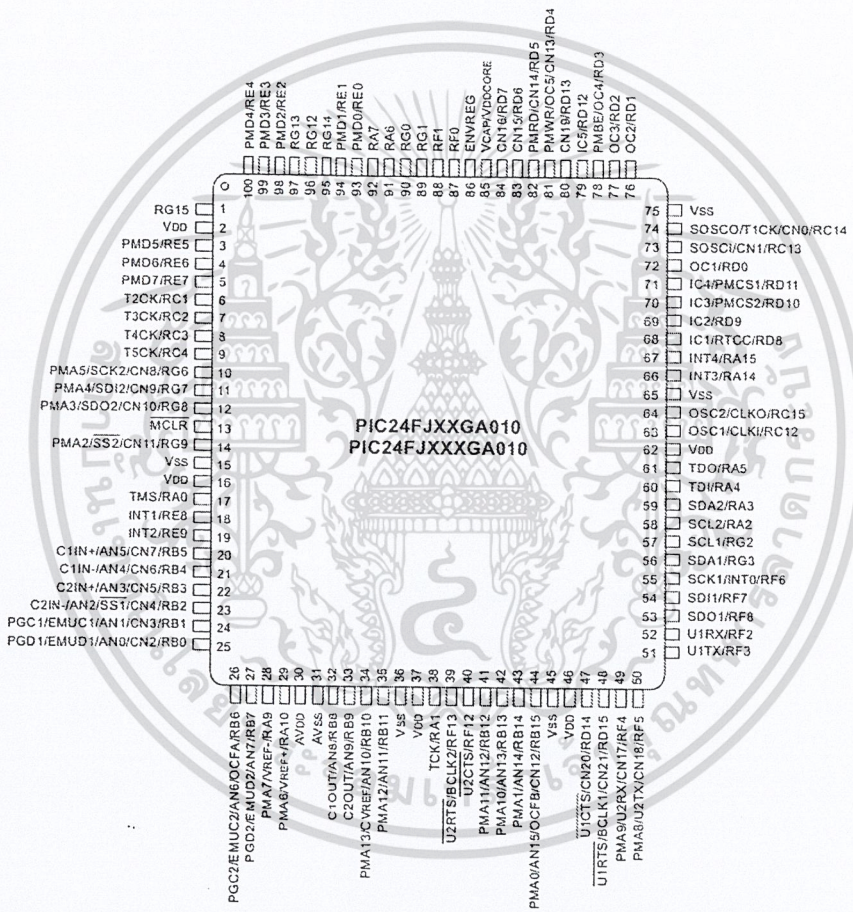
| Device | Pins | Program Memory (Bytes) | SRAM (Bytes) | Timers 16-Bit | Capture Input | Compare/ PWM Output | UART | SPI | $I^2C$™ | 10-Bit A/D (ch) | Comparators | PMP/PSP | JTAG |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PIC24FJ64GA006 | 64 | 64K | 8K | 5 | 5 | 5 | 2 | 2 | 2 | 16 | 2 | Y | Y |
| PIC24FJ96GA006 | 64 | 96K | 8K | 5 | 5 | 5 | 2 | 2 | 2 | 16 | 2 | Y | Y |
| PIC24FJ128GA006 | 64 | 128K | 8K | 5 | 5 | 5 | 2 | 2 | 2 | 16 | 2 | Y | Y |
| PIC24FJ64GA008 | 80 | 64K | 8K | 5 | 5 | 5 | 2 | 2 | 2 | 16 | 2 | Y | Y |
| PIC24FJ96GA008 | 80 | 96K | 8K | 5 | 5 | 5 | 2 | 2 | 2 | 16 | 2 | Y | Y |
| PIC24FJ128GA008 | 80 | 128K | 8K | 5 | 5 | 5 | 2 | 2 | 2 | 16 | 2 | Y | Y |
| PIC24FJ64GA010 | 100 | 64K | 8K | 5 | 5 | 5 | 2 | 2 | 2 | 16 | 2 | Y | Y |
| PIC24FJ96GA010 | 100 | 96K | 8K | 5 | 5 | 5 | 2 | 2 | 2 | 16 | 2 | Y | Y |
| PIC24FJ128GA010 | 100 | 128K | 8K | 5 | 5 | 5 | 2 | 2 | 2 | 16 | 2 | Y | Y |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น การดัดแปลงเนื้อหา และต้องอ้างถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# PIC24FJ128GA010 FAMILY

**Pin Diagrams (Continued))**

**100-Pin TQFP**

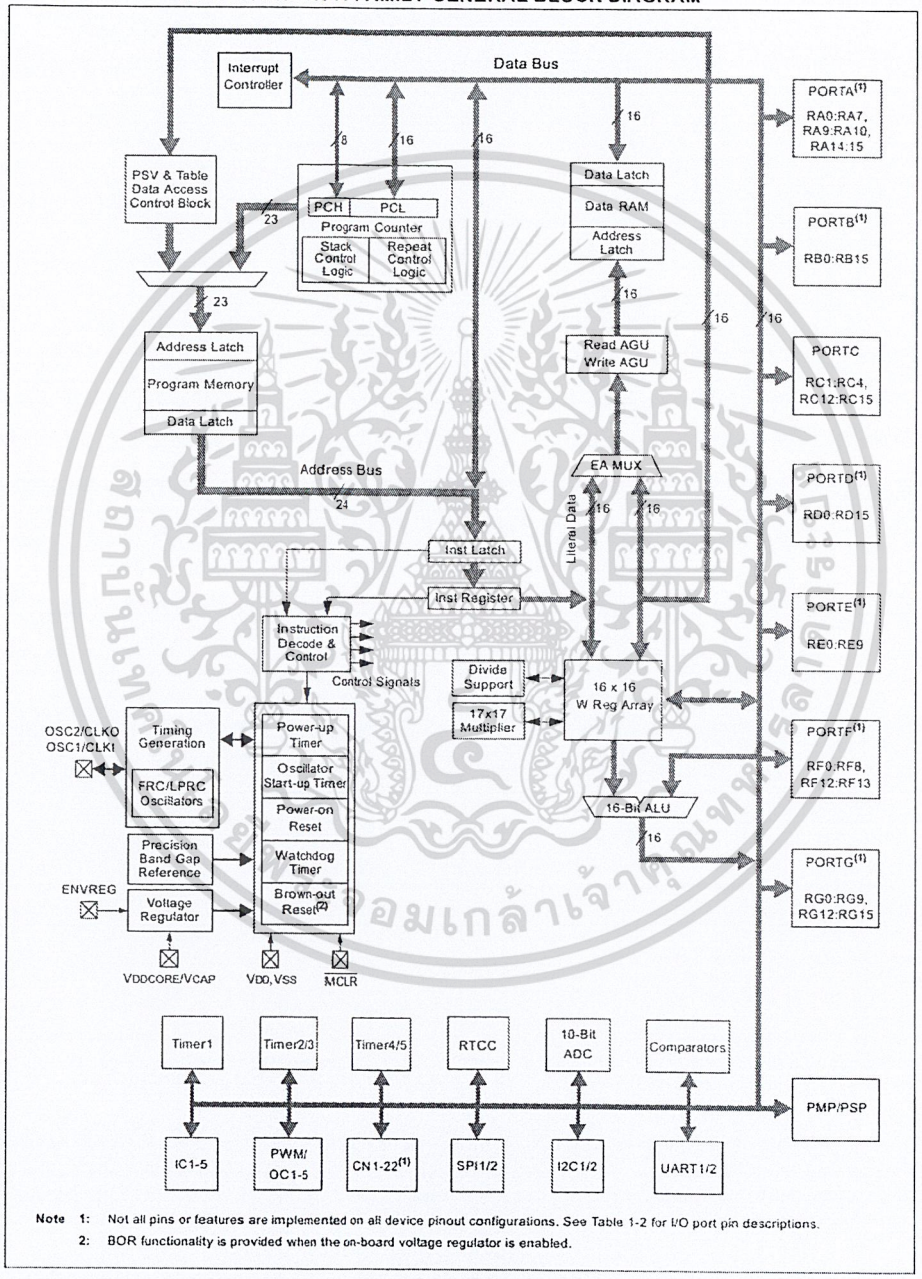



 © 2007 Microchip Technology Inc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# PIC24FJ128GA010 FAMILY

**FIGURE 1-1: PIC24FJ128GA010 FAMILY GENERAL BLOCK DIAGRAM**




**Note 1:** Not all pins or features are implemented on all device pinout configurations. See Table 1-2 for I/O port pin descriptions.

**2:** BOR functionality is provided when the on-board voltage regulator is enabled.


© 2007 Microchip Technology Inc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**TABLE 1-2: PIC24FJ128GA010 FAMILY PINOUT DESCRIPTIONS**

| Function | Pin Number | | | I/O | Input Buffer | Description |
|---|---|---|---|---|---|---|
| | 64-Pin | 80-Pin | 100-Pin | | | |
| AN0 | 16 | 20 | 25 | I | ANA | A/D Analog Inputs. |
| AN1 | 15 | 19 | 24 | I | ANA | |
| AN2 | 14 | 18 | 23 | I | ANA | |
| AN3 | 13 | 17 | 22 | I | ANA | |
| AN4 | 12 | 16 | 21 | I | ANA | |
| AN5 | 11 | 15 | 20 | I | ANA | |
| AN6 | 17 | 21 | 26 | I | ANA | |
| AN7 | 18 | 22 | 27 | I | ANA | |
| AN8 | 21 | 27 | 32 | I | ANA | |
| AN9 | 22 | 28 | 33 | I | ANA | |
| AN10 | 23 | 29 | 34 | I | ANA | |
| AN11 | 24 | 30 | 35 | I | ANA | |
| AN12 | 27 | 33 | 41 | I | ANA | |
| AN13 | 28 | 34 | 42 | I | ANA | |
| AN14 | 29 | 35 | 43 | I | ANA | |
| AN15 | 30 | 36 | 44 | I | ANA | |
| AVDD | 19 | 25 | 30 | P | — | Positive Supply for Analog Modules. |
| AVSS | 20 | 26 | 31 | P | — | Ground Reference for Analog Modules. |
| BCLK1 | 35 | 38 | 48 | O | — | UART1 IrDA® Baud Clock. |
| BCLK2 | 29 | 35 | 39 | O | — | UART2 IrDA® Baud Clock. |
| C1IN- | 12 | 16 | 21 | I | ANA | Comparator 1 Negative Input. |
| C1IN+ | 11 | 15 | 20 | I | ANA | Comparator 1 Positive Input. |
| C1OUT | 21 | 27 | 32 | O | — | Comparator 1 Output. |
| C2IN- | 14 | 18 | 23 | I | ANA | Comparator 2 Negative Input. |
| C2IN+ | 13 | 17 | 22 | I | ANA | Comparator 2 Positive Input. |
| C2OUT | 22 | 28 | 33 | O | — | Comparator 2 Output. |
| CLKI | 39 | 49 | 63 | I | ANA | Main Clock Input Connection. |
| CLKO | 40 | 50 | 64 | O | — | System Clock Output. |
| CN0 | 48 | 60 | 74 | I | ST | Interrupt-on-Change Inputs. |
| CN1 | 47 | 59 | 73 | I | ST | |
| CN2 | 16 | 20 | 25 | I | ST | |
| CN3 | 15 | 19 | 24 | I | ST | |
| CN4 | 14 | 18 | 23 | I | ST | |
| CN5 | 13 | 17 | 22 | I | ST | |
| CN6 | 12 | 16 | 21 | I | ST | |
| CN7 | 11 | 15 | 20 | I | ST | |
| CN8 | 4 | 6 | 10 | I | ST | |
| CN9 | 5 | 7 | 11 | I | ST | |
| CN10 | 6 | 8 | 12 | I | ST | |
| CN11 | 8 | 10 | 14 | I | ST | |
| CN12 | 30 | 36 | 44 | I | ST | |
| CN13 | 52 | 66 | 81 | I | ST | |
| CN14 | 53 | 67 | 82 | I | ST | |
| CN15 | 54 | 68 | 83 | I | ST | |
| CN16 | 55 | 69 | 84 | I | ST | |
| CN17 | 31 | 39 | 49 | I | ST | |

**Legend:** TTL = TTL input buffer
ANA = Analog level input/output
ST = Schmitt Trigger input buffer
$I^2C$™ = $I^2C$/SMBus input buffer



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# PIC24FJ128GA010 FAMILY

**TABLE 1-2: PIC24FJ128GA010 FAMILY PINOUT DESCRIPTIONS (CONTINUED)**

| Function | Pin Number | | | I/O | Input Buffer | Description |
|---|---|---|---|---|---|---|
| | 64-Pin | 80-Pin | 100-Pin | | | |
| CN18 | 32 | 40 | 50 | I | ST | Interrupt-on-Change Inputs. |
| CN19 | — | 65 | 80 | I | ST | |
| CN20 | — | 37 | 47 | I | ST | |
| CN21 | — | 38 | 48 | I | ST | |
| CVREF | 23 | 29 | 34 | O | ANA | Comparator Voltage Reference Output. |
| EMUC1 | 15 | 19 | 24 | I/O | ST | In-Circuit Emulator Clock Input/Output. |
| EMUD1 | 16 | 20 | 25 | I/O | ST | In-Circuit Emulator Data Input/Output. |
| EMUC2 | 17 | 21 | 26 | I/O | ST | In-Circuit Emulator Clock Input/Output. |
| EMUD2 | 18 | 22 | 27 | I/O | ST | In-Circuit Emulator Data Input/Output. |
| ENVREG | 57 | 71 | 86 | I | ST | Enable for On-Chip Voltage Regulator. |
| IC1 | 42 | 54 | 68 | I | ST | Input Capture Inputs. |
| IC2 | 43 | 55 | 69 | I | ST | |
| IC3 | 44 | 56 | 70 | I | ST | |
| IC4 | 45 | 57 | 71 | I | ST | |
| IC5 | 52 | 64 | 79 | I | ST | |
| INT0 | 35 | 45 | 55 | I | ST | External Interrupt Inputs. |
| INT1 | 42 | 13 | 18 | I | ST | |
| INT2 | 43 | 14 | 19 | I | ST | |
| INT3 | 44 | 52 | 66 | I | ST | |
| INT4 | 45 | 53 | 67 | I | ST | |
| $\overline{MCLR}$ | 7 | 9 | 13 | I | ST | Master Clear (Device Reset) Input. This line is brought low to cause a Reset. |
| OC1 | 46 | 58 | 72 | O | — | Output Compare/PWM Outputs. |
| OC2 | 49 | 61 | 76 | O | — | |
| OC3 | 50 | 62 | 77 | O | — | |
| OC4 | 51 | 63 | 78 | O | — | |
| OC5 | 52 | 66 | 81 | O | — | |
| OCFA | 17 | 21 | 26 | I | ST | Output Compare Fault A Input. |
| OCFB | 30 | 36 | 44 | I | ST | Output Compare Fault B Input. |
| OSC1 | 39 | 49 | 63 | I | ANA | Main Oscillator Input Connection. |
| OSC2 | 40 | 50 | 64 | O | ANA | Main Oscillator Output Connection. |
| PGC1 | 15 | 19 | 24 | I/O | ST | In-Circuit Debugger and ICSP™ Programming Clock. |
| PGD1 | 16 | 20 | 25 | I/O | ST | In-Circuit Debugger and ICSP Programming Data. |
| PGC2 | 17 | 21 | 26 | I/O | ST | In-Circuit Debugger and ICSP™ Programming Clock. |
| PGD2 | 18 | 22 | 27 | I/O | ST | In-Circuit Debugger and ICSP Programming Data. |

**Legend:** TTL = TTL input buffer ST = Schmitt Trigger input buffer
ANA = Analog level input/output $I^2C$™ = $I^2C$/SMBus input buffer


© 2007 Microchip Technology Inc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**TABLE 1-2: PIC24FJ128GA010 FAMILY PINOUT DESCRIPTIONS (CONTINUED)**

| Function | Pin Number | | | I/O | Input Buffer | Description |
|---|---|---|---|---|---|---|
| | 64-Pin | 80-Pin | 100-Pin | | | |
| PMA0 | 30 | 36 | 44 | I/O | ST/TTL | Parallel Master Port Address Bit 0 Input (Buffered Slave modes) and Output (Master modes). |
| PMA1 | 29 | 35 | 43 | I/O | ST/TTL | Parallel Master Port Address Bit 1 Input (Buffered Slave modes) and Output (Master modes). |
| PMA2 | 8 | 10 | 14 | O | — | Parallel Master Port Address (Demultiplexed Master modes). |
| PMA3 | 6 | 8 | 12 | O | — | |
| PMA4 | 5 | 7 | 11 | O | — | |
| PMA5 | 4 | 6 | 10 | O | — | |
| PMA6 | 16 | 24 | 29 | O | — | |
| PMA7 | 22 | 23 | 28 | O | — | |
| PMA8 | 32 | 40 | 50 | O | — | |
| PMA9 | 31 | 39 | 49 | O | — | |
| PMA10 | 28 | 34 | 42 | O | — | |
| PMA11 | 27 | 33 | 41 | O | — | |
| PMA12 | 24 | 30 | 35 | O | — | |
| PMA13 | 23 | 29 | 34 | O | — | |
| PMBE | 51 | 63 | 78 | O | — | Parallel Master Port Byte Enable Strobe. |
| PMCS1 | 45 | 57 | 71 | I/O | ST/TTL | Parallel Master Port Chip Select 1 Strobe/Address bit 14. |
| PMCS2 | 44 | 56 | 70 | O | — | Parallel Master Port Chip Select 2 Strobe/Address bit 15. |
| PMD0 | 60 | 76 | 93 | I/O | ST/TTL | Parallel Master Port Data (Demultiplexed Master mode) or Address/Data (Multiplexed Master modes). |
| PMD1 | 61 | 77 | 94 | I/O | ST/TTL | |
| PMD2 | 62 | 78 | 98 | I/O | ST/TTL | |
| PMD3 | 63 | 79 | 99 | I/O | ST/TTL | |
| PMD4 | 64 | 80 | 100 | I/O | ST/TTL | |
| PMD5 | 1 | 1 | 3 | I/O | ST/TTL | |
| PMD6 | 2 | 2 | 4 | I/O | ST/TTL | |
| PMD7 | 3 | 3 | 5 | I/O | ST/TTL | |
| PMRD | 53 | 67 | 82 | I/O | ST/TTL | Parallel Master Port Read Strobe. |
| PMWR | 52 | 66 | 81 | I/O | ST/TTL | Parallel Master Port Write Strobe. |

**Legend:** TTL = TTL input buffer ST = Schmitt Trigger input buffer
ANA = Analog level input/output $I^2C$™ = $I^2C$/SMBus input buffer



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# PIC24FJ128GA010 FAMILY

**TABLE 1-2: PIC24FJ128GA010 FAMILY PINOUT DESCRIPTIONS (CONTINUED)**

| Function | Pin Number | | | I/O | Input Buffer | Description |
|---|---|---|---|---|---|---|
| | 64-Pin | 80-Pin | 100-Pin | | | |
| RA0 | — | — | 17 | I/O | ST | PORTA Digital I/O. |
| RA1 | — | — | 38 | I/O | ST | |
| RA2 | — | — | 58 | I/O | ST | |
| RA3 | — | — | 59 | I/O | ST | |
| RA4 | — | — | 60 | I/O | ST | |
| RA5 | — | — | 61 | I/O | ST | |
| RA6 | — | — | 91 | I/O | ST | |
| RA7 | — | — | 92 | I/O | ST | |
| RA9 | — | 23 | 28 | I/O | ST | |
| RA10 | — | 24 | 29 | I/O | ST | |
| RA14 | — | 52 | 66 | I/O | ST | |
| RA15 | — | 53 | 67 | I/O | ST | |
| RB0 | 16 | 20 | 25 | I/O | ST | PORTB Digital I/O. |
| RB1 | 15 | 19 | 24 | I/O | ST | |
| RB2 | 14 | 18 | 23 | I/O | ST | |
| RB3 | 13 | 17 | 22 | I/O | ST | |
| RB4 | 12 | 16 | 21 | I/O | ST | |
| RB5 | 11 | 15 | 20 | I/O | ST | |
| RB6 | 17 | 21 | 26 | I/O | ST | |
| RB7 | 18 | 22 | 27 | I/O | ST | |
| RB8 | 21 | 27 | 32 | I/O | ST | |
| RB9 | 22 | 28 | 33 | I/O | ST | |
| RB10 | 23 | 29 | 34 | I/O | ST | |
| RB11 | 24 | 30 | 35 | I/O | ST | |
| RB12 | 27 | 33 | 41 | I/O | ST | |
| RB13 | 28 | 34 | 42 | I/O | ST | |
| RB14 | 29 | 35 | 43 | I/O | ST | |
| RB15 | 30 | 36 | 44 | I/O | ST | |
| RC1 | — | 4 | 6 | I/O | ST | PORTC Digital I/O. |
| RC2 | — | — | 7 | I/O | ST | |
| RC3 | — | 5 | 8 | I/O | ST | |
| RC4 | — | — | 9 | I/O | ST | |
| RC12 | 39 | 49 | 63 | I/O | ST | |
| RC13 | 47 | 59 | 73 | I/O | ST | |
| RC14 | 48 | 60 | 74 | I/O | ST | |
| RC15 | 40 | 50 | 64 | I/O | ST | |

**Legend:** TTL = TTL input buffer ST = Schmitt Trigger input buffer
ANA = Analog level input/output $I^2C$™ = $I^2C$/SMBus input buffer


© 2007 Microchip Technology Inc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**TABLE 1-2: PIC24FJ128GA010 FAMILY PINOUT DESCRIPTIONS (CONTINUED)**

| Function | Pin Number | | | I/O | Input Buffer | Description |
|---|---|---|---|---|---|---|
| | 64-Pin | 80-Pin | 100-Pin | | | |
| RD0 | 46 | 58 | 72 | I/O | ST | PORTD Digital I/O. |
| RD1 | 49 | 61 | 76 | I/O | ST | |
| RD2 | 50 | 62 | 77 | I/O | ST | |
| RD3 | 51 | 63 | 78 | I/O | ST | |
| RD4 | 52 | 66 | 81 | I/O | ST | |
| RD5 | 53 | 67 | 82 | I/O | ST | |
| RD6 | 54 | 68 | 83 | I/O | ST | |
| RD7 | 55 | 69 | 84 | I/O | ST | |
| RD8 | 42 | 54 | 68 | I/O | ST | |
| RD9 | 43 | 55 | 69 | I/O | ST | |
| RD10 | 44 | 56 | 70 | I/O | ST | |
| RD11 | 45 | 57 | 71 | I/O | ST | |
| RD12 | — | 64 | 79 | I/O | ST | |
| RD13 | — | 65 | 80 | I/O | ST | |
| RD14 | — | 37 | 47 | I/O | ST | |
| RD15 | — | 38 | 48 | I/O | ST | |
| RE0 | 60 | 76 | 93 | I/O | ST | PORTE Digital I/O. |
| RE1 | 61 | 77 | 94 | I/O | ST | |
| RE2 | 62 | 78 | 98 | I/O | ST | |
| RE3 | 63 | 79 | 99 | I/O | ST | |
| RE4 | 64 | 80 | 100 | I/O | ST | |
| RE5 | 1 | 1 | 3 | I/O | ST | |
| RE6 | 2 | 2 | 4 | I/O | ST | |
| RE7 | 3 | 3 | 5 | I/O | ST | |
| RE8 | — | 13 | 18 | I/O | ST | |
| RE9 | — | 14 | 19 | I/O | ST | |
| RF0 | 58 | 72 | 87 | I/O | ST | PORTF Digital I/O. |
| RF1 | 59 | 73 | 88 | I/O | ST | |
| RF2 | 34 | 42 | 52 | I/O | ST | |
| RF3 | 33 | 41 | 51 | I/O | ST | |
| RF4 | 31 | 39 | 49 | I/O | ST | |
| RF5 | 32 | 40 | 50 | I/O | ST | |
| RF6 | 35 | 45 | 55 | I/O | ST | |
| RF7 | — | 44 | 54 | I/O | ST | |
| RF8 | — | 43 | 53 | I/O | ST | |
| RF12 | — | — | 40 | I/O | ST | |
| RF13 | — | — | 39 | I/O | ST | |

**Legend:** TTL = TTL input buffer ST = Schmitt Trigger input buffer
ANA = Analog level input/output $I^2C$™ = $I^2C$/SMBus input buffer



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# PIC24FJ128GA010 FAMILY

**TABLE 1-2: PIC24FJ128GA010 FAMILY PINOUT DESCRIPTIONS (CONTINUED)**

| Function | Pin Number | | | I/O | Input Buffer | Description |
|---|---|---|---|---|---|---|
| | 64-Pin | 80-Pin | 100-Pin | | | |
| RG0 | — | 75 | 90 | I/O | ST | PORTG Digital I/O. |
| RG1 | — | 74 | 89 | I/O | ST | |
| RG2 | 37 | 47 | 57 | I/O | ST | |
| RG3 | 36 | 46 | 56 | I/O | ST | |
| RG6 | 4 | 6 | 10 | I/O | ST | |
| RG7 | 5 | 7 | 11 | I/O | ST | |
| RG8 | 6 | 8 | 12 | I/O | ST | |
| RG9 | 8 | 10 | 14 | I/O | ST | |
| RG12 | — | — | 96 | I/O | ST | |
| RG13 | — | — | 97 | I/O | ST | |
| RG14 | — | — | 95 | I/O | ST | |
| RG15 | — | — | 1 | I/O | ST | |
| RTCC | 42 | 54 | 68 | O | — | Real-Time Clock Alarm Output. |
| SCK1 | 35 | 45 | 55 | O | — | SPI1 Serial Clock Output. |
| SCK2 | 4 | 6 | 10 | I/O | ST | SPI2 Serial Clock Output. |
| SCL1 | 37 | 47 | 57 | I/O | $I^2C$ | I2C1 Synchronous Serial Clock Input/Output. |
| SCL2 | 32 | 52 | 58 | I/O | $I^2C$ | I2C2 Synchronous Serial Clock Input/Output. |
| SDA1 | 36 | 46 | 56 | I/O | $I^2C$ | I2C1 Data Input/Output. |
| SDA2 | 31 | 53 | 59 | I/O | $I^2C$ | I2C2 Data Input/Output. |
| SDI1 | 34 | 44 | 54 | I | ST | SPI1 Serial Data Input. |
| SDI2 | 5 | 7 | 11 | I | ST | SPI2 Serial Data Input. |
| SDO1 | 33 | 43 | 53 | O | — | SPI1 Serial Data Output. |
| SDO2 | 6 | 8 | 12 | O | — | SPI2 Serial Data Output. |
| SOSCI | 47 | 59 | 73 | I | ANA | Secondary Oscillator/Timer1 Clock Input. |
| SOSCO | 48 | 60 | 74 | O | ANA | Secondary Oscillator/Timer1 Clock Output. |
| $\overline{SS1}$ | 14 | 18 | 23 | I/O | ST | Slave Select Input/Frame Select Output (SPI1). |
| $\overline{SS2}$ | 8 | 10 | 14 | I/O | ST | Slave Select Input/Frame Select Output (SPI2). |
| T1CK | 48 | 60 | 74 | I | ST | Timer1 Clock. |
| T2CK | — | 4 | 6 | I | ST | Timer2 External Clock Input. |
| T3CK | — | — | 7 | I | ST | Timer3 External Clock Input. |
| T4CK | — | 5 | 8 | I | ST | Timer4 External Clock Input. |
| T5CK | — | — | 9 | I | ST | Timer5 External Clock Input. |
| TCK | 27 | 33 | 38 | I | ST | JTAG Test Clock/Programming Clock Input. |
| TDI | 28 | 34 | 60 | I | ST | JTAG Test Data/Programming Data Input. |
| TDO | 24 | 14 | 61 | O | — | JTAG Test Data Output. |
| TMS | 23 | 13 | 17 | I | ST | JTAG Test Mode Select Input. |

**Legend:** TTL = TTL input buffer ST = Schmitt Trigger input buffer
ANA = Analog level input/output $I^2C$™ = $I^2C$/SMBus input buffer

 © 2007 Microchip Technology Inc.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

**TABLE 1-2: PIC24FJ128GA010 FAMILY PINOUT DESCRIPTIONS (CONTINUED)**

| Function | Pin Number | | | I/O | Input Buffer | Description |
|---|---|---|---|---|---|---|
| | 64-Pin | 80-Pin | 100-Pin | | | |
| U1CTS | 43 | 37 | 47 | I | ST | UART1 Clear to Send Input. |
| U1RTS | 35 | 38 | 48 | O | — | UART1 Request to Send Output. |
| U1RX | 34 | 42 | 52 | I | ST | UART1 Receive. |
| U1TX | 33 | 41 | 51 | O | DIG | UART1 Transmit Output. |
| U2CTS | 21 | 27 | 40 | I | ST | UART2 Clear to Send Input. |
| U2RTS | 29 | 35 | 39 | O | — | UART2 Request to Send Output. |
| U2RX | 31 | 39 | 49 | I | ST | UART 2 Receive Input. |
| U2TX | 32 | 40 | 50 | O | — | UART2 Transmit Output. |
| VDD | 10, 26, 38 | 12, 32, 48 | 2, 16, 37, 46, 62 | P | — | Positive Supply for Peripheral Digital Logic and I/O Pins. |
| VDDCAP | 56 | 70 | 85 | P | — | External Filter Capacitor Connection (regulator enabled). |
| VDDCORE | 56 | 70 | 85 | P | — | Positive Supply for Microcontroller Core Logic (regulator disabled). |
| VREF- | 15 | 23 | 28 | I | ANA | A/D and Comparator Reference Voltage (Low) Input. |
| VREF+ | 16 | 24 | 29 | I | ANA | A/D and Comparator Reference Voltage (High) Input. |
| VSS | 9, 25, 41 | 11, 31, 51 | 15, 36, 45, 65, 75 | P | — | Ground Reference for Logic and I/O Pins. |

**Legend:** TTL = TTL input buffer
ANA = Analog level input/output
ST = Schmitt Trigger input buffer
$I^2C$™ = $I^2C$/SMBus input buffer



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

# เอกสารอ้างอิง

[1] ประจิน พลังสันติกุล. **dsPIC30F Programming กับ MPLAB C คอมไพเลอร์.** กรุงเทพฯ : แอพซอฟต์เทค. 2551.

[2] บัณฑิต จามรภูติ. **คู่มือการใช้งาน Protel DXP [Altium Designer].** เชียงใหม่ : บัณฑิตเพรส. 2550.

[3] Microchip. " **MiWi.**" [Online]. Available : http://www.microchip.com. 2011.

[4] Electoday 3.0. " **Microcontroller PIC.** " [Online]. Available : http://www.electoday.com.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้