

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

TCP/IP – ZigBee Gateway

TCP/IP – ZigBee เกตเวย์

โดย

นายชวพล กุลเวชกิจ

นางสาววิรัตน์ บุรพาเดชะ

รพ.
82767
2550

เลขหมู่.....
83075
เลขทะเบียน.....
- 5 ส.ค. 2551
วัน,เดือน,ปี.....

b. 11964194
i.....

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

ภาควิชาวิศวกรรมสารสนเทศ

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2550

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

TCP/IP – ZIGBEE GATEWAY



**A PROJECT SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENT FOR THE DEGREE OF
BACHELOR IN DEPARTMENT OF INFORMATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG**

2007

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อปริญญานิพนธ์ TCP/IP – ZigBee เกตเวย์
TCP/IP – ZigBee Gateway
ชื่อนักศึกษา นายชวพล กุลเวชกิจ รหัสประจำตัว 47010155
นางสาวชวิรัตน์ บุรพาเดชะ รหัสประจำตัว 47010158
อาจารย์ที่ปรึกษา ผศ. มนต์ชัย แซ่มซ้อย
ระดับการศึกษา ปริญญาตรี วิศวกรรมศาสตรบัณฑิต
สาขาวิศวกรรมสารสนเทศ
ภาควิชา วิศวกรรมสารสนเทศ
ปีการศึกษา 2550

ปริญญานิพนธ์ฉบับนี้ได้รับความเห็นชอบจากอาจารย์ที่ปรึกษาเป็นที่เรียบร้อยแล้ว

(ผศ. มนต์ชัย แซ่มซ้อย)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัวข้อวิทยานิพนธ์	TCP/IP – ZigBee เกตเวย์		
ชื่อนักศึกษา	นายชวพล	กุลเวชกิจ	รหัสประจำตัว 47010155
	นางสาววริรัตน์	นุรพาเดชะ	รหัสประจำตัว 47010158
อาจารย์ที่ปรึกษา	ผศ. มนต์ชัย แซ่มซ้อย		
ระดับการศึกษา	ปริญญาตรี วิศวกรรมศาสตรบัณฑิต		
	สาขาวิศวกรรมสารสนเทศ		
ภาควิชา	วิศวกรรมสารสนเทศ		
ปีการศึกษา	2550		

บทคัดย่อ

เนื่องจากในปัจจุบันนี้เทคโนโลยีและการติดต่อสื่อสารยิ่งทวีความสำคัญมากขึ้น อีกทั้งยังมีการใช้เทคโนโลยีต่างๆที่ทันสมัยมากมาย และเทคโนโลยี ZigBee คือการสื่อสารแบบไร้สายเป็นมาตรฐาน IEEE 802.15.4 มีค่าใช้จ่ายต่ำ ใช้พลังงานน้อย อัตราการส่งข้อมูลไม่มากนัก แต่มีความน่าเชื่อถือในการส่งข้อมูลสูง ในโครงการนี้ได้นำเสนอ TCP/IP – ZigBee เกตเวย์ ทำหน้าที่ในการเชื่อมต่อระหว่างโปรโตคอล TCP/IP กับโปรโตคอล ZigBee เพื่อให้สามารถติดต่อโมดูล ZigBee ผ่านทาง TCP/IP ได้ ซึ่งแต่ละโปรโตคอลมีรูปแบบการสื่อสารที่แตกต่างกัน ทำให้ต้องสร้างเกตเวย์เพื่อให้โปรโตคอลที่มีความแตกต่างกันสามารถติดต่อสื่อสารกันได้ โดยใช้ไมโครคอนโทรลเลอร์ PIC18F86J65 ซึ่งเป็นไมโครคอนโทรลเลอร์ในตระกูล 18F97J60 ของบริษัทไมโครชิป เป็นตัวจัดการโปรโตคอลที่ไม่เหมือนกัน เพื่อสร้างเกตเวย์เชื่อมต่อระหว่างโปรโตคอล TCP/IP กับ ZigBee ทำให้การใช้งาน ZigBee มีความสะดวกสบายมากขึ้น ง่ายต่อการควบคุม และนำไปพัฒนาโปรแกรมประยุกต์การใช้งานผ่านอินเทอร์เน็ตให้มีความหลากหลายได้มากยิ่งขึ้น

Thesis Title	TCP/IP – ZigBee Gateway		
Student	Mr. Chawapon	Kulvechakij	ID. 47010155
	Miss Chavirat	Burapadecha	ID. 47010158
Advisor	Asst.Prof. Monchai Chamchoy		
Graduate Level	Bachelor Degree of Information Engineering		
Department	Information Engineering		
Academic Year	2007		

Abstract

Resulting, Nowadays communication technologies have much more important and new technology has been growing up. The ZigBee is wireless technology and the IEEE 802.15.4 standard. The ZigBee specification is a standard that defines a stack protocol enabling the interoperability of wireless devices in a low-cost, low-power consumption, low-data-rate network and high data transmission reliability between devices. This project's present TCP/IP – ZigBee Gateway which assign connect between protocol TCP/IP and protocol ZigBee which each protocol is different perform communication. So building gateway to use communication by using a microcontroller PIC18F86J65 is controller in PIC18F97J60 family by Microchip and controller Zigbee to make a connection gateway between TCP/IP and ZigBee protocol. Using ZigBee has more comfortable, easier to control and can develop more variety applications to use ZigBee protocol via Ethernet.

กิตติกรรมประกาศ

การจัดทำปฏิญานិพนธ์ฉบับนี้จะไม่สามารถสำเร็จลุล่วงไปด้วยดีได้เลย ถ้าไม่ได้รับการช่วยเหลือและขอขอบคุณ ผศ. มนต์ชัย แซ่มซ้อย ซึ่งเป็นอาจารย์ที่ปรึกษาโครงการที่ให้โอกาสในการทำโครงการนี้กับข้าพเจ้า และยังช่วยให้คำปรึกษา การแนะนำขั้นตอนการทำงานอย่างเป็นระบบ แนวทางในการหาข้อมูลที่ใช้ประกอบในรายงาน และ การจัดรูปแบบของรายงานที่ถูกต้อง ตลอดจนรูปแบบการนำเสนอโครงการที่ดี และทุกสิ่งทุกอย่างที่อาจารย์พูดและสอน ความทุ่มเทที่มีให้กับนักศึกษาอย่างเต็มที่ ผู้จัดทำรู้สึกซาบซึ้งและขอกราบขอบพระคุณเป็นอย่างสูง

ขอขอบคุณทางภาควิชาวิศวกรรมสารสนเทศที่หล่อหลอมให้ผู้จัดทำทั้งสองคนจนมาถึงจุดนี้ได้ มีความรู้ความสามารถทำให้เกิดความกระฉ่างในเส้นทางของการดำเนินชีวิตที่ชัดเจนยิ่งขึ้น ที่ขาดไม่ได้เลยก็คือเหล่าคณาจารย์ในภาควิชาทุกท่านที่ทุ่มเททำการสอนวิชาต่าง ๆ ด้วยดีตลอดมา เมื่อมีปัญหาใด ๆ ถึงแม้ทางภาควิชาจะไม่ได้จัดให้มีอาจารย์ที่ปรึกษาอย่างชัดเจน ตัวของนักศึกษาเองก็สามารถสอบถามเรื่องราวและข้อสงสัยต่าง ๆ ได้จากทุกท่าน

ขอขอบคุณ เพื่อนๆ พี่ๆ ทุกคนที่ให้คำแนะนำที่ดี ทั้งในเรื่องการทำโครงการนี้ และการแบ่งเวลาในการทำโครงการ และยังคงเป็นกำลังใจให้ข้าพเจ้าให้การทำโครงการครั้งนี้ ให้ผ่านช่วงเวลาที่ยากลำบาก ให้สำเร็จไปได้ด้วยดี

และที่สำคัญที่สุดคือ กำลังใจและความห่วงใยจากครอบครัวของผู้จัดทำทั้งสองคนที่มีเสมอมา ไม่เคยขาด ขอขอบคุณบิดา มารดาและญาติ ๆ ในทุก ๆ เรื่อง รวมไปถึงการสนับสนุนด้านการเงินในเรื่องเรียนตั้งแต่จบจนปีสุดท้ายของการเรียนที่คณะวิศวกรรมศาสตร์ในสถาบันแห่งนี้

สุดท้าย ผู้จัดทำ(อีกคน) ขอขอบคุณที่รวมทุกข์ร่วมสุข ฝ่าฟันอุปสรรคต่างๆในการทำงานมาด้วยกัน การทำงานร่วมกันครั้งนี้จะเป็นประสบการณ์ที่ยิ่งใหญ่ และความทรงจำที่ดีตลอดไป หากไม่มีทุกคนที่กล่าวมาทั้งหมดนี้ โครงการนี้ก็คงมีอาจสำเร็จลุล่วงไปได้ด้วยดี ตรงตามจุดประสงค์ที่แท้จริงของโครงการได้

คณะผู้จัดทำ

สารบัญ

	หน้า
บทคัดย่อภาษาไทย	ก
บทคัดย่อภาษาอังกฤษ	ข
สารบัญ	ค
สารบัญรูป	จ
สารบัญตาราง	ช
บทที่ 1 บทนำ	
1.1 ความสำคัญและที่มาของปัญหา	1
1.2 วัตถุประสงค์ของโครงการ	1
1.3 ขอบเขตของโครงการ	2
1.4 ขั้นตอนการดำเนินงาน	3
บทที่ 2 ทฤษฎีและหลักการ	
2.1 สเตกโปรโตคอล TCP/IP	4
2.1.1 ชั้นแอปพลิเคชัน	6
2.1.2 ชั้นทรานสปอร์ต	7
2.1.3 ชั้นอินเทอร์เน็ตเวิร์ค	12
2.1.4 ชั้นเน็ตเวิร์คอินเทอร์เน็ตเฟส	15
2.1.5 กลไกของโปรโตคอล IP	17
2.2 หลักการพื้นฐานของ ZigBee	18
2.2.1 IEEE 802.15.4 กับ ZigBee	18
2.2.2 จุดเด่นของ ZigBee	19
2.2.3 การตั้งค่าเครือข่าย	23
2.2.4 โพรไฟล์โปรโตคอล ZigBee	25
2.2.5 ZigBee โปรโตคอลสเตก	26
2.2.6 การรักษาความปลอดภัย	28
2.2.7 การติดต่อสื่อสารของ ZigBee ผ่านทางพอร์ตอนุกรม	29
2.3 หลักการทำงานไมโครคอนโทรลเลอร์ตระกูล PIC18F97J60	30
2.3.1 ลักษณะหลักการ	30
2.3.2 อีเทอร์เน็ตโมดูล	34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญ(ต่อ)

	หน้า
บทที่ 3 การออกแบบโครงงาน	
3.1 การออกแบบฮาร์ดแวร์	36
3.1.1 บอร์ดไมโครคอนโทรลเลอร์	36
3.1.2 บอร์ด ZigBee	39
3.2 โครงสร้างของระบบ	43
3.2.1 การออกแบบฟังก์ชันการทำงานของโปรแกรมควบคุมตัวควบคุม	46
บทที่ 4 การทดลองและผลการทดลอง	
4.1 การรับส่งข้อมูลผ่านทางโปรโตคอล TCP/IP ไปยังพอร์ตอนุกรม	57
4.1.1 การใช้โปรแกรม command DOS ทดสอบpingหมายเลข IP	57
4.1.2 ใช้โปรแกรม Hercules ทดสอบการส่งค่าข้อมูลจาก TCP ไปยังพอร์ตอนุกรม	58
4.1.3 ใช้โปรแกรม Hercules ทดสอบการส่งค่าข้อมูลจาก TCP ไปยังพอร์ตอนุกรม	60
4.2 การรับส่งข้อมูลผ่านทางโปรโตคอล TCP/IP ไปยังโมดูล ZigBee	63
บทที่ 5 บทสรุปและวิจารณ์	
5.1 สรุปผลการทดลอง	67
5.2 ปัญหาที่พบในระหว่างการดำเนินโครงงาน	67
5.3 แนวทางแก้ไขและพัฒนา	67
บรรณานุกรม	68
ภาคผนวก	69

สารบัญรูป

	หน้า
รูปที่ 1.1 ลักษณะของโครงงาน โดยรวม	2
รูปที่ 2.1 สเตค TCP/IP เปรียบเทียบกับมาตรฐาน OSI	5
รูปที่ 2.2 ความสัมพันธ์ระหว่างโปรโตคอลต่างๆ ของ TCP/IP	5
รูปที่ 2.3 แอปพลิเคชันต่างๆ ในสเตค TCP/IP	6
รูปที่ 2.4 แอปพลิเคชันหรือโปรเซสต่างๆสื่อสารกับชั้นทรานสปอร์ต	7
รูปที่ 2.5 โปรเซสต่างๆที่เรียกใช้ชั้นทรานสปอร์ต	8
รูปที่ 2.6 รูปแบบของTCP แพ็กเก็ต	9
รูปที่ 2.7 รูปแบบของUDP แพ็กเก็ต	11
รูปที่ 2.8 รูปแบบของ IP แพ็กเก็ต	12
รูปที่ 2.9 โปรโตคอล IP ในการส่งผ่านข้อมูลระหว่างเครือข่าย	14
รูปที่ 2.10 โครงสร้างของโปรโตคอลTCP/IP	16
รูปที่ 2.11 เปรียบเทียบช่วงความถี่ของมาตรฐานต่างๆ	18
รูปที่ 2.12 เครือข่ายแบบสตาร์	23
รูปที่ 2.13 เครือข่ายแบบต้นไม้	24
รูปที่ 2.14 เครือข่ายแบบเมช	24
รูปที่ 2.15 สเตคโปรโตคอล ZigBee	27
รูปที่ 2.16 โครงสร้างสถาปัตยกรรมของสเตคโปรโตคอล ZigBee	28
รูปที่ 2.17ระบบคาล์วไฟฟ้าไร้สายในสิ่งแวดล้อมเชื่อมต่อ UART	29
รูปที่ 2.18 แพ็กเก็ตข้อมูล UART 0x1F (31 ในเลขฐานสิบ) ถูกส่งผ่าน โมดูล RF	29
รูปที่ 2.19ไดอะแกรมขาของ PIC ตระกูล PIC18F97J60 ชนิด 80 ขา	31
รูปที่ 2.20 บล็อกไดอะแกรมของ PIC ตระกูล PIC18F97J60 ชนิด 80 ขา	33
รูปที่ 2.21 บล็อกไดอะแกรมอีเทอร์เน็ต โมดูล	34
รูปที่ 2.22 องค์ประกอบที่ต้องการภายนอกของการปฏิบัติการอีเทอร์เน็ต	35
รูปที่ 3.1 วงจรพอร์ต RJ45	36
รูปที่ 3.2 วงจร EEPROM	37
รูปที่ 3.3 วงจรจ่ายกระแสไฟ	37

สารบัญรูป(ต่อ)

	หน้า
รูปที่ 3.4 วงจร MAX3232 และ RS232	37
รูปที่ 3.5 วงจรไมโครคอนโทรลเลอร์และพอร์ต ICD2	38
รูปที่ 3.6 ขาสัญญาณ ZigBee	39
รูปที่ 3.7 วงจรแสดงระดับสัญญาณ ZigBee	40
รูปที่ 3.8 บอร์ดไมโครคอนโทรลเลอร์	40
รูปที่ 3.9 บอร์ด ZigBee	41
รูปที่ 3.10 บอร์ดไมโครคอนโทรลเลอร์ต่อกับบอร์ด ZigBee	41
รูปที่ 3.11 บอร์ดเกตเวย์	42
รูปที่ 3.12 โครงสร้างของระบบการเชื่อมต่อโปรโตคอล TCP/IP- ZigBee เกตเวย์	43
รูปที่ 3.13 การทำงานของโปรโตคอลย่อยในสแตค TCP/IP	43
รูปที่ 3.14 ภาพรวมฟังก์ชันการทำงานของระบบการเชื่อมต่อโปรโตคอล TCP/IP-ZigBee เกตเวย์	44
รูปที่ 3.15 การออกแบบฟังก์ชันการทำงานของโปรแกรมควบคุมตัวควบคุม	46
รูปที่ 3.16 การทำงานของโปรแกรมหลัก	47
รูปที่ 3.17 การทำงานของ HTTP	48
รูปที่ 3.18 การทำงานของ FTP	48
รูปที่ 3.19 การทำงานของ DHCP	49
รูปที่ 3.20 การทำงานของ TCP	50
รูปที่ 3.21 การทำงานของ UDP	51
รูปที่ 3.22 การทำงานของ ICMP	52
รูปที่ 3.23 การทำงานของ IP	53
รูปที่ 3.24 การทำงานของ ARP	54
รูปที่ 3.25 การทำงานของ MAC	55
รูปที่ 4.1 การทดลอง	56
รูปที่ 4.2 การทดลองการpingหมายเลข IP	57
รูปที่ 4.3 โปรแกรม command DOS ที่ใช้ทดลองการpingหมายเลข IP	57
รูปที่ 4.4 การทดลองรับส่งค่าจากโปรโตคอล TCP ไปยังพอร์ตอนุกรม	58
รูปที่ 4.5 การเปิดพอร์ตอนุกรมที่ด้านผู้รับ	58
รูปที่ 4.6 การสร้างการเชื่อมต่อที่ด้านผู้ใช้	59

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญรูป(ต่อ)

	หน้า
รูปที่ 4.7 การส่งข้อมูลที่ด้านผู้ส่ง	59
รูปที่ 4.8 การรับข้อมูลที่ด้านผู้รับ	60
รูปที่ 4.9 การเปิดพอร์ตอนุกรมที่ด้านผู้รับ	60
รูปที่ 4.10 การเปิดพอร์ต UDP ที่ด้านผู้ส่ง	61
รูปที่ 4.11 การส่งข้อมูลที่ด้านผู้ส่ง	61
รูปที่ 4.12 การส่งข้อมูลที่ด้านผู้รับ	62
รูปที่ 4.13 การรับข้อมูลที่ด้านผู้ส่ง	62
รูปที่ 4.14 การทดลองการรับส่งข้อมูลผ่านทาง โปรโตคอล TCP/IP ไปยังโมดูล ZigBee	63
รูปที่ 4.15 ค่าหมายเลขที่อยู่ของ ZigBee ตัวที่ 1	64
รูปที่ 4.16 ค่าหมายเลขที่อยู่ของ ZigBee ตัวที่ 2	64
รูปที่ 4.17 การติดต่อของ ZigBee ตัวที่ 1	65
รูปที่ 4.18 การติดต่อของ ZigBee ตัวที่ 2	65
รูปที่ 4.18 การส่งข้อมูลจากคอมพิวเตอร์เครื่องส่ง	66
รูปที่ 4.19 การส่งข้อมูลจากคอมพิวเตอร์ฝั่งรับ	66

สารบัญตาราง

	หน้า
ตารางที่ 1.1 แผนภูมิขั้นตอนการดำเนินงาน โครงการงาน	3
ตารางที่ 2.1 ชนิดของอุปกรณ์ IEEE 802.15.4	19
ตารางที่ 2.2 ข้อมูลจำเพาะของโมดูล ZigBee	21
ตารางที่ 2.3 การกำหนดพินของโมดูล ZigBee	22
ตารางที่ 2.4 การเปรียบเทียบมาตรฐานการติดต่อสื่อสารแบบไร้สาย	25
ตารางที่ 2.5 ลักษณะอุปกรณ์ของ PIC ตระกูล PIC18F97J60 ชนิด 80 ขา	32



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของปัญหา

ในโลกยุคปัจจุบันการติดต่อสื่อสารเป็นปัจจัยที่สำคัญที่สุด ในการดำรงชีวิตและการดำเนินไป ซึ่งกิจกรรมต่างๆ การติดต่อสื่อสารในปัจจุบันพัฒนาไปอย่างรวดเร็วและมีประสิทธิภาพมากขึ้น จึงทำให้เกิดเทคโนโลยีใหม่เกิดขึ้นมากมาย เพื่ออำนวยความสะดวกในชีวิตประจำวันของมนุษย์ ทำให้ข้ามขีดจำกัดของการสื่อสารแบบเก่าที่มีระยะทางและการวางสายซึ่งเป็นอุปสรรคในการติดต่อสื่อสารในอดีต

เทคโนโลยี ZigBee เป็นเทคโนโลยีไร้สายชนิดหนึ่ง ในปัจจุบันได้มีการใช้ที่แพร่หลายมากขึ้น เพราะอำนวยความสะดวกสบายแก่ผู้ใช้และไม่ต้องเสียค่าบริการในการใช้งาน เมื่อเทียบกับเทคโนโลยีอื่นที่มีลักษณะเดียวกัน

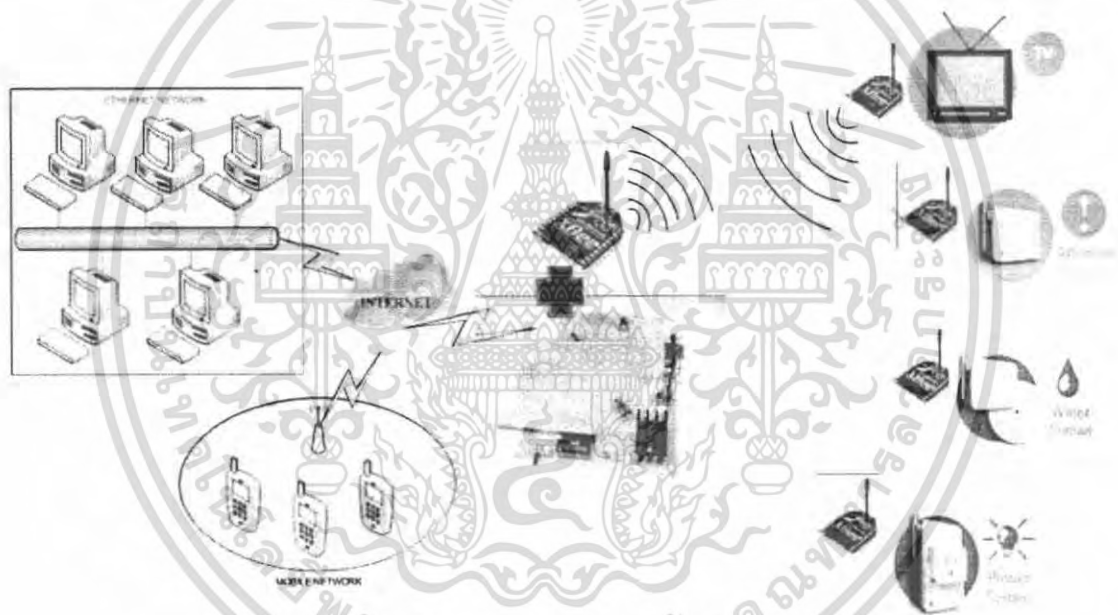
โครงการนี้จะนำความสามารถของเทคโนโลยีนี้ มาใช้ร่วมกับไมโครคอนโทรลเลอร์ (PIC18F86J65 ซึ่งเป็นไมโครคอนโทรลเลอร์ในตระกูลของ PIC18F97J60) เพื่อทำเกตเวย์เชื่อมต่อระหว่างโปรโตคอล TCP/IP กับ ZigBee ทำให้การใช้งาน ZigBee มีประสิทธิภาพมากขึ้น ง่ายต่อการควบคุม และนำไปพัฒนาโปรแกรมประยุกต์การใช้งานโปรโตคอล ZigBee ให้มีความหลากหลายได้มากขึ้น

1.2 วัตถุประสงค์ของโครงการ

- 1.2.1 เพื่อวิเคราะห์และออกแบบการรับส่งข้อมูลระหว่างโปรโตคอล TCP/IP กับ ZigBee โดยใช้ไมโครคอนโทรลเลอร์ (PIC18F86J65) บนบอร์ดได้
- 1.2.2 ศึกษาการทำงานของไมโครคอนโทรลเลอร์ สแตกโปรโตคอล TCP/IP และ ZigBee
- 1.2.3 เพื่อออกแบบบอร์ดที่ใช้ไมโครคอนโทรลเลอร์ร่วมกับโมดูล ZigBee ได้
- 1.2.4 ศึกษาเกี่ยวกับโปรแกรมภาษาซีในการควบคุมไมโครคอนโทรลเลอร์

1.3 ขอบเขตของโครงการ

- 1.3.1 สามารถวิเคราะห์และออกแบบระบบที่ใช้การรับส่งข้อมูลระหว่างโปรโตคอล TCP/IP และ โมดูล ZigBee โดยใช้ไมโครคอนโทรลเลอร์ (PIC18F86J65) ได้
- 1.3.2 มีความรู้ และ ความเข้าใจในการทำงานของไมโครคอนโทรลเลอร์ (PIC18F86J65) สแตกโปรโตคอล TCP/IP และ โมดูล ZigBee
- 1.3.3 สามารถออกแบบบอร์ดที่ใช้ไมโครคอนโทรลเลอร์ร่วมกับ โมดูล ZigBee ได้
- 1.3.4 มีความรู้ และ ความเข้าใจเกี่ยวกับโปรแกรมภาษาซีในการควบคุมไมโครคอนโทรลเลอร์
- 1.3.6 สามารถทำการเชื่อมต่อรับส่งข้อมูลระหว่างโปรโตคอล TCP/IP และ ZigBee ได้



รูปที่ 1.1 ลักษณะ โครงงาน โดยรวม

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1.4 ขั้นตอนการดำเนินงาน

ตารางที่ 1.1 แผนภูมิขั้นตอนการดำเนินงาน โครงการงาน

ขั้นตอนที่	วิธีการดำเนินโครงการงาน	ระยะเวลาการดำเนินงาน							
		มี.ย. 50	ก.ค. 50	ส.ค. 50	ก.ย. 50	ต.ค. 50	พ.ย. 50	ธ.ค. 50	ม.ค. 51
1	โครงการงาน TCP/IP – ZigBee Gateway	↔							
2	ออกแบบฮาร์ดแวร์ในส่วนวงจรควบคุมและวงจรทำงาน และซอฟต์แวร์ในส่วนโปรแกรมควบคุมการทำงาน		↔						
3	เขียนโปรแกรมควบคุมการทำงานโดยใช้ภาษา C ผ่านทาง MPLAB		↔						
4	จำลองการทำงานของวงจรและโปรแกรม		↔						
5	บัดกรีวงจรที่ออกแบบไว้ลงบนบอร์ดไมโครคอนโทรลเลอร์			↔					
6	ทำเอกสารปริ๊นตูปัญญาณิพนธ์ส่วนของโครงการงาน 1		↔						
7	ปรับปรุงโปรแกรมควบคุมให้รับส่งโปรโตคอล TCP/IP ทั้งแบบ TCP และ UDP					↔			
8	เพิ่มการเชื่อมต่อแบบอีเทอร์เน็ต และปรับปรุงส่วนติดต่อกับผู้ใช้งานให้สามารถควบคุมผ่านการเชื่อมต่อแบบอีเทอร์เน็ต					↔			
9	ทดสอบการทำงานของฮาร์ดแวร์และซอฟต์แวร์ โดยต้องสามารถสร้างการเชื่อมต่อระหว่าง TCP/IP กับ โมดูล ZigBee ได้						↔		
10	ออกแบบและก๊อปปี้แผ่นปริ๊นตูปัญญาณิพนธ์ตามวงจรที่ผ่านการทดสอบทั้งหมดแล้ว							↔	
11	ทดสอบและแก้ไขจุดบกพร่องของโครงการงานที่อาจเกิดขึ้น							↔	
12	ทำเอกสารปริ๊นตูปัญญาณิพนธ์							↔	

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งผู้ที่มีให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 2

ทฤษฎีที่เกี่ยวข้อง

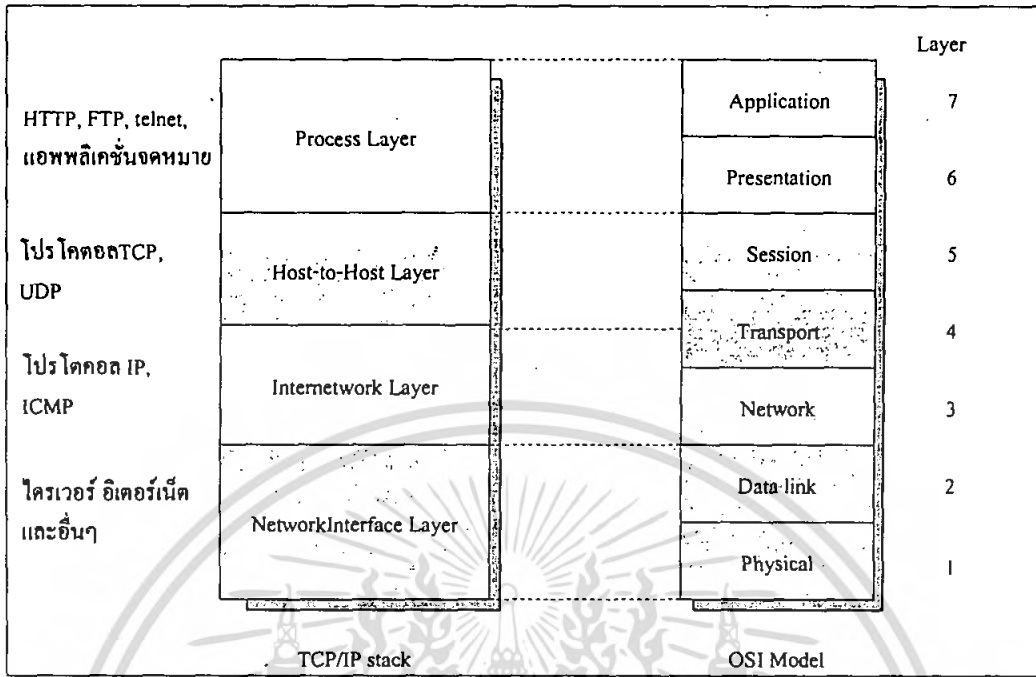
2.1 สแต็กโปรโตคอล TCP/IP

ในการใช้งานเครือข่ายคอมพิวเตอร์ทั่วไปหรือในเครือข่ายอินเทอร์เน็ตก็ตาม จะมีการส่งผ่านข้อมูลไปมาระหว่างเครื่องคอมพิวเตอร์ภายในเครือข่ายหรือข้ามเครือข่ายออกไป ระบบคอมพิวเตอร์ที่เชื่อมต่ออยู่ในแต่ละเครือข่ายอาจใช้ฮาร์ดแวร์และซอฟต์แวร์ที่เหมือนกันหรือแตกต่างกัน ดังนั้นการที่ส่งผ่านข้อมูลถึงกันและตีความได้อย่างถูกต้องตรงกันจะต้องทำข้อกำหนดร่วมกันในการสื่อสารเสียก่อน เรียกว่าต้องกำหนดระเบียบวิธีการในการติดต่อให้ตรงกัน ข้อตกลงนั้น ถูกเรียกว่า โปรโตคอล คือ ระเบียบวิธีที่กำหนดขึ้นสำหรับการสื่อสารข้อมูล ที่สามารถส่งผ่านไปยังปลายทางได้อย่างถูกต้อง

โปรโตคอล TCP/IP มีการจัดแบ่งกลไกการทำงานออกเป็นชั้นๆ เหมือนกับมาตรฐาน OSI model และสามารถเทียบเคียงกับมาตรฐานของ OSI model ได้ ซึ่งในแต่ละชั้นของโปรโตคอล TCP/IP จะประกอบด้วย

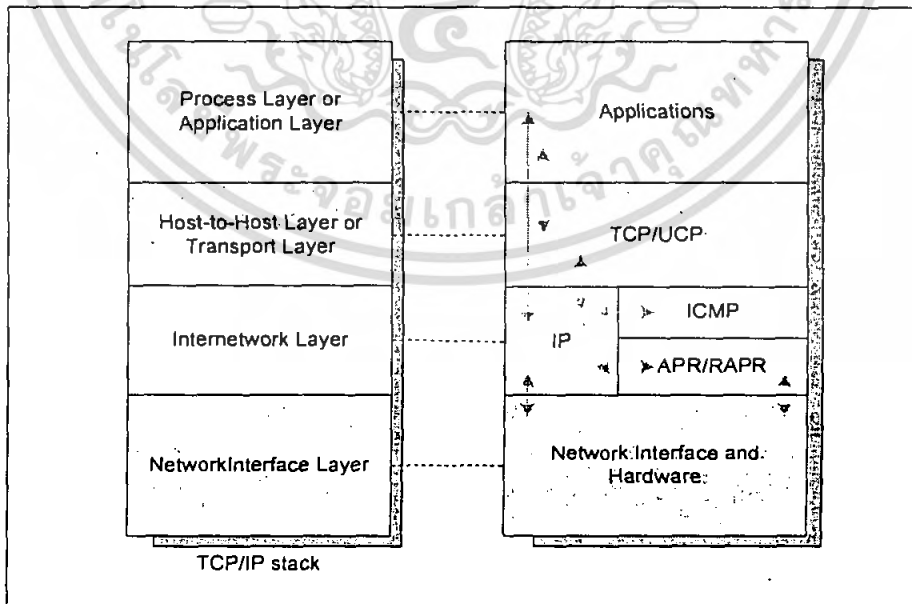
- ชั้นแอปพลิเคชัน (Application Layer)
- ชั้นทรานสปอร์ต (Transport Layer)
- ชั้นอินเทอร์เน็ตเวิร์ค (Internetwork Layer)
- ชั้นเน็ตเวิร์คอินเตอร์เฟส (Network Interface layer)

โดยเมื่อเทียบกับมาตรฐาน OSI model แล้วจะเป็นดังรูปที่ 2.1 ซึ่งเราจะเห็นว่าบางชั้นของโปรโตคอล TCP/IP เทียบได้กับมาตรฐาน OSI model ถึงสองชั้น และบางชั้นก็จะทำงานคาบเกี่ยวกับหลายๆชั้นของ OSI model



รูปที่ 2.1 สแตค TCP/IP เปรียบเทียบกับมาตรฐาน OSI

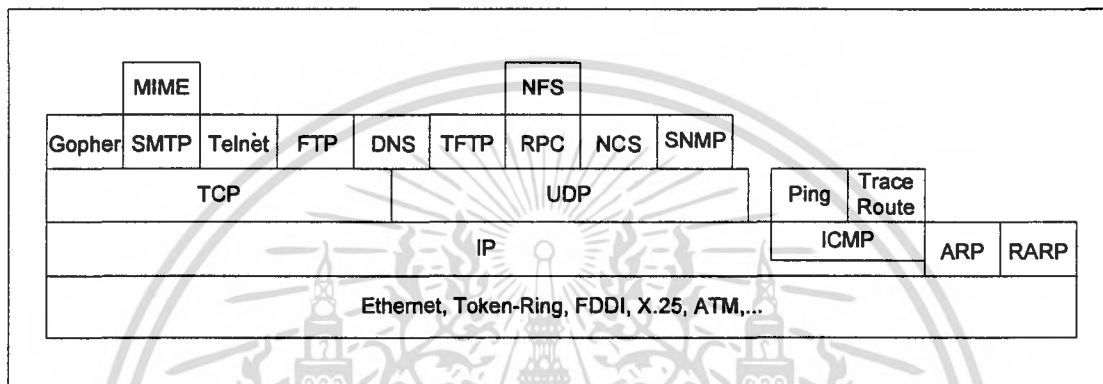
ในแต่ละกลไกของโปรโตคอล TCP/IP จะมีโปรโตคอลอื่นๆ ในชุดของ TCP/IP ร่วมทำงานอยู่ด้วย จึงทำให้เป็นที่มาของชื่อเรียก “โปรโตคอลสแตค (Protocol Stack)” เนื่องจากมีโปรโตคอลซ้อนทับกันอยู่เพื่อช่วยกันทำงานดังรูป



รูปที่ 2.2 ความสัมพันธ์ระหว่างโปรโตคอลต่างๆ ของ TCP/IP

จากรูปจะเห็นได้ว่า มีโปรโตคอลในแต่ละระดับซ้อนทับกันอยู่หลายตัวด้วยกัน การซ้อนกันเป็นชั้นๆ หรือแต่ละชั้นนี้ หากเป็น OSI model จะมีข้อบังคับให้แต่ละชั้นติดต่อกับเฉพาะชั้นที่ติดกับตนเองเท่านั้น แต่สำหรับสแตค TCP/IP แล้วจะเห็นว่าบางชั้นสามารถข้ามไปติดต่อกับชั้นอื่นที่ไม่ติดกับตนได้

2.1.1 ชั้นแอปพลิเคชัน



รูปที่ 2.3 แอปพลิเคชันต่างๆ ในสแตค TCP/IP

จากรูปแสดงลำดับชั้นการทำงานของโปรโตคอล TCP/IP เทียบกับมาตรฐาน OSI model นั้น ในชั้นบนสุดเรียกว่า ชั้นแอปพลิเคชัน ทำงาน 2 หน้าที่เทียบได้กับชั้นแอปพลิเคชันและชั้นพีรีเซนเทชัน ในชั้นนี้จะรองรับการทำงานของแอปพลิเคชันต่างๆ ที่ทำงานเป็นกระบวนการ อยู่ในเครื่องเซิร์ฟเวอร์ที่ให้บริการและเครื่องที่ขอใช้บริการ หรือไคลเอนต์ ซึ่งจะติดต่อกันผ่านโปรโตคอลเฉพาะแอปพลิเคชันอีกทีหนึ่ง

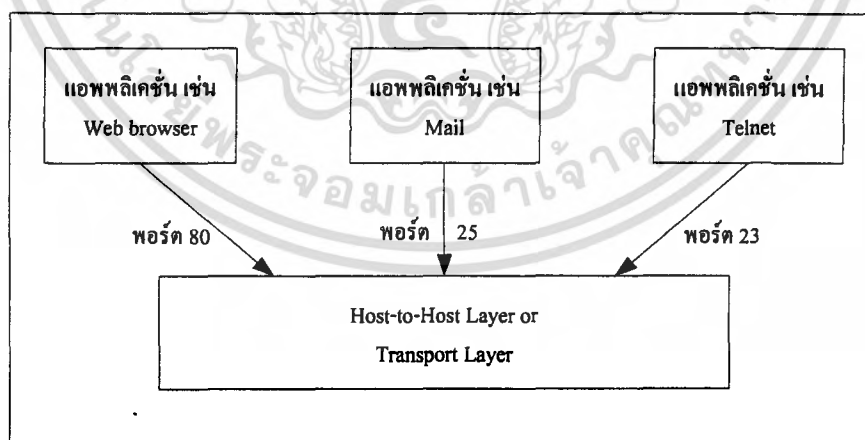
การทำงานของแอปพลิเคชันต่างๆ จะอยู่ที่ชั้นนี้ และมีการติดต่อกันตามแต่ละโปรโตคอลเฉพาะแล้วแต่แอปพลิเคชันที่ใช้งาน จากการที่ชั้นแอปพลิเคชันของ TCP/IP รองรับให้โปรโตคอลอื่นทำงานได้หลายกระบวนการและหลายโปรโตคอลได้พร้อมกันนั้น ทำให้ผู้ใช้สามารถเปิดโปรแกรมใช้งานได้หลายๆ อย่างพร้อมกัน เช่น เปิดโปรแกรมเว็บเบราว์เซอร์เพื่อเรียกดูเว็บเพจพร้อมกับใช้งานโปรแกรมรับส่งอีเมลไปพร้อมกันได้โดยไม่ต้องรอให้ทำงานอย่างหนึ่งอย่างใดเสร็จก่อน

โปรโตคอลหลักๆ ที่ทำงานในชั้นแอปพลิเคชัน ซึ่งผู้ใช้นักจะคุ้นเคยกันดีได้แก่ FTP, Telnet, HTTP และ SMTP นอกจากนี้ยังมีโปรโตคอลอื่นที่อยู่เบื้องหลัง ซึ่งทำงานโดยที่ผู้ใช้ไม่สามารถมองเห็นได้จากโปรแกรมหรือไม่ได้มีการใช้งานโดยตรง เช่น

- โพรโทคอล DNS ทำหน้าที่แปลงข้อมูลชื่อโดเมนเนม หรือชื่อเว็บไซต์ให้เป็นหมายเลข IP (IP Address)
- โพรโทคอล SNMP ใช้ในการควบคุมและตรวจสอบอุปกรณ์ที่อยู่ในเครือข่าย
- โพรโทคอล DHCP ทำหน้าที่แจกจ่ายข้อมูลพารามิเตอร์ของเครือข่ายให้กับเครื่องลูกข่ายที่เชื่อมต่ออยู่
- โพรโทคอล HTTP ใช้ในการทำเวปเพจ

2.1.2 ชั้นทรานสปอร์ต

การทำงานที่ชั้นนี้จะมีบทบาทในการจัดการต่อจากชั้นแอปพลิเคชันเรามาอีกเรียกชั้นโฮสต์-โฮสต์ (Host-to-Host) ว่าเป็นชั้นทรานสปอร์ต ซึ่งไม่ใช่ชั้นของทรานสปอร์ตในมาตรฐาน OSI model การทำงานของชั้นนี้จะมีการสร้างการเชื่อมต่อกันระหว่างแอปพลิเคชันชั้นทรานสปอร์ต โดยจุดที่เชื่อมกันเพื่อรับส่งข้อมูลนี้เรียกว่า พอร์ต และในแต่ละแอปพลิเคชันก็จะสร้างการเชื่อมต่อผ่านพอร์ต ได้พร้อมกันหลายแอปพลิเคชัน ซึ่งการใช้งานพอร์ตของแต่ละแอปพลิเคชันที่อยู่ในชั้นทรานสปอร์ตจะแตกต่างกันตามหมายเลขที่กำหนดไว้ และแต่ละโปรโตคอลจะมีการใช้งานพอร์ตหมายเลขต่างๆไม่ซ้ำกัน



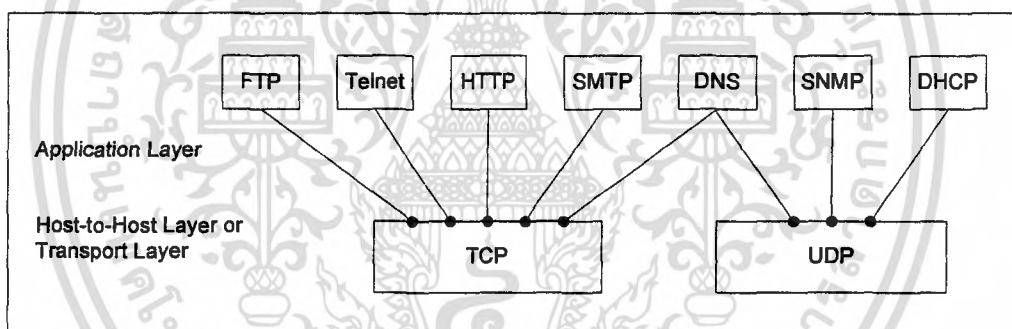
รูปที่ 2.4 แอปพลิเคชันหรือกระบวนการต่างๆสื่อสารกับชั้นทรานสปอร์ต

เมื่อแอปพลิเคชันทำงานผ่านโปรโตคอลในชั้นแอปพลิเคชัน จะมีการส่งผ่านข้อมูลไปยังชั้นทรานสปอร์ต ที่ชั้นนี้จะมีการเชื่อมต่อผ่านพอร์ตที่กำหนด ทำให้การรับส่งข้อมูลในแต่ละ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรโตคอลทำได้อีกต้องถึงแม้ว่าในเครื่องเซิร์ฟเวอร์ที่ให้บริการจะมีการทำงานอยู่หลายกระบวนการที่แตกต่างกันก็ตาม หรือมีผู้ใช้บริการเข้ามาใช้งานพร้อมกันจำนวนมาก และหลายแอปพลิเคชันในเวลาเดียวกัน ในชั้นทรานสปอร์ตนี้จะมีโปรโตคอลทำงานอยู่ 2 โปรโตคอลที่แตกต่างกัน คือ โปรโตคอล TCP และโปรโตคอล UDP ในการส่งผ่านข้อมูลลงไปที่ชั้นถัดๆไป จะเห็นว่า โปรโตคอล TCP และ UDP จะถูกผนึกเข้าไปในโปรโตคอล IP อีกทีหนึ่งและส่งต่อไปยังเครือข่ายอินเทอร์เน็ตต่อไป

ตัวโปรโตคอล TCP และโปรโตคอล UDP จะมีแอปพลิเคชันเฉพาะเพื่อเรียกใช้งานแยกกัน คือ แอปพลิเคชันที่ใช้โปรโตคอล FTP, Telnet, HTTP และ SMTP จะมีการส่งผ่านข้อมูลโดยเรียกใช้โปรโตคอล TCP ส่วนแอปพลิเคชันที่ใช้โปรโตคอล SNMP และ DHCP จะส่งผ่านข้อมูลโดยเรียกใช้โปรโตคอล UDP และสำหรับโปรโตคอล DNS นั้นจะสามารถเรียกใช้งานได้ทั้ง TCP และ UDP ซึ่งเหตุผลที่มีการเรียกใช้โปรโตคอล TCP และ UDP แยกต่างหาก เนื่องจากวิธีการทำงานของทั้งสองโปรโตคอลต่างกัมนั่นเอง



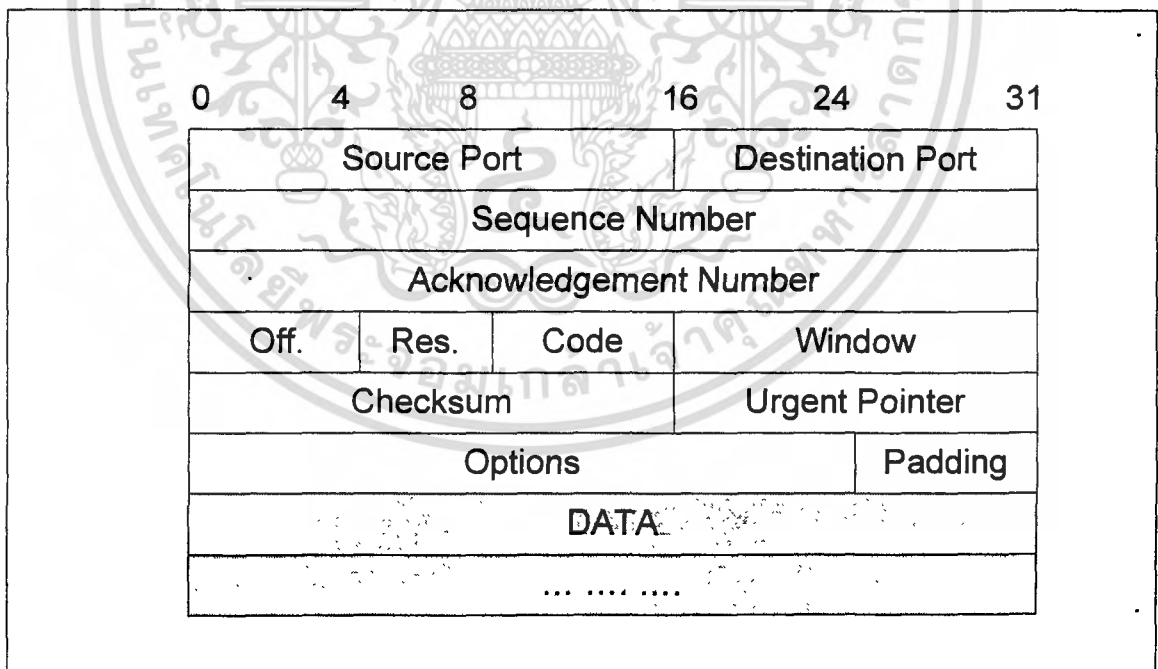
รูปที่ 2.5 กระบวนการต่างๆที่เรียกใช้ชั้นทรานสปอร์ต

2.1.2.1 โปรโตคอล TCP

เป็นโปรโตคอลที่มีการรับส่งข้อมูลแบบแบ่งเป็นส่วนย่อยๆก่อน โดยไม่คำนึงถึงปริมาณข้อมูลที่จะส่งออกไป แล้วจึงส่งไปยังปลายทางอย่างต่อเนื่องเป็นลำดับข้อมูล ในกรณีที่ข้อมูลส่วนใดส่วนหนึ่งสูญหายไป ก็จะส่งข้อมูลส่วนนั้นใหม่อีกครั้ง สำหรับปลายทางก็จะทำหน้าที่จัดเรียงส่วนของข้อมูลคาต้าแกรม (Datagram) ใหม่ให้ต่อเนื่องกัน และประกอบกลับเป็นข้อมูลทั้งหมดได้ ซึ่งจะแยกข้อมูลที่ไม่ถูกต้องออก ดังนั้นแอปพลิเคชันหรือกระบวนการใดที่อาศัยการส่งผ่านข้อมูลด้วยโปรโตคอล TCP จะต้องใช้หน่วยความจำและขนาดของช่องสัญญาณมากกว่าโปรโตคอล UDP

การติดต่อระหว่างกันจะมีการสร้างการติดต่อกัน ทั้ง 2 ด้าน (Connection-Oriented) เสียก่อน แล้วจึงจะรับส่งข้อมูลไปได้พร้อมกัน (full duplex) เหมือนกับการใช้โทรศัพท์ที่ติดต่อกัน เมื่อผู้ติดต่อต้นทางเรียกให้ฝ่ายตรงข้ามรับสายแล้ว จึงเริ่มการทักทายกันก่อนเพื่อให้แน่ใจว่าฝั่งตรงข้ามพร้อมที่จะติดต่อด้วย จากนั้นจึงเริ่มต้นติดต่อกัน และเมื่อต้องการจะเลิกการติดต่อก็จะมีการแจ้งบอกยกเลิกให้ฝั่งตรงข้ามทราบว่าจะเลิกการติดต่อและวางสายไป ซึ่งในระหว่างการติดต่อกันนั้น แม้ว่าฝ่ายใดฝ่ายหนึ่งหรือทั้งสองฝ่ายจะเงียบไป คือไม่พูดอะไรเป็นเวลานานๆ แต่การเชื่อมโยงจะยังคงอยู่ไม่ขาดไปจนกว่าฝ่ายใดฝ่ายหนึ่งจะวางสาย เช่นเดียวกับการติดต่อกันด้วยกลไกโปรโตคอล TCP เมื่อแอปพลิเคชันต้องการส่งผ่านข้อมูลจะใช้โปรโตคอลที่เหมาะสมในชั้น Process layer ติดต่อกันและมีการสร้างช่องส่งข้อมูลผ่านพอร์ตที่กำหนดเพื่อส่งผ่านข้อมูลไปยังโปรโตคอล TCP

ในระหว่างการรับส่งข้อมูลนี้ โปรโตคอล TCP จะเพิ่มขบวนการสอบทานข้อมูล เพื่อให้ข้อมูลมีความถูกต้องไม่ผิดพลาดไปจากเดิม โดยการส่งสัญญาณสอบทานข้อมูล (acknowledgement) และส่งข้อมูลให้ใหม่อีกครั้ง ถ้าปลายทางไม่ได้รับหรือเกิดความผิดพลาดขึ้น ความน่าเชื่อถือของการส่งผ่านข้อมูลโดยโปรโตคอล TCP จะมีมากกว่า แต่ก็ต้องอาศัยทรัพยากรของระบบมากกว่าในการทำงานเช่นกัน



รูปที่ 2.6 รูปแบบของ TCP แพ็กเก็ต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนหัวของ TCP มีรายละเอียด ดังนี้

Source Port Number: หมายเลขพอร์ตต้นทางที่ส่งคำคำแกรมนี้

Destination Port Number: หมายเลขพอร์ตปลายทางที่จะเป็นผู้รับคำคำแกรม

Sequence Number: พิลด์ที่ระบุหมายเลขลำดับอ้างอิงในการสื่อสารข้อมูลแต่ละครั้ง เพื่อใช้ในการแยกแยะว่าเป็นข้อมูลของชุดใด และนำมาจัดลำดับได้ถูกต้อง

Acknowledgment Number: ทำหน้าที่เช่นเดียวกับ Sequence Number แต่ใช้ในการตอบรับ

Header Length: โดยปกติความยาวของส่วนหัว TCP จะมีความยาว 20 ไบต์ แต่อาจจะมากกว่านั้น ถ้ามีข้อมูลในฟิลด์ option แต่ต้องไม่เกิน 60 ไบต์

Flag: เป็นข้อมูลระดับบิตที่อยู่ในส่วนหัว TCP โดยใช้เป็นตัวบอกคุณสมบัติของแพ็กเก็ต TCP ขณะนั้นๆ และใช้เป็นตัวควบคุมจังหวะการรับส่งข้อมูลด้วย Flag มีอยู่ทั้งหมด 6 บิต แบ่งได้ดังตารางที่ 2.3

2.1.2.2 โพรโทคอล UDP

ในชั้นทรานสปอร์ตนอกจากจะมีโปรโตคอล TCP ทำงานแล้ว ก็ยังมีโปรโตคอล UDP ที่มีคุณสมบัติแตกต่างกันอยู่ด้วย ในการรับส่งข้อมูลผ่านโปรโตคอล UDP จะเป็นแบบที่ทั้งสองด้านไม่จำเป็นต้องอาศัยการสร้างช่องทางเชื่อมต่อกัน (connectionless) ระหว่างเครื่องเซิร์ฟเวอร์ให้บริการกับเครื่องที่ขอใช้บริการ โดยไม่ต้องแจ้งให้ฝ่ายรับข้อมูลเตรียมรับข้อมูลเหมือนโปรโตคอล TCP และไม่มีการตรวจสอบความถูกต้องครบถ้วนในการรับส่งข้อมูลนั้นๆด้วย เนื่องจากโปรโตคอล UDP ไม่มีสัญญาณสอบทานข้อมูลในการส่งข้อมูลแต่ละครั้ง และไม่มีการส่งข้อมูลให้อีกในกรณีที่เกิดความผิดพลาดของการส่งข้อมูล เมื่อเป็นเช่นนี้แอปพลิเคชันหรือกระบวนการใดที่ต้องอาศัยโปรโตคอล UDP ในการส่งผ่านข้อมูลก็อาจจะต้องสร้างกระบวนการตรวจสอบข้อมูลขึ้นมาเอง

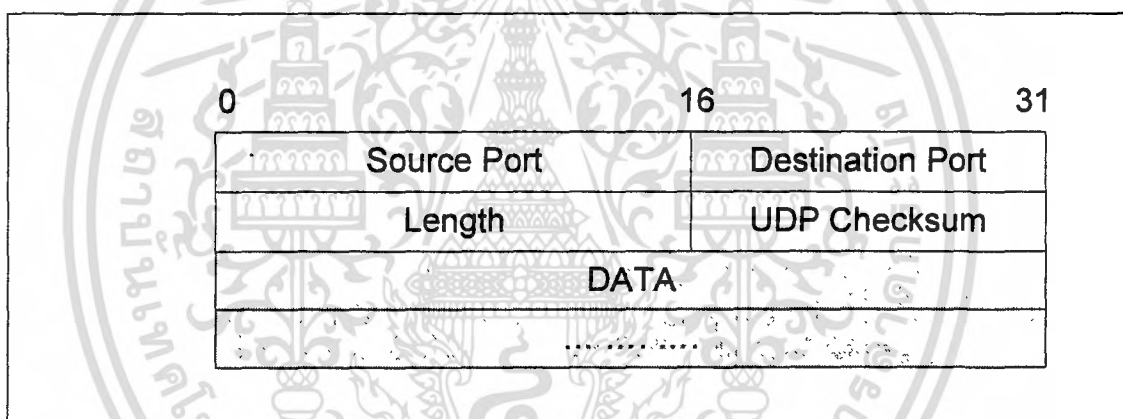
จากรูป 2.7 จะเห็นว่าโปรโตคอลชั้นบนขึ้นไป ที่ใช้การส่งผ่านข้อมูลโดยโปรโตคอล UDP เช่น โปรโตคอล SNMP (ใช้ควบคุมและจัดการอุปกรณ์ในเครือข่าย) หรือโปรโตคอล DHCP (ใช้ส่งข้อมูลพารามิเตอร์ของเครือข่ายให้กับเครื่องลูกข่ายได้ใช้งาน) การส่งข้อมูลเหล่านั้นไม่ต้องรับทราบหรือตรวจสอบว่าข้อมูลไปถึงปลายทางถูกต้องหรือไม่ แต่กลไกการตรวจสอบข้อมูลที่มีการรับส่งจะไปทำในชั้นตอนของโปรโตคอลชั้นที่สูงกว่าแทน

ตัวอย่างขั้นตอนกลไกการทำงานโดยใช้โปรโตคอล UDP มีดังต่อไปนี้

1. ในชั้นแอปพลิเคชัน เมื่อโปรแกรมควบคุมอุปกรณ์เครือข่ายต้องการส่งข้อมูลไปยังอุปกรณ์ที่ต้องการแอปพลิเคชันนั้นจะติดต่อผ่านโปรโตคอล SNMP ในชั้นแอปพลิเคชัน

2. โพรโทคอล SNMP จะติดต่อกับโพรโทคอล UDP ในชั้นถัดไป เพื่อขอติดต่อผ่านพอร์ตที่กำหนด
3. โพรโทคอล SNMP เตรียมข้อมูลที่จะส่ง รวมทั้งที่อยู่ปลายทาง
4. โพรโทคอล SNMP ส่งผ่านข้อมูลให้โพรโทคอล UDP ที่อยู่ในชั้นทรานสปอร์ต โพรโทคอล UDP ทำหน้าที่ผนึกข้อมูลนั้น ไปกับโพรโทคอล IP ในชั้นถัดลงไป เพื่อส่งข้อมูลออกจากเครื่อง

ซึ่งจะเห็นว่ามิกลไกที่ต่างจากการส่งข้อมูลด้วยโพรโทคอล TCP ซึ่งจะต้องมีการติดต่อกันก่อน และทั้งสองฝ่ายรับทราบการรับส่งข้อมูลของช่องการส่งข้อมูลนั้น



รูปที่ 2.7 รูปแบบของ UDP แพ็กเก็ต

ส่วนหัวของ UDP มีรายละเอียด ดังนี้

Source Port Number: หมายเลขพอร์ตต้นทางที่ส่งคาต้าแกรมนี้

Destination Port Number: หมายเลขพอร์ตปลายทางที่จะเป็นผู้รับคาต้าแกรม

UDP Length: ความยาวของคาต้าแกรม ทั้งส่วน Header และ Data นั้นหมายความว่าค่าที่น้อยที่สุดในฟิลด์นี้คือ 8 ซึ่งเป็นขนาดของ Header

Checksum: เป็นตัวตรวจสอบความถูกต้องของ UDP datagram และจะนำข้อมูลบางส่วนใน IP Header มาคำนวณด้วย

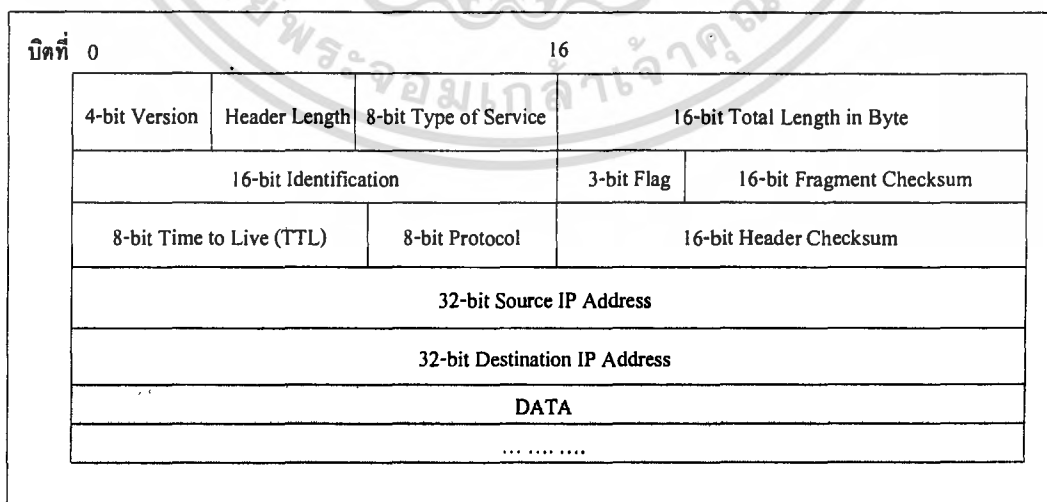
รูปแบบของแพ็กเก็ต UDP จะมีฟิลด์ข้อมูลส่วนหัว (header) น้อยมาก และไม่มีส่วนการตรวจสอบข้อมูลทำให้แพ็กเก็ต UDP มีขนาดเล็กและใช้หน่วยความจำ หรือ ทรัพยากรของระบบน้อย

2.1.3 ชั้นอินเทอร์เน็ตเวิร์ค

ในระดับล่างต่อมาในชั้นอินเทอร์เน็ตเวิร์ค มีหน้าที่ส่งผ่านข้อมูลในระหว่างเครือข่าย โดยมีโปรโตคอลที่ทำงานเป็นกลไกสำคัญในการส่งผ่านข้อมูลไปยังเครือข่ายใดๆบนอินเทอร์เน็ต คือ โปรโตคอล IP นอกจากนี้ในชั้นอินเทอร์เน็ตเวิร์ค ยังมีโปรโตคอลทำงานอยู่ด้วยอีก 2 ชนิดคือ โปรโตคอล ICMP และโปรโตคอล ARP

2.1.3.1 โปรโตคอล IP

โปรโตคอล IP ทำหน้าที่ให้บริการส่งผ่านข้อมูลที่มาจากชั้นทรานสปอร์ต เพื่อส่งข้ามไปยังเครือข่ายใดๆได้อย่างถูกต้อง แม้ว่าจะมีเครือข่ายเชื่อมต่อกันอยู่ในอินเทอร์เน็ตเป็นล้านๆ เครือข่ายก็ตาม เนื่องจากโปรโตคอล IP มีข้อมูลตำแหน่ง IP ปลายทางที่จะส่งข้อมูลไปให้ โดยทำงานร่วมกับอุปกรณ์เราเตอร์ (router) เพื่อส่งข้อมูลข้ามเครือข่ายออกไปได้ ตัวโปรโตคอล IP จะทำงานแบบแพ็กเก็ตสวิตซิง (packet switching) คือมีการส่งข้อมูลผ่านสวิทช์ไปยังปลายทาง โดยข้อมูลจะเดินทางไปยังเครือข่ายต่างๆ ผ่านสวิทช์นี้ไปเรื่อยๆจนกว่าจะถึงปลายทาง ตัววงจรผ่านหรือสวิทช์นี้อาจเป็นเกตเวย์ ในระบบเครือข่ายก็ได้ ซึ่งในข้อมูลของโปรโตคอล IP จะมีข้อมูลของหมายเลข IP ปลายทางที่จะส่งข้อมูลไปและเมื่อถึงเครือข่ายปลายทางแล้ว จะมีกลไกแปลงหมายเลข IP ให้เป็นหมายเลขฮาร์ดแวร์ประจำเครื่องที่ต้องการอีกทีหนึ่งด้วยโปรโตคอล ARP



รูปที่ 2.8 รูปแบบของ IP แพ็กเก็ต

ส่วนหัวของ IP โดยปกติจะมีขนาด 20 bytes ยกเว้นในกรณีที่มีการเพิ่ม option บางอย่าง ส่วนหัวของ IP มีรายละเอียดดังนี้

Version: หมายเลขเวอร์ชันของโปรโตคอล ที่ใช้งานในปัจจุบันคือ เวอร์ชัน 4 (IPv4) และเวอร์ชัน 6 (IPv6)

Header Length: ความยาวส่วนหัว โดยทั่วไปถ้าไม่มีส่วน option จะมีค่าเป็น 5*32 bit

Type of Service (TOS): ใช้เป็นข้อมูลสำหรับเราเตอร์ในการตัดสินใจเลือกการเส้นทางข้อมูลในแต่ละคาตาแกรม แต่ในปัจจุบันไม่ได้มีการนำไปใช้งานแล้ว

Length: ความยาวทั้งหมดเป็นจำนวนไบต์ของคาตาแกรม ด้วยขนาด 16 บิตของฟิลด์ จะหมายถึงความยาวสูงสุดของคาตาแกรม คือ 65535 ไบต์ (64KByte) แต่ในการส่งข้อมูลจริง ข้อมูลจะถูกแยกเป็นส่วนๆตามขนาดของ MTU ที่กำหนดในลิงค์ชั้น และนำมารวมกันอีกครั้งเมื่อส่งถึงปลายทาง แอปพลิเคชันส่วนใหญ่จะมีขนาดของคาตาแกรมไม่เกิน 512 ไบต์

Identification: เป็นหมายเลขของคาตาแกรมในกรณีที่มีการแยกคาตาแกรมเมื่อข้อมูลส่งถึงปลายทางจะนำข้อมูลที่มี identification เดียวกันมารวมกัน

Flag: ใช้ในกรณีที่มีการแยกคาตาแกรม

Fragment offset: ใช้ในการกำหนดตำแหน่งข้อมูลในคาตาแกรมที่มีการแยกส่วน เพื่อให้สามารถนำกลับมาเรียงต่อกันได้อย่างถูกต้อง

Time to live (TTL): กำหนดจำนวนครั้งที่มากที่สุดที่คาตาแกรมจะถูกส่งระหว่าง hop (การส่งผ่านข้อมูลระหว่างเน็ตเวิร์ค) เพื่อป้องกันไม่ให้เกิดการส่งข้อมูลโดยไม่สิ้นสุด โดยเมื่อข้อมูลถูกส่งไป 1 hop จะทำการลดค่า TTL ลง 1 เมื่อค่าของ TTL เป็น 0 และข้อมูลยังไม่ถึงปลายทาง ข้อมูลนั้นจะถูกยกเลิก และเราเตอร์สุดท้ายจะส่งข้อมูล ICMP แจ้งกลับมายังต้นทางว่าเกิด time out ในระหว่างการส่งข้อมูล

Protocol: ระบุโปรโตคอลที่ส่งในคาตาแกรม เช่น TCP, UDP หรือ ICMP

Header checksum: ใช้ในการตรวจสอบความถูกต้องของข้อมูลในส่วนหัว

Source IP address: ที่อยู่ IP ของผู้ส่งข้อมูล

Destination IP address: ที่อยู่ IP ของผู้รับข้อมูล

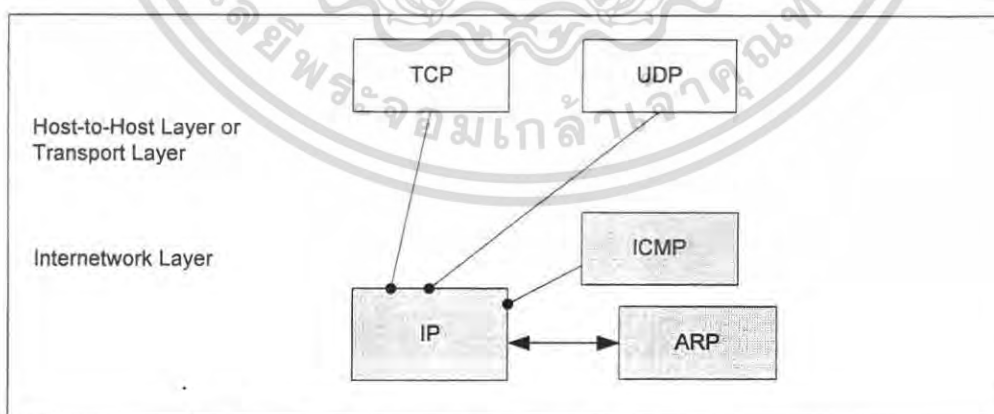
Data: ข้อมูลจากโปรโตคอลระดับบน

2.1.3.2 โพรโทคอล ICMP

หน้าที่หลักของโพรโทคอล ICMP คือการแจ้งหรือแสดงข้อความจากระบบ เพื่อบอกให้ผู้ใช้ทราบว่าเกิดอะไรขึ้นในการส่งผ่านข้อมูลนั้น ซึ่งปัญหาส่วนมากที่พบคือส่งไปไม่ได้ หรือปลายทางรับข้อมูลไม่ได้ เป็นต้น นอกจากนี้โพรโทคอล ICMP ยังถูกเรียกใช้งานจากเครื่องเซิร์ฟเวอร์ และเราเตอร์อีกด้วย เพื่อแลกเปลี่ยนข้อมูลที่ใช้ควบคุม ส่วนรูปแบบการทำงานของโพรโทคอล ICMP นั้นจะทำงานคู่กับโพรโทคอล IP ในระดับเดียวกัน และข้อความต่างๆที่แจ้งให้ทราบจะถูกผนึกอยู่ภายในข้อมูลของ IP อีกทีหนึ่ง

ข้อความที่โพรโทคอล ICMP ส่งนั้นแบ่งออกได้ 2 แบบคือ ข้อความแจ้งข้อผิดพลาด (ICMP error Message) และข้อความเรียกขอข้อมูลเพิ่มเติม (ICMP query) ตัวอย่างกลไกการทำงานของโพรโทคอล ICMP เช่น เมื่อมีการส่งผ่านข้อมูลจากผู้ใช้ไปยังปลายทางที่ไม่ถูกต้อง หรือขณะนั้นเครื่องปลายทางเกิดปัญหาจนไม่สามารถรับข้อมูลได้ ที่เราเตอร์จะส่งข้อความแจ้งเป็นข้อความที่ชื่อ “Destination Unreachable” ให้กับผู้ส่งข้อมูล นอกจากนี้ตัวข้อมูลที่แจ้งข้อความก็จะมีส่วนของข้อมูล IP คาด้าแกรมที่เกิดปัญหาด้วย ดังนั้นเมื่อผู้ส่งข้อมูลได้รับข้อความแจ้งแล้วก็จะทราบได้ว่าจุดที่เกิดปัญหานั้นอยู่ที่ใด

ดังนั้นโพรโทคอล ICMP จึงกลายมาเป็นเครื่องมืออย่างหนึ่งในการช่วยทดสอบเครือข่าย เช่น คำสั่ง “ping” ที่เรามักใช้ทดสอบว่าเครื่องเซิร์ฟเวอร์ที่ให้บริการหรืออุปกรณ์ที่ต่ออยู่ในเครือข่ายอินเทอร์เน็ตนั้น ยังทำงานเป็นปกติหรือไม่ แล้วคำสั่ง “ping” มีการเรียกใช้งานโพรโทคอล ICMP แจ้งเป็นข้อความให้ทราบอีกต่อหนึ่ง



รูปที่ 2.9 โพรโทคอล IP ในการส่งผ่านข้อมูลระหว่างเครือข่าย

โพรโทคอล TCP และ UDP ต้องอาศัย โพรโทคอล IP เพื่อการส่งผ่านข้อมูลระหว่างเครือข่าย และในชั้นอินเทอร์เน็ตเวิร์ค ยังมีโพรโทคอล ICMP ทำหน้าที่ส่งข้อความแจ้งเตือนและโพรโทคอล ARP ทำหน้าที่แปลงโพรโทคอล IP ไปเป็นหมายเลขของฮาร์ดแวร์จริง อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

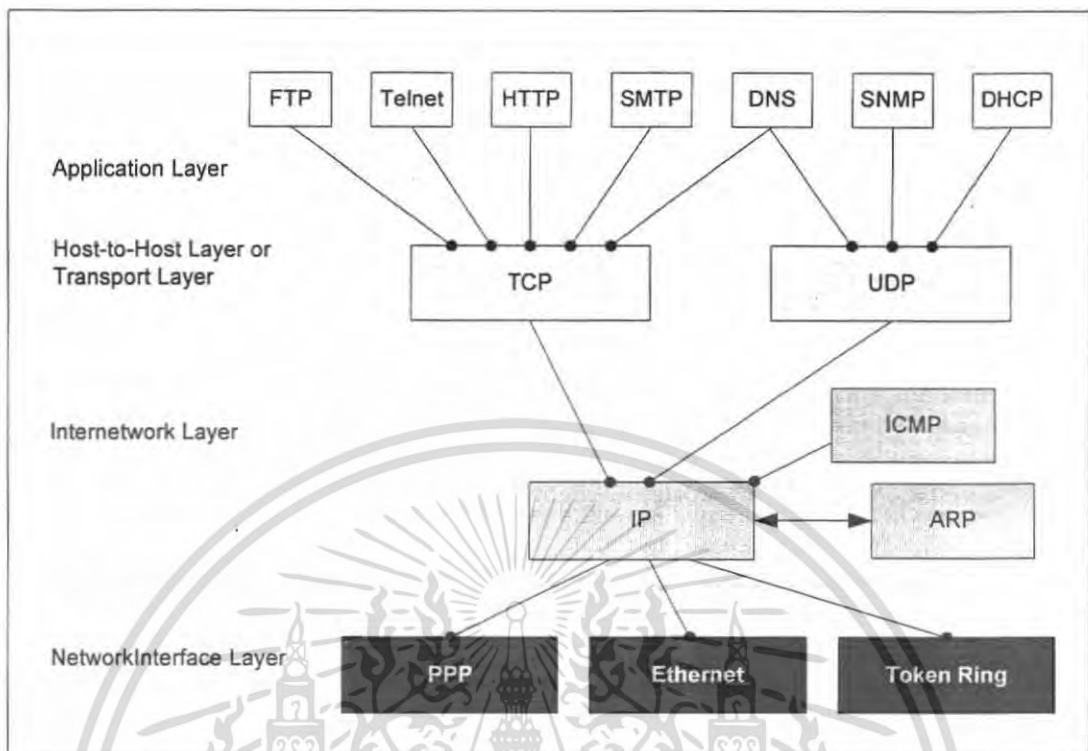
2.1.3.3 โพรโทคอล ARP

โพรโทคอล ARP ถูกเรียกใช้งานโดยโพรโทคอล IP เพื่อช่วยแปลงหมายเลข IP ไปเป็นหมายเลขฮาร์ดแวร์ปลายทาง เช่น เว็บเซิร์ฟเวอร์เครื่องหนึ่งเชื่อมต่ออยู่ในเครือข่ายอินเทอร์เน็ต และในการเชื่อมต่อนี้ต้องอาศัยการ์ดแลนที่ติดตั้งอยู่ที่การ์ดแลนนี้เองจะมีหมายเลขเฉพาะประจำฮาร์ดแวร์ที่ไม่ซ้ำกับใคร เพื่อใช้อ้างอิงการส่งข้อมูลในเครือข่าย แต่เมื่อมาใช้งานในโพรโทคอล TCP/IP ก็จะต้องมีการกำหนดหมายเลข IP ประจำตัวเพื่อใช้อ้างอิงกัน และโพรโทคอล ARP จะทำหน้าที่แปลงค่าหมายเลข IP ให้เป็นหมายเลขฮาร์ดแวร์จริงให้ในระดับการทำงานที่ชั้นอินเทอร์เน็ตเวิร์ค นี้ ซึ่งกลไกการแปลงนี้เรียกว่า “address resolution”

2.1.4 ชั้นเน็ตเวิร์คอินเทอร์เน็ตเฟส

เนื่องจากในด้านกายภาพของเครือข่ายนั้น มีหลายวิธีการและหลายรูปแบบในการเชื่อมต่อระบบให้เป็นเครือข่าย แต่อย่างไรก็ตามในเครือข่ายอินเทอร์เน็ตนี้ ข้อมูลจะถูกถ่ายทอดและส่งผ่านไปยังปลายทางโดยไม่คำนึงถึงรูปแบบการเชื่อมต่อทางกายภาพ ไม่ว่าจะเป็นการใช้เครือข่ายใยแก้วนำแสงหรือเครือข่ายสาย UTP (Unshielded Twist Pair) เชื่อมต่อเป็นแบบเครือข่ายอีเทอร์เน็ต ธรรมดาหรือเครือข่าย Token Ring, ATM, ISDN หรืออื่นๆก็ตาม

การทำงานระดับล่างสุดต่อจากชั้นอินเทอร์เน็ตเวิร์ค จะเป็นการแปลงข้อมูลให้อยู่ในรูปแบบที่เหมาะสม และแปลงเป็นสัญญาณไฟฟ้าส่งไปยังเครือข่ายต่อไป ซึ่งในชั้นเน็ตเวิร์คอินเทอร์เน็ตเฟส นี้ เมื่อเทียบกับมาตรฐาน OSI Model แล้วจะเป็นการรวม 2 ชั้นเข้าด้วยกัน คือ ชั้นดาต้าลิงค์ และชั้นฟิสิคอลล กล่าวโดยสรุปคือ การทำงานในชั้นต่างๆตามโครงสร้างของโพรโทคอล TCP/IP จะมีลักษณะดังรูปที่ 2.9



รูปที่ 2.10 โครงสร้างของโปรโตคอลTCP/IP

โครงสร้างของโปรโตคอล TCP/IP ในแต่ละชั้นจะมีโปรโตคอลหลักหน้าที่ต่างๆ และส่งผ่านข้อมูลไปยังเครือข่ายและออกสู่อินเทอร์เน็ต

กล่าวโดยสรุปก็คือ โปรโตคอล TCP/IP ทำงานโดยแบ่งเป็นชั้นเทียบกับ OSI model ได้ กลไกในการทำงานของโปรโตคอล TCP/IP มี 4 ชั้น ซึ่งในชั้นแรก คือชั้นแอปพลิเคชัน ทำหน้าที่ติดต่อกับแอปพลิเคชันและโปรโตคอลที่แอปพลิเคชันนั้นๆใช้งาน และส่งต่อมาให้ชั้นทรานสปอร์ต เพื่อติดต่อกันระหว่างเครื่องเซิร์ฟเวอร์ให้บริการกับเครื่องผู้ใช้บริการ ในชั้นนี้จะมีการสร้างการเชื่อมต่อระหว่างระบบขึ้นตามแต่ละโปรโตคอลที่ต้องการ ต่อมาเป็นการผนึกข้อมูลไปเป็น IP แพคเกจที่ชั้นอินเทอร์เน็ตเวิร์ค โดยอาศัยโปรโตคอล IP เพื่อให้สามารถติดต่อส่งข้อมูลข้ามเครือข่ายไปยังเครือข่ายและเครื่องที่ถูกต้องได้ และสุดท้ายการส่งข้อมูลออกสู่โลกภายนอกต้องอาศัยกลไกในชั้นเน็ตเวิร์คอินเทอร์เน็ตเฟสเพื่อแปลงข้อมูลใหม่ เพิ่มข้อมูลที่จำเป็นในการอ้างอิงตำแหน่งและแปลงข้อมูลเป็นสัญญาณไฟฟ้าส่งออกไปยังเครือข่าย และอาจจะออกไปยังเกตเวย์หรือเราเตอร์เพื่อข้ามเครือข่ายออกไปยังเส้นทางที่กำหนดไว้ในอินเทอร์เน็ตต่อไป

เราจะเห็นว่าในแต่ละชั้นของโครงสร้างสแตค TCP/IP มีการใช้งานโปรโตคอลต่างๆอยู่หนึ่งโปรโตคอลหรือมากกว่า ในแต่ละโปรโตคอลเหล่านี้ก็จะรับผิดชอบทำหน้าที่ของตน เพื่อส่งผ่านข้อมูลลงไปยังระดับล่าง และออกสู่เครือข่ายอินเทอร์เน็ตในที่สุด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.1.5 กลไกของโปรโตคอล IP

ในการส่งผ่านข้อมูลหรือไปยังเครือข่ายอินเทอร์เน็ตนั้น โปรโตคอล IP จะทำหน้าที่พิจารณาว่าปลายทางในการส่ง IP คาต้าแกรมนั้นจะเป็นภายในเครือข่ายของตนเองหรือจะต้องส่งข้อมูลข้ามเครือข่ายไปอีก โดยการพิจารณานี้โปรโตคอล IP จะตรวจสอบจากค่าหมายเลข IP ปลายทางว่าส่วนที่เป็นค่าหมายเลขเครือข่าย จะเหมือนกับค่าหมายเลขเครือข่ายของ IP ต้นทางหรือไม่ ถ้าค่าตรงกันแสดงว่าการส่งข้อมูลอยู่ภายในเครือข่ายเดียวกัน แต่ถ้าค่าต่างกัน แสดงว่าต้องส่งข้อมูลไปยังปลายทางที่อยู่คนละเครือข่ายกัน

การส่งข้อมูลภายในเครือข่ายเดียวกัน มีกลไกดังนี้

1. โปรโตคอล IP จะเรียกใช้บริการโปรโตคอล ARP (Address Resolution Protocol) เพื่อแปลงหมายเลข IP ปลายทางให้เป็นค่าหมายเลขฮาร์ดแวร์ (MAC address)

2. เมื่อโปรโตคอล IP ได้รับหมายเลขฮาร์ดแวร์แล้ว จะส่งข้อมูลนั้นไปยังฮาร์ดแวร์ที่ระบุไว้ การส่งข้อมูลข้ามเครือข่าย มีกลไกดังนี้

1. โปรโตคอล IP ตรวจสอบพบว่าหมายเลข IP Address ปลายทางอยู่คนละเครือข่ายกัน โดยโปรโตคอล IP จะอ่านค่า IP Address ของเราเตอร์ เพื่อเตรียมส่งข้อมูลไปที่เราเตอร์แทน ซึ่งในที่นี้จะมีการกำหนดเป็นค่าเราเตอร์มาตรฐาน

2. โปรโตคอล IP จะเรียกใช้บริการโปรโตคอล ARP เพื่อแปลงค่า IP address ให้เป็นค่าหมายเลขฮาร์ดแวร์

3. โปรโตคอล IP ส่งข้อมูล IP คาต้าแกรมไปยังที่กำหนดไว้

โปรโตคอล IP จะรู้ได้อย่างไรว่าเครือข่ายดังกล่าวมีการเชื่อมต่อเราเตอร์อยู่ และมีค่า IP อะไร ซึ่งในเรื่องนี้ผู้ใช้จะต้องกำหนดค่าที่เรียกว่าดีฟอลต์เราเตอร์ หรือ ดีฟอลต์เกตเวย์เสียก่อน ว่ามีค่าหมายเลข IP อะไร โดยสอบถามได้จากผู้ดูแลระบบ

83075

2.2 หลักการพื้นฐานของ ZigBee

ZigBee เป็นโปรโตคอลของเครือข่ายไร้สาย ที่ออกแบบมาใช้สำหรับการตรวจนับและควบคุมเครือข่ายที่ใช้อัตราการส่งข้อมูลต่ำโดยเฉพาะ โดยมีแอปพลิเคชันจำนวนมากที่ได้รับประโยชน์จากโปรโตคอลนี้ เช่น การสร้างระบบเครือข่ายอัตโนมัติ ระบบรักษาความปลอดภัยภายในบ้าน ระบบเครือข่ายในโรงงานอุตสาหกรรม และแอปพลิเคชันอื่นๆ

เมื่อเปรียบเทียบกับโปรโตคอลไร้สายอื่นๆแล้ว โปรโตคอล ZigBee มีความซับซ้อนน้อยกว่าจึงช่วยลดทรัพยากรที่ต้องใช้งาน และที่สำคัญที่สุดมาตรฐานที่กำหนดไว้ เนื่องจากมีช่วงความถี่ที่สามารถใช้งานได้มีสามช่วง และความสามารถในการปรับแต่งเครือข่ายและความปลอดภัยของระบบ ซึ่งสามารถนำไปทดแทนการเชื่อมต่อแบบ RS-422 และ RS-485 ได้

2.2.1 IEEE 802.15.4 กับ ZigBee

โปรโตคอล ZigBee ใช้ข้อกำหนด IEEE 802.15.4 เป็นชั้น Medium Access (MAC) และชั้นฟิสิกส์ของตัวมันเอง โดย IEEE 802.15.4 ได้กำหนดช่วงความถี่สำหรับการทำงานเป็นสามช่วงคือ 2.4 GHz, 915 MHz, 868 MHz ในแต่ละช่วงความถี่จะมีจำนวนช่องสัญญาณที่กำหนดอยู่ช่วงหนึ่ง เช่น ช่วงความถี่ 2.4 GHz จะมีช่องสัญญาณ 16 ช่อง (ช่องสัญญาณที่ 11-26), ช่วงความถี่ 915 MHz จะมีช่องสัญญาณ 10 ช่อง (ช่องสัญญาณที่ 1-10) และที่ 868 MHz จะมีช่องสัญญาณ 1 ช่อง (ช่องสัญญาณที่ 0)



รูปที่ 2.11 เปรียบเทียบช่วงความถี่ของมาตรฐานต่างๆ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อัตราการส่งข้อมูลของโปรโตคอลนี้ขึ้นอยู่กับช่วงความถี่ที่ใช้งาน ช่วงความถี่ 2.4 GHz จะให้อัตราการส่งข้อมูล 250 kbps, 915MHz จะให้อัตราการส่งข้อมูล 40 kbps, 868 MHz จะให้อัตราการส่งข้อมูล 20 kbps อัตราการส่งข้อมูลที่แท้จริงจะน้อยกว่าที่กำหนดไว้เนื่องจากการเพิ่มเสดเดอร์และความล่าช้าที่เกิดในกระบวนการทำงาน

ความยาวสูงสุดของ MAC แพ็กเก็ตคือ 127 ไบต์ รวมทั้งมีค่า CRC 16 บิตที่ทำหน้าที่ตรวจสอบความถูกต้องของเฟรม นอกจากนี้ IEEE 802.15.4 ใช้กระบวนการยืนยันการได้รับข้อมูล (acknowledgement) ซึ่งกระบวนการนี้ได้กล่าวถึงในเรื่อง TCP/IP ที่ผ่านมาแล้ว

ชนิดของอุปกรณ์

IEEE 802.15.4 กำหนดอุปกรณ์ไว้สองชนิด ตามตารางที่ 2.1

ตารางที่ 2.1 ชนิดของอุปกรณ์ IEEE 802.15.4

ชนิดของอุปกรณ์	งานที่สามารถใช้ได้	แหล่งพลังงาน	การตั้งค่าที่เครื่องรับ
Full Function Device (FFD)	ทุกประเภท	หลัก	เปิดเมื่อไม่ได้ใช้
Reduced Function Device (RFD)	จำกัด	แบตเตอรี่	ปิดเมื่อไม่ได้ใช้

2.2.2 จุดเด่นของ ZigBee

สามารถติดต่อกันได้ระยะไกล

- ประมาณ 100 เมตร ถ้ามีสิ่งกีดขวาง และ 1500 เมตรในที่โล่ง
- มีกำลังส่ง 100 mW (20 dBm)
- กำลังในการรับ -100 dBm
- อัตราการส่งข้อมูล 250,000 bps

ระบบเครือข่ายและความปลอดภัยสูง

- จะพยายามติดต่อใหม่ถ้าหลุดจากเครือข่ายและมีการใช้ Acknowledgements ด้วย
- DSSS (Direct Sequent Spread Spectrum)
- ในแต่ละช่องการสื่อสารสามารถใช้หมายเลขเครือข่ายได้ถึง 65,000 หมายเลข
- มีการใช้หมายเลขต้นทางและหมายเลขปลายทางในการติดต่อ
- สามารถสื่อสารกันแบบยูนิแคสและบรอดแคสได้

- สนับสนุนการเชื่อมต่อเครือข่ายแบบ point-to-point, point-to-multipoint และ peer-to-peer

ใช้พลังงานต่ำ

- กระแสของภาคส่ง 215 mA ที่ 3.3 V
- กระแสของภาครับ 55 mA ที่ 3.3 V
- กระแส Power-down น้อยกว่า 10 μ A

ง่ายต่อการนำไปใช้งาน

- ไม่จำเป็นต้องตั้งค่าสำหรับการสื่อสารแบบวิทยุ
- มีซอฟต์แวร์ X-CTU สำหรับการทดสอบและตั้งค่าอุปกรณ์
- มีโหมดคอมมานด์ 2 แบบคือ AT และ API สำหรับการตั้งค่าโมดูล
- มีชุดคำสั่งที่ครอบคลุม
- มีขนาดเล็ก

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.2 ข้อมูลจำเพาะของโมดูล ZigBee

Specification	XBee	XBee-PRO
Performance		
Indoor/Urban Range	up to 100 ft. (30 m)	Up to 300' (100 m)
Outdoor RF line-of-sight Range	up to 300 ft. (100 m)	Up to 1 mile (1500 m)
Transmit Power Output (software selectable)	1mW (0 dBm)	60 mW (18 dBm) conducted, 100 mW (20 dBm) EIRP*
RF Data Rate	250,000 bps	250,000 bps
Serial Interface Data Rate (software selectable)	1200 - 115200 bps (non-standard baud rates also supported)	1200 - 115200 bps (non-standard baud rates also supported)
Receiver Sensitivity	-92 dBm (1% packet error rate)	-100 dBm (1% packet error rate)
Power Requirements		
Supply Voltage	2.8 - 3.4 V	2.8 - 3.4 V
Transmit Current (typical)	45mA (@ 3.3V)	If PL=0 (10dBm): 137mA (@3.3V), 139mA (@3.0V) PL=1 (12dBm): 155mA (@3.3V), 153mA (@3.0V) PL=2 (14dBm): 170mA (@3.3V), 171mA (@3.0V) PL=3 (16dBm): 188mA (@3.3V), 195mA (@3.0V) PL=4 (18dBm): 215mA (@3.3V), 227mA (@3.0V)
Idle / Receive Current (typical)	50mA (@ 3.3V)	55mA (@ 3.3 V)
Power-down Current	< 10 μ A	< 10 μ A
General		
Operating Frequency	ISM 2.4 GHz	ISM 2.4 GHz
Dimensions	0.960" x 1.067" (2.438cm x 2.711cm)	0.960" x 1.297" (2.438cm x 3.294cm)
Operating Temperature	-40 to 85° C (industrial)	-40 to 85° C (industrial)
Antenna Options	Integrated Whip, Chip or U.FL Connector	Integrated Whip, Chip or U.FL Connector
Networking & Security		
Supported Network Topologies	Point-to-point, Point-to-multipoint & Peer-to-peer	
Number of Channels (software selectable)	16 Direct Sequence Channels	12 Direct Sequence Channels
Addressing Options	PAN ID, Channel and Addresses	PAN ID, Channel and Addresses
Agency Approvals		
United States (FCC Part 15.247)	OUR-XBEE	OUR-XBEEPRO
Industry Canada (IC)	4214A XBEE	4214A XBEEPRO
Europe (CE)	ETSI	ETSI (Max. 10 dBm transmit power output)
Japan	กฉ	005NYCA0378 (Max. 10 dBm transmit power output)**

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 การกำหนดพินของโมดูล ZigBee

หมายเลขพิน	ชื่อ	รายละเอียด
1	VCC	จ่ายกระแสเข้า
2	DOUT	ข้อมูล UART ออก
3	DIN / CONFIG	ข้อมูล UART เข้า
4	DO8	ดิจิตอลเอาต์พุต 8
5	RESET	รีเซ็ตโมดูล
6	PWM0 / RSSI	PWM เอาต์พุต 0 / วัดความแรงของสัญญาณภาครับ
7	PWM1	PWM เอาต์พุต 1
8	[reserved]	-
9	DTR / SLEEP_RQ / DI8	ควบคุมโหมดสลีป / ดิจิตอลเอาต์พุต 8
10	GND	กราวด์
11	AD4 / DIO4	อนาล็อกอินพุต 4 / ดิจิตอล I/O 4
12	CTS / DIO7	Clear-to-Send / ดิจิตอล I/O 7
13	ON / SLEEP	ตรวจสอบสถานะของโมดูล
14	VREF	ความต่างศักย์อ้างอิงสำหรับ A/D อินพุต
15	Associate / AD5 / DIO5	อนาล็อกอินพุต 5 / ดิจิตอล I/O 5
16	RTS / AD6 / DIO6	Request-to-Send / อนาล็อกอินพุต 6 / ดิจิตอล I/O 6
17	AD3 / DIO3	อนาล็อกอินพุต 3 / ดิจิตอล I/O 3
18	AD2 / DIO2	อนาล็อกอินพุต 2 / ดิจิตอล I/O 2
19	AD1 / DIO1	อนาล็อกอินพุต 1 / ดิจิตอล I/O 1
20	AD0 / DIO0	อนาล็อกอินพุต 0 / ดิจิตอล I/O 0

- การเชื่อมต่ออย่างน้อยต้องมีพิน VCC, GND, DOUT และ DIN
- การเชื่อมต่อสำหรับอแพคเฟิร์มแวร์ VCC, GND, DIN, DOUT, RTS และ DTR
- โมดูลจะต้องต่อตัวต้านทาน 50k โอห์ม ที่ขา RESET

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

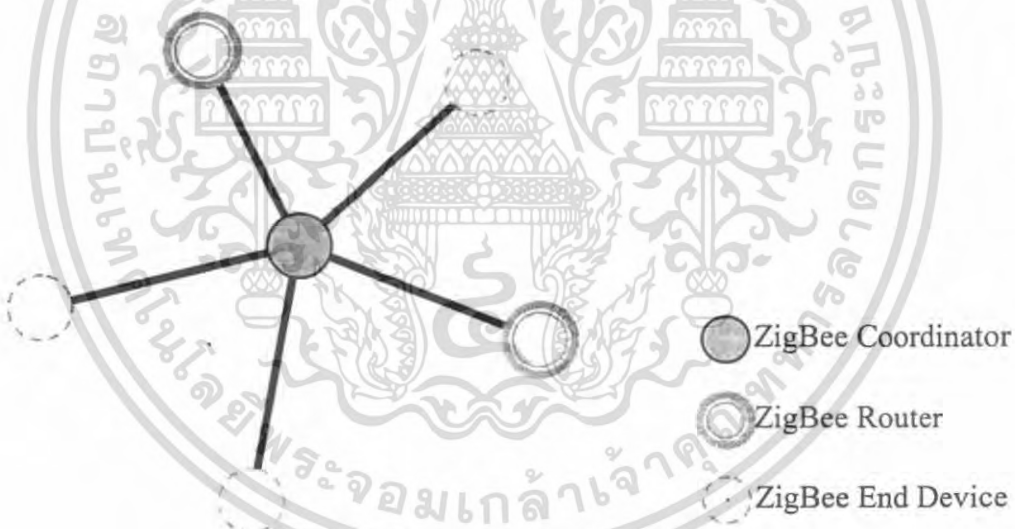
2.2.3 การตั้งค่าเครือข่าย

เครือข่ายไร้สายโดยใช้โปรโตคอล ZigBee สามารถตั้งค่าได้หลายแบบ ซึ่งสามารถแบ่งได้เป็นหมวดหมู่ใหญ่ๆ ได้ 2 แบบ คือ จุดเชื่อมต่อ และอุปกรณ์ปลายทาง

อุปกรณ์เชื่อมต่อของโปรโตคอล ZigBee เป็นอุปกรณ์ประเภท FFD ที่รวมการทำงานของโปรโตคอล ZigBee ไว้เป็นจำนวนมาก ส่วนอุปกรณ์ปลายทางสามารถเป็นได้ทั้ง FFD และ RFD ซึ่ง RFD เป็นอุปกรณ์ที่เล็กและง่ายที่สุดของโปรโตคอล ZigBee ที่มีการทำงานของโปรโตคอล ZigBee น้อยมาก

2.2.3.1 เครือข่ายแบบสตาร์ (Star Network)

ประกอบด้วยจุดเชื่อมต่อโปรโตคอล ZigBee 1 จุด และอุปกรณ์ปลายทางหลายๆชิ้น ในเครือข่ายแบบสตาร์ อุปกรณ์ปลายทางทั้งหมดจะสื่อสารกับอุปกรณ์เชื่อมต่อเท่านั้น ถ้าอุปกรณ์ปลายทางหนึ่ง ต้องการสื่อสารกับอุปกรณ์ปลายทางอื่นๆ ต้องส่งข้อมูลผ่านอุปกรณ์เชื่อมต่อเท่านั้น หรือกล่าวอีกนัยหนึ่งคือ อุปกรณ์เชื่อมต่อทำหน้าที่ส่งข้อมูลไปยังผู้รับ

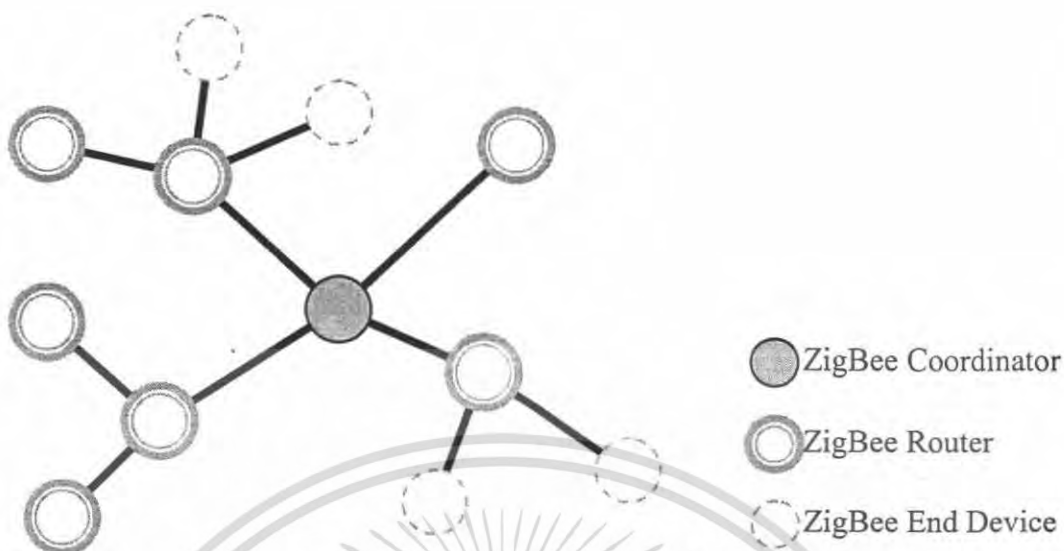


รูปที่ 2.12 เครือข่ายแบบสตาร์

2.2.3.2 เครือข่ายแบบต้นไม้ (Cluster Tree)

ในเครือข่ายนี้ อุปกรณ์ปลายทางจะสามารถเชื่อมต่อได้กับอุปกรณ์เชื่อมต่อหรือ ZigBee โปรโตคอลเร้าเตอร์ เร้าเตอร์ทำหน้าที่ 2 ประเภท คือ เพิ่มจำนวนโหนดที่สามารถเชื่อมต่ออยู่บนเครือข่ายและขยายขนาดของเครือข่าย เนื่องจากเร้าเตอร์จะทำหน้าที่ส่งข้อมูลไปยังจุดต่างๆของเครือข่ายได้โดยที่ อุปกรณ์ปลายทางไม่จำเป็นต้องอยู่ในระยะการส่งสัญญาณวิทยุ

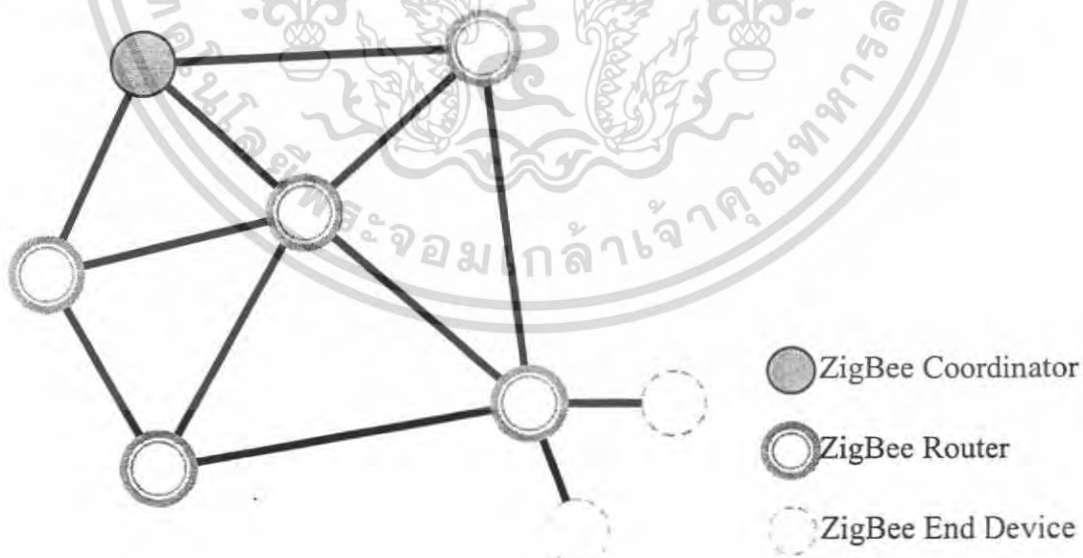
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.13 เครือข่ายแบบต้นไม้

2.2.3.3 เครือข่ายแบบเมช (Mesh Network)

เครือข่ายแบบเมชใช้กับเครือข่ายแบบต้นไม้ ยกเว้นอุปกรณ์ FFD สามารถส่งข้อมูลไปยัง FFD อื่นได้โดยตรง ไม่ต้องผ่านโครงสร้างต้นไม้ ข้อมูลที่ส่งไปยัง RFD จะต้องผ่านอุปกรณ์ RFD ก่อนหน้า ข้อดีของการเชื่อมต่อแบบนี้คือ ช่วยลดอัตราความล่าช้าของการส่งและเพิ่มความน่าเชื่อถือของระบบ



รูปที่ 2.14 เครือข่ายแบบเมช

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เครือข่ายแบบต้นไม้และเครือข่ายแบบเมชมีอีกชื่อว่า เครือข่ายหลายจุด (milti-hop) ขณะที่เครือข่ายแบบสตาร์เป็น เครือข่ายจุดเดียว (single-hop) เครือข่ายโปรโตคอล ZigBee เป็นเครือข่ายแบบเชื่อมต่อได้หลายอุปกรณ์พร้อมกัน ซึ่งหมายความว่า จุดเชื่อมต่อในเครือข่ายทุกจุดมีสิทธิในการเข้าถึงตัวกลางที่ใช้ในการสื่อสารเท่าๆกัน

มีวิธีการเชื่อมต่อแบบหลายอุปกรณ์พร้อมกัน 2 วิธี ได้แก่ beacon และ non-beacon การเชื่อมต่อแบบ non-beacon ทุกจุดเชื่อมต่อในเครือข่ายส่งข้อมูลได้ตลอดเวลาที่ช่องสัญญาณว่างอยู่ ในเครือข่ายแบบ beacon จุดเชื่อมต่อจะสามารถส่งข้อมูลได้ในเวลาที่ถูกกำหนดไว้ล่วงหน้าเท่านั้น

ตารางที่ 2.4 การเปรียบเทียบมาตรฐานการติดต่อสื่อสารแบบไร้สาย

มาตรฐาน	ZigBee® 802.15.4	Wi-Fi™ 802.11b	Bluetooth™ 802.15.1
ช่วงการส่ง(เมตร)	1 – 100*	1 - 100	1 – 10
อายุการใช้งาน(วัน)	100 – 1,000	0.5 – 5.0	1 - 7
ขนาดเครือข่าย(ต่อโหนด)	> 64,000	32	7
แอปพลิเคชัน	Monitoring & Control	Web, Email, Video	Cable Replacement
ขนาดสแต็ก(กิโลไบต์)	4 – 32	1,000	250
Throughput (kb/s)	20 – 250	11,000	720

2.2.4 โพรไฟล์โปรโตคอล ZigBee

โพรไฟล์โปรโตคอล ZigBee เป็นคำอธิบายขององค์ประกอบทางตรรกะของอุปกรณ์และส่วนติดต่อ ไม่ต้องมีรหัสในการสร้างโพรไฟล์ ข้อมูลโพรไฟล์แต่ละส่วนจะสามารถส่งผ่านระหว่างอุปกรณ์ได้คล้ายกับสถานะของสวิทช์ ถูกเรียกว่าแอททริบิวต์ (attribute) แต่ละแอททริบิวต์จะมีเอกลักษณ์เฉพาะตัว และแอททริบิวต์เหล่านี้จะถูกจัดกลุ่มอยู่ในคลัสเตอร์ (cluster) และแต่ละคลัสเตอร์ก็จะมีเอกลักษณ์เฉพาะตัว โดยส่วนติดต่อจะถูกกำหนดในระดับคลัสเตอร์ ไม่ใช่ที่ระดับแอททริบิวต์

โพรไฟล์จะกำหนดค่าหมายเลขของแอททริบิวต์ และหมายเลขคลัสเตอร์ นอกจากนี้ยังมีรูปแบบในแต่ละแอททริบิวต์ ตัวอย่างเช่น ในการควบคุมภายในบ้าน โพรไฟล์การส่องสว่าง คลัสเตอร์ “OnOffDRC” ของอุปกรณ์รีโมทควบคุมค่าความต้านทาน (Dimmer Remote Control) จะมีเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

แอมบิตรีบิตต์ “OnOff” จะต้องมีค่าเป็นสัญญาณ 8 บิต ซึ่งค่า 0xFF หมายถึง “เปิด” ค่า 0x00 หมายถึง “ปิด” และค่า 0xF0 หมายถึงสลับค่าเข้าที่พูด

โพรไฟล์ยังกำหนดคลัสเตอร์ที่จำเป็นและที่เป็นตัวเลือกสำหรับแต่ละอุปกรณ์ นอกจากนี้เพิ่มโพรไฟล์อาจนิยามตัวเลือกบางโปรโตคอล ZigBee บริการแมนคาทอรี่

ผู้ใช้สามารถนำค่านิยามและเขียนรหัสใช้มันได้ มันสามารถเขียนรหัสได้หลายแบบ หลายกลุ่มฟังก์ชันที่สนับสนุนแมนคาทอรี่คลัสเตอร์ และการบริการและใช้แอมบิตรีบิตต์ที่อยู่ในโพรไฟล์ ในทางนี้หนึ่งสวิตช์จะทำงานกับโพลคอนโทรลเลอร์ที่สร้างขึ้น

ตัวอย่างเช่น การควบคุมภายในบ้าน, โพรไฟล์การส่องสว่าง 6 อุปกรณ์พิเศษ สำหรับ ZigBee โปรโตคอลให้การสนับสนุนสำหรับค่าโพรไฟล์ หัวโพรไฟล์กับข้อมูลตามนี้

- Profile ID
- อุปกรณ์ IDs และ Version
- Cluster IDs
- แอมบิตรีบิตต์ IDs
- ชนิดข้อมูลแอมบิตรีบิตต์

แต่ละกลุ่มฟังก์ชันของรหัส การสนับสนุนหนึ่งคลัสเตอร์หรือมากกว่านั้น ถูกเรียกว่า Endpoint การสื่อสารอุปกรณ์ที่ต่างกันมี endpoint และคลัสเตอร์ที่สนับสนุน

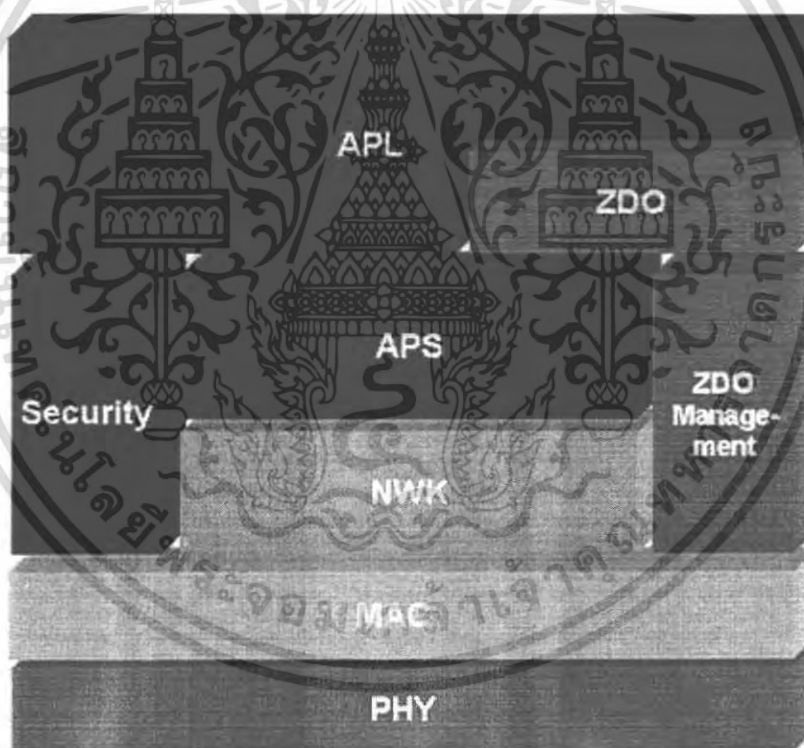
2.2.5 ZigBee โปรโตคอลสแตก

ชั้นสแตก ZigBee ประกอบด้วย ชั้นเครือข่าย ชั้นแอปพลิเคชัน และการให้บริการรักษาความปลอดภัย (SSP) อุปกรณ์แต่ละตัวของ ZigBee ควรมีลักษณะพิเศษที่สามารถใช้งานได้ทั้งแบบสาธารณะและส่วนตัว ซึ่งรูปแบบนั้นจะถูกนิยามตามสภาพแวดล้อมของแอปพลิเคชัน ชนิดของอุปกรณ์คลัสเตอร์ถูกใช้สำหรับการติดต่อสื่อสาร รูปแบบการติดต่อแบบสาธารณะรับรองการให้บริการที่ต่างกันของผู้ให้บริการสำหรับที่ว่างแอปพลิเคชันเดียวกัน

อุปกรณ์ถูกนิยาม โดยรูปแบบและถูกประกอบด้วยวัตถุแอปพลิเคชันต่างๆ แต่ละวัตถุแอปพลิเคชันถูกติดต่อกับส่วนพักของสแตก ZigBee โดย endpoint ซึ่งมีที่อยู่องค์ประกอบในอุปกรณ์ การติดต่อสื่อสารถูกสร้างจาก endpoint ไปยัง endpoint และส่งผ่านโครงสร้างข้อมูลที่ถูกเรียกโดยคลัสเตอร์ คลัสเตอร์จะมีกลุ่มของแอมบิตรีบิตต์ที่จำเป็นสำหรับการใช้ข้อมูลร่วมกัน ระหว่างวัตถุแอปพลิเคชัน คลัสเตอร์ถูกใช้ในแอปพลิเคชันพิเศษซึ่งถูกนิยามลงในรูปแบบของมันเอง แต่ละอินเตอร์เฟสสามารถส่งหรือรับข้อมูลในรูปแบบของคลัสเตอร์ได้

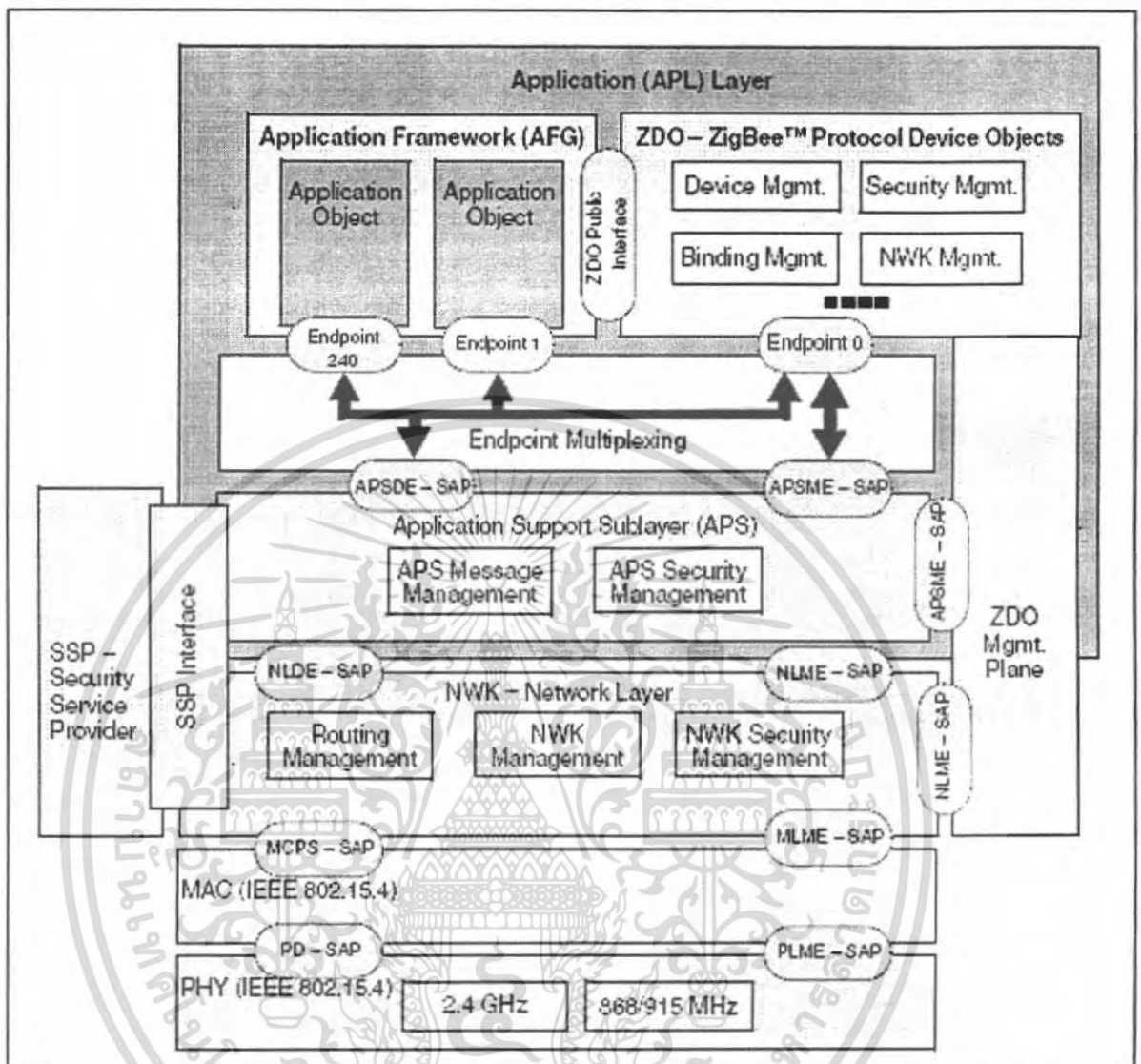
0 และ 255 เป็น endpoint พิเศษถูกใช้สำหรับการตั้งค่าและการจัดการของการเข้าถึงอุปกรณ์ ZigBee แอปพลิเคชันสามารถติดต่อสื่อสารกับชั้นอื่นๆของสแตก ZigBee ที่ติดตั้ง และตั้งค่าไว้แล้วได้ การเพิ่ม endpoint 0 เป็นวัตถุของอุปกรณ์ ZigBee (ZigBee Device Object) endpoint 255 ถูกใช้เพื่อส่งไปยังทุกๆ endpoint ขณะที่ endpoint 241 – 254 ถูกจองไว้ แอปพลิเคชันที่สนับสนุนชั้นที่ต่ำกว่าที่ติดต่อกับ endpoint กับ ชั้นเครือข่าย และ SSP เป็นตัวช่วยให้ endpoint ทั้งหมด กับการส่งข้อมูล การรักษาความปลอดภัย และการ Binding (การ Binding เป็นความสามารถในการจับความแตกต่างแต่ละอุปกรณ์เข้าด้วยกัน เช่น สวิตช์)

ชั้นเครือข่ายใช้อุปกรณ์ในการติดต่อสื่อสารกับอุปกรณ์อื่นๆ ในการหาเส้นทางของข้อความ และการค้นหาเครือข่าย การสนับสนุนแอปพลิเคชันในชั้นที่ต่ำกว่าถูกให้บริการในชั้นนี้ แอปพลิเคชันสามารถคอนฟลิกและส่งผ่านค่าพารามิเตอร์ของชั้นเครือข่ายผ่านทาง ZDO



รูปที่ 2.15 สแตกโปรโตคอล ZigBee

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 2.16 โครงสร้างสถาปัตยกรรมของสแตกโปรโตคอล ZigBee

2.2.5 การรักษาความปลอดภัย

SSP เป็นตัวรักษาความปลอดภัย อย่างไรก็ตาม การรักษาความปลอดภัยของระบบถูกนิยามที่รูปแบบระดับของโพรไฟล์ ซึ่งการนิยามชนิดขององค์ประกอบการรักษาความปลอดภัยในเครือข่ายพิเศษ แต่ละชั้น MAC เครือข่าย และชั้นแอปพลิเคชัน สามารถรักษาความปลอดภัย และใช้กุญแจรักษาความปลอดภัยร่วมกันได้เพื่อเป็นการลดการเก็บความต้องการ SSP ถูกติดตั้งและคอนฟลิกผ่านทาง ZDO และต้องการองค์ประกอบพิเศษ

การถอดรหัสพื้นฐาน ZigBee พิเศษนิยมให้การใช้ “trust center” หรือ การไว้ใจศูนย์กลาง เหมือนกับเป็นอุปกรณ์ในเครือข่ายที่ใช้กระจายกุญแจรักษาความปลอดภัย

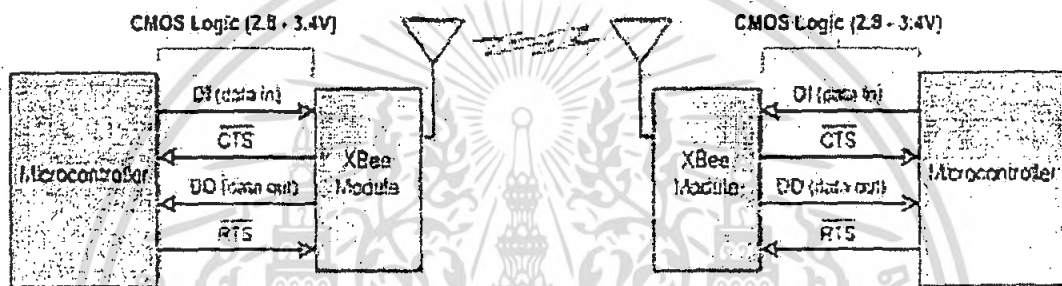
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.6 การติดต่อสื่อสารของ ZigBee ผ่านทางพอร์ตอนุกรม

ZigBee อินเทอร์เน็ตเฟสกับอุปกรณ์ผู้ใช้ (host) ผ่านพอร์ตอนุกรม อะซิงโครนัสระดับตรรกะส่งผ่าน โมดูลทางพอร์ตอนุกรม โมดูลสามารถติดต่อสื่อสารกับตรรกะต่างๆเข้ากับ UART หรือระดับตัวแปลกับอุปกรณ์อนุกรมต่างๆ

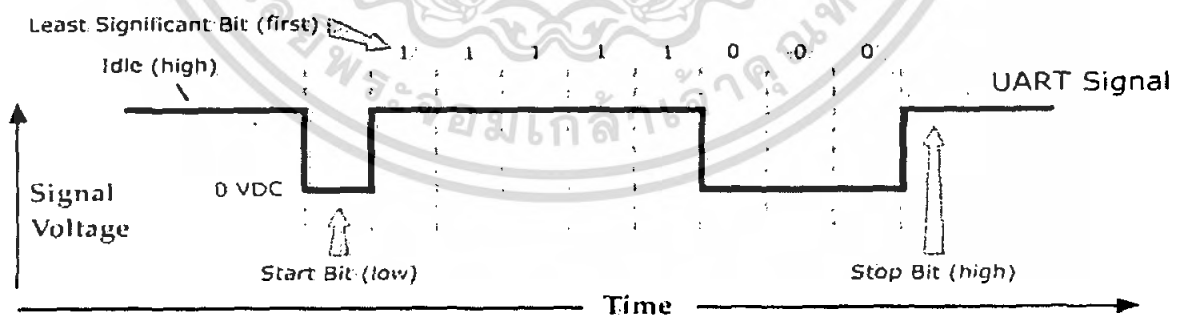
2.2.6.1 UART การไหลของข้อมูล

อุปกรณ์มีอินเทอร์เน็ตเฟส UART สามารถติดต่อโดยตรงกับพินของ RF โมดูล แสดงในรูปแบบด้านล่าง



รูปที่ 2.17 ระบบคาล์วไฟฟ้าไออะแกรมในอุปกรณ์ที่เชื่อมต่อ UART

ข้อมูลส่งเข้า UART โมดูล ผ่านขาได้อิน (ขา3) ด้วยสัญญาณอนุกรมอะซิงโครนัส สัญญาณจะว่างเมื่อไม่มีข้อมูลถูกส่ง แต่ละข้อมูลไบต์ประกอบด้วย บิตเริ่มต้น (ต่ำ) 8 ข้อมูลบิต และ บิตหยุด (สูง) ตามรูปอธิบายแผนบิตอนุกรมของการผ่านข้อมูลของโมดูล



รูปที่ 2.18 แฟ้มเกิดข้อมูล UART 0x1F (31 ในเลขฐานสิบ) ถูกส่งผ่านโมดูล RF

รูปแบบงาน โมดูล UART ดังนั้นเวลาและพาริตีตรวจสอบต้องการข้อมูลติดต่อสื่อสาร การสื่อสารอนุกรมขึ้นอยู่กับ 2 UARIS ถูกจัดโครงสร้างกับการจัดตั้ง (อัตราบอด, พาริตี, บิตเริ่มต้น, บิตหยุด, บิตข้อมูล)

2.3 หลักการทำงานไมโครคอนโทรลเลอร์ตระกูล PIC18F97J60

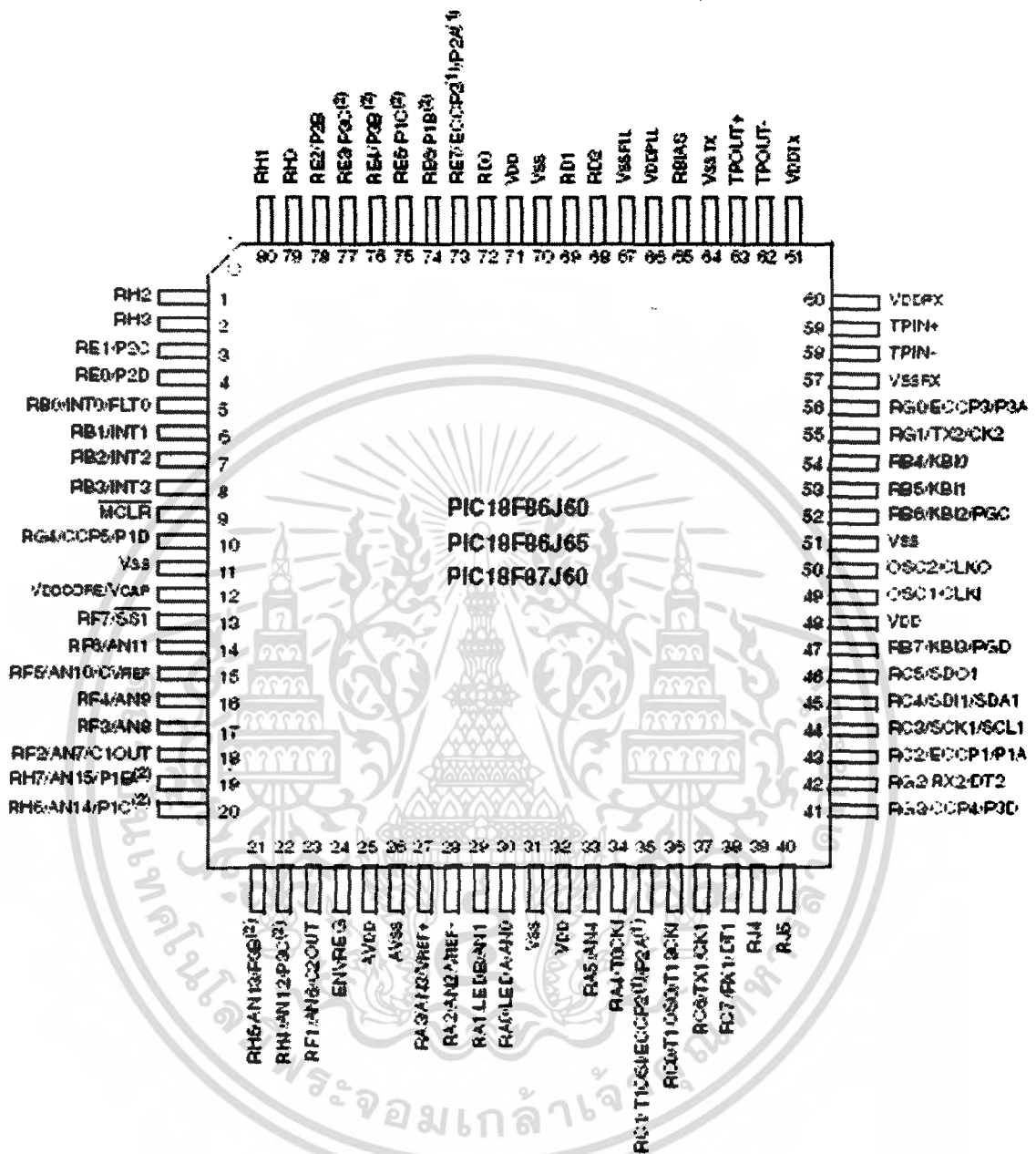
2.3.1 ลักษณะหลักการ

อุปกรณ์ทั้งหมดในตัวควบคุมตระกูลนี้ สัมพันธ์ช่วงของลักษณะที่สามารถเข้าสัญญาณการลดกำลังงานที่ใช้ระหว่างการปฏิบัติการ หัวใจสำคัญมีลักษณะดังนี้

- Alternate Run Mode สัญญาณนาฬิกาใส่ในไมโครคอนโทรลเลอร์ที่เวลา(Timer1)หรือ RC ภายในออสซิลเลเตอร์ กำลังงานที่ใช้ระหว่างการเข้ารหัสสามารถถูกลดมากถึง 90%

- Multiple Idle Mode ตัวควบคุมสามารถแสดงผล โดยไม่ใช่ซีพียูหลัก แต่กระบวนการยังคงทำงานอยู่ในสถานะนี้ กำลังงานที่ใช้สามารถถูกลดลงได้ในอนาคตเท่ากับ4%ของความต้องการในสถานะปกติ

- On-the-Fly Mode Switching การจัดการกำลังงานลักษณะนี้ถูกเรียกโดยรหัสของผู้ใช้ในระหว่างการปฏิบัติการ การอนุญาตผู้ใช้ต้องสอดคล้องกับความคิดประหยัดกำลังงานในการออกแบบแอปพลิเคชันซอฟต์แวร์



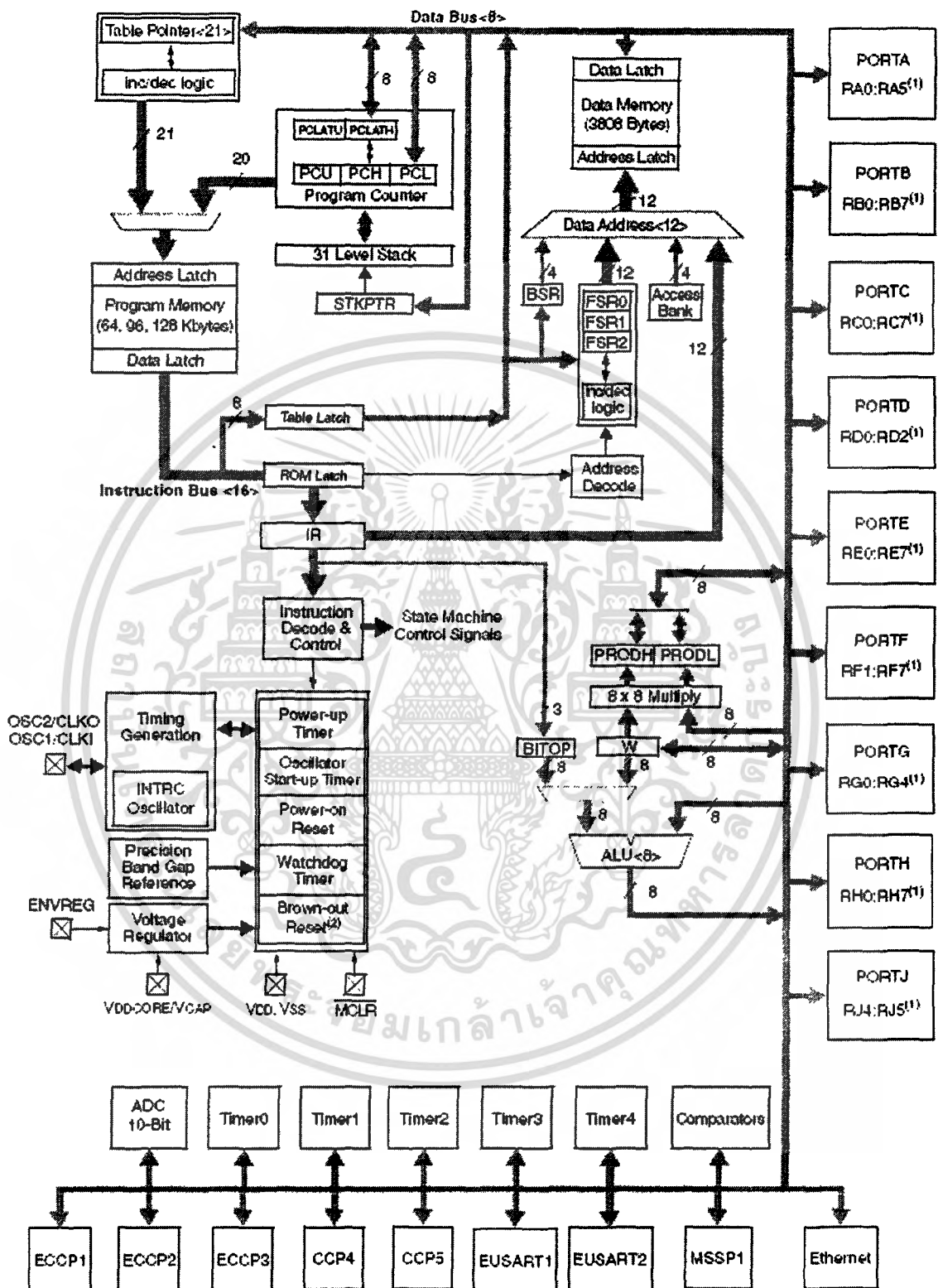
รูปที่ 2.19 ไดอะแกรมขาของ PIC ตระกูล PIC18F97J60 ชนิด 80 ขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 2.3 ลักษณะอุปกรณ์ของ PIC ตระกูล PIC18F97J60 ชนิด 80 ขา

ลักษณะ	PIC18F86J65
การปฏิบัติการความถี่	DC – 41.67 MHz
หน่วยความจำโปรแกรม(ไบต์)	96K
หน่วยความจำโปรแกรม(โครงสร้าง)	49148
หน่วยความจำข้อมูล(ไบต์)	3808
แหล่งที่มาอินเตอร์รัพ	27
พอร์ต เข้า/ออก	พอร์ต A, B, C, D, E, F, G, H, J
พิน เข้า/ออก	55
เวลา	5
การจับ/เปรียบเทียบ/โมดูลPWM	2
การจับประยุกต์/เปรียบเทียบ/โมดูลPWM	3
การติดต่อสื่อสารพอร์ตอนุกรม	MSSP (1), Enhanced USART (2)
การติดต่อสื่อสารพอร์ตอีเทอร์เน็ต(10Base-T)	ได้
การติดต่อสื่อสารพอร์ตสเลฟขนาน(PSP)	ไม่
บัสหน่วยความจำภายนอก	ไม่
10 บิต โมดูลนาฬิกา-ดิจิทัล	15ช่องสัญญาณขาเข้า
การเริ่มใหม่(และความหน่วงเวลา)	POR, BOR, โครงสร้างRESET,สแตกเต็ม, สแตกไม่ไหล, MCLR , WDT (PWRT, OST)
จัดโครงสร้าง	75 โครงสร้าง, 83 กีบจัดโครงสร้างประยุกต์ให้ ทำงาน
แพ็คเกจ	80-พิน TQFP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



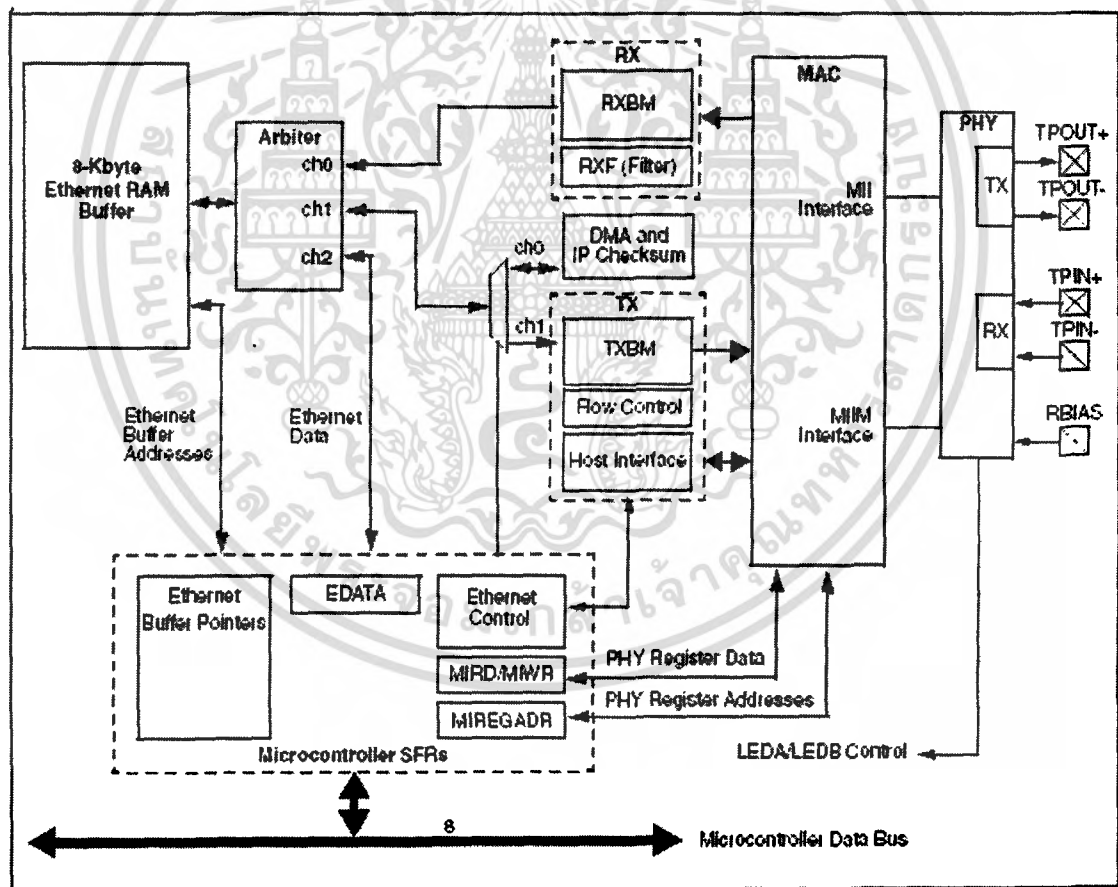
รูปที่ 2.20 บล็อกไดอะแกรมของ PIC ตระกูล PIC18F97J60 ชนิด 80 ขา

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.3.2 อีเทอร์เน็ตโมดูล

PIC ตระกูล PIC18F97J60 มีคุณสมบัติเป็นโมดูลควบคุมอีเทอร์เน็ตแบบฝังตัว ซึ่งใช้ในการแก้ปัญหาการเชื่อมต่อได้อย่างสมบูรณ์ ซึ่งประกอบด้วย ชั้น MAC และการส่งผ่านชั้น PHY การส่งสัญญาณทั้งสองและองค์ประกอบบางส่วนที่ส่งผ่านต้องเชื่อมต่อไมโครคอนโทรลเลอร์โดยผ่านทางเครือข่ายอีเทอร์เน็ต

อีเทอร์เน็ตโมดูลทั้งหมดของ IEEE 802.3 แบบพิเศษที่ใช้ในการเชื่อมต่อ 10-BaseT บนเครือข่ายสายเกลียวคู่ จะจัดการตัวเลขของตารางตัวกรองความถี่แพ็กเก็ต และจำกัดแพ็กเก็ตที่เข้ามาให้บริการโมดูล DMA ภายในสำหรับข้อมูลความเร็วอัตราการส่งผ่านข้อมูล และตัวช่วยเหลือนฮาร์ดแวร์ในการคำนวณความถูกต้องของข้อมูล และให้บริการในการสร้างขาออกของ LED และการทำงานของบล็อกไดอะแกรมของโมดูล ดังรูปข้างล่างนี้

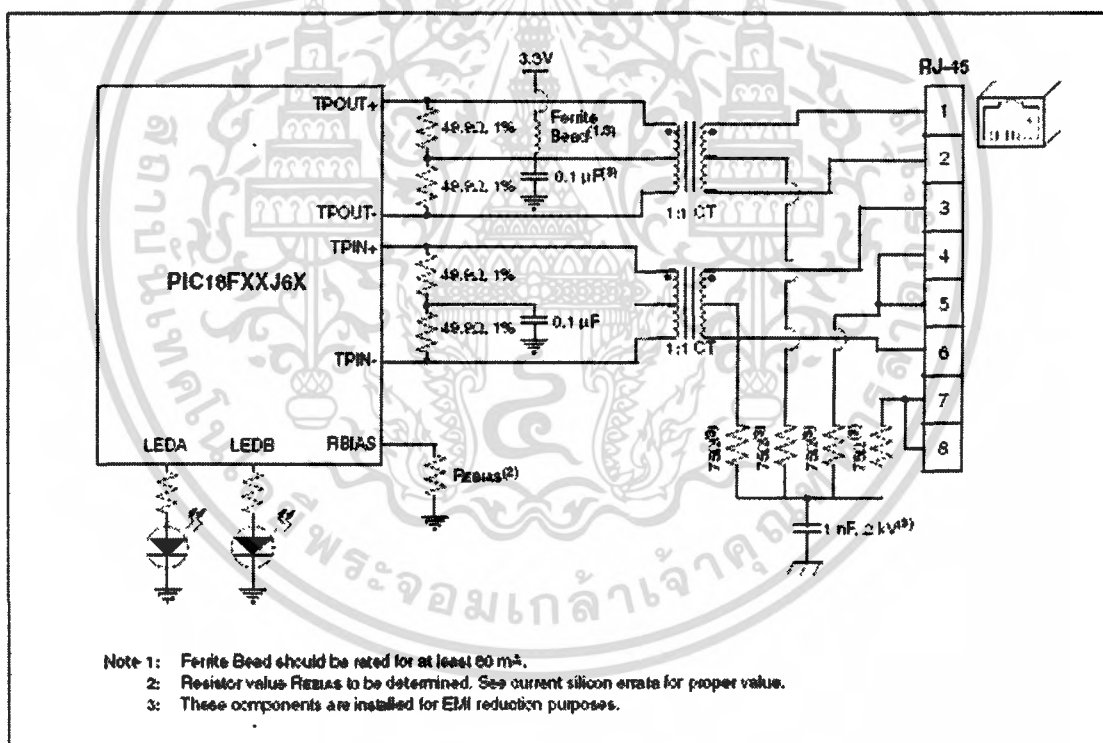


รูปที่ 2.21 บล็อกไดอะแกรมอีเทอร์เน็ตโมดูล

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อีเทอร์เน็ต โมดูลประกอบด้วยฟังก์ชันหลัก 5 ฟังก์ชัน

1. โมดูลการส่ง PHY เข้ารหัส และการถอดรหัสข้อมูลอนาล็อกถูกใช้ในอินเทอร์เฟซสาย เกลียวคู่ และส่งหรือรับบนเครือข่าย
2. โมดูล MAC องค์กรประกอบ IEEE 802.3 ตรวจจับ MAC และการให้บริการจัดการ อินเทอร์เน็ตสื่ออิสระ (MIIM) ที่ควบคุม PHY
3. ฟังก์ชันอิสระ บัฟเฟอร์ RAM 8 กิโลไบต์สำหรับการเก็บแพ็กเก็ตที่ถูกรับและส่ง
4. การตัดสินใจที่ส่งผ่านที่บัฟเฟอร์ RAM เมื่อความต้องการถูกสร้างขึ้นสำหรับไมโครคอลโทรเลอร์หลัก DMA การส่งและรับบล็อก
5. อินเทอร์เน็ตรีจิสเตอร์ที่ฟังก์ชันเหมือนการแสดงคำสั่ง และสถานะสัญญาณในระหว่าง โมดูลและ SFRs ของไมโครคอลโทรเลอร์



รูปที่ 2.22 องค์ประกอบที่ต้องการภายนอกของกระบวนการอีเทอร์เน็ต

- หมายเหตุ
1. Ferrite Bead ควรมีค่าอย่างน้อย 80 mA
 2. ค่าตัวต้านทาน REBIAS กำหนดโดยดูจากกระแสของซิลิกอน errata ที่เตรียมค่าไว้
 3. ส่วนประกอบเหล่านี้ต่อไว้เพื่อลดทอนค่า EMI

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 3

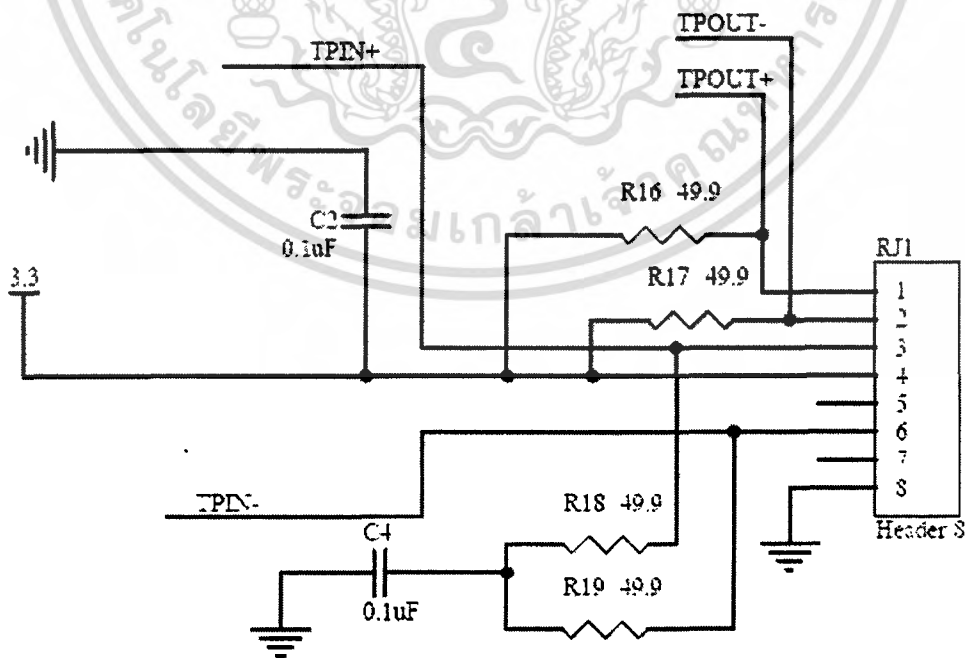
การออกแบบโครงงาน

3.1 การออกแบบฮาร์ดแวร์

การออกแบบฮาร์ดแวร์มีอยู่ 3 ส่วนคือ บอร์ดไมโครคอนโทรลเลอร์และบอร์ด ZigBee จากนั้นได้พัฒนาให้รวมวงจรทั้ง 2 บอร์ดเข้าด้วยกันและมีขนาดเล็กลง โดยมีวงจรต่างๆดังภาพ

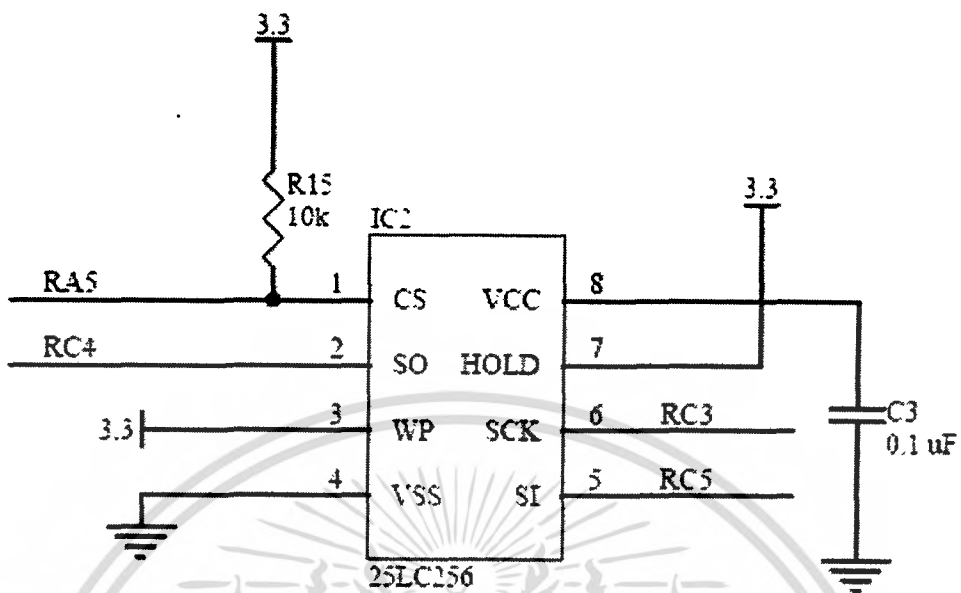
3.1.1 บอร์ดไมโครคอนโทรลเลอร์

- พอร์ต RJ45 ทำหน้าที่เป็นตัวรับข้อมูลจากอินเทอร์เน็ตส่งต่อให้ไมโครคอนโทรลเลอร์
- EEPROM เป็นหน่วยความจำภายนอกสำหรับไมโครคอนโทรลเลอร์เก็บข้อมูล
- ภาควิทยาระบบจะมีบริดจ์เพื่อแปลงกระแสสลับให้เป็นกระแสตรงแล้วจึงใช้เรกูเลเตอร์ลดแรงดันลงจาก 12 โวลต์ เป็น 3.3 โวลต์ โดยใช้ LM1117T3.3
- ไอซี 3232 สำหรับแปลงระดับแรงดันให้เป็นแบบ TTL สำหรับพอร์ตอนุกรมที่เชื่อมต่ออุปกรณ์อื่นๆต่อไป
- PIC16F86J65 เป็น MCU ซึ่งสามารถจัดการข้อมูลอินเทอร์เน็ตได้ภายในตัว ซึ่งจะใช้การโปรแกรมผ่านมาตรฐาน ICD2 และมีคริสตอลเพื่อป้อน clock ทำให้ MCU ทำงานได้

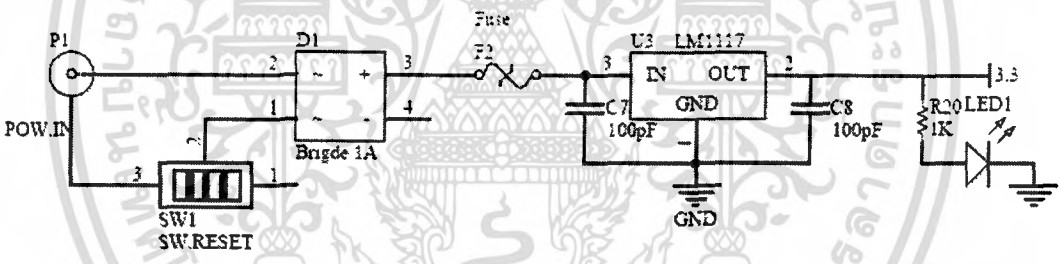


รูปที่ 3.1 วงจรพอร์ต RJ45

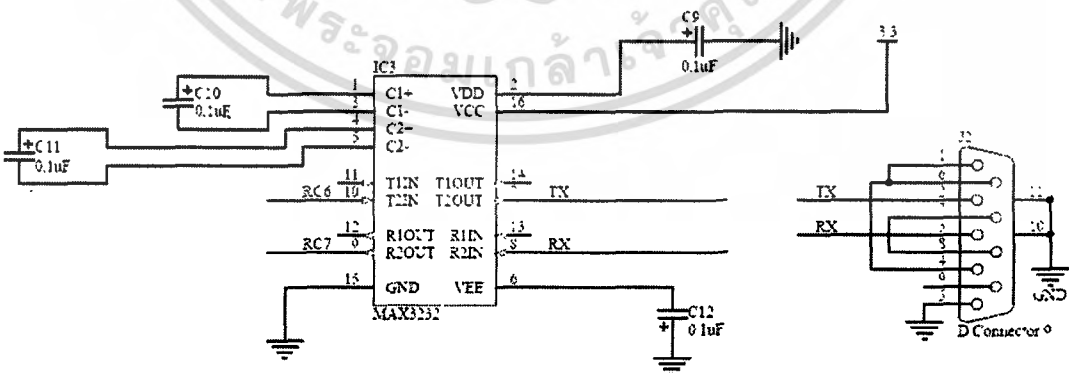
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.2 วงจร EEPROM

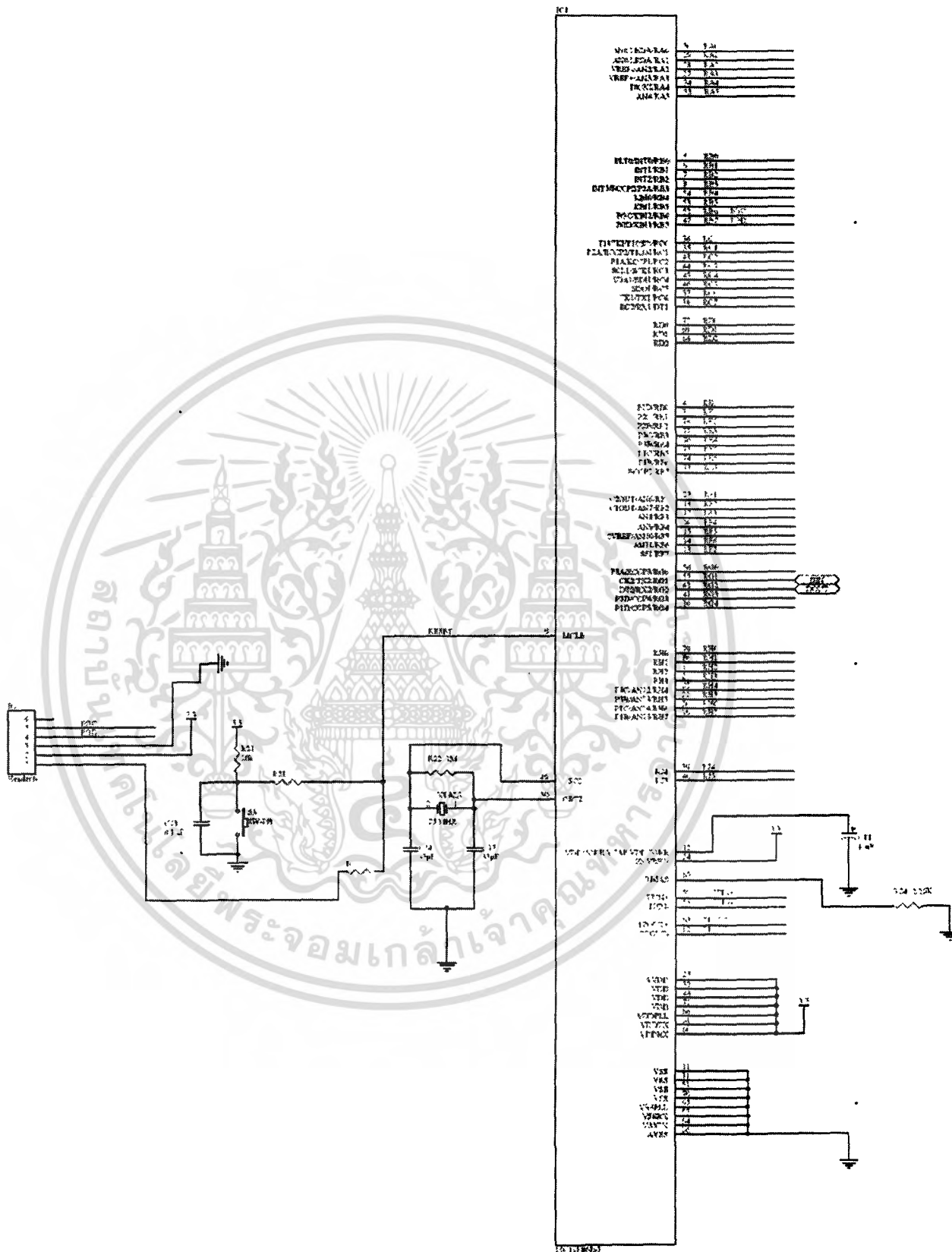


รูปที่ 3.3 วงจรจ่ายกระแสไฟ



รูปที่ 3.4 วงจร MAX3232 และ RS232

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

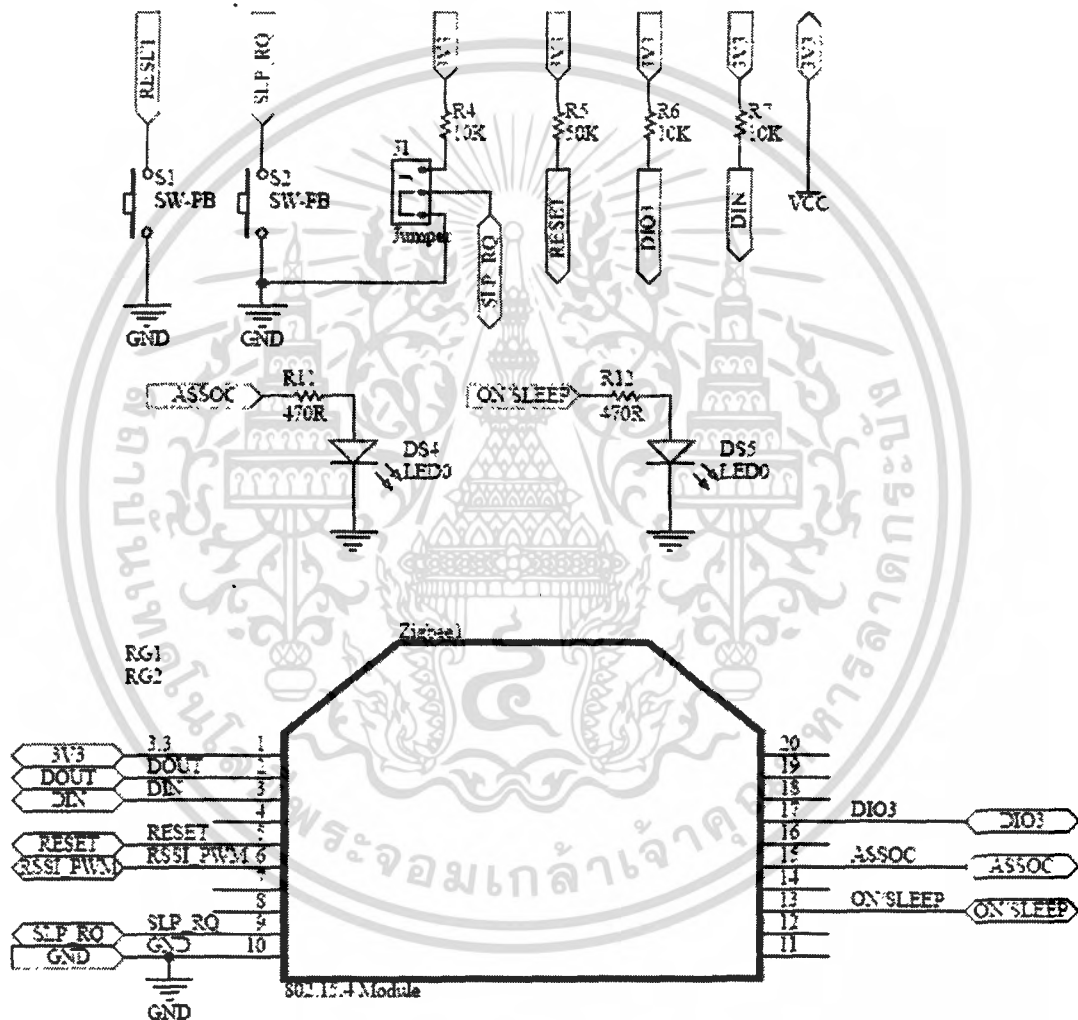


รูปที่ 3.5 วงจรไมโครคอนโทรลเลอร์และพอร์ต ICD2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

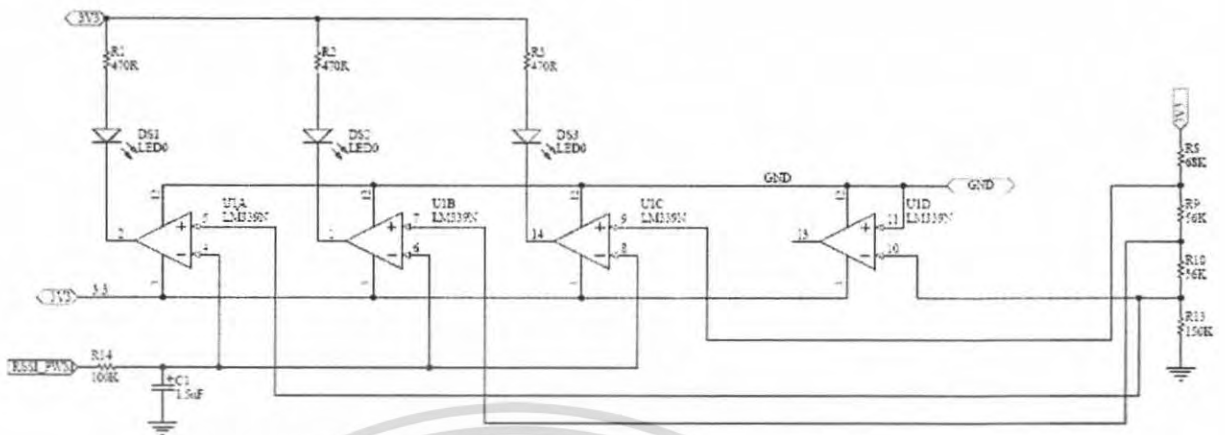
3.1.2 บอร์ด ZigBee

โมดูล ZigBee ต้องการแรงดันขนาด 3.3 โวลต์เช่นเดียวกับกับ MCU ดังนั้นจึงใช้ภาคจ่ายกระแสตรงเดียวกันได้โดยใช้ LM1117T3.3 สำหรับการต่อขาของจะ ใช้การเชื่อมต่อส่งข้อมูลทางพอร์ตอนุกรมเช่นเดียวกับกับ MCU หรือจะต่อตรงเข้าที่ขา DI และ DO ของไมโครคอนโทรลเลอร์เลยก็ได้ สำหรับ LED ASSOC (LED3) จะต่อกับขา 15 ของ Module Zigbee ซึ่งจะกระพริบทุกๆ 1 วินาที เพื่อบ่งบอกถึงสภาวะการทำงานของ โมดูล ZigBee



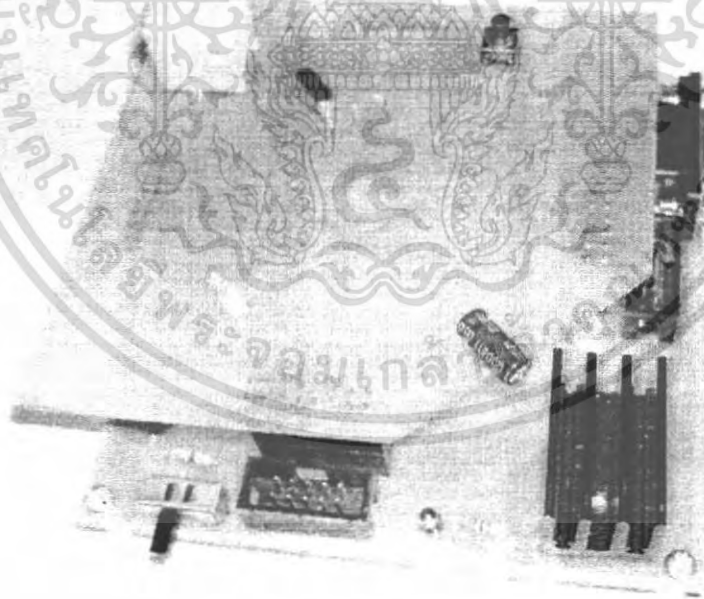
รูปที่ 3.6 ขาสัญญาณ ZigBee

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



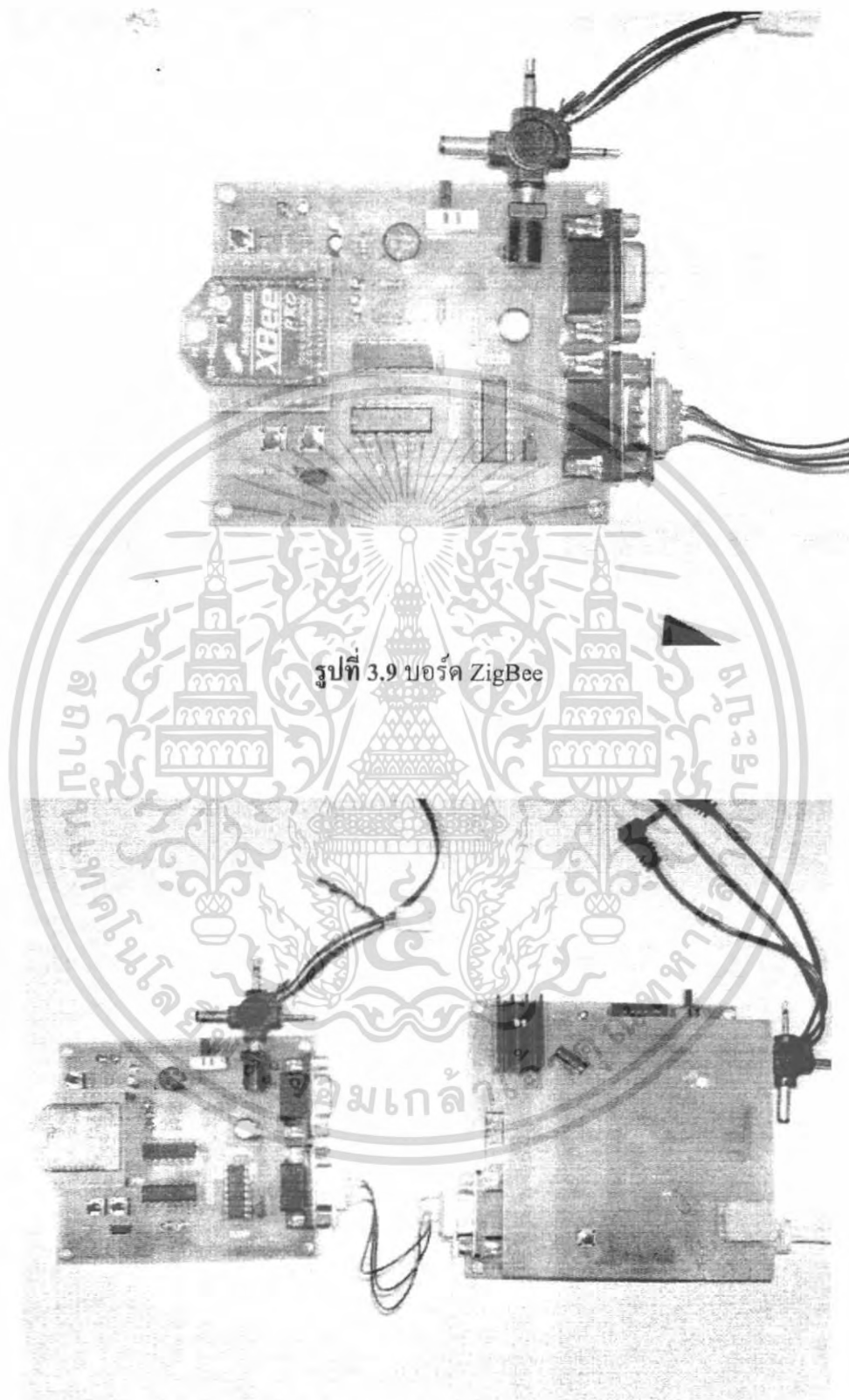
รูปที่ 3.7 วงจรแสดงระดับสัญญาณ ZigBee

โดยวงจรต่างๆ ได้ออกมาเป็นรูปร่างบอร์ดดังภาพ



รูปที่ 3.8 บอร์ดไมโครคอนโทรลเลอร์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

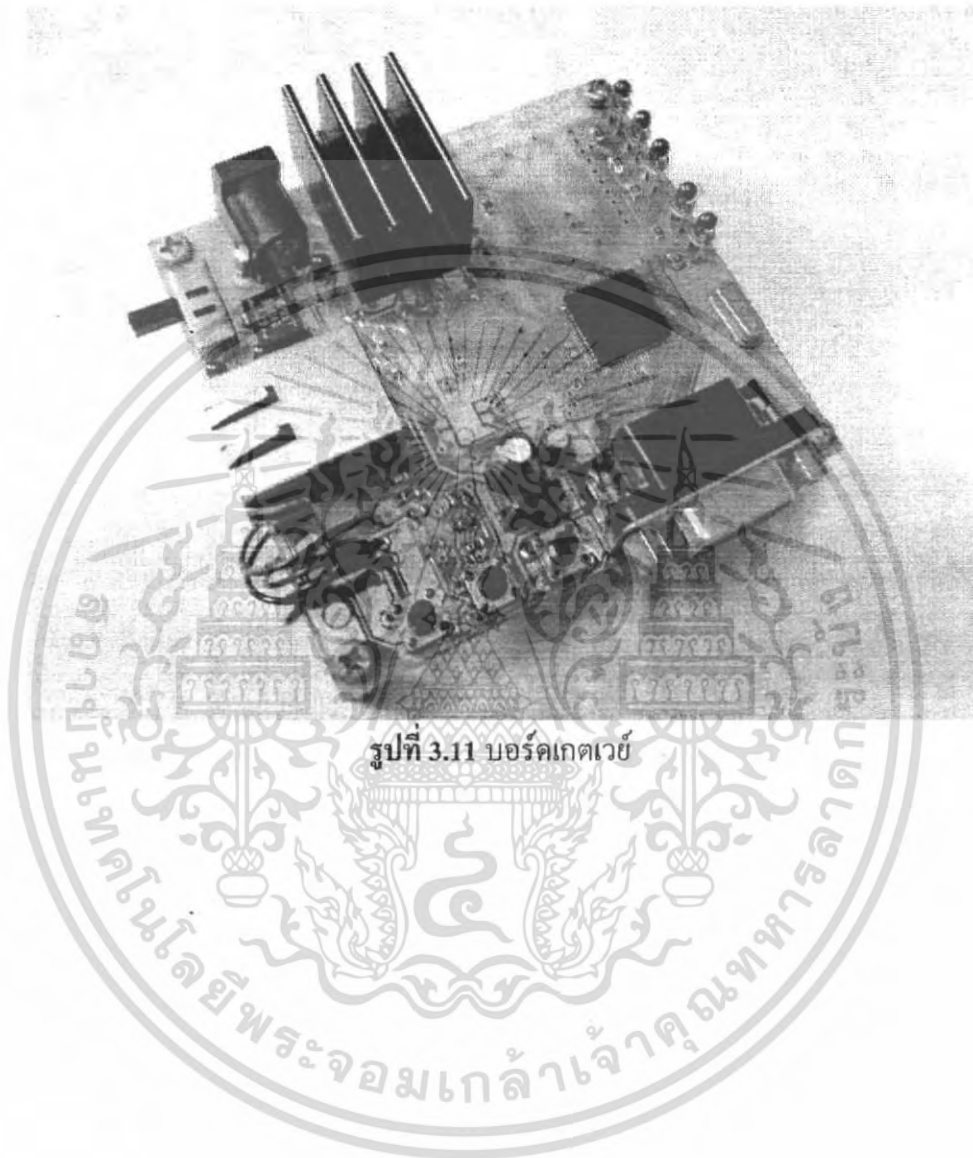


รูปที่ 3.9 บอร์ด ZigBee

รูปที่ 3.10 บอร์ดไมโครคอนโทรลเลอร์ต่อกับบอร์ด ZigBee

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

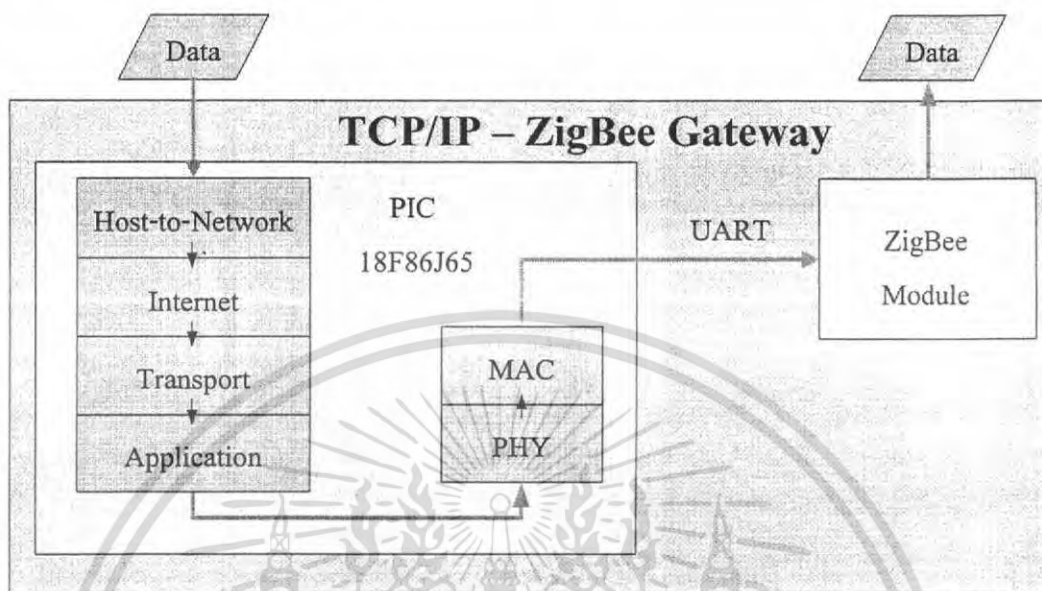
จากนั้นจึงได้พัฒนาให้มีไมโครคอนโทรลเลอร์และ ZigBee รวมกันในบอร์ดเดียวและมีขนาดเล็กลง



รูปที่ 3.11 บอร์ดเกตเวย์

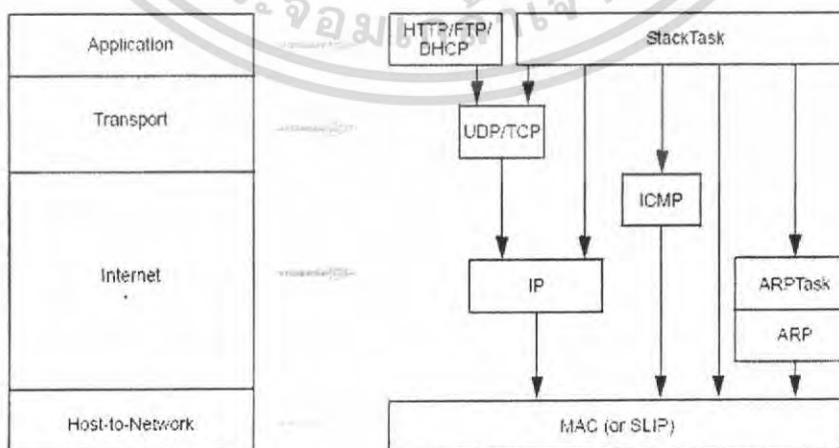
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 โครงสร้างของระบบ



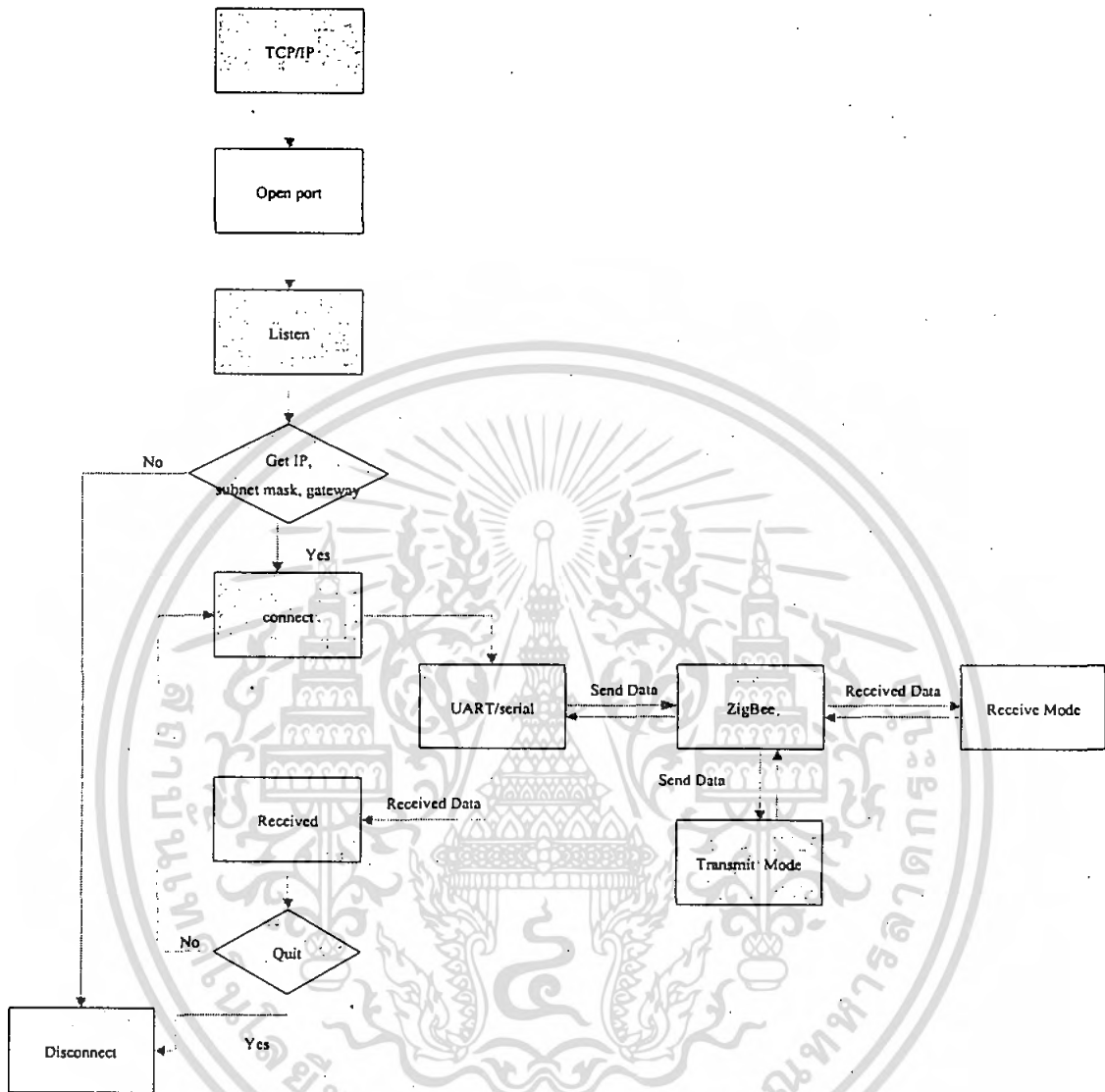
รูปที่ 3.12 โครงสร้างของระบบการเชื่อมต่อโปรโตคอล TCP/IP - ZigBee เกตเวย์

โครงสร้างของระบบการเชื่อมต่อโปรโตคอล TCP/IP - ZigBee เกตเวย์ แบ่งการทำงานเป็นสองส่วน คือ ส่วนของไมโครคอนโทรลเลอร์ (PIC18F86J65) ใช้ในการควบคุมการทำงานของสแตค TCP/IP และรับส่งผ่านข้อมูลผ่านโปรโตคอล UART ติดต่อกับพอร์ตอนุกรม เพื่อใช้ในการสื่อสารไปยังโมดูล ZigBee หรือ ใช้โปรโตคอล UART ในการติดต่อโมดูล ZigBee โดยตรง ซึ่งเราแสดงเป็นฟังก์ชันของรายละเอียดการทำงานของไมโครคอนโทรลเลอร์ (PIC18F86J65) ที่ใช้ควบคุมทำงาน สแตค TCP/IP และการทำงานของโมดูล ZigBee



รูปที่ 3.13 การทำงานของโปรโตคอลย่อยในสแตคของ TCP/IP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.14 ภาพรวมฟังก์ชันการทำงานของระบบการเชื่อมต่อโปรโตคอล TCP/IP-ZigBee เกตเวย์

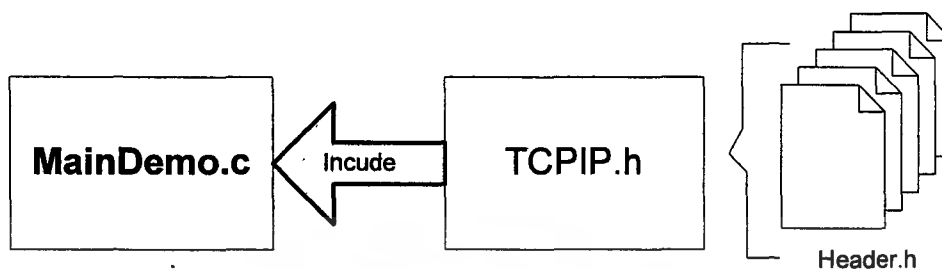
ขั้นตอนการทำงานของสแตก TCP/IP รับส่งข้อมูล โดยแบ่งเป็น โปรโตคอลย่อยดังนี้

1. เมื่อเราส่งข้อมูลจากชั้นแอปพลิเคชัน ในชั้นนี้จะมีโปรโตคอลย่อย คือ HTTP, FTP, DHCP ซึ่งการทำงานที่แตกต่างกัน โดยถ้าเราต้องการใช้โปรโตคอลใด ต้องนิยามการทำงานของโปรโตคอลนั้น เมื่อโปรโตคอลที่เรานิยามการใช้งานได้รับไฟล์ข้อมูลจะเพิ่มหัวไฟล์ไปหน้าข้อมูลแล้วทำเป็นเฟรมข้อมูลส่งไปยังชั้นทรานสปอร์ตต่อไป
2. ชั้นทรานสปอร์ตในชั้นนี้มีโปรโตคอลสองตัวซึ่งมีการทำงานที่แตกต่างกัน คือ TCP และ UDP ซึ่ง TCP จะรับโปรโตคอล HTTP และ FTP จากชั้นแอปพลิเคชันและ UDP รับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- โปรโตคอล DHCP จากชั้นแอปพลิเคชัน จากนั้นจะทำเฟรมใหญ่ในเป็นซอกเก็ตย่อยโดยเพิ่มหัวไฟล์เข้าไปหน้าซอกเก็ตเพิ่มระบุว่าเป็นของเฟรมใด และส่งไปยังชั้นอินเทอร์เน็ต
3. ชั้นอินเทอร์เน็ตเมื่อได้รับซอกเก็ตจากชั้นทรานสปอร์ตแล้ว ทำซอกเก็ตให้เป็นแพ็กเก็ตย่อยๆ ในชั้นนี้จะมีโปรโตคอล ICMP และ IP โดยโปรโตคอล ICMP ใช้ในการปิง หมายเลข IP ที่โปรโตคอล IP ระบุแล้วไปยังปลายทาง และโปรโตคอล IP ทำการระบุ หมายเลข IP ทางและปลายทางที่ต้องการส่ง โดยการเพิ่มหัวไฟล์เข้าไปที่แพ็กเก็ตและส่งไปยังชั้นโฮสต์-เครือข่าย
 4. ชั้นโฮสต์-เครือข่าย เมื่อได้รับแพ็กเก็ตจากชั้นอินเทอร์เน็ตแล้ว ทำแพ็กเก็ตให้เป็นข้อมูลไบต์ย่อยๆ ในชั้นนี้มีโปรโตคอล MAC และ SLIP เราต้องเลือกใช้โปรโตคอลใดโปรโตคอลหนึ่ง ไม่สามารถใช้งานพร้อมกันในเวลาเดียวโดยโปรโตคอล MAC จะรับข้อมูลจากชั้นอินเทอร์เน็ตมาระบุหมายเลข MAC ของต้นทางไปยังปลายทาง และโปรโตคอล SLIP ใช้ในการเชื่อมต่อออกพอร์ตอนุกรม
 5. ลำดับการทำงานนั้นเราใช้งานสแตค เข้าจัดการให้การทำงานของสแตค TCP/IP เป็นไปตามลำดับขั้นตอน และใช้งาน ARP (ARPTask) ในการแบ่งการปฏิบัติการเป็นสองลักษณะคือ เซิร์ฟเวอร์ และผู้ใช้ ซึ่งสามารถเรียกใช้งานจากโปรโตคอล MAC
 6. เมื่อเราต้องการส่งข้อมูลจากโปรโตคอล TCP/IP ไปยัง โปรโตคอล ZigBee เราสามารถทำได้สองวิธี คือ การเชื่อมต่อออกพอร์ตอนุกรมเพื่อรับส่งข้อมูลไปยัง โมดูล ZigBee หรือ การใช้โปรโตคอล UART ในการจัดการรับส่งข้อมูลจากโปรโตคอล TCP/IP ไปยังโมดูล ZigBee ซึ่งมีการจัดการสแตคแบบเข้าก่อนออกก่อน (FIFO) ก่อนที่จะรับส่งข้อมูลออกไปยัง โมดูล ZigBee

3.2.1 การออกแบบฟังก์ชันการทำงานของโปรแกรมควบคุมตัวควบคุม



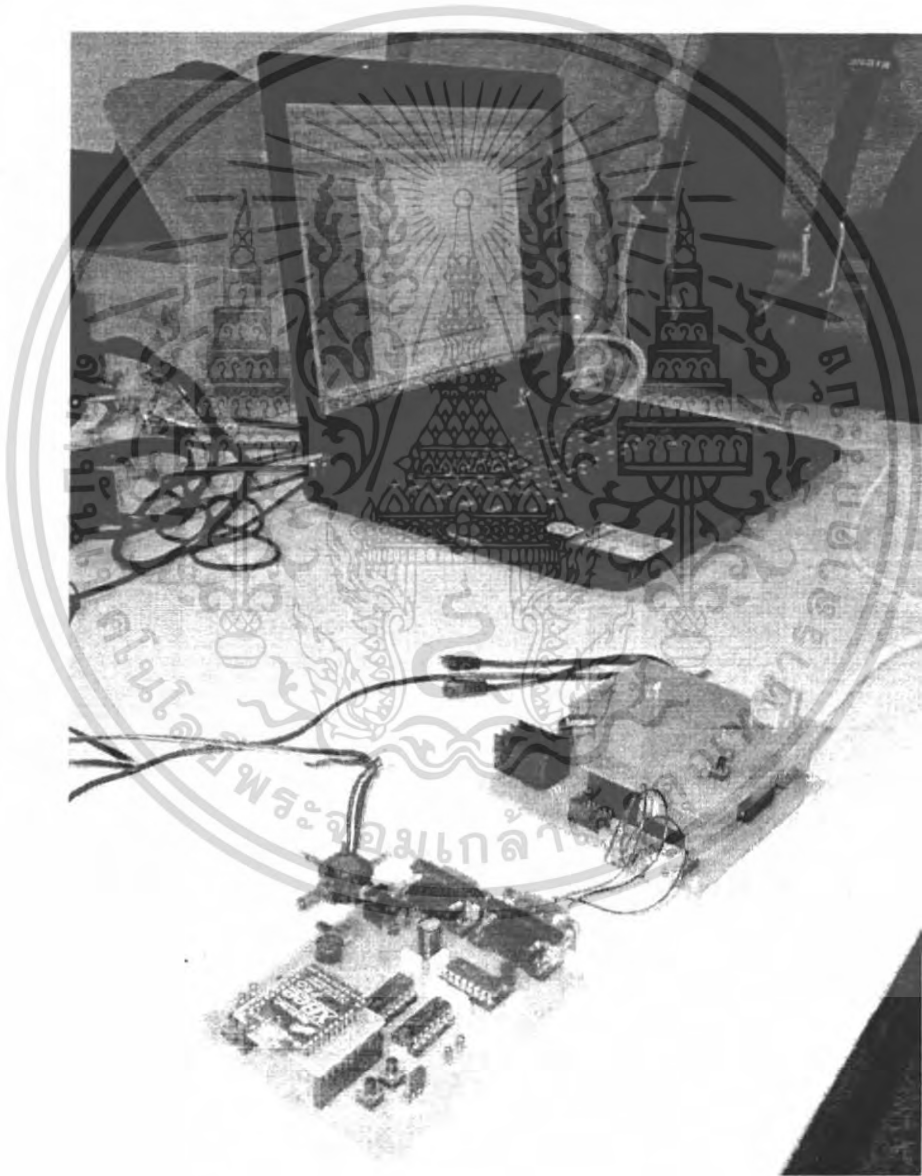
รูปที่ 3.15 การออกแบบฟังก์ชันการทำงานของโปรแกรมควบคุมตัวควบคุม

จากบทที่ 2 ได้แสดงการทำงานของสแตค TCP/IP หลักการพื้นฐานของโปรโตคอล ZigBee และความรู้พื้นฐานของตัวควบคุมไมโครคอนโทรลเลอร์ (PIC18F86J65) เราทำการออกแบบโปรแกรมควบคุมตัวควบคุมไมโครคอนโทรลเลอร์ โดยใช้ภาษาซีในการเขียนควบคุมการทำงานของตัวควบคุมไมโครคอนโทรลเลอร์ เราออกแบบให้สร้างไฟล์หลักขึ้นมาหนึ่งไฟล์ (MainDemo.c) เพื่อนิยามฟังก์ชันการทำงานในการบันทึกแอฟพลิเคชัน จัดการทำงานของแอฟพลิเคชัน และใช้รหัสคำสั่งในการนิยามเรียกใช้การทำงานของโปรโตคอลต่างๆ เช่น ถ้านิยามการใช้งานเซิร์ฟเวอร์ HTTP ใช้รหัสคำสั่งนิยามเรียกไฟล์ HTTP.c หรือ ถ้านิยามการใช้งานเซิร์ฟเวอร์ FTP ใช้รหัสคำสั่งนิยามเรียกไฟล์ FTP.c และจะต้องนิยามเรียกไฟล์ TCP.c เพราะเซิร์ฟเวอร์ HTTP และ FTP ทำงานโดยใช้โปรโตคอล TCP ในการส่งผ่านข้อมูลในชั้นทรานสปอร์ต และโปรโตคอล TCP ต้องทำงานในการส่งข้อมูลในชั้นอินเทอร์เน็ตโดยทำการส่งผ่านโปรโตคอล IP ดังนั้นจึงต้องนิยาม IP.c และนิยาม MAC.c เพื่อเป็นการแปลงหมายเลขไอพีแอสเดสจากชั้นอินเทอร์เน็ตเป็นหมายเลขของอุปกรณ์ที่ชั้นโฮสต์-เครือข่าย ดังนั้นเพื่อจะส่งผ่านข้อมูลออกไปยังโปรโตคอล ZigBee เราออกแบบให้ใช้ UART.c ในการรับและส่งข้อมูลออกไปยังพอร์ตอนุกรม ซึ่งโปรโตคอล ZigBee สามารถรับและส่งข้อมูลทางพอร์ตอนุกรมได้ จะเห็นได้ว่าการนิยามจำเป็นต้องนิยามการทำงานตั้งแต่ขั้นต่ำกว่าของโปรโตคอลที่เรานิยามการใช้งานลงมา และสร้างไฟล์ TCPIP.h ในการเก็บหัวไฟล์ย่อย เพื่อสะดวกในการเรียกใช้หัวไฟล์ตามโปรโตคอลที่เรานิยามการใช้งาน ซึ่งในแต่ละโปรโตคอลย่อยในสแตคการทำงาน TCP/IP มีฟังก์ชันการทำงานดังต่อไปนี้

บทที่ 4

การทดลองและผลการทดลอง

การทดลองนี้ได้จำลองการส่งข้อมูลผ่านคอมพิวเตอร์ 2 เครื่อง เครื่องแรกต่อกับบอร์ดไมโครคอนโทรลเลอร์กับ ZigBee ตัวที่ 1 และคอมพิวเตอร์อีกเครื่องต่อกับบอร์ด ZigBee ตัวที่ 2 โดยจะทำการส่งข้อมูลหากัน



รูปที่ 4.1 การทดลอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1 การรับส่งข้อมูลผ่านทางโปรโตคอล TCP/IP ไปยังพอร์ตคอมพิวเตอร์

4.1.1 การใช้โปรแกรม command DOS ทดสอบ ping หมายเลข IP

หลังจากเบิร์นโปรแกรมใส่ไมโครคอลลโทรเลอร์แล้ว ทำการทดลอง ping หมายเลข IP ผ่านทางโปรโตคอล ICMP แสดงผลการทดสอบโดยใช้โปรแกรม command DOS ผู้ใช้จะต้องตั้งค่าหมายเลข IP ให้อยู่ในวงเครือข่ายเดียวกันกับ หมายเลข IP ของไมโครคอลลโทรเลอร์ ที่มี

- หมายเลข IP 192.168.6.102
- ซับเน็ต มาร์ค 255.255.255.0



รูปที่ 4.2 การทดลองการ ping หมายเลข IP

```

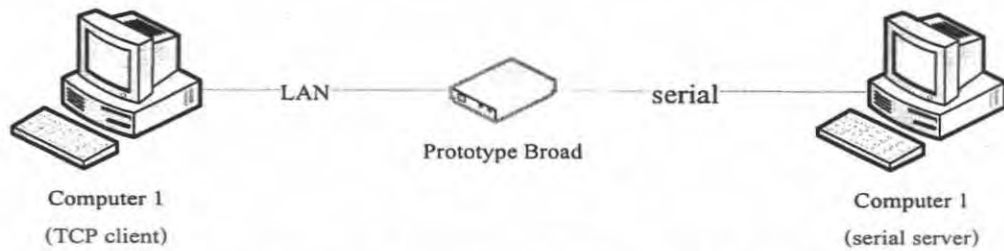
C:\WINDOWS\system32\CMD.exe
Microsoft Windows XP [Version 5.1.2600]
(C) Copyright 1985-2001 Microsoft Corp.
C:\Documents and Settings\Administrator>ping 192.168.6.102
Pinging 192.168.6.102 with 32 bytes of data:
Reply from 192.168.6.102: bytes=32 time=1ms TTL=100
Reply from 192.168.6.102: bytes=32 time=1ms TTL=100
Reply from 192.168.6.102: bytes=32 time=1ms TTL=100
Reply from 192.168.6.102: bytes=32 time=1ms TTL=100

Ping statistics for 192.168.6.102:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 1ms, Maximum = 1ms, Average = 1ms
C:\Documents and Settings\Administrator>
  
```

รูปที่ 4.3 โปรแกรม command DOS ที่ใช้ทดลองการ ping หมายเลข IP

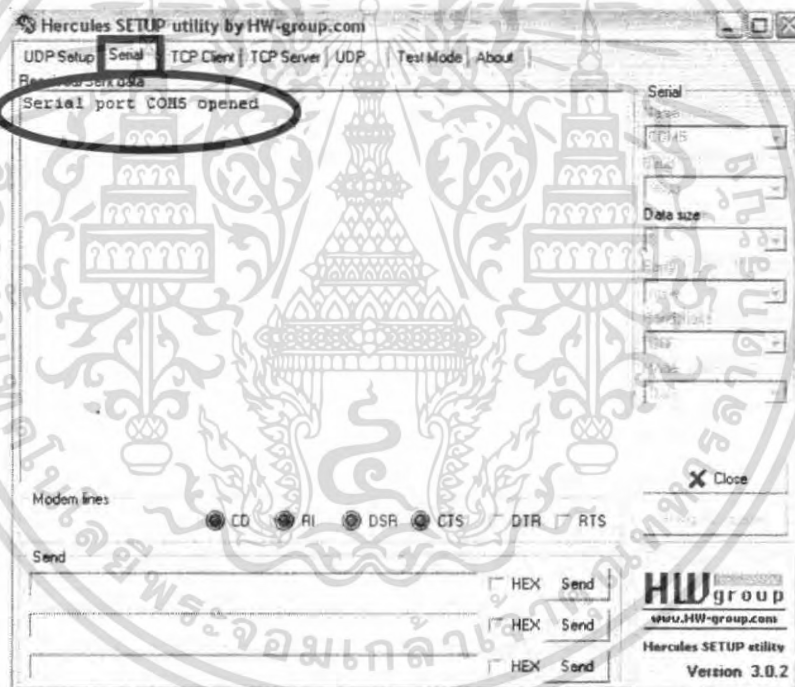
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.1.2 การใช้โปรแกรม Hercules ทดสอบการส่งค่าข้อมูลจาก TCP ไปยังพอร์ตอนุกรม



รูปที่ 4.4 การทดลองรับส่งค่าจากโปรโตคอล TCP ไปยังพอร์ตอนุกรม

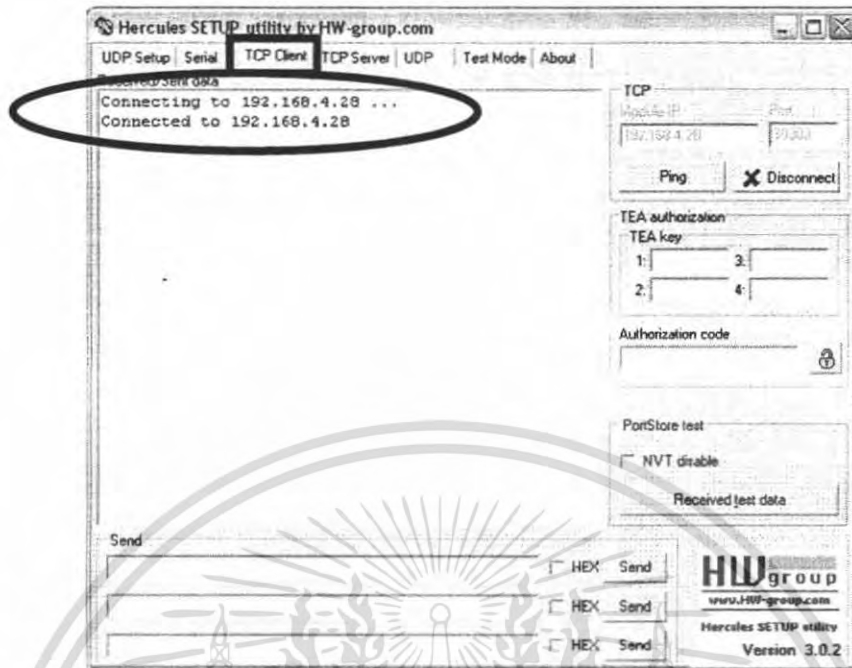
เริ่มจากการเปิดพอร์ตอนุกรมที่ฝั่งผู้รับ



รูปที่ 4.5 การเปิดพอร์ตอนุกรมที่ด้านผู้รับ

จากนั้นสร้างการเชื่อมต่อ TCP ที่ฝั่งผู้ใช้ และใส่หมายเลข IP 192.168.4.28 และพอร์ตที่เราติดต่อ แล้วกดปุ่ม Connect ในที่นี้เราได้ประกาศพอร์ต 30303 สำหรับใช้ติดต่อโปรโตคอล TCP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

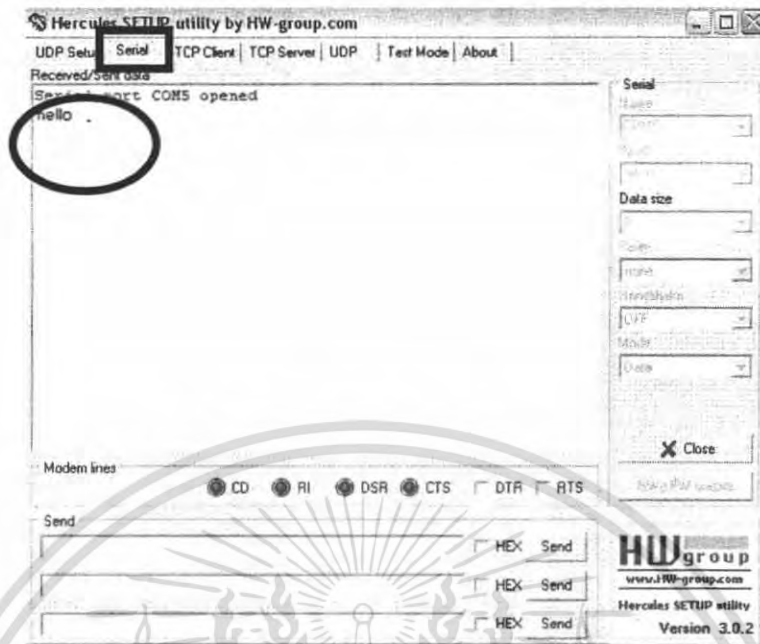


รูปที่ 4.6 การสร้างการเชื่อมต่อที่ด้านผู้ใช้
เมื่อสร้างการเชื่อมต่อได้แล้ว ทำการส่งข้อมูล "Hello" จากฝั่งผู้ใช้ ไปยังฝั่งผู้รับ



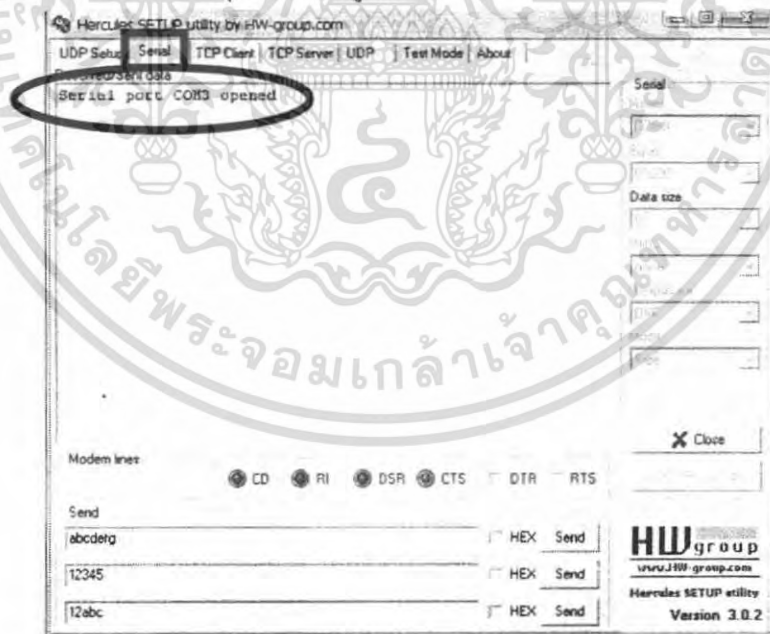
รูปที่ 4.7 การส่งข้อมูลที่ด้านผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.8 การรับข้อมูลที่ด้านผู้รับ

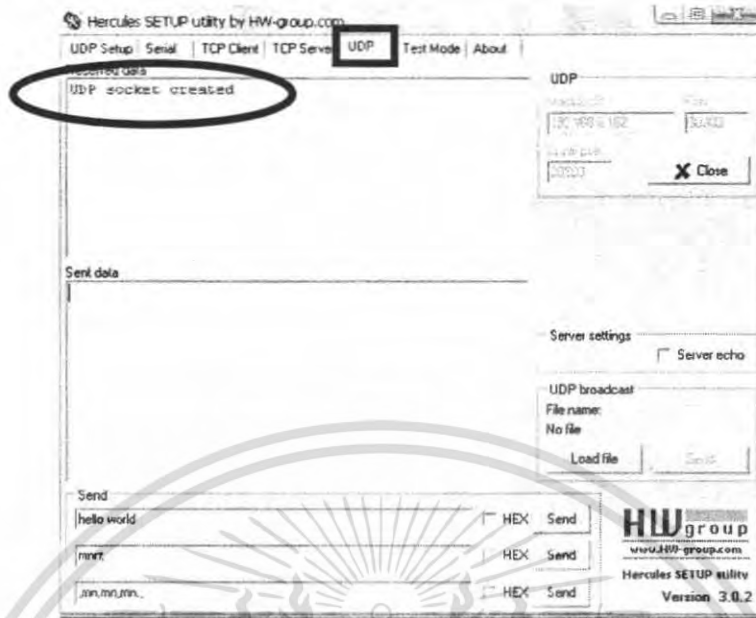
4.1.3 ใช้โปรแกรม Hercules ทดสอบการส่งค่าข้อมูลจาก UDP ไปยังพอร์ตอนุกรม เริ่มจากการเปิดพอร์ตอนุกรมที่ฝ่ายผู้รับ



รูปที่ 4.9 การเปิดพอร์ตอนุกรมด้านผู้รับ

จากนั้น ทำการเปิด UDP ที่ฝั่งผู้รับ และใส่หมายเลข IP 192.168.6.102 และพอร์ตที่เราติดต่อกับและโลกคอลพอร์ต แล้วกดปุ่ม Connect ในที่นี้เราได้ประกาศพอร์ต 30303 สำหรับใช้ติดต่อกับโปรโตคอล UDP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้าไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.10 การเปิดพอร์ต UDP ที่ด้านผู้ใช้

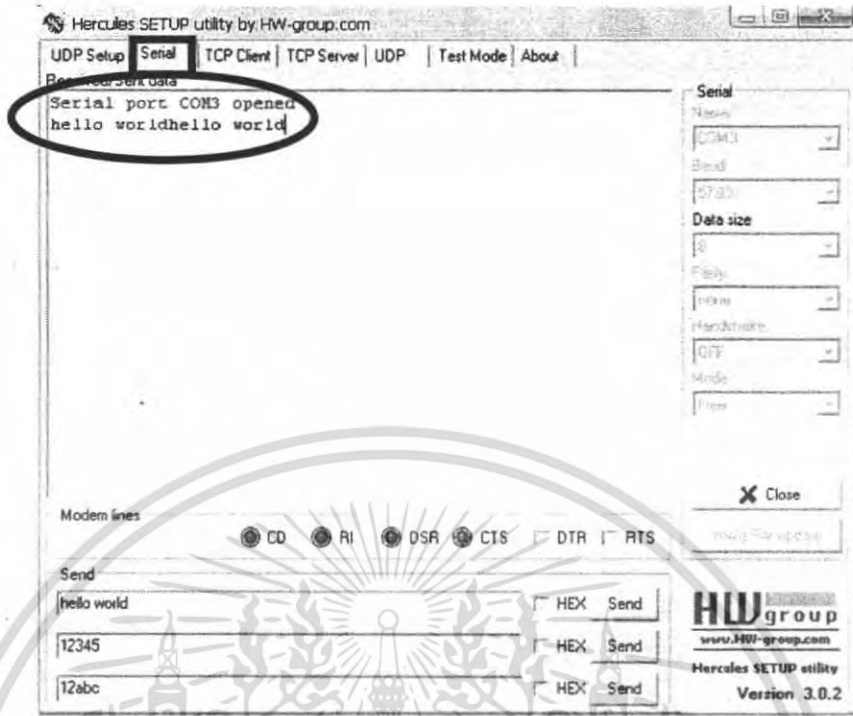
เมื่อทำการเปิดพอร์ตได้สำเร็จแล้ว ทำการทดลองรับส่งค่าจาก UDP ฟังผู้ใช้ ไปยังพอร์ตอนุกรมที่ฝั่งผู้รับ ทำการทดลองส่งค่า “hello world”



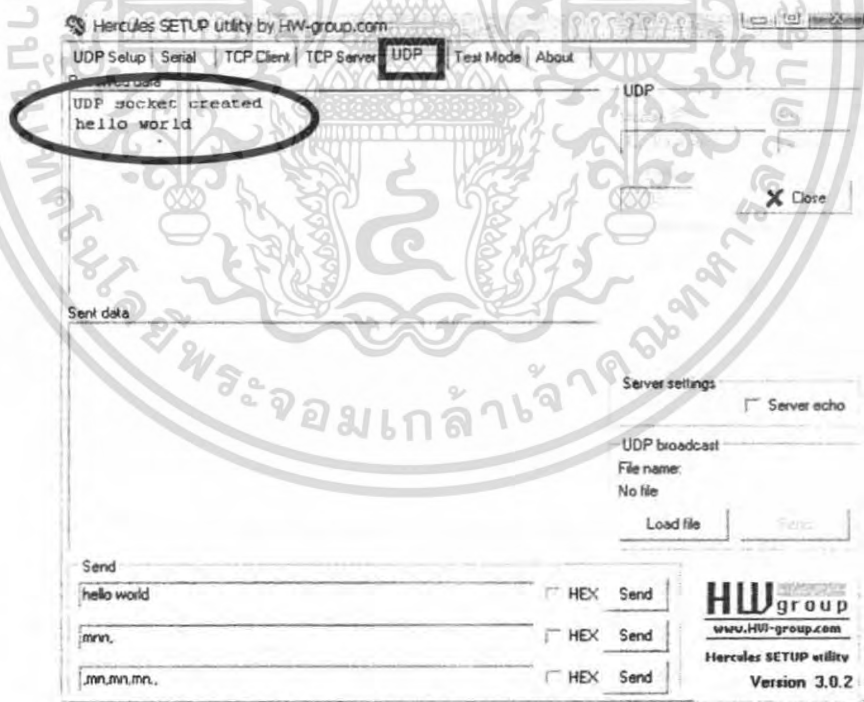
รูปที่ 4.11 การส่งข้อมูลที่ด้านผู้ใช้

ทำการเคลียร์หน้าต่าง(clear windows) และทำทดลองส่งค่า “hello world” จากฝั่งรับที่เชื่อมต่อกับพอร์ตอนุกรม ไปยังผู้ส่ง โดยผ่าน โปรโตคอล UDP

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 4.12 การส่งข้อมูลที่ด้านผู้รับ



รูปที่ 4.13 การรับข้อมูลที่ด้านผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การรับส่งข้อมูลผ่านทางโปรโตคอล TCP/IP ไปยังโมดูล ZigBee



รูปที่ 4.14 การทดลองการรับส่งข้อมูลผ่านทางโปรโตคอล TCP/IP ไปยังโมดูล ZigBee

เราเลือกใช้โปรโตคอล UDP ในการรับส่งข้อมูลจากโปรโตคอล TCP/IP ไปยังโมดูล ZigBee .
 ปลายทาง เพราะโปรโตคอล UDP มีข้อดีที่ไม่ต้องสร้างการเชื่อมต่อก่อนการใช้งาน ทำให้การรับส่ง
 ข้อมูลเป็นไปอย่างรวดเร็วและเป็นลดภาระในการประมวลผลข้อมูลของไมโครคอนโทรลเลอร์
 ขั้นแรกต้องทำการกำหนดหมายเลขที่อยู่ ZigBee แต่ละตัวจะส่งหากันโดยใช้โปรแกรม X-
 CTU มีคำสั่งพื้นฐาน 5 คำสั่ง

+++ เป็นการเปิดคอมมานด์โหมดสำหรับการปรับค่าต่างๆของ ZigBee

atsl : แสดงค่า Source Address Low

atsh : แสดงค่า Source Address High

โดยค่า atsl และ atsh ไม่สามารถปรับเปลี่ยนค่าได้

atdl : แสดงค่า Destination Address Low

atdh : แสดงค่า Destination Address High

atnd : แสดงค่า Node ใกล้เคียงที่ ZigBee ตรวจจับได้ โดยมีข้อมูลที่สำคัญคือค่า atsl และ
 atsh ของ ZigBee ที่ตรวจจับ

สำหรับ ZigBee ตัวที่ 1 จะมีค่า

atsl = 4008516D , atsh = 13A200

และตรวจจับ ZigBee ตัวที่ 2 ซึ่งจะได้ค่าดังนี้

atsl = 4008516E , atsh = 13A200

เมื่อได้ข้อมูลเบื้องต้นแล้วจึงทำการตั้งค่า atdl และ atdh ของ ZigBee ตัวที่ 1 ให้มีหมายเลข
 ปลายทางเป็นของ ZigBee ตัวที่ 2 จะได้ค่าทั้งหมดของ ZigBee ตัวที่ 1 ดังภาพ

```

X-CSR (COM1)
PC Settings | Range Test | Terminal | Modem Configuration
Line Status: CTS | CD | DSR
Assert: DTR [x] RTS [x] Break [ ]
Close Com Port Assemble Packet Clear Screen Hide Hex
+++OK
atdh 2B 2B 2B 4F 4B 0D
13A200 61 74 64 68 0D
atdl 31 33 41 32 30 30 0D
4008516E 61 74 64 6C 0D
atsh 34 30 30 38 35 31 36 45 0D
13A200 61 74 73 68 0D
atsl 31 33 41 32 30 30 0D
4008516D 61 74 73 6C 0D
atnd 34 30 30 38 35 31 36 44 0D
0 61 74 6E 64 0D
13A200 30 0D
4008516E 31 33 41 32 30 30 0D
27 34 30 30 38 35 31 36 45 0D
32 37 0D
20 0D
0D
0D
COM1 9600 8-N-1 FLOW/NONE Rx: 60 bytes

```

รูปที่ 4.15 ค่าหมายเลขที่อยู่ของ ZigBee ตัวที่ 1

เมื่อปรับตั้งค่า ZigBee ตัวที่ 1 แล้ว ZigBee ตัวที่ 2 ก็ใช้วิธีเดียวกันคือ ตั้งให้ค่า atdl และ atdh ตรงกับค่า atsl และ atsh ของ ZigBee ตัวที่ 1 จะ ได้ค่าทั้งหมดของ ZigBee ตัวที่ 2 ดังภาพ

```

X-CSR (COM1)
PC Settings | Range Test | Terminal | Modem Configuration
Line Status: CTS | CD | DSR
Assert: DTR [x] RTS [x] Break [ ]
Close Com Port Assemble Packet Clear Screen Hide Hex
+++OK
atdh 2B 2B 2B 4F 4B 0D
13A200 61 74 64 68 0D
atdl 31 33 41 32 30 30 0D
4008516D 61 74 64 6C 0D
atsh 34 30 30 38 35 31 36 44 0D
13A200 61 74 73 68 0D
atsl 31 33 41 32 30 30 0D
4008516E 61 74 73 6C 0D
atnd 34 30 30 38 35 31 36 45 0D
0 61 74 6E 64 0D
13A200 30 0D
4008516D 31 33 41 32 30 30 0D
29 34 30 30 38 35 31 36 44 0D
32 39 0D
20 0D
0D
0D
COM1 9600 8-N-1 FLOW/NONE Rx: 60 bytes

```

รูปที่ 4.16 ค่าหมายเลขที่อยู่ของ ZigBee ตัวที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อตั้งค่า ZigBee ทั้งสองตัวเรียบร้อยแล้วก็จะสามารถติดต่อกันได้ โดยแสดงดังภาพ



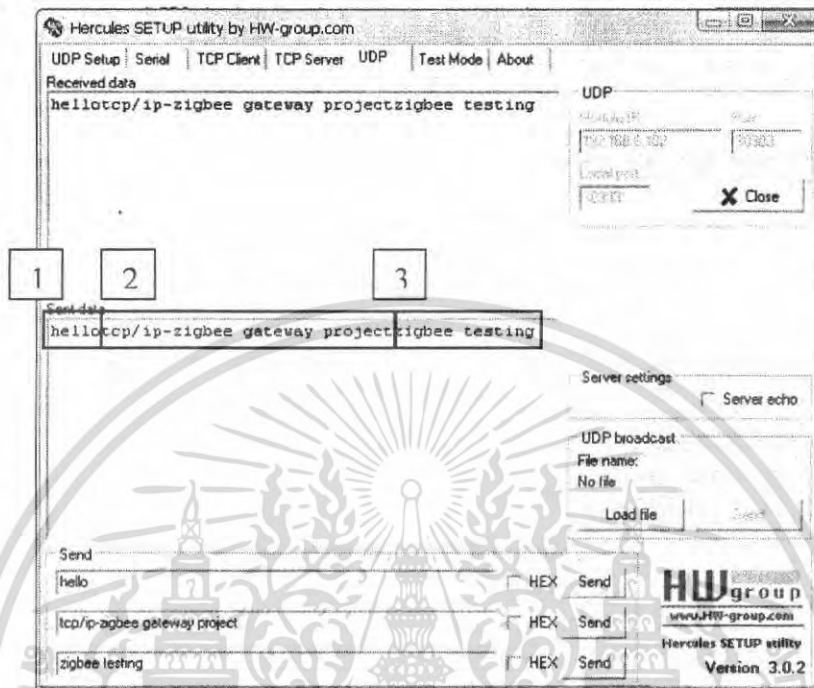
รูปที่ 4.17 การติดต่อของ ZigBee ตัวที่ 1



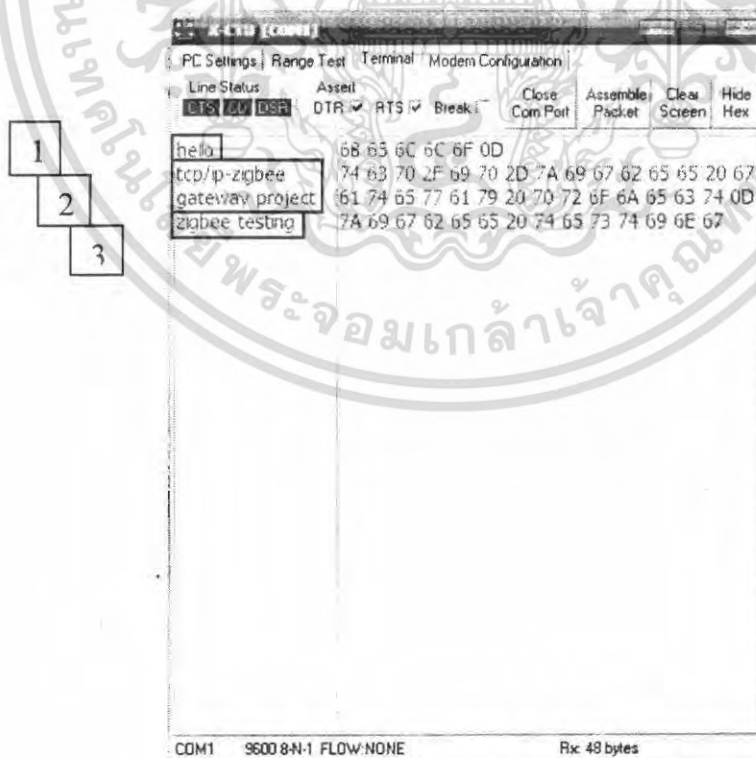
รูปที่ 4.18 การติดต่อของ ZigBee ตัวที่ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หลังจากทดสอบการติดต่อของ ZigBee เรียบร้อยแล้ว จะทำการส่งข้อมูลจาก UDP ผ่าน ZigBee ตัวที่ 1 และ ZigBee ตัวที่ 2 จะต่อกับคอมพิวเตอร์อีกเครื่องหนึ่ง โดยได้ผลการทดลองดังภาพ



รูปที่ 4.18 การส่งข้อมูลจากคอมพิวเตอร์เครื่องส่ง



รูปที่ 4.19 การรับข้อมูลที่คอมพิวเตอร์ฝั่งรับ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

บทสรุปและวิจารณ์

5.1 สรุปผลการทดลอง

1. สามารถทำการออกแบบบอร์ดไมโครคอนโทรลเลอร์และบอร์ดโมดูล ZigBee เพื่อใช้ในการสร้างการเชื่อมต่อระหว่างโปรโตคอล TCP/IP ไปยังโปรโตคอล ZigBee
2. สามารถสร้างการเชื่อมต่อโปรโตคอล TCP/IP ไปยังโมดูล ZigBee โดยใช้ไมโครคอนโทรลเลอร์เป็นตัวควบคุมการจัดการ โปรโตคอลทั้งสองให้ติดต่อสื่อสารกันได้

5.2 ปัญหาที่พบในระหว่างการดำเนินโครงการ

1. ความรู้ของผู้ทำโครงการมีไม่พอจึงทำให้เสียเวลาในการศึกษา รวบรวมข้อมูลใช้เวลามาก
2. เนื่องจากข้อแรกทำให้การเขียน โปรแกรมและออกแบบบอร์ดทดลอง เกิดข้อผิดพลาดขึ้น ส่งผลให้เกิดความล่าช้าในการดำเนิน โครงการ ทำให้ใช้เวลานานขึ้น
3. เนื่องจากอุปกรณ์ไมโครคอนโทรลเลอร์และโมดูล ZigBee หาได้ยากในประเทศทำให้ต้องสั่งซื้อผ่านเว็บจากต่างประเทศ อุปกรณ์จึงมีราคาสูงและล่าช้า

5.3 แนวทางแก้ไขและพัฒนา

1. สามารถนำไปประยุกต์ใช้กับการควบคุมอุปกรณ์ที่ใช้โมดูล ZigBee ในระยะไกล โดยผ่านทางอินเทอร์เน็ต หรือผ่านทางสแตก TCP/IP
2. สามารถพัฒนาแอปพลิเคชันมาสนับสนุนการใช้งานเครือข่าย ZigBee ให้สะดวกและง่ายสำหรับผู้ใช้งานมากขึ้น
3. สามารถใช้งานโมดูล ZigBee ได้หลากหลาย เช่น สามารถนำไปใช้ในการควบคุมอุปกรณ์ภายในบ้านอัจฉริยะ (Home Automation)

บรรณานุกรม

- [1] IEEE 802.15.4 Standard Specification, downloadable at standards.ieee.org/getieee802/.
- [2] ZIGBEE technical documents at <http://www.zigbee.org/>.
- [3] CC2420 Datasheet, downloadable at <http://www.chipcon.com/>.
- [4] Stack TCP 3.75 at <http://www.microchip.com/>
- [5] PIC18F86J65 family at <http://www.microchip.com/>



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

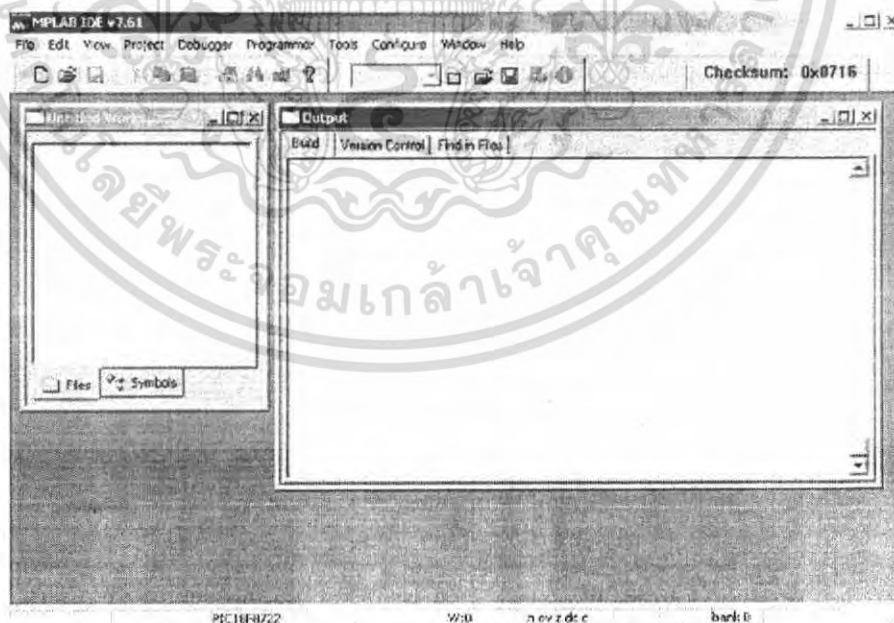
ภาคผนวก

วิธีการติดตั้งและใช้งานโปรแกรม MPLAB

1. สามารถลงได้จากแผ่นซีดี โดยเลือก menu.exe หรือสามารถดาวน์โหลดได้จาก www.microchip.com
2. ต้องทำการติดตั้งโปรแกรม MPLAB IDE 7.6 ขึ้นไป เพื่อที่จะสนับสนุนโปรแกรมภาษาซีได้
3. หลังจากติดตั้งเสร็จให้ทำการติดตั้งไฟล์ MCC18 ในไดรฟ์ที่อยู่เดียวกันกับที่ลงโปรแกรม MPLAB IDE vx.xx
4. ทำการติดตั้งโปรแกรม MPLAB MCC 18 เพื่อให้สามารถคอมไพล์โปรแกรมภาษาซีได้

การใช้งานโปรแกรม MPLAB

1. เมื่อทำการเปิดโปรแกรม MPLAB IDE vx.xx โดยการดับเบิลคลิกที่ไอคอนที่อยู่บน desktop หรือเลือกจาก Start>Programs>Microchip>MPLAB IDE vx.xx>MPLAB IDE จะแสดงเป็นหน้าจอดังภาพ

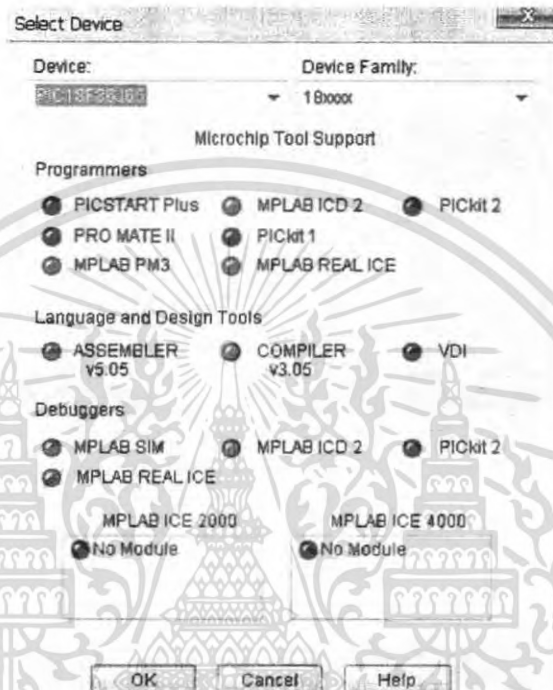


รูปที่ 1 หน้าต่างแสดงการทำงานของโปรแกรม MPLAB

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การเลือกใช้อุปกรณ์

1. ทำการเลือกที่ Configure>Select Device
2. ในช่องอุปกรณ์ ให้เลือก ชนิดอุปกรณ์ที่เราใช้ เช่นในกรณีนี้เลือกเป็น PIC18F86J65



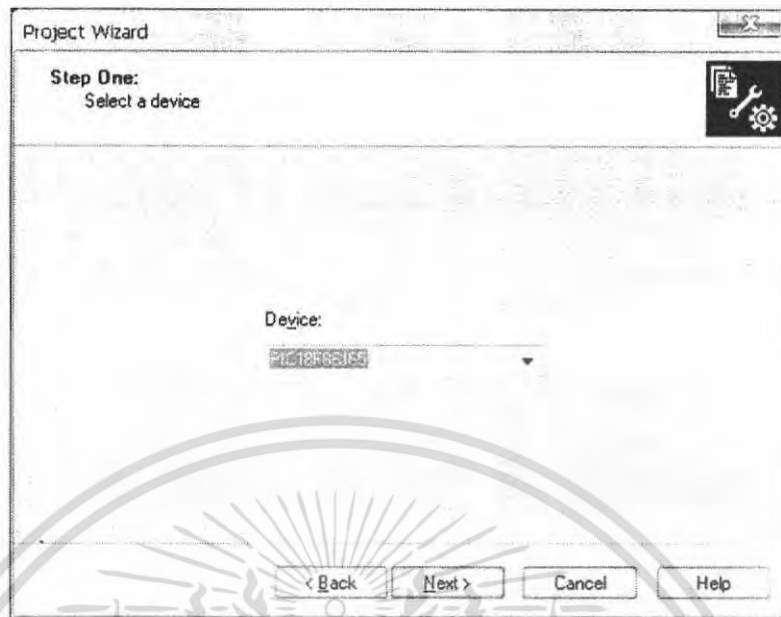
รูปที่ 2 หน้าต่างอุปกรณ์

3. สัญลักษณ์ ไฟสีเขียว หมายถึง อุปกรณ์ชนิดนี้ สามารถสนับสนุนการทำงานอะไรได้บ้างในโปรแกรมนี้
4. สัญลักษณ์ ไฟสีแดง หมายถึง อุปกรณ์ชนิดนี้ ไม่สามารถสนับสนุนการทำงานอะไรได้บ้างในโปรแกรมนี้

การสร้างโครงการ

1. เราสามารถใช้เครื่องมือ wizard โดยเลือกจาก Project>Project Wizard
2. จากนั้นทำการ Next แล้วเลือกอุปกรณ์ ตามชนิดที่ใช้ ในกรณีนี้เลือกเป็น PIC18F86J65

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3 หน้าต่าง Project Wizard 1

3. ทำการเลือกภาษาที่ใช้

MPASM assembler เป็น ไฟล์ที่อยู่ใน:

“C:\Program Files\Microchip\MPASM Suite\mpasmwin.exe”

the MPLINK linker เป็น ไฟล์ที่อยู่ใน:

“C:\Program Files\Microchip\MPASM Suite\mplink.exe”

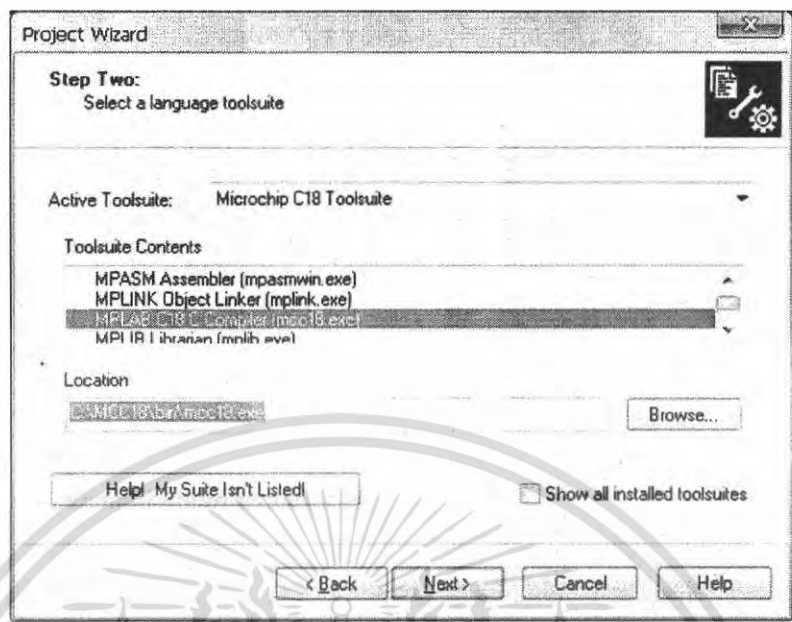
MPLAB C18 C Compiler เป็น ไฟล์ที่อยู่ใน:

“C:\MCC18\bin\Mcc18.exe”

และ the MPLIB librarian เป็น ไฟล์ที่อยู่ใน:

“C:\Program Files\Microchip\MPASM Suite\mplib.exe”

ในกรณีนี้เราใช้ภาษาซีเป็นภาษาในการเขียน โปรแกรมควบคุม จึงเลือก MCC18 เป็นภาษาที่ใช้ในการคอมไพล์

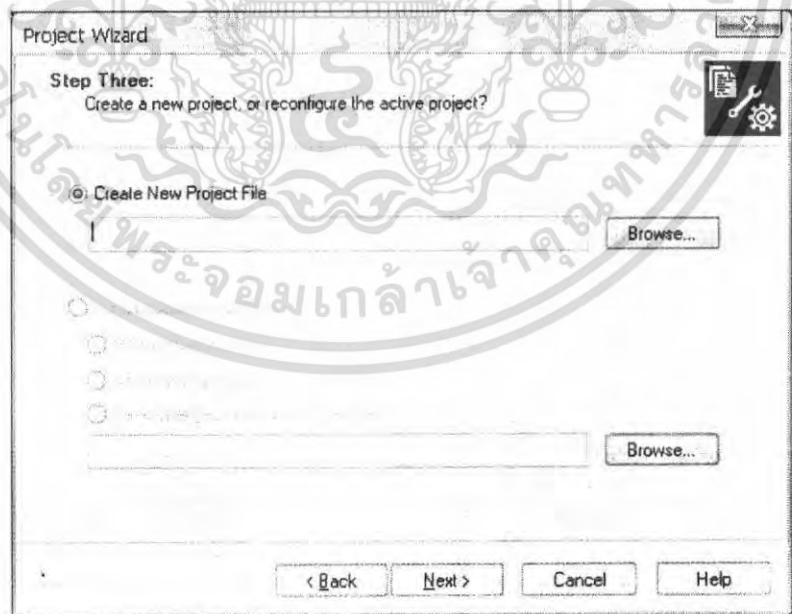


รูปที่ 4 หน้าต่าง Project Wizard 2

- 4. ทำการตั้งชื่อโครงการ หรือเลือกใช้โครงการที่มีอยู่แล้ว

ในกรณีเราเลือก tcpip.mcp สามารถดาวน์โหลดได้จาก www.microchip.com >stack tcp/ip

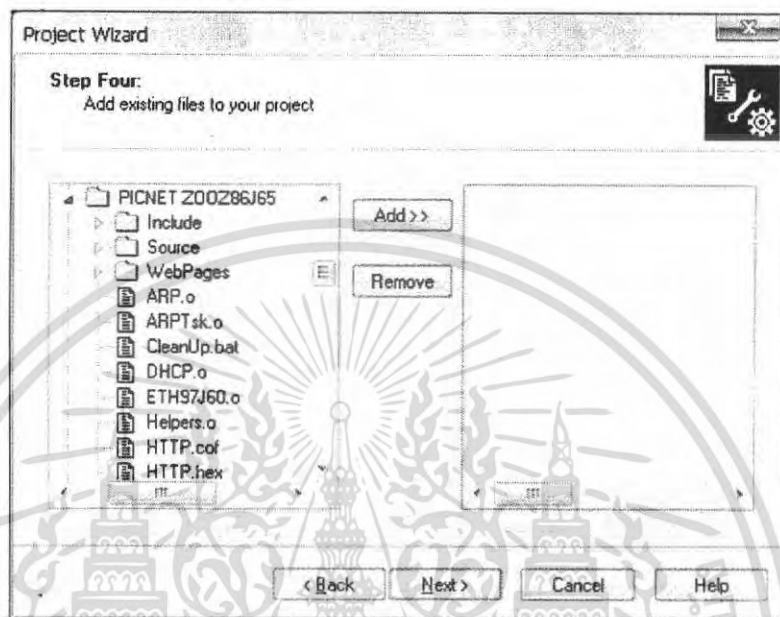
3.75



รูปที่ 5 หน้าต่าง Project Wizard 3

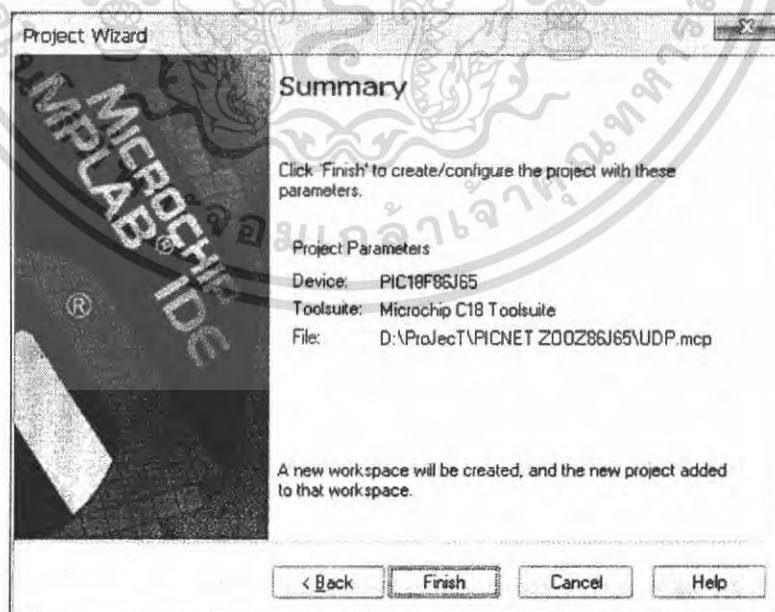
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5. จากนั้นทำการเพิ่มไฟล์เลือกที่ source, header และ lkr ที่ต้องการใช้เพิ่มเข้าไป เช่น ในกรณีสร้างโปรเจก udp.mcp ทำการเพิ่มไฟล์ของสแตคที่อยู่ในระดับชั้นที่ต่ำกว่าลงไป เมื่อทำการเพิ่มเสร็จแล้วทำการ Next



รูปที่ 6 หน้าต่าง Project Wizard 4

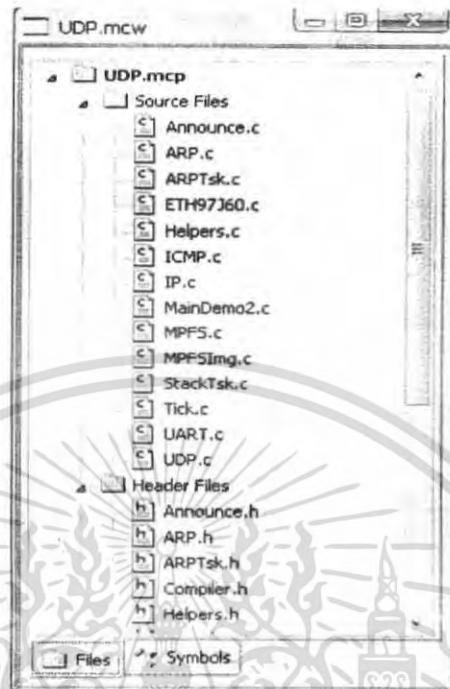
6. จะได้โครงการ workshop ที่เสร็จสมบูรณ์



รูปที่ 7 หน้าต่าง Project Wizard 5

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7. จะได้ผลลัพธ์ดังรูป



รูปที่ 8 หน้าต่างโครงการ

8. การแก้ไขโค้ดได้ เลือกไฟล์ที่ต้องการแก้ไข แล้วคลิกขวา เลือก edit โค้ดไฟล์ที่ต้องการเลือกจะปรากฏที่หน้าต่าง
9. เมื่อต้องการสร้างโครงการ เป็นไฟล์ .cof และ .hex เพื่อใช้ในการเบิร์นโปรแกรมลงในไมโครคอนโทรลเลอร์ โดยการเลือก project>build all หรือกดปุ่ม F10 ที่คีย์บอร์ด
10. สามารถเลือกการคอมไฟล์ในรูปแบบลักษณะต่างๆ ได้ โดยเลือกจาก project >project option
11. สามารถทำการแก้ไขการเข้ารหัสและคอนฟิกบิตได้ โดยเลือกจาก configure >configuration bits

Address	Value	Category	Setting
17FF8	10	Watchdog Timer Enable	Disabled
		Stack Overflow Reset Enable	Enabled
		Extended CPU Enable	Disabled
17FF9	04	Code Protect 000000-017FFF	Disabled
17FFA	C5	Oscillator	OSC1/OSC2 as primary, HS+PLL Osc
		Fail Safe Monitor Clock Enable	Enabled
		Internal External Switch Over Enable	Enabled
17FFB	0F	Watchdog Timer Postscale	1:32,768
17FFD	F7	CCP2 Mux	ECCP2 I/O Muxed With RC1
		ECCP Mux	P1B/P1C On RE6/RES, P3B/P3C On RE4/RES
		Ethernet LED Enable	LEDA/LEDB On RA0/PA1 With Ethernet, RA0/AN0 Without

รูปที่ 9 หน้าต่างการตั้งค่าบิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานโปรแกรม PICkit 2 v2.xx

1. ทำการติดตั้งโปรแกรม จากแผ่น โดยเลือกจาก install.exe หรือดาวน์โหลดมาจาก www.microchip.com หรือ www.download.com
2. ทำการเชื่อมต่อเครื่องเบรินเข้ากับบอร์ดไมโครคอนโทรลเลอร์
3. ทำการเปิดโปรแกรม PICkit 2 v2.xx
4. ตรวจสอบการเชื่อมต่อ เลือก Tools>Check communication
5. เลือก File>Import.Hex เพื่อนำเข้าไฟล์ .Hex ที่เรากอมไฟล์ได้จากโปรแกรม MPLAB IDE vx.xx
6. กดปุ่ม write เพื่อเขียนลงบอร์ดไมโครคอนโทรลเลอร์

การใช้งาน TCP/IP-ZigBee เกตย์เวย์

1. จ่ายกระแสไฟด้วยหม้อแปลง 12 โวลต์ เปิดสวิตซ์
2. ต่อสาย UTP ที่พอร์ต RJ-45 เชื่อมระหว่างเกตย์เวย์และคอมพิวเตอร์
3. ตั้งค่าที่คอมพิวเตอร์ดังนี้

IP address : 192.168.6.102

Subnet mask : 255.255.255.0

4. ใช้โปรแกรมทดสอบการรับส่งข้อมูล

1.2. Specifications

Table 1-01. Specifications of the XBee/XBee-PRO OEM RF Modules

Specification	XBee	XBee-PRO
Performance		
Indoor/Urban Range	up to 100 ft. (30 m)	Up to 300' (100 m)
Outdoor RF line-of-sight Range	up to 300 ft. (100 m)	Up to 1 mile (1500 m)
Transmit Power Output (software selectable)	1mW (0 dBm)	60 mW (18 dBm) conducted, 100 mW (20 dBm) EIRP*
RF Data Rate	250,000 bps	250,000 bps
Serial Interface Data Rate (software selectable)	1200 - 115200 bps (non-standard baud rates also supported)	1200 - 115200 bps (non-standard baud rates also supported)
Receiver Sensitivity	-92 dBm (1% packet error rate)	-100 dBm (1% packet error rate)
Power Requirements		
Supply Voltage	2.8 - 3.4 V	2.8 - 3.4 V
Transmit Current (typical)	45mA (@ 3.3 V)	If PL=0 (10dBm): 137mA(@3.3V), 139mA(@3.0V) PL=1 (12dBm): 155mA (@3.3V), 153mA(@3.0V) PL=2 (14dBm): 170mA (@3.3V), 171mA(@3.0V) PL=3 (16dBm): 188mA (@3.3V), 195mA(@3.0V) PL=4 (18dBm): 215mA (@3.3V), 227mA(@3.0V)
Idle / Receive Current (typical)	50mA (@ 3.3 V)	55mA (@ 3.3 V)
Power-down Current	< 10 μ A	< 10 μ A
General		
Operating Frequency	ISM 2.4 GHz	ISM 2.4 GHz
Dimensions	0.960" x 1.087" (2.438cm x 2.761cm)	0.960" x 1.297" (2.438cm x 3.294cm)
Operating Temperature	-40 to 85° C (Industrial)	-40 to 85° C (Industrial)
Antenna Options	Integrated Whip, Chip or U.F.L Connector	Integrated Whip, Chip or U.F.L Connector
Networking & Security		
Supported Network Topologies	Point-to-point, Point-to-multipoint & Peer-to-peer	
Number of Channels (software selectable)	16 Direct Sequence Channels	12 Direct Sequence Channels
Addressing Options	PAN ID, Channel and Addresses	
Agency Approvals		
United States (FCC Part 15.247)	OUR-XBEE	OUR-XBEEPRO
Industry Canada (IC)	4214A XBEE	4214A XBEEPRO
Europe (CE)	ETSI	ETSI (Max. 10 dBm transmit power output)*
Japan	n/a	005NYCA0378 (Max. 10 dBm transmit power output)**

* When operating in Europe: XBee-PRO RF Modules must be configured to operate at a maximum transmit power output level of 10 dBm. The power output level is set using the PL command. The PL parameter must equal "0" (10 dBm).

Additionally, European regulations stipulate an EIRP power maximum of 12.86 dBm (19 mW) for the XBee-PRO and 12.11 dBm for the XBee when integrating high-gain antennas.

** When operating in Japan: Transmit power output is limited to 10 dBm. A special part number is required when ordering modules approved for use in Japan. Contact MaxStream for more information [call 1-801-765-9885 or send e-mails to sales@maxstream.net].

Antenna Options: The ranges specified are typical when using the integrated Whip (1.5 dBi) and Dipole (2.1 dBi) antennas. The Chip antenna option provides advantages in its form factor; however, it typically yields shorter range than the Whip and Dipole antenna options when transmitting outdoors. For more information, refer to the "XBee Antenna" application note located on MaxStream's web site (<http://www.maxstream.net/support/knowledgebase/article.php?kb=153>).

1.5. Pin Signals

Figure 1-03. XBee/XBee-PRO RF Module Pin Numbers (top sides shown - shields on bottom)

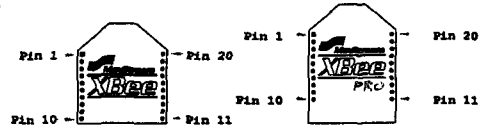


Table 1-02. Pin Assignments for the XBee and XBee-PRO Modules (Low-asserted signals are distinguished with a horizontal line above signal name.)

Pin #	Name	Direction	Description
1	VCC	-	Power supply
2	DOUT	Output	UART Data Out
3	DIN / CONFIG	Input	UART Data In
4	DO8*	Output	Digital Output 8
5	RESET	Input	Module Reset (reset pulse must be at least 200 ns)
6	PWM0 / RSSI	Output	PWM Output 0 / RX Signal Strength Indicator
7	PWM1	Output	PWM Output 1
8	[reserved]	-	Do not connect
9	DTR / SLEEP_RQ / DI8	Input	Pin Sleep Control Line or Digital Input 8
10	GND	-	Ground
11	AD4 / DIO4	Either	Analog Input 4 or Digital I/O 4
12	CTS / DIO7	Either	Clear-to-Send Flow Control or Digital I/O 7
13	ON / SLEEP	Output	Module Status Indicator
14	VREF	Input	Voltage Reference for A/D Inputs
15	Associate / AD5 / DIO5	Either	Associated Indicator, Analog Input 5 or Digital I/O 5
16	RTS / AD6 / DIO6	Either	Request-to-Send Flow Control, Analog Input 6 or Digital I/O 6
17	AD3 / DIO3	Either	Analog Input 3 or Digital I/O 3
18	AD2 / DIO2	Either	Analog Input 2 or Digital I/O 2
19	AD1 / DIO1	Either	Analog Input 1 or Digital I/O 1
20	AD0 / DIO0	Either	Analog Input 0 or Digital I/O 0

* Function is not supported at the time of this release

Design Notes:

q ncaWdWkUa UNRaIOk mM nK L91K K k
 q ncaWdWkUa UNRaIKRdKDptRnKdwAfd OK nKH mK mK L91K GK K 1s
 q GeaFK d UNRaMKip UctKAdN.Kdp UNRN. WRDI
 q nRtDIKaUDt IKFIOsXn pDmDpK dNRdNNEU KNRESET
 SS T T EST T T T ST ST T EST T
 ST TE T ST ST SS

1.6. Electrical Characteristics

Table 1-03. DC Characteristics (VCC = 2.8 - 3.4 VDC)

Symbol	Characteristic	Condition	Min	Typical	Max	Unit
V _{IL}	Input Low Voltage	All Digital Inputs	-	-	0.35 * VCC	V
V _{IH}	Input High Voltage	All Digital Inputs	0.7 * VCC	-	-	V
V _{OL}	Output Low Voltage	I _{OL} = 2 mA, VCC >= 2.7 V	-	-	0.5	V
V _{OH}	Output High Voltage	I _{OH} = -2 mA, VCC >= 2.7 V	VCC - 0.5	-	-	V
I _{IN}	Input Leakage Current	V _{IN} = VCC or GND, all inputs, per pin	-	0.025	1	µA
I _{OZ}	High Impedance Leakage Current	V _{IN} = VCC or GND, all I/O High-Z, per pin	-	0.025	1	µA
TX	Transmit Current	VCC = 3.3 V	-	45 (XBee) 215 (PRO)	-	mA
RX	Receive Current	VCC = 3.3 V	-	50 (XBee) 55 (PRO)	-	mA
PWR-DWN	Power-down Current	SM parameter = 1	-	< 10	-	µA

Table 1-04. ADC Characteristics (Operating)

Symbol	Characteristic	Condition	Min	Typical	Max	Unit
V _{REFH}	VREF - Analog-to-Digital converter reference range	-	2.08	-	V _{DDAD}	V
I _{REF}	VREF - Reference Supply Current	Enabled	-	200	-	µA
		Disabled or Sleep Mode	-	< 0.01	0.02	µA
V _{INDC}	Analog Input Voltage ¹	-	V _{SSAD} - 0.3	-	V _{DDAD} + 0.3	V

1. Maximum electrical operating range, not valid conversion range.

Table 1-05. ADC Timing/Performance Characteristics¹

Symbol	Characteristic	Condition	Min	Typical	Max	Unit
R _{AS}	Source Impedance at Input ²	-	-	-	10	kΩ
V _{AIN}	Analog Input Voltage ³	-	V _{REFL}	-	V _{REFH}	V
RES	Ideal Resolution (1 LSB) ⁴	2.08V ≤ V _{DDAD} ≤ 3.6V	2.031	-	3.516	mV
DNL	Differential Non-linearity ⁵	-	-	±0.5	±1.0	LSB
INL	Integral Non-linearity ⁶	-	-	±0.5	±1.0	LSB
E _{ZS}	Zero-scale Error ⁷	-	-	±0.4	±1.0	LSB
F _{FS}	Full-scale Error ⁸	-	-	±0.4	±1.0	LSB
E _{IL}	Input Leakage Error ⁹	-	-	±0.05	±5.0	LSB
E _{TU}	Total Unadjusted Error ¹⁰	-	-	±1.1	±2.5	LSB

1. All ACCURACY numbers are based on processor and system being in WAIT state (very little activity and no IO switching) and that adequate low-pass filtering is present on analog input pins (filter with 0.01 µF to 0.1 µF capacitor between analog input and VREFL). Failure to observe these guidelines may result in system or microcontroller noise causing accuracy errors which will vary based on board layout and the type and magnitude of the activity.

Data transmission and reception during data conversion may cause some degradation of these specifications, depending on the number and timing of packets. It is advisable to test the ADCs in your installation if best accuracy is required.

2. R_{AS} is the real portion of the impedance of the network driving the analog input pin. Values greater than this amount may not fully charge the input circuitry of the ATD resulting in accuracy error.

3. Analog input must be between V_{REFL} and V_{REFH} for valid conversion. Values greater than V_{REFH} will convert to \$3FF.

4. The resolution is the ideal step size or 1LSB = (V_{REFH} - V_{REFL})/1024

5. Differential non-linearity is the difference between the current code width and the ideal code width (1LSB). The current code width is the difference in the transition voltages to and from the current code.

6. Integral non-linearity is the difference between the transition voltage to the current code and the adjusted ideal transition voltage for the current code. The adjusted ideal transition voltage is (Current Code - 1/2) * (1 / ((V_{REFH} + E_{FS}) - (V_{REFL} + E_{ZS}))).

7. Zero-scale error is the difference between the transition to the first valid code and the ideal transition to that code. The ideal transition voltage to a given code is (Code - 1/2) * (1 / (V_{REFH} - V_{REFL})).

8. Full-scale error is the difference between the transition to the last valid code and the ideal transition to that code. The ideal transition voltage to a given code is (Code - 1/2) * (1 / (V_{REFH} - V_{REFL})).

9. Input leakage error is error due to input leakage across the real portion of the impedance of the network driving the analog pin. Reducing the impedance of the network reduces this error.

10. Total unadjusted error is the difference between the transition voltage to the current code and the ideal straight-line transfer function. This measure of error includes inherent quantization error (1/2LSB) and circuit error (differential, integral, zero-scale, and full-scale) error. The specified value of E_{TU} assumes zero E_{IL} (no leakage or zero real source impedance).



MICROCHIP

PIC18F97J60 Family Data Sheet

**64/80/100-Pin High-Performance,
1-Mbit Flash Microcontrollers
with Ethernet**



PIC18F97J60 FAMILY

64/80/100-Pin High-Performance, 1-Mbit Flash Microcontrollers with Ethernet

Ethernet Features:

- IEEE 802.3™ Compatible Ethernet Controller
- Fully Compatible with 10/100/1000Base-T Networks
- Integrated MAC and 10Base-T PHY
- 8-Kbyte Transmit/Receive Packet Buffer SRAM
- Supports One 10Base-T Port
- Programmable Automatic Retransmit on Collision
- Programmable Padding and CRC Generation
- Programmable Automatic Rejection of Erroneous Packets
- Activity Outputs for 2 LED Indicators
- Buffer:
 - Configurable transmit/receive buffer size
 - Hardware-managed circular receive FIFO
 - Byte-wide random and sequential access
 - Internal DMA for fast memory copying
 - Hardware assisted checksum calculation for various protocols
- MAC:
 - Support for Unicast, Multicast and Broadcast packets
 - Programmable Pattern Match of up to 64 bytes within packet at user-defined offset
 - Programmable wake-up on multiple packet formats
- PHY:
 - Wave shaping output filter

Flexible Oscillator Structure:

- Selectable System Clock derived from Single 25 MHz External Source:
 - 2.778 to 41.667 MHz
- Internal 31 kHz Oscillator
- Secondary Oscillator using Timer1 @ 32 kHz
- Fail-Safe Clock Monitor:
 - Allows for safe shutdown if oscillator stops
- Two-Speed Oscillator Start-up

External Memory Bus (100-pin devices only):

- Address Capability of up to 2 Mbytes
- 8-Bit or 16-Bit Interface
- 12-Bit, 16-Bit and 20-Bit Addressing modes

Peripheral Highlights:

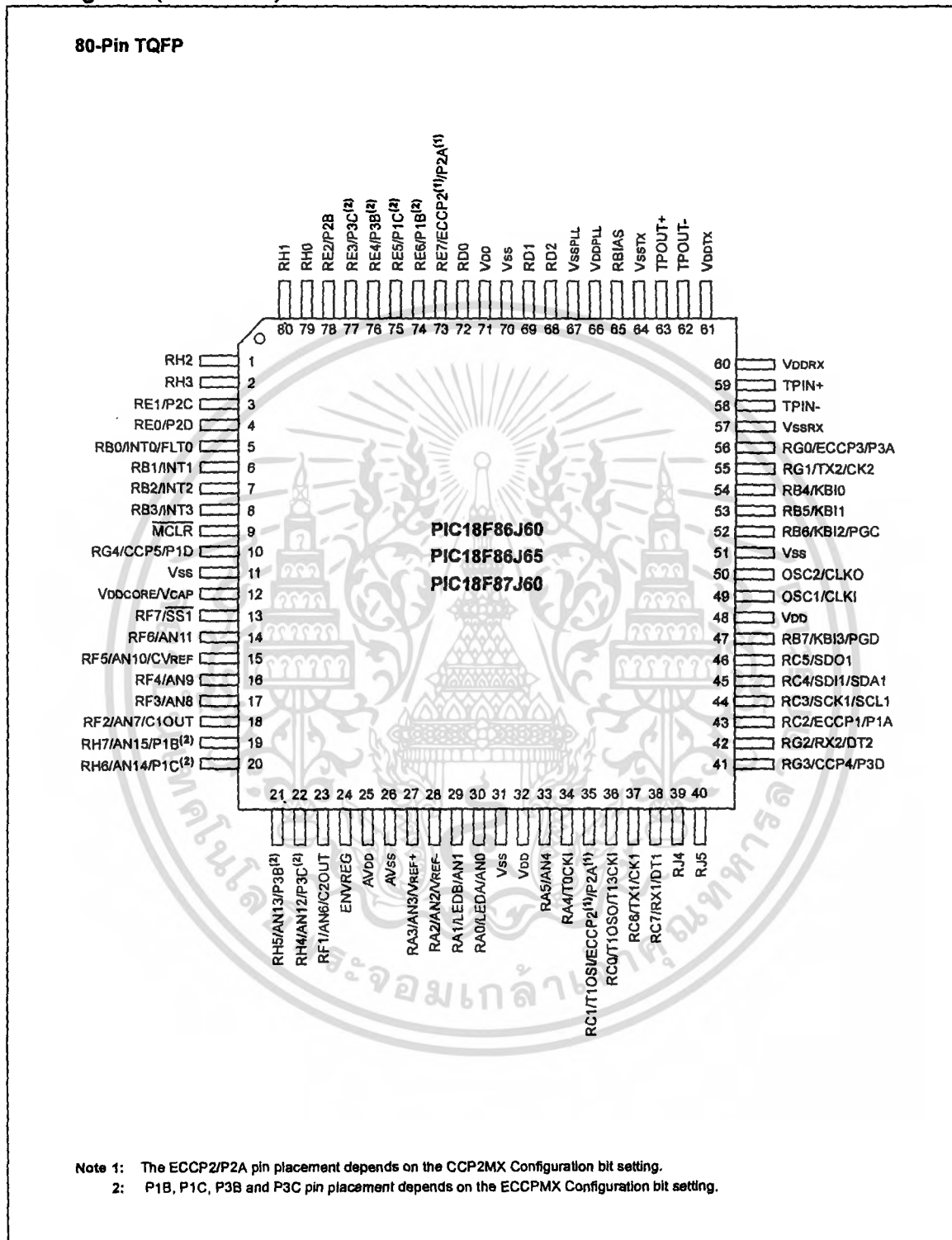
- High-Current Sink/Source: 25 mA/25 mA on PORTB and PORTC
- Five Timer modules (Timer0 to Timer4)
- Four External Interrupt pins
- Two Capture/Compare/PWM (CCP) modules
- Three Enhanced Capture/Compare/PWM (ECCP) modules:
 - One, two or four PWM outputs
 - Selectable polarity
 - Programmable dead time
 - Auto-shutdown and auto-restart
- Up to Two Master Synchronous Serial Port (MSSP) modules supporting SPI (all 4 modes) and I²C™ Master and Slave modes
- Up to Two Enhanced USART modules:
 - Supports RS-485, RS-232 and LIN 1.2
 - Auto-wake-up on Start bit
 - Auto-Baud Detect (ABD)
- 10-Bit, Up to 16-Channel Analog-to-Digital Converter module (A/D):
 - Auto-acquisition capability
 - Conversion available during Sleep
- Dual Analog Comparators with Input Multiplexing
- Parallel Slave Port (PSP) module (100-pin devices only)

Special Microcontroller Features:

- 5.5V Tolerant Inputs (digital-only pins)
- Low-Power, High-Speed CMOS Flash Technology:
 - Self-reprogrammable under software control
- C compiler Optimized Architecture for Reentrant Code
- Power Management Features:
 - Run: CPU on, peripherals on
 - Idle: CPU off, peripherals on
 - Sleep: CPU off, peripherals off
- Priority Levels for Interrupts
- 8 x 8 Single-Cycle Hardware Multiplier
- Extended Watchdog Timer (WDT):
 - Programmable period from 4 ms to 134s
- Single-Supply 3.3V In-Circuit Serial Programming™ (ICSP™) via Two Pins
- In-Circuit Debug (ICD) with 3 Breakpoints via Two Pins
- Operating Voltage Range of 2.35V to 3.6V (3.1V to 3.6V using Ethernet module)
- On-Chip 2.5V Regulator

PIC18F97J60 FAMILY

Pin Diagrams (Continued)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIC18F97J60 FAMILY

1.0 DEVICE OVERVIEW

This document contains device-specific information for the following devices:

- PIC18F66J60
- PIC18F66J65
- PIC18F67J60
- PIC18F86J60
- PIC18F86J65
- PIC18F87J60
- PIC18F96J60
- PIC18F96J65
- PIC18F97J60

This family introduces a new line of low-voltage devices with the foremost traditional advantage of all PIC18 microcontrollers – namely, high computational performance and a rich feature set at an extremely competitive price point. These features make the PIC18F97J60 family a logical choice for many high-performance applications where cost is a primary consideration.

1.1 Core Features

1.1.1 OSCILLATOR OPTIONS AND FEATURES

All of the devices in the PIC18F97J60 family offer five different oscillator options, allowing users a range of choices in developing application hardware. These options include:

- Two Crystal modes, using crystals or ceramic resonators.
- Two External Clock modes, offering the option of a divide-by-4 clock output.
- A Phase Lock Loop (PLL) frequency multiplier, available to the external oscillator modes, which allows clock speeds of up to 41.667 MHz.
- An internal RC oscillator with a fixed 31 kHz output which provides an extremely low-power option for timing-insensitive applications.

The internal oscillator block provides a stable reference source that gives the family additional features for robust operation:

- **Fail-Safe Clock Monitor:** This option constantly monitors the main clock source against a reference signal provided by the internal oscillator. If a clock failure occurs, the controller is switched to the internal oscillator, allowing for continued low-speed operation or a safe application shutdown.
- **Two-Speed Start-up:** This option allows the internal oscillator to serve as the clock source from Power-on Reset, or wake-up from Sleep mode, until the primary clock source is available.

1.1.2 EXPANDED MEMORY

The PIC18F97J60 family provides ample room for application code, from 64 Kbytes to 128 Kbytes of code space. The Flash cells for program memory are rated to last 100 erase/write cycles. Data retention without refresh is conservatively estimated to be greater than 20 years.

The PIC18F97J60 family also provides plenty of room for dynamic application data with 3808 bytes of data RAM.

1.1.3 EXTERNAL MEMORY BUS

In the unlikely event that 128 Kbytes of memory are inadequate for an application, the 100-pin members of the PIC18F97J60 family also implement an External Memory Bus (EMB). This allows the controller's internal program counter to address a memory space of up to 2 Mbytes, permitting a level of data access that few 8-bit devices can claim. This allows additional memory options, including:

- Using combinations of on-chip and external memory up to the 2-Mbyte limit
- Using external Flash memory for reprogrammable application code or large data tables
- Using external RAM devices for storing large amounts of variable data

1.1.4 EXTENDED INSTRUCTION SET

The PIC18F97J60 family implements the optional extension to the PIC18 instruction set, adding eight new instructions and an Indexed Addressing mode. Enabled as a device configuration option, the extension has been specifically designed to optimize reentrant application code originally developed in high-level languages, such as C.

1.1.5 EASY MIGRATION

Regardless of the memory size, all devices share the same rich set of peripherals, allowing for a smooth migration path as applications grow and evolve.

PIC18F97J60 FAMILY

1.2 Other Special Features

- **Communications:** The PIC18F97J60 family incorporates a range of serial communication peripherals, including up to two independent Enhanced USARTs and up to two Master SSP modules, capable of both SPI and I²C™ (Master and Slave) modes of operation. In addition, one of the general purpose I/O ports can be reconfigured as an 8-bit Parallel Slave Port for direct processor-to-processor communications.
- **CCP Modules:** All devices in the family incorporate two Capture/Compare/PWM (CCP) modules and three Enhanced CCP (ECCP) modules to maximize flexibility in control applications. Up to four different time bases may be used to perform several different operations at once. Each of the three ECCP modules offers up to four PWM outputs, allowing for a total of twelve PWMs. The ECCP modules also offer many beneficial features, including polarity selection, programmable dead time, auto-shutdown and restart and Half-Bridge and Full-Bridge Output modes.
- **10-Bit A/D Converter:** This module incorporates programmable acquisition time, allowing for a channel to be selected and a conversion to be initiated without waiting for a sampling period and thus, reducing code overhead.
- **Extended Watchdog Timer (WDT):** This enhanced version incorporates a 16-bit prescaler, allowing an extended time-out range. See Section 27.0 "Electrical Characteristics" for time-out periods.

1.3 Details on Individual Family Members

Devices in the PIC18F97J60 family are available in 64-pin, 80-pin and 100-pin packages. Block diagrams for the three groups are shown in Figure 1-1, Figure 1-2 and Figure 1-3.

The devices are differentiated from each other in four ways:

1. Flash program memory (three sizes, ranging from 64 Kbytes for PIC18FX6J60 devices to 128 Kbytes for PIC18FX7J60 devices).
2. A/D channels (eleven for 64-pin devices, fifteen for 80-pin pin devices and sixteen for 100-pin devices).
3. Serial communication modules (one EUSART module and one MSSP module on 64-pin devices, two EUSART modules and one MSSP module on 80-pin devices and two EUSART modules and two MSSP modules on 100-pin devices).
4. I/O pins (39 on 64-pin devices, 55 on 80-pin devices and 70 on 100-pin devices).

All other features for devices in this family are identical. These are summarized in Table 1-1, Table 1-2 and Table 1-3.

The pinouts for all devices are listed in Table 1-4, Table 1-5 and Table 1-6.

PIC18F97J60 FAMILY

TABLE 1-1: DEVICE FEATURES FOR THE PIC18F97J60 FAMILY (64-PIN DEVICES)

Features	PIC18F66J60	PIC18F66J65	PIC18F67J60
Operating Frequency	DC – 41.667 MHz	DC – 41.667 MHz	DC – 41.667 MHz
Program Memory (Bytes)	64K	96K	128K
Program Memory (Instructions)	32764	49148	65532
Data Memory (Bytes)	3808		
Interrupt Sources	26		
I/O Ports	Ports A, B, C, D, E, F, G		
I/O Pins	39		
Timers	5		
Capture/Compare/PWM Modules	2		
Enhanced Capture/Compare/PWM Modules	3		
Serial Communications	MSSP (1), Enhanced USART (1)		
Ethernet Communications (10Base-T)	Yes		
Parallel Slave Port Communications (PSP)	No		
External Memory Bus	No		
10-Bit Analog-to-Digital Module	11 Input Channels		
Resets (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow, MCLR, WDT (PWRT, OST)		
Instruction Set	75 Instructions, 83 with Extended Instruction Set Enabled		
Packages	64-Pin TQFP		

TABLE 1-2: DEVICE FEATURES FOR THE PIC18F97J60 FAMILY (80-PIN DEVICES)

Features	PIC18F86J60	PIC18F86J65	PIC18F87J60
Operating Frequency	DC – 41.667 MHz	DC – 41.667 MHz	DC – 41.667 MHz
Program Memory (Bytes)	64K	96K	128K
Program Memory (Instructions)	32764	49148	65532
Data Memory (Bytes)	3808		
Interrupt Sources	27		
I/O Ports	Ports A, B, C, D, E, F, G, H, J		
I/O Pins	55		
Timers	5		
Capture/Compare/PWM Modules	2		
Enhanced Capture/Compare/PWM Modules	3		
Serial Communications	MSSP (1), Enhanced USART (2)		
Ethernet Communications (10Base-T)	Yes		
Parallel Slave Port Communications (PSP)	No		
External Memory Bus	No		
10-Bit Analog-to-Digital Module	15 Input Channels		
Resets (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow, MCLR, WDT (PWRT, OST)		
Instruction Set	75 Instructions, 83 with Extended Instruction Set Enabled		
Packages	80-Pin TQFP		

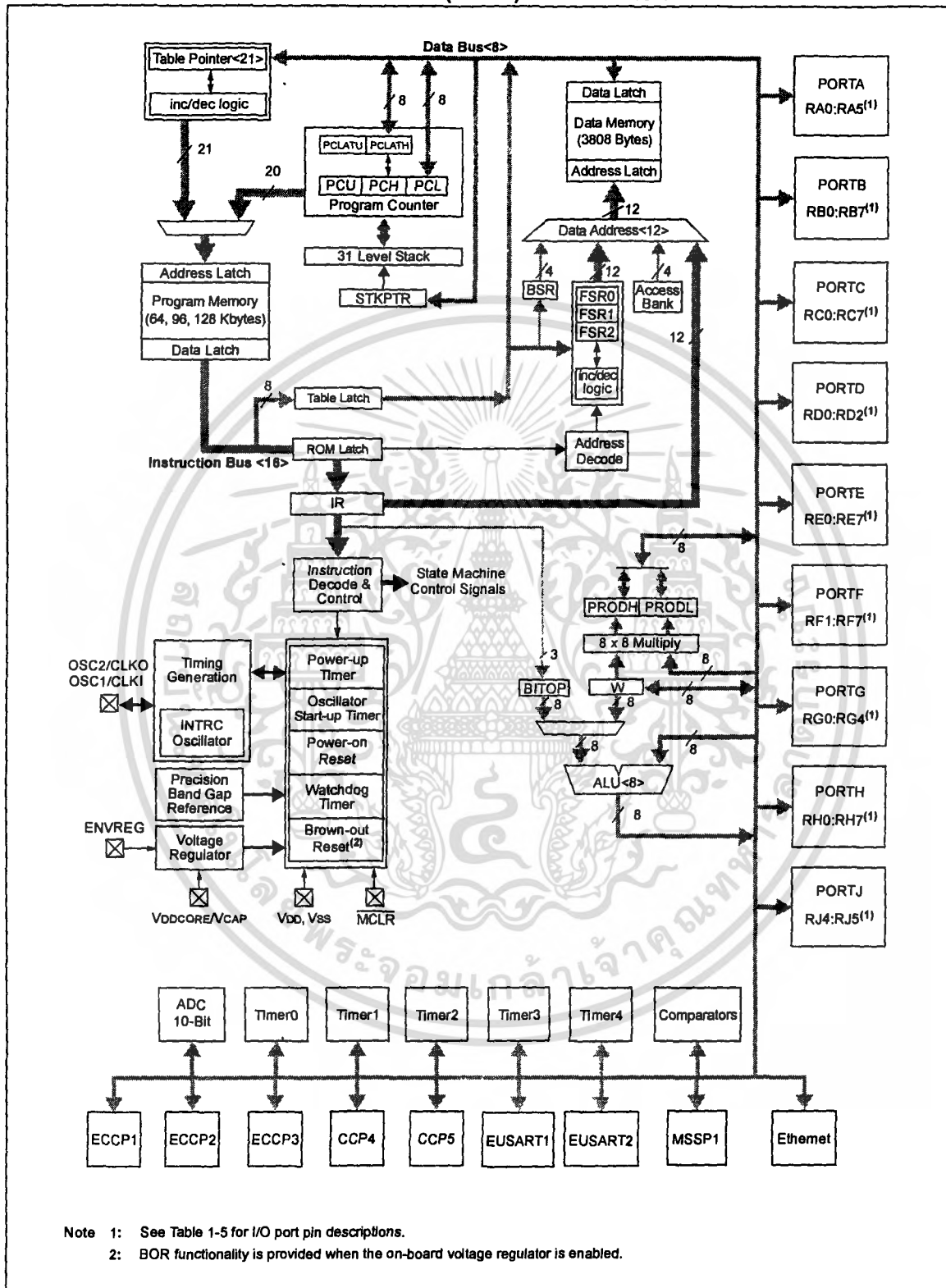
PIC18F97J60 FAMILY

TABLE 1-3: DEVICE FEATURES FOR THE PIC18F97J60 FAMILY (100-pin DEVICES)

Features	PIC18F96J60	PIC18F96J65	PIC18F97J60
Operating Frequency	DC – 41.667 MHz	DC – 41.667 MHz	DC – 41.667 MHz
Program Memory (Bytes)	64K	96K	128K
Program Memory (Instructions)	32764	49148	65532
Data Memory (Bytes)	3808		
Interrupt Sources	29		
I/O Ports	Ports A, B, C, D, E, F, G, H, J		
I/O Pins	70		
Timers	5		
Capture/Compare/PWM Modules	2		
Enhanced Capture/Compare/PWM Modules	3		
Serial Communications	MSSP (2), Enhanced USART (2)		
Ethernet Communications (10Base-T)	Yes		
Parallel Slave Port Communications (PSP)	Yes		
External Memory Bus	Yes		
10-Bit Analog-to-Digital Module	16 Input Channels		
Resets (and Delays)	POR, BOR, RESET Instruction, Stack Full, Stack Underflow, MCLR, WDT (PWRT, OST)		
Instruction Set	75 Instructions, 83 with Extended Instruction Set Enabled		
Packages	100-Pin TQFP		

PIC18F97J60 FAMILY

FIGURE 1-2: PIC18F86J60/86J65/87J60 (80-PIN) BLOCK DIAGRAM



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIC18F97J60 FAMILY

TABLE 1-5: PIC18F86J60/86J65/87J60 PINOUT I/O DESCRIPTIONS

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	TQFP			
MCLR	9	I	ST	Master Clear (Reset) input. This pin is an active-low Reset to the device.
OSC1/CLKI OSC1 CLKI	49	I I	ST CMOS	Oscillator crystal or external clock input. Oscillator crystal input or external clock source input. ST buffer when configured in internal RC mode; CMOS otherwise. External clock source input. Always associated with pin function OSC1. (See related OSC2/CLKO pin.)
OSC2/CLKO OSC2 CLKO	50	O O	— —	Oscillator crystal or clock output. Oscillator crystal output. Connects to crystal or resonator in Crystal Oscillator mode. In internal RC mode, OSC2 pin outputs CLKO which has 1/4 the frequency of OSC1 and denotes the instruction cycle rate.
RA0/LEDA/AN0 RA0 LEDA AN0	30	I/O O I	TTL — Analog	PORTA is a bidirectional I/O port. Digital I/O. Ethernet LEDA indicator output. Analog input 0.
RA1/LEDB/AN1 RA1 LEDB AN1	29	I/O O I	TTL — Analog	Digital I/O. Ethernet LEDB indicator output. Analog input 1.
RA2/AN2/VREF- RA2 AN2 VREF-	28	I/O I I	TTL Analog Analog	Digital I/O. Analog input 2. A/D reference voltage (low) input.
RA3/AN3/VREF+ RA3 AN3 VREF+	27	I/O I I	TTL Analog Analog	Digital I/O. Analog input 3. A/D reference voltage (high) input.
RA4/T0CKI RA4 T0CKI	34	I/O I	ST ST	Digital I/O. Timer0 external clock input.
RA5/AN4 RA5 AN4	33	I/O I	TTL Analog	Digital I/O. Analog input 4.

Legend: TTL = TTL compatible input CMOS = CMOS compatible input or output
 ST = Schmitt Trigger input with CMOS levels Analog = Analog input
 I = Input O = Output
 P = Power OD = Open-Drain (no P diode to VDD)

- Note 1:** Default assignment for ECCP2/P2A when CCP2MX Configuration bit is set.
Note 2: Default assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is set).
Note 3: Alternate assignment for ECCP2/P2A when CCP2MX Configuration bit is cleared.
Note 4: Alternate assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is cleared).

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
 ไม่ว่ากรณีใดๆ ทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

PIC18F97J60 FAMILY

TABLE 1-5: PIC18F86J60/86J65/87J60 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	TQFP			
RC0/T1OSO/T13CKI RC0 T1OSO T13CKI	36	I/O O I	ST — ST	PORTC is a bidirectional I/O port. Digital I/O. Timer1 oscillator output. Timer1/Timer3 external clock input.
RC1/T1OSI/ECCP2/P2A RC1 T1OSI ECCP2 ⁽¹⁾ P2A ⁽¹⁾	35	I/O I I/O O	ST — CMOS ST —	Digital I/O. Timer1 oscillator input. Capture 2 input/Compare 2 output/PWM2 output. ECCP2 PWM output A.
RC2/ECCP1/P1A RC2 ECCP1 P1A	43	I/O I/O O	ST ST —	Digital I/O. Capture 1 input/Compare 1 output/PWM1 output. ECCP1 PWM output A.
RC3/SCK1/SCL1 RC3 SCK1 SCL1	44	I/O I/O I/O	ST ST ST	Digital I/O. Synchronous serial clock input/output for SPI mode. Synchronous serial clock input/output for I ² C™ mode.
RC4/SDI1/SDA1 RC4 SDI1 SDA1	45	I/O I I/O	ST ST ST	Digital I/O. SPI data in. I ² C data I/O.
RC5/SDO1 RC5 SDO1	46	I/O O	ST —	Digital I/O. SPI data out.
RC6/TX1/CK1 RC6 TX1 CK1	37	I/O O I/O	ST — ST	Digital I/O. EUSART1 asynchronous transmit. EUSART1 synchronous clock (see related RX1/DT1 pin).
RC7/RX1/DT1 RC7 RX1 DT1	38	I/O I I/O	ST ST ST	Digital I/O. EUSART1 asynchronous receive. EUSART1 synchronous data (see related TX1/CK1 pin).

Legend: TTL = TTL compatible input
 ST = Schmitt Trigger input with CMOS levels
 I = Input
 P = Power
 CMOS = CMOS compatible input or output
 Analog = Analog input
 O = Output
 OD = Open-Drain (no P diode to V_{DD})

- Note 1:** Default assignment for ECCP2/P2A when CCP2MX Configuration bit is set.
Note 2: Default assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is set).
Note 3: Alternate assignment for ECCP2/P2A when CCP2MX Configuration bit is cleared.
Note 4: Alternate assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is cleared).

PIC18F97J60 FAMILY

TABLE 1-5: PIC18F86J60/86J65/87J60 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	TQFP			
RF1/AN6/C2OUT	23	I/O	ST	PORTF is a bidirectional I/O port. Digital I/O. Analog input 6. Comparator 2 output.
RF1		I	Analog	
AN6 C2OUT		O	—	
RF2/AN7/C1OUT	18	I/O	ST	Digital I/O. Analog input 7. Comparator 1 output.
RF2		I	Analog	
AN7 C1OUT		O	—	
RF3/AN8	17	I/O	ST	Digital I/O. Analog input 8.
RF3 AN8		I	Analog	
RF4/AN9	16	I/O	ST	Digital I/O. Analog input 9.
RF4 AN9		I	Analog	
RF5/AN10/CVREF	15	I/O	ST	Digital I/O. Analog input 10. Comparator reference voltage output.
RF5		I	Analog	
AN10 CVREF		O	—	
RF6/AN11	14	I/O	ST	Digital I/O. Analog input 11.
RF6 AN11		I	Analog	
RF7/SS1	13	I/O	ST	Digital I/O. SPI slave select input.
RF7 SS1		I	TTL	

Legend: TTL = TTL compatible input CMOS = CMOS compatible input or output
 ST = Schmitt Trigger input with CMOS levels Analog = Analog input
 I = Input O = Output
 P = Power OD = Open-Drain (no P diode to V_{DD})

- Note 1:** Default assignment for ECCP2/P2A when CCP2MX Configuration bit is set.
Note 2: Default assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is set).
Note 3: Alternate assignment for ECCP2/P2A when CCP2MX Configuration bit is cleared.
Note 4: Alternate assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is cleared).

PIC18F97J60 FAMILY

TABLE 1-5: PIC18F86J60/86J65/87J60 PINOUT I/O DESCRIPTIONS (CONTINUED)

Pin Name	Pin Number	Pin Type	Buffer Type	Description
	TQFP			
RH0	79	I/O	ST	PORTH is a bidirectional I/O port. Digital I/O.
RH1	80	I/O	ST	Digital I/O.
RH2	1	I/O	ST	Digital I/O.
RH3	2	I/O	ST	Digital I/O.
RH4/AN12/P3C	22	I/O	ST	Digital I/O.
RH4		I	Analog	Analog input 12.
AN12		O	—	ECCP3 PWM output C.
P3C ⁽⁴⁾				
RH5/AN13/P3B	21	I/O	ST	Digital I/O.
RH5		I	Analog	Analog input 13.
AN13		O	—	ECCP3 PWM output B.
P3B ⁽⁴⁾				
RH6/AN14/P1C	20	I/O	ST	Digital I/O.
RH6		I	Analog	Analog input 14.
AN14		O	—	ECCP1 PWM output C.
P1C ⁽⁴⁾				
RH7/AN15/P1B	19	I/O	ST	Digital I/O.
RH7		I	Analog	Analog input 15.
AN15		O	—	ECCP1 PWM output B.
P1B ⁽⁴⁾				

Legend: TTL = TTL compatible input CMOS = CMOS compatible input or output
 ST = Schmitt Trigger input with CMOS levels Analog = Analog input
 I = Input O = Output
 P = Power OD = Open-Drain (no P diode to V_{DD})

- Note 1:** Default assignment for ECCP2/P2A when CCP2MX Configuration bit is set.
2: Default assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is set).
3: Alternate assignment for ECCP2/P2A when CCP2MX Configuration bit is cleared.
4: Alternate assignments for P1B/P1C/P3B/P3C (ECCPMX Configuration bit is cleared).

