การประยุกต์ใช้ซิกบีสำหรับระบบเครือข่ายเซนเซอร์แบบไร้สาย

THE APPLICATION OF ZIGBEE FOR WIRELESS SENSOR NETWORK

อภิลักษณ์ เพ็งจันทร์
อภิสิทธิ์ นำชัยชนะกิจ

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2553

THE APPLICATION OF ZIGBEE FOR WIRELESS SENSOR NETWORK

APILUCK PENGJAN
APISIT NUMCHAICHANAKIJ

THIS THESIS IS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN DEPARTMENT OF ELECTRONICS
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2010

การประยุกต์ใช้ซิกบีสำหรับระบบเครือข่ายเซนเซอร์แบบไร้สาย

THE APPLICATION OF ZIGBEE FOR WIRELESS SENSOR NETWORK

โดย

นายอภิลักษณ์ เพ็งจันทร์ 50011864

นายอภิสิทธิ์ นำชัยชนะกิจ 50011870

อาจารย์ที่ปรึกษา

ดร.กสิน วิเชียรชม

ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต

สาขาวิชาวิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

ปีการศึกษา 2553

ปริญญานิพนธ์ ปีการศึกษา 2553

สาขาวิชา วิศวกรรมอิเล็กทรอนิกส์

คณะวิศวกรรมศาสตร์ สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เรื่อง การประยุกต์ใช้ซิกบีสำหรับระบบเครือข่ายเซนเซอร์แบบไร้สาย

(The Application of Zigbee for Wireless Sensor Network)

ผู้จัดทำ นายอภิลักษณ์ เพ็งจันทร์ รหัส 50011864 ชั้นปีที่ 4

นายอภิสิทธิ์ นำชัยชนะกิจ รหัส 50011870 ชั้นปีที่ 4

.... อาจารย์ที่ปรึกษา

(ดร.กสิน วิเชียรชม)

การประยุกต์ใช้ซิกบีสำหรับระบบเครือข่ายเซนเซอร์แบบไร้สาย

นาย อภิลักษณ์ เพ็งจันทร์ รหัส 50011864
นาย อภิสิทธิ์ นำชัยชนะกิจ รหัส 50011870
ดร. กสิน วิเชียรชม
ปีการศึกษา 2553

บทคัดย่อ

ปริญญานิพนธ์นี้กล่าวถึงการศึกษาและพัฒนาระบบเครือข่ายเซนเซอร์แบบไร้สายโดยในปริญญานิพนธ์นี้ได้เลือกใช้เทคโนโลยีซิกบี ซึ่งจัดอยู่ในมาตรฐาน IEEE 802.15.4 ทำหน้าที่เป็นอุปกรณ์รับ – ส่งสัญญาณ โดยเทคโนโลยีซิกบี นี้มีอัตราการรับส่งข้อมูลต่ำ ราคาถูก และมีการจัดการ โดยใช้พลังงานต่ำ ทำให้สะดวกในการติดตั้งและดูแลรักษา ในการออกแบบปริญญานิพนธ์นี้ได้ใช้โมดูล X-Bee ซึ่งเป็นโมดูลซิกบี ที่มีขายตามท้องตลาด นำมาใช้งานร่วมกับไมโครคอนโทรลเลอร์ ARM7 Cortex-M3 ในการติดต่อกับซิกบี ที่ทำหน้าที่เป็นแม่ข่าย เพื่อประมวลผลและรวบรวมข้อมูล โดยจะส่งข้อมูลทั้งหมดกลับมายังคอมพิวเตอร์ เพื่อแสดงผลภาพรวมทั้งหมดของระบบ

THE APPLICATION OF ZIGBEE FOR WIRELESS SENSOR NETWORK

Mr. Apiluck Pengjan ID.50011864

Mr. Apisit Numchaichanakij ID.50011870

Dr. Kasin Vichienchom Advisor

Education Year 2010

Abstract

This report describes the design and implementation of wireless sensor network (WSN) using Zigbee technology. The network is composed of a gateway and sensor nodes distributed over the designated area. Each node collects the environmental data such as temperature and humidity and periodically sends it to the gateway via the Zigbee compliant wireless link. In this design, the X-Bee modules and the Arm-7 based cortex-m3 development board are used as the Zigbee transceiver for the sensor nodes and the gateway respectively. The client/server system using a PC as a server and gateways as clients has been developed. The server receives and stores the data from each gateway using MySQL database management. The system allows one to monitor each sensor node and in some cases to control switches attached to the node using its interactive GUI. In addition, all the collected data and system status can be viewed over the internet through the system webpage. The system was tested under both open and closed environments. It performs as designed. This prototype system is suitable for applications such as the environmental monitoring and control system.

กิตติกรรมประกาศ

ข้าพเจ้าขอขอบพระคุณอาจารย์ที่ปรึกษา ดร.กสิน วิเชียรชม ที่ให้ความช่วยเหลือ และเป็นผู้ที่ให้คำปรึกษาในการเลือกใช้อุปกรณ์ต่างๆในการนำมาใช้งาน ตลอดจนให้คำแนะนำต่างๆจนทำให้ปริญญานิพนธ์ฉบับนี้สำเร็จลุล่วงได้ด้วยดี ซึ่งข้าพเจ้ารู้สึกซาบซึ้งเป็นอย่างยิ่ง รวมทั้งเพื่อนร่วมชั้นที่ให้คำแนะนำในทุกๆด้าน สุดท้ายบุคคลที่ลืมมิได้เลยคือ ผู้ปกครองซึ่งคอยสนับสนุนและคอยให้กำลังใจเสมอมา

นายอภิลักษณ์ เพ็งจันทร์
นายอภิสิทธิ์ นำชัยชนะกิจ
ผู้จัดทำ

สารบัญ

สารบัญ(ต่อ)

สารบัญ(ต่อ)

สารบัญ(ต่อ)

สารบัญ(ต่อ)

สารบัญ(ต่อ)

สารบัญรูป

สารบัญรูป(ต่อ)

สารบัญรูป(ต่อ)

สารบัญรูป(ต่อ)

สารบัญตาราง

บทที่ 1

บทนำ

1.1 ความสำคัญและที่มาของปัญหา

เนื่องจากในปัจจุบันเทคโนโลยีทางด้านการสื่อสาร และการส่งข้อมูลได้รับความสนใจมากขึ้น และยังมีการพัฒนาขึ้นอย่างรวดเร็ว โดยเฉพาะการสื่อสาร และการส่งข้อมูลแบบไร้สายนั้นเป็นที่นิยมใช้กันอย่างแพร่หลาย ตัวอย่างเช่น โทรศัพท์เคลื่อนที่ไร้สาย, GPRS, Wireless และ Bluetooth เป็นต้น จะพบว่าในการสื่อสารแบบไร้สายนั้นมีด้วยกันหลายประเภท แต่ละประเภทจะมีความเหมาะสมของการสื่อสารที่แตกต่างทั้งรูปแบบการใช้งาน สภาพแวดล้อมที่เหมาะสมต่อการใช้งาน และที่สำคัญคือ ระบบรักษาความปลอดภัย เป็นต้น ดังนั้นในการสื่อสารแบบไร้สายนั้นจำเป็นต้องเลือกระบบการสื่อสารแบบไร้สายให้เหมาะสมกับข้อมูลที่ต้องการรับ-ส่ง และสภาพแวดล้อม จึงทำให้การส่งข้อมูลมีความถูกต้อง และใช้ประโยชน์ได้อย่างคุ้มค่า

Zigbee ก็เป็นหนึ่งในระบบการสื่อสารแบบไร้สาย คือ ระบบเครือข่ายไร้สายส่วนบุคคล (WPAN) อยู่ภายใต้มาตรฐาน IEEE 802.15.4 โดยมาตรฐานนี้ใช้งานสำหรับการสื่อสารความเร็วต่ำ ใช้กำลังไฟฟ้าน้อย อุปกรณ์ราคาถูก และมีคุณสมบัติการจัดการตัวเองได้ เป็นเทคโนโลยีไร้สายที่ร่วมกันสื่อสารข้อมูลผ่านเซ็นเซอร์ขนาดเล็ก เป็นการรับ-ส่งคลื่นสัญญาณข้อมูลจุดต่อจุดไปเรื่อยๆ จนถึงปลายทางที่ต้องการดาวน์โหลดข้อมูลลงในเครื่องคอมพิวเตอร์เพื่อใช้ในการวิเคราะห์ข้อมูล ข้อมูลที่ได้อาจจะเป็นการวัดอุณหภูมิ การเคลื่อนไหวของสิ่งมีชีวิต ปริมาณมลพิษในอากาศ ปริมาณน้ำ ท่อแก๊ส โดยใช้พลังงานแสงอาทิตย์หรือแบตเตอรี่ขนาดเล็กที่ใช้พลังงานน้อยมาก จึงเหมาะสมกับการรับ-ส่งที่มีขนาดไม่เกิน16 บิต ภายในระยะทางไม่เกิน 50 เมตร และมีการส่งข้อมูลแบบไม่ต่อเนื่อง จึงทำให้มีอายุการใช้งานที่นาน อาจจะเป็น 10 ปีก็ได้ จึงทำให้ Zigbee เป็นอีกทางเลือกหนึ่งที่ได้รับความนิยมในการใช้งาน

1.2 วัตถุประสงค์ของโครงงาน

1. เพื่อศึกษา ทดลอง และออกแบบการสื่อสารไร้สายโดยใช้โปรโตคอล Zigbee
2. เพื่อศึกษาการรับ-ส่งข้อมูลผ่านทางมาตรฐาน IEEE 802.15.4
3. เพื่อศึกษาและออกแบบระบบรักษาความปลอดภัยในการส่งข้อมูลแบบไร้สาย
4. ประยุกต์การใช้งาน Zigbee ในรูปแบบของการส่งข้อมูลแบบต่างๆเช่น Star Topology, Tree Topology และ Mesh Topology เพื่อให้เหมาะสมกับงานที่ใช้
5. เพื่อออกแบบและพัฒนาโปรแกรมบริหารจัดการระบบเครือข่าย Zigbee
6. เพื่อศึกษาตัวแปรต่างๆที่มีผลต่อผลผลิตทางการเกษตร
7. เพื่อนำระบบมาประยุกต์ใช้ช่วยเหลือและจัดการงานด้านเกษตรกรรม

1.3 ขอบเขตของโครงงาน

1. สามารถวิเคราะห์ และออกแบบระบบการสื่อสารไร้สายผ่าน Zigbee ได้

2. สามารถอธิบายโครงสร้างการรับ-ส่งข้อมูลของมาตรฐาน IEEE 802.15.4

3. สามารถวิเคราะห์โครงสร้างการใช้งานโปรโตคอลของ Zigbee ได้

4. ระบบมีการรับ-ส่งข้อมูลได้ถูกต้อง แม่นยำ และมีการป้องกันไม่ให้ผู้อื่นสามารถรับรู้ข้อมูลในขณะที่ทำการรับ-ส่งกันได้

5. ระบบการสื่อสารไร้สายผ่าน Zigbee มีการทำงานได้จริงตามแผนการออกแบบในการรับ-ส่งข้อมูล

6. ระบบที่พัฒนาขึ้นมาสารถบริการจัดการระบบเครือข่าย Zigbee ได้ถูกต้องและมีประสิทธิภาพที่น่าพอใจ

7. สามารถเก็บข้อมูลและควบคุมอุณหภูมิและความชื้นที่มีผลต่อผลผลิตทางการเกษตร

8. ระบบสามารถช่วยเหลือและจัดการงานด้านเกษตรกรรมได้อย่างมีประสิทธิภาพ

1.4 ขั้นตอนการดำเนินงาน

แผนภูมิแสดงขั้นตอนการดำเนินโครงงาน

	ขั้นตอนการปฏิบัติงาน	ระยะเวลา								
		มิ.ย.	ก.ค.	ส.ค.	ก.ย.	ต.ค.	พ.ย.	ธ.ค.	ม.ค.	ก.พ.
1.	กำหนดหัวข้อโครงงานและบทคัดย่อ									
2.	ศึกษา วิเคราะห์ และออกแบบชิ้นงาน									
3.	ทดลอง และแก้ไขชิ้นงานที่ออกแบบ									
4.	จัดทำรายงานความคืบหน้าโครงงาน									
5.	ศึกษาและเลือกเซ็นเซอร์ไปใช้ในโครงงานต้นแบบ									
6.	ทดสอบประสิทธิภาพของเซ็นเซอร์									
7.	ออกแบบระบบโครงงานต้นแบบ									
8.	สร้างวงจรที่ใช้ในระบบโครงงานต้นแบบ									
9.	ทดลอง และแก้ไขชิ้นงานที่สร้างขึ้นมา									
10.	จัดทำรายงานโครงงานทั้งหมด									

บทที่ 2

ทฤษฎีและเทคโนโลยี Zigbee

2.1 เทคโนโลยีการสื่อสารไร้สายระยะสั้นด้วยคลื่นวิทยุ

เป็นเทคโนโลยีการสื่อสารไร้สายที่ต้องการกำลังการส่งไม่สูงมากนัก อุปกรณ์มีขนาดเล็กกะทัดรัด สะดวกในการติดตั้งและใช้งาน สามารถใช้งานได้ในบริเวณที่มีคลื่นวิทยุหนาแน่น ทนต่อการเปลี่ยนแปลงของอุณหภูมิ การเพิ่มขนาดของเครือข่ายทำได้โดยสะดวก ลงทุนต่ำกว่าการสื่อสารแบบใช้สาย แต่มีข้อจำกัดคือโปรโตคอลที่ใช้ในการติดต่อสื่อสารต้องมีระบบการจัดการที่ดีเพื่อหลีกเลี่ยงปัญหาการซับซ้อนของระบบเครือข่าย ระยะทางที่รับส่งได้มีจำกัด ถ้าจะส่งข้อมูลระยะไกลก็ต้องใช้อุปกรณ์เพิ่มเติมเป็นจำนวนมากเพื่อให้ข้อมูลไปถึงปลายทางได้อย่างถูกต้อง

เทคโนโลยีการสื่อสารผ่านคลื่นวิทยุนั้นสามารถประยุกต์ใช้งานได้หลายด้าน เช่น ด้านสิ่งแวดล้อม ตัวอย่างคือ ระบบตรวจสอบการเกิดไฟป่า ระบบตรวจสอบสภาพดิน ด้านสุขภาพตัวอย่างคือ การตรวจสอบอัตราการเต้นของหัวใจ ด้านอุตสาหกรรม ตัวอย่างคือ การตรวจสอบการทำงานของเครื่องจักร การสื่อสารผ่านคลื่นวิทยุในระยะสั้นนี้เมื่อแบ่งตามความเร็วในการส่งข้อมูลผ่านเครือข่าย (Throughput) จะแบ่งได้เป็น 3 ประเภทคือ Low, Medium และ High โดย Low Throughput มีความเร็ว 250 Kbit/s ใช้ในงานด้านการตรวจจับ (sensing) การติดตาม(tracking) และการเฝ้าดู (monitoring) ตัวอย่างเช่น Wireless Sensor Network ซึ่งอยู่บนมาตรฐาน IEEE 802.15.4 ส่วน Medium Throughput มีความเร็วในการส่งข้อมูล 1-11 Mbit/s ได้แก่ เทคโนโลยี Bluetooth, IEEE 802.11b, WLAN, Home RF, Personal Area Network (PAN) สำหรับ High throughput มีความเร็วในการส่งข้อมูล 50-100 Mbit/s ได้แก่ เทคโนโลยี IEEE 802.11a

2.2 Zigbee และ มาตรฐาน IEEE 802.15.4

2.2.1 ข้อมูลทั่วไปของ Zigbee

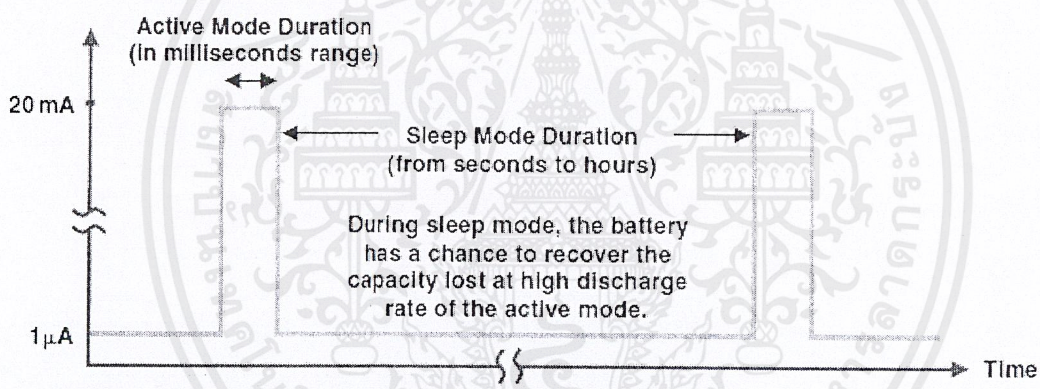
มาตรฐาน Zigbee เป็นโปรโตคอลสำหรับ การส่งข้อมูลที่มีความเร็วต่ำ, ระยะสั้น โดยพื้นฐานของ Zigbee wireless sensor นั้นทำงานอยู่ในช่วง 3 ความถี่ คือ 868 MHz, 915MHz และ 2.4GHz มีความเร็วสูงสุดอยู่ที่ 250 Kbps

จุดมุ่งหมายสำคัญที่สุดของ Zigbee คือ ความเร็วในการส่งที่ต่ำ, ราคาถูก และ ประหยัดพลังงาน เนื่องจากโดยปกติแล้ว เวลาส่วนใหญ่ของ อุปกรณ์ Zigbee นั้นจะอยู่ในโหมดการทำงานแบบ power-saving หรือเรียกอีกอย่างหนึ่งว่า โหมดการทำงานแบบ sleep ซึ่งจะทำให้ Zigbee มีความสามารถคงอยู่ได้แม้เวลาผ่านไปเป็นปี โดยที่ไม่ต้องกลับมาเปลี่ยนแบตเตอรี่บ่อยๆ

ตัวอย่างการประยุกต์ใช้งานของ Zigbee สำหรับการเฝ้าดูผู้ป่วยในบ้าน เช่น ความดันในเลือดของคนไข้ และ ระดับการเต้นของหัวใจ โดยให้คนไข้สวมใส่อุปกรณ์ที่มี Zigbee และมีเซ็นเซอร์ติดอยู่ และทำการส่งข้อมูลไปยัง local server เช่น คอมพิวเตอร์ภายในบ้าน ซึ่งทำให้ นางพยาบาลหรือหมอ สามารถรับรูได้ผ่านเครือข่าย อินเตอร์เน็ตเป็นต้น

มาตรฐานของ Zigbee นั้นถูกพัฒนาโดย Zigbee Alliance ซึ่งมีหลายร้อยบริษัทที่เป็นสมาชิก โดย Zigbee Alliance ถูกก่อตั้งขึ้นเมื่อปี ค.ศ.2002 และเป็นองค์กรที่ไม่แสวงหาผลกำไรและเปิดให้บุคคลอื่นๆ สามารถเข้ามาร่วมใช้งาน Zigbee Standard ซึ่ง Zigbee Standard นั้นได้เลือกใช้งาน IEEE 802.15.4 ในส่วนของ Physical Layer (PHY) และ โปรโตคอล Medium Access Control (MAC) ดังนั้นจึงถือได้ว่ามาตรฐาน Zigbee นั้นสอดคล้องกับมาตรฐานของIEEE 802.15.4

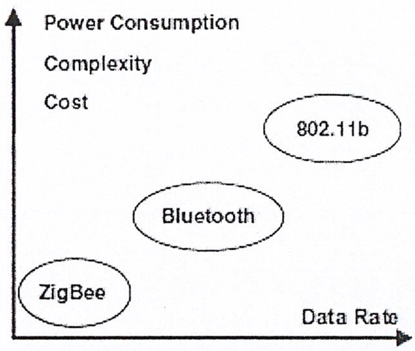
จากรูป 2.1 อัตราส่วนของ duty cycle ของเวลาที่ Zigbee ใช้งาน เทียบกับเวลาทั้งหมดนั้น สามารถเห็นได้ว่า อุปกรณ์ใช้เวลาประมาณ 1% ของเวลาทั้งหมดเพื่อทำการตื่นขึ้นมาซึ่ง ในการประยุกต์ใช้งาน Zigbee นั้น ส่วนใหญ่แล้วจะมี duty cycle น้อยกว่า 1% เพื่อประโยชน์ในด้านการประหยัดพลังงาน



รูป 2.1 เปรียบเทียบ การใช้พลังงาน และ ช่วงเวลาการใช้งานของ Zigbee เทียบกับเวลาทั้งหมด [1]

2.2.2 เปรียบเทียบ Zigbee กับ Bluetooth และ มาตรฐาน IEEE 802.11

อ้างอิงจากมาตรฐาน IEEE 802.11 นั้น IEEE 802.11b นั้นใช้ความถี่ในช่วง 2.4GHz ซึ่ง IEEE 802.11b แบบความเร็วสูง สามารถมีความเร็วในการส่ง ได้สูงสุดถึง 11Mbps โดยระยะทางในการส่งข้อมูลจะอยู่ในช่วง 30-100 เมตร ในส่วนของ Bluetooth ซึ่งจะมีความเร็วที่ต่ำกว่า มีความเร็วในการส่งข้อมูลต่ำกว่า 3Mbps และระยะในการส่งข้อมูลโดยปกติแล้วจะอยู่ในช่วง 2-10 เมตร Zigbee มีความเร็วในการส่งที่น้อยที่สุด สามารถส่งข้อมูลในความเร็วที่อยู่ในช่วง 20-250 Kbps และระยะทางจะอยู่ในช่วง 10-100 เมตร จากรูป 2.2 เมื่อเปรียบเทียบทั้ง 3 แบบ ถือว่า Zigbee สามารถประหยัดพลังงานและต้นทุนได้มากที่สุด



	Data Rate	Typical Range	Application Examples
ZigBee	20 to 250 Kbps	10–100 m	Wireless Sensor Networks
Bluetooth	1 to 3 Mbps	2–10 m	Wireless Headset Wireless Mouse
IEEE 802.11b	1 to 11 Mbps	30–100 m	Wireless Internet Connection

รูป 2.2 เปรียบเทียบ มาตรฐาน Zigbee กับ Bluetooth และ IEEE 802.11b [1]

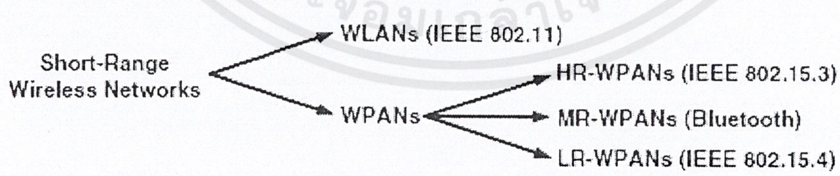
2.2.3 ประเภทเครือข่ายระยะสั้น

เครือข่ายระยะสั้น สามารถแบ่งออกได้เป็น 2 ประเภทใหญ่ๆ คือ local area network (WLANs) และ wireless personal area network (WPANs)

WLANs นั้นเป็นเครือข่ายสำหรับ เครือข่ายแบบมีสาย (LANs) เช่น Ethernet (IEEE 802.3) ซึ่งเป้าหมายของ WLAN นั้นคือ สามารถส่งข้อมูลได้ระยะไกลมากที่สุด และ ความเร็วสูงสุด

ในส่วนของ WPANs นั้น พัฒนาขึ้นมาเพื่อส่งข้อมูลแบบไร้สายในบริเวณ personal operating space (POS) ซึ่ง POS เป็นบริเวณที่มีสัญญาณ กระจายอยู่ทั่วในรัศมี 10เมตร (33ฟุต)

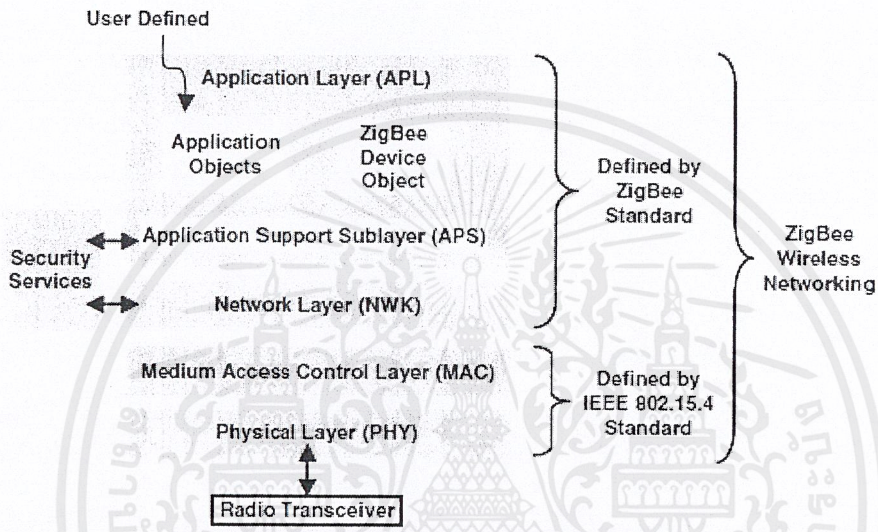
WPLANs ยังสามารถจำแนกออกได้เป็น 3 ประเภท คือ high-rate (HR) WPANs, medium-rate (MR) WPANs และ low-rate (LR) WPANs ตัวอย่างของ HR-WPAN คือ IEEE 802.15.3 ซึ่งสามารถส่งข้อมูลด้วยความเร็วในการส่ง 11-55 Mbps ถูกใช้ในระบบเครือข่าย วีดิโอไร้สาย เป็นต้น ในส่วนของ MR-WAN เช่น Bluetooth ที่มีความเร็วในการส่ง 1-3 Mbps เช่นการใช้งานในการส่งข้อมูลเสียงผ่านหูฟัง และ Zigbee ซึ่งมีความเร็วสูงสุด 250 Kbps นั้นถูกจัดอยู่ในประเภท LR-WPAN ตามรูป 2.3



รูป 2.3 ประเภทของเครือข่ายระยะสั้น [1]

2.3 ความสัมพันธ์ระหว่าง Zigbee และ มาตรฐาน IEEE802.15.4

จากรูป 2.4 แสดง Zigbee โปรโตคอลเลเยอร์ ซึ่งมีพื้นฐานมาจาก Open System Interconnect (OSI) ซึ่งสามารถแบ่งออกเป็นชั้นเลเยอร์ต่างๆ สังเกตเห็นได้ว่า ในการพัฒนาแอพพลิเคชั่นต่างๆ ในส่วนเลเยอร์ของโปรโตคอลที่อยู่ต่ำกว่าจะถูกแยกอิสระจากแอพพลิเคชั่นเลเยอร์ ทำให้ง่ายต่อการพัฒนาแอพพลิเคชั่น แต่ยังคงสามารถพัฒนาแอพพลิเคชั่นให้สนับสนุนการทำงานของโปรโตคอลที่อยู่ต่ำกว่าได้ เราสามารถเรียกซอฟท์แวร์ที่พัฒนาให้มีผลต่อโปรโตคอล ซึ่งเป็นที่รู้จักว่า protocol stack software



รูปที่ 2.4 Zigbee โปรโตคอลเลเยอร์ [1]

มาตรฐานของ Zigbee นั้น ถูกระบุด้วย Networking Layer, Application Layer และ Security Layer เท่านั้น ในส่วนของ MAC และ PHY นั้น เป็นส่วนของ Network Protocol ของ Zigbee ที่ได้นำ มาตรฐาน IEEE 802.15.4 มาใช้งาน อุปกรณ์ที่ได้มาตรฐานของ Zigbee จึงเป็นอุปกรณ์ที่ได้มาตรฐาน IEEE 802.15.4 ด้วยเช่นกัน

ในส่วนของลักษณะ Physical-level นั้น คือ ข้อมูลจำเพาะของ PHY Layer ซึ่งเป็นตัวที่จะระบุ ความถี่ที่ใช้งาน, ความเร็วในการส่งข้อมูล, ความไวในการรับข้อมูล และ รูปแบบอุปกรณ์ ซึ่งเป็นมาตรฐานของ IEEE 802.15.4

2.4 ความถี่ที่ใช้งานและความเร็วในการส่งข้อมูล

ความถี่ที่ใช้งานตามมาตรฐานของ IEEE 802.15.4 มีทั้งหมด 3 ความถี่ดังนี้คือ

1. 868-868.6 MHz (868 MHz Band)
2. 902-928 MHz (915 MHz Band)
3. 2400-2483.5 MHz (2.4 GHz Band)

ในส่วนของ 868 MHz Band นั้นเป็นช่วงความถี่ที่ใช้ในแถบประเทศยุโรปสำหรับเครือข่ายต่างๆ อีกสองความถี่นั้นคือ 915 MHz และ 2.4 GHz นั้นเป็นการกำหนดไว้เพื่อ อุตสาหกรรม, วิทยาศาสตร์ และ ทางการแพทย์ (ISM Band) ซึ่ง 915 MHz นั้นจะใช้ในแถบอเมริกาเหนือ และ 2.4 GHz เป็นย่านความถี่ที่ใช้ทั่วโลก

	Frequency (MHz)	Number of Channels	Modulation	Chip Rate (Kchip/s)	Bit Rate (Kb/s)	Symbol Rate (Ksymbol/s)	Spreading Method
	868-868.6	1	BPSK	300	20	20	Binary DSSS
	902-928	10	BPSK	600	40	40	Binary DSSS
Optional	868-868.6	1	ASK	400	250	12.5	20-bit PSSS
	902-928	10	ASK	1600	250	50	5-bit PSSS
Optional	868-868.6	1	O-QPSK	400	100	25	16-array orthogonal
	902-928	10	O-QPSK	1000	250	62.5	16-array orthogonal
	2400-2483.5	16	O-QPSK	2000	250	62.5	16-array orthogonal

ตารางที่ 2.1 ความเร็วในการส่งข้อมูลและความถี่ที่ใช้ มาตรฐาน IEEE 802.15.4 [1]

ตารางที่ 2.1 แสดงถึงความถี่ทั้ง 3 ย่าน ซึ่งเป็นไปตามมาตรฐาน IEEE 802.15.4 ซึ่งถ้าต้องการใช้ความเร็วในการส่งข้อมูลที่มากกว่า 40 Kbps นั้นจะต้องเลือกใช้ความถี่ย่าน 2.4 GHz ในการส่งข้อมูล (หากไม่พิจารณาที่ความถี่ทางเลือก) ซึ่งการมอดูเลชั่นในมาตรฐาน IEEE 802.15.4 นั้นแบบ BPSK และ O-QPSK ข้อมูลดิจิตอลจะอยู่ในเฟสของสัญญาณ แต่แบบ ASK นั้นข้อมูลดิจิตอลจะอยู่ในแอมพลิจูตของสัญญาณ

2.5 รูปแบบเครือข่ายแบบต่างๆของ Zigbee

มาตรฐาน IEEE 802.15.4 เครือข่ายแบบไร้สายนั้น สามารถแบ่งอุปกรณ์ออกเป็น 2 ประเภท คือ full-function devices (FFDs) และ reduced-function devices (RFDs) โดยที่ FFD นั้น สามารถรองรับหน้าที่ต่างๆ ที่เป็นไปตาม IEEE 802.15.4 และสามารถตอบรับการเข้าถึงของบทบาทต่างๆในเครือข่ายได้ทั้งหมด ในส่วนของ RFD นั้นมีการจำกัดความสามารถในการทำหน้าที่ต่างๆ เช่น FFD สามารถเข้าถึงกับทุกๆอุปกรณ์ในเครือข่ายได้ทั้งหมด แต่ RFD สามารถเข้าถึงเพียงแค่ FFD เท่านั้น เป็นต้น ซึ่งโดยทั่วไปแล้ว อัตราการสูญเสียพลังงาน และหน่วยความจำของอุปกรณ์ที่ทำหน้าที่เป็น RFD นั้น จะน้อยกว่าอุปกรณ์ที่ทำหน้าที่เป็น FFD

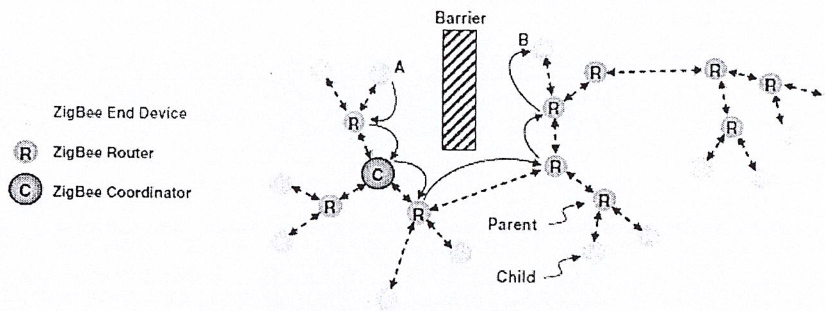
2.5.1 เครือข่ายแบบสตาร์ (Star Topology)

รูป 2.4 ประกอบด้วยจุดเชื่อมต่อโปรโตคอล Zigbee 1 จุด ทำหน้าที่เป็น Personal Area Network (PAN) Coordinator โดยคุณสมบัติมีลักษณะเป็น full-function devices (FFD) เท่านั้น และอุปกรณ์ปลายทางหลาย ๆ ชิ้นในเครือข่ายแบบสตาร์ อุปกรณ์ปลายทางทั้งหมดจะสื่อสารกับ PAN Coordinator เท่านั้น ถ้าอุปกรณ์ปลายทางหนึ่งต้องการสื่อสารกับอุปกรณ์ปลายทางอื่น ๆ ต้องส่งข้อมูลผ่าน PAN Coordinator เท่านั้นหรือกล่าวอีกนัยหนึ่งคือ PAN Coordinator ทำหน้าที่ส่งข้อมูลไปยังผู้รับ

รูปที่ 2.4 การเชื่อมต่อเครือข่ายแบบสตาร์ [1]

2.5.2 เครือข่ายแบบต้นไม้ (Tree Topology)

จากรูป 2.5 แสดงความต้องการที่จะส่งข้อมูลจากอุปกรณ์หนึ่งไปยังอีกอุปกรณ์หนึ่ง โดยมีสิ่งกีดขวางกั้นอยู่ ซึ่งในตัวอย่าง อุปกรณ์จุด A ต้องการส่งข้อมูลไปยังอุปกรณ์จุด B ซึ่งมีกำแพงกั้นระหว่างอุปกรณ์สองตัวนี้ ซึ่งเป็นสิ่งที่ยากที่จะลอดผ่านกำแพงนี้



รูปที่ 2.5 การเชื่อมต่อเครือข่ายแบบต้นไม้ [1]

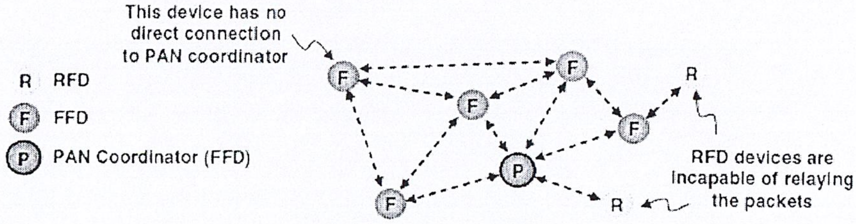
การเชื่อมต่อแบบต้นไม้ช่วยแก้ไขปัญหาดังกล่าวโดยการสามารถถ่ายทอดข้อมูลไปยัง จุด B ได้ โดยใช้วิธี multihopping เนื่องจากข้อมูลสามารถกระโดดจากจุดหนึ่งไปยังอีกจุดหนึ่ง จนกระทั่งถึงจุดหมายปลายทาง

ในเครือข่ายนี้อุปกรณ์ปลายทางจะสามารถเชื่อมต่อได้กับอุปกรณ์เชื่อมต่อหรือ ZigBee โปรโตคอลเราท์เตอร์ ซึ่งเราท์เตอร์ทำหน้าที่ 2 ประเภท คือ เพิ่มจำนวน โนดที่สามารถเชื่อมต่ออยู่บนเครือข่ายและขยายขนาดของเครือข่าย เนื่องจากเราท์เตอร์จะทำหน้าที่ส่งข้อมูลไปยังจุดต่างๆของเครือข่ายได้โดยที่ อุปกรณ์ปลายทางไม่จำเป็นต้องอยู่ในระยะการส่งสัญญาณวิทยุ

2.5.3 เครือข่ายแบบเมซ (Mesh Topology)

การเชื่อต่อเครือข่ายแบบ Mesh เป็นโครงข่ายที่มีประสิทธิภาพสูงเนื่องจาก ข้อมูลสามารถส่งไปถึงเป้าหมายได้หลายทาง ทำให้ระบบนี้สามารถรับส่งข้อมูล ไปยังจุดหมายปลายทางได้ แม้จะเกิดความเสียหายของระบบในบางส่วนก็ตาม (ขึ้นอยู่กับ การออกแบบระบบของผู้ใช้ด้วย) ระบบนี้จึงเป็นระบบที่ได้รับความนิยมเป็นอย่างมาก

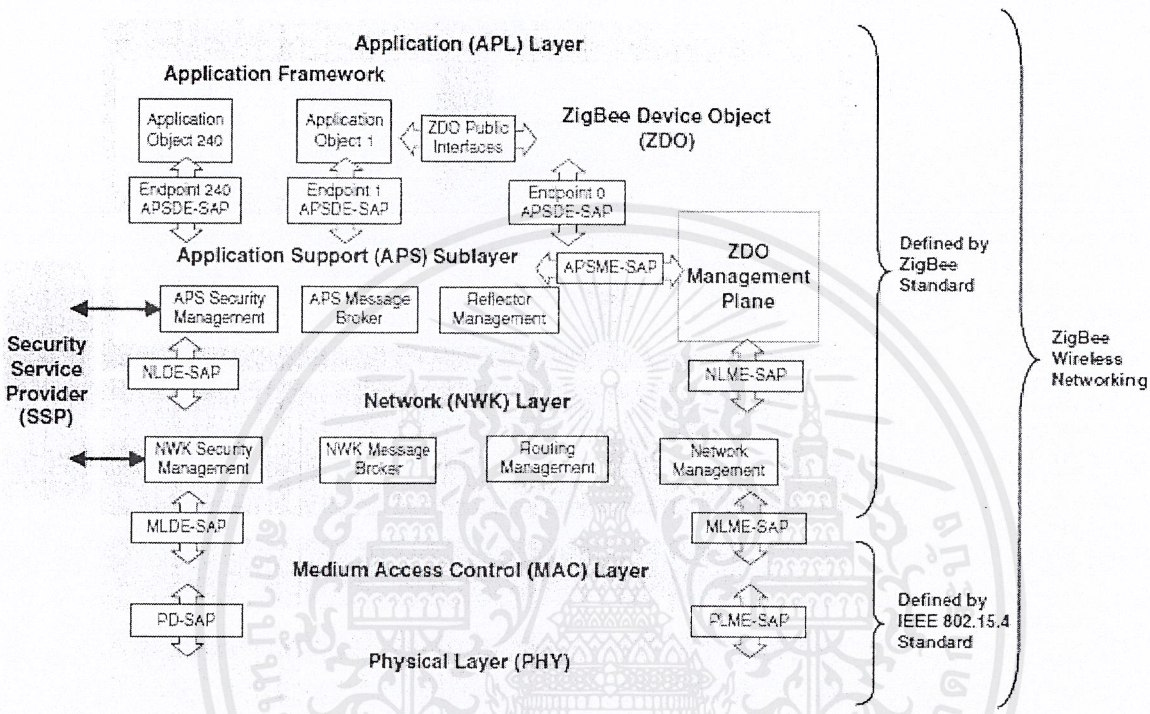
รูปที่ 2.6 แสดงการเชื่อมต่อแบบเมซ (Mesh Topology) หรือ peer-to-peer network อุปกรณ์แต่ละตัวสามารถติดต่อกับอุปกรณ์อื่นๆได้อย่างอิสระ โดยมีจุดที่ทำหน้าที่ PAN Coordinator และ Router จะมีคุณสมบัติเป็น full-function devices (FFD) เท่านั้น อย่างไรก็ตาม RFD ยังสามารถเป็นส่วนหนึ่งของ ระบบเครือข่ายชนิดนี้ได้ เพียงแต่จะสามารถติดต่อกับ FFD เพียงตัวใดตัวหนึ่งเท่านั้น



รูปที่ 2.6 การเชื่อมต่อเครือข่ายแบบเมซ [1]

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

2.6 Zigbee โปรโตคอลสแตก



รูปที่ 2.7 Zigbee Protocol Layer [1]

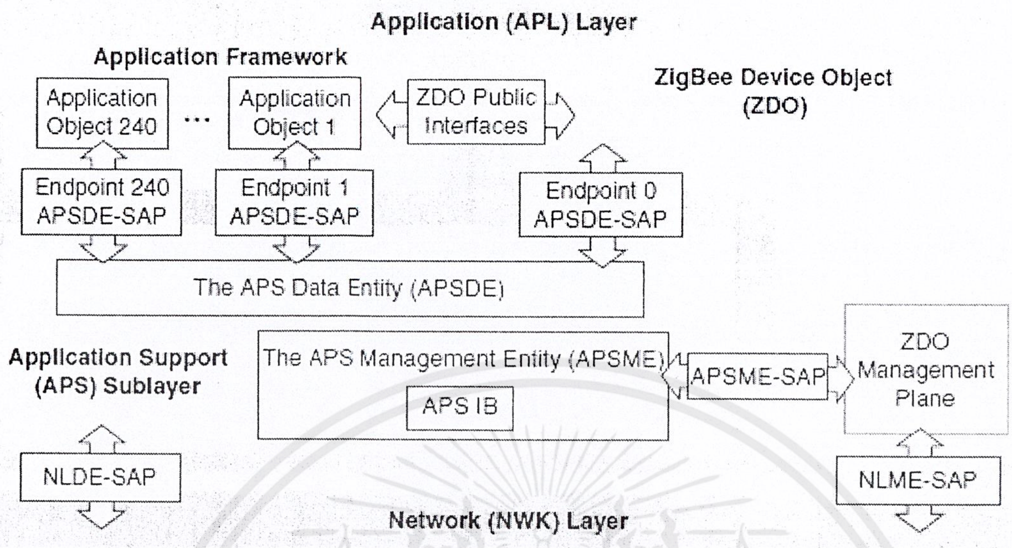
2.6.1 Application (APL) Layer

ในส่วนของ Application Layer หรือ (APL) นั้นเป็นส่วนที่อยู่สูงสุดของโปรโตคอลเลเยอร์ ของ Zigbee Wireless Sensor โดยสังเกตจากรูปที่ 2.8 ในส่วนของ APL นั้นสามารถแบ่งออกเป็น 3 ส่วนใหญ่ๆดังนี้คือ Application Framework, Application Support (APS) Sublayer และ Zigbee Device Object (ZDO)

Application Support (APS) Sublayer นั้นมีหน้าที่เชื่อมโยงการทำงานระหว่างในส่วนของ Network (NWK) Layer กับ Application Layer (APL) ในส่วนของ APS นี้สามารถแบ่งหน้าที่ออกได้เป็น 2 ส่วนใหญ่ๆคือ การบริหารข้อมูล (APS Data Entity) และ การจัดการ (APS Management Entity)

APSDE นั้นสามารถเข้าถึงได้โดยพัฒนาวัตถุแอพพลิเคชั่นผ่าน Endpoint จุดต่างๆ โดยเรียกช่องทางสำหรับการสื่อสารเหล่านี้ว่า APSDE Service Access Point (SAP) ซึ่งแอพพลิเคชั่นออบเจ็กต์เหล่านี้จะพัฒนาจากผู้ผลิต ซึ่งในตัวอุปกรณ์ Zigbee สามารถถูกปรับปรุงจากผู้ผลิตได้หลายรูปแบบ ทั้งนี้ แอพพลิเคชั่นออบเจ็กต์ต่างๆเหล่านี้

สามารถมีช่องการสื่อสารรองรับได้มากถึง 240 endpoint หรือ 240 แอพพลิเคชั่นต่อ 1 อุปกรณ์



รูปที่ 2.8 Zigbee Application Layer [1]

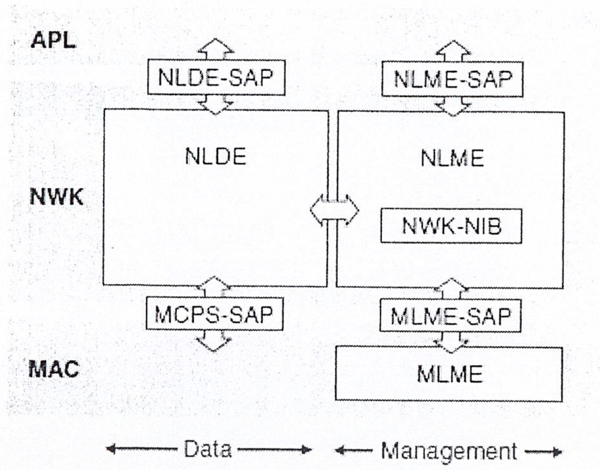
Zigbee Device Object (ZDO) เป็นช่องทางการสื่อสารกับ APS Sublayer และ Application Framework ใน ZDO นั้นจะเก็บฟังก์ชั่นการทำงานต่างๆซึ่งเป็นฟังก์ชั่นการทำงานพื้นฐานของทุกๆแอพพลิเคชั่น ที่จะนำมาใช้งานใน Zigbee โปรโตคอลแสตก เช่น ทำหน้าที่เก็บโหมดการทำงานต่างๆ ไม่ว่าจะระบุโหมดการทำงานเป็น Coordinator, Router หรือ End device โดยจะส่งข้อมูลเหล่านี้ผ่านทาง endpoint ที่ 0 (APSDE-SAP) ซึ่งช่องทางนี้มีไว้สำหรับ ZDO เท่านั้น หน้าที่อื่นๆ เช่น ใช้ในการหาเส้นทางของข้อมูล, การค้นหาเครือข่าย หน้าที่ต่างๆเหล่านี้จะถูกส่งไปที่ APS Management Entity โดยผ่านช่องทางที่เรียกว่า APSME-SAP

2.6.2 Network (NWK) Layer

จากรูป 2.9 แสดงถึง Zigbee Network Layer ซึ่งเป็นส่วนที่อยู่ระหว่าง Application Layer และ MAC Layer โดย NWK สามารถแบ่งหน้าที่ออกได้เป็น 2 ส่วนใหญ่ๆคือ การบริหารข้อมูล (NWK Layer Data Entity) และ การจัดการ (NWK Layer Management Entity)

NWK Layer Data Entity (NLDE) ทำหน้าที่เกี่ยวกับ การส่งข้อมูล โดยส่วนบริหารด้านข้อมูลนั้น ถูกส่งผ่านมาจาก NLDE Service Access Point (SAP) และ NWK Layer Management Entity (NLME) ทำหน้าที่ด้านการจัดการหน้าที่ต่างๆ ซึ่งจะเชื่อมต่อกับ NLME-SAP ที่อยู่ในส่วนของ ZDO ใน Application Layer

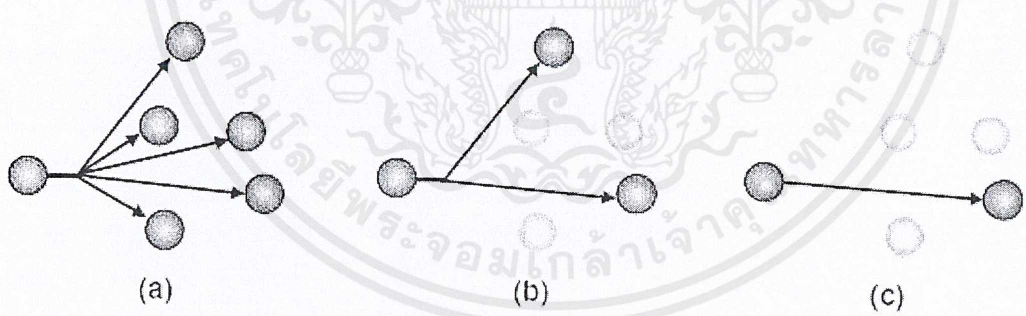
ในส่วนของ Network Layer นี้ได้เก็บค่าคงที่และคุณลักษณะต่างๆ ซึ่งเรียกว่า Network Information Base (NIB) ซึ่งผู้พัฒนา Zigbee นั้นสามารถ นำค่าคงที่และคุณลักษณะต่างๆเหล่านี้ออกมาพัฒนาในส่วนของ Application Layer ได้



รูปที่ 2.9 Zigbee Network Layer ติดต่อกับ Application Layer และ MAC Layer [1]

ในชั้น Network Layer ของ Zigbee ที่เป็นแม่ข่าย (Coordinator) นั้นจะกำหนด 16-bit network address ให้กับแต่ละอุปกรณ์ในเครือข่าย โดยที่การกำหนดนั้นจะเป็นไปตาม IEEE 802.15.4 MAC address ซึ่งหากมีอุปกรณ์ใหม่ๆ ภายในเครือข่ายเข้ามาจะต้องกำหนด 16-bit IEEE 802.15.4 MAC short address ให้เหมือนกันด้วย

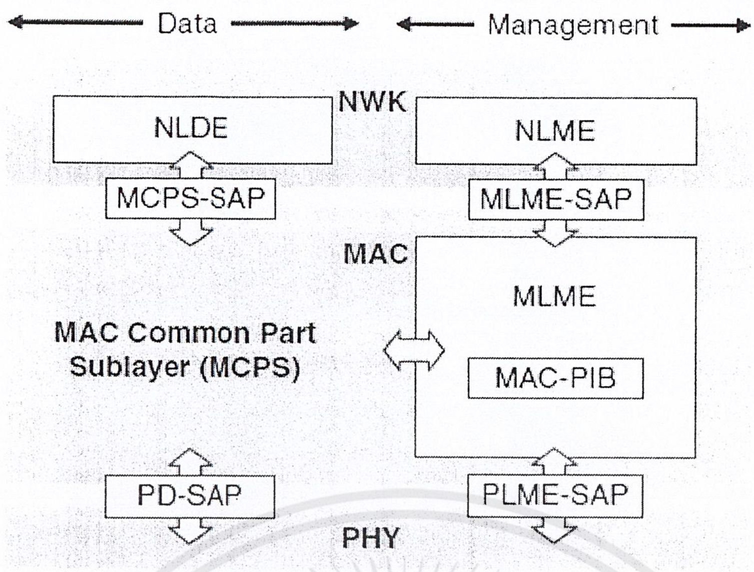
กลไกลการเชื่อมต่อนั้นมี 3 ประเภทหลักๆดังรูป 2.10 คือ boardcast, multicast และ unicast ซึ่ง boardcast นั้นตั้งใจที่จะส่งข้อมูลให้กับทุกๆอุปกรณ์ในเครือข่าย multicast เป็นการระบุข้อมูลที่จะส่งให้เป็นบางอุปกรณ์เท่านั้น ส่วน unicast เป็นการเจาะจงส่งข้อมูลไปที่อุปกรณ์ใดอุปกรณ์หนึ่งภายในเครือข่าย



รูปที่ 2.10 กลไกลในการเชื่อมต่อรูปแบบต่างๆ (a) Boardcast [1], (b) Multicast [1], (c) Unicast [1]

2.6.3 IEEE 802.15.4 MAC Layer

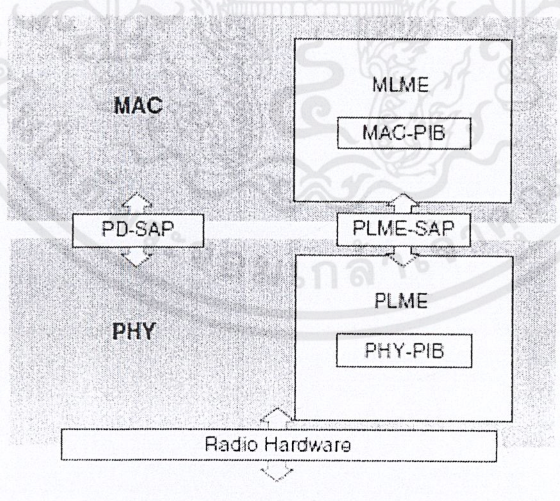
จากรูป 2.11 แสดงถึง MAC Layer ซึ่งเป็นส่วนของเลเยอร์ที่ติดต่อกับ Network Layer ด้านบน และ Physical Layer ด้านล่าง โดยจะมี การบริหารส่วนของหน้าที่ (MAC Layer Management Entity) หรือ MLME ซึ่งจะมีการเก็บฐานข้อมูลที่จำเป็นของตัวเอง โดยจะอ้างอิงจาก MAC PAN Information Base (MAC-PIB) ซึ่งคุณลักษณะรวมถึงค่าคงที่ต่างๆนี้จะถูกอ้างอิงตาม IEEE 802.15.4



รูปที่ 2.11 Zigbee MAC Layer [1]

2.6.4 IEEE 802.15.4 Physical Layer

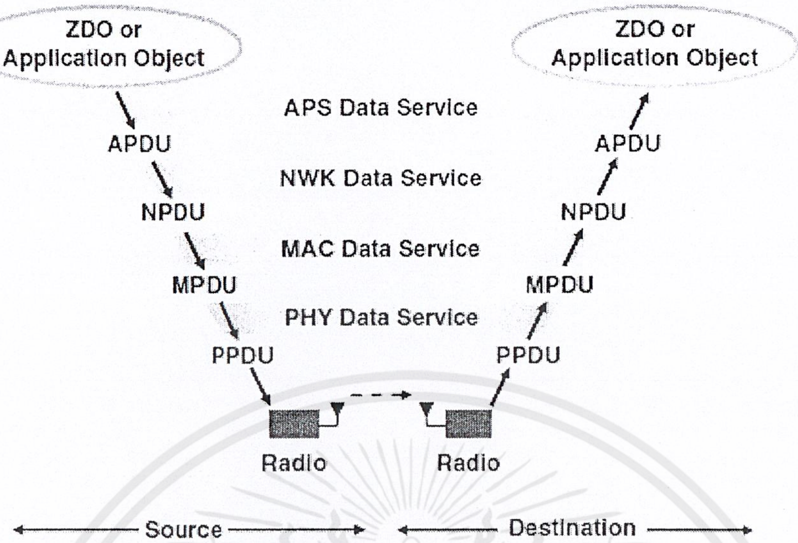
มาตรฐานของ IEEE 802.15.4 นั้นไม่ได้กำหนดหน้าที่ต่างๆของ Physical Layer หรือส่วนที่ติดต่อกับ MAC Layer ทั้งหมด แต่จะมีเพียงบางส่วนที่ถูกกำหนดเป็นพื้นฐาน เช่น ความไวในการรับสัญญาณ, กำลังส่งข้อมูลทางออก เป็นต้น



รูปที่ 2.12 Zigbee Physical Layer [1]

การทำงานในชั้นนี้มี 2 ส่วนใหญ่ๆคือ การบริหารข้อมูล (PHY data service) และ การจัดการ (PHY management service) โดยจากรูป 2.13 PHY data service จะเป็นตัวกำหนดการเปิดใช้งาน PHY Protocol Data Unit

(PPDU) ซึ่งมีหน้าที่ ทำการส่งข้อมูล, การตอบรับ ผ่านช่องทางการสื่อสารของคลื่นวิทยุ



รูปที่ 2.13 การส่งข้อมูลระหว่างอุปกรณ์ 2 ตัว [1]

2.7 โหมดการทำงานของ Zigbee

2.7.1 ขั้นตอนการทำงานของ Zigbee Coordinator

ZigBee Coordinator จะเริ่มต้นเครือข่าย โดยการตรวจสอบการใช้ช่องสัญญาณวิทยุภายในบริเวณรอบๆ ถ้ามีช่องสัญญาณที่ไม่ถูกใช้โดย coordinator ตัวอื่น ก็สามารถเริ่มต้นเครือข่ายได้ หลังจากนั้น coordinator ก็จะทำหน้าที่เป็นศูนย์กลางของเครือข่าย รองรับการเข้าร่วมเครือข่ายของ Zigbee end-device และรองรับการร้องขออื่นๆ ตามมาตรฐานด้วยเช่นกัน ในโครงงานนี้ coordinator รองรับการเข้าร่วมเครือข่าย การออกจากเครือข่าย และการร้องขอการ Binding เท่านั้น

2.7.2 ขั้นตอนการทำงานของ Zigbee End-Device

Zigbee end-device จะเริ่มต้นการทำงานโดยการร้องขอการเข้าร่วมเครือข่ายไปยัง coordinator ประจำเครือข่ายนั้นๆ โดยการตรวจสอบผ่านช่องสัญญาณต่างๆ ว่า coordinator ใช้ช่องสัญญาณใดอยู่เมื่อเข้าร่วมเครือข่ายแล้ว end-device จึงสามารถทำการร้องขอคำสั่งอื่นๆ ผ่านทาง coordinator ได้ เช่น การส่งข้อความทั่วไป (Message), การร้องขอการ binding (binding request), การขอออกจากเครือข่าย

2.8 หลักการติดต่อสื่อสารของ Zigbee

2.8.1 CSMA-CA

IEEE 802.15.4 ได้กำหนดวิธีการการเข้าถึงหลายๆอุปกรณ์โดยใช้ช่องสัญญาณความถี่ช่องเดียวกัน โดยกลไกลวิธีนี้เรียกว่า Carrier Sense Multiple Access with Collision Avoidance (CSMA-CA) เมื่อครั้งใด อุปกรณ์ต้องการส่งข้อมูล ขั้นตอนแรกจะทำการประเมิณว่าช่องสัญญาณว่างหรือไม่ หรือ Clear Channel Assessment (CCA) เพื่อให้แน่ใจว่าช่องสัญญาณนั้นไม่ถูกใช้โดยอุปกรณ์อื่นๆในเครือข่าย หลังจากนั้น อุปกรณ์จึงจะทำการส่งข้อมูลของตัวเองออกไป ในการตัดสินใจว่าช่องสัญญาณนั้นว่างหรือไม่ว่าง ใช้วิธีการวัดความเข้มของพลังงานในช่องความถี่นั้นๆ

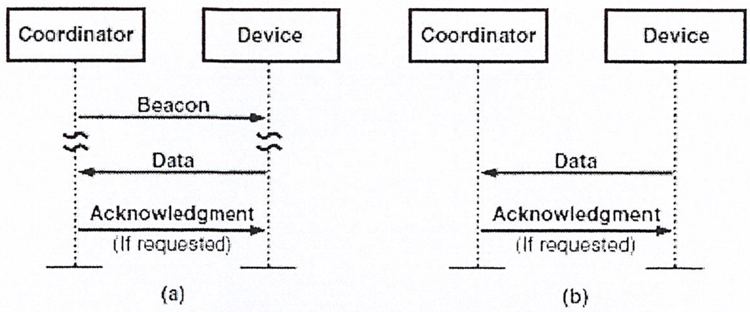
เมื่อตัดสินใจที่จะส่งสัญญาณออก อันดับแรกจะอยู่ในโหมดการรับ ซึ่งจะทำการวัดระดับความเข้มของสัญญาณในช่องสัญญาณที่ได้ทำการเลือก ซึ่งเรียกว่า Energy Detection (ED) ซึ่งใน ED นี้จะไม่สามารถตีความสัญญาณที่ตรวจจับได้ เพียงแต่สามารถประมาณค่าความเข้มของสัญญาณได้เท่านั้น ซึ่งถ้าพบสัญญาณใดๆ ในช่องสัญญาณที่สนใจ ED จะไม่สามารถตัดสินใจได้ว่าสัญญาณนี้เป็นสัญญาณมาตรฐานของ IEEE 802.15.4 หรือไม่

อีกวิธีในการตัดสินใจเปรียบเทียบช่องสัญญาณนั้นว่า “ว่าง” หรือ “ไม่ว่าง” จะเป็นหน้าที่ของ carrier sense (CS) ซึ่งใน CS จะแตกต่างกับ ED กล่าวคือ รูปแบบ CS จะสามารถตรวจสอบรูปแบบสัญญาณว่าเป็นสัญญาณมาตรฐานของ IEEE 802.15.4 ซึ่งจะทำให้อุปกรณ์สามารถตัดสินใจเลือกช่องสัญญาณเหล่านั้น จนกระทั่งความเข้มของสัญญาณอ่อนตัวลง จึงจะสามารถส่งข้อมูลออกได้

ถ้าหาก ช่องสัญญาณนั้นๆไม่ว่าง อุปกรณ์จะทำการรอช่วงเวลาหนึ่ง จากนั้นจึงจะทำการตรวจสอบอีกครั้ง และจะทำซ้ำในลักษณะแบบนี้จนถึงค่าที่ผู้ใช้กำหนดเอาไว้

2.8.2 เครือข่าย Beacon-Enabled และ Non-Beacon

วิธีการเข้าถึงช่องสัญญาณนั้น สามารถแบ่งออกเป็น 2 วิธีคือ contention based และ contention free ใน contention-based channel access นั้น ทุกอุปกรณ์ในเครือข่ายที่ต้องการส่งข้อมูลในช่องสัญญาณเดียวกัน จะใช้กลไกล CSMA-CA ที่ได้กล่าวไปข้างต้น อุปกรณ์ไหนที่สามารถพบช่องสัญญาณว่างก่อนจะได้ส่งข้อมูลก่อน ส่วนอีกประเภทหนึ่งคือ contention-free นั้น โดยวิธีนี้อุปกรณ์แม่เครือข่าย (PAN Coordinator) จะเป็นตัวมอบสิทธิพิเศษให้กับอุปกรณ์เจาะจงที่ต้องการส่งข้อมูล ซึ่งจะเรียกว่า guaranteed time slot (GTS) ดังนั้นอุปกรณ์ที่ใช้วิธีรับ GTS นี้จะส่งข้อมูลได้โดยปราศจากกลไกล CSMA-CA



รูปที่ 2.14 การส่งข้อมูลให้ Coordinator โดยวิธี (a) Beacon-Enabled [1] (b) Non-Beacon [1]

ในกรณี GTS นี้ อุปกรณ์แม่เครือข่ายจะต้องติดต่อกับลูกข่ายได้ทั้งหมด จากรูป 2.14(a) beacon คือข้อความเฉพาะที่แม่ข่ายจะส่งไปหาทุกๆจุดในเครือข่ายเพื่ออนุญาต GTS ให้กับบางจุดในเครือข่าย ข้อเสียของวิธีนี้คือ อุปกรณ์ทุกๆตัวจะต้องตื่นขึ้นมารับ beacon และถ้าไม่ได้รับการอนุญาต GTS ก็จะกลับเข้าสู่โหมด sleep เหมือนเดิม นั่นหมายความว่าจะมีอุปกรณ์ในเครือข่ายหลายๆตัวที่จะต้องตื่นขึ้นมา นอกเหนือจากการทำงานปกติของอุปกรณ์นั้นๆ ทำให้เครือข่ายแบบ Beacon-Enabled นั้นสิ้นเปลืองพลังงานมากกว่าเครือข่ายแบบ Non-Beacon

ในเครือข่ายที่แม่ข่ายไม่ต้องส่งข้อความ Beacon ให้กับลูกๆในเครือข่ายนั้น เรียกว่า Non-Beacon Network ซึ่งจะไม่มีการส่ง GTS และอุปกรณ์แต่ละตัวไม่จำเป็นที่จะต้องเชื่อมโยงถึงกันหมด ซึ่งการสิ้นเปลืองพลังงานจะน้อยกว่าเนื่องจากอุปกรณ์ที่เชื่อมต่อแบบ Non-Beacon ไม่ต้องตื่นขึ้นมาทำงานบ่อยครั้ง

2.8.3 ลักษณะการทำงานของ Self-Forming และ Self-Healing

ในระบบเครือข่ายแบบเมซ ยกตัวอย่าง อุปกรณ์ตัวแรกที่เป็น FFD เมื่อเริ่มต้นการทำงานสื่อสาร สามารถระบุตัวตนเป็นแม่ข่ายสำหรับเครือข่ายนั้นๆ (Zigbee Coordinator) และอุปกรณ์อื่นๆสามารถเข้ามาร่วมในเครือข่ายโดนการส่งการขอร้องการร่วมกลุ่มเครือข่าย การระบุตัวตนในลักษณะแบบนี้เรียกว่า self-forming networks

นอกจากนี้ในเครือข่ายแบบเมซ สามารถส่งข้อมูลได้หลายเส้นจากอุปกรณ์หนึ่งไปยังอีกอุปกรณ์หนึ่ง ซึ่งอาจจะมีบางจุดที่มีปัญหาไม่สามารถทำงานได้ แต่เครือข่ายยังสามารถส่งข้อมูลไปยังจุดหมายได้ เนื่องจากมีลักษณะการเลือกเส้นทางด้วยตนเอง เรียกวิธีการนี้ว่าคุณลักษณะ self-healing ของเครือข่ายแบบเมซ

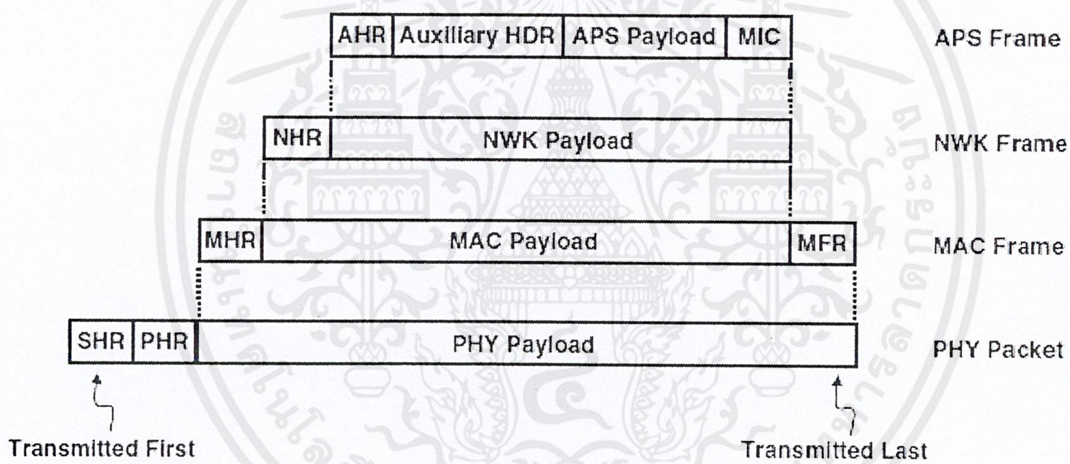
2.9 ฟังก์ชั่นการทำงานของ Zigbee และ IEEE 802.15.4 Networking Layer

2.9.1 Physical Layer

ในส่วนของ Physical Layer ในเครือข่าย Zigbee wireless sensor นั้นจัดอยู่ในมาตรฐาน IEEE 802.15.4 ซึ่งเป็นเลเยอร์ที่อยู่ต่ำที่สุด ซึ่งสามารถควบคุมโดยตรงกับตัวส่งสัญญาณคลื่นวิทยุ โดยที่ PHY Layer นั้น มีหน้าที่เปิดสัญญาณการส่งข้อมูลออก และ รับข้อมูลเข้า โดยการส่งหรือรับข้อมูลนั้น PHY Layer จะต้องมั่นใจว่าช่องสัญญาณที่ได้ทำการเลือกนั้น ไม่ถูกใช้โดยอุปกรณ์ในเครือข่ายอื่นๆอยู่

2.9.2 รูปแบบทั่วไปของโครงสร้าง Physical Packet

โครงสร้างของ Physical Packet นั้น สามารถแบ่งองค์ประกอบได้ 3 องค์ประกอบ ได้แก่ Synchronization header (SHR), PHY header (PHR) และ PHY payload โดยวิธีการส่งข้อมูลออกทางคลื่นวิทยุ จะเรียงตามลำดับที่กล่าวมาข้างต้น



รูปที่ 2.15 โครงสร้างของ Zigbee Packet [1]

Synchronization header (SHR) จะมีหน้าที่บอกอุปกรณ์ตัวรับให้อนุญาตการรับข้อมูล และ ล็อกข้อมูลที่ส่งต่อจาก SHR ทั้งหมดเอาไว้ ต่อจากนั้นจะเป็นหน้าที่ส่วนของ PHY header (PHR) ที่จะระบุความยาวขนาดของข้อมูลทั้งหมด ในส่วน PHY Payload นั้นจะขึ้นอยู่กับ เลเยอร์ส่วนบน ซึ่งจะรวมข้อมูลที่แท้จริงและคำสั่งต่างๆ สำหรับอุปกรณ์ผู้รับ

ในส่วนของข้อมูล MAC Frame ซึ่งจะถูกส่งผ่าน PHY Packet ในรูปแบบของ PHY Payload นั้น มี 3 องค์ประกอบเช่นกัน โดยจะมีส่วน MAC Header (MHR) จะรวบรวมข้อมูลซึ่งเป็น ที่อยู่และการรักษาความปลอดภัย ส่วน MAC Payload นั้น จะเก็บขนาดของข้อมูล, เก็บรหัสคำสั่งและข้อมูลต่างๆ และในส่วนสุดท้าย MAC Footer (MFR) จะบรรจุ ข้อมูลขนาด 16 bit สำหรับตรวจสอบลำดับการเรียงของข้อมูลที่อยู่ใน MAC Payload (16-bit Frame Check Sequence) หรือ FCS

จากรูป 2.15 ในส่วนของ Network Frame (NWK) สามารถแบ่งองค์ประกอบออกเป็น 2 องค์ประกอบ คือ NWK Header (NHR) และ NWK Payload ซึ่ง NWK Header นั้นจะเป็นที่เก็บ ที่อยู่ของเครือข่าย และ การควบคุม และ ในส่วนของ NWK Payload นั้นเป็นข้อมูลที่ส่งมาจาก Application Support Sublayer (APS Sublayer) ที่อยู่ในชั้นของ Application

APS Frame นั้น สามารถแบ่งองค์ประกอบได้ 4 องค์ประกอบ คือ APS Header (AHR) ซึ่งมีข้อมูลการควบคุมในชั้น Application Layer และ ข้อมูลที่อยู่ ส่วนต่อมาคือ Auxiliary Frame Header หรือ (Auxiliary HDR) จะเก็บกลไกลซึ่งใช้ในการรักษาความปลอดภัยของแต่ละ Frame และรหัสรักษาความปลอดภัย ซึ่งรหัสรักษาความปลอดภัยนี้จะต้องตรงกับอุปกรณ์ตัวรับเพื่อถอดรหัสออก ส่วนต่อมาคือ APS Payload จะรวบรวมข้อมูลและคำสั่งต่างๆ และส่วนสุดท้ายคือ Message Integrity Code (MIC) เป็นการรักษาความปลอดภัยในส่วนของ APS Frame

2.9.3 MAC Layer

MAC Layer หรือ Medium Access Control นั้น เป็นเลเยอร์ที่อยู่ระหว่าง PHY Layer และ NWK Layer โดยมีหน้าที่กำเนิด beacons และทำให้อุปกรณ์อื่นๆเข้าใจ beacons (โดยจะต้องเป็นเครือข่ายที่ใช้งาน beacon-enable เท่านั้น) และหน้าที่อื่นๆคือ ทำให้อุปกรณ์สามารถเชื่อมต่อ และ ไม่เชื่อมต่อกัน

2.9.4 รูปแบบทั่วไปโครงสร้าง MAC Frame

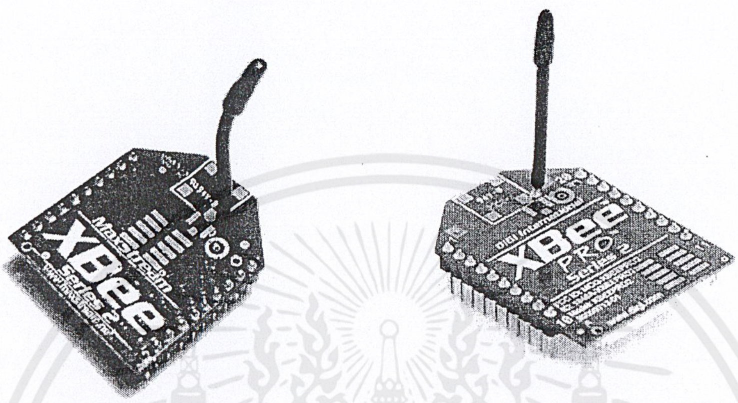
ในมาตรฐาน IEEE 802.15.4 นั้นระบุโครงสร้างของ MAC Frame ไว้ดังนี้คือ

1. Beacon Frame
2. Data Frame
3. Acknowledge Frame
4. MAC Command Frame

ใน Beacon Frame นั้นจะถูกใช้โดยอุปกรณ์เครือข่ายแม่ (Coordinator) โดยที่ beacons จะถูกใช้ในการส่งสัญญาณไปให้กับอุปกรณ์ทุกๆตัวในเครือข่าย ในส่วนของ Data Frame และ Acknowledge Frame นั้นใช้สำหรับส่งข้อมูลก็ต่อเมื่อได้รับสัญญาณ Acknowledge ตอบกลับมาจากอุปกรณ์ที่ต้องการส่งไปให้ ส่วนคำสั่งของ MAC นั้นจะถูกส่งออกโดยข้อมูลใน MAC Command Frame

2.10 โมดูล X-Bee และ X-Bee Pro

Zigbee เป็นชื่อของเครื่องหมายทางการค้า บริษัทที่ต้องการจะนำมาพัฒนาจะต้องซื้อลิขสิทธิ์จาก Zigbee Alliance ซึ่ง สามารถพัฒนาอุปกรณ์ Zigbee ให้ทำงานในความสามารถต่างๆเช่น FFD และ RFD ซึ่งในโครงงานนี้ ได้เลือกใช้อุปกรณ์ Zigbee ซึ่งผลิตโดยบริษัท Digi ซึ่งมีขายในประเทศไทย โดยเป็นอุปกรณ์ประเภท FFD หรือ เรียกอีกชื่อหนึ่งว่า X-Bee Series2 โดยจะมีกำลังส่ง 2 แบบคือ X-Bee Series2 และ X-Bee Pro Series2



รูปที่ 2.16 X-Bee (ซ้าย) และ X-Bee Pro (ขวา) [2]

X-Bee เป็นอุปกรณ์ที่มี ไมโครคอนโทรลเลอร์ และ RF IC อยู่ภายใน ทำหน้าที่เป็น อุปกรณ์ Transceiver (อุปกรณ์รับ-ส่งสัญญาณ) แบบ Half Duplex ซึ่งใช้ความถี่ย่าน 2.4 GHz มีการจัดการ โดยใช้พลังงานต่ำ ใช้งานง่าย มี interface ที่ใช้รับและส่งข้อมูลกับ X-Bee เป็น UART (TTL) ซึ่งสำหรับทางด้านไมโครคอนโทรลเลอร์ โดยนำขาที่ใช้ติดต่อสื่อสาร UART ของ X-Bee ต่อเข้ากับ UART ของ ไมโครคอนโทรลเลอร์

X-Bee สามารถใช้งานตามมาตรฐาน Zigbee ได้ โดยที่ไม่ต้องเขียนโปรแกรมสร้าง เครือข่าย Zigbee เนื่องจากทางผู้ผลิตได้จัดทำ firmware ที่จะเข้าไปโปรแกรมในตัว X-Bee ทำให้เราสามารถโปรแกรมตัวแปรต่างๆ ผ่าน Software Interface (X-CTU หรือโปรแกรมที่เขียนขึ้นเอง), ผ่านทาง AT Commands โดยใช้โปรแกรม Hyperterminal ต่างๆในการรับส่งข้อมูล ระหว่าง Serial Port หรือ ผ่านทางการรับส่งข้อมูลด้วย ไมโครคอนโทรลเลอร์ โดยเมื่อ set X-Bee ให้ทำงานเป็นอุปกรณ์ในเครือข่าย Zigbee แล้ว เราจะเรียก X-Bee แต่ละตัวว่าเป็น Node

คุณสมบัติของ X-Bee และ X-Bee Pro

1. Operating Frequency ISM Band 2.4 GHz (ISM Band หมายถึง ย่านความถี่ใช้งานเพื่อการวิจัย ซึ่งจะอนุญาตให้ใช้กับ อุตสาหกรรม (Industrial) วิทยาศาสตร์ (Scientific) และ ทางการแพทย์ (Medical) รวมเป็น ISM)

2. มีสายอากาศให้เลือกใช้หลายแบบ คือ แบบ Chip Ant , Whip Ant , UFL con , RPSMA con โดย 2 แบบหลัง ต้องใช้เสาอากาศย่าน 2.4 GHz ที่เป็น connector แบบ UFL หรือ SMA ในการต่อใช้งาน

3. ใช้ Supply Voltage 2.8-3.4 V

4. Power Down Current น้อยกว่า 10uA

5. มี RF Data Rate อยู่ที่ 250 Kbps (เป็นส่วนของ สัญญาณที่ส่งผ่านอากาศ)

6. มี Serial Interface Data Rate อยู่ระหว่าง 1200 – 115200 Bps (เป็นส่วนที่ติดต่อสื่อสารกับไมโครคอนโทรเลอร์)

7. ใช้ Spread Spectrum ชนิด DSSS (Direct Sequence)

8. ลักษณะการกำหนด Addressing คือ กำหนด PAN ID สำหรับเครือข่ายหนึ่งๆ, กำหนด Channel และกำหนด Address ของแต่ละตัว

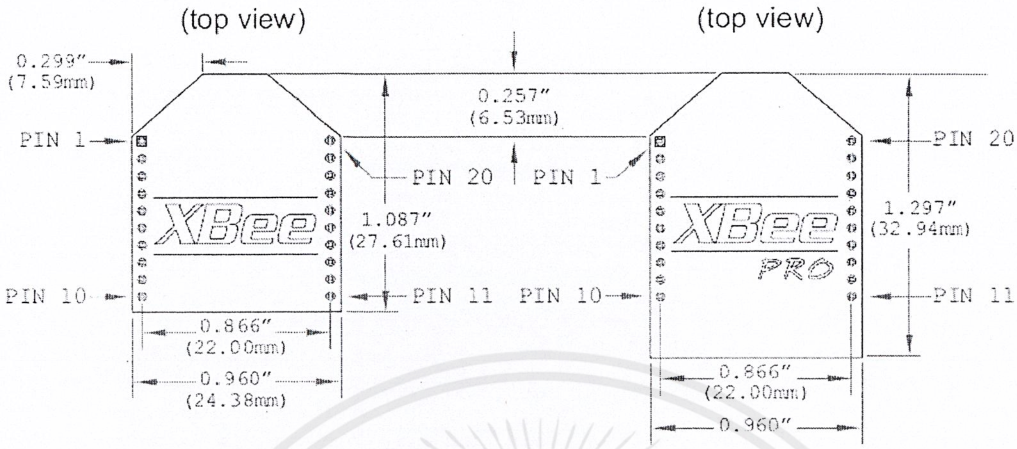
X-Bee และ X-Bee Pro จะมีอยู่ 2 รุ่นคือ รุ่น series1 (รุ่น IEEE802.15.4) มีลักษณะเป็น reduced-function devices (RFDs) และ รุ่น series 2 (รุ่น ZNET2.5 / รุ่น ZB) มีลักษณะเป็น full-function devices (FFDs) และยังมีขนาด Power ให้เลือกอีก 2 แบบ คือ แบบธรรมดา (1 mw – 2 mw) และ แบบ PRO (50 mw – 60 mw) ซึ่งจะมีผลเรื่องระยะทางการรับส่งข้อมูล โดยแต่ละ series นั้น สามารถสร้างเครือข่ายได้หลายแบบ แต่จะมีเพียง series 2 เท่านั้นที่จะทำเครือข่ายแบบ mesh ได้

2.10.1 ตารางคุณสมบัติ X-Bee และ X-Bee Pro

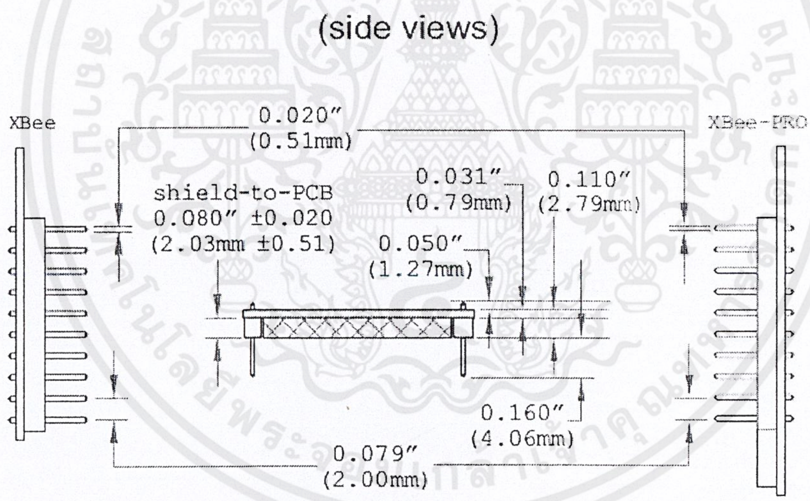
Feature	**X-Bee Series2**	**X-Bee Series2 Pro**
Power Input	2.8-3.3V @ 40mA	2.8-3.3V @ 295mA
Max data rate (Air)	250kbps	250kbps
Power Output	2mW output (+3dBm)	50mW output (+17dBm)
Distance	400ft (120m) range	1 mile (1600m) range
Antenna	Wire, Chip, UFL, SMA	Wire, Chip, UFL, SMA
Peripheral	6 10-bit ADC input pins 8 digital I/O pins	6 10-bit ADC input pins 8 digital I/O pins
Upgrade Firmware	Local(ZNET2.5) or over-air configuration(ZB)	Local(ZNET2.5) or over-air configuration(ZB)
Network	Point to point / multi-point / Mesh Network	Point to point / multi-point / Mesh Network

ตาราง 2.2 เปรียบเทียบคุณสมบัติ X-Bee และ X-Bee Pro [3]

2.10.2 ลักษณะทางกายภาพของ X-Bee และ X-Bee Pro

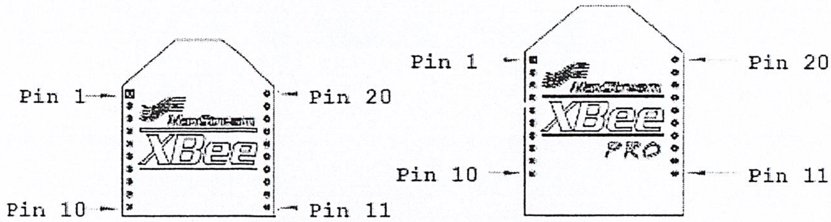


รูปที่ 2.17 ลักษณะทางกายภาพของโมดูลด้านบนของ X-Bee (ซ้าย) และ X-Bee Pro (ขวา) [3]



รูปที่ 2.18 ลักษณะทางกายภาพของโมดูล X-Bee และ X-Bee Pro ด้านข้าง [3]

2.10.3 การกำหนด PIN ของโมดูล X-Bee



รูปที่ 2.19 การเรียงลำดับขาของโมดูล X-Bee [3]

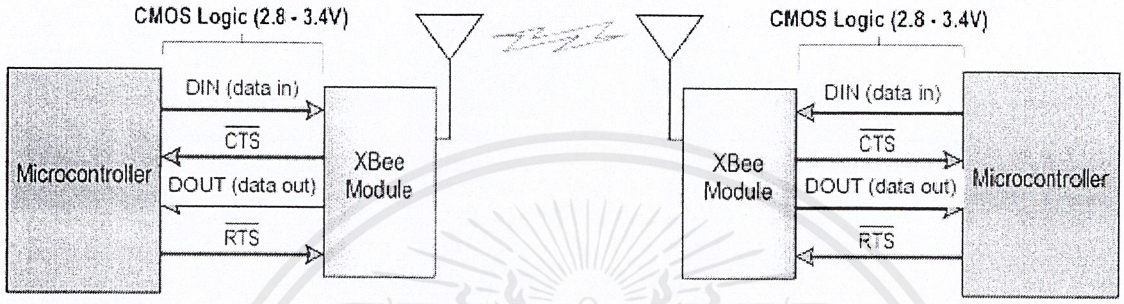
ขาที่	ชื่อขา	ทิศทาง	การทำงาน
1	VCC	-	ขาต่อไฟเลี้ยง 2.8-3.3V
2	DOUT	Output	ส่งข้อมูลอนุกรม (UART Data Out)
3	DIN/CONFIG	Input	รับข้อมูลอนุกรม (UART Data In)
4	DO8	Output	Digital Output 8
5	RESET	Input	ขารีเซตหลักของโมดูล
6	PWM0/RSSI	Output	PWM เอาท์พุต 0 / ความแรงสัญญาณภาครับ
7	PWM1	Output	PWM เอาท์พุต 1
8	Reserved	-	
9	DTR/SLEEP_RQ/DI8	Input	ควบคุมโหมด SLEEP / Digital Output 8
10	GND	-	กราวด์
11	AD4/DIO4	Either	อนาล๊อกอินพุต 4 / ดิจิตอล I/O 4
12	CTS/DIO7	Either	Clear-to-send / ดิจิตอล I/O 7
13	ON/SLEEP	Output	ตรวจสอบสถานะของโมดูล
14	VREF	Input	ความต่างศักย์อ้างอิงสำหรับ A/D อินพุต
15	Associated/AD5/DIO5	Either	อนาล๊อกอินพุต 5 / ดิจิตอล I/O 5
16	RTS/AD6/DIO6	Either	Request-to-send / อนาล๊อกอินพุต 6 / ดิจิตอล I/O 6
17	AD3/DIO3	Either	อนาล๊อกอินพุต 3 / ดิจิตอล I/O 3
18	AD2/DIO2	Either	อนาล๊อกอินพุต 2 / ดิจิตอล I/O 2
19	AD1/DIO1	Either	อนาล๊อกอินพุต 1 / ดิจิตอล I/O 1
20	AD0/DIO0	Either	อนาล๊อกอินพุต 0 / ดิจิตอล I/O 0

ตาราง 2.3 การทำงานของขาโมดูล X-Bee

2.10.4 การติดต่อสื่อสารของ X-Bee ผ่านพอร์ตอนุกรม

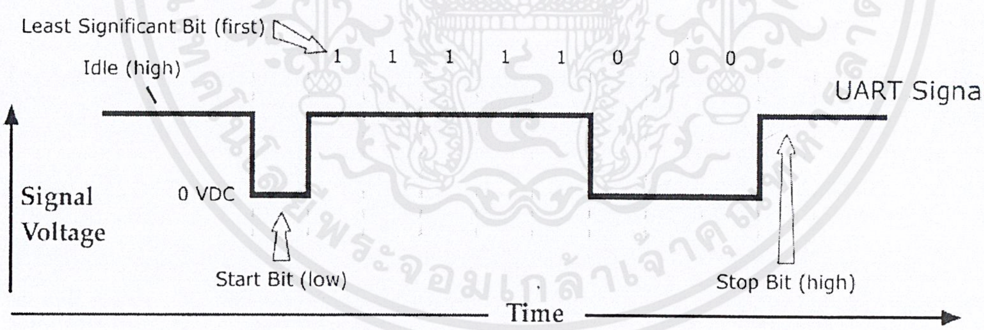
2.10.4.1 ทิศทางการไหลของข้อมูล UART

จากรูป 2.20 โมดูล X-Bee อินเตอร์เฟสกับอุปกรณ์ผู้ใช้ (Host) ผ่านทางพอร์ตอนุกรมอะซิงโครนัสระดับตรรกะ ส่งผ่านโมดูลทางพอร์ตอนุกรม โมดูลสามารถติดต่อสื่อสารกับตรรกะต่างๆเข้ากับ UART โดยอุปกรณ์มีอินเตอร์เฟส UART สามารถติดต่อโดยตรงกับขาของโมดูล X-Bee ดังรูป



รูปที่ 2.20 ระบบการไหลของข้อมูลในอุปกรณ์ที่เชื่อมต่อผ่าน UART [3]

ข้อมูลส่งขาเข้า UART โมดูล ผ่านขา Data In (ขา 3) ด้วยสัญญาณอนุกรมอะซิงโครนัส สัญญาณจะว่างเมื่อไม่มีข้อมูลถูกส่ง แต่ละข้อมูลไบต์ประกอบด้วย บิตเริ่มต้น (บิตต่ำ) 8 ข้อมูลบิต และบิตหยุด (บิตสูง) ตามรูป 2.21 อธิบายแผนบิตอนุกรมของการผ่านข้อมูลของโมดูล



รูปที่ 2.21 Packet ข้อมูล UART 0x1F ถูกส่งผ่านโมดูล X-Bee [3]

รูปแบบงานโมดูล UART ดังนั้นเวลาและบิทพาริตี้ ตรวจสอบต้องการข้อมูลติดต่อสื่อสาร การสื่อสารอนุกรมขึ้นอยู่กับ 2 UARTS ถูกจัดโครงสร้างกับการจัดตั้ง (อัตราบอด, บิทพาริตี้, บิตเริ่มต้น, บิตหยุด, บิตข้อมูล)

2.10.4.2 การส่งข้อมูลของ X-Bee

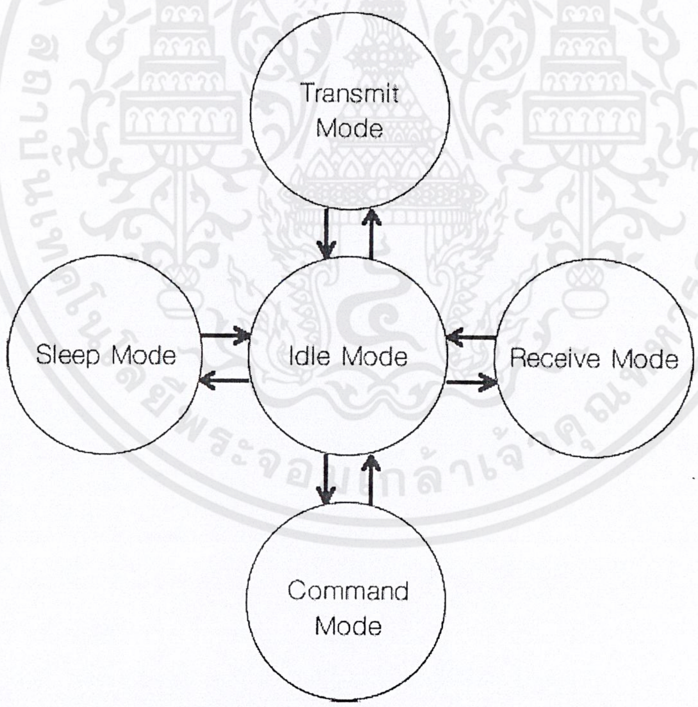
การส่งข้อมูลแบบ RF ของแต่ละแพ็กเกจในส่วนของ header จะประกอบไปด้วย Source Address และ Destination Address โดยที่ IEEE 802.15.4 จะมีโครงสร้าง 2 แบบ นั่นคือแบบ short 16-bit addresses และแบบ long 64-bit addresses ซึ่ง64-bit จะสามารถอ่านคำสั่ง SL (Serial Number Low) และ SH (Serial Number High) และการส่งข้อมูลแบบ RF จะส่งได้ 2 โหมด คือ Unicast Mode และ Broadcast Mode

- การส่งแพ็คเกจโดยใช้โครงสร้าง 16-bit addressing ให้ตั้งค่า ตัวแปร DL (Destination Address Low) ให้เท่ากับ ตัวแปร MY และตั้งค่าตัวแปร DH (Destination Address High) เป็น ‘0’

- การส่งแพ็คเกจโดยใช้โครงสร้าง 64-bit addressing ให้ตั้งค่า Destination Address (DL + DH) ให้ตรงกับ Source Address (SL + SH) ของปลายทางที่เราจะส่งแพ็คเกจไป

2.10.5 โหมดการทำงานของโมดูล X-Bee

การทำงานของ X-Bee สามารถแบ่งออกได้เป็น 5 โหมดการทำงาน ดังรูป 2.22 ดังต่อไปนี้



รูปที่ 2.22 โหมดต่างๆของการทำงานโมดูล X-Bee

- ***Idle Mode*** เป็นโหมดที่ไม่มีการรับส่งข้อมูล และ เป็นโหมดกลางที่สามารถเปลี่ยนไปยังโหมดต่างๆได้
- ***Transmit Mode*** มีการส่งข้อมูลได้สองวิธี

1. Direct Transmission ข้อมูลทั้งหมดจะถูกส่งไปยัง Destination Address ทันที

2. Indirect Transmission Packet ของข้อมูลจะถูกเก็บไว้จนกว่าจะถึงเวลาส่งเท่านั้น และจะส่งไปยังที่มีการตอบรับมา (Source Address = Destination Address)

- ***Receive Mode*** ข้อมูลคลื่นวิทยุจะถูกรับทางสายอากาศ

- ***Sleep Mode RF*** อยู่ในสถานะที่มีการใช้กำลังไฟฟ้าต่ำหรือไม่มีการใช้ การเข้ามาอยู่ในโหมดนี้นั้นจะต้องเป็นไปตามเงื่อนไขต่อไปนี้อย่างใดอย่างหนึ่ง คือ

1. มีการใช้งานที่ Sleep_RQ (Pin 9)

2. อยู่ในโหมด Idle โดยไม่มีการรับส่งข้อมูลเป็นเวลานานมากกว่าที่กำหนดไว้ที่ตัวแปร ST (Time Before Sleep)

- ***Command Mode*** เป็นโหมดคำสั่งโดยจะเป็นโหมดที่โมดูล X-bee ให้สำคัญสูงสุด

2.11 บทสรุป

ในบทที่ผ่านมาได้อธิบายถึงโครงสร้างภายในของ Zigbee Stack ซึ่งได้อธิบายข้อมูลทั่วไปของ Zigbee, การเปรียบเทียบ Zigbee กับเครือข่าย Wireless Network อื่นๆ ซึ่งเป็นไปตามมาตรฐาน IEEE 802.11, อธิบายประเภทของเครือข่ายระยะสั้น ทำให้เข้าใจประเภทของเครือข่ายของ Zigbee มากยิ่งขึ้น, ความถี่ที่ใช้งานและความเร็วในการส่งข้อมูล นอกจากนี้ยังได้อธิบายรูปแบบเครือข่ายต่างๆของ Zigbee ซึ่งสามารถแบ่งได้ 3 ประเภทคือ เครือข่ายแบบสตาร์ เครือข่ายแบบต้นไม้ และเครือข่ายแบบเมฆ สุดท้ายได้กล่าวถึง Zigbee โปรโตคอลสแตก ทั้งหมด ทำให้ทราบข้อมูลเบื้องต้นของ Zigbee และสามารถนำความรู้ที่ได้ศึกษานำมาประยุกต์ใช้กับโครงงานที่ออกแบบไว้

บทที่ 3

การออกแบบและทดสอบโครงงานเบื้องต้น

3.1 ภาพรวมของโครงงานเบื้องต้น

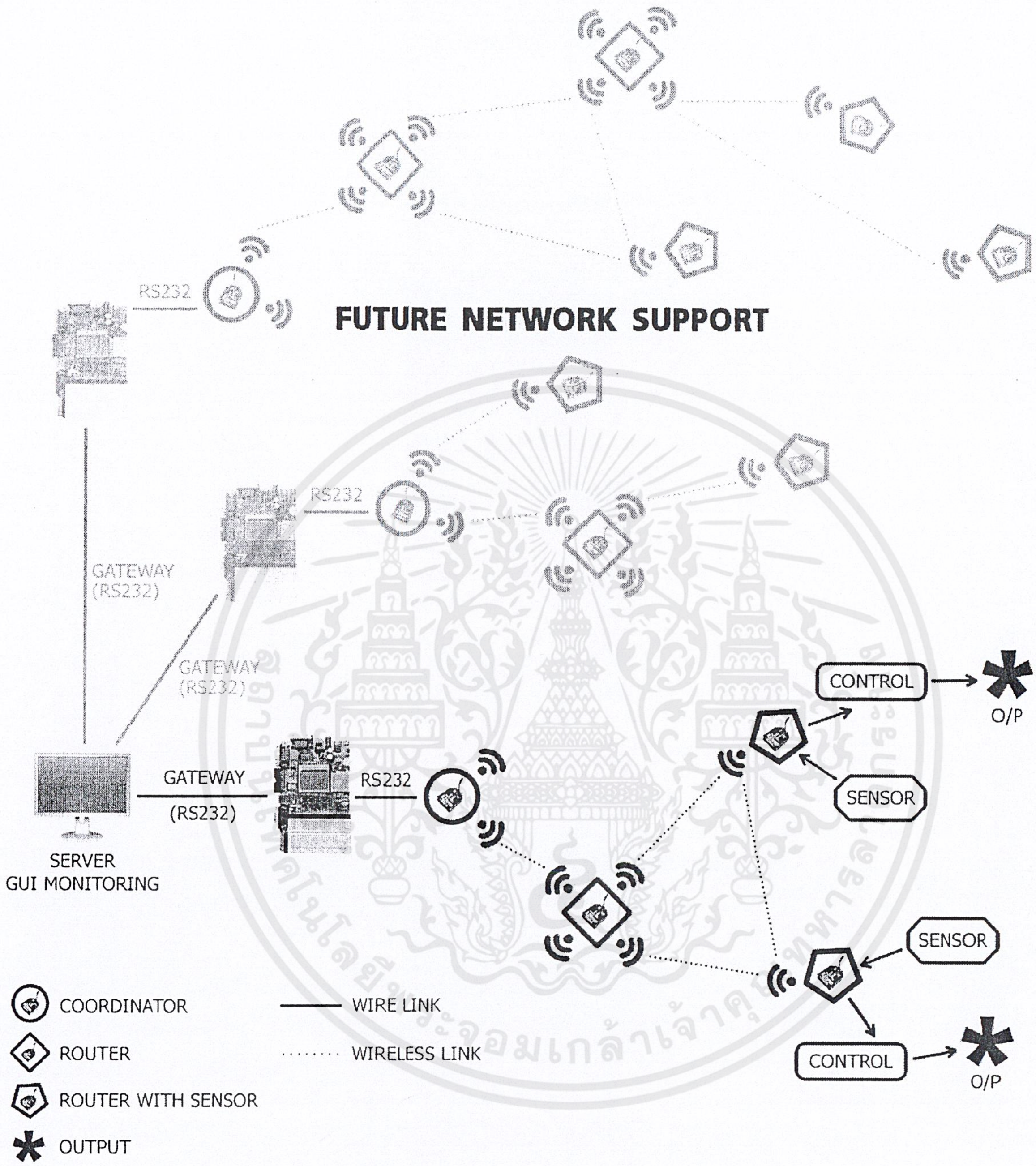
ภาพรวมของโครงงานที่ได้ออกแบบไว้เบื้องต้น สามารถแสดงดังรูป 3.1 จะเห็นว่าระบบโดยรวมที่สมบูรณ์ ประกอบด้วยเครือข่าย Zigbee ย่อยหลายๆเครือข่าย แต่ละเครือข่ายจะมีเซ็นเซอร์ต่างๆ กระจายอยู่เป็นบริเวณกว้าง โดยแต่ละเครือข่ายนั้น จะรับ-ส่งข้อมูล ผ่านเครือข่ายโดยใช้โมดูล X-bee ซึ่งทำหน้าที่เป็นทั้ง FFD และ RFD จากนั้นข้อมูลจากเซนเซอร์ในแต่ละเครือข่ายจะถูกประมวลผลและรวบรวมด้วย ARM ซึ่งทำหน้าที่เป็น gateway เชื่อมต่อกับเซิร์ฟเวอร์ผ่านทางระบบ LAN (ซึ่งในโครงงานนี้ได้เลือกใช้ RS-232 แทน) โดย เซิร์ฟเวอร์นั้นจะมีโปรแกรมเฝ้าระวังและควบคุมซึ่งมี GUI (Graphic User Interface) คอยทำหน้าที่แสดงผลภาพรวมของระบบเมื่อตรวจพบความผิดพลาด หรือ ค่าที่ได้จากเซ็นเซอร์ตามจุดนั้นๆของข้อมูลที่ถูกส่งมา ต่ำกว่า หรือ สูงกว่า ค่าที่ได้ตั้งเอาไว้ เซิร์ฟเวอร์จะสั่งการทำงานควบคุมกลับไปยังจุดที่โมดูล X-bee ส่งมา เพื่อนำไปควบคุมระบบให้เกิดสภาวะเหมาะสมที่ได้ตั้งค่าเอาไว้ ทำให้ผู้ใช้งานสามารถใช้งานระบบได้โดยง่าย

จากรูป 3.1 จะเห็นว่า ระบบใหญ่ที่ทำหน้าที่ดูภาพรวมของระบบนั้น ยังสามารถรองรับระบบที่จะเกิดขึ้นในอนาคตได้อีกหลายระบบ ซึ่งตามมาตรฐานของ Zigbee นั้น ในแต่ละระบบสามารถรองรับเครือข่าย Zigbee ได้มากถึง 64,000 จุด

การทดลองในภาคการศึกษาแรกนี้เป็นการทดลองทางด้าน การใช้งานโมดูล X-Bee ร่วมกับไมโครคอนโทรลเลอร์ ARM7 Cortex-M3 ซึ่งเป็น การตั้งค่าโมดูล X-Bee สำหรับการใช้งาน, การทดสอบใช้งานโมดูล X-Bee รูปแบบการเชื่อมต่อโครงข่ายในแบบต่างๆ, การทดสอบระยะทางในการส่งข้อมูลไร้สาย, การอ่านข้อมูลทาง Packet ของ Zigbee, การประยุกต์ใช้งานพอร์ท A/D และ Digital I/O ที่อยู่ในโมดูล X-Bee นำมาใช้งาน, การพัฒนาโปรแกรมใน ARM7 Cortex-M3 เพื่อตีความ Packet ของข้อมูลที่ได้รับและแสดงผลออกทางจอ GLCD5110

การนำโมดูล X-Bee มาใช้งานนั้น ผู้พัฒนาจะต้องทำการโปรแกรมค่าต่างๆลงในโมดูล X-Bee ซึ่งค่าต่างๆเหล่านี้จะเป็นตัวกำหนดความสามารถที่โมดูล X-Bee นั้นจะทำได้เช่น ทำงานเป็นแม่ข่ายของระบบ (Coordinator), เราท์เตอร์ (Router) หรือ ลูกข่ายของระบบ (End Device) ก็ตาม

โดยก่อนการโปรแกรมค่าต่างๆเหล่านั้นลงในโมดูล X-Bee ได้ จะต้องสร้างวงจรสำหรับ โปรแกรมค่าต่างๆสำหรับโมดูล X-Bee ซึ่งใช้การส่งข้อมูลทางพอร์ตอนุกรม RS-232 ผ่านทางคอมพิวเตอร์ โดยจะต้องใช้โปรแกรมที่ทางผู้ผลิตพัฒนาไว้ สำหรับการตั้งค่าต่างๆ โดยบริษัท Digi ได้พัฒนาซอฟท์แวร์ X-CTU ไว้สำหรับผู้พัฒนาโมดูล X-Bee ในการตั้งค่าต่างๆลงสู่โมดูล



รูปที่ 3.1 ภาพรวมของระบบที่ออกแบบไว้

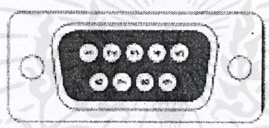
3.1.1 การตั้งค่าต่างๆสำหรับโมดูล X-Bee

3.1.1.1 พอร์ตอนุกรม RS-232

การสื่อสารแบบอนุกรม นับว่ามีความสำคัญ ต่อการใช้งาน ไมโครคอนโทรลเลอร์มาก เพราะสามารถใช้แป้นพิมพ์ และจอภาพของ PC เป็น อินพุต และ เอาต์พุต ในการติดต่อ หรือ ควบคุม ไมโครคอนโทรลเลอร์ ด้วยสัญญาณอย่างน้อย เพียง 3 เส้นเท่านั้น คือ

1. สายส่งสัญญาณ TxD
2. สายส่งสัญญาณ RxD
3. สาย Ground

ซึ่งโดยปกติพอร์ตอนุกรม RS-232 จะสามารถต่อสายได้ยาว 50 ฟุตโดยประมาณ ขึ้นอยู่กับ ชนิดของสายสัญญาณ, ระยะทาง และ ปริมาณสัญญาณรบกวน



รูปที่ 3.2 คอนเน็กเตอร์แบบ DB9 (ตัวผู้) เมื่อมองจากด้านหลัง[5]

Pin	Description	Type
1	Data Carrier Detect (DCD)	Input
2	Received Data (RXD)	Input
3	Transmitted Data (TXD)	Output
4	Data Terminal Ready (DTR)	Output
5	Signal Ground (GND)	Input
6	Data Set Ready (DSR)	Input
7	Request To Send (RTS)	Output
8	Clear to Send (CTS)	Input
9	Ring Indicator (RI)	Input

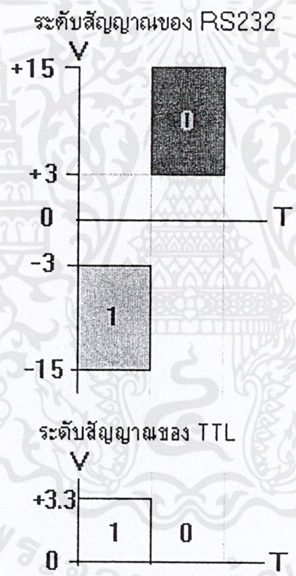
ตารางที่ 3.1 การจัดขาของ DB9 และหน้าที่ต่างๆในการใช้งาน

การทำงานของขา DB9

1. TxD เป็นขาที่ใช้ในการส่งข้อมูลออก
2. RxD เป็นขาที่ใช้ในการรับข้อมูลเข้า

3. Ground เป็นขาที่ใช้ในการอ้างอิงระดับสัญญาณกราวนด์ร่วม
4. DTR แสดงสภาวะพอร์ตว่าเปิดใช้งาน, DSR ตรวจสอบว่าพอร์ต ที่ติดต่อด้วย เปิดอยู่หรือไม่
   - เมื่อเปิดพอร์ตอนุกรม ขา DTR จะ ON เพื่อให้อุปกรณ์ได้รับทราบว่าต้องการติดต่อด้วย
   - ในขณะเดียวกันก็จะตรวจสอบขา DSR ว่าอุปกรณ์พร้อมหรือไม่
5. RTS แสดงสภาวะพอร์ตว่าต้องการส่งข้อมูล, CTS ตรวจสอบว่าพอร์ตที่ติดต่ออยู่ ต้องการส่งข้อมูลหรือไม่
   - เมื่อต้องการส่งข้อมูลขา RTS จะ ON และจะส่งข้อมูลออกที่ขา TXD เมื่อส่งเสร็จก็จะ OFF
   - ในขณะเดียวกันก็จะตรวจสอบขา CTS ว่าอุปกรณ์ต้องการที่จะส่งข้อมูลหรือไม่

ระดับสัญญาณของ RS-232 และ TTL



รูปที่ 3.3 ระดับสัญญาณ RS-232 (บน) และ TTL (ล่าง) [5]

จากรูป 3.3 สัญญาณรบกวนที่เกิดขึ้น ในสายนำสัญญาณ มักจะมีแรงดันเป็นบวก เมื่อเทียบกับกราวนด์

- เพื่อป้องกันสัญญาณรบกวนนี้ จึงออกแบบแรงดัน ของโลจิก "1" เป็นลบ คืออยู่ในช่วง -3V ถึง -15V ส่วนแรงดัน ของโลจิก "0" อยู่ในช่วง +3V ถึง +15V

- เหตุที่ระดับสัญญาณของ RS232 อยู่ในช่วง +15V ถึง -15V ก็เพื่อให้ต่อสายสัญญาณไปได้ไกลขึ้นดังนั้น จึงจำเป็นจะต้องมีวงจรเปลี่ยนระดับแรงดันของ RS232 มาเป็นระดับแรงดันของ TTL

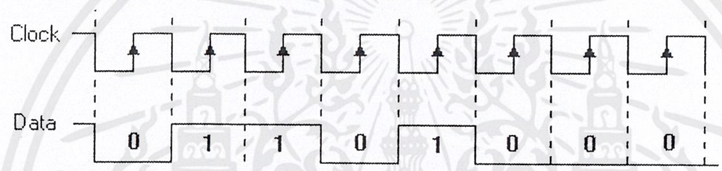
อัตราการส่งข้อมูล (Baud rate)

คือความเร็วของการรับ-ส่งข้อมูล เป็นจำนวนบิตต่อวินาที เช่น 300, 1200, 2400, 4800 , 9600 ,14400 , 19200, 38400, 56000 เป็นต้น ซึ่งการเลือกอัตราการส่งข้อมูลขึ้นอยู่กับ ชนิดของสายสัญญาณ, ระยะทาง และปริมาณสัญญาณรบกวน

รูปแบบการสื่อสารแบบอนุกรม

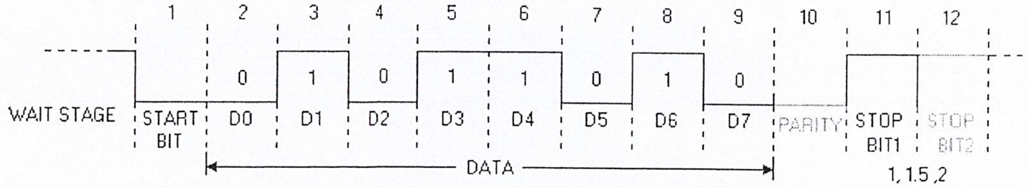
มีด้วยกันอยู่ 2 แบบ คือแบบซิงโครนัส (Synchronous) และ แบบอะซิงโครนัส (Asynchronous)

- การสื่อสารแบบซิงโครนัส (Synchronous) เมื่อมีการรับส่งข้อมูล จะมีสัญญาณนาฬิกา ซึ่งเป็นตัวกำหนด จังหวะเวลา การส่งข้อมูล ร่วมอยู่ด้วยอีกเส้นหนึ่ง ใช้คู่กับสัญญาณข้อมูล ตัวอย่างเช่น การส่งสัญญาณจากคีย์บอร์ด



รูปที่ 3.4 การส่งข้อมูลแบบซิงโครนัส [5]

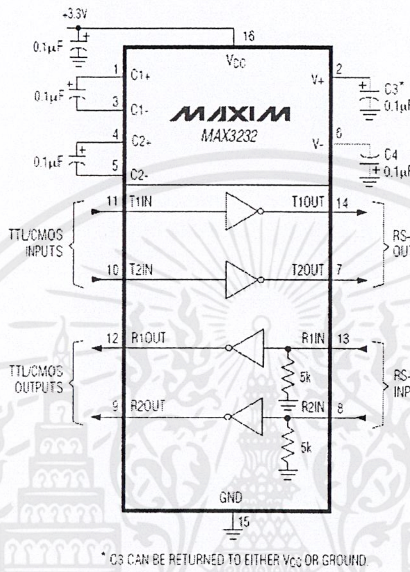
- การสื่อสารแบบอะซิงโครนัส (Asynchronous) เป็นการรับส่งข้อมูล โดยที่ไม่จำเป็นต้องมีสัญญาณนาฬิกา ร่วมด้วย แต่จะใช้ให้ตัวส่ง และตัวรับ มีอัตราส่งข้อมูล ที่เท่ากัน โดยรูปแบบข้อมูลแบบอะซิงโครนัส ประกอบด้วย 4 ส่วนคือ
  1. บิตเริ่มต้น (Start bit) มีขนาด 1 บิต
  2. บิตข้อมูล (Data) มีขนาด 5,6,7 หรือ 8 บิต
  3. บิตตรวจสอบพาริตี้ (Parity bit) มีขนาด 1 บิตหรือไม่มี
  4. บิตหยุด (Stop bit) มีขนาด 1, 1.5, 2 บิต



รูปที่ 3.5 การส่งข้อมูลแบบอะซิงโครนัส [5]

3.1.1.2 ไอซีแปลงระดับสัญญาณ RS232

การออกแบบชิ้นงานนั้น วงจรในส่วนแรกจะเป็นการออกแบบวงจรสำหรับตั้งค่าต่างๆที่จำเป็นในการใช้งานโมดูล X-Bee โดยวงจรนี้จะใช้สำหรับโปรแกรมโมดูล X-Bee ผ่านพอร์ทอนุกรมของคอมพิวเตอร์ ซึ่งใช้ไอซี MAX3232 (จากรูป 3.6) ในการแปลงระดับสัญญาณของ RS-232 มาเป็นระดับ TTL และในทำนองเดียวกัน ไอซี MAX3232 ยังแปลงระดับสัญญาณ TTL ไปเป็นระดับสัญญาณ RS-232

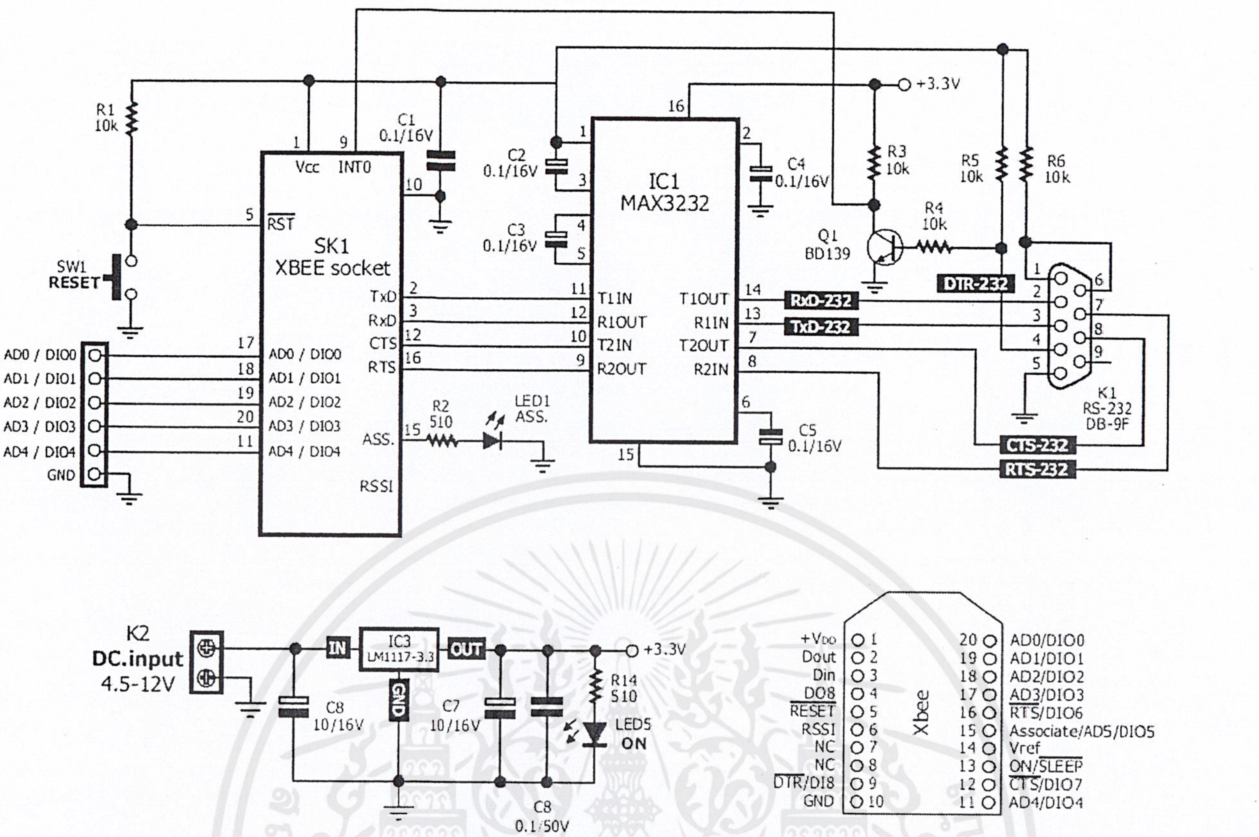


รูปที่ 3.6 วงจรแปลงระดับสัญญาณ TTL เป็น RS-232 และ RS-232 เป็น TTL [4]

เนื่องจาก โมดูล X-Bee ทำงานไฟเลี้ยงย่าน 3.3V จึงต้องเลือกวงจรแปลงระดับสัญญาณในการโปรแกรมค่าต่างๆที่จำเป็นลงในโมดูล X-Bee ผ่านไอซีที่ทำงานด้วยไฟเลี้ยงย่านเดียวกันคือ MAX3232

3.1.1.3 วงจรการตั้งค่าโมดูล X-Bee

ในการตั้งค่าโมดูล X-Bee นั้นจะต้องใช้ขาที่จำเป็นในการสื่อสารของ X-Bee ดังนี้คือ Data In, Data Out, CTS, RTS เป็นต้น ซึ่งจะต้องทำการออกแบบวงจรสำหรับโปรแกรม X-Bee ตามรูปที่ 3.7



รูปที่ 3.7 วงจรโปรแกรมโมดูล X-Bee

3.1.1.4 ตัวแปรที่จำเป็นในการตั้งค่า

ในการตั้งค่าโมดูล X-Bee นั้นจะต้องตั้งค่าผ่านซอฟท์แวร์จากผู้ผลิต คือ X-CTU ซึ่งจะสอดคล้องกับ โมดูล X-Bee ที่ใช้ เนื่องจาก X-Bee ที่ใช้เป็นรุ่น X-Bee Series2 ซึ่งสามารถ โปรแกรมทั้งผ่านอากาศ และ ผ่านทางพอร์ตอนุกรมได้ ดังนั้นจึงต้องเข้าใจการตั้งค่าตัวแปรต่างๆให้กับ X-Bee ในโหมดต่างๆได้ดังนี้

ฟังก์ชั่นที่ตั้งค่า	หน้าที่หลัก	หน้าที่ในเครือข่าย
ZNET 2.5 COORDINATOR API	ตอบสนองการสื่อสารด้วย API Command	Coordinator
ZNET 2.5 COORDINATOR AT	ตอบสนองการสื่อสารด้วย AT Command	Coordinator
ZNET 2.5 ROUTER/END DEVICE API	ตอบสนองการสื่อสารด้วย API Command ในการเลือกใช้งาน ROUTER/END DEVICE นั้น กำหนดได้จาก SLEEP TIME โดย ROUTER	Router / End Device

	ไม่สามารถใช้งาน SLEEP TIME ได้	
ZNET 2.5 ROUTER/END DEVICE AT	ตอบสนองการสื่อสารด้วย AT Command ในการเลือกใช้งาน ROUTER/END DEVICE นั้น กำหนดได้จาก SLEEP TIME โดย ROUTER ไม่สามารถใช้งาน SLEEP TIME ได้	Router / End Device

ตาราง 3.2 การตั้งค่าฟังก์ชั่นการทำงานของโมดูล X-Bee

นอกจากนั้น ยังมีตัวแปรจำเป็นที่จะต้องตั้งค่า ซึ่งในโครงงานนี้เน้นประสิทธิภาพสูงที่สุดของระบบ ซึ่งในการทดลองเบื้องต้นจะใช้ X-Bee ทั้งหมด 4 จุด โดยกำหนด Coordinator 1 จุด และเป็น Router อีก 3 จุด เพื่อทำการส่งข้อมูลในเครือข่ายแบบ Mesh Topology ซึ่งในการกำหนดตัวแปรต่างๆสำหรับแต่ละโมดูล มีดังนี้

หน้าที่ในเครือข่าย	ฟังก์ชั่นที่ใช้งาน	ตัวแปรที่จำเป็นในการตั้งค่า
Coordinator	ZNET 2.5 COORDINATOR API	ID – PAN ID : 100 DH – Destination High : 0 DL – Destination Low : FFFF NI – Node Identifier : COORDINATOR BH – Broadcast Radius : 0 EE – Encryption Enable : 1 KY – AES Encrypt Key : A1B2C3D4E5F6 BD – Baud Rate : 3 NB – Parity : 0 AP – API Enable : 1
Router Node 1	ZNET 2.5 ROUTER AT	ID – PAN ID : 100 DH – Destination High : 0 DL – Destination Low : 0 NI – Node Identifier : ROUTER_1 BH – Broadcast Radius : 0 EE – Encryption Enable : 1 KY – AES Encrypt Key : A1B2C3D4E5F6 BD – Baud Rate : 3

		NB – Parity : 0 D0 – DIO : 3 IC – Digital IO Change Detection : FF
Router Node 2	ZNET 2.5 ROUTER AT	ID – PAN ID : 100 DH – Destination High : 0 DL – Destination Low : 0 NI – Node Identifier : ROUTER_2 BH – Broadcast Radius : 0 EE – Encryption Enable : 1 KY – AES Encrypt Key : A1B2C3D4E5F6 BD – Baud Rate : 3 NB – Parity : 0
Router Node 3	ZNET 2.5 ROUTER AT	ID – PAN ID : 100 DH – Destination High : 0 DL – Destination Low : 0 NI – Node Identifier : ROUTER_3 BH – Broadcast Radius : 0 EE – Encryption Enable : 1 KY – AES Encrypt Key : A1B2C3D4E5F6 BD – Baud Rate : 3 NB – Parity : 0 D0 – ADC : 2 IR – IO Sampling Rate : 4E20

ตารางที่ 3.3 การตั้งค่าตัวแปรต่างๆในโมดูลแต่ละจุด

ซึ่งความสามารถของ Router ต่างๆ มีดังนี้คือ

Router Node 1 ทำการตั้งค่า D0 : 3 และ IC : FF นั้น จะทำให้ Module คอยจับความเปลี่ยนแปลงทาง Port Digital IO ขาที่ 0 เมื่อมีการเปลี่ยนจากสถานะ 0 เป็น 1 หรือ 1 เป็น 0 จะทำการส่งข้อมูลในรูปแบบ Packet กลับมาที่ Coordinator

Router Node 2 ทำหน้าที่เป็นลูกข่ายในระบบ ทำหน้าที่เป็นผู้ส่งข้อมูล จากลูกข่ายตัวอื่นๆ ไปยังแม่ข่าย ในกรณีที่ลูกข่ายตัวอื่นๆมีปัญหา จะทำการส่งข้อมูลผ่าน Router2 ไปยัง Coordinator

Router Node 3 ทำการตั้งค่า D0 : 2 และ IR : 4E20 โดย IR คือช่วงเวลาที่กำหนดในการ Sampling นำค่าสัญญาณส่งกลับไปที่ Coordinator ซึ่งในที่นี้ กำหนดค่าเป็น 4E20 จะมีค่าเท่ากับ 20 วินาที (0x4E20 = 0100111000100000 ในเลขฐานสอง หรือเท่ากับ 20000 ในเลขฐานสิบ) หน่วยที่กำหนดเป็น mS ดังนั้นจะได้ 20000 x ms จะได้ 20 วินาที ดังนั้น หน้าที่ของ Router Node 3 คือ ทุกๆ 20 วินาที จะทำการนำค่า Analog ส่งกลับไปที่ Coordinator ในรูปแบบ Packet

	Frame Fields		Offset	Example	Description
API Packet	Start Delimiter		0	0x7E	
	Length		MSB 1	0x00	Number of bytes between the length and the checksum
			LSB 2	0x13	
	Frame-specific Data	Frame Type	3	0x97	
		Frame ID	4	0x55	This is the same value passed in to the request.
		64-bit Source (remote) Address	MSB 5	0x00	The address of the remote radio returning this response.
			6	0x13	
			7	0xA2	
			8	0x00	
			9	0x40	
			10	0x52	
			11	0x2B	
			LSB 12	0xAA	
		16-bit Source (remote) Address	MSB 13	0x7D	Set to the 16-bit network address of the remote. Set to 0xFFFE if unknown.
			LSB 14	0x84	
		AT Commands	15	0x53	Name of the command
			16	0x4C	
		Command Status	17	0x00	0 = OK 1 = ERROR 2 = Invalid Command 3 = Invalid Parameter
		Command Data	18	0x40	Register data in binary format. If the register was set, then this field is not returned.
			19	0x52	
			20	0x2B	
			21	0xAA	
	Checksum		22	0xF0	0xFF - the 8 bit sum of bytes from offset 3 to this byte.

รูปที่ 3.8 ตัวอย่าง Packet ที่ถูกส่งกลับมาที่ Coordinator [3]

จากรูป 3.8 Packet ที่ถูกส่งกลับมาที่ Coordinator สามารถแบ่งแยกได้เป็นหลายๆส่วนข้อมูลดังนี้

Start Delimiter เป็นการส่งข้อมูลขนาด 1 byte โดยจะต้องเริ่มต้นที่ 0x7E เท่านั้น

Length มีขนาด 2 bytes ซึ่งจะเก็บค่าความยาวของ Frame-specific Data ว่ามีกี่ bytes เช่นจากรูป 3.8 Frame-specific Data มีทั้งหมด 19 bytes ดังนั้น 19_{10} จึงเท่ากับ 13_{16} หรือ 0x13

API ID จากรูป 3.7 ในกรณีที่เป็นการถูกส่งกลับไปที่แม่ข่ายนั้น จะมีค่า 0x97

Frame ID เป็นชื่อของ Packet นั้นๆ ถ้าไม่ใช้จะถูกกำหนดเป็น 0x00

64-bit address เป็นค่า Serial High และ Serial Low Address จำนวน 8 bytes ของอุปกรณ์ที่ส่งข้อมูลมาให้

16-bit address ในกรณีเลือกใช้ 64-bit address ค่านี้จะถูกกำหนด 2 bytes นี้เป็น 0xFFFE

AT commands เป็นคำสั่งที่ถูกส่งมา เช่น 0x4953 คือ IS ในรหัส ASCII ซึ่งคำสั่ง ATIS คือคำสั่งการ Polling ค่าต่างๆทาง GPIO ที่ถูกกำหนดไว้

Command Data ข้อมูลต่างๆที่ถูก polling กลับมา

Checksum จากรูป 3.8 ค่า Checksum เกิดจาก นำ 0xFF หักออกจากจำนวน bytes ทั้งหมดของ Frame-specific Data ซึ่งจากในรูป 3.8 ค่า Checksum คือ

Checksum = 0xFFFF – (0x97 + 0x55 + 0x00 + 0x13 + 0xA2 + 0x00 + 0x40 + 0x52 + 0x2B + 0xAA + 0x7D + 0x84 + 0x53 + 0x4C + 0x00 + 0x40 + 0x52 + 0x2B + 0xAA)

Checksum = 0xFFFF – 60F

Checksum = 0xF9F0 ดังนั้นค่า Checksum 1 byte สุดท้ายคือ 0xF0

API Frame Names	API ID
AT Command	0x08
AT Command - Queue Parameter Value	0x09
ZigBee Transmit Request	0x10
Explicit Addressing ZigBee Command Frame	0x11
Remote Command Request	0x17
Create Source Route	0x21
AT Command Response	0x88
Modem Status	0x8A
ZigBee Transmit Status	0x8B
ZigBee Receive Packet (AO=0)	0x90
ZigBee Explicit Rx Indicator (AO=1)	0x91
ZigBee IO Data Sample Rx Indicator	0x92
XBee Sensor Read Indicator (AO=0)	0x94
Node Identification Indicator (AO=0)	0x95
Remote Command Response	0x97
Over-the-Air Firmware Update Status	0xA0
Route Record Indicator	0xA1

รูปที่ 3.9 หน้าที่ของ API ID ในการส่งข้อมูลระหว่างโมดูล X-Bee [3]

3.1.2 การทดลองใช้งาน

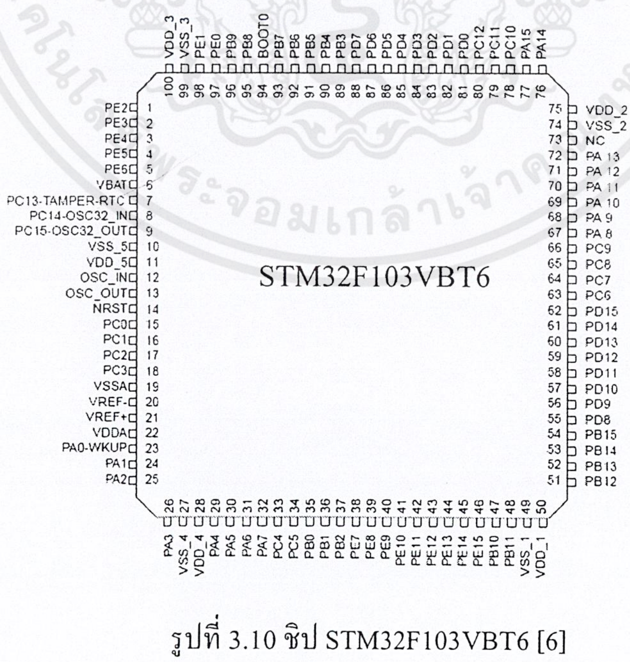
ในภาคการศึกษานี้ เป็นการทดสอบการทำงานของโมดูล X-Bee ร่วมกับอุปกรณ์อื่นๆ เช่น ไมโคร-คอนโทรลเลอร์ ซึ่งในการออกแบบการทดลองนั้น ได้ใช้ ไมโครคอนโทรลเลอร์ ARM7 Cortex-M3 ในการประมวลผลของระบบเครือข่ายจำลอง ซึ่งมีหน้าที่รวบรวมข้อมูลที่ถูกส่งมาจาก X-Bee แต่ละจุด โดยทำการแสดงผลผ่านจอ GLCD5110 ทำให้สามารถมองเห็นภาพรวมของระบบโดยรวมได้ง่ายขึ้น

3.1.2.1 ไมโครคอนโทรลเลอร์ ARM7 Cortex-M3

Cortex-M3 เป็นชื่อของโปรเซสเซอร์ 32 บิตที่พัฒนาโดย ARM Limited จากประเทศอังกฤษ และมีผู้ผลิตไมโครคอนโทรลเลอร์นำมาพัฒนาต่อ โดยบรรจุอุปกรณ์เพอริเฟอรัล และหน่วยความจำเข้าไปทำให้เกิดเป็นชิปไมโครคอนโทรลเลอร์ Cortex-M3

Cortex-M3 เป็นโปรเซสเซอร์ที่พัฒนามาจากสถาปัตยกรรม ARMv7-M มีคุณสมบัติเด่นในด้านประสิทธิภาพในการทำงานที่สูง โดยใช้ขนาดของหน่วยความจำไม่ใหญ่ ทำให้มีราคาถูกมาก เมื่อเทียบกับไมโครคอนโทรลเลอร์ 32 บิตที่พัฒนามาในระยะเวลาที่ใกล้เคียงกัน ด้วยการใช้ชุดคำสั่ง Thumb-2 ที่รองรับทั้งคำสั่ง 16 และ 32 บิต ทำให้ Cortex-M3 สามารถทำงานได้เร็วกว่า ARM7TDMI

จากรูป 3.10 บอร์ด STM32 ซึ่งบรรจุชิป STM32F103VBT6 ซึ่งมีขาต่อใช้งาน 100 ขา มีพอร์ตอินพุตเอาท์พุตมากถึง 80 ขา มีหน่วยความจำแฟลช 128 กิโลไบต์และ Static Ram 20 กิโลไบต์ มีโมดูลแปลงสัญญาณอะนาลอกเป็นดิจิตอลความละเอียด 12 บิต มากถึง 16 ช่อง มีตัวควบคุมอินเตอร์รัปต์ NCIV ทำให้รองรับการอินเตอร์รัปต์ได้อย่างดี และกำหนดความสำคัญได้ถึง 16 ระดับ นอกจากนี้ยังมี DMA ที่ทำให้โมดูลเพอริเฟอรัลต่างๆ สามารถเข้าถึงหน่วยความจำได้โดยตรง ไม่ต้องผ่านการจัดการจากซีพียู ทำให้การทำงานเร็วขึ้นอย่างมาก



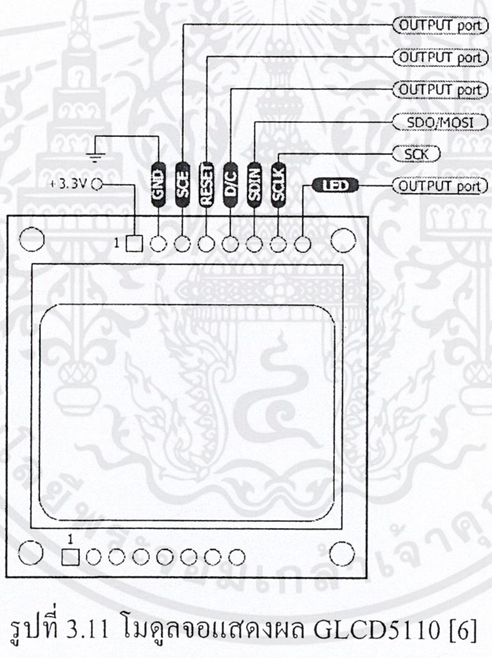
รูปที่ 3.10 ชิป STM32F103VBT6 [6]

3.1.2.2 โมดูลจอแสงผล GLCD5110

โมดูล LCD เป็นอุปกรณ์แสดงผลที่ได้รับความนิยมสูงในการนำมาประยุกต์ใช้งานร่วมกับไมโครคอนโทรลเลอร์เพื่อแสดงค่าตัวเลข ข้อความ รวมทั้งสัญลักษณ์และเครื่องหมายต่างๆ จากแบบอักขระหรือแบบกราฟิก เพื่อให้ผู้ใช้งานสามารถสร้างสรรค์งานแสดงผลได้อย่างกว้างขวางมากขึ้น เนื่องจากโมดูลกราฟิก LCD สามารถแสดงผลได้อิสระลงถึงระดับจุด หรือที่เรียกว่า พิกเซล ผู้ใช้งานจึงสามารถนำภาพลงไปแสดงผลของจอกราฟิก LCD นี้ได้ โดยคุณภาพจะขึ้นกับความละเอียดของจอแสดงผล

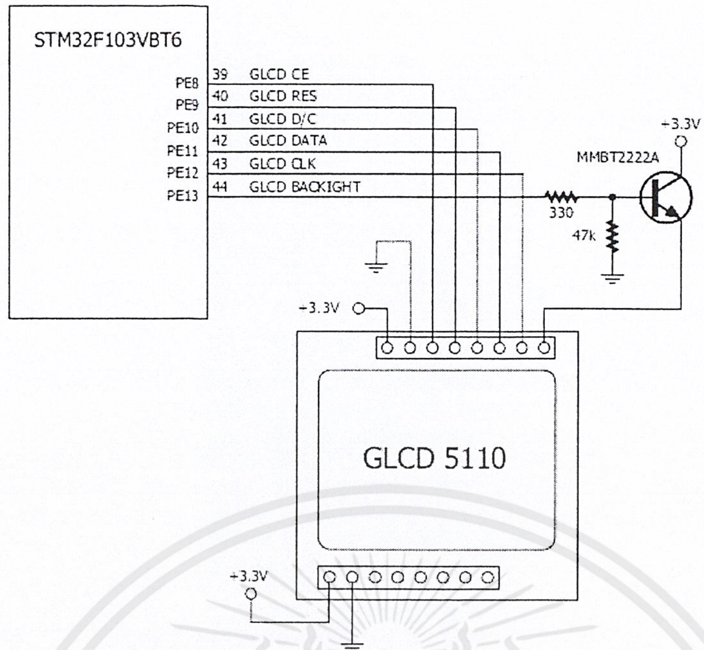
โมดูล LCD 5110 ที่นำมาใช้เป็น LCD ที่ใช้งานตัวเดียวกับโทรศัพท์เคลื่อนที่ Nokia 5110 ซึ่งในปัจจุบันสามารถจัดหาเพื่อนำมาใช้ในการแสดงผลของระบบไมโครคอนโทรลเลอร์ได้ง่ายขึ้น ทั้งยังมีราคาไม่สูงเมื่อเทียบกับประโยชน์ที่ได้

ซึ่งคุณสมบัติทางเทคนิคของ GLCD 5110 นี้ มีความละเอียด 84 x 48 จุด โดยใช้ตัวควบคุมเบอร์ PCD8544 ติดต่อด้วย SPI ใช้ไฟเลี้ยงย่าน +3 ถึง 3.3V ซึ่งเป็นไฟเลี้ยงย่านเดียวกับ ไมโครคอนโทรลเลอร์ ARM7 Cortex-M3



รูปที่ 3.11 โมดูลจอแสดงผล GLCD5110 [6]

ในการใช้งานโมดูล GLCD 5110 จะต้องเขียนโปรแกรมเพื่อติดต่อกับตัวควบคุมภายในโมดูล PCD8544 โดยติดต่อแบบ SPI ซึ่งจะต้องใช้สายสัญญาณข้อมูลอนุกรม 1 เส้น และสัญญาณนาฬิกา ส่วนขาควบคุมอื่นๆที่เหลือจะเป็นการส่งสัญญาณลอจิกไปควบคุมตรงๆ ซึ่งวิธีการต่อวงจรโมดูล GLCD5110 กับ ARM7 Cortex-M3 เป็นไปตามรูป 3.12



รูปที่ 3.12 การต่อโมดูลจอแสดงผล GLCD5110 กับ ARM7 Cortex-M3 [6]

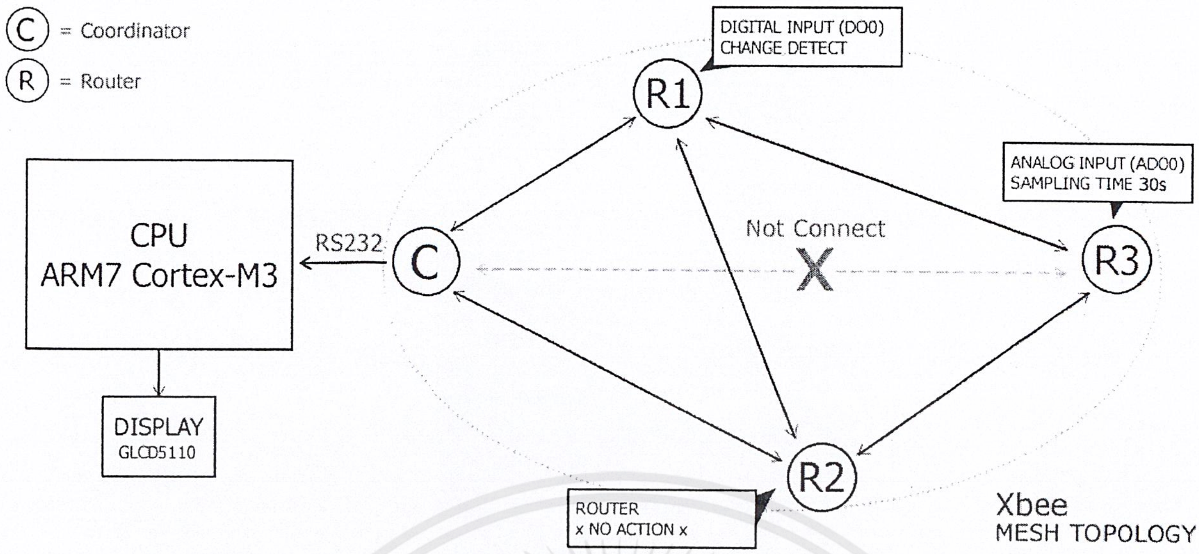
3.1.2.3 ทดสอบโมดูล X-Bee กับไมโครคอนโทรลเลอร์ ARM7 Cortex-M3

ในส่วนของการทดสอบนั้น ได้ทำการออกแบบระบบดังนี้คือ กำหนด X-Bee ทำหน้าที่เป็นแม่ข่าย 1 จุด ซึ่งในจุดของแม่ข่ายนั้น จะมีไมโครคอนโทรลเลอร์ ต่ออยู่กับ X-Bee ซึ่งมีหน้าที่เป็นตัวประมวลผลกลาง

นอกจากนี้ X-Bee จุดที่เหลือนั้น ทำหน้าที่ต่างๆดังนี้

1. จุดที่ 1 ทำหน้าที่เป็น Router เมื่อมีการเปลี่ยนแปลงที่สถานะดิจิตอลที่ขา DO0 จะทำการส่งข้อมูลกลับไปที่ Coordinator เพื่อแจ้งสถานะ ซึ่งจะส่งก็ต่อเมื่อมีการเปลี่ยนแปลงสถานะของขา DO0 เท่านั้น
2. จุดที่ 2 ทำหน้าที่เป็น Router เพื่อกำหนดเป็นจุดทางผ่านจุดหนึ่งของระบบ
3. จุดที่ 3 ทำหน้าที่เป็น Router ในจุดนี้จะกำหนดให้ทำการส่งข้อมูลสถานะที่ขา AD0 ทุกๆ 30 วินาที เพื่อแจ้งกลับไปที่ Coordinator ให้บันทึกข้อมูลสถานะข้อมูลอนาลอกทุกๆ 30 วินาที

ในการวางแต่ละจุดของ Router นั้น จากรูป 3.13 เป็นการวาง Router 1 และ 2 ให้สามารถสื่อสารกับ Coordinator ได้ แต่จุดที่วาง Router 3 นั้น ไม่สามารถเชื่อมต่อกับ Coordinator ได้โดยตรง ทำให้ต้องทำการเลือกเส้นทางด้วยโปรโตคอลของ Zigbee เอง ซึ่งในการทดสอบนั้นสามารถทดลองโดย ปิด Router 1 และสังเกตว่าข้อมูลจาก Router 3 สามารถส่งกลับมายังที่ตัว Coordinator ได้อยู่ หรือ ปิด Router 2 และสังเกตว่าข้อมูลจาก Router 3 สามารถส่งกลับมายังที่ตัว Coordinator ได้เช่นกัน ซึ่งการสื่อสารโครงข่ายในลักษณะแบบนี้จะใช้ประสิทธิภาพของ Zigbee อย่างสูงสุด คือ Mesh Topology



รูปที่ 3.13 รูปแบบการทดสอบโมดูล X-Bee กับไมโครคอนโทรลเลอร์ ARM7 Cortex-M3

ในส่วนของ ARM7 Cortex-M3 จะเป็นตัวรวบรวมข้อมูลทั้งหมด โดยการอ่าน API Packet Frame จาก X-Bee จุดต่างๆ ที่ถูกส่งมาที่ Coordinator และนำมาแสดงผลในจอ LCD เป็นลักษณะการรวบรวมข้อมูลนำมา Monitor เพื่อให้ผู้ใช้สามารถตรวจเช็คภาพรวมของระบบได้

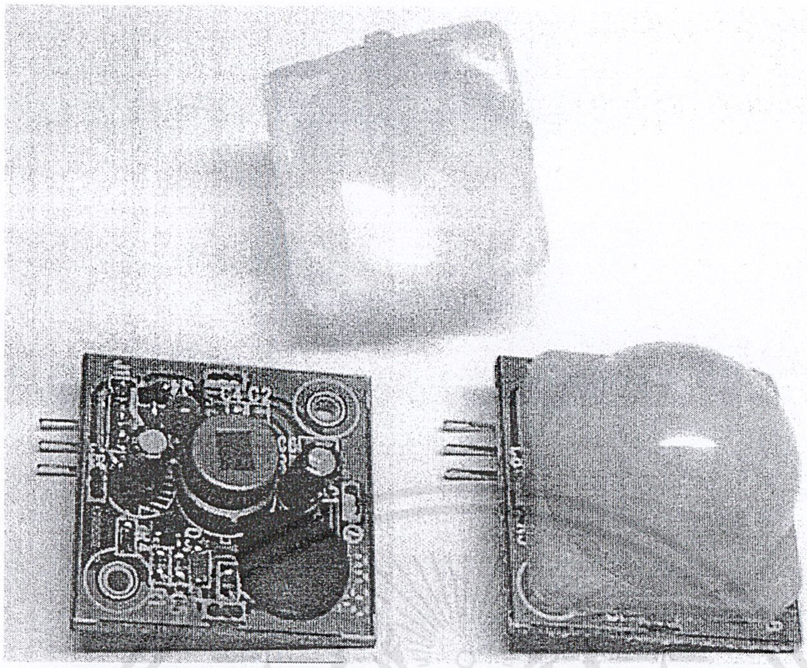
3.1.3 เซ็นเซอร์และการประยุกต์ใช้งาน

ในการทดลองภาคการศึกษานี้ จะเป็นการศึกษาทฤษฎีเซนเซอร์แบบต่างๆ ซึ่งใช้สำหรับการทดสอบการทำงานของ Zigbee เพื่อส่งข้อมูลกลับไปยัง Coordinator โดยเซนเซอร์ที่ใช้ทดสอบมีดังต่อไปนี้

3.1.3.1 เซ็นเซอร์ตรวจจับการเคลื่อนไหว ZX-PIR

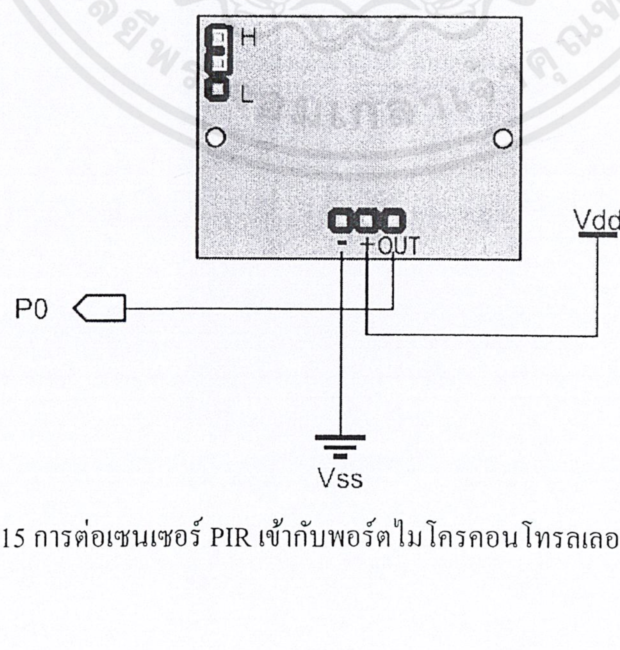
เซนเซอร์การตรวจจับความเคลื่อนไหวของสิ่งมีชีวิต (Motion Sensor) ที่ได้รับความนิยมและใช้งานง่ายคือตรวจจับแบบอินฟราเรด ซึ่งใช้หลักการตรวจจับที่เรียกว่า ไพโรอิเล็กคริก (pyro-electric) อันเป็นการตรวจจับการแผ่รังสีอินฟราเรด หากระดับของการแผ่รังสีไม่เปลี่ยนแปลง แสดงว่า สิ่งมีชีวิตที่ต้องการตรวจจับนั้นไม่มีการเคลื่อนไหว แต่ ถ้าหากมีการเคลื่อนไหวเกิดขึ้น ระดับของการแผ่รังสีอินฟราเรดจะเปลี่ยนแปลง

สิ่งมีชีวิตไม่ว่าจะเป็นมนุษย์หรือสัตว์เลือดอุ่นในภาวะที่ยังมีชีวิตอยู่ จะมีการกระจายพลังงานความร้อนออกมาจากตัวเองในรูปของการแผ่รังสีอินฟราเรดอยู่ตลอดเวลา โดยจะมีปริมาณมากหรือน้อยขึ้นอยู่กับสภาพของร่างกายในขณะนั้น เมื่อมีการเคลื่อนไหวปริมาณของการแผ่รังสีก็จะเปลี่ยนแปลง รังสีอินฟราเรดจากมนุษย์หรือสัตว์เลือดอุ่นที่มีระดับความเข้มสูงสุดจะมีความยาวคลื่นประมาณ 9.4 ไมโครเมตร



รูปที่ 3.14 ส่วนประกอบภายในของโมดูล PIR พร้อมใช้งาน ด้านบนคือเลนส์ไฟรเนล, ด้านซ้ายคือแผงวงจรตัวตรวจจับ PIR, ด้านขวาคือ โมดูล PIR เมื่อประกอบเลนส์และแผงวงจรเข้าด้วยกัน [7]

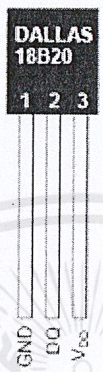
การใช้งานเนื่องจากเอาท์พุตของ ZX-PIR เป็นสัญญาณดิจิตอลที่มีสองสถานะคือ ลอจิกสูงหรือ “1” และลอจิกต่ำหรือ “0” จึงสามารถเชื่อมต่อกับขาพอร์ตดิจิตอลของไมโครคอนโทรลเลอร์ได้ทุกขา โดยต้องกำหนดให้ขาพอร์ตที่เชื่อมต่อนั้นเป็นอินพุตดิจิตอลก่อน และไม่ต้องต่อตัวต้านทานพูลอัพที่ขาพอร์ตของไมโครคอนโทรลเลอร์ซึ่งนำมาต่อกับโมดูล ZX-PIR



รูปที่ 3.15 การต่อเซนเซอร์ PIR เข้ากับพอร์ตไมโครคอนโทรลเลอร์ P0 [8]

3.1.3.2 เซ็นเซอร์วัดอุณหภูมิ DS18B20

DS18B20 เป็น IC วัดอุณหภูมิแบบดิจิตอล สามารถวัดอุณหภูมิเป็นหน่วยองศา C ในช่วง -55C ถึง 125C ที่ความละเอียด 9-12 บิต และมีความแม่นยำอยู่ที่ 0.5C ในช่วง -10C ถึง 85C ในกรณีที่เป็นตัวถังแบบ TO-92 นั้นจะมีโครงสร้าง และขาดังแสดงในรูปที่ 3.16



รูปที่ 3.16 ขาการใช้งาน DS18B20 [9]

การสื่อสารและควบคุม DS18B20 นั้นสามารถทำได้โดยใช้บัสข้อมูลแบบ 1-wire ซึ่งใช้สายสัญญาณเพียงแค่เส้นเดียวเท่านั้น ภายใน DS18B20 แต่ละตัวมีโค๊ดประจำตัวขนาด 64 บิต ทำให้สามารถใช้งาน DS18B20 หลายตัวทำงานบนบัสแบบ 1 wire พร้อมกันได้ นอกจากนี้ DS18B20 ยังสามารถทำงานในโหมดพาราสิต (Parasite Power Mode) ซึ่งเป็นการทำงานโดยไม่ใช้ไฟเลี้ยง แต่ใช้พลังงานจากสายสัญญาณ 1-wire ซึ่งมีประโยชน์มากสำหรับการวัดอุณหภูมิระยะไกล หรือในการใช้งานในที่ๆ มีเนื้อที่จำกัด

ข้อมูลอุณหภูมิที่วัดได้จะถูกเก็บอยู่ในรีจิสเตอร์ Temperature ซึ่งมีขนาด 16 บิต ดังแสดงในรูปที่ 3.17 ถ้าข้อมูลอุณหภูมิเป็นบวก S จะเป็น “1” แต่ถ้าข้อมูลอุณหภูมิเป็นลบ S จะเป็น “0” ในกรณีที่ DS18B20 ทำงานในโหมดความละเอียด 12 บิต บิตทุกบิตในรีจิสเตอร์ Temperature จะถูกใช้หมด แต่ในกรณีที่ทำงานในโหมด 9-11 บิต บิตล่าง (บิต 0 – บิต 2) จะไม่ถูกใช้งาน ซึ่งในการกำหนดโหมดความละเอียดการทำงานของ DS18B20 นั้นสามารถกำหนดได้ที่รีจิสเตอร์ Configuration ซึ่งโดยปกติเริ่มต้น DS18B20 จะทำงานในโหมด 12 บิต

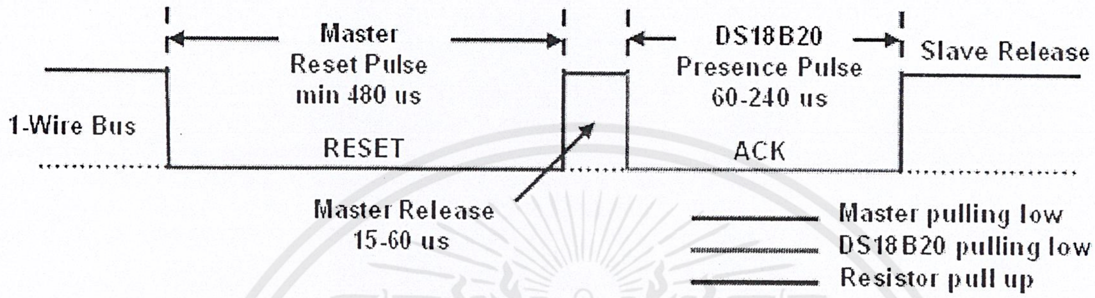
	bit7	bit6	bit5	bit4	bit3	bit2	bit1	bit0
LS Byte	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}	2^{-4}
	bit15	**bit14**	**bit13**	**bit12**	**bit11**	**bit10**	**bit9**	**bit8**
MS Byte	S	S	S	S	S	2^6	2^5	2^4

รูปที่ 3.17 โครงสร้างภายในรีจิสเตอร์ Temperature LSB และ MSB [9]

การสื่อสารแบบ 1-wire เป็นระบบบัสข้อมูลแบบ Half-duplex นั้นคือสามารถสื่อสารได้ 2 ทิศทาง แต่ไม่สามารถรับ และส่งข้อมูลพร้อมกันในช่วงเวลาเดียวกันได้ ระบบบัสมีการทำงานเป็นแบบ Master/Slave โดย

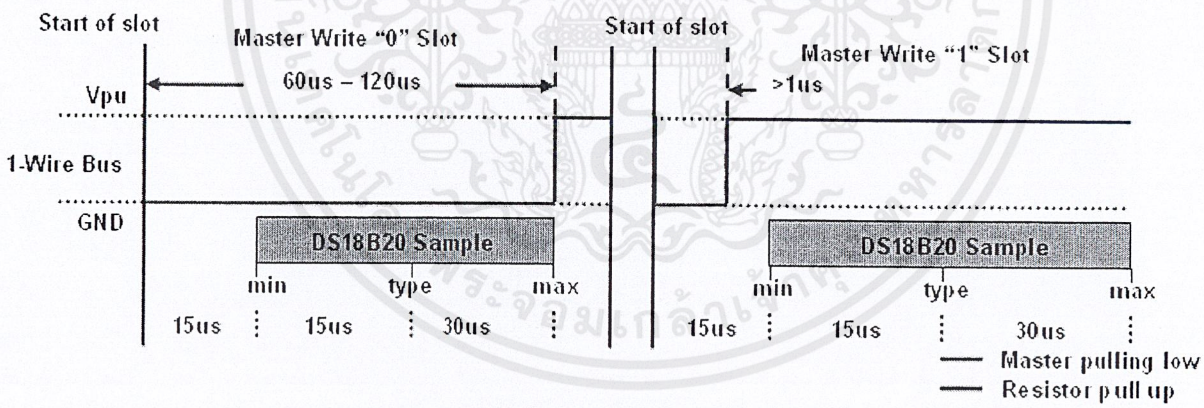
อุปกรณ์ Master จะเป็นตัวควบคุมสถานะ และจังหวะการรับส่งของบัสข้อมูล ในขณะที่อุปกรณ์ Slave จะทำงานตามการควบคุมของอุปกรณ์ Master เท่านั้น

รูปแบบของสัญญาณบนบัส 1-wire สามารถแบ่งออกได้เป็น 6 รูปแบบ คือ Reset pulse, Presence pulse, write 0, write 1, read 0, read 1 ในกระบวนการเริ่มต้นการสื่อสารแบบ 1-wire ทั้งหมดนั้น อุปกรณ์ Master ต้องขอเริ่มการสื่อสารด้วยการสร้าง Reset pulse ก่อน เมื่ออุปกรณ์ Slave ได้รับ Reset pulse ก็จะสร้าง Presence pulse เพื่อตอบรับการขอเริ่มการสื่อสารนั้น ซึ่งมีรายละเอียดของช่วงเวลาต่าง ๆ ดังแสดงในรูปที่ 3.18



รูปที่ 3.18 การเริ่มการติดต่อสื่อสารแบบ 1-wire ด้วย Reset Pulse และ Presence Pulse [9]

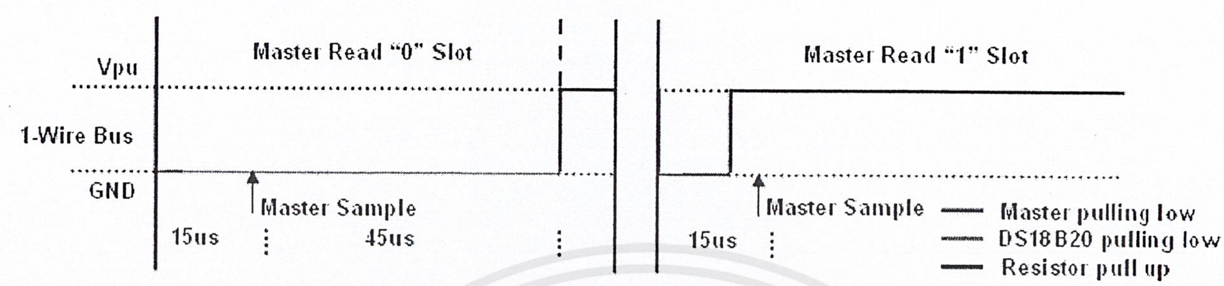
ในการเขียนข้อมูลแบ่งออกเป็น 2 ชนิดคือการเขียนข้อมูล “1” และการเขียนข้อมูล “0” ดังแสดงในรูปที่ 3.19 การเขียนข้อมูลลง DS18B20 ต้องใช้ช่วงเวลาของไทม์สล๊อตอย่างต่ำ 60uS และต้องมีช่วงเวลาระหว่างไทม์สล๊อตอย่างต่ำ 1uS



รูปที่ 3.19 การเขียนข้อมูลลง DS18B20 [9]

การเขียนข้อมูลทั้ง 2 ชนิด เริ่มแรกอุปกรณ์ Master ต้องดึงสัญญาณบนบัส 1-wire ลงมาให้อยู่ในสถานะลอจิกต่ำก่อน ในกรณีที่ต้องการเขียนข้อมูล “0” ลงใน DS18B20 อุปกรณ์ Master ต้องดึงสัญญาณบนบัสให้เป็นลอจิกต่ำต่อ จนกว่าจะครบช่วงเวลาไทม์สล๊อต (อย่างต่ำ 60 uS) ส่วนในกรณีที่ต้องการเขียนข้อมูล “1” ลง DS18B20 อุปกรณ์ Master ต้องปล่อยบัส เพื่อให้บัสกลับไปอยู่ในสถานะลอจิกสูงก่อนการ Sampling ของ DS18B20 ซึ่งจะอยู่ในช่วง 15 uS - 60 uS หลังจากที่อุปกรณ์ Master ดึงสัญญาณบัส 1-wire ลงมา

ในการอ่านค่าภายใน SRAM ของ DS18B20 สามารถทำได้ก็ต่อเมื่ออุปกรณ์ Master ได้เขียนข้อมูลเพื่อขอทำการอ่านค่าใน SRAM (Read Scratchpad) ซึ่งมีค่าเป็น 0xBE ลงไปที่ DS18B20 เสียก่อน จากนั้นจึงเริ่มอ่านข้อมูลจากบัส 1-wire โดยไทม์สล๊อตของการอ่านต้องมีช่วงเวลาอย่างต่ำ 60 uS และต้องมีช่วงเวลาระหว่างไทม์สล๊อตอย่างต่ำ 1 uS ดังแสดงในรูปที่ 3.19



รูปที่ 3.19 การอ่านข้อมูลจาก DS18B20 [9]

การอ่านข้อมูลจากบัส 1-wire เริ่มแรกอุปกรณ์ Master จะต้องดึงบัส 1-wire ลงให้อยู่ในสถานะลอจิกต่ำเป็นช่วงเวลาอย่างน้อย 1 uS จากนั้นจึงค่อยปล่อยบัส ในกรณีที่ DS18B20 ส่งข้อมูล "0" DS18B20 จะดึงบัสให้เป็นลอจิกต่ำจนจนกว่าจะสิ้นสุดไทม์สล๊อตถึงจึงจะปล่อยบัสให้กลับไปอยู่ในสถานะลอจิกสูง ส่วนในกรณีที่ DS18B20 ส่งข้อมูล "1" DS18B20 จะปล่อยบัสให้อยู่ในสภานะลอจิกสูงตลอด ในการ Sample เพื่อรับข้อมูลจาก DS18B20 ควรทำภายใน 15 uS หลังจากจุดเริ่มของไทม์สล๊อตดังแสดงในรูปที่ 3.19

ขั้นตอนการเข้าใช้งาน DS18B20 มี 3 ขั้นตอนด้วยกันคือ 1. Initialization 2. ROM Command 3. DS18B20 Function Command

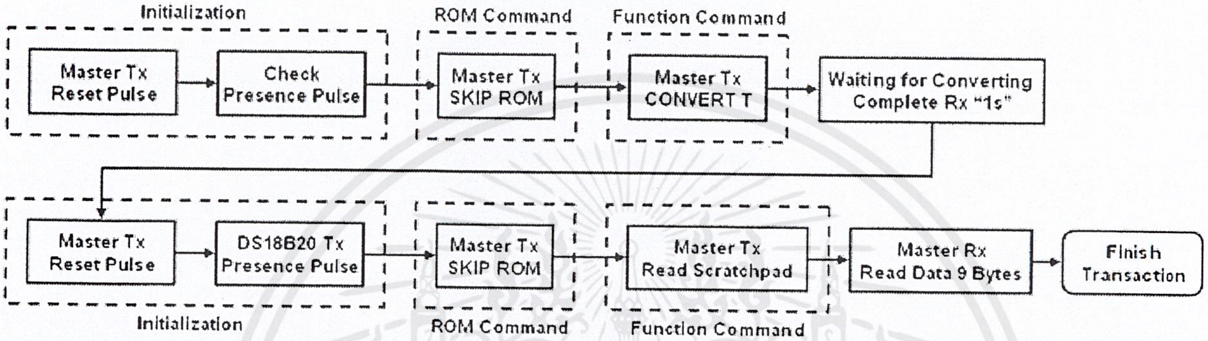
การ Initialization ประกอบไปด้วยการส่ง Reset Pulse จากอุปกรณ์ Master ตามด้วย Presence Pulse ซึ่งตอบรับโดย DS18B20 เพื่อบ่งบอกว่าอุปกรณ์พร้อมทำงาน

หลังจากการทำ Initialization เสร็จเรียบร้อยแล้ว อุปกรณ์ Master ต้องส่ง ROM Command ไปยัง DS18B20 ROM Command นั้นแบ่งออกได้เป็น 5 คำสั่งด้วยกันคือ SEARCH ROM [0xF0], READ ROM [0x33], MATCH ROM [0x55], SKIP ROM [0xCC], ALARM SEARCH [0xEC] ซึ่งในกรณีที่ต่อใช้งาน DS18B20 เพียงตัวเดียวนั้นจะใช้ ROM Command ได้แค่ 2 คำสั่งนั้นคือ READ ROM ซึ่งเป็นการอ่านค่า ROM Code ขนาด 64 บิต บนตัว DS18B20 อีกคำสั่งคือ SKIP ROM ซึ่งเป็นคำสั่งที่ใช้ในกรณีที่อุปกรณ์ Master ต้องการส่งคำสั่งควบคุม DS18B20 ทุกตัว ซึ่งไม่จำเป็นต้องระบุ ROM Code

หลังจากที่อุปกรณ์ Master ส่ง Rom Command ไปยัง DS18B20 แล้ว อุปกรณ์ Master จะสามารถใช้ Function Command เพื่อเข้าไปควบคุมการทำงานของ DS18B20 ได้ Function Command ประกอบไปด้วย CONVERT T[0x44], WRITE SCRATCHPAD[0x4E], READ SCRATCHPAD[0xBE], COPY SCRATCHPAD [0x48], RECALL E2[0xB8], READ POWER SUPPLY [0xB4]

ขั้นตอนการอ่านค่าอุณหภูมิจาก DS18B20 นั้น แบ่งออกเป็น 2 ขั้นตอนใหญ่ ๆ คือ Converting Temperature ซึ่งเป็นการแปลงอุณหภูมิให้อยู่ในรูปข้อมูลดิจิตอลเก็บไว้ใน Scratchpad และ Read Scratchpad ซึ่ง

เป็นการอ่านข้อมูลที่เก็บใน Scratchpad นั้นออกมา ดังแสดงในรูปที่ 3.20 จะเห็นได้ว่าไมโครคอนโทรเลอร์ซึ่งเป็นอุปกรณ์ Master จะต้องทำการ Initialization จากนั้นจึงค่อยคำสั่ง SKIP ROM [0xCC] ซึ่งเป็น ROM Command ที่ใช้ในกรณีที่ต่อใช้งาน DS18B20 เพียงตัวเดียวจากนั้นจึงค่อยส่ง Function Command CONVERT T [0x44] เพื่อสั่งให้ DS18B20 ทำการแปลงอุณหภูมิให้อยู่ในรูปแบบดิจิตอลและเก็บไว้ใน Scratchpad กระบวนการแปลงนี้จะใช้เวลาประมาณ 750ms ซึ่งเราสามารถตรวจสอบสถานการณ์แปลงข้อมูลได้จากการตรวจสอบสถานะของบัส 1-wire ในกรณีที่ DS18B20 ยังทำการแปลงข้อมูลอยู่บัส 1-wire จะมีสถานะเป็นลอจิกต่ำ แต่ถ้า DS18B20 ได้ทำการแปลงข้อมูลเสร็จเรียบร้อยแล้วบัส 1-wire จะกลับมาอยู่ในสถานะปกติคือลอจิกสูง

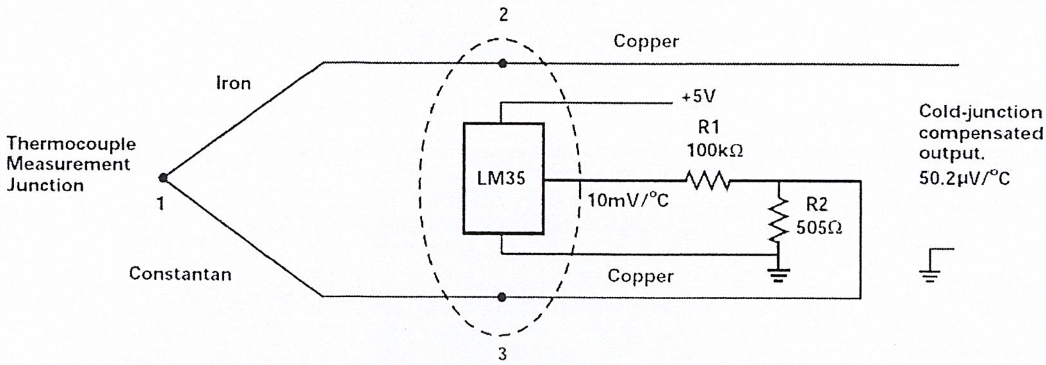


รูปที่ 3.20 ขั้นตอนการแปลง และอ่านอุณหภูมิจาก DS18B20

เมื่อ DS18B20 แปลงข้อมูลอุณหภูมิเป็นดิจิตอลเก็บไว้ใน Scratchpad แล้ว ไมโครคอนโทรลเลอร์สามารถอ่านข้อมูลภายใน Scratchpad นั้นโดยทำการ Initialization บัส 1-wire อีกครั้ง จากนั้นจึงส่งคำสั่ง SKIP ROM [0xCC] ซึ่งเป็น ROM Command และส่งคำสั่ง READ SCRATCHPAD [0xBE] ซึ่งเป็น Function Command เพื่อขออ่านข้อมูล Scratchpad ภายใน DS18B20 จากนั้นไมโครคอนโทรลเลอร์จึงค่อยทำการอ่านข้อมูลภายใน Scratchpad ของ DS18B20 มาทีละไบต์จนครบ 9 ไบต์ และแสดงผลข้อมูลอุณหภูมิที่อยู่ซึ่งเป็นข้อมูลในไบต์แรกและไบต์ 2 ที่อ่านมาจาก Scratchpad นั้นออกมาทางพอร์ตอนุกรม ดังแสดงในรูปที่ 3.20

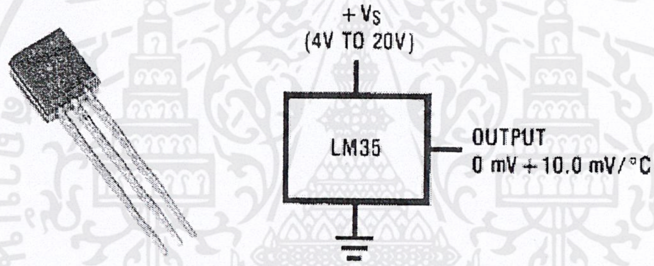
3.1.3.3 เซ็นเซอร์วัดอุณหภูมิ LM35 DZ

LM35 DZ เป็นเซ็นเซอร์วัดอุณหภูมิโดยใช้หลักการของ เทอร์โมคัปเปิล (Thermocouple) ซึ่งเป็นการเปลี่ยนแปลงอุณหภูมิหรือความร้อนเป็นแรงเคลื่อนไฟฟ้า (emf) เทอร์โมคัปเปิลทำมาจากโลหะตัวนำที่ต่างชนิดกัน 2 ตัว (แตกต่างกันทางโครงสร้างของอะตอม) นำมาเชื่อมต่อปลายทั้งสองเข้าด้วยกันที่ปลายด้านหนึ่ง เรียกว่าจุดวัดอุณหภูมิ ส่วนปลายอีกด้านหนึ่งปล่อยเปิดไว้ เรียกว่าจุดอ้างอิง หากจุดวัดอุณหภูมิและจุดอ้างอิงมีอุณหภูมิต่างกันจะทำให้มีการนำกระแสในวงจรเทอร์โมคัปเปิลทั้งสองข้างดังรูปที่ 3.21



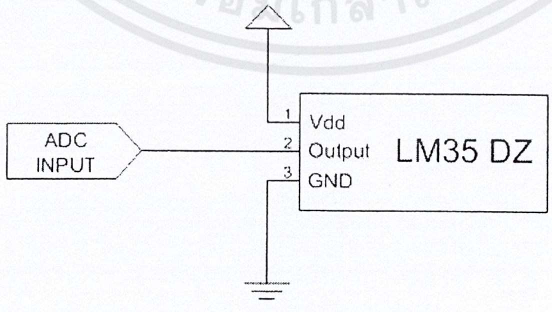
รูปที่ 3.21 หลักการเทอร์โมคัปเปิลภายในเซ็นเซอร์วัดอุณหภูมิ LM35 DZ

การใช้งานเนื่องจากเอาท์พุตของ LM35 DZ เป็นสัญญาณอนาลอกที่มีลักษณะเป็นเชิงเส้น โดยใน Datasheet ของเซ็นเซอร์ตัวนี้ ได้ระบุว่า มีความเป็นเชิงเส้น +10.0 mV ต่อ 1 องศาเซลเซียส โดยครอบคลุมช่วงการใช้งานตั้งแต่ 0 – 100 องศาเซลเซียส



รูปที่ 3.22 เซ็นเซอร์วัดอุณหภูมิ LM35 DZ และวิธีการต่อใช้งาน

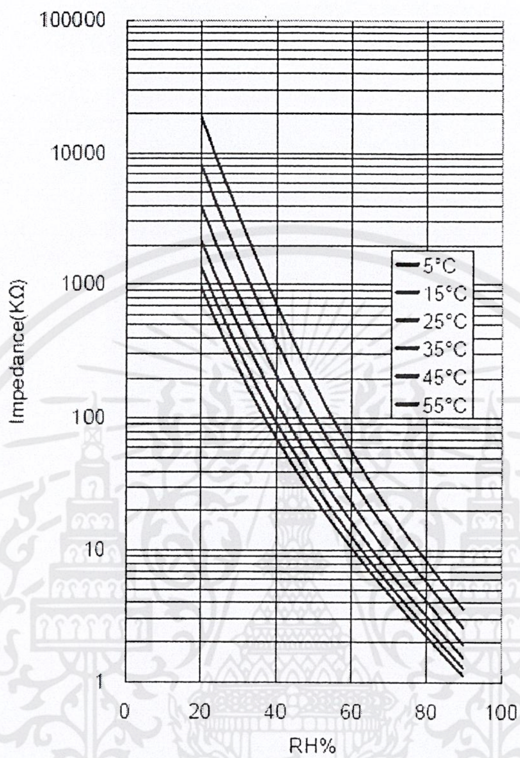
วิธีการต่อใช้งานเซ็นเซอร์วัดอุณหภูมิ LM35 DZ นั้น สามารถต่อเข้ากับ ขา Input Analog ของไมโครคอนโทรลเลอร์ หรือ ถ้านำมาใช้กับ โมดูล X-Bee นั้น ก็สามารถต่อเข้าทางพอร์ต Input Analog ของโมดูล X-Bee ได้ ดังรูปที่ 3.23



รูปที่ 3.23 การต่อเซนเซอร์วัดอุณหภูมิ LM35 DZ เข้ากับพอร์ต Analog to Digital Convertor

3.1.3.4 โมดูลวัดความชื้นและอุณหภูมิ SK 2L 94V-0

SK 2L 97V-0 เป็นโมดูลวัดความชื้นและอุณหภูมิ โดยใช้หัววัดความชื้น HCZ-J32 ซึ่ง เป็น Resistive Humidity Sensor โดยหลักการของเซ็นเซอร์ชนิดนี้ ใช้วิธีการวัดความเปลี่ยนแปลงของความต้านทานภายใน ดังรูปที่ 3.24

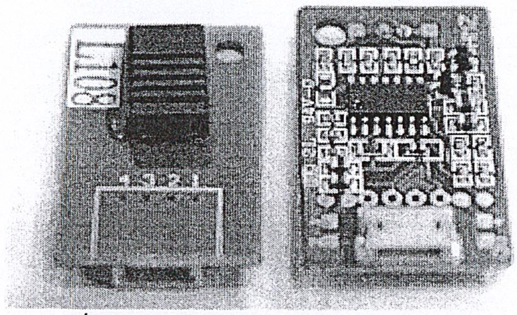


รูปที่ 3.24 ความสัมพันธ์ระหว่าง ความชื้นสัมพัทธ์ (%RH) และ ความต้านทานภายในหัววัด

ซึ่งภายในโมดูล SK 2L 97V-0 นี้ได้มีวงจรภายใน ในการแปลงค่าความต้านทานภายในของหัววัด ให้เป็นแรงดัน ทำให้สามารถใช้งานได้ง่ายขึ้น ซึ่งผู้ใช้ สามารถนำค่าความสัมพันธ์ของแรงดันเหล่านี้ มาใช้ในการวัดค่าความชื้นสัมพัทธ์ในอากาศ ดังตารางที่ 3.5

%RH / Temp	10	15	20	25	30	35	40
10	0.97	0.95	0.95	0.95	0.85	0.82	0.81
20	1.15	1.12	1.13	1.09	0.98	0.96	0.88
30	1.48	1.45	1.44	1.42	1.36	1.37	1.31
40	1.76	1.73	1.74	**1.75**	**1.73**	**1.73**	**1.7**
50	2.06	2.06	2.08	**2.09**	**2.08**	**2.07**	**2.06**
60	2.36	2.37	2.37	**2.37**	**2.37**	**2.37**	**2.35**
70	2.64	2.63	2.62	**2.63**	**2.64**	**2.64**	**2.61**
80	2.92	2.91	2.92	**2.91**	**2.9**	**2.91**	**2.9**
90	3.09	3.08	3.07	3.09	3.07	3.04	3.03

ตารางที่ 3.5 ความสัมพันธ์ระหว่างแรงดันของความชื้นที่อุณหภูมิต่างๆ



รูปที่ 3.25 ลักษณะของโมดูล SK 2L 97V-0

3.2 การทดสอบและผลการทดสอบโครงงานเบื้องต้น

3.2.1 วิธีการทดสอบ

3.2.1.1 ทดสอบการส่งข้อมูลแบบ Star Topology

1.) กำหนดหน้าที่ของโมดูล X-Bee โดย 1 ตัวสำหรับทำหน้าที่เป็น Coordinator และ 1 ตัวสำหรับทำหน้าที่เป็น End Device โดยกำหนดค่าพารามิเตอร์สำคัญๆดังนี้

1.1) Coordinator กำหนด DH = 0000 และ DL = FFFF

1.2) End Device กำหนด Destination Address ให้ตรงกับ SH และ SL ของ Coordinator

2.) ทดสอบการส่งข้อมูลขนาด 1 byte จาก Coordinator และตรวจสอบการรับข้อมูลที่ End Device โดยเปลี่ยนระยะทางและสถานที่ต่างๆกัน

รูป 3.25 ออกแบบการทดลองสื่อสารโครงข่าย Star Topology [10]

3.2.1.2 ทดสอบการส่งข้อมูลแบบ Tree Topology

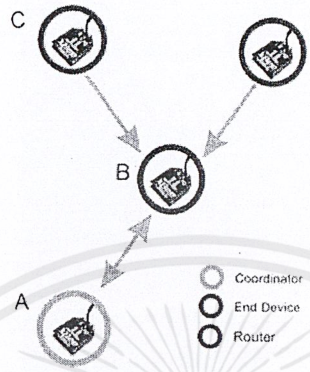
1.) กำหนดหน้าที่ของโมดูล X-Bee โดย 1 ตัวสำหรับทำหน้าที่เป็น Coordinator, 1 ตัวสำหรับทำหน้าที่เป็น Router และ 1 ตัวสำหรับทำหน้าที่เป็น End Device โดยกำหนดค่าพารามิเตอร์สำคัญๆดังนี้

1.1) Coordinator กำหนด Destination Address ให้ตรงกับ SH และ SL ของ End Device

1.2) Router กำหนด DL = 0000 และ DH = 0000

1.3) End Device กำหนด Destination Address ให้ตรงกับ SH และ SL ของ Coordinator

2.) ทดสอบการส่งข้อมูลขนาด 1 byte จาก Coordinator โดยส่งข้อมูลกันแบบ Star Topology แล้วจึงค่อยๆ เคลื่อนที่ End Device ออกห่างจาก Coordinator จน X-Bee ทั้ง 2 ตัวไม่สามารถรับส่งข้อมูลกันได้ จากนั้นจึงนำ X-Bee ที่ทำหน้าที่เป็น Router มาคั่นกลางระหว่าง Coordinator และ End Device และสังเกตผลที่เกิดขึ้นว่าสามารถกลับมารับข้อมูลได้อีกหรือไม่



รูป 3.26 ออกแบบการทดลองสื่อสารโครงข่าย Tree Topology [10]

3.2.1.3 ทดสอบการส่งข้อมูลแบบ Mesh Topology

1.) กำหนดหน้าที่ของโมดูล X-Bee โดย 1 ตัวสำหรับทำหน้าที่เป็น Coordinator และ 3 ตัวสำหรับทำหน้าที่เป็น Router โดยกำหนดค่าพารามิเตอร์สำคัญๆดังนี้

1.1) Coordinator กำหนด DH = 0000 และ DL = FFFF

1.2) Router1 กำหนด DL = 0000 และ DH = 0000

1.3) Router2 กำหนด DL = 0000 และ DH = 0000

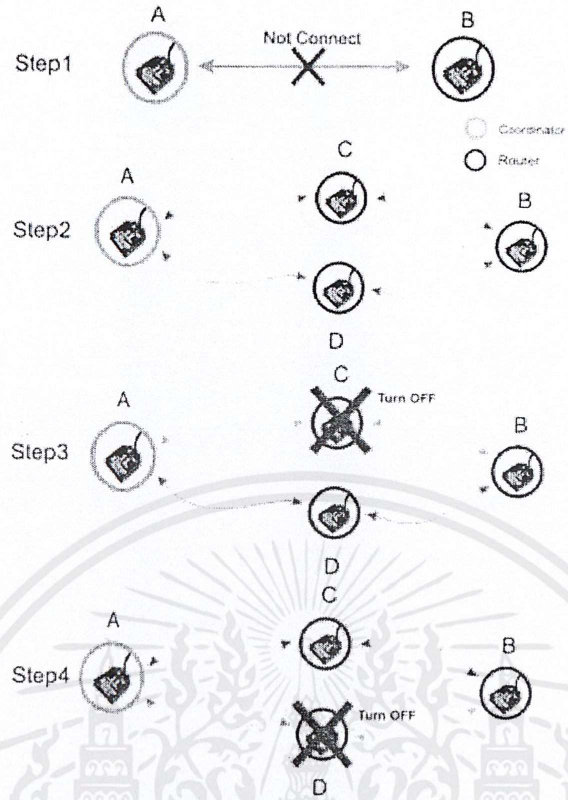
1.4) Router3 กำหนด DL = 0000 และ DH = 0000

2.) นำโมดูล X-Bee ที่ตั้งค่าแล้ว คือ Coordinator (A) และ Router (B) สองตัวมาทดลองส่งข้อมูลหากันโดยตั้ง DH, DL ของทั้ง 2 ตัวให้สามารถส่งข้อมูลหากันได้ จากนั้น แยก X-Bee ทั้ง 2 ตัวออกจากกันจนไม่สามารถติดต่อกันได้อีก

3.) จากนั้น นำเอา X-Bee อีก 2 ตัวที่ตั้งค่าเป็น Router (C, D) มาวางคั่นกลาง จะพบว่า เราจะสามารถ ติดต่อระหว่าง Coordinator (A) และ Router (B) ได้อีกครั้ง

4.) ปิดแต่ Router (C) แล้วสังเกตว่า Router (A) ยังสามารถคุยกับ Router (B) ได้ หรือไม่

5.) ปิดแต่ Router (D) แล้วสังเกตว่า Router (A) ยังสามารถคุยกับ Router (B) ได้ หรือไม่



รูป 3.27 ออกแบบการทดลองสื่อสาร โครงข่าย Mesh Topology [10]

3.2.1.4 ทดสอบการใช้งาน PIR (Motion) Sensor

นำ PIR (Motion) Sensor ที่ใช้งานมาวัดสัญญาณที่เปลี่ยนแปลง ในขณะที่มีคนเดินผ่านและไม่มีคนเดินผ่าน ในระยะ 5 ฟุต 10 ฟุต 15 ฟุต 20 ฟุต 25 ฟุต และ 30 ฟุตโดยวัดสัญญาณที่ได้จากขา Output

3.2.1.5 ทดสอบการใช้งานเซ็นเซอร์ ร่วมกับ โมดูล X-Bee และ ไมโครคอนโทรลเลอร์ ARM7

1.) นำ X-Bee ทำหน้าที่ Coordinator ต่อเข้ากับ RS232 ของ ARM7 Cortex-M3 โดยโปรแกรม ARM7 ให้รองรับการทำงานคอยอ่านค่าจาก Packet ที่ได้รับจาก Coordinator และ แสดงผลผ่านจอ LCD

2.) กำหนดหน้าที่ของโมดูล X-Bee โดย 1 ตัวสำหรับทำหน้าที่เป็น Coordinator และ 3 ตัวสำหรับทำหน้าที่เป็น Router โดยกำหนดค่าพารามิเตอร์สำคัญๆดังนี้

1.1) Coordinator กำหนด DH = 0000 และ DL = FFFF

1.2) Router1 กำหนด DL = 0000 และ DH = 0000 ทำหน้าที่เป็น Router

1.3) Router2 กำหนด DL = 0000 และ DH = 0000, IR=4E20 โดยต่อพอร์ต AD0 กับตัวต้านทานปรับค่าได้ เพื่อรับค่า Analog Input ทุกๆ 20 วินาที ซึ่งโดยปกติแล้วสามารถตั้งค่าได้มากสุด IR=FFFF เท่ากับ 65535 วินาที หรือประมาณ 1 ชั่วโมง

1.4) Router3 กำหนด DL = 0000 และ DH = 0000, IC=FFFF โดยต่อพอร์ต DIO0 กับขา Output ของ PIR Sensor เพื่อตรวจจับในกรณีมีการเปลี่ยนแปลงสถานะที่ขา DIO0 จาก Low เป็น High หรือจาก High เป็น Low

3.) กำหนดจุดวาง Router2 และ Router3 ไม่ให้สามารถสื่อสารกับ Coordinator ได้โดยตรง โดยวางระยะให้ไกลออกจาก Coordinator จนไม่สามารถสื่อสารกันได้ หลังจากนั้นจึงนำ Router1 มาวางตรงกลางระหว่าง Router 2 และ Router3

4.) บันทึกค่า ในทุกๆ 20 วินาทีสำหรับ ค่า Analog โดยการเปลี่ยนแปลง Voltage ที่ขา AD0 ผ่านตัวต้านทานปรับค่าได้ โดยกำหนด Voltage ให้มีค่า 0V, 0.3V, 0.6V, 0.9V และ 1.2V ตามลำดับ ทำการบันทึกระดับ Bit Level ที่ปรากฏบนหน้าจอ LCD

5.) บันทึกค่า การตรวจจับคลื่นรังสีอินฟราเรด โดยมีการเคลื่อนไหวหน้า Sensor และ ไม่มีการเคลื่อนไหวหน้า Sensor ทำการบันทึกการตรวจจับที่ปรากฏบนหน้าจอ LCD

3.2.2 ผลการทดสอบโครงงาน

3.2.2.1 ผลการทดสอบการส่งข้อมูลแบบ Star Topology

สถานที่	ข้อมูลต้นทาง ขนาด 1 byte	ข้อมูลที่ได้รับในระยะทางต่างๆ					
		10เมตร	20เมตร	30เมตร	40เมตร	50เมตร	60เมตร
กลางแจ้ง	0x7E	0x7E	0x7E	0x7E	0x7E	0x7E	ไม่ได้รับข้อมูล
ในตึกที่มีสิ่งกีดขวาง	0x7E	0x7E	0x7E	0x7E	ไม่ได้รับข้อมูล	ไม่ได้รับข้อมูล	ไม่ได้รับข้อมูล

3.2.2.2 ผลการทดสอบการส่งข้อมูลแบบ Tree Topology

เหตุการณ์	ข้อมูลต้นทาง Coordinator ขนาด 1 byte	ข้อมูลปลายทาง Router ที่ได้รับ
Coordinator และ End Device สามารถติดต่อกันได้	0x7E	0x7E
แยก Coordinator และ End Device ออกจากกัน	0x7E	ไม่ได้รับข้อมูล
End Device ติดต่อ Coordinator ผ่าน Router	0x7E	0x7E

3.2.2.3 ผลการทดสอบการส่งข้อมูลแบบ Mesh Topology

เหตุการณ์	ข้อมูลต้นทาง Coordinator ขนาด 1 byte	ข้อมูลปลายทาง Router ที่ได้รับ
Coordinator (A) ติดต่อกับ Router (B)	0x7E	0x7E
แยก Coordinator (A) และ Router (B)	0x7E	ไม่ได้รับข้อมูล
นำ Router (C) และ Router (D) มาวางตรงกลางระหว่าง Coordinator (A) และ Router (B)	0x7E	0x7E
ปิด Router (C) เปิด Router (D)	0x7E	0x7E
ปิด Router (D) เปิด Router (C)	0x7E	0x7E

3.2.2.4 ผลการทดสอบการตอบสนอง PIR (Motion) Sensor

เหตุการณ์	ลอจิกที่เกิดขึ้นที่ขา Output ของ PIR (Motion) Sensor	
	เมื่อมีคนเดินผ่าน	เมื่อไม่มีคนเดินผ่าน
ระยะทาง 5 เมตร	High	Low
ระยะทาง 10 เมตร	High	Low
ระยะทาง 15 เมตร	High	Low
ระยะทาง 20 เมตร	Low	Low
ระยะทาง 25 เมตร	Low	Low
ระยะทาง 30 เมตร	Low	Low

3.2.2.5 ผลการทดสอบใช้งานเซนเซอร์ ร่วมกับ โมดูล X-Bee และ ไมโครคอนโทรลเลอร์ ARM7

ก) ทดสอบการส่งข้อมูลการเปลี่ยนแปลงขา AD0 จาก Router2 ทุกๆ 20 วินาที

	0 V	0.3 V	0.6 V	0.9 V	1.2 V
	วินาทีที่ 20	วินาทีที่ 40	วินาทีที่ 60	วินาทีที่ 80	วินาทีที่ 100
ค่า Bit Level ที่ปรากฏบนจอ LCD	0	254	513	767	1023

ข) ทดสอบการการตรวจจับคลื่นรังสีอินฟราเรด ผ่านการเปลี่ยนแลงขา DO0 จาก Router3

	ค่าบนจอ LCD ที่ปรากฏ
มีการเคลื่อนไหวผ่านหน้า Sensor ระยะ 8 เมตร	Detect
ไม่มีการเคลื่อนไหวหน้า Sensor ระยะ 8 เมตร	None

3.3 สรุปผลการทดสอบโครงงานเบื้องต้น

จากการทดสอบหัวข้อ 3.2.2.1 มีจุดประสงค์เพื่อหาระยะทางของโมดูล X-Bee ที่สามารถสื่อสารกันได้จริง โดยในการทดสอบเป็นการสื่อสารกันระหว่างโมดูล X-Bee Series2 จำนวน 2 ตัว สื่อสารกันแบบ Star Topology ซึ่งทำหน้าที่เป็น Coordinator และ End Device ผลของการทดสอบนั้นพบว่า เมื่อส่งข้อมูลหากันในบริเวณภายในตึก ซึ่งมีกำแพงเป็นตัวกั้นนั้น สามารถส่งข้อมูลหากันได้ระยะไกลสุด 30 เมตร และบริเวณที่เป็นพื้นที่โล่งแจ้ง สามารถส่งข้อมูลหากันได้ระยะไกลสูงสุด 50 เมตรซึ่งผิดไปจากตาราง 2.2 ซึ่งเป็นตารางคุณสมบัติ X-Bee จะสามารถส่งข้อมูลหากันได้ระยะไกลสุด 120 เมตร

การทดสอบหัวข้อ 3.2.2.2 เป็นการทดสอบ โครงข่ายสื่อสารแบบ Tree Topology ซึ่งจากการทดสอบนั้นเป็นการนำโมดูล X-Bee จำนวน 2 ตัวให้แยกอิสระจากกัน แล้วทำการวางโมดูล X-Bee ที่ทำหน้าที่เป็น Router ไว้บริเวณตรงกลางระหว่าง โมดูล X-Bee 2 ตัว ซึ่งผลการทดสอบนั้น สามารถส่งข้อมูลจาก End Device ไปยัง Coordinator ผ่าน Router ได้ตามทฤษฎีโครงข่ายการสื่อสารแบบ Tree Topology

การทดสอบหัวข้อ 3.2.2.3 เป็นการทดสอบ โครงข่ายแบบ Mesh Topology โดยการทำ Coordinator และ Router จุดปลายทางไว้ ไม่ให้สามารถให้สื่อสารกันได้ แต่ให้สื่อสารผ่าน Router 2 จุดที่อยู่จุดกึ่งกลางระหว่าง Coordinator และ Router โดยสลับการเลือกเส้นทาง โดยการสลับเปิดปิด Router ที่อยู่จุดกึ่งกลางระหว่าง 2 จุด จะพบว่า เมื่อปิด Router จุดใดจุดหนึ่ง ข้อมูลก็ยังสามารถส่งข้อมูลไปยังต้นทางได้ โดยการเลือกเส้นทางการส่งข้อมูลกับ Router ตัวอื่นที่อยู่บริเวณใกล้เคียง ซึ่งสามารถส่งข้อมูลแบบ Mesh Topology ได้ตามทฤษฎี Zigbee ที่กล่าวไว้

การทดสอบหัวข้อ 3.2.2.4 เป็นการทดสอบการใช้งาน PIR (Motion) Sensor ซึ่งเป็นการวัดการตอบสนองของ PIR (Motion) Sensor ในระยะต่างๆ ซึ่งจากผลการทดสอบจะเห็นว่า PIR (Motion) Sensor สามารถตอบสนองได้ไกลมากที่สุด 15 เมตรเมื่อมีการเปลี่ยนแปลงรังสีอินฟราเรดผ่านบริเวณหน้า PIR (Motion) Sensor

ในการทดสอบหัวข้อสุดท้าย 3.2.2.5 เป็นการทดสอบระบบตัวอย่าง ซึ่งจะเห็นว่า โมดูล X-Bee Router2 สามารถให้ค่า Analog ส่งกลับมายัง Coordinator และตีความข้อมูลที่ได้รับมาผ่าน ARM7 Cortex M3 ได้ใกล้เคียงกับค่าที่เป็นจริงมาก และในส่วนของการตรวจจับคลื่นรังสีอินฟราเรดนั้น สามารถทำงานการตอบสนองได้ดีในช่วง 1-8 เมตร ซึ่งในจากการทดสอบหัวข้อ ข. นั้น PIR (Motion) Sensor สามารถตอบสองได้ไกลจนถึง 15 เมตร แต่ยังมีความผิดพลาดบ้างในบางครั้ง จึงสรุปได้ว่า สามารถทำงานได้ดีในย่าน 1-8 เมตร

จากการทดสอบข้างต้น จะเห็นว่าโมดูล X-Bee Series2 นั้น สามารถส่งข้อมูลระยะไกลสุดได้ 50 เมตร ซึ่งในความเป็นจริงนั้นระบบที่ต้องการสร้างขึ้น อาจจะใช้พื้นที่บริเวณกว้าง ซึ่งอาจจะมีปัญหาเกี่ยวกับระยะทางการส่ง

ข้อมูลได้ ดังนั้นจึงแนะนำให้เลือกใช้ โมดูล X-Bee-Pro Series2 จะดีกว่า เนื่องจากสามารถส่งข้อมูลได้ระยะทางที่ไกลกว่า แต่เมื่อเปรียบเทียบคุณสมบัติระหว่าง X-Bee และ X-Bee-Pro นั้น จะเห็นว่า X-Bee-Pro จะใช้พลังงานมากกว่า X-Bee จึงเป็นข้อเสีย ซึ่ง ถ้าต้องการระยะทางที่มากขึ้น จะต้องแลกกับการบำรุงรักษาที่ใช้เวลาบ่อยกว่า ระยะทางที่สั้นกว่า แต่การบำรุงรักษาจะน้อยครั้งกว่า

สำหรับการใช้งาน PIR (Motion) Sensor นั้น จะเห็นได้ว่ารัศมีการตอบสนองของตัวเซนเซอร์นั้น มีรัศมีค่อนข้างไกล จึงเหมาะกับการติดตั้งที่บริเวณกลางสถานที่ เช่น กลางบ้าน กลางห้อง โดยใช้วิธีการติดตั้งบนเพดานจะเหมาะสมมากที่สุด

3.4 บทสรุป

ในบทที่ผ่านมาได้บรรยายและอธิบายถึงการทำงานโดยรวมของโครงงานเบื้องต้น ที่มีความสามารถในการรับ-ส่งข้อมูลผ่านเครือข่าย Zigbee, การตั้งค่าต่างๆสำหรับโมดูล X-Bee, การทดลองใช้งานโมดูล X-Bee ร่วมกับไมโครคอนโทรลเลอร์ ARM7 รวมถึงการประยุกต์ใช้เข้ากับเซ็นเซอร์ต่างๆ, การทดสอบการรับ-ส่งข้อมูลในรูปแบบเครือข่ายแบบต่างๆ เช่น แบบสตาร์, แบบต้นไม้ และ แบบเมซ สุดท้ายเป็นการทดสอบและสรุปผลการทดสอบในการใช้งานโมดูล X-Bee ทั้งหมดที่กล่าวมานี้ ทำให้ทราบว่า ประสิทธิภาพของโมดูล X-Bee มีความเหมาะสมที่นำมาใช้งานในการรับ-ส่งข้อมูลในรูปแบบของครือข่ายในระบบของโครงงานเบื้องต้นที่ได้ออกแบบไว้เพื่อนำมาพัฒนาต่อไปได้ในอนาคต

บทที่ 4
การออกแบบโครงงานต้นแบบ

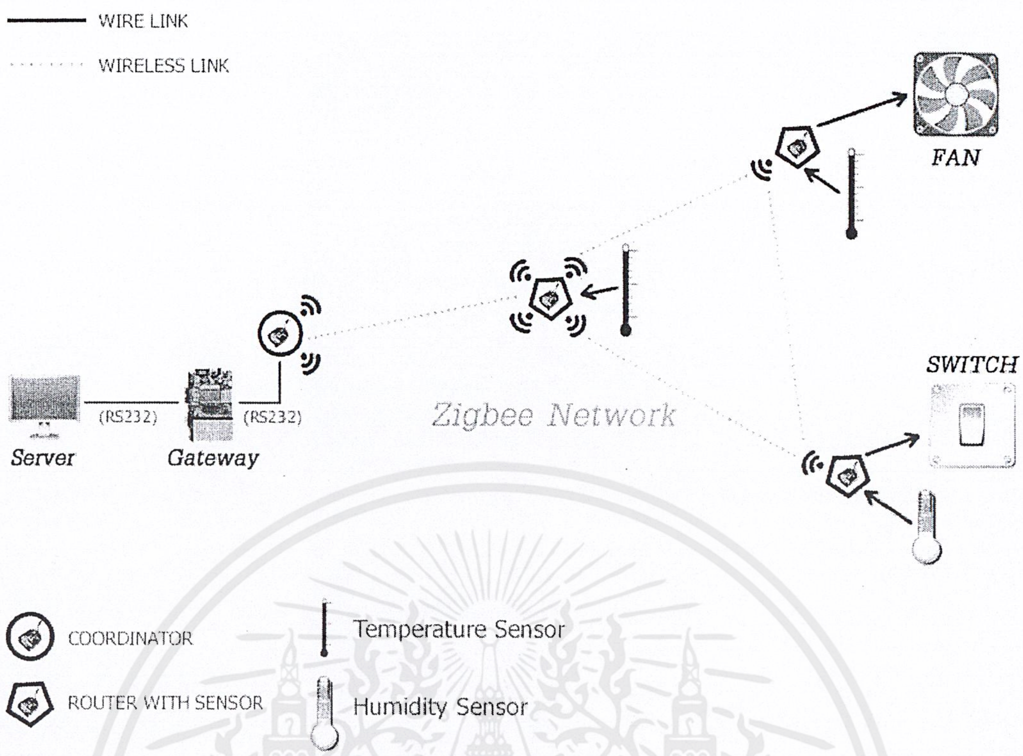
4.1 ระบบการทำงานของโครงงานต้นแบบ

ระบบการทำงานของโครงงานต้นแบบ แสดงดังรูปที่ 4.1 จะเห็นว่าระบบโครงงานต้นแบบนี้เป็นระบบ Wireless Network Sensor ซึ่งรับ-ส่งข้อมูลผ่านทางเครือข่าย Zigbee โดยทำงานร่วมกับไมโครคอนโทรลเลอร์ ARM7 Cortex-M3 ซึ่งทำหน้าที่เป็น gateway โดยเชื่อมต่อกับหน่วยประมวลผลระบบหรือเซิร์ฟเวอร์ ผ่านทางพอร์ต RS-232 โดยเซิร์ฟเวอร์ของระบบต้นแบบนั้น จะมีโปรแกรมเฝ้าระวังและควบคุม คอยทำหน้าที่แสดงผลภาพรวมของระบบเพื่อทำการเก็บข้อมูลทางวิทยาศาสตร์และเทคโนโลยี หรือ ข้อมูลเฉพาะเจาะจง ที่ต้องการศึกษาในเรื่องนั้นๆ เช่น การเก็บข้อมูลอุณหภูมิ หรือ ความชื้น ในระบบปิด เพื่อที่จะนำไปใช้ประโยชน์ในด้านต่างๆ ได้ในอนาคตต่อไป

ข้อมูลที่ได้จากเซ็นเซอร์ตามโหนดปลายทางของระบบที่อยู่บริเวณต่างๆนั้น จะนำมาประมวลผลผ่านโปรแกรมเฝ้าระวังและควบคุม หากข้อมูลเหล่านั้นมีการเปลี่ยนแปลงที่ ต่ำกว่า หรือ สูงกว่า ตามค่าที่ถูกกำหนดไว้ จะมีการส่งสัญญาณบางอย่างออกไปควบคุม ณ จุดปลายทางเหล่านั้น เพื่อทำให้เกิดสภาวะที่เหมาะสม ทำให้ผู้ใช้งานสามารถรักษาสภาวะ ในระบบปิดได้โดยง่าย ผ่านทางโปรแกรมเฝ้าระวังและควบคุม นอกจากนี้ระบบโครงงานต้นแบบ สามารถรองรับฟังก์ชั่นการทำงาน เฝ้าระวัง และ ควบคุม ได้หลายรูปแบบ เช่น การเปลี่ยนอุปกรณ์การวัดเป็นแบบอื่น หรือ การเปลี่ยนแปลงการควบคุมอุปกรณ์ปลายทางต่างๆ ได้โดยง่าย ทำให้สามารถเก็บข้อมูลทางวิทยาศาสตร์และเทคโนโลยีได้หลายรูปแบบ

ในการเก็บข้อมูลทางด้านวิทยาศาสตร์และเทคโนโลยีนั้น จำเป็นที่จะต้องเก็บข้อมูลในระยะยาว เพื่อนำข้อมูลเหล่านั้น มาแสดงผลในรูปแบบต่างๆ เช่น กราฟ หรือ แสดงผลผ่านเว็บไซต์ ได้โดยง่าย เนื่องจาก เซิร์ฟเวอร์กลาง ที่ใช้ในการประมวลผลกลางของระบบนั้น ยังสามารถทำหน้าที่เป็นเซิร์ฟเวอร์ในลักษณะอื่นๆ เช่นในโครงงานต้นแบบนี้ สามารถนำข้อมูลทางด้านวิทยาศาสตร์และเทคโนโลยีเก็บในลักษณะเป็นฐานข้อมูลออนไลน์ และ สามารถตรวจสอบข้อมูลเหล่านั้น ได้ผ่านทางเครือข่ายอินเตอร์เน็ตอีกด้วย

ระบบของโครงงานต้นแบบนั้นยังสามารถปรับเปลี่ยนใช้งานได้หลายรูปแบบ ยกตัวอย่างเช่น ใช้ร่วมกับระบบรักษาความปลอดภัยภายในอาคารหรือสำนักงาน, ใช้ร่วมกับระบบดูแลการผลิตภายในโรงงาน หรือ ใช้กับงานด้านเกษตรกรรม เป็นต้น ซึ่งในโครงงานนี้จะนำระบบโครงงานต้นแบบมายกตัวอย่างใช้ในงานด้านเกษตรกรรม ซึ่งงานด้านเกษตรกรรมนั้นส่วนใหญ่ตัวแปรสำคัญของงานด้านนี้คือ อุณหภูมิ ความชื้น และแสงแดด ที่มีผลต่อการเจริญเติบต่อของพืช ตัวอย่างเช่น การเพาะเห็ดในโรงเรือน เนื่องจากการปลูกเห็ดในโรงเรือนปิดนั้นสามารถควบคุมอุณหภูมิกับความชื้นได้ดีกว่าโรงเรือนแบบเปิด ซึ่งจะทำให้เราสามารถเก็บข้อมูลสภาพอากาศภายในโรงเรือน และควบคุมปริมาณความชื้นและอุณหภูมิให้คงที่ได้ดีกว่านั้นเอง



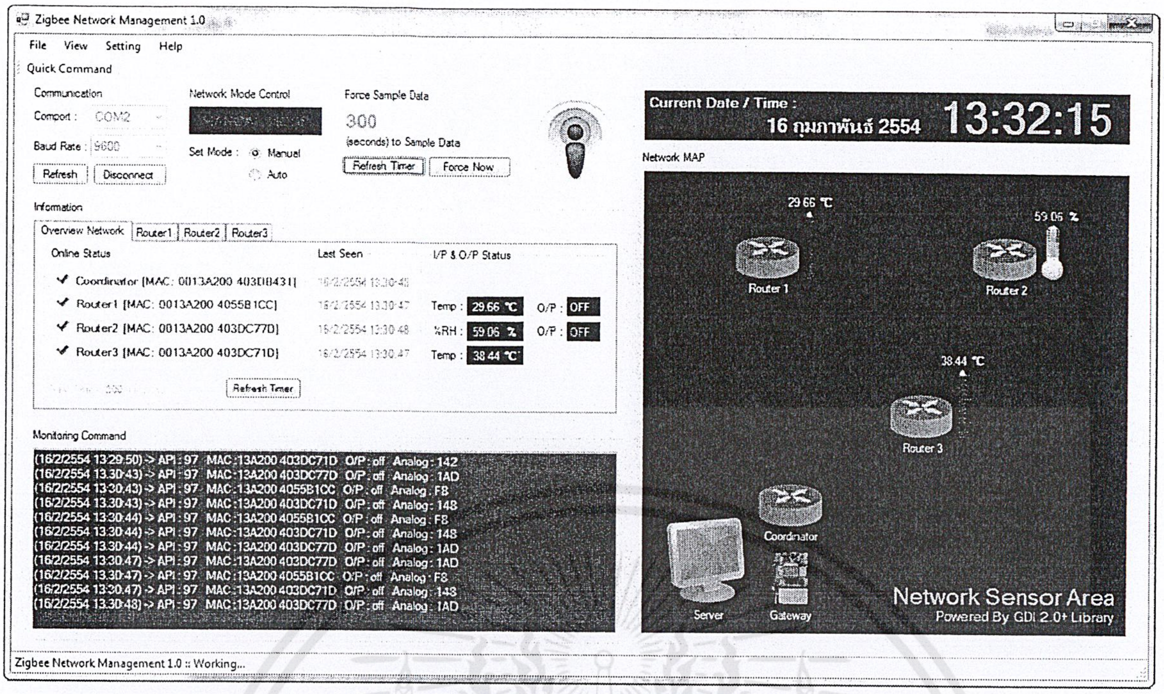
รูปที่ 4.1 ระบบในโครงงานต้นแบบ

ในระบบโครงงานต้นแบบนั้นจะมีส่วนหลักๆอยู่ 3 ส่วนคือดังนี้

4.1.1 หน่วยประมวลผลระบบ (Server)

หน่วยประมวลผลระบบ (Server) สามารถแบ่งออกเป็น 2 ส่วนหลักๆ ได้แก่ โปรแกรมเฝ้าระวังและควบคุม และ ในส่วนของเว็บเซิร์ฟเวอร์ ดังรูปที่ 4.3 โดยการพัฒนาโปรแกรมเฝ้าระวังและควบคุมนั้น ได้เลือกใช้ภาษา C#.NET ในการเขียน GUI (Graphic User Interface) ขึ้นมาให้ง่ายต่อการใช้งาน ในส่วนของการทำงานนั้น โปรแกรมเฝ้าระวังและควบคุม สามารถสื่อสารระหว่าง ข้อมูลที่ได้รับจาก Gateway ในระบบ และ ส่งข้อมูลบางอย่างเข้าสู่ระบบ เพื่อควบคุมการทำงานของอุปกรณ์ต่างๆได้ดังรูปที่ 4.2 นอกจากนี้ โปรแกรมเฝ้าระวังและควบคุม ยังมีหน้าที่ในการเก็บข้อมูลที่ได้รับจาก โหนดปลายทางต่างๆที่ส่งเข้ามา เพื่อทำสร้างระบบฐานข้อมูล ที่สามารถนำข้อมูลเหล่านี้มาประมวลผลได้ในอนาคต ในการเก็บข้อมูลได้เลือกใช้ซอฟท์แวร์ MySQL สำหรับจัดการฐานข้อมูล

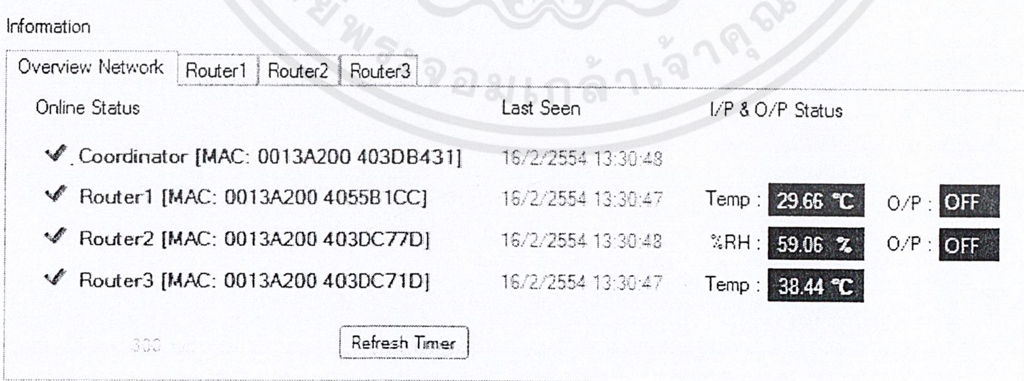
ดังนั้นในการพัฒนาโปรแกรมเฝ้าระวังและควบคุม จึงต้อง พัฒนาให้สามารถสื่อสารกับซอฟท์แวร์การจัดการฐานข้อมูลได้ด้วย นอกจากนั้น เพื่อให้สามารถแสดงผลข้อมูลย้อนหลัง หรือข้อมูลที่เก็บอย่างต่อเนื่องเป็นระยะเวลานาน ผ่านทางเว็บไซต์ ในระบบอินเตอร์เน็ต เซิร์ฟเวอร์ของระบบจึงได้ติดตั้งโปรแกรม Apache เพื่อให้ทำหน้าที่เป็นเว็บเซิร์ฟเวอร์ได้ด้วย



รูปที่ 4.2 แสดง GUI โปรแกรมเฝ้าระวังและควบคุมที่ได้ออกแบบไว้

จากรูปที่ 4.2 โปรแกรมเฝ้าระวังและควบคุมนั้น มีฟังก์ชั่นการทำงานที่หลากหลาย ซึ่งสามารถอธิบายการทำงานเป็นส่วนๆ ได้ดังนี้

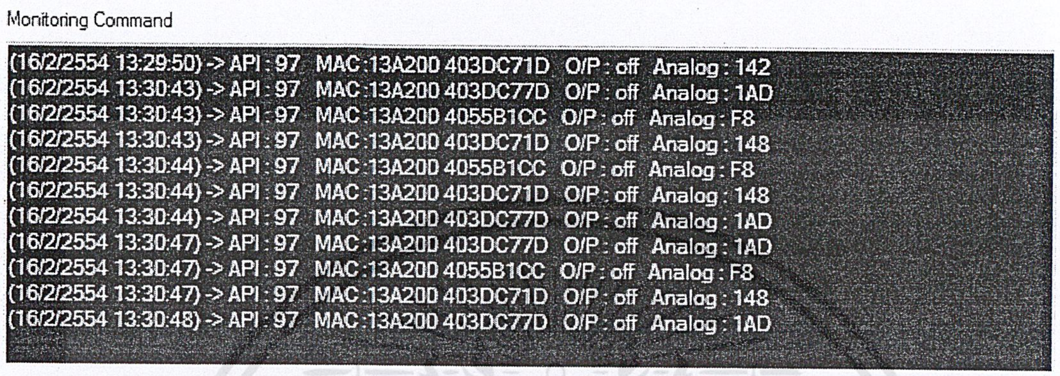
ส่วนแรก เป็นการแสดงผลภาพรวมของระบบ จากรูปที่ 4.3 เป็นการแสดงผลข้อมูลที่ได้ล่าสุด จากโหนดต่างๆ เพื่อนำมาแสดงผลให้กับผู้ใช้อย่างคร่าวๆ โดยข้อมูลเหล่านั้น จะมีเวลาระบุ ทำให้สามารถตรวจสอบได้ว่ามีความผิดพลาดในการทำงานของแต่ละโหนดหรือไม่ นอกจากนี้ ผู้ใช้ยังสามารถเลือกแถบเมนูระบุแสดงรายละเอียดเฉพาะโหนดนั้นๆ เพื่อดูข้อมูลเฉพาะเจาะจง เช่น ข้อมูลระดับบิตที่ได้จาก A/D Converter, ข้อมูลดิบที่ผ่านการคำนวณแล้ว ตลอดจนสถานะทางด้านเอาท์พุท ของโหนดนั้นๆ อุปกรณ์ที่เชื่อมต่ออยู่นั้น อยู่ในสถานะเปิดหรือปิด



รูปที่ 4.3 แสดง GUI โปรแกรมเฝ้าระวังและควบคุมในส่วนการแสดงภาพรวมของระบบ

ส่วนที่สอง เป็นการแสดงผลข้อมูลทางด้านเทคนิคที่เกิดเหตุการณ์ต่างๆขึ้นกับระบบ จากรูปที่ 4.4 ข้อมูลที่แสดง จะอยู่ในรูปของตัวอักษร เช่น วันและเวลาที่เกิดเหตุการณ์นั้นๆ, แพ็กเก็ตข้อมูลที่ได้รับ, ลักษณะการทำงาน

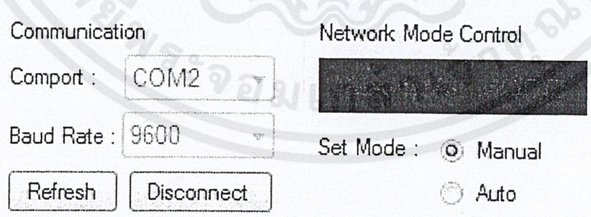
ของแพ็กเก็ตเฟรมนั้นๆ, ที่อยู่ของโหนดปลายทางที่ส่งมา ตลอดจน สถานะข้อมูล Analog หรือ Digital ในขณะนั้น นอกจากการแสดงผลในลักษณะดังข้างต้นแล้ว โปรแกรมเฝ้าระวังและควบคุม จะมีการเก็บข้อมูลเหล่านี้ ในรูปแบบของ Text File ทำให้สามารถตรวจสอบข้อมูลย้อนหลังผ่านไฟล์เหล่านั้นผ่านการเปิดไฟล์เหล่านั้น โดยไม่ต้องลงซอฟท์แวร์อื่นๆเพิ่มเติม นอกจากเก็บข้อมูลในรูปแบบ Text File แล้ว โปรแกรมเฝ้าระวังและควบคุมยังเก็บข้อมูลลงในฐานข้อมูลอัตโนมัติ เพื่อผู้ใช้งานผ่านเว็บไซต์ สามารถเห็นข้อมูลเหล่านั้นด้วยเช่นกัน



รูปที่ 4.4 แสดง GUI โปรแกรมเฝ้าระวังและควบคุมในส่วนของหน้าจอบันทึกเหตุการณ์

ส่วนที่สาม เป็นส่วนของโหมดการทำงานของระบบ ซึ่งสามารถกำหนดได้จากรูป 4.5 โดยการทำงานของระบบนั้นจะมีอยู่สองส่วนคือ การทำงานแบบอัตโนมัติ และ การทำงานแบบมีผู้ควบคุม

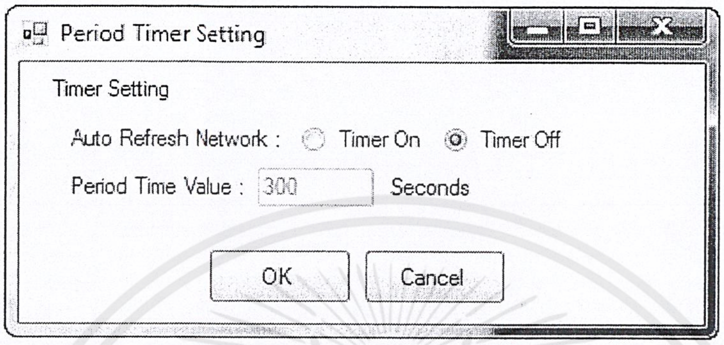
เมื่อระบบเข้าสู่โหมดการทำงานแบบอัตโนมัตินั้น ข้อมูลที่ที่ได้รับจากโหนดต่างๆ ที่มีอุปกรณ์ควบคุมต่ออยู่ เมื่อส่งข้อมูลมายังโปรแกรมเฝ้าระวังและควบคุมแล้วพบว่า ข้อมูลเหล่านั้น มากกว่า หรือ น้อยกว่าค่าที่ถูกกำหนดไว้ โปรแกรมเฝ้าระวังและควบคุม จะส่งสัญญาณบางอย่างออกไป ผ่าน Gateway ของระบบเพื่อให้แม่ข่ายของระบบนั้น ส่งสัญญาณเหล่านั้นไปยังโหนดปลายทาง เพื่อทำงานเปิดหรือปิดอุปกรณ์ควบคุมอัตโนมัติ ทำให้ระบบสามารถควบคุมสภาพแวดล้อมที่อยู่ในระบบปิด ให้กลับมาอยู่ในสภาวะที่เหมาะสม ที่เราต้องการได้



รูปที่ 4.5 แสดง GUI โปรแกรมเฝ้าระวังและควบคุมในส่วนของโหมดการทำงานของระบบ

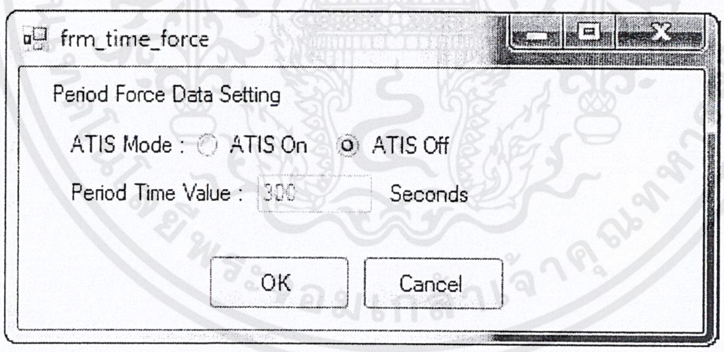
หากการทำงานของโปรแกรมเฝ้าระวังและควบคุม อยู่ในโหมดการทำงานแบบมีผู้ควบคุม ระบบจะสามารถทำงานในลักษณะของมอนิเตอร์ของระบบเพียงอย่างเดียว หากผู้ควบคุมเห็นการเปลี่ยนแปลงของสภาวะแวดล้อมที่ไม่ปกติ สามารถส่งสัญญาณไปควบคุมอุปกรณ์ตามโหนดนั้นๆ ด้วยเพียงการกดปุ่มสั่งงาน ให้สถานะลอจิกทางด้านเอาท์พุทของอุปกรณ์เหล่านั้น เป็นแบบเปิดหรือปิด ได้โดยตรง

ส่วนที่สี่เป็นการกำหนดเวลาในการทำงานฟังก์ชั่นต่างๆ เช่น ระบบมีการใช้งานในระยะยาว บางโหนดของระบบ อาจจะเกิดความผิดพลาดใดๆ ที่ไม่สามารถทำงานได้ จึงจำเป็นอย่างยิ่ง ที่จะต้องมีการตรวจสอบสถานะของระบบอย่างน้อย 12 ชั่วโมง ต่อ 1 ครั้ง หรือ 1 วัน ต่อ 1 ครั้ง เป็นต้น เพื่อทำการดูแต่ละโหนดว่ายังทำงานได้ตามปกติหรือไม่ ซึ่งการสามารถกำหนดได้จาก แถบเมนู Setting > Timer Setting > Period Network Refresh ซึ่งจะเป็นการตั้งค่าเวลาที่จะทำการตรวจสอบระบบ 1 ครั้ง ดังรูปที่ 4.6



รูปที่ 4.6 แสดง GUI โปรแกรมเฝ้าระวังและควบคุมในส่วนการตั้งค่าเวลาในการตรวจสอบสถานะ

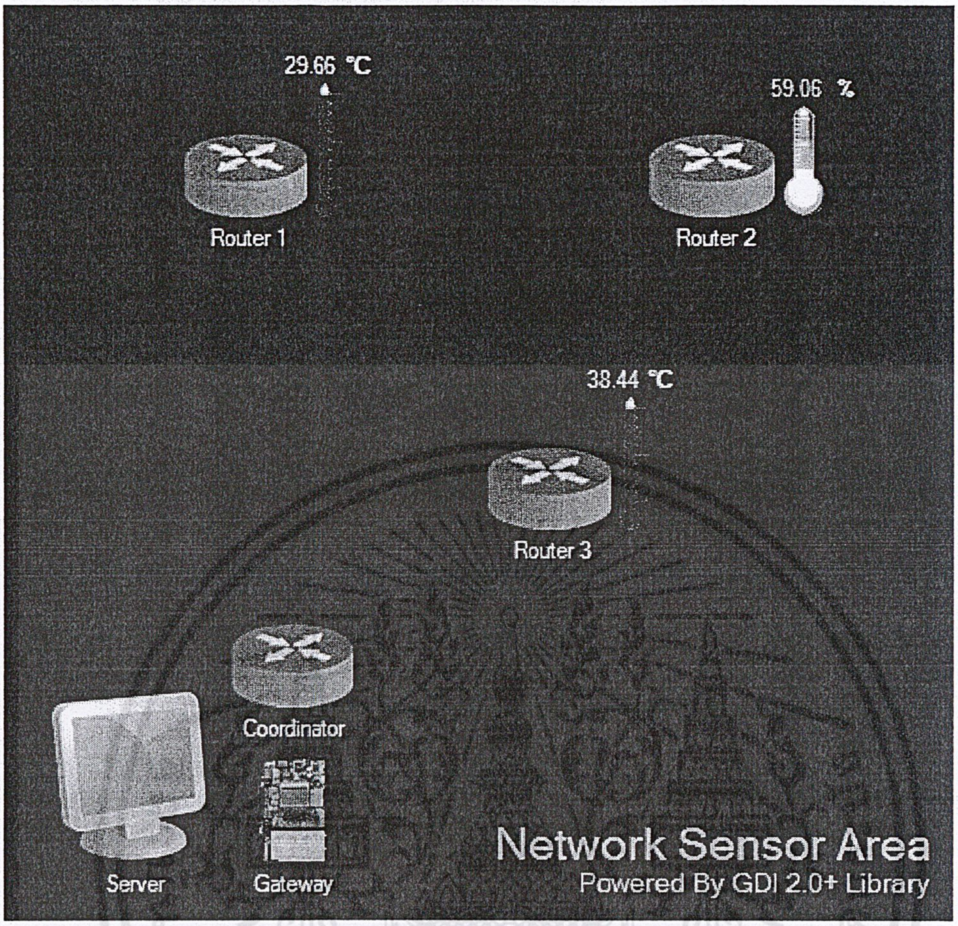
นอกจากนี้ ยังมีการกำหนดเวลาทางด้านอื่นๆอีก เช่น ระยะเวลาในการเก็บข้อมูลเพื่อนำมาแสดงผล เช่น ระบบเก็บข้อมูล ทุกๆ 24 ชั่วโมง หรือ ทุกๆ 1 อาทิตย์เป็นต้น ซึ่งสามารถกำหนดเวลาเหล่านี้ได้จาก แถบเมนู Setting > Timer Setting > Period Force Data ซึ่งจะเป็นการตั้งค่าเวลาที่จะเก็บข้อมูล 1 ครั้งตามเวลาที่ถูกกำหนด ดังรูปที่ 4.7



รูปที่ 4.7 แสดง GUI โปรแกรมเฝ้าระวังและควบคุมในส่วนการตั้งค่าเวลาในการเก็บข้อมูลในระบบ

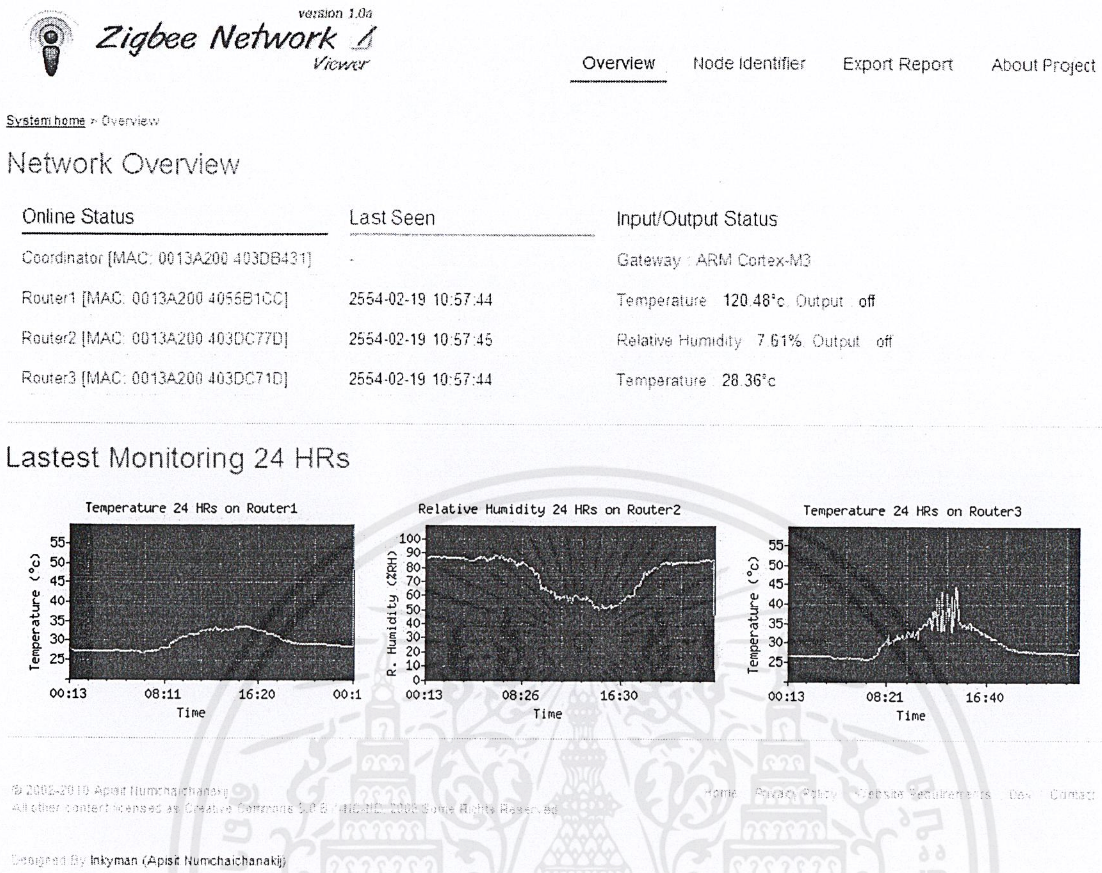
จากรูปที่ 4.8 เป็นการแสดงให้สามารถเห็นภาพรวมของระบบ ว่าแต่ละโหนดมีการเชื่อมต่อเซ็นเซอร์แบบใด และอยู่จุดใดของพื้นที่ ซึ่งสามารถคลิกที่โหนดต่างๆ เพื่อเห็นข้อมูลสถานะของโหนดเหล่านั้นได้

Network MAP



รูปที่ 4.7 แสดง GUI โปรแกรมเฝ้าระวังและควบคุมในส่วนการแสดงผลในรูปของกราฟฟิค

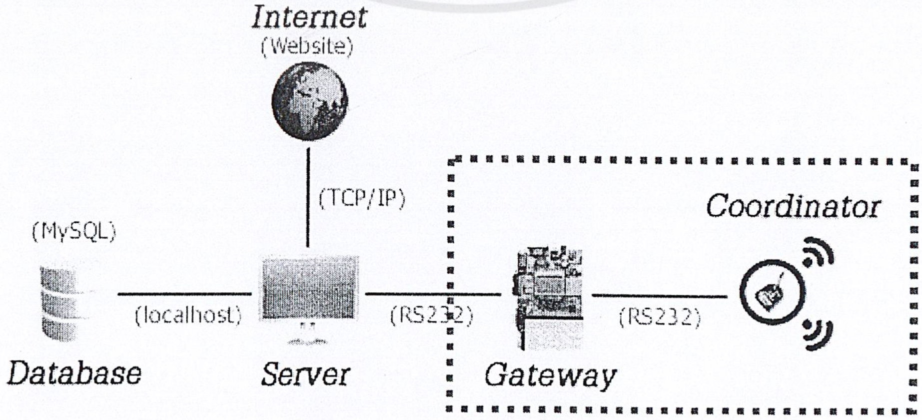
ในการเข้าถึงข้อมูลโดยผ่านทางเว็บเซิร์ฟเวอร์ สุดท้ายจากรูปที่ 4.8 แสดงหน้าต่างเว็บไซต์ที่ได้ออกแบบไว้ ซึ่งแสดงผลข้อมูลที่จำเป็น หรือ แสดงผลในรูปของกราฟก็ตาม ให้ผู้ใช้งานสามารถเข้าถึงข้อมูลเหล่านั้นได้โดยง่าย โดยในเว็บไซต์ของระบบที่ได้ออกแบบไว้ นอกจากแสดงผลเพียงอย่างเดียวแล้ว ยังสามารถนำข้อมูลที่ถูกเก็บอยู่ใน ฐานข้อมูล ส่งออกตามช่วงเวลาที่ผู้ใช้ต้องการได้ โดยทำการส่งออกอยู่ในรูปไฟล์ตาราง Excel ซึ่งข้อมูลเหล่านั้น สามารถนำไปใช้ในการวิเคราะห์ทางวิทยาศาสตร์และเทคโนโลยี ต่อไปได้ในอนาคต



รูปที่ 4.8 แสดงหน้าต่างเว็บไซต์ที่ได้ออกแบบไว้

4.1.2 หน่วยประมวลผลเครือข่าย (Gateway and Coordinator)

ในส่วนของหน่วยประมวลผลเครือข่าย (Gateway and Coordinator) นี้จะประกอบด้วยอุปกรณ์ดังรูปที่ 4.9 ซึ่งจะมีโมดูล X-Bee ทำหน้าที่เป็นโหนดแม่ข่าย (Coordinator) ในการสร้างเครือข่าย Zigbee ขึ้นมา ต่อกับไมโครคอนโทรลเลอร์ ARM7 Cortex-M3 ซึ่งจะเป็นตัวรับ-ส่ง และประมวลผลข้อมูลที่ได้รับจากเครือข่าย Zigbee ผ่านทางโหนดแม่ข่าย ไปยัง โปรแกรมเฝ้าระวังและควบคุม ที่รอรับ-ส่ง ข้อมูลจากเซิร์ฟเวอร์ของระบบ



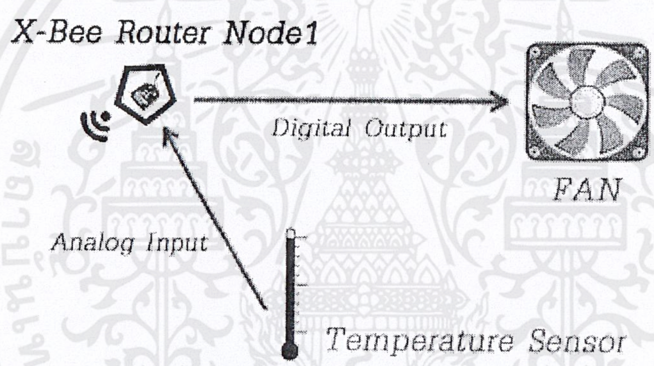
รูปที่ 4.9 หน่วยประมวลผลเครือข่าย (Gateway and Coordinator)

4.1.3 โหนดต่างๆภายในเครือข่าย (Router)

4.1.3.1 โหนดวัดอุณหภูมิ

จากรูปที่ 4.10 ในส่วนของโหนดวัดอุณหภูมินี้จะมีโมดูล X-Bee ที่ทำหน้าที่เป็น Router ต่อกับเข้ากับเซ็นเซอร์วัดอุณหภูมิ LM35 DZ และ วงจรขับรีเลย์ซึ่งใช้สำหรับควบคุมอุปกรณ์ปรับอากาศ (ในโครงงานต้นแบบนี้จำลองเป็นพัดลม 12V)

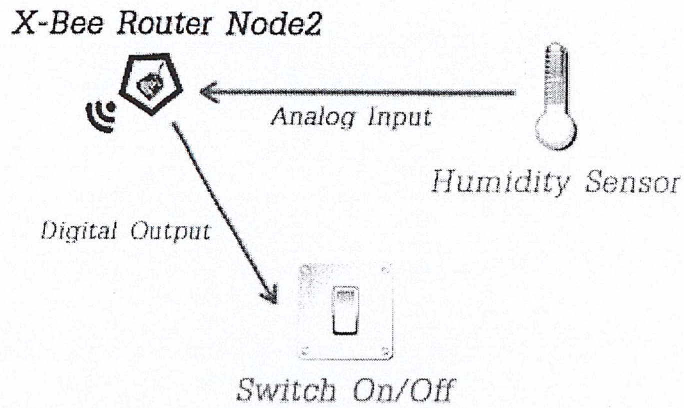
การทำงานของโหนดวัดอุณหภูมิ คือ โมดูล X-Bee จะส่งค่าที่ข้อมูลอุณหภูมิที่ได้จาก LM35 DZ โดยทางด้าน Output ที่ออกจาก LM35 DZ นั้นจะเป็นค่าของแรงดันไฟฟ้า โมดูล X-Bee จะนำค่าแรงดันที่ได้รับ ผ่าน A/D เพื่อแปลงแรงดันเป็นค่า Digital จากนั้นส่งข้อมูลไปยัง Gateway ผ่านโหนดแม่ข่าย เพื่อส่งต่อไปยังเซิร์ฟเวอร์ โดยจะถูกส่งไปยังแม่ข่ายตามคาบเวลาที่เรากำหนด เช่น ทุกๆ 5 นาที ในกรณีที่โปรแกรมเฝ้าระวังและควบคุมทำงานในโหมดอัตโนมัติ หากอุณหภูมิที่วัดได้มีค่ามากกว่าอุณหภูมิที่ถูกตั้งไว้ในระบบ ระบบจะส่งสัญญาณกลับไปยังโหนดวัดอุณหภูมิสั่งให้วงจรขับรีเลย์ทำงาน ทำให้พัดลมหมุดจนกระทั่งอุณหภูมิกลับเข้าสู่สภาวะปกติ



รูปที่ 4.10 ภาพรวมของโหนดวัดอุณหภูมิ

4.1.3.2 โหนดวัดปริมาณความชื้นในอากาศ

โหนดวัดปริมาณความชื้นในอากาศนี้จะมีโมดูล X-Bee ที่ทำหน้าที่เป็น Router ต่อกับเข้ากับโมดูลวัดความชื้น ซึ่งค่า Output ของโมดูลวัดความชื้นจะเป็นค่าแรงดันไฟฟ้าระหว่าง 0-3.3V แต่เนื่องจากโมดูล X-Bee สามารถรับค่าแรงดันที่ขาอินพุทได้ไม่เกิน 1.2V ต้องใช้วงจรลดทอนค่าแรงดันให้เหลือ 1/3 เมื่อผ่านวงจรลดทอนแรงดันไปยังอินพุทของโมดูล X-Bee จะผ่านวงจร A/D เพื่อแปลงค่าแรงดันที่ถูกลดทอนแล้วเป็นค่า Digital จากนั้นส่งข้อมูลไปยัง Gateway ผ่านโหนดแม่ข่าย เพื่อส่งต่อไปยังเซิร์ฟเวอร์ โดยค่าที่โหนดวัดความชื้นในอากาศนี้จะส่งข้อมูลไปยังแม่ข่ายตามคาบเวลาที่เรากำหนด ในกรณีที่โปรแกรมเฝ้าระวังและควบคุมทำงานในโหมดอัตโนมัติ เมื่อได้ข้อมูลความชื้นในอากาศ ที่แตกต่างจากค่าความชื้นในอากาศที่ถูกตั้งไว้ในระบบ เช่น หากค่าที่วัดได้มีค่าต่ำกว่าระบบจะมีการส่งสัญญาณกลับไปยังโหนดวัดความชื้นในอากาศนี้ เพื่อสร้างสัญญาณให้วงจรขับรีเลย์ทำงาน ทำให้อุปกรณ์เพิ่มความชื้นทำงาน จนกระทั่งความชื้นในระบบปิดกลับเข้าสู่สภาวะปกติ ซึ่งแสดงภาพรวมของระบบดังรูป 4.11

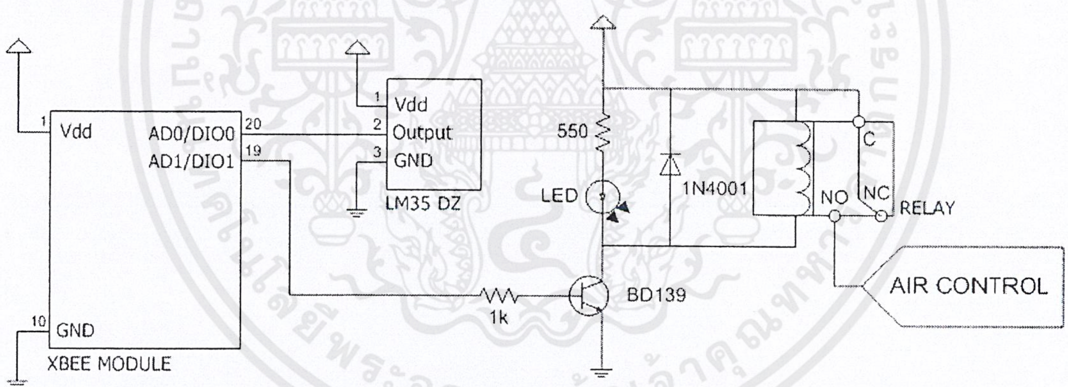


รูปที่ 4.11 ภาพรวมของโหนดวัดความชื้นในอากาศ

4.2 การออกแบบวงจร

4.2.1 โหนดวัดอุณหภูมิ

การออกแบบโหนดที่ใช้สำหรับวัดอุณหภูมินั้น จากรูปที่ 4.12 เบื้องต้นได้เลือกใช้เซ็นเซอร์วัดอุณหภูมิ LM35 DZ โดยนำขา Output ของ LM35 DZ ต่อเข้ากับ ขา Input ของโมดูล X-Bee และ นำขา Digital I/O ของโมดูล X-Bee ต่อเข้ากับวงจรขับรีเลย์ ซึ่งถูกนำไปต่อกับภาคควบคุมต่อไป (ในกรณีนี้เลือกใช้พัดลม 12V แทน)



รูปที่ 4.12 การออกแบบวงจรสำหรับโหนดวัดอุณหภูมิในเครือข่าย Zigbee

4.2.2 โหนดวัดปริมาณความชื้นในอากาศ

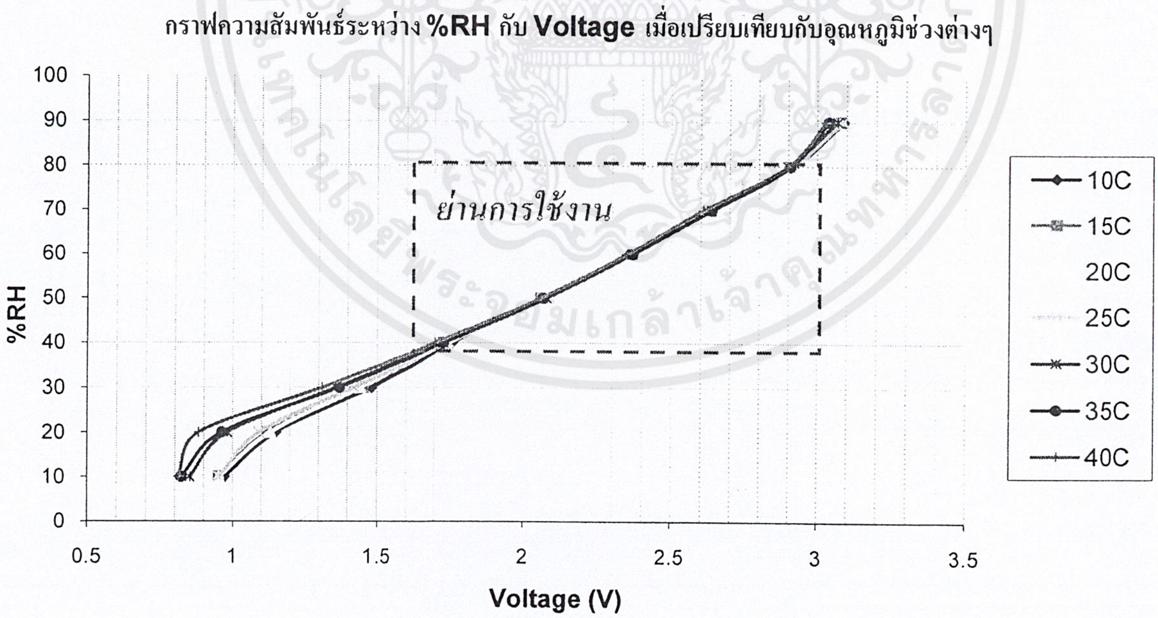
4.2.2.1 การใช้งานเซ็นเซอร์วัดความชื้น

ในการใช้งานจริง เนื่องจากเซ็นเซอร์มีข้อจำกัดในการใช้งาน จึงต้องมีการเลือกช่วงการใช้งานจาก Datasheet ของบริษัทผู้ผลิต ซึ่งได้กำหนดช่วงแรงดัน Output ที่เปลี่ยนไปตามระดับความชื้นในอากาศดังนี้

RH/Temp	10	15	20	25	30	35	40
10	0.97	0.95	0.95	0.95	0.85	0.82	0.81
20	1.15	1.12	1.13	1.09	0.98	0.96	0.88
30	1.48	1.45	1.44	1.42	1.36	1.37	1.31
40	1.76	1.73	1.74	**1.75**	**1.73**	**1.73**	**1.7**
50	2.06	2.06	2.08	**2.09**	**2.08**	**2.07**	**2.06**
60	2.36	2.37	2.37	**2.37**	**2.37**	**2.37**	**2.35**
70	2.64	2.63	2.62	**2.63**	**2.64**	**2.64**	**2.61**
80	2.92	2.91	2.92	**2.91**	**2.9**	**2.91**	**2.9**
90	3.09	3.08	3.07	3.09	3.07	3.04	3.03

ตารางที่ 4.1 แรงดันของความชื้นที่อุณหภูมิต่าง ๆ

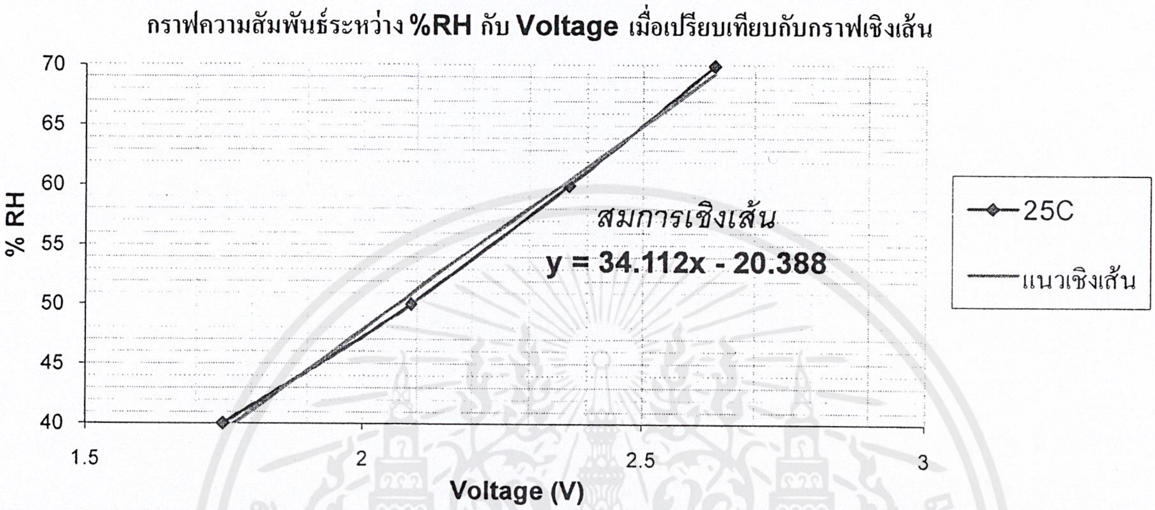
ในการนำข้อมูลมาใช้งานนั้น ทางผู้จัดทำได้นำข้อมูลดังตารางที่ 4.1 มาพล็อตกราฟ ซึ่งจะได้ ช่วงแรงดัน Output ของระดับความชื้นในอากาศ ที่มีค่าแตกต่างกันตามช่วงของอุณหภูมิ ดังนี้



รูปที่ 4.13 กราฟแสดงการเปรียบเทียบแรงดันกับความชื้นที่อุณหภูมิช่วงต่าง ๆ

จากรูปที่ 4.13 กราฟดังกล่าวจะมีความเป็นเชิงเส้นที่ความชื้น ตั้งแต่ 40%RH ถึง 85%RH ซึ่งเป็นความชื้นที่ใช้งานทั่วไป หากความความชื้นต่ำกว่านี้ (น้อยกว่า 40%RH) ในการอ่านค่าความชื้นต้องพิจารณาอุณหภูมิด้วย ซึ่งต้องพิจารณาแต่ละกราฟของอุณหภูมินั้นๆ

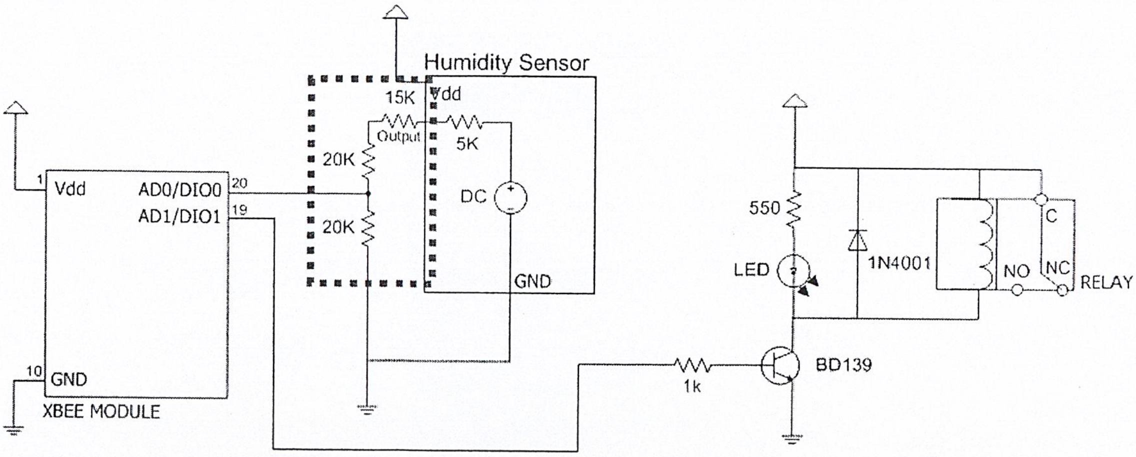
ต่อมาเมื่อพิจารณาเฉพาะช่วงที่นำไปใช้งานจริง จึงตัดกราฟบางส่วนออกมาได้ดังรูปที่ 4.14 พบว่า ทุกเส้นของอุณหภูมิ มีความชื้นกับแรงดันใกล้เคียงกัน จึงเลือกใช้ ที่ 25°C พร้อมกับสร้างสมการออกมา



รูปที่ 4.14 กราฟความสัมพันธ์ระหว่างความชื้นและแรงดันที่ 25°C

4.2.2.2 วงจรลดทอนแรงดัน

เนื่องจากบริษัทผู้ผลิตได้กำหนดไว้ว่าขา Input Analog ของโมดูล X-Bee นั้น สามารถทนแรงดันได้ไม่เกิน 1.2V แต่ค่าแรงดันเอาท์พุทของโมดูลวัดความชื้นมีค่าสูงสุดประมาณ 3.0V จึงทำให้ต้องมีการลดแรงดันเอาท์พุทและเพื่อให้ง่ายต่อการใช้งานจึงใช้วงจรแบ่งแรงดัน Voltage Divider เพื่อให้แรงดันที่ออกมาลดลง 3 เท่า โดยเลือกค่าความต้านทานในวงจรแบ่งแรงดันให้ที่ให้ค่าความต้านทานด้านที่ต่อกับ โมดูลวัดความชื้น มีค่ามากกว่าความต้านทานภายในด้านเอาท์พุทของโมดูลวัดความชื้น (5 KΩ) มากๆ เพื่อลดผล loading ในขณะเดียวกันความต้านทานของวงจรแบ่งแรงดันเมื่อพิจารณาจากทางด้าน Input Analog ของ X-Bee ต้องมีค่าน้อยกว่าความต้านทานด้าน Input ของวงจร A/D (1 MΩ) ดังนั้นจึงเลือกใช้ ค่าความต้านทาน ดังรูปคือ 20 KΩ, 20 KΩ และ 15 KΩ ดังรูปที่ 4.15



รูปที่ 4.15 การออกแบบวงจรสำหรับโหนดวัดปริมาณความชื้นในเครือข่าย Zigbee

4.3 บทสรุป

ในบทที่ผ่านมาอธิบายถึงการออกแบบและการทำงานของวงจรภายในระบบโครงงานต้นแบบว่า มีการทำงานอย่างไร, การออกแบบและการทำงานของหน่วยประมวลผลระบบ, การออกแบบและการทำงานของหน่วยประมวลผลเครือข่าย, การออกแบบและการทำงานของโหนดวัดอุณหภูมิและโหนดวัดความชื้น รวมทั้งศึกษาการทำงานของระบบโครงงานต้นแบบที่สามารถนำไปประยุกต์ใช้กับงานได้หลากหลายประเภท

บทที่ 5

การทดลองและสรุปผลการทดลองโครงงานต้นแบบ

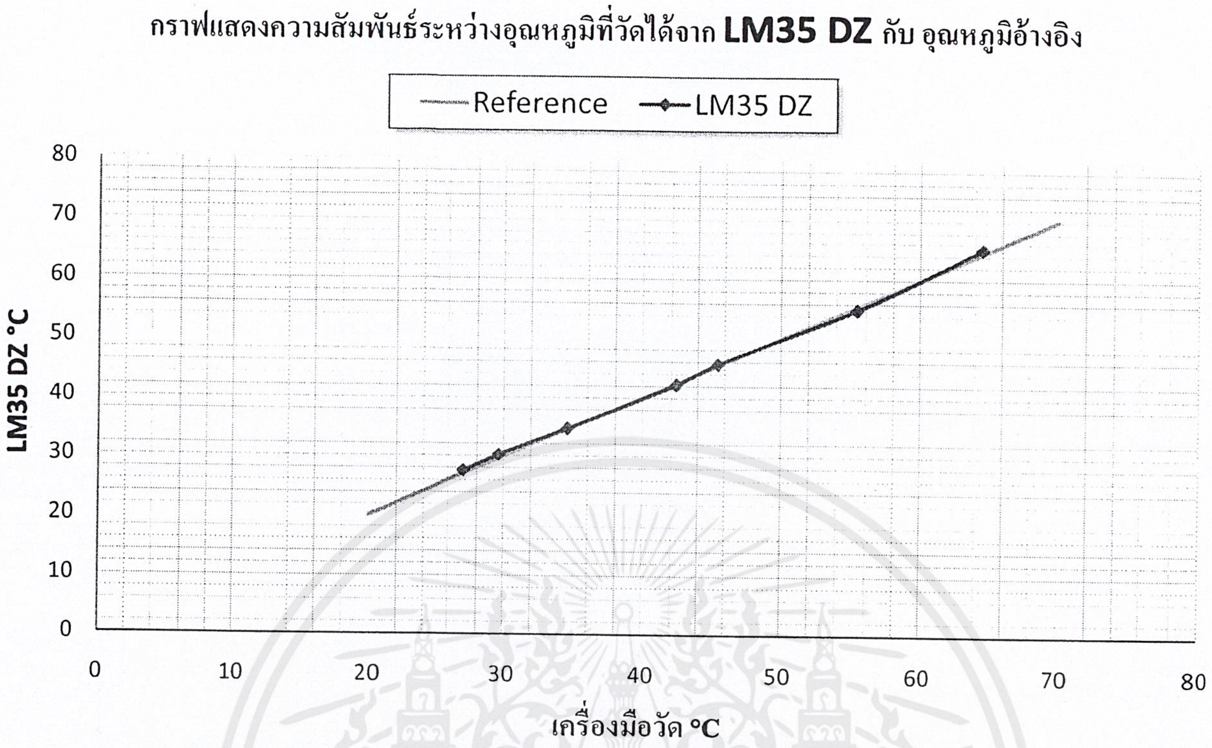
5.1 การสอบเทียบเซ็นเซอร์วัดอุณหภูมิ LM35 DZ

5.1.1 วิธีการสอบเทียบ

ในการทดลองนี้จึงนำวงจร LM35 ต่อเข้ากับโมดูล X-Bee ที่ทำหน้าที่เป็น Router ส่งค่าของอุณหภูมิทุกๆ 1 นาที ไปยังโมดูล X-Bee ที่ทำหน้าที่เป็น Coordinator ในเครือข่ายต่ออยู่กับ Gateway และส่งข้อมูลไปยังโปรแกรมเฝ้าระวังและควบคุมที่พัฒนาขึ้น เพื่อคำนวณค่าของอุณหภูมิ และแสดงผลทางเซิร์ฟเวอร์ จากนั้นนำมาเปรียบเทียบค่าอุณหภูมิที่อ่านได้จากเครื่องมือวัดอุณหภูมิอินฟราเรด

5.1.2 ผลการสอบเทียบ

LM35 DZ			อุณหภูมิที่ปรากฏบนเครื่องมือวัดอุณหภูมิ (°C)	ความคลาดเคลื่อน (%)
ค่าที่ปรากฏ (Hex)	Voltage (mV)	ค่าที่แปลงได้ (°C)		
0xE7	270.70	27.07	27.6	1.9%
0xFD	296.48	29.65	30.2	1.8%
0x127	345.70	34.57	34.7	0.4%
0x16A	424.21	42.42	42.2	0.5%
0x184	454.68	45.47	45.7	0.5%
0x1D9	554.29	55.43	54.9	0.9%
0x227	645.70	64.57	65.2	1.0%



รูปที่ 5.1 กราฟแสดงความแตกต่างระหว่าง อุณหภูมิที่วัดได้จาก LM35 DZ กับ อุณหภูมิจริง

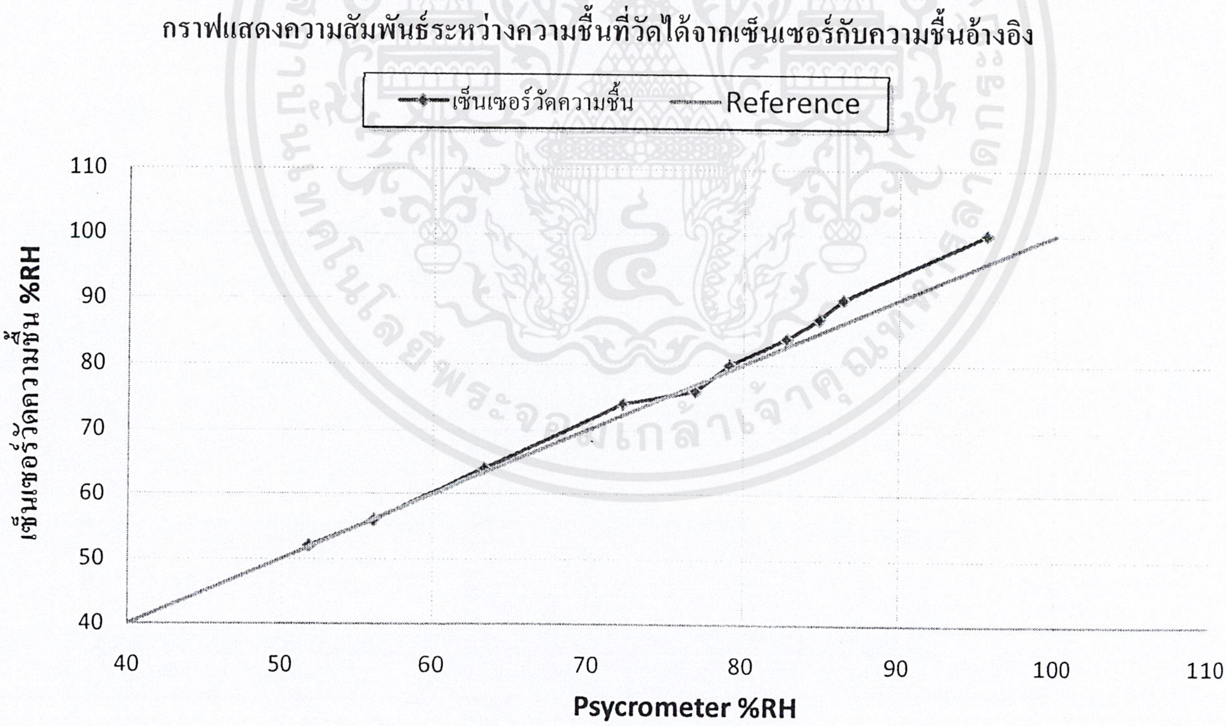
5.2 การสอบเทียบโมดูลวัดความชื้น

5.2.1 วิธีทำการสอบเทียบ

นำวงจรวัดความชื้น ต่อเข้ากับโมดูล X-bee ที่ทำหน้าที่เป็น Router ส่งข้อมูลความชื้นทุกๆ 1 นาที ไปยังโมดูล X-bee ที่เป็น Coordinator ในเครือข่ายต่ออยู่กับ Gateway และส่งข้อมูลไปยังโปรแกรมเฝ้าระวังและควบคุมที่พัฒนาขึ้น เพื่อคำนวณค่าของความชื้นสัมพัทธ์และแสดงผลทางเซิร์ฟเวอร์ จากนั้นนำมาเปรียบเทียบค่าความชื้นสัมพัทธ์ที่คำนวณได้จากอุปกรณ์วัดความชื้นแบบกระเปาะเปียกกระเปาะแห้ง (Psychrometer)

5.2.2 ผลการสอบเทียบ

โมดูลวัดความชื้น (%RH)	คำนวณความชื้นจาก Psychrometer (%RH)	ความคลาดเคลื่อน (%)
51.79	52	0.4%
56.14	56	0.25%
63.40	64	0.94%
72.25	74	2.36%
76.93	76	1.22%
79.13	80	1.08%
82.77	84	1.46%
84.88	87	2.44%
86.4	90	4.00%
95.67	100	4.33%



รูปที่ 5.2 กราฟแสดงความแตกต่างระหว่างความชื้นที่วัดได้จากเซ็นเซอร์ กับ ความชื้นอ้างอิง

5.3 ทดสอบประสิทธิภาพของวงจรลดทอนแรงดัน และ A/D Converter ของโมดูล X-Bee

5.3.1 วิธีการทดลอง

นำแหล่งจ่ายแรงดันไฟฟ้าที่สามารถบอกค่าแรงดันไฟฟ้าที่จ่ายออกมาได้ ต่อเข้ากับวงจรแบ่งแรงดัน และต่อเข้ากับโมดูล X-Bee ที่ทำหน้าที่เป็น Router ส่งข้อมูลทุกๆ 1 นาที ไปยังโมดูล X-Bee ที่เป็น Coordinator ในเครือข่ายต่ออยู่กับ Gateway และส่งข้อมูลไปยังโปรแกรมเฝ้าระวังและควบคุมที่พัฒนาขึ้น และนำ โวลต์มิเตอร์วัดแรงดันที่ออกมาจากวงจรแบ่งแรงดัน จากนั้นนำมาเปรียบเทียบกัน

5.3.2 ผลการทดลอง

แรงดันที่แหล่งจ่ายไฟตรง		แรงดันไฟฟ้าที่วงจรแบ่งแรงดัน		แรงดันที่ปรากฏบนโปรแกรม	
ค่าแรงดัน (V)	ค่าแรงดันหาร3 (mV)	วัด (mV)	ความคลาดเคลื่อน (%)	แสดงผล (mV)	ความคลาดเคลื่อน (%)
0.495	165	164.5	0.30%	165	0.30%
0.847	282.3	282.5	0.07%	282.3	0.07%
1.533	511	509	0.39%	511	0.39%
2.084	694	692	0.28%	694.7	0.39%
2.617	872	870	0.22%	872.3	0.26%
3.023	1007	1005	0.19%	1008	0.29%

5.4 ทดสอบประสิทธิภาพการทำงานของระบบเฝ้าระวัง

5.4.1 วิธีการทดสอบ

ทดสอบระบบโดยทำการเชื่อมต่อระบบเฝ้าระวังและควบคุมดังรูปที่ 4.1 ซึ่งในการทดลองนี้เป็นการเฝ้าระวังเพียงอย่างเดียว โดยการทำงานของระบบนั้น โปรแกรมเฝ้าระวัง จะต้องตั้งค่าคาบเวลา ที่ต้องการเก็บข้อมูลจากทุกๆโหนด ซึ่งในที่นี้ได้เก็บข้อมูลทุกๆ 5 นาที ตลอด 24 ชั่วโมง เพื่อสังเกตความเปลี่ยนแปลงของอุณหภูมิและความชื้นสัมพัทธ์ในอากาศตลอดระยะเวลา 1 วัน จากนั้นนำข้อมูลที่ได้มาพล๊อตกราฟเพื่อสังเกตความเปลี่ยนแปลงดังกล่าว โดยลักษณะการติดตั้งของโหนดต่างๆเป็นดังนี้

1. โหนด Router 1 เชื่อมต่อเซ็นเซอร์วัดอุณหภูมิ(LM 35DZ) ติดตั้งบริเวณในร่ม ที่มีลมพัดตลอดเวลา
2. โหนด Router 2 เชื่อมต่อโมดูลวัดความชื้น(SK 2L 97V-0) ติดตั้งบริเวณกลางแจ้งที่มีแสงแดดส่องถึง
3. โหนด Router 3 เชื่อมต่อเซ็นเซอร์วัดอุณหภูมิ(LM 35DZ) ติดตั้งบริเวณกลางแจ้งที่มีแสงแดดส่องถึง

5.4.2 ผลการทดสอบ

วันที่	เวลา	โหนด Router 1 วัดอุณหภูมิ (°C)	โหนด Router 2 วัดความชื้นสัมพัทธ์ (%RH)	โหนด Router 3 วัดอุณหภูมิ (°C)
17/2/2554	0:13:34	28.26	84.67	26.72
17/2/2554	0:18:36	28.26	84.91	26.72
17/2/2554	0:23:42	27.79	86.11	26.72
17/2/2554	0:28:47	27.55	86.59	26.72
17/2/2554	0:33:52	27.55	86.82	26.72
17/2/2554	0:38:57	27.55	87.30	26.72
17/2/2554	0:44:04	27.32	87.54	26.72
17/2/2554	0:49:08	27.32	87.54	26.72
17/2/2554	0:54:13	27.32	87.78	26.72
17/2/2554	0:59:18	27.32	87.54	26.72
17/2/2554	1:04:23	27.32	87.30	26.72
17/2/2554	1:09:29	27.32	87.30	26.72
17/2/2554	1:14:34	27.32	87.30	26.72
17/2/2554	1:19:39	27.32	87.30	26.72
17/2/2554	1:24:44	27.32	87.30	26.72
17/2/2554	1:29:49	27.32	87.30	26.72
17/2/2554	1:34:55	27.32	87.06	26.72
17/2/2554	1:40:00	27.32	87.06	26.72
17/2/2554	1:45:06	27.32	87.30	26.72
17/2/2554	1:50:10	27.32	87.54	26.72
17/2/2554	1:55:15	27.32	87.30	26.72
17/2/2554	2:00:21	27.32	87.30	26.48
17/2/2554	2:05:26	27.32	86.82	26.72
17/2/2554	2:10:31	27.32	86.82	26.72
17/2/2554	2:15:36	27.32	86.82	26.72
17/2/2554	2:20:42	27.32	86.82	26.72
17/2/2554	2:25:47	27.32	87.06	26.72

17/2/2554	2:30:52	27.32	87.06	26.72
17/2/2554	2:35:57	27.32	86.82	26.72
17/2/2554	2:41:02	27.08	86.82	26.72
17/2/2554	2:46:08	27.08	86.82	26.72
17/2/2554	2:51:13	27.08	87.06	26.48
17/2/2554	2:56:19	27.08	87.06	26.48
17/2/2554	3:01:23	27.08	87.06	26.48
17/2/2554	3:06:28	27.08	87.06	26.48
17/2/2554	3:11:34	27.08	86.82	26.48
17/2/2554	3:16:40	27.08	86.59	26.48
17/2/2554	3:21:44	27.08	86.11	26.72
17/2/2554	3:26:50	27.08	86.11	26.48
17/2/2554	3:31:55	27.32	86.35	26.48
17/2/2554	3:37:00	27.32	86.35	26.48
17/2/2554	3:42:05	27.32	85.87	26.48
17/2/2554	3:47:10	27.32	85.39	26.48
17/2/2554	3:52:15	27.32	85.63	26.25
17/2/2554	3:57:21	27.32	85.39	26.25
17/2/2554	4:02:26	27.55	84.91	26.02
17/2/2554	4:07:31	27.55	84.67	26.02
17/2/2554	4:12:36	27.55	84.91	26.02
17/2/2554	4:17:43	27.55	85.15	26.25
17/2/2554	4:22:47	27.55	84.91	26.02
17/2/2554	4:27:52	27.55	85.15	26.02
17/2/2554	4:32:57	27.32	85.63	26.02
17/2/2554	4:38:02	27.32	86.11	26.02
17/2/2554	4:43:08	27.32	86.35	26.02
17/2/2554	4:48:13	27.32	86.82	26.02
17/2/2554	4:53:18	27.32	86.59	25.78
17/2/2554	4:58:23	27.08	87.30	25.78
17/2/2554	5:03:28	27.08	87.30	26.02

17/2/2554	5:08:34	27.08	87.06	26.25
17/2/2554	5:13:39	27.08	87.30	26.02
17/2/2554	5:18:44	27.08	87.06	26.02
17/2/2554	5:23:50	27.08	86.35	26.25
17/2/2554	5:28:54	27.55	84.91	26.02
17/2/2554	5:34:00	27.55	84.91	26.02
17/2/2554	5:39:06	27.32	86.11	25.78
17/2/2554	5:44:10	27.32	86.11	25.78
17/2/2554	5:49:15	27.32	86.35	25.78
17/2/2554	5:54:21	27.55	85.63	25.78
17/2/2554	5:59:26	27.32	86.35	25.78
17/2/2554	6:04:33	27.08	87.06	25.78
17/2/2554	6:09:36	27.08	88.02	25.78
17/2/2554	6:14:42	26.85	88.26	25.78
17/2/2554	6:19:47	26.85	88.74	25.55
17/2/2554	6:24:53	26.85	88.26	25.78
17/2/2554	6:29:57	27.08	87.54	25.55
17/2/2554	6:35:02	27.08	86.82	25.55
17/2/2554	6:40:08	26.85	88.26	25.55
17/2/2554	6:45:13	27.08	87.06	25.55
17/2/2554	6:50:18	27.32	86.59	25.55
17/2/2554	6:55:23	27.32	85.63	25.55
17/2/2554	7:00:28	27.32	86.11	25.78
17/2/2554	7:05:35	27.32	86.11	25.78
17/2/2554	7:10:39	27.32	85.15	26.02
17/2/2554	7:15:44	27.55	84.43	26.25
17/2/2554	7:20:49	27.32	85.15	26.48
17/2/2554	7:25:54	27.32	86.35	26.48
17/2/2554	7:31:00	27.55	84.91	26.95
17/2/2554	7:36:05	27.32	85.39	27.19
17/2/2554	7:41:10	27.55	84.19	27.89

17/2/2554	7:46:15	27.55	84.91	28.36
17/2/2554	7:51:20	27.79	83.71	28.59
17/2/2554	7:56:26	27.79	81.67	29.06
17/2/2554	8:01:31	28.02	81.91	29.30
17/2/2554	8:06:36	28.02	80.71	29.53
17/2/2554	8:11:43	28.02	83.23	29.77
17/2/2554	8:16:47	28.26	82.27	30.23
17/2/2554	8:21:52	28.02	83.71	30.47
17/2/2554	8:26:57	28.02	83.47	30.00
17/2/2554	8:32:02	28.26	83.95	31.88
17/2/2554	8:37:07	28.26	83.23	29.30
17/2/2554	8:42:45	29.49	82.51	30.23
17/2/2554	8:47:50	29.49	81.19	30.23
17/2/2554	8:52:56	29.49	81.43	30.70
17/2/2554	8:58:01	29.73	80.95	30.70
17/2/2554	9:03:06	29.73	79.99	29.77
17/2/2554	9:08:11	29.96	79.99	31.17
17/2/2554	9:13:17	30.19	78.55	30.94
17/2/2554	9:18:23	30.19	79.27	30.94
17/2/2554	9:23:28	30.43	78.07	32.11
17/2/2554	9:28:32	30.66	75.19	31.17
17/2/2554	9:33:39	30.66	75.67	30.94
17/2/2554	9:38:43	30.66	76.15	31.64
17/2/2554	9:43:48	30.90	74.23	31.88
17/2/2554	9:48:53	30.90	70.16	31.41
17/2/2554	9:53:58	31.13	67.52	31.17
17/2/2554	9:59:04	31.13	65.84	31.41
17/2/2554	10:04:09	31.13	65.36	31.64
17/2/2554	10:09:14	31.13	65.12	32.81
17/2/2554	10:14:19	31.37	66.08	32.81
17/2/2554	10:19:24	31.60	65.12	32.58

17/2/2554	10:24:30	31.60	63.44	31.64
17/2/2554	10:29:35	31.60	64.16	31.64
17/2/2554	10:34:41	31.60	62.48	33.05
17/2/2554	10:39:45	31.60	63.68	32.11
17/2/2554	10:44:52	31.37	62.24	31.41
17/2/2554	10:49:56	31.37	62.72	31.88
17/2/2554	10:55:01	31.83	62.48	32.81
17/2/2554	11:00:07	31.60	61.40	31.17
17/2/2554	11:05:11	31.60	60.92	32.81
17/2/2554	11:10:18	31.83	58.52	33.98
17/2/2554	11:15:23	32.30	59.00	33.52
17/2/2554	11:20:27	32.30	59.24	33.98
17/2/2554	11:25:32	32.30	59.24	34.22
17/2/2554	11:30:37	32.54	58.04	34.45
17/2/2554	11:35:43	32.77	58.28	34.92
17/2/2554	11:40:48	32.54	59.48	34.92
17/2/2554	11:45:53	32.77	59.48	34.69
17/2/2554	11:50:58	33.01	58.52	35.39
17/2/2554	11:56:04	32.77	58.52	35.63
17/2/2554	12:01:09	33.01	59.96	36.56
17/2/2554	12:06:14	32.77	59.24	37.75
17/2/2554	12:11:19	33.01	57.08	38.20
17/2/2554	12:16:24	32.77	59.48	38.30
17/2/2554	12:21:30	32.77	59.24	38.44
17/2/2554	12:26:35	33.01	58.04	38.20
17/2/2554	12:31:40	33.24	54.92	39.40
17/2/2554	12:36:45	33.48	56.60	41.48
17/2/2554	12:41:50	33.01	57.80	40.42
17/2/2554	12:46:56	33.01	59.24	41.02
17/2/2554	12:52:02	32.77	59.96	42.00
17/2/2554	12:57:06	33.01	59.96	43.13

17/2/2554	13:02:11	32.54	59.72	41.50
17/2/2554	13:07:18	32.30	59.00	40.10
17/2/2554	13:12:22	32.77	58.76	40.02
17/2/2554	13:17:27	32.77	57.80	40.08
17/2/2554	13:22:32	33.01	58.04	42.66
17/2/2554	13:27:37	33.01	57.80	41.30
17/2/2554	13:32:43	33.01	58.76	40.31
17/2/2554	13:37:48	32.77	58.28	42.20
17/2/2554	13:42:54	32.54	57.80	41.05
17/2/2554	13:47:58	33.01	57.80	41.95
17/2/2554	13:53:03	33.01	59.00	42.70
17/2/2554	13:58:09	33.01	58.28	43.47
17/2/2554	14:03:14	33.01	57.08	44.53
17/2/2554	14:08:20	33.24	57.08	42.66
17/2/2554	14:13:24	33.24	55.88	41.48
17/2/2554	14:18:30	33.48	56.12	39.75
17/2/2554	14:23:35	33.71	53.01	38.20
17/2/2554	14:28:40	33.71	51.57	36.48
17/2/2554	14:33:45	33.24	52.05	35.07
17/2/2554	14:38:50	33.48	52.05	34.45
17/2/2554	14:43:56	33.48	51.09	34.45
17/2/2554	14:49:01	33.71	50.37	34.92
17/2/2554	14:54:06	33.48	51.09	33.98
17/2/2554	14:59:11	33.48	52.29	34.22
17/2/2554	15:04:16	33.48	54.45	34.22
17/2/2554	15:09:22	33.48	51.57	33.98
17/2/2554	15:14:27	33.71	51.09	35.16
17/2/2554	15:19:32	33.94	51.57	34.69
17/2/2554	15:24:39	33.48	51.81	33.75
17/2/2554	15:29:43	33.48	51.09	34.22
17/2/2554	15:34:48	33.48	52.05	34.22

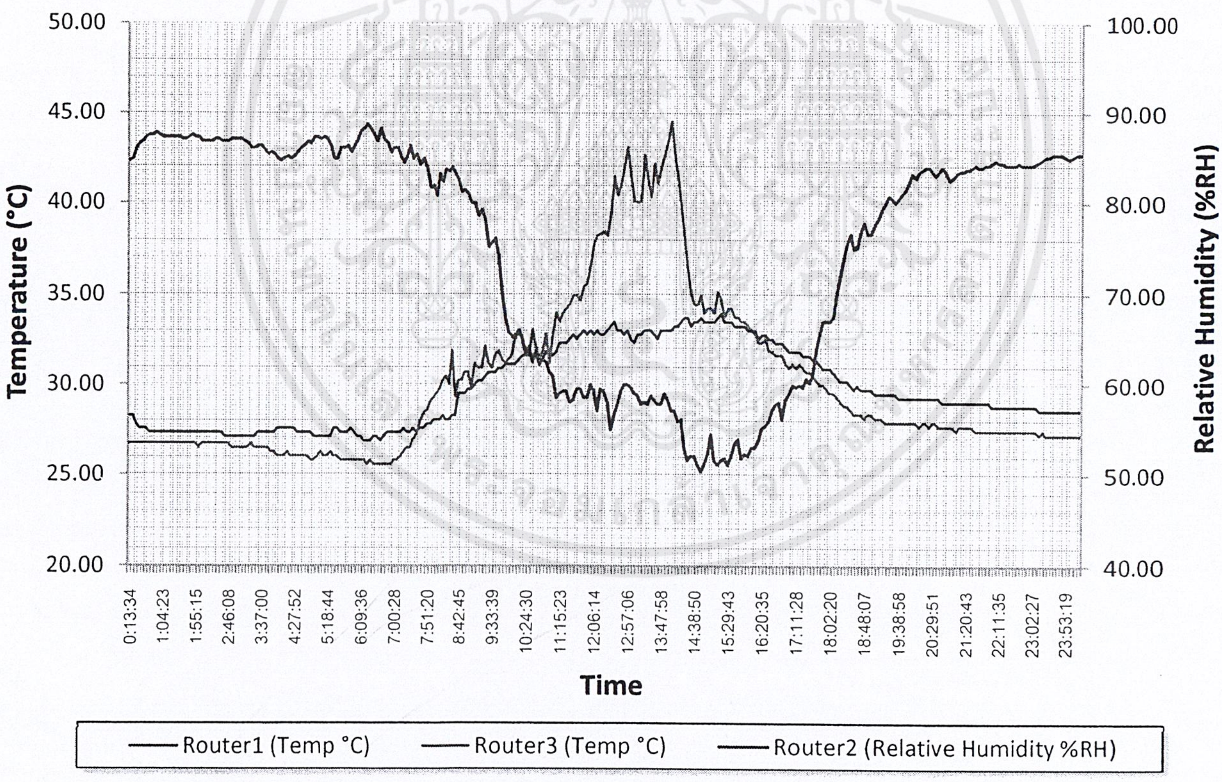
17/2/2554	15:39:53	33.24	53.49	33.75
17/2/2554	15:44:58	33.24	53.97	33.75
17/2/2554	15:50:03	33.24	51.81	33.52
17/2/2554	15:55:09	33.24	52.29	33.52
17/2/2554	16:00:14	33.01	52.05	33.28
17/2/2554	16:05:19	33.01	53.01	33.05
17/2/2554	16:10:24	33.01	53.01	32.81
17/2/2554	16:15:31	32.54	53.73	32.34
17/2/2554	16:20:35	32.77	55.16	32.34
17/2/2554	16:25:40	32.77	55.40	32.58
17/2/2554	16:30:45	32.54	55.88	31.88
17/2/2554	16:35:50	32.54	57.08	31.88
17/2/2554	16:40:56	32.30	57.80	31.64
17/2/2554	16:46:01	32.30	58.04	31.64
17/2/2554	16:51:07	32.30	56.12	31.64
17/2/2554	16:56:11	32.07	58.28	31.17
17/2/2554	17:01:16	31.83	59.00	30.94
17/2/2554	17:06:22	31.83	59.96	31.17
17/2/2554	17:11:28	31.83	59.72	30.94
17/2/2554	17:16:32	31.83	59.96	31.17
17/2/2554	17:21:37	31.60	59.72	30.94
17/2/2554	17:26:42	31.60	60.68	30.70
17/2/2554	17:31:48	31.60	60.20	30.70
17/2/2554	17:36:53	31.37	61.16	30.47
17/2/2554	17:41:58	31.37	63.92	30.23
17/2/2554	17:47:03	31.13	65.60	30.00
17/2/2554	17:52:09	30.90	67.04	30.00
17/2/2554	17:57:14	30.90	67.04	29.77
17/2/2554	18:02:20	30.90	67.04	29.53
17/2/2554	18:07:24	30.66	67.52	29.53
17/2/2554	18:12:29	30.43	70.16	29.30

17/2/2554	18:17:35	30.19	72.07	29.06
17/2/2554	18:22:40	30.19	73.99	28.83
17/2/2554	18:27:45	30.19	75.67	28.83
17/2/2554	18:32:50	29.96	76.63	28.59
17/2/2554	18:37:56	29.73	74.95	28.36
17/2/2554	18:43:01	29.96	75.19	28.36
17/2/2554	18:48:06	29.73	76.83	28.36
17/2/2554	18:48:07	29.73	77.83	28.13
17/2/2554	18:53:11	29.73	76.63	28.36
17/2/2554	18:58:16	29.73	76.63	28.36
17/2/2554	19:03:22	29.49	77.59	28.13
17/2/2554	19:08:27	29.49	78.31	28.13
17/2/2554	19:13:32	29.49	79.27	28.13
17/2/2554	19:18:37	29.49	79.99	27.89
17/2/2554	19:23:42	29.49	80.71	27.89
17/2/2554	19:28:48	29.49	80.47	27.89
17/2/2554	19:33:53	29.49	79.99	27.89
17/2/2554	19:38:58	29.26	80.47	27.89
17/2/2554	19:44:03	29.26	80.95	27.89
17/2/2554	19:49:09	29.26	81.43	27.89
17/2/2554	19:54:14	29.26	82.27	27.89
17/2/2554	19:59:19	29.26	83.23	27.89
17/2/2554	20:04:24	29.26	82.75	27.89
17/2/2554	20:09:29	29.26	83.47	27.66
17/2/2554	20:14:35	29.26	83.71	27.89
17/2/2554	20:19:40	29.26	83.95	27.89
17/2/2554	20:24:45	29.26	83.95	27.66
17/2/2554	20:29:51	29.26	83.47	27.89
17/2/2554	20:34:55	29.26	82.99	27.89
17/2/2554	20:40:01	29.26	83.47	27.66
17/2/2554	20:45:06	29.02	83.95	27.66

17/2/2554	20:50:11	29.02	83.47	27.66
17/2/2554	20:55:18	29.02	82.51	27.66
17/2/2554	21:00:22	29.02	82.75	27.66
17/2/2554	21:05:27	29.02	83.23	27.42
17/2/2554	21:10:32	29.02	83.47	27.66
17/2/2554	21:15:37	29.02	83.47	27.66
17/2/2554	21:20:43	29.02	83.71	27.66
17/2/2554	21:25:48	29.02	83.71	27.66
17/2/2554	21:30:53	29.02	83.95	27.66
17/2/2554	21:35:58	29.02	84.19	27.42
17/2/2554	21:41:03	29.02	83.95	27.42
17/2/2554	21:46:09	29.02	83.95	27.42
17/2/2554	21:51:14	29.02	84.19	27.42
17/2/2554	21:56:19	29.02	84.19	27.42
17/2/2554	22:01:25	28.79	84.43	27.42
17/2/2554	22:06:29	28.79	84.67	27.42
17/2/2554	22:11:35	28.79	84.43	27.42
17/2/2554	22:16:40	28.79	84.43	27.42
17/2/2554	22:21:45	28.79	84.19	27.42
17/2/2554	22:26:50	28.79	84.19	27.42
17/2/2554	22:31:55	28.79	84.19	27.42
17/2/2554	22:37:01	28.79	84.19	27.42
17/2/2554	22:42:06	28.79	84.43	27.42
17/2/2554	22:47:11	28.79	84.19	27.42
17/2/2554	22:52:16	28.79	84.19	27.42
17/2/2554	22:57:23	28.79	84.19	27.42
17/2/2554	23:02:27	28.79	84.19	27.42
17/2/2554	23:07:32	28.79	84.43	27.42
17/2/2554	23:12:37	28.55	84.67	27.19
17/2/2554	23:17:42	28.55	84.91	27.42
17/2/2554	23:22:48	28.55	85.15	27.19

17/2/2554	23:27:53	28.55	85.15	27.19
17/2/2554	23:32:58	28.55	85.39	27.19
17/2/2554	23:38:03	28.55	85.39	27.19
17/2/2554	23:43:08	28.55	85.39	27.19
17/2/2554	23:48:14	28.55	85.39	27.19
17/2/2554	23:53:19	28.55	85.15	27.19
17/2/2554	23:58:24	28.55	84.91	27.19
18/2/2554	0:03:29	28.55	85.15	27.19
18/2/2554	0:08:35	28.55	85.39	27.19
18/2/2554	0:13:40	28.55	85.39	27.19

กราฟแสดงความสัมพันธ์ระหว่างอุณหภูมิและความชื้นที่วัดได้จากโหนดต่างๆ ตลอด ระยะเวลา 24 ชั่วโมง



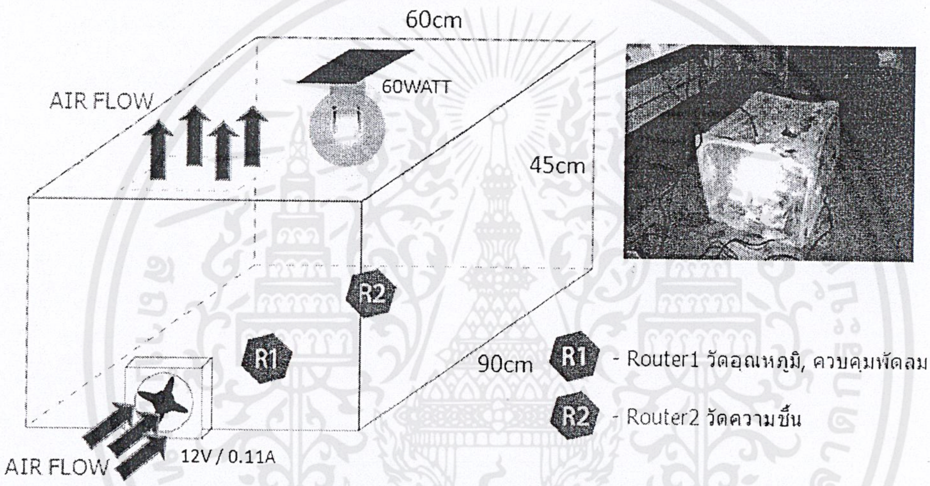
รูปที่ 5.3 กราฟแสดงความสัมพันธ์ระหว่างอุณหภูมิและความชื้นที่วัดได้จากโหนดต่างๆตลอด 24 ชั่วโมง

5.5 ทดสอบประสิทธิภาพการทำงานของระบบเฝ้าระวังและควบคุม

5.5.1 ทดสอบประสิทธิภาพการควบคุมอุณหภูมิ

5.5.1.1 วิธีการทดสอบ

ก่อนการทดสอบประสิทธิภาพการควบคุมอุณหภูมินั้น จึงได้สร้างโรงเรือนจำลอง สำหรับใช้ในการทดสอบ โรงเรือนจำลองที่สร้างขึ้น มีขนาด 60 x 90 x 45 เซนติเมตร ซึ่งในการทดสอบนี้เป็นการทดสอบควบคุมอุณหภูมิ จึงเลือกใช้หลอดไฟ ติดเหนือโรงเรือนจำลอง เพื่อให้ความร้อนแก่ระบบปิด ทำให้อุณหภูมิสูงขึ้น นอกจากนี้ เราสามารถลดอุณหภูมิในระบบปิดได้ โดยเจาะรูด้านข้างของระบบปิด เพื่อเป็นทางเข้าของลมเย็น เมื่อลมเย็นถูกเป่าเข้าสู่ระบบ ความร้อนที่อยู่ในระบบจะลอยตัวสูงขึ้น และถูกระบายออก ทางช่องด้านบนที่ถูกเจาะไว้ ดังรูป 5.1



รูปที่ 5.4 ระบบโรงเรือนปิดที่ได้ออกแบบไว้สำหรับการทดสอบควบคุมอุณหภูมิ

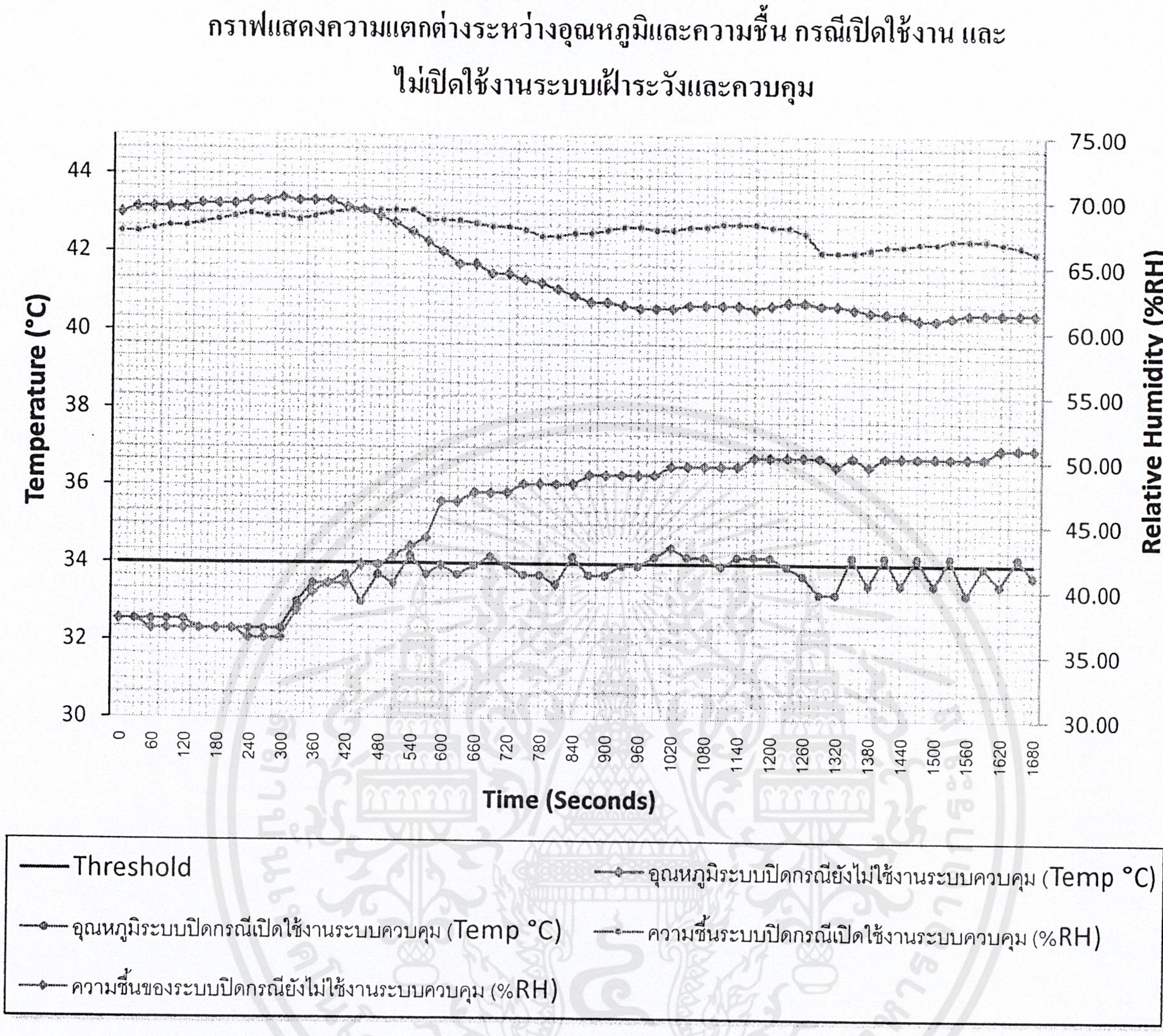
ทดสอบวัดค่าอุณหภูมิในโรงเรือนจำลอง โดยข้อมูลชุดแรกนั้น เป็นการมอนิเตอร์ข้อมูลอุณหภูมิและเก็บข้อมูล ทุกๆ 30 วินาที ตลอด 30 นาที หลังจากนั้นทำการเก็บข้อมูลชุดที่สอง โดยกำหนดค่าในโปรแกรมเฝ้าระวังและควบคุมให้อุณหภูมิในระบบไม่เกิน 34 องศา จากนั้นทำการมอนิเตอร์ข้อมูลและควบคุม โดยทำการเก็บข้อมูลทุกๆ 30 วินาทีตลอด 30 นาที เช่นเดียวกับข้อมูลชุดแรก สุดท้ายทำการพล๊อตกราฟเปรียบเทียบกันระหว่าง กรณีเปิดใช้งาน และ ไม่เปิดใช้งานระบบเฝ้าระวังและควบคุม

ทั้งนี้ ตลอดการทดลองนั้น นอกจากทำการเก็บข้อมูลอุณหภูมิแล้ว ยังทำการเก็บข้อมูลความชื้นด้วย เพื่อทำการศึกษาความสัมพันธ์ระหว่างอุณหภูมิและความชื้นในโรงเรือนจำลอง

5.5.1.2 ผลการทดสอบ

วินาที	กรณีไม่เปิดใช้งานระบบเฝ้าระวังและควบคุม		กรณีเปิดใช้งานระบบเฝ้าระวังและควบคุม	
	อุณหภูมิ (°C)	ความชื้นสัมพัทธ์ (%RH)	อุณหภูมิ (°C)	ความชื้นสัมพัทธ์ (%RH)
1	32.54	68.96	32.54	67.52
30	32.54	69.44	32.54	67.52
60	32.30	69.44	32.54	67.76
90	32.30	69.44	32.54	68.00
120	32.30	69.44	32.54	68.00
150	32.30	69.68	32.30	68.24
180	32.30	69.68	32.30	68.48
210	32.30	69.68	32.30	68.72
240	32.07	69.92	32.30	68.96
270	32.07	69.92	32.30	68.72
300	32.07	70.16	32.30	68.72
330	32.77	69.92	33.01	68.48
360	33.24	69.92	33.48	68.72
390	33.48	69.92	33.48	68.96
420	33.48	69.44	33.71	69.20
450	33.94	69.20	33.01	69.20
480	33.94	68.72	33.71	69.20
510	34.18	68.24	33.48	69.20
540	34.41	67.52	34.18	69.20
570	34.65	66.80	33.71	68.48
600	35.58	66.08	33.94	68.48
630	35.58	65.12	33.71	68.48
660	35.82	65.12	33.94	68.24
690	35.82	64.40	34.18	68.00
720	35.82	64.40	33.94	68.00
750	36.05	63.92	33.71	67.76

780	36.05	63.68	33.71	67.28
810	36.05	63.20	33.48	67.28
840	36.05	62.72	34.18	67.52
870	36.29	62.24	33.71	67.52
900	36.29	62.24	33.71	67.76
930	36.29	62.00	33.94	68.00
960	36.29	61.76	33.94	68.00
990	36.29	61.76	34.18	67.76
1020	36.52	61.76	34.41	67.76
1050	36.52	62.00	34.18	68.00
1080	36.52	62.00	34.18	68.00
1110	36.52	62.00	33.94	68.24
1140	36.52	62.00	34.18	68.24
1170	36.76	61.76	34.18	68.24
1200	36.76	62.00	34.18	68.00
1230	36.76	62.24	33.94	68.00
1260	36.76	62.24	33.71	67.52
1290	36.76	62.00	33.24	66.08
1320	36.52	62.00	33.24	66.08
1350	36.76	61.76	34.18	66.08
1380	36.52	61.52	33.48	66.32
1410	36.76	61.40	34.18	66.56
1440	36.76	61.40	33.48	66.56
1470	36.76	60.92	34.18	66.80
1500	36.76	60.92	33.48	66.80
1530	36.76	61.16	34.18	67.04
1560	36.76	61.40	33.24	67.04
1590	36.76	61.40	33.94	67.04
1620	36.99	61.40	33.48	66.80
1650	36.99	61.40	34.18	66.56
1680	36.99	61.40	33.71	66.08

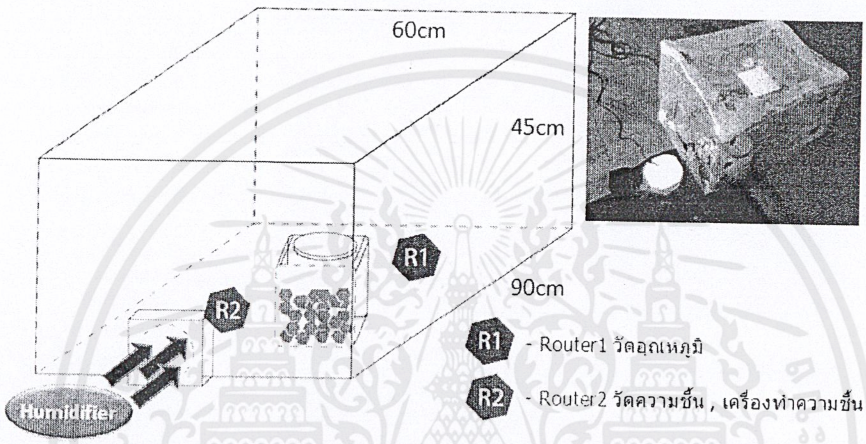


รูปที่ 5.5 กราฟแสดงความแตกต่างระหว่าง อุณหภูมิและความชื้นกรณีเปิดใช้งานและไม่เปิดใช้งานระบบเฝ้าระวังและควบคุม

5.5.2 ทดสอบประสิทธิภาพการควบคุมความชื้นสัมพัทธ์

5.5.2.1 วิธีการทดสอบ

ก่อนการทดสอบประสิทธิภาพการควบคุมความชื้นสัมพัทธ์นั้น จึงได้ใช้โรงเรือนจำลองเดียวกับการทดสอบที่ผ่านมา ซึ่งในการทดสอบนี้เป็นการทดสอบควบคุมความชื้นสัมพัทธ์ในระบบ จึงเลือกใช้ถังใส่น้ำแข็งวางไว้ในระบบปิด เพื่อให้เกิดการควบแน่นในระบบ ทำให้ความชื้นในระบบต่ำลง นอกจากนี้ เราสามารถเพิ่มความชื้นในระบบปิดได้ โดยเจาะรูด้านข้างของระบบปิด เพื่อเป็นทางเข้าของเครื่องสร้างละอองน้ำ ทำให้ระบบปิดมีความชื้นเพิ่มมากขึ้น ดังรูป 5.2



รูปที่ 5.6 ระบบโรงเรือนปิดที่ได้ออกแบบไว้สำหรับการทดสอบควบคุมความชื้นสัมพัทธ์

ทดสอบวัดค่าความชื้นในโรงเรือนจำลอง โดยช่วงแรกนั้น ทำการเก็บข้อมูลกำหนดค่าในโปรแกรมเฝ้าระวังและควบคุมให้ความชื้นในระบบไม่น้อยกว่า 70% และ 80% ตามลำดับ จากนั้นทำการมอนิเตอร์ข้อมูลและควบคุม โดยทำการเก็บข้อมูลทุกๆ 30 วินาทีตลอด 35 นาที โดยการทดลองเก็บข้อมูลนั้นแบ่งเป็น 3 ช่วงคือ ช่วงแรกทำการลดความชื้นในโรงเรือนให้เหลือประมาณ 60% โดยหลักการควบแน่นของน้ำ หลังจากนั้น ช่วงที่สองเป็นการเปิดระบบเฝ้าระวังและควบคุม โดยที่ควบคุมความชื้นในโรงเรือนจำลองให้มีความชื้นสัมพัทธ์ 70% และช่วงสุดท้าย เป็นการควบคุมความชื้นในโรงเรือนจำลองให้มีความชื้นสัมพัทธ์ 80%

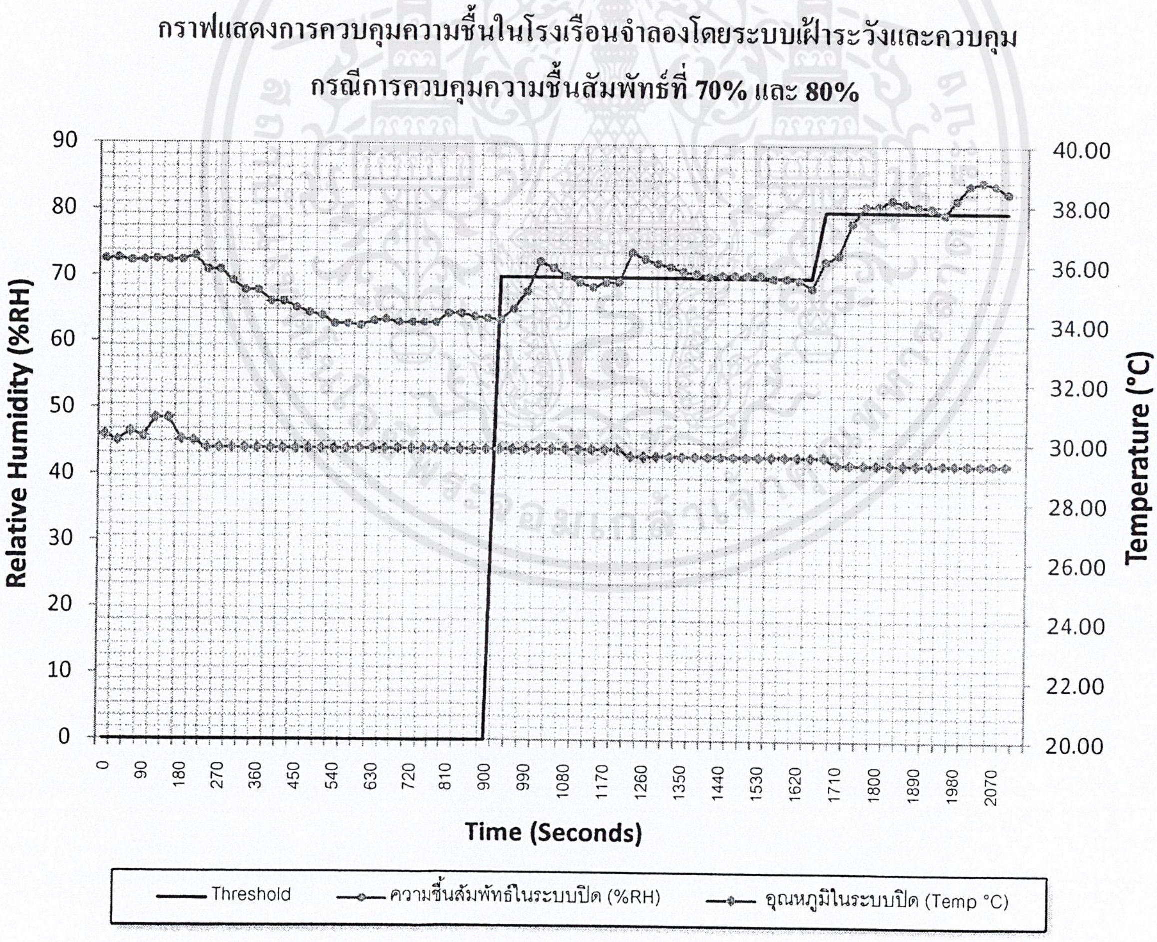
ทั้งนี้ ตลอดการทดลองนั้น นอกจากทำการเก็บข้อมูลความชื้นสัมพัทธ์แล้ว ยังทำการเก็บข้อมูลอุณหภูมิด้วย เพื่อทำการศึกษาความสัมพันธ์ระหว่างความชื้นและอุณหภูมิในโรงเรือนจำลอง

5.5.2.2 ผลการทดสอบ

วินาที	อุณหภูมิในโรงเรือนจำลอง (°C)	ความชื้นสัมพัทธ์ในโรงเรือนจำลอง (%RH)
1	30.20	72.48
30	30.00	72.59
60	30.30	72.23
90	30.15	72.31
120	30.76	72.47
150	30.76	72.32
180	30.05	72.31
210	30.00	72.96
240	29.77	70.87
270	29.77	70.87
300	29.77	69.20
330	29.77	67.76
360	29.77	67.76
390	29.77	66.08
420	29.77	66.08
450	29.77	65.12
480	29.77	64.40
510	29.77	63.92
540	29.77	62.72
570	29.77	62.72
600	29.77	62.48
630	29.77	63.20
660	29.77	63.44
690	29.77	62.96
720	29.77	62.96
750	29.77	62.96
780	29.77	62.96
810	29.77	64.40

840	29.77	64.40
870	29.77	63.92
900	29.77	63.68
930	29.77	63.44
960	29.77	65.12
990	29.77	67.76
1020	29.77	72.31
1050	29.77	71.35
1080	29.77	70.16
1110	29.77	69.20
1140	29.77	68.48
1170	29.77	69.20
1200	29.77	69.20
1230	29.53	73.75
1260	29.53	72.79
1290	29.53	72.07
1320	29.53	71.59
1350	29.53	70.87
1380	29.53	70.64
1410	29.53	70.16
1440	29.53	70.40
1470	29.53	70.40
1500	29.53	70.40
1530	29.53	70.40
1560	29.53	69.92
1590	29.53	69.92
1620	29.53	69.68
1650	29.53	68.48
1680	29.53	72.55
1710	29.30	73.51
1740	29.30	78.31

1770	29.30	80.95
1800	29.30	80.95
1830	29.30	81.91
1860	29.30	81.43
1890	29.30	80.95
1920	29.30	80.71
1950	29.30	79.75
1980	29.30	81.91
2010	29.30	84.19
2040	29.30	84.67
2070	29.30	84.19
2100	29.30	82.99



รูปที่ 5.7 กราฟแสดงการควบคุมความชื้นใน โรงเรือน กรณีควบคุมความชื้นที่ 70% และ 80%

5.6 สรุปผลการทดสอบ

จากการทดลองการใช้งาน LM35 DZ ในการวัดอุณหภูมิ ปรากฏว่า LM35 DZ ให้ผลการตอบสนองต่ออุณหภูมิได้อย่างถูกต้องแม่นยำ ดังที่ได้แสดงกราฟจากรูปที่ 5.1 เมื่อเทียบกับอุณหภูมิอ้างอิงแล้วมีความใกล้เคียงกันมาก ดังนั้นการวัดอุณหภูมิของ LM35 DZ ที่แสดงผลทางออกมาทางเซิร์ฟเวอร์มีความถูกต้องและเชื่อถือได้

จากการทดลองการใช้งานเซ็นเซอร์วัดความชื้น ปรากฏว่า เซ็นเซอร์วัดความชื้นมีผลต่อการตอบสนองได้ถูกต้อง ที่ช่วงความชื้น 40-85 %RH นอกเหนือจากช่วงความชื้นนี้ ค่าความชื้นที่วัดได้จะมีค่าคลาดเคลื่อนมากขึ้นเรื่อยๆ ดังที่ได้แสดงจากกราฟจากรูปที่ 5.2 เมื่อเทียบกับความชื้นอ้างอิง ดังนั้นการใช้งานเซ็นเซอร์วัดความชื้นจำเป็นต้องมีใช้งานที่ช่วงความชื้นที่ 40-85 %RH จึงจะได้ค่าความชื้นที่ถูกต้องและแม่นยำ

ในทดสอบการใช้งานวงจรลดทอนแรงดันและ A/D Converter ของโมดูล X-Bee ปรากฏว่า วงจรลดทอนแรงดันสามารถแบ่งแรงดันจากแหล่งจ่ายไฟให้เหลือ 1/3 ของแรงดันที่จ่ายออกมา เมื่อดูจากตารางผลการทดลองที่ 5.3.2 ค่าแรงดันที่แสดงบนโปรแกรมเฝ้าระวังและควบคุม และที่วัดได้จากวงจรลดทอนแรงดันนั้น มีค่าความคลาดเคลื่อนที่น้อยมาก ซึ่งสามารถยอมรับได้ ดังนั้นการทำงานของวงจรลดทอนแรงดันและ A/D Converter ของโมดูล X-bee มีความถูกต้องและเชื่อถือได้

ในการทดสอบประสิทธิภาพระบบเฝ้าระวังนั้น เป็นการทดสอบประสิทธิภาพของซอฟท์แวร์เฝ้าระวังและควบคุม และ ประสิทธิภาพของเซ็นเซอร์วัดอุณหภูมิ (LM35 DZ) และโมดูลวัดความชื้น ตลอดระยะเวลา 24 ชั่วโมง โดยเป็นการเก็บข้อมูลทุกๆ 5 นาที จำลองเหตุการณ์สภาพอากาศจริงตลอดระยะเวลา 24 ชั่วโมง ซึ่งจากการทดสอบประสิทธิภาพของระบบเฝ้าระวังอย่างเต็มรูปแบบนั้น ปรากฏว่า โปรแกรมเฝ้าระวังและควบคุม ตลอดจนฐานข้อมูลที่ใช้ในการเก็บข้อมูลจากโหนดปลายทางนั้น มีความสามารถในการจัดการข้อมูลที่ได้รับจากโหนดปลายทางได้อย่างมีประสิทธิภาพ ในส่วนของการแสดงผลของข้อมูล ผู้ดูแลระบบสามารถใช้งานได้สะดวกและเข้าใจง่าย นอกจากนั้น การทำงานของเซ็นเซอร์ตามโหนดต่างๆนั้น สามารถตอบสนองการทำงานตลอด 24 ชั่วโมงได้อย่างมีประสิทธิภาพเช่นกัน ซึ่งข้อมูลดังกล่าวได้แสดงออกมาดังกราฟผลการทดลองหัวข้อ 5.4.2 เมื่อดูจากความสัมพันธ์ความแตกต่างระหว่าง อุณหภูมิบริเวณกลางแจ้ง และ อุณหภูมิห้องที่มีอากาศถ่ายเทนั้น ปรากฏว่า ช่วงเวลา 11.00น. ถึง 14.00น. นั้น อุณหภูมิบริเวณกลางแจ้ง จะสูงขึ้นแตกต่างจากอุณหภูมิห้องอย่างเห็นได้ชัด ซึ่งมีความแตกต่างมากสุดถึง 10 องศาเซลเซียส นอกจากนี้ เมื่อสังเกตช่วงเวลาอื่นๆ โดยส่วนใหญ่แล้ว บริเวณกลางแจ้งในเวลาเย็นถึงช่วงเช้ามืด จะมีอุณหภูมิที่ต่ำกว่าอุณหภูมิห้องเพียงเล็กน้อย ซึ่งอาจเกิดจากกระแสลมเย็นที่พัดผ่านบริเวณกลางแจ้งในช่วงเวลาเย็นถึงช่วงเช้ามืด ทำให้มีอุณหภูมิที่ต่ำกว่า

ข้อมูลดังกราฟหัวข้อผลการทดลองที่ 5.4.2 ยังมีข้อมูลของประสิทธิภาพการทดสอบเซ็นเซอร์วัดความชื้นสัมพัทธ์ตลอดระยะเวลา 24 ชั่วโมง ซึ่งข้อมูลในช่วงเช้ามืด ตั้งแต่เวลา 5.00น. – 6.00น. เป็นช่วงที่มีความชื้นสัมพัทธ์ในอากาศมากที่สุด เหตุผลที่เป็นเช่นนั้น เนื่องจากอุณหภูมิที่ลดลงค่าความชื้นสัมพัทธ์จะสูงขึ้น ทั้งนี้เนื่องจาก ค่าความชื้นสัมพัทธ์เท่ากับปริมาณไอน้ำที่มีอยู่จริงในอากาศ ต่อ ปริมาณไอน้ำอิ่มตัวที่อุณหภูมิและปริมาตรเดียวกัน แต่เนื่องจาก หากอุณหภูมิสูงอากาศจะได้รับไอน้ำมากกว่าอากาศที่มีอุณหภูมิต่ำ สาเหตุนี้จึงเป็นเหตุผลที่ทำให้ ช่วงที่มีอุณหภูมิต่ำ จะมีความชื้นสัมพัทธ์ในอากาศที่สูง และช่วงที่มีอุณหภูมิสูง จะมีความชื้น

สัมพัทธ์ในอากาศที่ต่ำ สังเกตจากกราฟในช่วงเวลาตอนกลางวันที่มีแดดจัด (11.00น. ถึง 14.00น.) ความชื้นสัมพัทธ์ในอากาศได้ลดลงอย่างรวดเร็ว จนกระทั่งเพิ่มขึ้นอีกครั้งในช่วงเวลาเย็น เนื่องจากสภาวะแวดล้อมมีอุณหภูมิต่ำลง

จากหัวข้อการทดสอบ 5.5.1 เป็นการทดสอบประสิทธิภาพการทำงานของระบบเฝ้าระวังและควบคุม โดยการทดลองนี้เป็นการทดสอบการควบคุมอุณหภูมิในโรงเรือนจำลองแบบปิด ซึ่งผลการทดสอบเป็นสามารถทำงานได้อย่างมีประสิทธิภาพในการควบคุมอุณหภูมิในระดับหนึ่งเท่านั้น เนื่องจากระบบปรับอากาศที่ได้ออกแบบไว้มีขนาดเล็กเกินไป หากต้องการควบคุมอุณหภูมิที่มีประสิทธิภาพมากกว่านี้ ต้องทำการเปลี่ยนระบบปรับอากาศให้มีขนาดใหญ่ขึ้น ทำให้สามารถระบายความร้อนได้มากกว่าเดิม

จากกราฟแสดงผลการทดสอบในหัวข้อ 5.5.1.2 นั้น ภาพรวมของอุณหภูมิระหว่างการเปิดใช้งานระบบเฝ้าระวังและควบคุมกับไม่เปิดใช้งานนั้น สังเกตได้ว่า อุณหภูมิสามารถถูกควบคุมให้อยู่ในช่วงอุณหภูมิที่ตั้งค่าไว้จากโปรแกรมเฝ้าระวังและควบคุม (ในการทดลองนี้ตั้งค่าจำกัดอุณหภูมิไว้ที่ 34 องศาเซลเซียส)

ในช่วง 1 – 300 วินาทีแรกของการทดลอง เป็นการมอนิเตอร์สภาพแวดล้อมของระบบ ซึ่งมีอุณหภูมิประมาณ 32 – 33 องศาเซลเซียสจากนั้นได้เริ่มทำการเปิดแหล่งความร้อน ทำให้ช่วงเวลาวินาทีที่ 300 – 660 อุณหภูมิในโรงเรือนจำลองนั้น มีค่าขึ้นสูงอย่างต่อเนื่อง เมื่อไม่เปิดใช้งานระบบเฝ้าระวังและควบคุม อุณหภูมิจะเริ่มอิ่มตัวในช่วง 36 – 37 องศาเซลเซียส แต่เมื่อพิจารณาเส้นกราฟอุณหภูมิการเปิดใช้งานระบบเฝ้าระวังและควบคุมจะเห็นว่า เมื่ออุณหภูมิสูงขึ้นกว่าค่าที่ตั้งไว้ในโปรแกรมเฝ้าระวังและควบคุม จะถูกพัดลมระบายอากาศเป่าให้อากาศภายนอกเข้ามาภายในระบบ ทำให้อุณหภูมิถูกจำกัดไว้ค่าๆหนึ่ง ซึ่งจากเส้นกราฟพบว่า มีการแกว่งของอุณหภูมิ อยู่ในช่วง 33.7 องศาเซลเซียส ถึง 34.2 องศาเซลเซียส ซึ่งเป็นขีดความสามารถของเครื่องปรับอากาศที่สามารถควบคุมอุณหภูมิในระบบได้เพียงเท่านี้

นอกจากนี้ เมื่อพิจารณาเส้นกราฟความชื้นสัมพัทธ์นั้น ปรากฏว่า ความชื้นของระบบปิดกรณียังไม่ใช้งานระบบเฝ้าระวังและควบคุมมีค่าสูงกว่าระบบปิดกรณีใช้งานระบบเฝ้าระวังและควบคุม เนื่องจาก ระบบปิดที่ใช้งานเฝ้าระวังและควบคุมนั้น สามารถควบคุมให้อยู่ในระดับที่ต่ำกว่า ทำให้ความชื้นในระบบปิดมีค่าสูงกว่านั่นเอง

การทดลองสุดท้าย จากหัวข้อ 5.5.2 เป็นการทดสอบระบบเฝ้าระวังและควบคุมโดยการทดลองนี้เป็นการทดสอบการควบคุมความชื้นในโรงเรือนจำลองแบบปิด ซึ่งในการทดลองนี้ แบ่งการทดลองออกเป็นเป็น 3 ช่วง ในช่วงแรก เป็นการลดความชื้นในระบบ โดยใช้หลักการควบแน่นของไอน้ำในอากาศอย่างง่ายๆ จะเห็นว่าในช่วงวินาทีที่ 180 – 480 นั้น ความชื้นสัมพัทธ์ได้ลดลงจาก 70% ลงมาถึง 62% จากนั้นในช่วง วินาทีที่ 480 – 900 นั้น ความชื้นสัมพัทธ์ได้อิ่มตัวในช่วง 62% – 64% จากนั้น เป็นการทดลองในช่วงถัดมา โดยการทำการเปิดใช้งานระบบเฝ้าระวังและควบคุม โดยทำการกำหนดความชื้นในระบบไม่ให้ต่ำกว่า 70% สังเกตได้จากกราฟในช่วงวินาทีที่ 900 – 1620 นั้น ความชื้นในระบบปิดได้เพิ่มสูงขึ้นอยู่ในช่วง 69% - 73% ซึ่งนับว่าเป็นการควบคุมความชื้นในระบบปิดที่ค่าอยู่ในช่วงยอมรับได้ หลังจากนั้น เป็นการทดลองช่วงสุดท้าย โดยการกำหนดความชื้นในโปรแกรมเฝ้าระวังและควบคุม ไม่ให้ต่ำกว่า 80% สังเกตจากกราฟในช่วงวินาทีที่ 1620 – 2100 นั้น เส้นกราฟความชื้นจะสูงขึ้น ไปอยู่ในช่วง 78% - 82% ซึ่งถือว่าทำงานได้ปกติ แต่อาจจะมีความผิดพลาดเกิดขึ้นบ้าง เนื่องจากช่วงการใช้งานของโมดูล

ชนิดนี้ สามารถทำงานได้อย่างเป็นเชิงเส้น ในช่วงความชื้นสัมพัทธ์ 40% - 80% เท่านั้น หากความชื้นเกินมากกว่านี้ ข้อมูลอาจจะมีความผิดพลาดอยู่บ้าง

5.7 วิจารณ์ผลการทดสอบ

จากการทดสอบระบบเฝ้าระวังและควบคุมของโครงงานนี้ทั้งหมดนั้น มีหลายปัจจัย ที่ทำให้โครงงานนี้มีข้อจำกัด เช่น จากงบประมาณที่มีจำกัด ทำให้ไม่สามารถเลือกใช้อุปกรณ์เซ็นเซอร์ในระบบที่มีความแม่นยำมากได้ เช่น การใช้งานโมดูลวัดความชื้นนั้น ยังมีข้อจำกัดการใช้งานในช่วงความชื้นสัมพัทธ์ในอากาศที่ 40% - 80% หากความชื้นสัมพัทธ์มีค่ามากกว่านี้ หรือ ต่ำกว่านี้ จะทำให้การใช้งานมีความคลาดเคลื่อนมาก

โปรแกรมเฝ้าระวังและควบคุมที่ได้ออกแบบไว้นั้น มีประสิทธิภาพในการทำงานได้อย่างดีมาก แต่อย่างไรก็ตาม พบว่า โปรแกรมเฝ้าระวังและควบคุมใช้ทรัพยากรเครื่องเซิร์ฟเวอร์อย่างมาก จึงจำเป็นต้องใช้คอมพิวเตอร์เซิร์ฟเวอร์ที่มีความเร็วในการประมวลผลที่อยู่ในระดับสูง ซึ่งในอนาคต ผู้พัฒนาสามารถดัดแปลงอัลกอริทึม ของการการทำงานให้มีประสิทธิภาพมากขึ้น ซึ่งจะใช้ทรัพยากรเครื่องเซิร์ฟเวอร์น้อยลง

ระบบโครงงานต้นแบบดังกล่าวนี้ สามารถใช้งานได้มีประสิทธิภาพในระดับหนึ่งเท่านั้น เนื่องจากทางผู้จัดทำไม่สามารถทดสอบระบบโครงงานต้นแบบนี้ได้จริงในระบบปิดที่มีขนาดใหญ่ได้ เพียงแต่สามารถทดสอบได้เพียงระบบปิดเล็กๆ ที่ได้จำลองขึ้นมาสำหรับโครงงานต้นแบบนี้เท่านั้น แต่ผู้ใช้งานยังสามารถนำระบบต้นแบบนี้ไปพัฒนาให้เป็นระบบที่มีขนาดใหญ่ขึ้น หรือ ใช้ระบบโครงงานต้นแบบนี้ สำหรับเก็บข้อมูลทางวิทยาศาสตร์และเทคโนโลยี เช่น รูปแบบการเพาะเห็ดในเกษตรกรรม ที่มีลักษณะรูปแบบ ความชื้นกับอุณหภูมิ ที่มีค่าแตกต่างกัน ทำให้สามารถนำข้อมูลมาเปรียบกับผลผลิตที่ได้ ทำให้สามารถคิดค้นวิธีการเพาะเห็ดที่สามารถเพิ่มผลผลิตให้กับเกษตรกรได้มากขึ้น

5.8 บทสรุป

สรุปโครงงานต้นแบบระบบเครือข่ายเซนเซอร์แบบไร้สายซึ่งพัฒนาบนระบบเครือข่ายซิกบีที่ได้นำเสนอสามารถทำงานได้ตามที่ออกแบบ ระบบดังกล่าวยังสามารถนำไปประยุกต์ใช้งานระบบเฝ้าระวังและควบคุมอื่นๆได้ โดยเปลี่ยนชนิดของเซ็นเซอร์และอุปกรณ์ควบคุมให้เหมาะสมกับลักษณะของงาน เช่น ระบบตรวจสอบแก๊สพิษในโรงงานอุตสาหกรรม ระบบตรวจสอบมลพิษทางน้ำและอากาศ เป็นต้น

นอกจากนี้ โปรแกรมเฝ้าระวังและควบคุม ที่ได้ออกแบบไว้ ในอนาคตสามารถรองรับการทำงานที่มีจำนวนโหนดในเครือข่ายมากขึ้น ทำให้สามารถใช้กับโรงเรือนที่มีขนาดใหญ่ขึ้น หรือ สามารถเพิ่มจำนวนโหนดเซ็นเซอร์ ที่สามารถนำไปใช้กับเซ็นเซอร์อื่นๆได้หลายแบบ ดังนั้นโปรแกรมเฝ้าระวังและควบคุมที่ได้ออกแบบไว้นั้น จึงมีความยืดหยุ่นสูง และง่ายต่อการใช้ประยุกต์ใช้ในอนาคต

เอกสารอ้างอิง

[1] Shahin Farahani, Zigbee Wireless Network and Transceiver, Newnes, Oxford, UK, 2008

[2] X-Bee Basic Configuration in Network Application, available at http://www.thaieasyelec.com

[3] X-Bee / X-Bee-Pro ZB RF Modules, Zigbee RF Modules, Digi International, 2009

[4] MAXIM, MAX3222/MAX3232/MAX3237/MAX3241 Technical Datasheet, 2003

[5] How to use RS232, available at http://www.thaimicrotron.com

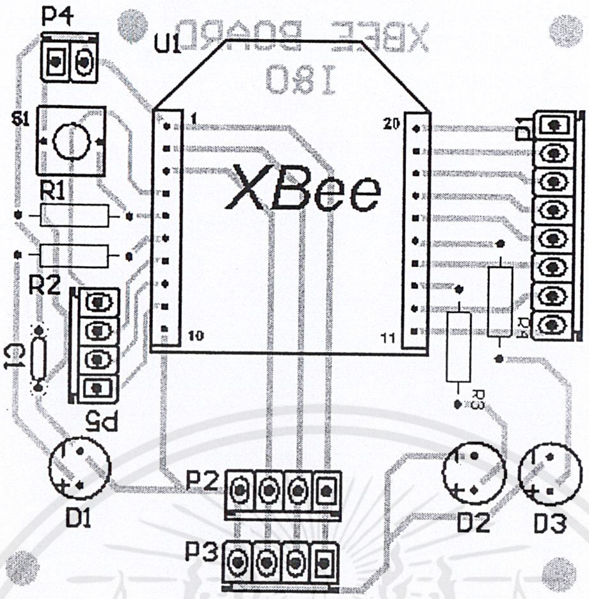
[6] ARM-based 32-bit MCU, STM32F103x8 Technical Datasheet, 2007

[7] PIR Motion Sensor, available at http://www.electronics123.com

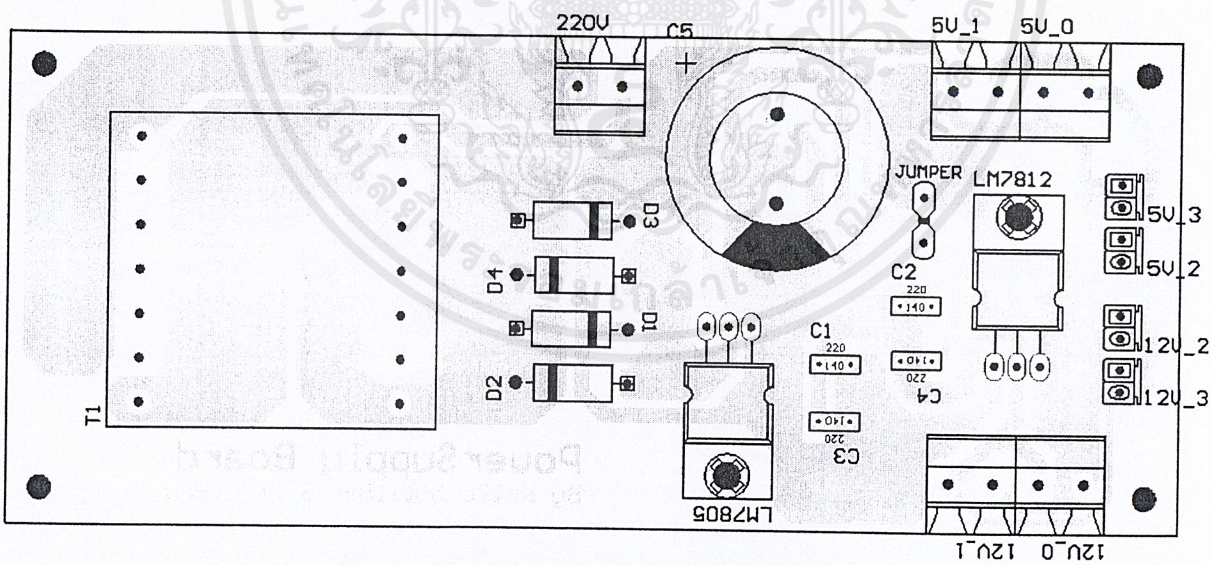
[8] ZX-PIR Manual, INEX, available at http://www.inex.co.th

[9] DS18B20+ Technical Datasheet, DALLDAS Semiconductor, 2008

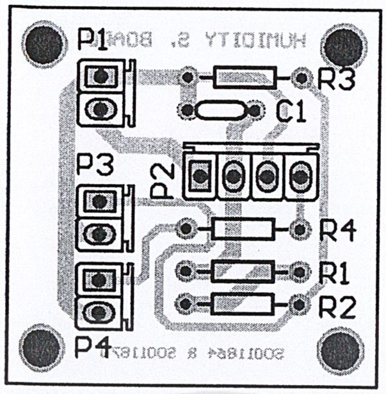
ภาคผนวก (ก)
ลายวงจรที่ใช้งาน



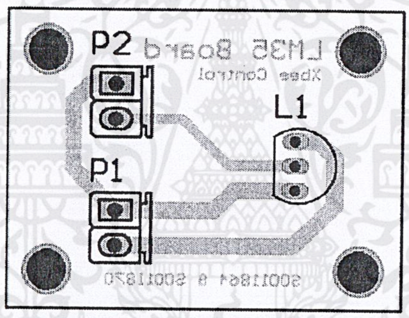
ลายวงจรบอร์ด X-Bee สำหรับ รับ-ส่งข้อมูล สำหรับโหนดปลายทาง



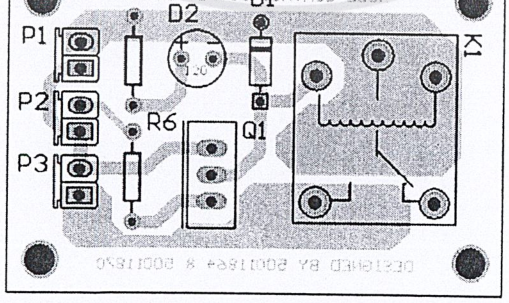
ลายวงจรบอร์ด Power Supply สำหรับ โหนดปลายทาง



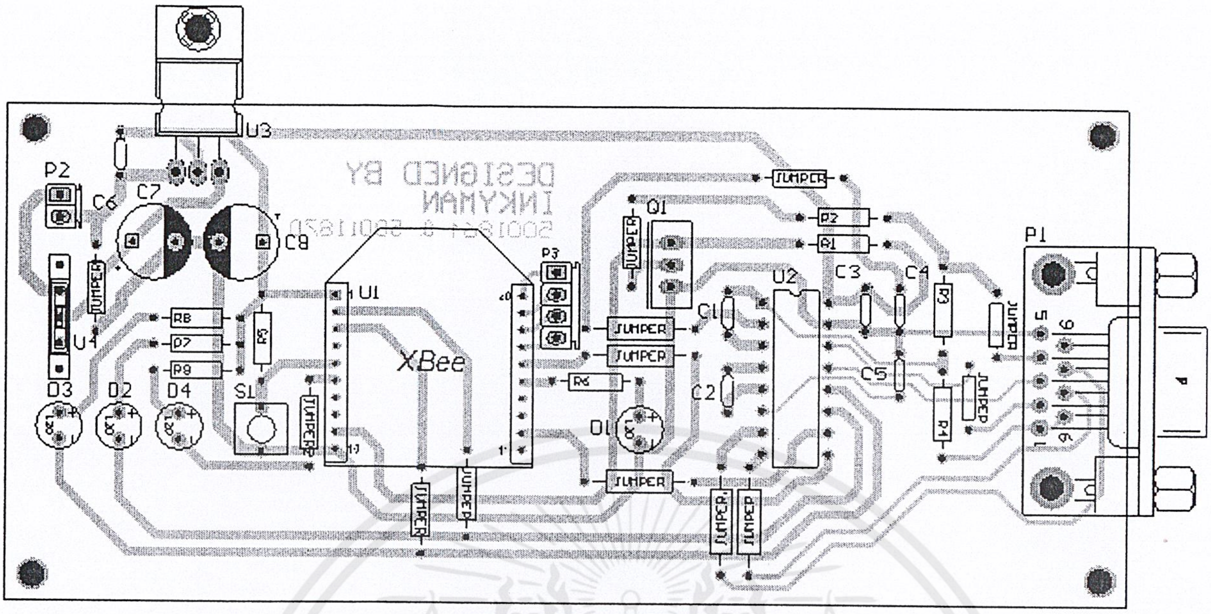
ลายวงจรหาร3 สำหรับเซ็นเซอร์วัดความชื้น



ลายวงจรเซ็นเซอร์วัดอุณหภูมิ



ลายวงจรรีเลย์ สำหรับควบคุมอุปกรณ์เชื่อมต่อภายนอก



ลายวงจรบอร์ดตั้งค่าโมดูล X-Bee

ภาคผนวก (ข)
โปรแกรมที่ใช้ในการควบคุมไมโครคอนโทรลเลอร์โครงงานเบื้องต้น

```
//-------------------------------------- Header -------------------------------------------------//
#include "stm32f10x_lib.h"      // Include STM32 header file
#include "glcd5110.h"           // Include library file for GLCD5110
#include "string.h"             //Include library file for string function

//--------------------------- Define for Joystick --------------------------------------------//
#define JOY_UP  1
#define JOY_DOWN  2
#define JOY_LEFT  3
#define JOY_RIGHT  4
#define JOY_SELECT  5
#define JOY_NONE  0

//---------------------------- Function Delay 1MS -----------------------------------------//
void delay(unsigned long ms)  // delay 1 ms per count @ Crystal 8.0 MHz and PLL9x or SYSCLK = 72 MHz
{
        volatile unsigned long i,j;
        for (i = 0; i < ms; i++ )
        for (j = 0; j < 5525; j++ );
}

//--------------------------- Function RCC Configuration -----------------------------------//
void RCC_setup()
{
  ErrorStatus HSEStartUpStatus;                    //  Keep error status
  RCC_DeInit();                                    //  RCC system reset(for debug purpose)
  RCC_HSEConfig(RCC_HSE_ON);                       //  Enable HSE
  HSEStartUpStatus = RCC_WaitForHSEStartUp();      //  Wait till HSE is ready
  if(HSEStartUpStatus == SUCCESS)
  {
    RCC_HCLKConfig(RCC_SYSCLK_Div1);           // HCLK = SYSCLK
    RCC_PCLK2Config(RCC_HCLK_Div1);            // PCLK2 = HCLK
    RCC_PCLK1Config(RCC_HCLK_Div2);            // PCLK1 = HCLK/2
    RCC_ADCCLKConfig(RCC_PCLK2_Div4);          // ADCCLK = PCLK2/4
    FLASH_SetLatency(FLASH_Latency_2);         // Flash 2 wait state
    FLASH_PrefetchBufferCmd(FLASH_PrefetchBuffer_Enable);    // Enable Prefetch Buffer
    RCC_PLLConfig(RCC_PLLSource_HSE_Div1,RCC_PLLMul_9);   // PLLCLK = 8MHz * 9 = 72 MHz

    RCC_PLLCmd(ENABLE);                        // Enable PLL
    while(RCC_GetFlagStatus(RCC_FLAG_PLLRDY) == RESET);   // Wait until PLL is ready

    RCC_SYSCLKConfig(RCC_SYSCLKSource_PLLCLK);  // Select PLL as system clock source
    while(RCC_GetSYSCLKSource() != 0x08); // Wait till PLL is used as system clock source
  }
}

//---------------------------- Function Timer1 Configuration ---------------------------//
void TIM4_setup()
{
  TIM_TimeBaseInitTypeDef  TIM_TimeBaseStructure;
  TIM_OCInitTypeDef  TIM_OCInitStructure;
  // Enable TIM4 clock
  RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM4,ENABLE);
  // Time base configuration
  TIM_TimeBaseStructure.TIM_Prescaler = 71;  //  1 MHz
  TIM_TimeBaseStructure.TIM_Period = 999;  // Period = 1000 count Frequency 10 kHz(1 MHz/100)
  TIM_TimeBaseStructure.TIM_ClockDivision = 0x0;
```

```
  TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up;
  TIM_TimeBaseInit(TIM4, &TIM_TimeBaseStructure);
  // PWM1 Mode configuration: Channel3
  TIM_OCInitStructure.TIM_OCMode = TIM_OCMode_PWM1;
  TIM_OCInitStructure.TIM_Channel = TIM_Channel_3;
  TIM_OCInitStructure.TIM_Pulse = 500;
  TIM_OCInit(TIM4, &TIM_OCInitStructure);
  TIM_OC3PreloadConfig(TIM4, TIM_OCPreload_Enable);
  TIM_ARRPreloadConfig(TIM4, ENABLE); // Enable Preload Auto reload register
}

//------------------------------ Function GPIO Configuration -----------------------------------------//
void GPIO_setup()
{
  GPIO_InitTypeDef GPIO_InitStructure;
  // Enable GPIOC clock
  RCC_APB2PeriphClockCmd( RCC_APB2Periph_GPIOA |
                    RCC_APB2Periph_GPIOB |
                    RCC_APB2Periph_GPIOC |
                    RCC_APB2Periph_GPIOD |
                    RCC_APB2Periph_GPIOE |
                    RCC_APB2Periph_AFIO
                    ,ENABLE);

  GPIO_PinRemapConfig(GPIO_Remap_USART2,ENABLE);

  // Configure USART1 Tx (PA9) as alternate function push-pull
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_9;
  GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
  GPIO_Init(GPIOA, &GPIO_InitStructure);
  // Configure USART1 Rx (PA10) as input floating
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_10;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
  GPIO_Init(GPIOA, &GPIO_InitStructure);

  // Configure USART2 Tx (PD5) as alternate function push-pull
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_5;
  GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
  GPIO_Init(GPIOD, &GPIO_InitStructure);
  // Configure USART2 Rx (PD6) as input floating
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_6;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
  GPIO_Init(GPIOD, &GPIO_InitStructure);

  // GPIOB Configuration: TIM4 channel3  // For sound
  GPIO_InitStructure.GPIO_Pin =  GPIO_Pin_8;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
  GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
  GPIO_Init(GPIOB, &GPIO_InitStructure);

  // Configure PC4 as analog input(ADC Channel14)
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_4;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AIN;
  GPIO_Init(GPIOC, &GPIO_InitStructure);
```

```
  // Configure PC6 as Output push-pull(connect with LED)
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_6 | GPIO_Pin_9;;
  GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
  GPIO_Init(GPIOC, &GPIO_InitStructure);

  // Configure PE2,PE3,PE4 and PE5 as Output push-pull(connect with LED)
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_2 | GPIO_Pin_3 | GPIO_Pin_4 | GPIO_Pin_5;
  GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
  GPIO_Init(GPIOE, &GPIO_InitStructure);

 // Configure the JoyStick IOs
 // Key up + Key down
 GPIO_InitStructure.GPIO_Pin = GPIO_Pin_8 | GPIO_Pin_14 | GPIO_Pin_12;
 GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
 GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IPU;
 GPIO_Init(GPIOD, &GPIO_InitStructure);

 // Key left + Key right
 GPIO_InitStructure.GPIO_Pin = GPIO_Pin_1 | GPIO_Pin_0;
 GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
 GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IPU;
 GPIO_Init(GPIOE, &GPIO_InitStructure);

  // Configure PC13 as input floating (EXTI Line13)
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_13;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
  GPIO_Init(GPIOC, &GPIO_InitStructure);

  // Configure PA0 as input floating (EXTI Line0)
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_0;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
  GPIO_Init(GPIOA, &GPIO_InitStructure);
}

//----------------------------- Function NVIC Configuration -----------------------------------------//
void NVIC_setup()
{
  NVIC_InitTypeDef NVIC_InitStructure;

  // Configure one bit for preemption priority
  NVIC_PriorityGroupConfig(NVIC_PriorityGroup_1);
  // Enable the EXTI0 Interrupt
  NVIC_InitStructure.NVIC_IRQChannel = EXTI0_IRQChannel;
  NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
  NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;
  NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
  NVIC_Init(&NVIC_InitStructure);
  // Enable the EXTI1 Interrupt
  NVIC_InitStructure.NVIC_IRQChannel = EXTI1_IRQChannel;
  NVIC_InitStructure.NVIC_IRQChannelSubPriority = 1;
  NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
  NVIC_Init(&NVIC_InitStructure);
}
```

```
//------------------------------ Function EXTI Configuration -----------------------------------------//
void EXTI_setup()
{
   EXTI_InitTypeDef EXTI_InitStructure;

   // Enable AFIO for EXTI module
   RCC_APB2PeriphClockCmd(RCC_APB2Periph_AFIO,ENABLE);
   GPIO_EXTILineConfig(GPIO_PortSourceGPIOE, GPIO_PinSource0);    // Connect EXTI Line0 to PA0
   // Configure EXTI Line0 to generate an interrupt on falling edge
   EXTI_InitStructure.EXTI_Line = EXTI_Line0;
   EXTI_InitStructure.EXTI_Mode = EXTI_Mode_Interrupt;
   EXTI_InitStructure.EXTI_Trigger = EXTI_Trigger_Falling;
   EXTI_InitStructure.EXTI_LineCmd = ENABLE;
   EXTI_Init(&EXTI_InitStructure);

   GPIO_EXTILineConfig(GPIO_PortSourceGPIOE, GPIO_PinSource1);   // Connect EXTI Line13 to PC13
   // Configure EXTI Line0 to generate an interrupt on falling edge
   EXTI_InitStructure.EXTI_Line = EXTI_Line1;
   EXTI_InitStructure.EXTI_Mode = EXTI_Mode_Interrupt;
   EXTI_InitStructure.EXTI_Trigger = EXTI_Trigger_Falling;
   EXTI_InitStructure.EXTI_LineCmd = ENABLE;
   EXTI_Init(&EXTI_InitStructure);
}

//------------------------------ Function USART1 Configuration ------------------------------------//
void USART1_setup()
{
   USART_InitTypeDef USART_InitStructure;
   // Enable USART1 clock
   RCC_APB2PeriphClockCmd(RCC_APB2Periph_USART1,ENABLE);
   // USART1 configured
   USART_InitStructure.USART_BaudRate = 9600;
   USART_InitStructure.USART_WordLength = USART_WordLength_8b;
   USART_InitStructure.USART_StopBits = USART_StopBits_1;
   USART_InitStructure.USART_Parity = USART_Parity_No;
   USART_InitStructure.USART_HardwareFlowControl = USART_HardwareFlowControl_None;
   USART_InitStructure.USART_Mode = USART_Mode_Rx | USART_Mode_Tx;
   USART_InitStructure.USART_Clock = USART_Clock_Disable;
   USART_InitStructure.USART_CPOL = USART_CPOL_Low;
   USART_InitStructure.USART_CPHA = USART_CPHA_2Edge;
   USART_InitStructure.USART_LastBit = USART_LastBit_Disable;

   USART_Init(USART1, &USART_InitStructure); // Configure the USART1
   // Enable the USART1 Transmit and Receive interrupt
   USART_ITConfig(USART1, USART_IT_TXE, DISABLE);
   // Enable the USART Receive interrupt
   USART_ITConfig(USART1, USART_IT_RXNE, ENABLE);
   USART_Cmd(USART1,ENABLE);  // Enable USART1
}
void USART2_setup()
{
   USART_InitTypeDef USART_InitStructure;
   // Enable USART2 clock
   RCC_APB1PeriphClockCmd(RCC_APB1Periph_USART2,ENABLE);
   // USART2 configured
   USART_InitStructure.USART_BaudRate = 9600;
   USART_InitStructure.USART_WordLength = USART_WordLength_8b;
```

```
    USART_InitStructure.USART_StopBits = USART_StopBits_1;
    USART_InitStructure.USART_Parity = USART_Parity_No;
    USART_InitStructure.USART_HardwareFlowControl = USART_HardwareFlowControl_None;
    USART_InitStructure.USART_Mode = USART_Mode_Rx | USART_Mode_Tx;
    USART_InitStructure.USART_Clock = USART_Clock_Disable;
    USART_InitStructure.USART_CPOL = USART_CPOL_Low;
    USART_InitStructure.USART_CPHA = USART_CPHA_2Edge;
    USART_InitStructure.USART_LastBit = USART_LastBit_Disable;

    USART_Init(USART2, &USART_InitStructure); // Configure the USART2
    USART_Cmd(USART2, ENABLE);  // Enable USART2
  }

//----------------------------- Function USART1,2 send string ---------------------------------------//
void usart1_putc(unsigned char c)
{
  while(USART_GetFlagStatus(USART1,USART_FLAG_TXE)==RESET);   // Wait until transmition ready
  USART_SendData(USART1,(int)c);   // Send character
}
void usart2_putc(unsigned char c)
{
  while(USART_GetFlagStatus(USART2,USART_FLAG_TXE)==RESET);   // Wait until transmition ready
  USART_SendData(USART2,(int)c);   // Send character
}
void usart1_puthex(u8 c)
{
  while(USART_GetFlagStatus(USART1,USART_FLAG_TXE)==RESET);   // Wait until transmition ready
  USART_SendData(USART1,c);        // Send Hex
}

//----------------------------- Function USART1,2 send string ---------------------------------------//
void usart1_puts(unsigned char * s)
{
  while(*s)            // Check end of string
  {
    usart1_putc(*s++);   // Send charracter 1 time
  }
}
void usart2_puts(unsigned char *s)
{
  while(*s)            // Check end of string
  {
    usart2_putc(*s++);   // Send charracter 1 time
  }
}

//----------------------------- Function USART1,2 wait character ------------------------------------//
int usart1_getc()
{
  while(USART_GetFlagStatus(USART1,USART_FLAG_RXNE)==RESET);   // Wait until receive data
  return(USART_ReceiveData(USART1)); // Return  character
}
int usart2_getc()
{
  while(USART_GetFlagStatus(USART2,USART_FLAG_RXNE)==RESET);   // Wait until receive data
  return(USART_ReceiveData(USART2)); // Return  character
}
```

```
//------------------------------------------ Sound ---------------------------------------------------//
void sound(int freq,long tm)
{
   int newPeriod;
   newPeriod = (u16)(1000000/freq)-1;  // Calculate new period value to set
    TIM_SetAutoreload(TIM4,newPeriod);  // Set new period value
    TIM_SetCompare3(TIM4,newPeriod/2);  // Set duty cycle maintain at 50%
    TIM_Cmd(TIM4, ENABLE); // TIM4 signal enable
    delay(tm);             // Delay for generate signal 1 sec
    TIM_Cmd(TIM4, DISABLE); // TIM4 signal disable
}

//------------------------------------------ ETC,, ---------------------------------------------------//
void lcd_initial()
{
   lcdString(1,2,"   ARM7 CPU   ");        // Set message on line 3
   lcdString(1,3," Initializing ");  // Set message on line 3
   lcdString(1,4," Please Wait..");          // Set message on line 4
   lcdBackLight(1);
   lcdUpdate();
   delay(5000);
   sound(500,100);
}

void lcd_startup_const()
{
   lcdString(1,1,"ZIGBEE APP");     // Set message on line 1
   lcdString(1,2,"Monitoring...."); // Set message on line 2
   lcdString(1,3,"--------------");  // Set message on line 3
   lcdString(1,4,"PIR : none    ");// Set message on line 4
   lcdString(1,5,"ANALOG : none ");       // Set message on line 5
   lcdBackLight(1);
   lcdUpdate();                 // Display message
}

//--------------------------------- Function ISR line 0 -----------------------------------------//
void EXTI0_IRQHandler(void)
{
 if(EXTI_GetITStatus(EXTI_Line0) != RESET)  // Ensure get EXTI line 0 pending bit RIGHT
 {
   // Toggle PC6 pin
   //GPIO_WriteBit(GPIOC, GPIO_Pin_6, (BitAction)((1-GPIO_ReadOutputDataBit(GPIOC, GPIO_Pin_6))));
   sound(4000,30);
   lcdClear();
   lcdUpdate();

   char strx[15];
   int percentage=0;
   while(1)
   {
         lcdString(1,2," X-Bee Network ");        // Set message on line 1
   lcdString(1,3,"Identifying...");// Set message on line 1
   sprintf(strx,"    %d%%      ",percentage);   // Convert string to display percentage
         lcdString(1,4,strx);                   // Set percentage value display on line 1
         lcdProgBar(0,34,83,3,percentage);      // Display progress bar at(0,10),width 83 pixel,high 5 pixel

         lcdUpdate();                           // Display message
```

```
        delay(20);                              // Delay for display
        percentage++;                           // Increase percentage
      if(percentage==100)                       // Check percentage over 100?
      { break; }
    }
    usart1_puthex(0x7E);
    usart1_puthex(0x00);
    usart1_puthex(0x04);
    usart1_puthex(0x08);
    usart1_puthex(0x01);
    usart1_puthex(0x4E);
    usart1_puthex(0x44);
    usart1_puthex(0x64);
    delay(10000);
    lcd_startup_const();
    EXTI_ClearITPendingBit(EXTI_Line0);         // Clear the EXTI line 0 pending bit
  }
}

//-------------------------------- Function ISR line 10 to 15 ---------------------------------//
void EXTI1_IRQHandler(void)
{
  if(EXTI_GetITStatus(EXTI_Line1) != RESET) // Ensure get EXTI line 1 pending bit LEFT
  {
    // Toggle PC9 pin
    GPIO_WriteBit(GPIOC, GPIO_Pin_9, (BitAction)((1-GPIO_ReadOutputDataBit(GPIOC, GPIO_Pin_9))));
    EXTI_ClearITPendingBit(EXTI_Line1);  // Clear the EXTI line 13 pending bit
  }
}

//-------------------------------------- MAIN PROGRAM --------------------------------------------//
void prog_main()
{
  int i,j;                // Keep character
  char str[50];           // Keep string convert
  start:                  // Label
  while(1)                // Infinite loop
  {
    lcdBackLight(1);
    i=0;
    while(i<=21)
    {
      str[i] = usart1_getc();     // Wait until receive character
      if(str[0]!=0x7E) {          // Check start with 0x7E
      goto start;
      }
      else {
        i++;
      }
    }

    int length  = str[2];
    int duty    = str[3];
    char addb7  = str[4];
    char addb6  = str[5];
    char addb5  = str[6];
    char addb4  = str[7];
```

```
char addb3  = str[8];
char addb2  = str[9];
char addb1  = str[10];
char addb0  = str[11];
int dcmH    = str[16];
int dcmL    = str[17];
int acm     = str[18];
char dataH  = str[19];
char dataL  = str[20];
int chksum  = str[21];
u8 sumall   = 0x00;

for(j=3;j<=length+2;j++)
{
  sumall = sumall + str[j];
}

sumall        =  sumall & 0x00FF;
u8 chkmul     =  0xFF - sumall;     // sum all with 8bit LSB only
u8 delResult  =  chksum - chkmul;

if(delResult!=0x00) {
  goto start;
}

//CASE ROUTER DETECTION (SH=0013A200,SL=403DC71D)
if(addb0==(char)0x1D)
{
  if(dataH==(char)0x00 && dataL==(char)0x00) {
    lcdString(1,4,"PIR : none  ");   // Set message on line 4
    lcdUpdate();                     // Display message
    goto start;
  }
  else if (dataH==(char)0x00 && dataL==(char)0x01) {
    lcdString(1,4,"PIR : detect");   // Set message on line 4
    lcdUpdate();                     // Display message
    sound(2000,150);
    goto start;
  }
  goto start;
}

//CASE ROUTER ANALOG SENSOR (SH=0013A200,SL=4055B1CC)
if(addb0==(char)0xCC)
{
  int byteHigh = dataH<<8;           // shift left 8 bit
  int byteLow  = dataL&0x00FF;       // Clear to 1byte only
  int byteReal = byteHigh|byteLow;   // OR together

  char decReal[3];
  sprintf(decReal,"%d   ",byteReal); // Convert analog value to string

  lcdString(1,5,"ANALOG : ");        // Set message on line 2
  lcdGotoXY(10,5);
  lcdString(10,5,decReal);
  lcdUpdate();                       // Display message
  goto start;
```

```
    }
  }
}

//--------------------------------------- Main Program ---------------------------------------------//
int main()
{
  RCC_setup();          // Reset/Clock Configuration
  GPIO_setup();         // GPIO Configuration
  USART1_setup();       // USART1 Configuration
  USART2_setup();       // USART2 Configuration
  TIM4_setup();         // TIMER4 Configuration for PWM
  EXTI_setup();         // EXTI configuration
  NVIC_setup();         // NVIC configuration

  lcd_initial();
  lcd_startup_const();

  while(1)
  {
    prog_main();
  }

}
```

ภาคผนวก (ค)
โปรแกรมที่ใช้ในการควบคุมไมโครคอนโทรลเลอร์ (Gateway)

```
//--------------------------------------------------------------------------------------------------//
// Program              :  Zigbee Gateway
// Description          : by Inkyman (Electronics Engineering 4C)
// Filename             : usart_01.c
// C complier           : RkitARM 1.03.0003 for RIDE7
// Firmwar Library      : STMicroelectronics FWLib V1.0
//--------------------------------------------------------------------------------------------------//
#include "stm32f10x_lib.h"      // Include STM32 header file
#include "glcd5110.h"           // Include library file for GLCD5110
#include "string.h"     //Include library file for string function
#define TxBufferSize   100 // Set TX buffer(maximum transmission byte)
#define TxBufferSize2  100 // Set TX buffer(maximum transmission byte)

u8 TxBuffer[TxBufferSize]={0}; // Initialize all data to zero UART1
u8 TxBuffer2[TxBufferSize2]={0}; // Initialize all data to zero UART2

vu8 TxCounter=0;  // Keep counter for transmission data UART1
vu8 RxMessage=0;  // Keep data from receive UART1
vu8 RxUpdate=0;   // Flag for indicate data incomming UART1
vu8 TxReady=1;    // Flag for indicate TX ready UART1

vu8 TxCounter2=0;  // Keep counter for transmission data UART2
vu8 RxMessage2=0;  // Keep data from receive UART2
vu8 RxUpdate2=0;   // Flag for indicate data incomming UART2
vu8 TxReady2=1;    // Flag for indicate TX ready UART2

unsigned long adc_val=0; // Keep analog value
char str[14];  // Keep string convert

//--------------------------------------------------------------------------------------------------//
//------------------------------ Function Delay 1MS -------------------------------------------------//
//--------------------------------------------------------------------------------------------------//
void delay(unsigned long ms)  // delay 1 ms per count @ Crystal 8.0 MHz and PLL9x or SYSCLK = 72 MHz
{
        volatile unsigned long i,j;
        for (i = 0; i < ms; i++ )
        for (j = 0; j < 5525; j++ );
}
//--------------------------------------------------------------------------------------------------//
//------------------------------ Function RCC Configuration -----------------------------------------//
//--------------------------------------------------------------------------------------------------//
void RCC_setup()
{
  ErrorStatus HSEStartUpStatus; // Keep error status
  RCC_DeInit();      // RCC system reset(for debug purpose)
  RCC_HSEConfig(RCC_HSE_ON);  // Enable HSE
  HSEStartUpStatus = RCC_WaitForHSEStartUp();   // Wait till HSE is ready
  if(HSEStartUpStatus == SUCCESS)
  {
    RCC_HCLKConfig(RCC_SYSCLK_Div1); // HCLK = SYSCLK
    RCC_PCLK2Config(RCC_HCLK_Div1); // PCLK2 = HCLK
    RCC_PCLK1Config(RCC_HCLK_Div2); // PCLK1 = HCLK/2
    RCC_ADCCLKConfig(RCC_PCLK2_Div4);  // ADCCLK = PCLK2/4
    FLASH_SetLatency(FLASH_Latency_2); // Flash 2 wait state
    FLASH_PrefetchBufferCmd(FLASH_PrefetchBuffer_Enable);  // Enable Prefetch Buffer
    RCC_PLLConfig(RCC_PLLSource_HSE_Div1,RCC_PLLMul_9);   // PLLCLK = 8MHz * 9 = 72 MHz
```

```
    RCC_PLLCmd(ENABLE);  // Enable PLL
    while(RCC_GetFlagStatus(RCC_FLAG_PLLRDY) == RESET);   // Wait until PLL is ready

    RCC_SYSCLKConfig(RCC_SYSCLKSource_PLLCLK);  // Select PLL as system clock source
    while(RCC_GetSYSCLKSource() != 0x08); // Wait till PLL is used as system clock source
  }
}
//----------------------------------------------------------------------------------------------------//
//----------------------------- Function Timer1 Configuration ----------------------------//
//----------------------------------------------------------------------------------------------------//
void TIM4_setup()
{
  TIM_TimeBaseInitTypeDef  TIM_TimeBaseStructure;
  TIM_OCInitTypeDef  TIM_OCInitStructure;
  // Enable TIM4 clock
  RCC_APB1PeriphClockCmd(RCC_APB1Periph_TIM4,ENABLE);
  // Time base configuration
  TIM_TimeBaseStructure.TIM_Prescaler = 71;  //  1 MHz
  TIM_TimeBaseStructure.TIM_Period = 999;  // Period = 1000 count Frequency 10 kHz(1 MHz/100)
  TIM_TimeBaseStructure.TIM_ClockDivision = 0x0;
  TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up;
  TIM_TimeBaseInit(TIM4, &TIM_TimeBaseStructure);
  // PWM1 Mode configuration: Channel3
  TIM_OCInitStructure.TIM_OCMode = TIM_OCMode_PWM1;
  TIM_OCInitStructure.TIM_Channel = TIM_Channel_3;
  TIM_OCInitStructure.TIM_Pulse = 500;
  TIM_OCInit(TIM4, &TIM_OCInitStructure);
  TIM_OC3PreloadConfig(TIM4, TIM_OCPreload_Enable);
  TIM_ARRPreloadConfig(TIM4, ENABLE); // Enable Preload Auto reload register
}
//----------------------------------------------------------------------------------------------------//
//----------------------------- Function GPIO Configuration ----------------------------------------//
//----------------------------------------------------------------------------------------------------//
void GPIO_setup()
{
  GPIO_InitTypeDef GPIO_InitStructure;
  // Enable GPIOC clock
  RCC_APB2PeriphClockCmd( RCC_APB2Periph_GPIOA |
                          RCC_APB2Periph_GPIOB |
                          RCC_APB2Periph_GPIOC |
                          RCC_APB2Periph_GPIOE
                          ,ENABLE);
  RCC_APB2PeriphClockCmd( RCC_APB2Periph_GPIOD | RCC_APB2Periph_AFIO, ENABLE);
  GPIO_PinRemapConfig(GPIO_Remap_USART2,ENABLE);

  // Configure USART1 Tx (PA9) as alternate function push-pull
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_9;
  GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
  GPIO_Init(GPIOA, &GPIO_InitStructure);

  // Configure USART1 Rx (PA10) as input floating
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_10;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
  GPIO_Init(GPIOA, &GPIO_InitStructure);

  // Configure USART2 Tx (PD5) as alternate function push-pull
```

```
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_5;
  GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
  GPIO_Init(GPIOD, &GPIO_InitStructure);

  // Configure USART2 Rx (PD6) as input floating
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_6;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_IN_FLOATING;
  GPIO_Init(GPIOD, &GPIO_InitStructure);

  // Configure PC4 as analog input(ADC Channel14)
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_4;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AIN;
  GPIO_Init(GPIOC, &GPIO_InitStructure);

  // Configure PC6 as Output push-pull(connect with LED)
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_6;
  GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
  GPIO_Init(GPIOC, &GPIO_InitStructure);

  // Configure PE7 PORT RESET DSPIC
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_7;
  GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
  GPIO_Init(GPIOE, &GPIO_InitStructure);

  // Configure PE2,PE3,PE4 and PE5 as Output push-pull(connect with LED)
  GPIO_InitStructure.GPIO_Pin = GPIO_Pin_2 | GPIO_Pin_3 | GPIO_Pin_4 | GPIO_Pin_5;
  GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_Out_PP;
  GPIO_Init(GPIOE, &GPIO_InitStructure);

  // GPIOB Configuration: TIM4 channel3  // For sound
  GPIO_InitStructure.GPIO_Pin =  GPIO_Pin_8;
  GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP;
  GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
  GPIO_Init(GPIOB, &GPIO_InitStructure);
}
//-------------------------------------------------------------------------------------------------//
//----------------------------- Function ADC Configuration ------------------------------------------//
//-------------------------------------------------------------------------------------------------//
void ADC_setup()
{
  ADC_InitTypeDef ADC_InitStructure;
  // Enable ADC1 clock
  RCC_APB2PeriphClockCmd(RCC_APB2Periph_ADC1,ENABLE);
  // ADC1 configuration
  ADC_InitStructure.ADC_Mode = ADC_Mode_Independent;
  ADC_InitStructure.ADC_ScanConvMode = DISABLE;
  ADC_InitStructure.ADC_ContinuousConvMode = DISABLE;
  ADC_InitStructure.ADC_ExternalTrigConv = ADC_ExternalTrigConv_None;
  ADC_InitStructure.ADC_DataAlign = ADC_DataAlign_Right;
  ADC_InitStructure.ADC_NbrOfChannel = 1;
  ADC_Init(ADC1, &ADC_InitStructure);
  // ADC1 regular channel14 configuration
  ADC_RegularChannelConfig(ADC1, ADC_Channel_14, 1, ADC_SampleTime_13Cycles5);
```

```
  ADC_Cmd(ADC1, ENABLE);  // Enable ADC1

  ADC_ResetCalibration(ADC1);   // Enable ADC1 reset calibaration register
  while(ADC_GetResetCalibrationStatus(ADC1));

  ADC_StartCalibration(ADC1);   // Start ADC1 calibaration
  while(ADC_GetCalibrationStatus(ADC1)); // Check the end of ADC1 calibration
}
//-------------------------------------------------------------------------------------------------//
//----------------------------- Function NVIC Configuration ----------------------------------------//
//-------------------------------------------------------------------------------------------------//
void NVIC_setup(void)
{
  NVIC_InitTypeDef NVIC_InitStructure;
  // Enable the USART1 Interrupt
  NVIC_InitStructure.NVIC_IRQChannel = USART1_IRQChannel;
  NVIC_InitStructure.NVIC_IRQChannelPreemptionPriority = 0;
  NVIC_InitStructure.NVIC_IRQChannelSubPriority = 0;
  NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
  NVIC_Init(&NVIC_InitStructure);
  NVIC_InitStructure.NVIC_IRQChannelSubPriority = 4;
  NVIC_InitStructure.NVIC_IRQChannel = USART2_IRQChannel;
  NVIC_InitStructure.NVIC_IRQChannelCmd = ENABLE;
  NVIC_Init(&NVIC_InitStructure);
}
//-------------------------------------------------------------------------------------------------//
//----------------------------- Function USART1 Configuration ---------------------------------------//
//-------------------------------------------------------------------------------------------------//
void USART1_setup()
{
  USART_InitTypeDef USART_InitStructure;
  // Enable USART1 clock
  RCC_APB2PeriphClockCmd(RCC_APB2Periph_USART1,ENABLE);
  // USART1 configured
  USART_InitStructure.USART_BaudRate = 9600;
  USART_InitStructure.USART_WordLength = USART_WordLength_8b;
  USART_InitStructure.USART_StopBits = USART_StopBits_1;
  USART_InitStructure.USART_Parity = USART_Parity_No;
  USART_InitStructure.USART_HardwareFlowControl = USART_HardwareFlowControl_None;
  USART_InitStructure.USART_Mode = USART_Mode_Rx | USART_Mode_Tx;
  USART_InitStructure.USART_Clock = USART_Clock_Disable;
  USART_InitStructure.USART_CPOL = USART_CPOL_Low;
  USART_InitStructure.USART_CPHA = USART_CPHA_2Edge;
  USART_InitStructure.USART_LastBit = USART_LastBit_Disable;

  USART_Init(USART1, &USART_InitStructure); // Configure the USART1
  // Enable the USART1 Transmit and Receive interrupt
  USART_ITConfig(USART1, USART_IT_TXE, DISABLE);
  // Enable the USART Receive interrupt
  USART_ITConfig(USART1, USART_IT_RXNE, ENABLE);
  USART_Cmd(USART1,ENABLE);  // Enable USART1
}
void USART2_setup()
{
  USART_InitTypeDef USART_InitStructure;
  // Enable USART2 clock
  RCC_APB1PeriphClockCmd(RCC_APB1Periph_USART2,ENABLE);
```

```
  // USART2 configured
  USART_InitStructure.USART_BaudRate = 9600;
  USART_InitStructure.USART_WordLength = USART_WordLength_8b;
  USART_InitStructure.USART_StopBits = USART_StopBits_1;
  USART_InitStructure.USART_Parity = USART_Parity_No;
  USART_InitStructure.USART_HardwareFlowControl = USART_HardwareFlowControl_None;
  USART_InitStructure.USART_Mode = USART_Mode_Rx | USART_Mode_Tx;
  USART_InitStructure.USART_Clock = USART_Clock_Disable;
  USART_InitStructure.USART_CPOL = USART_CPOL_Low;
  USART_InitStructure.USART_CPHA = USART_CPHA_2Edge;
  USART_InitStructure.USART_LastBit = USART_LastBit_Disable;

  USART_Init(USART2, &USART_InitStructure); // Configure the USART2
  // Enable the USART1 Transmit and Receive interrupt
  USART_ITConfig(USART2, USART_IT_TXE, DISABLE);
  // Enable the USART Receive interrupt
  USART_ITConfig(USART2, USART_IT_RXNE, ENABLE);
  USART_Cmd(USART2, ENABLE);  // Enable USART2
}
//-----------------------------------------------------------------------------------------------------//
//------------------------------ Function USART1,2 send string ---------------------------------------//
//-----------------------------------------------------------------------------------------------------//
void usart1_putc(unsigned char c)
{
  /*
  while(TxReady==0);    // Check TX ready flag
  TxCounter=0;   // Clear counter of data byte
  TxBuffer[0] = c;  // Load data to send(1 byte)
  TxBuffer[1] = '\0';  // End of string
  USART_ITConfig(USART1, USART_IT_TXE,ENABLE); // Enable TX interrupt start to send
  TxReady=0;  // Clear flag indicate(TX busy)
  */
  while(USART_GetFlagStatus(USART1,USART_FLAG_TXE)==RESET);   // Wait until transmition ready
  USART_SendData(USART1,(int)c);   // Send character
}
void usart2_putc(unsigned char c)
{
  while(USART_GetFlagStatus(USART2,USART_FLAG_TXE)==RESET);   // Wait until transmition ready
  USART_SendData(USART2,(int)c);   // Send character
}
//-----------------------------------------------------------------------------------------------------//
//------------------------------ Function USART1,2 send string ---------------------------------------//
//-----------------------------------------------------------------------------------------------------//
void usart1_puts(unsigned char * s)
{
  while(TxReady==0);   // Check TX ready flag
  TxCounter=0;      // Clear counter of data byte
  strcpy(TxBuffer,s);  // Load data to send
  USART_ITConfig(USART1,USART_IT_TXE,ENABLE);  // Enable TX interrupt start to send
  TxReady=0;  // Clear flag indicate(TX busy)
}
void usart2_puts(unsigned char *s)
{
  while(*s)   // Check end of string
  {
    usart2_putc(*s++);   // Send charracter 1 time
  }
```

```
}
//--------------------------------------------------------------------------------------------------//
//---------------------------------- Function USART1 ISR ------------------------------------------//
//--------------------------------------------------------------------------------------------------//
void USART1_IRQHandler(void)
{
  if(USART_GetITStatus(USART1, USART_IT_RXNE) != RESET)  // Check RX interrupt?
  {
    RxMessage = USART_ReceiveData(USART1); // Read one byte from the receive data register
    RxUpdate=1; // Flag indicate receive data

    USART_ClearITPendingBit(USART1, USART_IT_RXNE);  // Clear the USART1 Receive interrupt
  }
  if(USART_GetITStatus(USART1, USART_IT_TXE) != RESET)   // Check TX interrupt?
  {
    // Write one byte to the transmit data register
    USART_SendData(USART1,TxBuffer[TxCounter++]);
    // Clear the USART1 transmit interrupt
    USART_ClearITPendingBit(USART1, USART_IT_TXE);
    // Check End of transmission or over range of data buffer
    if(TxBuffer[TxCounter] == 0 || TxCounter>TxBufferSize)
    {
      // Disable the USART1 Transmit interrupt
      USART_ITConfig(USART1, USART_IT_TXE, DISABLE);
      TxReady=1; // Set flag for Tx ready
    }
  }
}
void USART2_IRQHandler(void)
{
  if(USART_GetITStatus(USART2, USART_IT_RXNE) != RESET)  // Check RX interrupt?
  {
    RxMessage2 = USART_ReceiveData(USART2); // Read one byte from the receive data register
    RxUpdate2=1; // Flag indicate receive data

    USART_ClearITPendingBit(USART2, USART_IT_RXNE);  // Clear the USART1 Receive interrupt
  }
  if(USART_GetITStatus(USART2, USART_IT_TXE) != RESET)   // Check TX interrupt?
  {
    // Write one byte to the transmit data register
    USART_SendData(USART2,TxBuffer2[TxCounter2++]);
    // Clear the USART1 transmit interrupt
    USART_ClearITPendingBit(USART2, USART_IT_TXE);
    // Check End of transmission or over range of data buffer
    if(TxBuffer2[TxCounter2] == 0 || TxCounter2>TxBufferSize2)
    {
      // Disable the USART1 Transmit interrupt
      USART_ITConfig(USART2, USART_IT_TXE, DISABLE);
      TxReady2=1; // Set flag for Tx ready
    }
  }
}
//--------------------------------------------------------------------------------------------------//
//------------------------------------------ Sound ------------------------------------------------//
//--------------------------------------------------------------------------------------------------//
void sound(int freq,long tm)
{
```

```
  int newPeriod;
  newPeriod = (u16)(1000000/freq)-1;  // Calculate new period value to set
   TIM_SetAutoreload(TIM4,newPeriod);  // Set new period value
   TIM_SetCompare3(TIM4,newPeriod/2);  // Set duty cycle maintain at 50%
   TIM_Cmd(TIM4, ENABLE); // TIM4 signal enable
   delay(tm);            // Delay for generate signal 1 sec
   TIM_Cmd(TIM4, DISABLE); // TIM4 signal disable
}
//-----------------------------------------------------------------------------------------------//
//------------------------------------- Main Program -------------------------------------------//
//-----------------------------------------------------------------------------------------------//
int main()
{
  //char str[14];     // Keep string convert
  RCC_setup();      // Reset/Clock Configuration
  GPIO_setup();     // GPIO Configuration
  ADC_setup();      // ADC Configuration
  USART1_setup();   // USART1 Configuration
  USART2_setup();   // USART2 Configuration
  NVIC_setup();     // NVIC configuration
  TIM4_setup();
  /*
  GPIO_WriteBit(GPIOC, GPIO_Pin_6,1); // LED at PC6 ON
  GPIO_WriteBit(GPIOE, GPIO_Pin_7,1); // TRIG ON
  delay(1500);
  GPIO_WriteBit(GPIOC, GPIO_Pin_6,0); // LED at PC6 OFF
  GPIO_WriteBit(GPIOE, GPIO_Pin_7,0); // TRIG OFF
*/
  lcdString(1,1,"XBee Gateway");          // Set message on line 1
        lcdString(1,2,"Started...");      // Set message on line 2
  lcdString(1,4,"USART1 : Ready");        // Set message on line 4
  lcdString(1,5,"USART2 : Ready");        // Set message on line 5
  lcdBackLight(1);
  lcdUpdate();  // Display message

  while(1)    // Infinite loop
  {
    lcdBackLight(1);

    if(RxUpdate==1)   // Check incoming data to dspic
    {
      RxUpdate = 0;                  // Clear flag for next time
      usart2_putc(RxMessage);         // Send character from receive
      //lcdString(1,4,"U1->U2: GET ");  // Set message on line 2
      //lcdUpdate();                  // Display message
      //sound(500,50);
    }
    if(RxUpdate2==1)
    {
      RxUpdate2 = 0;
      usart1_putc(RxMessage2);
      //lcdString(1,5,"U2->U1: GET ");  // Set message on line 2
      //lcdUpdate();                  // Display message
    }
    /*
    lcdString(1,4,"U1->U2: none");      // Set message on line 4
    lcdString(1,5,"U2->U1: none");      // Set message on line 5
```

```
    lcdUpdate();          // Display message
    */
  }
}
```

ภาคผนวก (ง)
ซอฟท์แวร์เฝ้าระวังและควบคุม ระบบ Zigbee
ในส่วนของ Process Code

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.Windows.Forms;
using System.IO.Ports;
using System.Configuration;
using System.Reflection;
using System.IO;
using MySql.Data.MySqlClient;
using MySql.Data.Types;

namespace XbeeSerialHex
{
  public partial class Form1 : Form
  {
    bool flagled01 = false, flagled02 = false;
    bool networkMode = false;
    bool statusCoordinator = false;  bool statusRouter1 = false, statusRouter2 = false, statusRouter3 = false;

    double router1levelSample1, router1VoltSample1, router1ConvertSample1;
    double router2levelSample1, router2VoltSample1, router2ConvertSample1;
    double router3levelSample1, router3VoltSample1, router3ConvertSample1;

    public Form1()
    {
      InitializeComponent();
      InitializeControlValues();
      // Enable Event handler
      serialPortxbee.DataReceived += new SerialDataReceivedEventHandler(serialPortxbee_DataReceived);
    }

    private SerialPort serialPortxbee = new SerialPort();
    private DateTime CurrTime = DateTime.Now;
    private string strProvider = "Data Source=localhost;Database=zigbee_analysis;User ID=root;Password=123456";

    private void InitializeControlValues()
    {
      foreach (string portname in SerialPort.GetPortNames())
      {
        serialCombox.Items.Add(portname);
        // If can no find any port in computer
        if (serialCombox.Items.Count <= 0)
        {
          serialCombox.Enabled = false;
        }
      }
      serialBaudbox.SelectedIndex = 7;
      //Disable Tab Index
      tabXbee.Enabled = false;
      button1.Enabled = false;
      refreshForceTimer.Enabled = false;
      //Maximum text Monitor Line

      //Config data monitoring textbox
      int numOfLines = 5;
      var lines = this.txtData.Lines;
```

```
            var newLines = lines.Skip(numOfLines);
            //Timer Refresh Network Default
            Configuration config = ConfigurationManager.OpenExeConfiguration(Application.ExecutablePath);
            var appConfig = ConfigurationManager.OpenExeConfiguration(Assembly.GetExecutingAssembly().Location);
            int timerRefreshValue = Convert.ToInt32(appConfig.AppSettings.Settings["timerRefreshValue"].Value);
            timeLabel.Text = timerRefreshValue.ToString();
        }

        private void serialCombox_SelectedIndexChanged_1(object sender, EventArgs e)
        {
            serialCombox.DropDownStyle = ComboBoxStyle.DropDownList;
            btnConnect.Enabled = true;
        }

        private void serialBaudbox_SelectedIndexChanged_1(object sender, EventArgs e)
        {
            serialBaudbox.DropDownStyle = ComboBoxStyle.DropDownList;
        }

        private void setModebt1_Click(object sender, EventArgs e)
        {
            networkMode = false;
            if (networkMode == false)
            {
                modeControlText.Text = "MANUAL MODE";
                modeControlText.ForeColor = Color.Red;
                //Enable all button
                btRouter1On1.Enabled = true;
                btRouter1Off1.Enabled = true;
                btRouter2On1.Enabled = true;
                btRouter2Off1.Enabled = true;
                //Disable Threshold tab
                thresholdRouter1.Text = "- - - -";
                thresholdRouter2.Text = "- - - -";
            }
            else if (networkMode == true)
            {
                modeControlText.Text = "AUTO MODE";
                modeControlText.ForeColor = Color.GreenYellow;
                //Disable all button
                btRouter1On1.Enabled = false;
                btRouter1Off1.Enabled = false;
                btRouter2On1.Enabled = false;
                btRouter2Off1.Enabled = false;
                //Threshold for Router1 Get Value
                var appConfig =
ConfigurationManager.OpenExeConfiguration(Assembly.GetExecutingAssembly().Location);
                decimal CThreshold1max =
Convert.ToDecimal(appConfig.AppSettings.Settings["CThreshold1max"].Value);
                thresholdRouter1.Text = CThreshold1max.ToString();
                //Threshold for Router2 Get Value
                decimal CThreshold2max =
Convert.ToDecimal(appConfig.AppSettings.Settings["CThreshold2max"].Value);
                thresholdRouter2.Text = CThreshold2max.ToString();
            }
        }

        private void setModebt2_Click(object sender, EventArgs e)
        {
            networkMode = true;
```

```
            if (networkMode == false)
            {
                modeControlText.Text = "MANUAL MODE";
                modeControlText.ForeColor = Color.Red;
                //Enable all button
                btRouter1On1.Enabled = true;
                btRouter1Off1.Enabled = true;
                btRouter2On1.Enabled = true;
                btRouter2Off1.Enabled = true;
                //Disable Threshold tab
                thresholdRouter1.Text = "- - - -";
                thresholdRouter2.Text = "- - - -";
            }
            else if (networkMode == true)
            {
                modeControlText.Text = "AUTO MODE";
                modeControlText.ForeColor = Color.GreenYellow;
                //Disable all button
                btRouter1On1.Enabled = false;
                btRouter1Off1.Enabled = false;
                btRouter2On1.Enabled = false;
                btRouter2Off1.Enabled = false;
                //Threshold for Router1 Get Value
                var appConfig =
ConfigurationManager.OpenExeConfiguration(Assembly.GetExecutingAssembly().Location);
                decimal CThreshold1max =
Convert.ToDecimal(appConfig.AppSettings.Settings["CThreshold1max"].Value);
                thresholdRouter1.Text = CThreshold1max.ToString();
                //Threshold for Router2 Get Value
                decimal CThreshold2max =
Convert.ToDecimal(appConfig.AppSettings.Settings["CThreshold2max"].Value);
                thresholdRouter2.Text = CThreshold2max.ToString();
            }
        }

        private void btnConnect_Click(object sender, EventArgs e)
        {
            if (serialPortxbee.IsOpen)
            {
                serialPortxbee.Close();
                serialCombox.Enabled = true;
                serialBaudbox.Enabled = true;
                btnConnect.Text = "Connect";
                btnRefresh.Enabled = false;
                tabXbee.Enabled = false;
                button1.Enabled = false;
                refreshForceTimer.Enabled = false;
                //Reset Label
                statusCoordinator = false;
                statusRouter1 = false;
                statusRouter2 = false;
                statusRouter3 = false;
                router1in1.Text = "N/A";
                router2in1.Text = "N/A";
                router3in1.Text = "N/A";
                router1out1.Text = "N/A";
                router2out1.Text = "N/A";
                //Reset Tab1
                rawRouter1.Text = "- - -";
                analogRouter1.Text = "- - -";
```

```
            tempRouter1.Text = "- - -";
            outputRouter1.Text = "- - -";
            timeConfRouter1.Text = "- - -";
            //Reset Tab2
            rawRouter2.Text = "- - -";
            analogRouter2.Text = "- - -";
            tempRouter2.Text = "- - -";
            outputRouter2.Text = "- - -";
            timeConfRouter2.Text = "- - -";
            //Reset Tab3
            rawRouter3.Text = "- - -";
            analogRouter3.Text = "- - -";
            tempRouter3.Text = "- - -";
            timeConfRouter3.Text = "- - -";
            this.Invoke(new EventHandler(DetectUpdate));
            tmrRefresh.Stop();
            lblClockText.Text = "- - - - - - - - -";
            lblClockText2.Text = "-- : -- : --";
            lblClockText.ForeColor = Color.Silver;
            lblClockText2.ForeColor = Color.Silver;
            //Load Default Parameter Timer
            onlineTimeCheck.Enabled = false;  //Stop timer online label check
            Configuration config = ConfigurationManager.OpenExeConfiguration(Application.ExecutablePath);
            var appConfig =
ConfigurationManager.OpenExeConfiguration(Assembly.GetExecutingAssembly().Location);
            int timerRefreshValue = Convert.ToInt32(appConfig.AppSettings.Settings["timerRefreshValue"].Value);
            iCount = timerRefreshValue;
            timeLabel.Text = iCount.ToString();
            int timerForceValue = Convert.ToInt32(appConfig.AppSettings.Settings["timerForceValue"].Value);
            iCount2 = timerForceValue;
            timeForceLabel.Text = iCount2.ToString();
        }
        else
        {
            serialPortxbee.PortName = serialCombox.Text;
            serialPortxbee.BaudRate = int.Parse(serialBaudbox.Text);
            serialPortxbee.DataBits = 8;
            serialPortxbee.StopBits = StopBits.One;
            serialPortxbee.Parity = Parity.None;
            // Then open comport
            serialPortxbee.Open();
            if (serialPortxbee.IsOpen)
            {
                //ZIGBEE NODE IDENTIFIER
                serialPortxbee.Write(new byte[] { 0x7E, 0x00, 0x04, 0x08, 0x01, 0x4E, 0x44, 0x64 }, 0, 8);
                txtData.Text = txtData.Text + "(" + DateTime.Now + ") -> " + "API : 08   Search Coordinator... Forming
Network..." + "\r\n";
                    ///// WRITE FILE LOG /////
                    string fileNameTemp = "logger-" + DateTime.Now.Date.ToString("yyyy-M");
                    if (File.Exists("log/" + fileNameTemp + ".txt"))
                    {
                        TextReader tr = new StreamReader("log/" + fileNameTemp + ".txt");
                        string contentReader = tr.ReadToEnd();
                        tr.Close();
                        TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                        tw.Write(contentReader + "\r\n" + "(" + DateTime.Now + ") -> " + "API : 08   Search
Coordinator... Forming Network...");
                        tw.Close();
                    }
                    else
```

```
                {
                    TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                    tw.Write("(" + DateTime.Now + ") -> " + "API : 08   Search Coordinator... Forming Network...");
                    tw.Close();
                }
                ///// END WRITE FILE TO LOG /////
            //ALWAYS BOTTOM
            this.txtData.SelectionStart = txtData.Text.Length;
            this.txtData.ScrollToCaret();
            serialCombox.Enabled = false;
            serialBaudbox.Enabled = false;
            btnRefresh.Enabled = true;
            btnConnect.Enabled = true;
            tabXbee.Enabled = true;
            button1.Enabled = true;
            refreshForceTimer.Enabled = true;
            btnConnect.Text = "Disconnect";

            // Load the current application's configuration file
            Configuration config = ConfigurationManager.OpenExeConfiguration(Application.ExecutablePath);
            var appConfig =
ConfigurationManager.OpenExeConfiguration(Assembly.GetExecutingAssembly().Location);
            string timerRefreshNetwork =
Convert.ToString(appConfig.AppSettings.Settings["timerRefreshNetwork"].Value);
            int timerRefreshValue = Convert.ToInt32(appConfig.AppSettings.Settings["timerRefreshValue"].Value);
            iCount = timerRefreshValue;
            timeLabel.Text = iCount.ToString();
            if (timerRefreshNetwork == "on")
            {
                onlineTimeCheck.Enabled = true;  //Start timer online label check
            }
            if (timerRefreshNetwork == "off")
            {
                onlineTimeCheck.Enabled = false;  //Stop timer online label check
            }
            //**
            string timerForceSample =
Convert.ToString(appConfig.AppSettings.Settings["timerForceSample"].Value);
            int timerForceValue = Convert.ToInt32(appConfig.AppSettings.Settings["timerForceValue"].Value);
            iCount2 = timerForceValue;
            timeForceLabel.Text = iCount2.ToString();
            if (timerForceSample == "on")
            {
                onlineForceCheck.Enabled = true;  //Start timer online label check
            }
            if (timerForceSample == "off")
            {
                onlineForceCheck.Enabled = false;  //Stop timer online label check
            }
            //**
            tmrRefresh.Start(); //Enable Real time Clock
        }
    }
}


private void refreshTimer_Click(object sender, EventArgs e)
{
    // Load the current application's configuration file
    Configuration config = ConfigurationManager.OpenExeConfiguration(Application.ExecutablePath);
```

```
            var appConfig = ConfigurationManager.OpenExeConfiguration(Assembly.GetExecutingAssembly().Location);
            string timerRefreshNetwork =
    Convert.ToString(appConfig.AppSettings.Settings["timerRefreshNetwork"].Value);
            int timerRefreshValue = Convert.ToInt32(appConfig.AppSettings.Settings["timerRefreshValue"].Value);
            iCount = timerRefreshValue;
            timeLabel.Text = iCount.ToString();
            if (timerRefreshNetwork == "on")
            {
                onlineTimeCheck.Enabled = true; //Start timer online label check
            }
            if (timerRefreshNetwork == "off")
            {
                onlineTimeCheck.Enabled = false; //Stop timer online label check
            }
        }

        private void btnRefresh_Click(object sender, EventArgs e)
        {
            statusCoordinator = false;
            statusRouter1 = false;
            statusRouter2 = false;
            statusRouter3 = false;
            timeLabelCoordinator.Text = "Waiting...";
            timeLabelRouter1.Text = "Waiting...";
            timeLabelRouter2.Text = "Waiting...";
            timeLabelRouter3.Text = "Waiting...";
            timeConfRouter1.Text = "Waiting...";
            timeConfRouter2.Text = "Waiting...";
            timeConfRouter3.Text = "Waiting...";
            //ZIGBEE NODE IDENTIFIER
            serialPortxbee.Write(new byte[] { 0x7E, 0x00, 0x04, 0x08, 0x01, 0x4E, 0x44, 0x64 }, 0, 8);
            txtData.Text = txtData.Text + "(" + DateTime.Now + ") -> " + "API : 08   Refreshing Network... Identify Node
    Xbee" + "\r\n";
                ///// WRITE FILE LOG /////
                string fileNameTemp = "logger-" + DateTime.Now.Date.ToString("yyyy-M");
                if (File.Exists("log/" + fileNameTemp + ".txt"))
                {
                    TextReader tr = new StreamReader("log/" + fileNameTemp + ".txt");
                    string contentReader = tr.ReadToEnd();
                    tr.Close();
                    TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                    tw.Write(contentReader + "\r\n" + "(" + DateTime.Now + ") -> " + "API : 08   Refreshing Network...
    Identify Node Xbee");
                    tw.Close();
                }
                else
                {
                    TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                    tw.Write("(" + DateTime.Now + ") -> " + "API : 08   Refreshing Network... Identify Node Xbee");
                    tw.Close();
                }
                ///// END WRITE FILE TO LOG /////
            this.Invoke(new EventHandler(DetectUpdate));
            //ALWAYS BOTTOM
            this.txtData.SelectionStart = txtData.Text.Length;
            this.txtData.ScrollToCaret();
        }

        private void btRouter1On1_Click(object sender, EventArgs e)
        {
```

```
        router1out1.Text = "ON";
        flagled01 = true;
        serialPortxbee.Write(new byte[] { 0x7E, 0x00, 0x10, 0x17, 0x05, 0x00, 0x13, 0xA2, 0x00, 0x40, 0x55, 0xB1,
0xCC, 0xFF, 0xFE, 0x02, 0x44, 0x31, 0x05, 0xA3 }, 0, 20);
        txtData.Text = txtData.Text + "(" + DateTime.Now + ") -> " + "API : 17   Manual Mode, Turn on DIO Channel
1 on Router1" + "\r\n";
            ///// WRITE FILE LOG /////
            string fileNameTemp = "logger-" + DateTime.Now.Date.ToString("yyyy-M");
            if (File.Exists("log/" + fileNameTemp + ".txt"))
            {
                TextReader tr = new StreamReader("log/" + fileNameTemp + ".txt");
                string contentReader = tr.ReadToEnd();
                tr.Close();
                TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                tw.Write(contentReader + "\r\n" + "(" + DateTime.Now + ") -> " + "API : 17   Manual Mode, Turn on
DIO Channel 1 on Router1");
                tw.Close();
            }
            else
            {
                TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                tw.Write("(" + DateTime.Now + ") -> " + "API : 17   Manual Mode, Turn on DIO Channel 1 on
Router1");
                tw.Close();
            }
            ///// END WRITE FILE TO LOG /////
    }

    private void btRouter1Off1_Click(object sender, EventArgs e)
    {
        router1out1.Text = "OFF";
        flagled01 = false;
        serialPortxbee.Write(new byte[] { 0x7E, 0x00, 0x10, 0x17, 0x05, 0x00, 0x13, 0xA2, 0x00, 0x40, 0x55, 0xB1,
0xCC, 0xFF, 0xFE, 0x02, 0x44, 0x31, 0x04, 0xA4 }, 0, 20);
        txtData.Text = txtData.Text + "(" + DateTime.Now + ") -> " + "API : 17   Manual Mode, Turn off DIO Channel
1 on Router1" + "\r\n";
            ///// WRITE FILE LOG /////
            string fileNameTemp = "logger-" + DateTime.Now.Date.ToString("yyyy-M");
            if (File.Exists("log/" + fileNameTemp + ".txt"))
            {
                TextReader tr = new StreamReader("log/" + fileNameTemp + ".txt");
                string contentReader = tr.ReadToEnd();
                tr.Close();
                TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                tw.Write(contentReader + "\r\n" + "(" + DateTime.Now + ") -> " + "API : 17   Manual Mode, Turn off
DIO Channel 1 on Router1");
                tw.Close();
            }
            else
            {
                TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                tw.Write("(" + DateTime.Now + ") -> " + "API : 17   Manual Mode, Turn off DIO Channel 1 on
Router1");
                tw.Close();
            }
            ///// END WRITE FILE TO LOG /////
    }

    private void btRouter2On1_Click(object sender, EventArgs e)
    {
```

```
            router2out1.Text = "ON";
            flagled02 = true;
            serialPortxbee.Write(new byte[] { 0x7E, 0x00, 0x10, 0x17, 0x05, 0x00, 0x13, 0xA2, 0x00, 0x40, 0x3D, 0xC7,
0x7D, 0xFF, 0xFE, 0x02, 0x44, 0x31, 0x05, 0xF4 }, 0, 20);
            txtData.Text = txtData.Text + "(" + DateTime.Now + ") -> " + "API : 17   Manual Mode, Turn on DIO Channel
1 on Router2" + "\r\n";
                ///// WRITE FILE LOG /////
                string fileNameTemp = "logger-" + DateTime.Now.Date.ToString("yyyy-M");
                if (File.Exists("log/" + fileNameTemp + ".txt"))
                {
                  TextReader tr = new StreamReader("log/" + fileNameTemp + ".txt");
                  string contentReader = tr.ReadToEnd();
                  tr.Close();
                  TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                  tw.Write(contentReader + "\r\n" + "(" + DateTime.Now + ") -> " + "API : 17   Manual Mode, Turn on
DIO Channel 1 on Router2");
                  tw.Close();
                }
                else
                {
                  TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                  tw.Write("(" + DateTime.Now + ") -> " + "API : 17   Manual Mode, Turn on DIO Channel 1 on
Router2");
                  tw.Close();
                }
                ///// END WRITE FILE TO LOG /////
        }

        private void btRouter2Off1_Click(object sender, EventArgs e)
        {
            router2out1.Text = "OFF";
            flagled02 = false;
            serialPortxbee.Write(new byte[] { 0x7E, 0x00, 0x10, 0x17, 0x05, 0x00, 0x13, 0xA2, 0x00, 0x40, 0x3D, 0xC7,
0x7D, 0xFF, 0xFE, 0x02, 0x44, 0x31, 0x04, 0xF5 }, 0, 20);
            txtData.Text = txtData.Text + "(" + DateTime.Now + ") -> " + "API : 17   Manual Mode, Turn off DIO Channel
1 o Router2" + "\r\n";
                ///// WRITE FILE LOG /////
                string fileNameTemp = "logger-" + DateTime.Now.Date.ToString("yyyy-M");
                if (File.Exists("log/" + fileNameTemp + ".txt"))
                {
                  TextReader tr = new StreamReader("log/" + fileNameTemp + ".txt");
                  string contentReader = tr.ReadToEnd();
                  tr.Close();
                  TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                  tw.Write(contentReader + "\r\n" + "(" + DateTime.Now + ") -> " + "API : 17   Manual Mode, Turn off
DIO Channel 1 on Router2");
                  tw.Close();
                }
                else
                {
                  TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                  tw.Write("(" + DateTime.Now + ") -> " + "API : 17   Manual Mode, Turn off DIO Channel 1 on
Router2");
                  tw.Close();
                }
                ///// END WRITE FILE TO LOG /////
        }

        private void serialPortxbee_DataReceived(object sender, SerialDataReceivedEventArgs e)
        {
```

```
            this.Invoke(new EventHandler(DoUpdate));
            this.Invoke(new EventHandler(DetectUpdate));
            this.Invoke(new EventHandler(VariableUpdate));
        }

        private void DoUpdate(object s, EventArgs e)
        {
         byte[] MyPacket;
         int[] analogSamples;
         MyPacket = new byte[64];
         analogSamples = new int[4];

         byte StartDiameter = (byte)serialPortxbee.ReadChar();
         if(StartDiameter == 0x7E )
         {
            statusCoordinator = true;
            timeLabelCoordinator.Text = DateTime.Now.ToString(); // Update Recent Coordinator
            //START DIAMETER
            MyPacket[0] = StartDiameter;
            //DATA LENGTH
            MyPacket[1] = (byte)serialPortxbee.ReadByte();
            MyPacket[2] = (byte)serialPortxbee.ReadByte();
            int dataLength = (MyPacket[1] << 8 | MyPacket[2]);
            //PRELOAD
            for (int i=1; i <= dataLength ; i++)
            {
               MyPacket[2 + i] = (byte)serialPortxbee.ReadByte();
            }
            //NAME API ID = OFFSET 3
            byte apiID = MyPacket[3];
            int macIDHigh = (((((MyPacket[4] << 8) | MyPacket[5]) << 8 ) | MyPacket[6]) << 8 ) | MyPacket[7] ;
            int macIDLow = (((((MyPacket[8] << 8) | MyPacket[9]) << 8) | MyPacket[10]) << 8) | MyPacket[11];
            //CHECKSUM
            MyPacket[3 + dataLength] = (byte)serialPortxbee.ReadByte();
            //SUM ALL PACKET (datalength + 1byte start diameter + 2 byte length + 1 checksum)
            int allMyPacket = dataLength + 4;
            int MyChecksum = MyPacket[3 + dataLength];
            int CalCheck = 0;
            for (int k=1 ; k <= dataLength ; k++ ) {
              CalCheck = CalCheck + MyPacket[2+k];
            }
            //TAKE SUM ALL TO BYTE LOW ONLY
            CalCheck = CalCheck & 0x00FF;
            //CAL 0xFF - SUM ALL BYTE LOW
            CalCheck = 0xFF - CalCheck;

            //CHECK ORIGINAL CHECKSUM = CAL CHECKSUM?
            if (MyChecksum == CalCheck)
            {
              if (apiID == 0x8A)                        // *** Hardware Reset
              {
                 if (MyPacket[4] == 0x00)
                 {
                   txtData.Text = txtData.Text + "(" + DateTime.Now + ") -> " + "API : 8A   Hardware Coordinator
Reset" + "\r\n";
                     ///// WRITE FILE LOG /////
                     string fileNameTemp = "logger-" + DateTime.Now.Date.ToString("yyyy-M");
                     if (File.Exists("log/" + fileNameTemp + ".txt"))
                     {
                        TextReader tr = new StreamReader("log/" + fileNameTemp + ".txt");
```

```
                    string contentReader = tr.ReadToEnd();
                    tr.Close();
                    TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                    tw.Write(contentReader + "\r\n" + "(" + DateTime.Now + ") -> " + "API : 8A   Hardware
Coordinator Reset");
                    tw.Close();
                  }
                  else
                  {
                    TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                    tw.Write("(" + DateTime.Now + ") -> " + "API : 8A   Hardware Coordinator Reset");
                    tw.Close();
                  }
                  ///// END WRITE FILE TO LOG /////
              //Reset Label
              statusCoordinator = false; statusRouter1 = false; statusRouter2 = false; statusRouter3 = false;
              this.Invoke(new EventHandler(DetectUpdate));
            }
            if (MyPacket[4] == 0x06)
            {
              txtData.Text = txtData.Text + "(" + DateTime.Now + ") -> " + "API : 8A   Coordinator Started...
Forming Network..." + "\r\n";
                  ///// WRITE FILE LOG /////
                  string fileNameTemp = "logger-" + DateTime.Now.Date.ToString("yyyy-M");
                  if (File.Exists("log/" + fileNameTemp + ".txt"))
                  {
                    TextReader tr = new StreamReader("log/" + fileNameTemp + ".txt");
                    string contentReader = tr.ReadToEnd();
                    tr.Close();
                    TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                    tw.Write(contentReader + "\r\n" + "(" + DateTime.Now + ") -> " + "API : 8A   Coordinator
Started... Forming Network...");
                    tw.Close();
                  }
                  else
                  {
                    TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                    tw.Write("(" + DateTime.Now + ") -> " + "API : 8A   Coordinator Started... Forming
Network...");
                    tw.Close();
                  }
                  ///// END WRITE FILE TO LOG /////
              statusCoordinator = true;
            }
          }
          //CHECK MODE SELECT API ID
          if (apiID == 0x88)                        // *** AT Command Response
          {
            int cmdName = (MyPacket[5] << 8) | MyPacket[6];
            int cmdNamePreConv1 = MyPacket[5];
            int cmdNamePreConv2 = MyPacket[6];
            //CONVERT INT TO CHAR TO SHOW COMMAND
            char cmdNameChar1 = (char) cmdNamePreConv1;
            char cmdNameChar2 = (char)cmdNamePreConv2;
            int macIDHighATCommand = (((((MyPacket[10] << 8) | MyPacket[11]) << 8) | MyPacket[12]) << 8) |
MyPacket[13];
            int macIDLowATCommand = (((((MyPacket[14] << 8) | MyPacket[15]) << 8) | MyPacket[16]) << 8) |
MyPacket[17];
            if (cmdName == 0x4E44)              // *** --- 4E 44 = ATND Polling Name in Network
            {
```

```
                    txtData.Text = txtData.Text + "(" + DateTime.Now + ") -> " + "API : " + apiID.ToString("X2") + "   System Found : " + "AT" + cmdNameChar1 + cmdNameChar2 + "   MAC : " + macIDHighATCommand.ToString("X2") + " " + macIDLowATCommand.ToString("X2") + " joined to network...\r\n";
                        ///// WRITE FILE LOG /////
                        string fileNameTemp = "logger-" + DateTime.Now.Date.ToString("yyyy-M");
                        if (File.Exists("log/" + fileNameTemp + ".txt"))
                        {
                            TextReader tr = new StreamReader("log/" + fileNameTemp + ".txt");
                            string contentReader = tr.ReadToEnd();
                            tr.Close();
                            TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                            tw.Write(contentReader + "\r\n" + "(" + DateTime.Now + ") -> " + "API : " + apiID.ToString("X2") + "   System Found : " + "AT" + cmdNameChar1 + cmdNameChar2 + "   MAC : " + macIDHighATCommand.ToString("X2") + " " + macIDLowATCommand.ToString("X2") + " joined to network...");
                            tw.Close();
                        }
                        else
                        {
                            TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                            tw.Write("(" + DateTime.Now + ") -> " + "API : " + apiID.ToString("X2") + "   System Found : " + "AT" + cmdNameChar1 + cmdNameChar2 + "   MAC : " + macIDHighATCommand.ToString("X2") + " " + macIDLowATCommand.ToString("X2") + " joined to network...");
                            tw.Close();
                        }
                        ///// END WRITE FILE TO LOG /////
                }
                //Do Update label
                if (macIDHighATCommand == 0x0013A200 && macIDLowATCommand == 0x4055B1CC)
                {
                    statusRouter1 = true;
                    timeLabelRouter1.Text = DateTime.Now.ToString();
                    timeConfRouter1.Text = DateTime.Now.ToString();
                }
                else if (macIDHighATCommand == 0x0013A200 && macIDLowATCommand == 0x403DC77D)
                {
                    statusRouter2 = true;
                    timeLabelRouter2.Text = DateTime.Now.ToString();
                    timeConfRouter2.Text = DateTime.Now.ToString();
                }
                else if (macIDHighATCommand == 0x0013A200 && macIDLowATCommand == 0x403DC71D)
                {
                    statusRouter3 = true;
                    timeLabelRouter3.Text = DateTime.Now.ToString();
                    timeConfRouter3.Text = DateTime.Now.ToString();
                }
            }
            if (apiID == 0x92)                          // *** Zigbee IO Data Sample Rx Indicator
            {
                int analogSampleindex = 19;
                int digitalChannelMask = (MyPacket[16] << 8) | MyPacket[17];
                // IF DIGITAL CHANNEL MASK MORE THAN ZERO
                //if (digitalChannelMask > 0)
               // {
                    int digitalSamples = (MyPacket[19] << 8) | MyPacket[20];
                    /*
                    for (int j = 0; j < 4; j++)              // 4 = number digital samples
                    {
                        digitalSamples[j] = ((d >> j) & 1);
                    }
```

```
        */
        analogSampleindex = 21;
      // }
      // TAKE ANALOG CHANNEL MASK
      int analogChannelMask = MyPacket[18];
      for (int m = 0; m < 4; m++)
      {
        if ( ((analogChannelMask >> m) & 1) != 0)
        {
          analogSamples[m] = (MyPacket[analogSampleindex] << 8) | MyPacket[analogSampleindex + 1];
          analogSampleindex += 2;
        }
        else
        {
          analogSamples[m] = -1;
        }
      }
      int readAnalogSample1 = analogSamples[0];

      //DISPLAY ON MONITOR SCREEN
      txtData.Text = txtData.Text + "(" + DateTime.Now + ") -> " + "API : " + apiID.ToString("X2") + "
 MAC :" + macIDHigh.ToString("X2") + " " + macIDLow.ToString("X2");
          ///// WRITE FILE LOG /////
          string fileNameTemp = "logger-" + DateTime.Now.Date.ToString("yyyy-M");
          if (File.Exists("log/" + fileNameTemp + ".txt"))
          {
            TextReader tr = new StreamReader("log/" + fileNameTemp + ".txt");
            string contentReader = tr.ReadToEnd();
            tr.Close();
            TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
            tw.Write(contentReader + "\r\n" + "(" + DateTime.Now + ") -> " + "API : " +
apiID.ToString("X2") + "   MAC :" + macIDHigh.ToString("X2") + " " + macIDLow.ToString("X2"));
            tw.Close();
          }
          else
          {
            TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
            tw.Write("(" + DateTime.Now + ") -> " + "API : " + apiID.ToString("X2") + "   MAC :" +
macIDHigh.ToString("X2") + " " + macIDLow.ToString("X2"));
            tw.Close();
          }
          ///// END WRITE FILE TO LOG /////
      if (digitalSamples == 0x0002)
      {
        txtData.Text = txtData.Text + "   O/P : on";
          ///// WRITE FILE LOG /////
          if (File.Exists("log/" + fileNameTemp + ".txt"))
          {
            TextReader tr = new StreamReader("log/" + fileNameTemp + ".txt");
            string contentReader = tr.ReadToEnd();
            tr.Close();
            TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
            tw.Write(contentReader + "   O/P : on ");
            tw.Close();
          }
          ///// END WRITE FILE TO LOG /////
          //ToMySQL
          string insertSQL = "INSERT INTO zb_datatabledigital (zb2_macHigh, zb2_macLow,
zb2_upDate, zb2_upTime, zb2_data_io1) VALUES (' " + macIDHigh.ToString("X2") + " ',' " +
```

```
macIDLow.ToString("X2") + " ',' " + DateTime.Now.ToString("yyyy-MM-dd") + " ',' " +
DateTime.Now.ToString("HH:mm:ss") + " ',' " + "on" + " ')";
                    MySqlConnection objMyCon = new MySqlConnection(strProvider);
                    MySqlCommand cmd = new MySqlCommand(insertSQL, objMyCon);
                    objMyCon.Open();
                    int executed = 0;
                    executed = cmd.ExecuteNonQuery();
                    objMyCon.Close();
                    //End ToMySQL

                if(macIDHigh == 0x0013A200 && macIDLow == 0x4055B1CC)
                {
                    flagled01 = true;
                    router1out1.Text = "ON";
                }
                if (macIDHigh == 0x0013A200 && macIDLow == 0x403DC77D)
                {
                    flagled02 = true;
                    router2out1.Text = "ON";
                }
                if (macIDHigh == 0x0013A200 && macIDLow == 0x403DC71D)
                {
                }
            }
            else if (digitalSamples == 0x0000)
            {
                txtData.Text = txtData.Text + "    O/P : off";
                    ///// WRITE FILE LOG /////
                    if (File.Exists("log/" + fileNameTemp + ".txt"))
                    {
                        TextReader tr = new StreamReader("log/" + fileNameTemp + ".txt");
                        string contentReader = tr.ReadToEnd();
                        tr.Close();
                        TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                        tw.Write(contentReader + "    O/P : off");
                        tw.Close();
                    }
                    ///// END WRITE FILE TO LOG /////
                    //ToMySQL
                    string insertSQL = "INSERT INTO zb_datatabledigital (zb2_macHigh, zb2_macLow,
zb2_upDate, zb2_upTime, zb2_data_io1) VALUES (' " + macIDHigh.ToString("X2") + " ',' " +
macIDLow.ToString("X2") + " ',' " + DateTime.Now.ToString("yyyy-MM-dd") + " ',' " +
DateTime.Now.ToString("HH:mm:ss") + " ',' " + "off" + " ')";
                    MySqlConnection objMyCon = new MySqlConnection(strProvider);
                    MySqlCommand cmd = new MySqlCommand(insertSQL, objMyCon);
                    objMyCon.Open();
                    int executed = 0;
                    executed = cmd.ExecuteNonQuery();
                    objMyCon.Close();
                    //End ToMySQL

                if (macIDHigh == 0x0013A200 && macIDLow == 0x4055B1CC)
                {
                    flagled01 = false;
                    router1out1.Text = "OFF";
                }
                if (macIDHigh == 0x0013A200 && macIDLow == 0x403DC77D)
                {
                    flagled02 = false;
                    router2out1.Text = "OFF";
```

```
            }
            if (macIDHigh == 0x0013A200 && macIDLow == 0x403DC71D)
            {
            }
        }
        txtData.Text = txtData.Text  + "   Analog : " + readAnalogSample1.ToString("X2") + "\r\n";
            ///// WRITE FILE LOG /////
            if (File.Exists("log/" + fileNameTemp + ".txt"))
            {
               TextReader tr = new StreamReader("log/" + fileNameTemp + ".txt");
               string contentReader = tr.ReadToEnd();
               tr.Close();
               TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
               tw.Write(contentReader + "   Analog : " + readAnalogSample1.ToString("X2"));
               tw.Close();
            }
            else
            {
               TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
               tw.Write("   Analog : " + readAnalogSample1.ToString("X2"));
               tw.Close();
            }
            ///// END WRITE FILE TO LOG /////
            // Again Update I/P and O/P Status
            if (macIDHigh == 0x0013A200 && macIDLow == 0x4055B1CC)
            {
               statusRouter1 = true;
               timeLabelRouter1.Text = DateTime.Now.ToString();
               timeConfRouter1.Text = DateTime.Now.ToString();
               router1levelSample1 = readAnalogSample1;
               router1VoltSample1 = (router1levelSample1 * 1200) / 1024;
               router1ConvertSample1 = (router1VoltSample1 / 10) ;
               //ToMySQL
               string insertSQL = "INSERT INTO zb_datatableanalog (zb_macHigh, zb_macLow, zb_upDate,
zb_upTime, zb_raw_io1, zb_data_io1) VALUES ( ' " + macIDHigh.ToString("X2") + " ',' " +
macIDLow.ToString("X2") + " ',' " + DateTime.Now.ToString("yyyy-MM-dd") + " ',' " +
DateTime.Now.ToString("HH:mm:ss") + " ',' " + readAnalogSample1.ToString("X2") + " ',' " +
router1ConvertSample1.ToString() + " ' )";
               MySqlConnection objMyCon = new MySqlConnection(strProvider);
               MySqlCommand cmd = new MySqlCommand(insertSQL, objMyCon);
               objMyCon.Open();
               int executedx = 0;
               executedx = cmd.ExecuteNonQuery();
               objMyCon.Close();
               //End ToMySQL
               //Case Auto Mode
               if (networkMode == true)
               {
                  var appConfig =
ConfigurationManager.OpenExeConfiguration(Assembly.GetExecutingAssembly().Location);
                  double measureCThreshold1max =
Convert.ToDouble(appConfig.AppSettings.Settings["CThreshold1max"].Value);
                  //Calculate Threshold
                  if (router1ConvertSample1 > measureCThreshold1max)
                  {
                     router1out1.Text = "ON";
                     flagled01 = true;
                     serialPortxbee.Write(new byte[] { 0x7E, 0x00, 0x10, 0x17, 0x05, 0x00, 0x13, 0xA2, 0x00,
0x40, 0x55, 0xB1, 0xCC, 0xFF, 0xFE, 0x02, 0x44, 0x31, 0x05, 0xA3 }, 0, 20);
```

```
                    txtData.Text = txtData.Text + "(" + DateTime.Now + ") -> " + "API : 17  Auto Mode, Turn
on DIO Channel 1 on Router1" + "\r\n";
                    ///// WRITE FILE LOG /////
                    if (File.Exists("log/" + fileNameTemp + ".txt"))
                    {
                      TextReader tr = new StreamReader("log/" + fileNameTemp + ".txt");
                      string contentReader = tr.ReadToEnd();
                      tr.Close();
                      TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                      tw.Write(contentReader + "\r\n" + "(" + DateTime.Now + ") -> " + "API : 17   Auto Mode,
Turn on DIO Channel 1 on Router1");
                      tw.Close();
                    }
                    else
                    {
                      TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                      tw.Write("(" + DateTime.Now + ") -> " + "API : 17   Auto Mode, Turn on DIO Channel 1
on Router1");

                      tw.Close();
                    }
                    ///// END WRITE FILE TO LOG /////
                  }
                  else if (router1ConvertSample1 <= measureCThreshold1max)
                  {
                    router1out1.Text = "OFF";
                    flagled01 = false;
                    serialPortxbee.Write(new byte[] { 0x7E, 0x00, 0x10, 0x17, 0x05, 0x00, 0x13, 0xA2, 0x00,
0x40, 0x55, 0xB1, 0xCC, 0xFF, 0xFE, 0x02, 0x44, 0x31, 0x04, 0xA4 }, 0, 20);
                    txtData.Text = txtData.Text + "(" + DateTime.Now + ") -> " + "API : 17   Auto Mode, Turn
off DIO Channel 1 on Router1" + "\r\n";
                    ///// WRITE FILE LOG /////
                    if (File.Exists("log/" + fileNameTemp + ".txt"))
                    {
                      TextReader tr = new StreamReader("log/" + fileNameTemp + ".txt");
                      string contentReader = tr.ReadToEnd();
                      tr.Close();
                      TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                      tw.Write(contentReader + "\r\n" + "(" + DateTime.Now + ") -> " + "API : 17   Manual
Mode, Turn off DIO Channel 1 on Router1");
                      tw.Close();
                    }
                    else
                    {
                      TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                      tw.Write("(" + DateTime.Now + ") -> " + "API : 17   Auto Mode, Turn off DIO Channel 1
on Router1");

                      tw.Close();
                    }
                    ///// END WRITE FILE TO LOG /////
                  }
                }
              }
              if (macIDHigh == 0x0013A200 && macIDLow == 0x403DC77D)
              {
                statusRouter2 = true;
                timeLabelRouter2.Text = DateTime.Now.ToString();
                timeConfRouter2.Text = DateTime.Now.ToString();
                router2levelSample1 = readAnalogSample1;
                router2VoltSample1 = (router2levelSample1 * 1.2) / 1024;
                router2VoltSample1 = router2VoltSample1 * 3;
```

```
router2ConvertSample1 = (34.112 * router2VoltSample1) - 20.388;
//ToMySQL
string insertSQL = "INSERT INTO zb_datatableanalog (zb_macHigh, zb_macLow, zb_upDate,
zb_upTime, zb_raw_io1, zb_data_io1) VALUES (' " + macIDHigh.ToString("X2") + " ',' " +
macIDLow.ToString("X2") + " ',' " + DateTime.Now.ToString("yyyy-MM-dd") + " ',' " +
DateTime.Now.ToString("HH:mm:ss") + " ',' " + readAnalogSample1.ToString("X2") + " ',' " +
router2ConvertSample1.ToString() + " ')";
MySqlConnection objMyCon = new MySqlConnection(strProvider);
MySqlCommand cmd = new MySqlCommand(insertSQL, objMyCon);
objMyCon.Open();
int executed = 0;
executed = cmd.ExecuteNonQuery();
objMyCon.Close();
//End ToMySQL
//Case Auto Mode
if (networkMode == true)
{
  var appConfig =
ConfigurationManager.OpenExeConfiguration(Assembly.GetExecutingAssembly().Location);
  double measureCThreshold2max =
Convert.ToDouble(appConfig.AppSettings.Settings["CThreshold2max"].Value);
  //Calculate Threshold
  if (router2ConvertSample1 <= measureCThreshold2max)
  {
    router2out1.Text = "ON";
    flagled02 = true;
    serialPortxbee.Write(new byte[] { 0x7E, 0x00, 0x10, 0x17, 0x05, 0x00, 0x13, 0xA2, 0x00,
0x40, 0x3D, 0xC7, 0x7D, 0xFF, 0xFE, 0x02, 0x44, 0x31, 0x05, 0xF4 }, 0, 20);
    txtData.Text = txtData.Text + "(" + DateTime.Now + ") -> " + "API : 17   Auto Mode, Turn
on DIO Channel 1 on Router2" + "\r\n";
    ///// WRITE FILE LOG /////
    if (File.Exists("log/" + fileNameTemp + ".txt"))
    {
      TextReader tr = new StreamReader("log/" + fileNameTemp + ".txt");
      string contentReader = tr.ReadToEnd();
      tr.Close();
      TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
      tw.Write(contentReader + "\r\n" + "(" + DateTime.Now + ") -> " + "API : 17   Auto Mode,
Turn on DIO Channel 1 on Router2");
      tw.Close();
    }
    else
    {
      TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
      tw.Write("(" + DateTime.Now + ") -> " + "API : 17   Auto Mode, Turn on DIO Channel 1
on Router2");

      tw.Close();
    }
    ///// END WRITE FILE TO LOG /////
  }
  else if (router2ConvertSample1 > measureCThreshold2max)
  {
    router2out1.Text = "OFF";
    flagled02 = false;
    serialPortxbee.Write(new byte[] { 0x7E, 0x00, 0x10, 0x17, 0x05, 0x00, 0x13, 0xA2, 0x00,
0x40, 0x3D, 0xC7, 0x7D, 0xFF, 0xFE, 0x02, 0x44, 0x31, 0x04, 0xF5 }, 0, 20);
    txtData.Text = txtData.Text + "(" + DateTime.Now + ") -> " + "API : 17   Auto Mode, Turn
off DIO Channel 1 o Router2" + "\r\n";
    ///// WRITE FILE LOG /////
    if (File.Exists("log/" + fileNameTemp + ".txt"))
```

```
                    {
                        TextReader tr = new StreamReader("log/" + fileNameTemp + ".txt");
                        string contentReader = tr.ReadToEnd();
                        tr.Close();
                        TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                        tw.Write(contentReader + "\r\n" + "(" + DateTime.Now + ") -> " + "API : 17   Auto Mode,
Turn off DIO Channel 1 on Router2");
                        tw.Close();
                    }
                    else
                    {
                        TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                        tw.Write("(" + DateTime.Now + ") -> " + "API : 17   Auto Mode, Turn off DIO Channel 1
on Router2");

                        tw.Close();
                    }
                    ///// END WRITE FILE TO LOG /////
                }
            }
        }
        if (macIDHigh == 0x0013A200 && macIDLow == 0x403DC71D)
        {
            statusRouter3 = true;
            timeLabelRouter3.Text = DateTime.Now.ToString();
            router3levelSample1 = readAnalogSample1;
            router3VoltSample1 = (router3levelSample1 * 1200) / 1024;
            router3ConvertSample1 = router3VoltSample1 / 10;
            //ToMySQL
            string insertSQL = "INSERT INTO zb_datatableanalog (zb_macHigh, zb_macLow, zb_upDate,
zb_upTime, zb_raw_io1, zb_data_io1) VALUES (' " + macIDHigh.ToString("X2") + " ',' " +
macIDLow.ToString("X2") + " ',' " + DateTime.Now.ToString("yyyy-MM-dd") + " ',' " +
DateTime.Now.ToString("HH:mm:ss") + " ',' " + readAnalogSample1.ToString("X2") + " ',' " +
router3ConvertSample1.ToString() + " ')";
            MySqlConnection objMyCon = new MySqlConnection(strProvider);
            MySqlCommand cmd = new MySqlCommand(insertSQL, objMyCon);
            objMyCon.Open();
            int executed = 0;
            executed = cmd.ExecuteNonQuery();
            objMyCon.Close();
            //End ToMySQL
        }
        //ALWAYS BOTTOM
        this.txtData.SelectionStart = txtData.Text.Length;
        this.txtData.ScrollToCaret();
    }
    //CHECK MODE SELECT API ID
    if (apiID == 0x97)                          // *** Remote Command Response
    {
        macIDHigh = (((((MyPacket[5] << 8) | MyPacket[6]) << 8) | MyPacket[7]) << 8) | MyPacket[8];
        macIDLow = (((((MyPacket[9] << 8) | MyPacket[10]) << 8) | MyPacket[11]) << 8) | MyPacket[12];
        int cmdFunc = (MyPacket[15] << 8) | MyPacket[16];
        int digitalSamples = (MyPacket[22] << 8) | MyPacket[23];
        int readAnalogSample1 = (MyPacket[24] << 8) | MyPacket[25];

        if (cmdFunc == 0x4953)
        {
            //DISPLAY ON MONITOR SCREEN
            txtData.Text = txtData.Text + "(" + DateTime.Now + ") -> " + "API : " + apiID.ToString("X2") + "
MAC :" + macIDHigh.ToString("X2") + " " + macIDLow.ToString("X2");
            ///// WRITE FILE LOG /////
```

```
                string fileNameTemp = "logger-" + DateTime.Now.Date.ToString("yyyy-M");
                if (File.Exists("log/" + fileNameTemp + ".txt"))
                {
                  TextReader tr = new StreamReader("log/" + fileNameTemp + ".txt");
                  string contentReader = tr.ReadToEnd();
                  tr.Close();
                  TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                  tw.Write(contentReader + "\r\n" + "(" + DateTime.Now + ") -> " + "API : " + apiID.ToString("X2")
+ "   MAC :" + macIDHigh.ToString("X2") + " " + macIDLow.ToString("X2"));
                  tw.Close();
                }
                else
                {
                  TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                  tw.Write("(" + DateTime.Now + ") -> " + "API : " + apiID.ToString("X2") + "   MAC :" +
macIDHigh.ToString("X2") + " " + macIDLow.ToString("X2"));
                  tw.Close();
                }
                ///// END WRITE FILE TO LOG /////
                if (digitalSamples == 0x0002)
                {
                  txtData.Text = txtData.Text + "   O/P : on";
                  ///// WRITE FILE LOG /////
                  if (File.Exists("log/" + fileNameTemp + ".txt"))
                  {
                    TextReader tr = new StreamReader("log/" + fileNameTemp + ".txt");
                    string contentReader = tr.ReadToEnd();
                    tr.Close();
                    TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                    tw.Write(contentReader + "   O/P : on ");
                    tw.Close();
                  }
                  ///// END WRITE FILE TO LOG /////
                  //ToMySQL
                  string insertSQL = "INSERT INTO zb_datatabledigital (zb2_macHigh, zb2_macLow, zb2_upDate,
zb2_upTime, zb2_data_io1) VALUES (' " + macIDHigh.ToString("X2") + " ',' " + macIDLow.ToString("X2") + " ',' "
+ DateTime.Now.ToString("yyyy-MM-dd") + " ',' " + DateTime.Now.ToString("HH:mm:ss") + " ',' " + "on" + " ')";
                  MySqlConnection objMyCon = new MySqlConnection(strProvider);
                  MySqlCommand cmd = new MySqlCommand(insertSQL, objMyCon);
                  objMyCon.Open();
                  int executed = 0;
                  executed = cmd.ExecuteNonQuery();
                  objMyCon.Close();
                  //End ToMySQL
                  if (macIDHigh == 0x0013A200 && macIDLow == 0x4055B1CC)
                  {
                    flagled01 = true;
                    router1out1.Text = "ON";
                  }
                  if (macIDHigh == 0x0013A200 && macIDLow == 0x403DC77D)
                  {
                    flagled02 = true;
                    router2out1.Text = "ON";
                  }
                  if (macIDHigh == 0x0013A200 && macIDLow == 0x403DC71D)
                  {
                  }
                }
                else if (digitalSamples == 0x0000)
                {
```

```
            txtData.Text = txtData.Text + "   O/P : off";
            ///// WRITE FILE LOG /////
            if (File.Exists("log/" + fileNameTemp + ".txt"))
            {
               TextReader tr = new StreamReader("log/" + fileNameTemp + ".txt");
               string contentReader = tr.ReadToEnd();
               tr.Close();
               TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
               tw.Write(contentReader + "   O/P : off");
               tw.Close();
            }
            ///// END WRITE FILE TO LOG /////
            //ToMySQL
            string insertSQL = "INSERT INTO zb_datatabledigital (zb2_macHigh, zb2_macLow, zb2_upDate,
zb2_upTime, zb2_data_io1) VALUES (' " + macIDHigh.ToString("X2") + " ',' " + macIDLow.ToString("X2") + " ',' "
+ DateTime.Now.ToString("yyyy-MM-dd") + " ',' " + DateTime.Now.ToString("HH:mm:ss") + " ',' " + "off" + " ')";
            MySqlConnection objMyCon = new MySqlConnection(strProvider);
            MySqlCommand cmd = new MySqlCommand(insertSQL, objMyCon);
            objMyCon.Open();
            int executed = 0;
            executed = cmd.ExecuteNonQuery();
            objMyCon.Close();
            //End ToMySQL
            if (macIDHigh == 0x0013A200 && macIDLow == 0x4055B1CC)
            {
               flagled01 = false;
               router1out1.Text = "OFF";
            }
            if (macIDHigh == 0x0013A200 && macIDLow == 0x403DC77D)
            {
               flagled02 = false;
               router2out1.Text = "OFF";
            }
            if (macIDHigh == 0x0013A200 && macIDLow == 0x403DC71D)
            {
            }
         }
         txtData.Text = txtData.Text + "   Analog : " + readAnalogSample1.ToString("X2") + "\r\n";
         ///// WRITE FILE LOG /////
         if (File.Exists("log/" + fileNameTemp + ".txt"))
         {
            TextReader tr = new StreamReader("log/" + fileNameTemp + ".txt");
            string contentReader = tr.ReadToEnd();
            tr.Close();
            TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
            tw.Write(contentReader + "   Analog : " + readAnalogSample1.ToString("X2"));
            tw.Close();
         }
         else
         {
            TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
            tw.Write("   Analog : " + readAnalogSample1.ToString("X2"));
            tw.Close();
         }
         ///// END WRITE FILE TO LOG /////
         // Again Update I/P and O/P Status
         if (macIDHigh == 0x0013A200 && macIDLow == 0x4055B1CC)
         {
            statusRouter1 = true;
            timeLabelRouter1.Text = DateTime.Now.ToString();
```

```
                    timeConfRouter1.Text = DateTime.Now.ToString();
                    router1levelSample1 = readAnalogSample1;
                    router1VoltSample1 = (router1levelSample1 * 1200) / 1024;
                    router1ConvertSample1 = (router1VoltSample1 / 10) ;
                    //ToMySQL
                    string insertSQL = "INSERT INTO zb_datatableanalog (zb_macHigh, zb_macLow, zb_upDate,
zb_upTime, zb_raw_io1, zb_data_io1) VALUES ( ' " + macIDHigh.ToString("X2") + " ',' " +
macIDLow.ToString("X2") + " ',' " + DateTime.Now.ToString("yyyy-MM-dd") + " ',' " +
DateTime.Now.ToString("HH:mm:ss") + " ',' " + readAnalogSample1.ToString("X2") + " ',' " +
router1ConvertSample1.ToString() + " ' )";
                    MySqlConnection objMyCon = new MySqlConnection(strProvider);
                    MySqlCommand cmd = new MySqlCommand(insertSQL, objMyCon);
                    objMyCon.Open();
                    int executedx = 0;
                    executedx = cmd.ExecuteNonQuery();
                    objMyCon.Close();
                    //End ToMySQL
                    //Case Auto Mode
                    if (networkMode == true)
                    {
                      var appConfig =
ConfigurationManager.OpenExeConfiguration(Assembly.GetExecutingAssembly().Location);
                      double measureCThreshold1max =
Convert.ToDouble(appConfig.AppSettings.Settings["CThreshold1max"].Value);
                      //Calculate Threshold
                      if (router1ConvertSample1 > measureCThreshold1max)
                      {
                        router1out1.Text = "ON";
                        flagled01 = true;
                        serialPortxbee.Write(new byte[] { 0x7E, 0x00, 0x10, 0x17, 0x05, 0x00, 0x13, 0xA2, 0x00,
0x40, 0x55, 0xB1, 0xCC, 0xFF, 0xFE, 0x02, 0x44, 0x31, 0x05, 0xA3 }, 0, 20);
                        txtData.Text = txtData.Text + "(" + DateTime.Now + ") -> " + "API : 17  Auto Mode, Turn on
DIO Channel 1 on Router1" + "\r\n";
                        ///// WRITE FILE LOG /////
                        if (File.Exists("log/" + fileNameTemp + ".txt"))
                        {
                          TextReader tr = new StreamReader("log/" + fileNameTemp + ".txt");
                          string contentReader = tr.ReadToEnd();
                          tr.Close();
                          TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                          tw.Write(contentReader + "\r\n" + "(" + DateTime.Now + ") -> " + "API : 17   Auto Mode,
Turn on DIO Channel 1 on Router1");
                          tw.Close();
                        }
                        else
                        {
                          TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                          tw.Write("(" + DateTime.Now + ") -> " + "API : 17   Auto Mode, Turn on DIO Channel 1 on
Router1");

                          tw.Close();
                        }
                        ///// END WRITE FILE TO LOG /////
                      }
                      else if (router1ConvertSample1 <= measureCThreshold1max)
                      {
                        router1out1.Text = "OFF";
                        flagled01 = false;
                        serialPortxbee.Write(new byte[] { 0x7E, 0x00, 0x10, 0x17, 0x05, 0x00, 0x13, 0xA2, 0x00,
0x40, 0x55, 0xB1, 0xCC, 0xFF, 0xFE, 0x02, 0x44, 0x31, 0x04, 0xA4 }, 0, 20);
```

```
                txtData.Text = txtData.Text + "(" + DateTime.Now + ") -> " + "API : 17   Auto Mode, Turn off DIO Channel 1 on Router1" + "\r\n";
                ///// WRITE FILE LOG /////
                if (File.Exists("log/" + fileNameTemp + ".txt"))
                {
                  TextReader tr = new StreamReader("log/" + fileNameTemp + ".txt");
                  string contentReader = tr.ReadToEnd();
                  tr.Close();
                  TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                  tw.Write(contentReader + "\r\n" + "(" + DateTime.Now + ") -> " + "API : 17   Auto Mode, Turn off DIO Channel 1 on Router1");
                  tw.Close();
                }
                else
                {
                  TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                  tw.Write("(" + DateTime.Now + ") -> " + "API : 17   Auto Mode, Turn off DIO Channel 1 on Router1");

                  tw.Close();
                }
                ///// END WRITE FILE TO LOG /////
              }
            }
          }
          if (macIDHigh == 0x0013A200 && macIDLow == 0x403DC77D)
          {
            statusRouter2 = true;
            timeLabelRouter2.Text = DateTime.Now.ToString();
            timeConfRouter2.Text = DateTime.Now.ToString();
            router2levelSample1 = readAnalogSample1;
            router2VoltSample1 = (router2levelSample1 * 1.2) / 1024;
            router2VoltSample1 = router2VoltSample1 * 3;
            router2ConvertSample1 = (34.112 * router2VoltSample1) - 20.388;
            //ToMySQL
            string insertSQL = "INSERT INTO zb_datatableanalog (zb_macHigh, zb_macLow, zb_upDate, zb_upTime, zb_raw_io1, zb_data_io1) VALUES (' " + macIDHigh.ToString("X2") + " ',' " + macIDLow.ToString("X2") + " ',' " + DateTime.Now.ToString("yyyy-MM-dd") + " ',' " + DateTime.Now.ToString("HH:mm:ss") + " ',' " + readAnalogSample1.ToString("X2") + " ',' " + router2ConvertSample1.ToString() + " ')";
            MySqlConnection objMyCon = new MySqlConnection(strProvider);
            MySqlCommand cmd = new MySqlCommand(insertSQL, objMyCon);
            objMyCon.Open();
            int executed = 0;
            executed = cmd.ExecuteNonQuery();
            objMyCon.Close();
            //End ToMySQL
            //Case Auto Mode
            if (networkMode == true)
            {
              var appConfig = ConfigurationManager.OpenExeConfiguration(Assembly.GetExecutingAssembly().Location);
              double measureCThreshold2max = Convert.ToDouble(appConfig.AppSettings.Settings["CThreshold2max"].Value);
              //Calculate Threshold
              if (router2ConvertSample1 <= measureCThreshold2max)
              {
                router2out1.Text = "ON";
                flagled02 = true;
                serialPortxbee.Write(new byte[] { 0x7E, 0x00, 0x10, 0x17, 0x05, 0x00, 0x13, 0xA2, 0x00, 0x40, 0x3D, 0xC7, 0x7D, 0xFF, 0xFE, 0x02, 0x44, 0x31, 0x05, 0xF4 }, 0, 20);
```

```
                            txtData.Text = txtData.Text + "(" + DateTime.Now + ") -> " + "API : 17   Auto Mode, Turn on
DIO Channel 1 on Router2" + "\r\n";
                            ///// WRITE FILE LOG /////
                            if (File.Exists("log/" + fileNameTemp + ".txt"))
                            {
                              TextReader tr = new StreamReader("log/" + fileNameTemp + ".txt");
                              string contentReader = tr.ReadToEnd();
                              tr.Close();
                              TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                              tw.Write(contentReader + "\r\n" + "(" + DateTime.Now + ") -> " + "API : 17   Auto Mode,
Turn on DIO Channel 1 on Router2");
                              tw.Close();
                            }
                            else
                            {
                              TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                              tw.Write("(" + DateTime.Now + ") -> " + "API : 17   Auto Mode, Turn on DIO Channel 1 on
Router2");

                              tw.Close();
                            }
                            ///// END WRITE FILE TO LOG /////
                          }
                          else if (router2ConvertSample1 > measureCThreshold2max)
                          {
                            router2out1.Text = "OFF";
                            flagled02 = false;
                            serialPortxbee.Write(new byte[] { 0x7E, 0x00, 0x10, 0x17, 0x05, 0x00, 0x13, 0xA2, 0x00,
0x40, 0x3D, 0xC7, 0x7D, 0xFF, 0xFE, 0x02, 0x44, 0x31, 0x04, 0xF5 }, 0, 20);
                            txtData.Text = txtData.Text + "(" + DateTime.Now + ") -> " + "API : 17   Auto Mode, Turn off
DIO Channel 1 o Router2" + "\r\n";
                            ///// WRITE FILE LOG /////
                            if (File.Exists("log/" + fileNameTemp + ".txt"))
                            {
                              TextReader tr = new StreamReader("log/" + fileNameTemp + ".txt");
                              string contentReader = tr.ReadToEnd();
                              tr.Close();
                              TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                              tw.Write(contentReader + "\r\n" + "(" + DateTime.Now + ") -> " + "API : 17   Auto Mode,
Turn off DIO Channel 1 on Router2");
                              tw.Close();
                            }
                            else
                            {
                              TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                              tw.Write("(" + DateTime.Now + ") -> " + "API : 17   Auto Mode, Turn off DIO Channel 1 on
Router2");

                              tw.Close();
                            }
                            ///// END WRITE FILE TO LOG /////
                          }
                        }
                      }
                      if (macIDHigh == 0x0013A200 && macIDLow == 0x403DC71D)
                      {
                        statusRouter3 = true;
                        timeLabelRouter3.Text = DateTime.Now.ToString();
                        router3levelSample1 = readAnalogSample1;
                        router3VoltSample1 = (router3levelSample1 * 1200) / 1024;
                        router3ConvertSample1 = router3VoltSample1 / 10;
                        //ToMySQL
```

```
                    string insertSQL = "INSERT INTO zb_datatableanalog (zb_macHigh, zb_macLow, zb_upDate,
zb_upTime, zb_raw_io1, zb_data_io1) VALUES (' " + macIDHigh.ToString("X2") + " ',' " +
macIDLow.ToString("X2") + " ',' " + DateTime.Now.ToString("yyyy-MM-dd") + " ',' " +
DateTime.Now.ToString("HH:mm:ss") + " ',' " + readAnalogSample1.ToString("X2") + " ',' " +
router3ConvertSample1.ToString() + " ')";
                    MySqlConnection objMyCon = new MySqlConnection(strProvider);
                    MySqlCommand cmd = new MySqlCommand(insertSQL, objMyCon);
                    objMyCon.Open();
                    int executed = 0;
                    executed = cmd.ExecuteNonQuery();
                    objMyCon.Close();
                    //End ToMySQL
                }
                //ALWAYS BOTTOM
                this.txtData.SelectionStart = txtData.Text.Length;
                this.txtData.ScrollToCaret();
            }
        }
        if (apiID == 0x95)                    // *** Node Identification Indicator (AO=0)
        {
            //NAME API ID = OFFSET 3
            int macIDHighNii = (((((MyPacket[4] << 8) | MyPacket[5]) << 8) | MyPacket[6]) << 8) | MyPacket[7];
            int macIDLowNii = (((((MyPacket[8] << 8) | MyPacket[9]) << 8) | MyPacket[10]) << 8) | MyPacket[11];
            txtData.Text = txtData.Text + "(" + DateTime.Now + ") -> " + "API : " + apiID.ToString("X2") + "
Accept Node to join... " + "MAC :" + macIDHighNii.ToString("X2") + " " + macIDLowNii.ToString("X2") + "\r\n";
                ///// WRITE FILE LOG /////
                string fileNameTemp = "logger-" + DateTime.Now.Date.ToString("yyyy-M");
                if (File.Exists("log/" + fileNameTemp + ".txt"))
                {
                    TextReader tr = new StreamReader("log/" + fileNameTemp + ".txt");
                    string contentReader = tr.ReadToEnd();
                    tr.Close();
                    TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                    tw.Write(contentReader + "\r\n" + "(" + DateTime.Now + ") -> " + "API : " +
apiID.ToString("X2") + "   Accept Node to join... " + "MAC :" + macIDHighNii.ToString("X2") + " " +
macIDLowNii.ToString("X2"));
                    tw.Close();
                }
                else
                {
                    TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                    tw.Write("(" + DateTime.Now + ") -> " + "API : " + apiID.ToString("X2") + "   Accept Node to
join... " + "MAC :" + macIDHighNii.ToString("X2") + " " + macIDLowNii.ToString("X2"));
                    tw.Close();
                }
                ///// END WRITE FILE TO LOG /////
            //ALWAYS BOTTOM
            this.txtData.SelectionStart = txtData.Text.Length;
            this.txtData.ScrollToCaret();
            //Do Update label
            if (macIDHighNii == 0x0013A200 && macIDLowNii == 0x4055B1CC)
            {
                timeLabelRouter1.Text = DateTime.Now.ToString();
                statusRouter1 = true;
            }
            else if (macIDHighNii == 0x0013A200 && macIDLowNii == 0x403DC77D)
            {
                timeLabelRouter2.Text = DateTime.Now.ToString();
                statusRouter2 = true;
            }
```

```
            else if (macIDHighNii == 0x0013A200 && macIDLowNii == 0x403DC71D)
            {
                timeLabelRouter3.Text = DateTime.Now.ToString();
                statusRouter3 = true;
            }
        }

    }
    // ELSE WRONG CHECK SUM
    else { }
  }
}

private void VariableUpdate(object s, EventArgs e)
{
   //Have any value in router1 Input1
   if (router1ConvertSample1 > 000.00)
   {
      router1in1.Text = router1ConvertSample1.ToString("##.##");
      router1in1view.Text = router1ConvertSample1.ToString("##.##");
      rawRouter1.Text = router1levelSample1.ToString("####") + " / 1023";
      analogRouter1.Text = router1VoltSample1.ToString("##.##");
      tempRouter1.Text = router1ConvertSample1.ToString("##.##");

      //Digital Label Status
      if (flagled01 == true)
      {
         outputRouter1.Text = "Power On";
      }
      else
      {
         outputRouter1.Text = "Power Off";
      }
   }
   else
   {
      router1in1.Text = router1ConvertSample1.ToString("N/A");
      router1in1view.Text = router1ConvertSample1.ToString("N/A");
   }
   //Have any value in router2 Input1
   if (router2ConvertSample1 > 00)
   {
      router2in1.Text = router2ConvertSample1.ToString("##.##");
      router2in1view.Text = router2ConvertSample1.ToString("##.##");
      rawRouter2.Text = router2levelSample1.ToString("####") + " / 1023";
      analogRouter2.Text = router2VoltSample1.ToString("#.##");
      tempRouter2.Text = router2ConvertSample1.ToString("##");

      //Digital Label Status
      if (flagled02 == true)
      {
         outputRouter2.Text = "Power On";
      }
      else
      {
         outputRouter2.Text = "Power Off";
      }
   }
   else
   {
```

```
            router2in1.Text = router2ConvertSample1.ToString("N/A");
            router2in1view.Text = router2ConvertSample1.ToString("N/A");
        }
        //Have any value in router3 Input1
        if (router3ConvertSample1 > 000.00)
        {
            router3in1.Text = router3ConvertSample1.ToString("##.##");
            router3in1view.Text = router3ConvertSample1.ToString("##.##");
            rawRouter3.Text = router3levelSample1.ToString("####") + " / 1023";
            analogRouter3.Text = router3VoltSample1.ToString("##.##");
            tempRouter3.Text = router3ConvertSample1.ToString("##.##");
        }
        else
        {
            router3in1.Text = router3ConvertSample1.ToString("N/A");
            router3in1view.Text = router3ConvertSample1.ToString("N/A");
        }
    }

    private void DetectUpdate(object s, EventArgs e)
    {
        //Label Coordinator
        if (statusCoordinator == true)
        {
            coordinatorLabel.ForeColor = Color.Green;
            imgStatusCoordinator.ImageLocation = Application.StartupPath + "\\image/Accept.png";
            timeLabelCoordinator.Text = DateTime.Now.ToString();
        }
        else if (statusCoordinator == false)
        {
            coordinatorLabel.ForeColor = Color.Red;
            imgStatusCoordinator.ImageLocation = Application.StartupPath + "\\image/Alert.png";
        }
        //Label Router1
        if (statusRouter1 == true)
        {
            statRouter1.Text = "Online";
            statRouter1.ForeColor = Color.Green;
            routerLabel1.ForeColor = Color.Green;
            imgStatusRouter1.ImageLocation = Application.StartupPath + "\\image/Accept.png";
        }
        else if (statusRouter1 == false)
        {
            statRouter1.Text = "Offline";
            statRouter1.ForeColor = Color.Red;
            routerLabel1.ForeColor = Color.Red;
            imgStatusRouter1.ImageLocation = Application.StartupPath + "\\image/Alert.png";
        }
        //Label Router2
        if (statusRouter2 == true)
        {
            statRouter2.Text = "Online";
            statRouter2.ForeColor = Color.Green;
            routerLabel2.ForeColor = Color.Green;
            imgStatusRouter2.ImageLocation = Application.StartupPath + "\\image/Accept.png";
        }
        else if (statusRouter2 == false)
        {
            statRouter2.Text = "Offline";
            statRouter2.ForeColor = Color.Red;
```

```
                routerLabel2.ForeColor = Color.Red;
                imgStatusRouter2.ImageLocation = Application.StartupPath + "\\image/Alert.png";
            }
                //Label Router3
            if (statusRouter3 == true)
            {
                statRouter3.Text = "Online";
                statRouter3.ForeColor = Color.Green;
                routerLabel3.ForeColor = Color.Green;
                imgStatusRouter3.ImageLocation = Application.StartupPath + "\\image/Accept.png";
            }
            else if (statusRouter3 == false)
            {
                statRouter3.Text = "Offline";
                statRouter3.ForeColor = Color.Red;
                routerLabel3.ForeColor = Color.Red;
                imgStatusRouter3.ImageLocation = Application.StartupPath + "\\image/Alert.png";
            }
        }

        private int iCount, iCount2;

        private void onlineTimeCheck_Tick(object sender, EventArgs e)
        {
            iCount--;
            if (iCount > 0)
            {
                timeLabel.Text = iCount.ToString();
            }
            else if (iCount == 0)
            {
                timeLabel.Text = "Refreshing....";
                labelSecond.Text = "";
                //Reset Label
                statusCoordinator = false;
                statusRouter1 = false;
                statusRouter2 = false;
                statusRouter3 = false;
                timeLabelCoordinator.Text = "Waiting...";
                timeLabelRouter1.Text = "Waiting...";
                timeLabelRouter2.Text = "Waiting...";
                timeLabelRouter3.Text = "Waiting...";
                timeConfRouter1.Text = "Waiting...";
                timeConfRouter2.Text = "Waiting...";
                timeConfRouter3.Text = "Waiting...";
                serialPortxbee.Write(new byte[] { 0x7E, 0x00, 0x04, 0x08, 0x01, 0x4E, 0x44, 0x64 }, 0, 8);
                txtData.Text = txtData.Text + "(" + DateTime.Now + ") -> " + "API : 08   Peroid Network Checking..." +
"\r\n";
                        ///// WRITE FILE LOG /////
                        string fileNameTemp = "logger-" + DateTime.Now.Date.ToString("yyyy-M");
                        if (File.Exists("log/" + fileNameTemp + ".txt"))
                        {
                            TextReader tr = new StreamReader("log/" + fileNameTemp + ".txt");
                            string contentReader = tr.ReadToEnd();
                            tr.Close();
                            TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                            tw.Write(contentReader + "(" + DateTime.Now + ") -> " + "API : 08   Peroid Network
Checking...");
                            tw.Close();
                        }
```

```
                else
                {
                    TextWriter tw = new StreamWriter("log/" + fileNameTemp + ".txt");
                    tw.Write("(" + DateTime.Now + ") -> " + "API : 08   Peroid Network Checking...");
                    tw.Close();
                }
                ///// END WRITE FILE TO LOG /////
            //ALWAYS BOTTOM
            this.txtData.SelectionStart = txtData.Text.Length;
            this.txtData.ScrollToCaret();
            this.Invoke(new EventHandler(DetectUpdate));
        }
        else
        {
            Configuration config = ConfigurationManager.OpenExeConfiguration(Application.ExecutablePath);
            var appConfig =
ConfigurationManager.OpenExeConfiguration(Assembly.GetExecutingAssembly().Location);
            int timerRefreshValue = Convert.ToInt32(appConfig.AppSettings.Settings["timerRefreshValue"].Value);
            iCount = timerRefreshValue;
            labelSecond.Text = "seconds";
        }
    }

    private void aboutToolStripMenuItem_Click(object sender, EventArgs e)
    {
        frm_about frm_about = new frm_about();
        frm_about.Show();
    }

    private void tmrRefresh_Tick(object sender, EventArgs e)
    {
        lblClockText.Text = DateTime.Now.ToLongDateString();
        lblClockText2.Text = DateTime.Now.ToLongTimeString();
        lblClockText.ForeColor = Color.Yellow;
        lblClockText2.ForeColor = Color.Yellow;
    }

    private void button1_Click(object sender, EventArgs e)
    {
        serialPortxbee.Write(new byte[] { 0x7E, 0x00, 0x0F, 0x17, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF,
0xFF, 0xFF, 0xFE, 0x00, 0x49, 0x53, 0x50 }, 0, 19);
    }

    private void thresholdSettingToolStripMenuItem_Click(object sender, EventArgs e)
    {
        frm_thresholdrouter frm_thresholdrouter = new frm_thresholdrouter();
        frm_thresholdrouter.Show();
    }

    private void periodRefreshNetworkToolStripMenuItem_Click(object sender, EventArgs e)
    {
        frm_time_period frm_time_period = new frm_time_period();
        frm_time_period.Show();
    }

    private void periodForceDataATISToolStripMenuItem_Click_1(object sender, EventArgs e)
    {
        frm_time_force frm_time_force = new frm_time_force();
        frm_time_force.Show();
    }
```

```
private void txtData_TextChanged(object sender, EventArgs e)
{
    int txtDataLineCount = txtData.Lines.Count();
    if (txtDataLineCount >= 50)
    {
        txtData.Clear();
    }
}

private void refreshForceTimer_Click(object sender, EventArgs e)
{
    // Load the current application's configuration file
    Configuration config = ConfigurationManager.OpenExeConfiguration(Application.ExecutablePath);
    var appConfig = ConfigurationManager.OpenExeConfiguration(Assembly.GetExecutingAssembly().Location);
    string timerForceSample = Convert.ToString(appConfig.AppSettings.Settings["timerForceSample"].Value);
    int timerForceValue = Convert.ToInt32(appConfig.AppSettings.Settings["timerForceValue"].Value);
    iCount2 = timerForceValue;
    timeLabel.Text = iCount.ToString();
    if (timerForceSample == "on")
    {
        onlineForceCheck.Enabled = true;  //Start timer online label check
    }
    if (timerForceSample == "off")
    {
        onlineForceCheck.Enabled = false;  //Stop timer online label check
    }
}

private void onlineForceCheck_Tick(object sender, EventArgs e)
{
    iCount2--;
    if (iCount2 > 0)
    {
        timeForceLabel.Text = iCount2.ToString();
    }
    else if (iCount2 == 0)
    {
        timeForceLabel.Text = "Refreshing....";
        serialPortxbee.Write(new byte[] { 0x7E, 0x00, 0x0F, 0x17, 0x01, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0xFF,
0xFF, 0xFF, 0xFE, 0x00, 0x49, 0x53, 0x50 }, 0, 19);
    }
    else
    {
        Configuration config = ConfigurationManager.OpenExeConfiguration(Application.ExecutablePath);
        var appConfig =
ConfigurationManager.OpenExeConfiguration(Assembly.GetExecutingAssembly().Location);
        int timerForceValue = Convert.ToInt32(appConfig.AppSettings.Settings["timerForceValue"].Value);
        iCount2 = timerForceValue;
    }
}

private void historyToolStripMenuItem_Click(object sender, EventArgs e)
{
    string myPath = @"C:\Users\Inkyman\Desktop\testXbeeOut\XbeeSerialHex\XbeeSerialHex\bin\Debug\log";
    System.Diagnostics.Process prc = new System.Diagnostics.Process();
    prc.StartInfo.FileName = myPath;
    prc.Start();
}
```

```
private void imgRouter1_Click(object sender, EventArgs e)
{
   tabXbee.SelectedTab = tabRouter1;
}

private void imgRouter2_Click(object sender, EventArgs e)
{
   tabXbee.SelectedTab = tabRouter2;
}

private void imgRouter3_Click(object sender, EventArgs e)
{
   tabXbee.SelectedTab = tabRouter3;
}

private void imgServer_Click(object sender, EventArgs e)
{
   tabXbee.SelectedTab = tabOverview;
}

//Blink when offline
private void ImageBlink_Tick(object sender, EventArgs e)
{
   //Check on Coordinator
   if (statusCoordinator == true)
   {
      imgWarnCoor.Visible = false;
   }
   else
   {
      //Coordinator
      if (imgWarnCoor.Visible == true)
      {
         imgWarnCoor.Visible = false;
      }
      else
      {
         imgWarnCoor.Visible = true;
      }
   }
   //Check on Router1
   if (statusRouter1 == true)
   {
      imgWarnR1.Visible = false;
   }
   else
   {
      //Router1
      if (imgWarnR1.Visible == true)
      {
         imgWarnR1.Visible = false;
      }
      else
      {
         imgWarnR1.Visible = true;
      }
   }
   //Check on Router2
   if (statusRouter2 == true)
   {
```

```
            imgWarnR2.Visible = false;
          }
          else
          {
            //Router2
            if (imgWarnR2.Visible == true)
            {
              imgWarnR2.Visible = false;
            }
            else
            {
              imgWarnR2.Visible = true;
            }
          }
          //Check on Router3
          if (statusRouter3 == true)
          {
            imgWarnR3.Visible = false;
          }
          else
          {
            //Router3
            if (imgWarnR3.Visible == true)
            {
              imgWarnR3.Visible = false;
            }
            else
            {
              imgWarnR3.Visible = true;
            }
          }
        }

      }
    }
```

ภาคผนวก (จ)

ซอฟท์แวร์เฝ้าระวังและควบคุม ระบบ Zigbee

ในส่วนของ Design Form

```
namespace XbeeSerialHex
{
  partial class Form1
  {
    /// <summary>
    /// Required designer variable.
    /// </summary>
    private System.ComponentModel.IContainer components = null;

    /// <summary>
    /// Clean up any resources being used.
    /// </summary>
    /// <param name="disposing">true if managed resources should be disposed; otherwise, false.</param>
    protected override void Dispose(bool disposing)
    {
      if (disposing && (components != null))
      {
        components.Dispose();
      }
      base.Dispose(disposing);
    }

    #region Windows Form Designer generated code

    /// <summary>
    /// Required method for Designer support - do not modify
    /// the contents of this method with the code editor.
    /// </summary>
    private void InitializeComponent()
    {
      this.components = new System.ComponentModel.Container();
      System.ComponentModel.ComponentResourceManager resources = new
System.ComponentModel.ComponentResourceManager(typeof(Form1));
      this.btnConnect = new System.Windows.Forms.Button();
      this.serialCombox = new System.Windows.Forms.ComboBox();
      this.serialBaudbox = new System.Windows.Forms.ComboBox();
      this.txtData = new System.Windows.Forms.TextBox();
      this.groupBox1 = new System.Windows.Forms.GroupBox();
      this.btnRefresh = new System.Windows.Forms.Button();
      this.label9 = new System.Windows.Forms.Label();
      this.label8 = new System.Windows.Forms.Label();
      this.groupBox2 = new System.Windows.Forms.GroupBox();
      this.groupBox3 = new System.Windows.Forms.GroupBox();
      this.setModebt2 = new System.Windows.Forms.RadioButton();
      this.setModebt1 = new System.Windows.Forms.RadioButton();
      this.label5 = new System.Windows.Forms.Label();
      this.panel1 = new System.Windows.Forms.Panel();
      this.modeControlText = new System.Windows.Forms.Label();
      this.menuStrip = new System.Windows.Forms.MenuStrip();
      this.fileToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
      this.toolStripSeparator5 = new System.Windows.Forms.ToolStripSeparator();
      this.exitToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
      this.viewToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
      this.historyToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
      this.internetMonitoringToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
      this.databaseToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
      this.settingToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
      this.thresholdSettingToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
      this.parametersSettingToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
      this.periodRefreshNetworkToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
```

```
this.periodForceDataATISToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
this.toolStripSeparator2 = new System.Windows.Forms.ToolStripSeparator();
this.helpToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
this.toolStripSeparator3 = new System.Windows.Forms.ToolStripSeparator();
this.aboutToolStripMenuItem = new System.Windows.Forms.ToolStripMenuItem();
this.statusStrip = new System.Windows.Forms.StatusStrip();
this.toolStripStatusLabel1 = new System.Windows.Forms.ToolStripStatusLabel();
this.toolStrip = new System.Windows.Forms.ToolStrip();
this.toolStripLabel1 = new System.Windows.Forms.ToolStripLabel();
this.toolStripSeparator1 = new System.Windows.Forms.ToolStripSeparator();
this.tabXbee = new System.Windows.Forms.TabControl();
this.tabOverview = new System.Windows.Forms.TabPage();
this.groupBox7 = new System.Windows.Forms.GroupBox();
this.panel5 = new System.Windows.Forms.Panel();
this.router2out1 = new System.Windows.Forms.Label();
this.panel4 = new System.Windows.Forms.Panel();
this.label17 = new System.Windows.Forms.Label();
this.router3in1 = new System.Windows.Forms.Label();
this.panel6 = new System.Windows.Forms.Panel();
this.router1out1 = new System.Windows.Forms.Label();
this.panel3 = new System.Windows.Forms.Panel();
this.label18 = new System.Windows.Forms.Label();
this.router2in1 = new System.Windows.Forms.Label();
this.label16 = new System.Windows.Forms.Label();
this.label14 = new System.Windows.Forms.Label();
this.label13 = new System.Windows.Forms.Label();
this.label12 = new System.Windows.Forms.Label();
this.label11 = new System.Windows.Forms.Label();
this.label10 = new System.Windows.Forms.Label();
this.panel2 = new System.Windows.Forms.Panel();
this.router1in1 = new System.Windows.Forms.Label();
this.label15 = new System.Windows.Forms.Label();
this.timeLabelRouter1 = new System.Windows.Forms.Label();
this.groupBox6 = new System.Windows.Forms.GroupBox();
this.timeLabelRouter3 = new System.Windows.Forms.Label();
this.timeLabelRouter2 = new System.Windows.Forms.Label();
this.timeLabelCoordinator = new System.Windows.Forms.Label();
this.groupBox5 = new System.Windows.Forms.GroupBox();
this.refreshTimer = new System.Windows.Forms.Button();
this.timeLabel = new System.Windows.Forms.Label();
this.textlabel10 = new System.Windows.Forms.Label();
this.imgStatusRouter3 = new System.Windows.Forms.PictureBox();
this.imgStatusRouter2 = new System.Windows.Forms.PictureBox();
this.imgStatusRouter1 = new System.Windows.Forms.PictureBox();
this.imgStatusCoordinator = new System.Windows.Forms.PictureBox();
this.coordinatorLabel = new System.Windows.Forms.Label();
this.routerLabel3 = new System.Windows.Forms.Label();
this.routerLabel2 = new System.Windows.Forms.Label();
this.routerLabel1 = new System.Windows.Forms.Label();
this.labelSecond = new System.Windows.Forms.Label();
this.tabRouter1 = new System.Windows.Forms.TabPage();
this.groupBox11 = new System.Windows.Forms.GroupBox();
this.statRouter1 = new System.Windows.Forms.Label();
this.groupBox10 = new System.Windows.Forms.GroupBox();
this.thresholdRouter1 = new System.Windows.Forms.Label();
this.label2 = new System.Windows.Forms.Label();
this.groupBox9 = new System.Windows.Forms.GroupBox();
this.btRouter1Off1 = new System.Windows.Forms.Button();
this.btRouter1On1 = new System.Windows.Forms.Button();
this.groupBox8 = new System.Windows.Forms.GroupBox();
```

```
this.label49 = new System.Windows.Forms.Label();
this.label48 = new System.Windows.Forms.Label();
this.label47 = new System.Windows.Forms.Label();
this.label46 = new System.Windows.Forms.Label();
this.outputRouter1 = new System.Windows.Forms.Label();
this.label23 = new System.Windows.Forms.Label();
this.tempRouter1 = new System.Windows.Forms.Label();
this.analogRouter1 = new System.Windows.Forms.Label();
this.rawRouter1 = new System.Windows.Forms.Label();
this.label22 = new System.Windows.Forms.Label();
this.label21 = new System.Windows.Forms.Label();
this.label20 = new System.Windows.Forms.Label();
this.pictureBox2 = new System.Windows.Forms.PictureBox();
this.pictureBox1 = new System.Windows.Forms.PictureBox();
this.label19 = new System.Windows.Forms.Label();
this.timeConfRouter1 = new System.Windows.Forms.Label();
this.tabRouter2 = new System.Windows.Forms.TabPage();
this.groupBox12 = new System.Windows.Forms.GroupBox();
this.statRouter2 = new System.Windows.Forms.Label();
this.groupBox13 = new System.Windows.Forms.GroupBox();
this.thresholdRouter2 = new System.Windows.Forms.Label();
this.label4 = new System.Windows.Forms.Label();
this.groupBox14 = new System.Windows.Forms.GroupBox();
this.btRouter2Off1 = new System.Windows.Forms.Button();
this.btRouter2On1 = new System.Windows.Forms.Button();
this.groupBox15 = new System.Windows.Forms.GroupBox();
this.label36 = new System.Windows.Forms.Label();
this.label37 = new System.Windows.Forms.Label();
this.label38 = new System.Windows.Forms.Label();
this.label39 = new System.Windows.Forms.Label();
this.outputRouter2 = new System.Windows.Forms.Label();
this.label41 = new System.Windows.Forms.Label();
this.tempRouter2 = new System.Windows.Forms.Label();
this.analogRouter2 = new System.Windows.Forms.Label();
this.rawRouter2 = new System.Windows.Forms.Label();
this.label45 = new System.Windows.Forms.Label();
this.label50 = new System.Windows.Forms.Label();
this.label51 = new System.Windows.Forms.Label();
this.pictureBox5 = new System.Windows.Forms.PictureBox();
this.pictureBox6 = new System.Windows.Forms.PictureBox();
this.label52 = new System.Windows.Forms.Label();
this.timeConfRouter2 = new System.Windows.Forms.Label();
this.tabRouter3 = new System.Windows.Forms.TabPage();
this.groupBox16 = new System.Windows.Forms.GroupBox();
this.statRouter3 = new System.Windows.Forms.Label();
this.groupBox19 = new System.Windows.Forms.GroupBox();
this.label40 = new System.Windows.Forms.Label();
this.label42 = new System.Windows.Forms.Label();
this.label43 = new System.Windows.Forms.Label();
this.label44 = new System.Windows.Forms.Label();
this.tempRouter3 = new System.Windows.Forms.Label();
this.analogRouter3 = new System.Windows.Forms.Label();
this.rawRouter3 = new System.Windows.Forms.Label();
this.label58 = new System.Windows.Forms.Label();
this.label59 = new System.Windows.Forms.Label();
this.label60 = new System.Windows.Forms.Label();
this.pictureBox7 = new System.Windows.Forms.PictureBox();
this.pictureBox8 = new System.Windows.Forms.PictureBox();
this.label61 = new System.Windows.Forms.Label();
this.timeConfRouter3 = new System.Windows.Forms.Label();
```

```
this.groupBox4 = new System.Windows.Forms.GroupBox();
this.notifyIcon1 = new System.Windows.Forms.NotifyIcon(this.components);
this.onlineTimeCheck = new System.Windows.Forms.Timer(this.components);
this.label24 = new System.Windows.Forms.Label();
this.button3 = new System.Windows.Forms.Button();
this.button4 = new System.Windows.Forms.Button();
this.label25 = new System.Windows.Forms.Label();
this.label26 = new System.Windows.Forms.Label();
this.label27 = new System.Windows.Forms.Label();
this.label28 = new System.Windows.Forms.Label();
this.label29 = new System.Windows.Forms.Label();
this.label30 = new System.Windows.Forms.Label();
this.label31 = new System.Windows.Forms.Label();
this.label32 = new System.Windows.Forms.Label();
this.pictureBox3 = new System.Windows.Forms.PictureBox();
this.pictureBox4 = new System.Windows.Forms.PictureBox();
this.label33 = new System.Windows.Forms.Label();
this.label34 = new System.Windows.Forms.Label();
this.groupBox17 = new System.Windows.Forms.GroupBox();
this.panel7 = new System.Windows.Forms.Panel();
this.label62 = new System.Windows.Forms.Label();
this.router3in1view = new System.Windows.Forms.Label();
this.router1in1view = new System.Windows.Forms.Label();
this.label64 = new System.Windows.Forms.Label();
this.label57 = new System.Windows.Forms.Label();
this.router2in1view = new System.Windows.Forms.Label();
this.label55 = new System.Windows.Forms.Label();
this.label54 = new System.Windows.Forms.Label();
this.label53 = new System.Windows.Forms.Label();
this.label35 = new System.Windows.Forms.Label();
this.label7 = new System.Windows.Forms.Label();
this.label6 = new System.Windows.Forms.Label();
this.imgWarnCoor = new System.Windows.Forms.PictureBox();
this.imgWarnR3 = new System.Windows.Forms.PictureBox();
this.imgWarnR2 = new System.Windows.Forms.PictureBox();
this.imgWarnR1 = new System.Windows.Forms.PictureBox();
this.pictureBox18 = new System.Windows.Forms.PictureBox();
this.pictureBox17 = new System.Windows.Forms.PictureBox();
this.pictureBox11 = new System.Windows.Forms.PictureBox();
this.imgRouter3 = new System.Windows.Forms.PictureBox();
this.imgRouter2 = new System.Windows.Forms.PictureBox();
this.imgRouter1 = new System.Windows.Forms.PictureBox();
this.imgCoordinator = new System.Windows.Forms.PictureBox();
this.pictureBox12 = new System.Windows.Forms.PictureBox();
this.imgServer = new System.Windows.Forms.PictureBox();
this.panel8 = new System.Windows.Forms.Panel();
this.lblClockText2 = new System.Windows.Forms.Label();
this.label1 = new System.Windows.Forms.Label();
this.lblClockText = new System.Windows.Forms.Label();
this.tmrRefresh = new System.Windows.Forms.Timer(this.components);
this.button1 = new System.Windows.Forms.Button();
this.pictureBox9 = new System.Windows.Forms.PictureBox();
this.groupBox18 = new System.Windows.Forms.GroupBox();
this.label3 = new System.Windows.Forms.Label();
this.refreshForceTimer = new System.Windows.Forms.Button();
this.timeForceLabel = new System.Windows.Forms.Label();
this.onlineForceCheck = new System.Windows.Forms.Timer(this.components);
this.ImageBlink = new System.Windows.Forms.Timer(this.components);
this.label56 = new System.Windows.Forms.Label();
this.label63 = new System.Windows.Forms.Label();
```

```
this.groupBox1.SuspendLayout();
this.groupBox2.SuspendLayout();
this.groupBox3.SuspendLayout();
this.panel1.SuspendLayout();
this.menuStrip.SuspendLayout();
this.statusStrip.SuspendLayout();
this.toolStrip.SuspendLayout();
this.tabXbee.SuspendLayout();
this.tabOverview.SuspendLayout();
this.groupBox7.SuspendLayout();
this.panel5.SuspendLayout();
this.panel4.SuspendLayout();
this.panel6.SuspendLayout();
this.panel3.SuspendLayout();
this.panel2.SuspendLayout();
this.groupBox6.SuspendLayout();
this.groupBox5.SuspendLayout();
((System.ComponentModel.ISupportInitialize)(this.imgStatusRouter3)).BeginInit();
((System.ComponentModel.ISupportInitialize)(this.imgStatusRouter2)).BeginInit();
((System.ComponentModel.ISupportInitialize)(this.imgStatusRouter1)).BeginInit();
((System.ComponentModel.ISupportInitialize)(this.imgStatusCoordinator)).BeginInit();
this.tabRouter1.SuspendLayout();
this.groupBox11.SuspendLayout();
this.groupBox10.SuspendLayout();
this.groupBox9.SuspendLayout();
this.groupBox8.SuspendLayout();
((System.ComponentModel.ISupportInitialize)(this.pictureBox2)).BeginInit();
((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).BeginInit();
this.tabRouter2.SuspendLayout();
this.groupBox12.SuspendLayout();
this.groupBox13.SuspendLayout();
this.groupBox14.SuspendLayout();
this.groupBox15.SuspendLayout();
((System.ComponentModel.ISupportInitialize)(this.pictureBox5)).BeginInit();
((System.ComponentModel.ISupportInitialize)(this.pictureBox6)).BeginInit();
this.tabRouter3.SuspendLayout();
this.groupBox16.SuspendLayout();
this.groupBox19.SuspendLayout();
((System.ComponentModel.ISupportInitialize)(this.pictureBox7)).BeginInit();
((System.ComponentModel.ISupportInitialize)(this.pictureBox8)).BeginInit();
this.groupBox4.SuspendLayout();
((System.ComponentModel.ISupportInitialize)(this.pictureBox3)).BeginInit();
((System.ComponentModel.ISupportInitialize)(this.pictureBox4)).BeginInit();
this.groupBox17.SuspendLayout();
this.panel7.SuspendLayout();
((System.ComponentModel.ISupportInitialize)(this.imgWarnCoor)).BeginInit();
((System.ComponentModel.ISupportInitialize)(this.imgWarnR3)).BeginInit();
((System.ComponentModel.ISupportInitialize)(this.imgWarnR2)).BeginInit();
((System.ComponentModel.ISupportInitialize)(this.imgWarnR1)).BeginInit();
((System.ComponentModel.ISupportInitialize)(this.pictureBox18)).BeginInit();
((System.ComponentModel.ISupportInitialize)(this.pictureBox17)).BeginInit();
((System.ComponentModel.ISupportInitialize)(this.pictureBox11)).BeginInit();
((System.ComponentModel.ISupportInitialize)(this.imgRouter3)).BeginInit();
((System.ComponentModel.ISupportInitialize)(this.imgRouter2)).BeginInit();
((System.ComponentModel.ISupportInitialize)(this.imgRouter1)).BeginInit();
((System.ComponentModel.ISupportInitialize)(this.imgCoordinator)).BeginInit();
((System.ComponentModel.ISupportInitialize)(this.pictureBox12)).BeginInit();
((System.ComponentModel.ISupportInitialize)(this.imgServer)).BeginInit();
this.panel8.SuspendLayout();
((System.ComponentModel.ISupportInitialize)(this.pictureBox9)).BeginInit();
```

```
this.groupBox18.SuspendLayout();
this.SuspendLayout();
//
// btnConnect
//
this.btnConnect.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.btnConnect.Location = new System.Drawing.Point(71, 79);
this.btnConnect.Name = "btnConnect";
this.btnConnect.Size = new System.Drawing.Size(78, 23);
this.btnConnect.TabIndex = 149;
this.btnConnect.Text = "Connect";
this.btnConnect.UseVisualStyleBackColor = true;
this.btnConnect.Click += new System.EventHandler(this.btnConnect_Click);
//
// serialCombox
//
this.serialCombox.Font = new System.Drawing.Font("Microsoft Sans Serif", 10F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.serialCombox.FormattingEnabled = true;
this.serialCombox.Location = new System.Drawing.Point(69, 19);
this.serialCombox.Name = "serialCombox";
this.serialCombox.Size = new System.Drawing.Size(80, 24);
this.serialCombox.TabIndex = 147;
this.serialCombox.SelectedIndexChanged += new
System.EventHandler(this.serialCombox_SelectedIndexChanged_1);
//
// serialBaudbox
//
this.serialBaudbox.Font = new System.Drawing.Font("Microsoft Sans Serif", 10F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.serialBaudbox.FormattingEnabled = true;
this.serialBaudbox.Items.AddRange(new object[] {
"300",
"600",
"1200",
"1800",
"2400",
"4800",
"7200",
"9600",
"14400",
"19200",
"38400",
"57600",
"115200",
"230400"});
this.serialBaudbox.Location = new System.Drawing.Point(68, 49);
this.serialBaudbox.Name = "serialBaudbox";
this.serialBaudbox.Size = new System.Drawing.Size(81, 24);
this.serialBaudbox.TabIndex = 148;
this.serialBaudbox.SelectedIndexChanged += new
System.EventHandler(this.serialBaudbox_SelectedIndexChanged_1);
//
// txtData
//
this.txtData.BackColor = System.Drawing.Color.Navy;
this.txtData.Font = new System.Drawing.Font("Microsoft Sans Serif", 9F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.txtData.ForeColor = System.Drawing.SystemColors.Window;
```

```
this.txtData.Location = new System.Drawing.Point(8, 20);
this.txtData.MaxLength = 0;
this.txtData.Multiline = true;
this.txtData.Name = "txtData";
this.txtData.ReadOnly = true;
this.txtData.Size = new System.Drawing.Size(616, 190);
this.txtData.TabIndex = 150;
this.txtData.TextChanged += new System.EventHandler(this.txtData_TextChanged);
//
// groupBox1
//
this.groupBox1.Controls.Add(this.btnRefresh);
this.groupBox1.Controls.Add(this.serialBaudbox);
this.groupBox1.Controls.Add(this.label9);
this.groupBox1.Controls.Add(this.label8);
this.groupBox1.Controls.Add(this.serialCombox);
this.groupBox1.Controls.Add(this.btnConnect);
this.groupBox1.Location = new System.Drawing.Point(12, 54);
this.groupBox1.Name = "groupBox1";
this.groupBox1.Size = new System.Drawing.Size(157, 112);
this.groupBox1.TabIndex = 155;
this.groupBox1.TabStop = false;
this.groupBox1.Text = "Communication";
//
// btnRefresh
//
this.btnRefresh.Enabled = false;
this.btnRefresh.Location = new System.Drawing.Point(7, 79);
this.btnRefresh.Name = "btnRefresh";
this.btnRefresh.Size = new System.Drawing.Size(60, 23);
this.btnRefresh.TabIndex = 152;
this.btnRefresh.Text = "Refresh";
this.btnRefresh.UseVisualStyleBackColor = true;
this.btnRefresh.Click += new System.EventHandler(this.btnRefresh_Click);
//
// label9
//
this.label9.AutoSize = true;
this.label9.BackColor = System.Drawing.Color.Transparent;
this.label9.Location = new System.Drawing.Point(5, 55);
this.label9.Name = "label9";
this.label9.Size = new System.Drawing.Size(67, 13);
this.label9.TabIndex = 151;
this.label9.Text = "Baud Rate : ";
//
// label8
//
this.label8.AutoSize = true;
this.label8.Location = new System.Drawing.Point(6, 24);
this.label8.Name = "label8";
this.label8.Size = new System.Drawing.Size(55, 13);
this.label8.TabIndex = 150;
this.label8.Text = "Comport : ";
//
// groupBox2
//
this.groupBox2.Controls.Add(this.txtData);
this.groupBox2.Location = new System.Drawing.Point(12, 409);
this.groupBox2.Name = "groupBox2";
this.groupBox2.Size = new System.Drawing.Size(632, 220);
```

```
this.groupBox2.TabIndex = 156;
this.groupBox2.TabStop = false;
this.groupBox2.Text = "Monitoring Command";
//
// groupBox3
//
this.groupBox3.Controls.Add(this.setModebt2);
this.groupBox3.Controls.Add(this.setModebt1);
this.groupBox3.Controls.Add(this.label5);
this.groupBox3.Controls.Add(this.panel1);
this.groupBox3.Location = new System.Drawing.Point(175, 54);
this.groupBox3.Name = "groupBox3";
this.groupBox3.Size = new System.Drawing.Size(157, 112);
this.groupBox3.TabIndex = 157;
this.groupBox3.TabStop = false;
this.groupBox3.Text = "Network Mode Control";
//
// setModebt2
//
this.setModebt2.AutoSize = true;
this.setModebt2.Location = new System.Drawing.Point(72, 81);
this.setModebt2.Name = "setModebt2";
this.setModebt2.Size = new System.Drawing.Size(47, 17);
this.setModebt2.TabIndex = 160;
this.setModebt2.Text = "Auto";
this.setModebt2.UseVisualStyleBackColor = true;
this.setModebt2.Click += new System.EventHandler(this.setModebt2_Click);
//
// setModebt1
//
this.setModebt1.AutoSize = true;
this.setModebt1.Checked = true;
this.setModebt1.Location = new System.Drawing.Point(72, 59);
this.setModebt1.Name = "setModebt1";
this.setModebt1.Size = new System.Drawing.Size(60, 17);
this.setModebt1.TabIndex = 159;
this.setModebt1.TabStop = true;
this.setModebt1.Text = "Manual";
this.setModebt1.UseVisualStyleBackColor = true;
this.setModebt1.Click += new System.EventHandler(this.setModebt1_Click);
//
// label5
//
this.label5.AutoSize = true;
this.label5.Location = new System.Drawing.Point(6, 60);
this.label5.Name = "label5";
this.label5.Size = new System.Drawing.Size(59, 13);
this.label5.TabIndex = 1;
this.label5.Text = "Set Mode :";
//
// panel1
//
this.panel1.BackColor = System.Drawing.SystemColors.ActiveCaptionText;
this.panel1.Controls.Add(this.modeControlText);
this.panel1.Location = new System.Drawing.Point(9, 20);
this.panel1.Name = "panel1";
this.panel1.Size = new System.Drawing.Size(140, 29);
this.panel1.TabIndex = 158;
//
// modeControlText
```

```
//
this.modeControlText.AutoSize = true;
this.modeControlText.BackColor = System.Drawing.Color.Transparent;
this.modeControlText.Font = new System.Drawing.Font("Microsoft Sans Serif", 10F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.modeControlText.ForeColor = System.Drawing.Color.Red;
this.modeControlText.Location = new System.Drawing.Point(10, 6);
this.modeControlText.Name = "modeControlText";
this.modeControlText.Size = new System.Drawing.Size(121, 17);
this.modeControlText.TabIndex = 4;
this.modeControlText.Text = "MANUAL MODE";
this.modeControlText.TextAlign = System.Drawing.ContentAlignment.MiddleCenter;
//
// menuStrip
//
this.menuStrip.BackColor = System.Drawing.SystemColors.ControlLight;
this.menuStrip.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {
this.fileToolStripMenuItem,
this.viewToolStripMenuItem,
this.settingToolStripMenuItem,
this.helpToolStripMenuItem});
this.menuStrip.Location = new System.Drawing.Point(0, 0);
this.menuStrip.Name = "menuStrip";
this.menuStrip.Size = new System.Drawing.Size(1184, 24);
this.menuStrip.TabIndex = 158;
this.menuStrip.Text = "menuStrip1";
//
// fileToolStripMenuItem
//
this.fileToolStripMenuItem.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] {
this.toolStripSeparator5,
this.exitToolStripMenuItem});
this.fileToolStripMenuItem.Name = "fileToolStripMenuItem";
this.fileToolStripMenuItem.Size = new System.Drawing.Size(37, 20);
this.fileToolStripMenuItem.Text = "File";
//
// toolStripSeparator5
//
this.toolStripSeparator5.Name = "toolStripSeparator5";
this.toolStripSeparator5.Size = new System.Drawing.Size(89, 6);
//
// exitToolStripMenuItem
//
this.exitToolStripMenuItem.Name = "exitToolStripMenuItem";
this.exitToolStripMenuItem.Size = new System.Drawing.Size(92, 22);
this.exitToolStripMenuItem.Text = "Exit";
//
// viewToolStripMenuItem
//
this.viewToolStripMenuItem.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] {
this.historyToolStripMenuItem,
this.internetMonitoringToolStripMenuItem,
this.databaseToolStripMenuItem});
this.viewToolStripMenuItem.Name = "viewToolStripMenuItem";
this.viewToolStripMenuItem.ShortcutKeyDisplayString = "";
this.viewToolStripMenuItem.Size = new System.Drawing.Size(44, 20);
this.viewToolStripMenuItem.Text = "View";
//
// historyToolStripMenuItem
//
```

```
this.historyToolStripMenuItem.Name = "historyToolStripMenuItem";
this.historyToolStripMenuItem.Size = new System.Drawing.Size(178, 22);
this.historyToolStripMenuItem.Text = "History (Log Files)";
this.historyToolStripMenuItem.Click += new System.EventHandler(this.historyToolStripMenuItem_Click);
//
// internetMonitoringToolStripMenuItem
//
this.internetMonitoringToolStripMenuItem.Name = "internetMonitoringToolStripMenuItem";
this.internetMonitoringToolStripMenuItem.Size = new System.Drawing.Size(178, 22);
this.internetMonitoringToolStripMenuItem.Text = "Internet Monitoring";
//
// databaseToolStripMenuItem
//
this.databaseToolStripMenuItem.Name = "databaseToolStripMenuItem";
this.databaseToolStripMenuItem.Size = new System.Drawing.Size(178, 22);
this.databaseToolStripMenuItem.Text = "Database";
//
// settingToolStripMenuItem
//
this.settingToolStripMenuItem.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] {
this.thresholdSettingToolStripMenuItem,
this.parametersSettingToolStripMenuItem,
this.toolStripSeparator2});
this.settingToolStripMenuItem.Name = "settingToolStripMenuItem";
this.settingToolStripMenuItem.Size = new System.Drawing.Size(56, 20);
this.settingToolStripMenuItem.Text = "Setting";
//
// thresholdSettingToolStripMenuItem
//
this.thresholdSettingToolStripMenuItem.Name = "thresholdSettingToolStripMenuItem";
this.thresholdSettingToolStripMenuItem.Size = new System.Drawing.Size(167, 22);
this.thresholdSettingToolStripMenuItem.Text = "Threshold Setting";
this.thresholdSettingToolStripMenuItem.Click += new
System.EventHandler(this.thresholdSettingToolStripMenuItem_Click);
//
// parametersSettingToolStripMenuItem
//
this.parametersSettingToolStripMenuItem.DropDownItems.AddRange(new
System.Windows.Forms.ToolStripItem[] {
this.periodRefreshNetworkToolStripMenuItem,
this.periodForceDataATISToolStripMenuItem});
this.parametersSettingToolStripMenuItem.Name = "parametersSettingToolStripMenuItem";
this.parametersSettingToolStripMenuItem.Size = new System.Drawing.Size(167, 22);
this.parametersSettingToolStripMenuItem.Text = "Timer Setting";
//
// periodRefreshNetworkToolStripMenuItem
//
this.periodRefreshNetworkToolStripMenuItem.Name = "periodRefreshNetworkToolStripMenuItem";
this.periodRefreshNetworkToolStripMenuItem.Size = new System.Drawing.Size(241, 22);
this.periodRefreshNetworkToolStripMenuItem.Text = "Period Network Refresh (ATND)";
this.periodRefreshNetworkToolStripMenuItem.Click += new
System.EventHandler(this.periodRefreshNetworkToolStripMenuItem_Click);
//
// periodForceDataATISToolStripMenuItem
//
this.periodForceDataATISToolStripMenuItem.Name = "periodForceDataATISToolStripMenuItem";
this.periodForceDataATISToolStripMenuItem.Size = new System.Drawing.Size(241, 22);
this.periodForceDataATISToolStripMenuItem.Text = "Period Force Data (ATIS)";
this.periodForceDataATISToolStripMenuItem.Click += new
System.EventHandler(this.periodForceDataATISToolStripMenuItem_Click_1);
```

```
//
// toolStripSeparator2
//
this.toolStripSeparator2.Name = "toolStripSeparator2";
this.toolStripSeparator2.Size = new System.Drawing.Size(164, 6);
//
// helpToolStripMenuItem
//
this.helpToolStripMenuItem.DropDownItems.AddRange(new System.Windows.Forms.ToolStripItem[] {
this.toolStripSeparator3,
this.aboutToolStripMenuItem});
this.helpToolStripMenuItem.Name = "helpToolStripMenuItem";
this.helpToolStripMenuItem.Size = new System.Drawing.Size(44, 20);
this.helpToolStripMenuItem.Text = "Help";
//
// toolStripSeparator3
//
this.toolStripSeparator3.Name = "toolStripSeparator3";
this.toolStripSeparator3.Size = new System.Drawing.Size(104, 6);
//
// aboutToolStripMenuItem
//
this.aboutToolStripMenuItem.Name = "aboutToolStripMenuItem";
this.aboutToolStripMenuItem.Size = new System.Drawing.Size(107, 22);
this.aboutToolStripMenuItem.Text = "About";
this.aboutToolStripMenuItem.Click += new System.EventHandler(this.aboutToolStripMenuItem_Click);
//
// statusStrip
//
this.statusStrip.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {
this.toolStripStatusLabel1});
this.statusStrip.Location = new System.Drawing.Point(0, 640);
this.statusStrip.MaximumSize = new System.Drawing.Size(1184, 22);
this.statusStrip.Name = "statusStrip";
this.statusStrip.Size = new System.Drawing.Size(1184, 22);
this.statusStrip.TabIndex = 159;
this.statusStrip.Text = "statusStrip1";
//
// toolStripStatusLabel1
//
this.toolStripStatusLabel1.Name = "toolStripStatusLabel1";
this.toolStripStatusLabel1.Size = new System.Drawing.Size(249, 17);
this.toolStripStatusLabel1.Text = "Zigbee Network Management 1.0 :: Working...";
//
// toolStrip
//
this.toolStrip.Items.AddRange(new System.Windows.Forms.ToolStripItem[] {
this.toolStripLabel1,
this.toolStripSeparator1});
this.toolStrip.Location = new System.Drawing.Point(0, 24);
this.toolStrip.Name = "toolStrip";
this.toolStrip.Size = new System.Drawing.Size(1184, 25);
this.toolStrip.TabIndex = 160;
this.toolStrip.Text = "toolStrip1";
//
// toolStripLabel1
//
this.toolStripLabel1.Name = "toolStripLabel1";
this.toolStripLabel1.Size = new System.Drawing.Size(98, 22);
this.toolStripLabel1.Text = "Quick Command";
```

```
//
// toolStripSeparator1
//
this.toolStripSeparator1.Name = "toolStripSeparator1";
this.toolStripSeparator1.Size = new System.Drawing.Size(6, 25);
//
// tabXbee
//
this.tabXbee.Controls.Add(this.tabOverview);
this.tabXbee.Controls.Add(this.tabRouter1);
this.tabXbee.Controls.Add(this.tabRouter2);
this.tabXbee.Controls.Add(this.tabRouter3);
this.tabXbee.Location = new System.Drawing.Point(9, 21);
this.tabXbee.Name = "tabXbee";
this.tabXbee.SelectedIndex = 0;
this.tabXbee.Size = new System.Drawing.Size(615, 197);
this.tabXbee.TabIndex = 161;
//
// tabOverview
//
this.tabOverview.Controls.Add(this.groupBox7);
this.tabOverview.Controls.Add(this.timeLabelRouter1);
this.tabOverview.Controls.Add(this.groupBox6);
this.tabOverview.Controls.Add(this.groupBox5);
this.tabOverview.Location = new System.Drawing.Point(4, 22);
this.tabOverview.Name = "tabOverview";
this.tabOverview.Size = new System.Drawing.Size(607, 171);
this.tabOverview.TabIndex = 3;
this.tabOverview.Text = "Overview Network";
this.tabOverview.UseVisualStyleBackColor = true;
//
// groupBox7
//
this.groupBox7.Controls.Add(this.panel5);
this.groupBox7.Controls.Add(this.panel4);
this.groupBox7.Controls.Add(this.panel6);
this.groupBox7.Controls.Add(this.panel3);
this.groupBox7.Controls.Add(this.label16);
this.groupBox7.Controls.Add(this.label14);
this.groupBox7.Controls.Add(this.label13);
this.groupBox7.Controls.Add(this.label12);
this.groupBox7.Controls.Add(this.label11);
this.groupBox7.Controls.Add(this.label10);
this.groupBox7.Controls.Add(this.panel2);
this.groupBox7.Location = new System.Drawing.Point(410, 4);
this.groupBox7.Name = "groupBox7";
this.groupBox7.Size = new System.Drawing.Size(190, 163);
this.groupBox7.TabIndex = 172;
this.groupBox7.TabStop = false;
this.groupBox7.Text = "I/P && O/P Status";
//
// panel5
//
this.panel5.BackColor = System.Drawing.Color.Black;
this.panel5.Controls.Add(this.router2out1);
this.panel5.Location = new System.Drawing.Point(146, 76);
this.panel5.Name = "panel5";
this.panel5.Size = new System.Drawing.Size(35, 20);
this.panel5.TabIndex = 10;
//
```

```
// router2out1
//
this.router2out1.AutoSize = true;
this.router2out1.BackColor = System.Drawing.Color.Transparent;
this.router2out1.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.router2out1.ForeColor = System.Drawing.Color.Yellow;
this.router2out1.Location = new System.Drawing.Point(1, 4);
this.router2out1.Name = "router2out1";
this.router2out1.Size = new System.Drawing.Size(27, 13);
this.router2out1.TabIndex = 9;
this.router2out1.Text = "N/A";
this.router2out1.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
//
// panel4
//
this.panel4.BackColor = System.Drawing.Color.Black;
this.panel4.Controls.Add(this.label17);
this.panel4.Controls.Add(this.router3in1);
this.panel4.Location = new System.Drawing.Point(44, 101);
this.panel4.Name = "panel4";
this.panel4.Size = new System.Drawing.Size(58, 20);
this.panel4.TabIndex = 8;
//
// label17
//
this.label17.AutoSize = true;
this.label17.BackColor = System.Drawing.Color.Transparent;
this.label17.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.label17.ForeColor = System.Drawing.Color.FromArgb(((int)(((byte)(224)))), ((int)(((byte)(224)))),
((int)(((byte)(224)))));
this.label17.Location = new System.Drawing.Point(35, 4);
this.label17.Name = "label17";
this.label17.Size = new System.Drawing.Size(20, 13);
this.label17.TabIndex = 7;
this.label17.Text = "°C";
//
// router3in1
//
this.router3in1.AutoSize = true;
this.router3in1.BackColor = System.Drawing.Color.Transparent;
this.router3in1.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.router3in1.ForeColor = System.Drawing.Color.Aqua;
this.router3in1.Location = new System.Drawing.Point(4, 4);
this.router3in1.Name = "router3in1";
this.router3in1.Size = new System.Drawing.Size(27, 13);
this.router3in1.TabIndex = 7;
this.router3in1.Text = "N/A";
this.router3in1.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
//
// panel6
//
this.panel6.BackColor = System.Drawing.Color.Black;
this.panel6.Controls.Add(this.router1out1);
this.panel6.Location = new System.Drawing.Point(146, 50);
this.panel6.Name = "panel6";
this.panel6.Size = new System.Drawing.Size(35, 20);
this.panel6.TabIndex = 9;
```

```
//
// router1out1
//
this.router1out1.AutoSize = true;
this.router1out1.BackColor = System.Drawing.Color.Transparent;
this.router1out1.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.router1out1.ForeColor = System.Drawing.Color.Aqua;
this.router1out1.Location = new System.Drawing.Point(1, 4);
this.router1out1.Name = "router1out1";
this.router1out1.Size = new System.Drawing.Size(27, 13);
this.router1out1.TabIndex = 7;
this.router1out1.Text = "N/A";
this.router1out1.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
//
// panel3
//
this.panel3.BackColor = System.Drawing.Color.Black;
this.panel3.Controls.Add(this.label18);
this.panel3.Controls.Add(this.router2in1);
this.panel3.Location = new System.Drawing.Point(44, 76);
this.panel3.Name = "panel3";
this.panel3.Size = new System.Drawing.Size(58, 20);
this.panel3.TabIndex = 8;
//
// label18
//
this.label18.AutoSize = true;
this.label18.BackColor = System.Drawing.Color.Transparent;
this.label18.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.label18.ForeColor = System.Drawing.Color.FromArgb(((int)(((byte)(224)))), ((int)(((byte)(224)))),
((int)(((byte)(224)))));
this.label18.Location = new System.Drawing.Point(38, 4);
this.label18.Name = "label18";
this.label18.Size = new System.Drawing.Size(16, 13);
this.label18.TabIndex = 8;
this.label18.Text = "%";
//
// router2in1
//
this.router2in1.AutoSize = true;
this.router2in1.BackColor = System.Drawing.Color.Transparent;
this.router2in1.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.router2in1.ForeColor = System.Drawing.Color.Yellow;
this.router2in1.Location = new System.Drawing.Point(3, 4);
this.router2in1.Name = "router2in1";
this.router2in1.Size = new System.Drawing.Size(27, 13);
this.router2in1.TabIndex = 7;
this.router2in1.Text = "N/A";
this.router2in1.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
//
// label16
//
this.label16.AutoSize = true;
this.label16.Location = new System.Drawing.Point(17, 53);
this.label16.Name = "label16";
this.label16.Size = new System.Drawing.Size(0, 13);
this.label16.TabIndex = 6;
```

```
//
// label14
//
this.label14.AutoSize = true;
this.label14.Location = new System.Drawing.Point(113, 79);
this.label14.Name = "label14";
this.label14.Size = new System.Drawing.Size(33, 13);
this.label14.TabIndex = 4;
this.label14.Text = "O/P :";
//
// label13
//
this.label13.AutoSize = true;
this.label13.Location = new System.Drawing.Point(113, 55);
this.label13.Name = "label13";
this.label13.Size = new System.Drawing.Size(33, 13);
this.label13.TabIndex = 3;
this.label13.Text = "O/P :";
//
// label12
//
this.label12.AutoSize = true;
this.label12.Location = new System.Drawing.Point(4, 104);
this.label12.Name = "label12";
this.label12.Size = new System.Drawing.Size(40, 13);
this.label12.TabIndex = 2;
this.label12.Text = "Temp :";
//
// label11
//
this.label11.AutoSize = true;
this.label11.Location = new System.Drawing.Point(7, 79);
this.label11.Name = "label11";
this.label11.Size = new System.Drawing.Size(37, 13);
this.label11.TabIndex = 1;
this.label11.Text = "%RH :";
//
// label10
//
this.label10.AutoSize = true;
this.label10.Location = new System.Drawing.Point(4, 53);
this.label10.Name = "label10";
this.label10.Size = new System.Drawing.Size(40, 13);
this.label10.TabIndex = 0;
this.label10.Text = "Temp :";
//
// panel2
//
this.panel2.BackColor = System.Drawing.Color.Black;
this.panel2.Controls.Add(this.router1in1);
this.panel2.Controls.Add(this.label15);
this.panel2.Location = new System.Drawing.Point(44, 50);
this.panel2.Name = "panel2";
this.panel2.Size = new System.Drawing.Size(58, 20);
this.panel2.TabIndex = 7;
//
// router1in1
//
this.router1in1.AutoSize = true;
this.router1in1.BackColor = System.Drawing.Color.Transparent;
```

```
this.router1in1.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.router1in1.ForeColor = System.Drawing.Color.Aqua;
this.router1in1.Location = new System.Drawing.Point(3, 4);
this.router1in1.Name = "router1in1";
this.router1in1.Size = new System.Drawing.Size(27, 13);
this.router1in1.TabIndex = 6;
this.router1in1.Text = "N/A";
this.router1in1.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
//
// label15
//
this.label15.AutoSize = true;
this.label15.BackColor = System.Drawing.Color.Transparent;
this.label15.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.label15.ForeColor = System.Drawing.Color.FromArgb(((int)(((byte)(224)))), ((int)(((byte)(224)))),
((int)(((byte)(224)))));
this.label15.Location = new System.Drawing.Point(35, 4);
this.label15.Name = "label15";
this.label15.Size = new System.Drawing.Size(20, 13);
this.label15.TabIndex = 5;
this.label15.Text = "°C";
//
// timeLabelRouter1
//
this.timeLabelRouter1.AutoSize = true;
this.timeLabelRouter1.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.timeLabelRouter1.ForeColor = System.Drawing.Color.DimGray;
this.timeLabelRouter1.Location = new System.Drawing.Point(292, 58);
this.timeLabelRouter1.Name = "timeLabelRouter1";
this.timeLabelRouter1.Size = new System.Drawing.Size(52, 13);
this.timeLabelRouter1.TabIndex = 171;
this.timeLabelRouter1.Text = "Waiting...";
//
// groupBox6
//
this.groupBox6.Controls.Add(this.timeLabelRouter3);
this.groupBox6.Controls.Add(this.timeLabelRouter2);
this.groupBox6.Controls.Add(this.timeLabelCoordinator);
this.groupBox6.Location = new System.Drawing.Point(286, 4);
this.groupBox6.Name = "groupBox6";
this.groupBox6.Size = new System.Drawing.Size(120, 163);
this.groupBox6.TabIndex = 1;
this.groupBox6.TabStop = false;
this.groupBox6.Text = "Last Seen";
//
// timeLabelRouter3
//
this.timeLabelRouter3.AutoSize = true;
this.timeLabelRouter3.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.timeLabelRouter3.ForeColor = System.Drawing.Color.DimGray;
this.timeLabelRouter3.Location = new System.Drawing.Point(6, 104);
this.timeLabelRouter3.Name = "timeLabelRouter3";
this.timeLabelRouter3.Size = new System.Drawing.Size(52, 13);
this.timeLabelRouter3.TabIndex = 173;
this.timeLabelRouter3.Text = "Waiting...";
//
```

```
// timeLabelRouter2
//
this.timeLabelRouter2.AutoSize = true;
this.timeLabelRouter2.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.timeLabelRouter2.ForeColor = System.Drawing.Color.DimGray;
this.timeLabelRouter2.Location = new System.Drawing.Point(6, 79);
this.timeLabelRouter2.Name = "timeLabelRouter2";
this.timeLabelRouter2.Size = new System.Drawing.Size(52, 13);
this.timeLabelRouter2.TabIndex = 172;
this.timeLabelRouter2.Text = "Waiting...";
//
// timeLabelCoordinator
//
this.timeLabelCoordinator.AutoSize = true;
this.timeLabelCoordinator.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.timeLabelCoordinator.ForeColor = System.Drawing.Color.DimGray;
this.timeLabelCoordinator.Location = new System.Drawing.Point(6, 30);
this.timeLabelCoordinator.Name = "timeLabelCoordinator";
this.timeLabelCoordinator.Size = new System.Drawing.Size(52, 13);
this.timeLabelCoordinator.TabIndex = 170;
this.timeLabelCoordinator.Text = "Waiting...";
//
// groupBox5
//
this.groupBox5.Controls.Add(this.refreshTimer);
this.groupBox5.Controls.Add(this.timeLabel);
this.groupBox5.Controls.Add(this.textlabel10);
this.groupBox5.Controls.Add(this.imgStatusRouter3);
this.groupBox5.Controls.Add(this.imgStatusRouter2);
this.groupBox5.Controls.Add(this.imgStatusRouter1);
this.groupBox5.Controls.Add(this.imgStatusCoordinator);
this.groupBox5.Controls.Add(this.coordinatorLabel);
this.groupBox5.Controls.Add(this.routerLabel3);
this.groupBox5.Controls.Add(this.routerLabel2);
this.groupBox5.Controls.Add(this.routerLabel1);
this.groupBox5.Controls.Add(this.labelSecond);
this.groupBox5.Location = new System.Drawing.Point(6, 5);
this.groupBox5.Name = "groupBox5";
this.groupBox5.Size = new System.Drawing.Size(276, 162);
this.groupBox5.TabIndex = 0;
this.groupBox5.TabStop = false;
this.groupBox5.Text = "Online Status";
//
// refreshTimer
//
this.refreshTimer.Font = new System.Drawing.Font("Microsoft Sans Serif", 7F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.refreshTimer.Location = new System.Drawing.Point(192, 135);
this.refreshTimer.Name = "refreshTimer";
this.refreshTimer.Size = new System.Drawing.Size(80, 22);
this.refreshTimer.TabIndex = 170;
this.refreshTimer.Text = "Refresh Timer";
this.refreshTimer.UseVisualStyleBackColor = true;
this.refreshTimer.Click += new System.EventHandler(this.refreshTimer_Click);
//
// timeLabel
//
this.timeLabel.AutoSize = true;
```

```
this.timeLabel.Font = new System.Drawing.Font("Microsoft Sans Serif", 7F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.timeLabel.ForeColor = System.Drawing.Color.Gray;
this.timeLabel.Location = new System.Drawing.Point(64, 141);
this.timeLabel.Name = "timeLabel";
this.timeLabel.Size = new System.Drawing.Size(22, 13);
this.timeLabel.TabIndex = 168;
this.timeLabel.Text = "-----";
this.timeLabel.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
//
// textlabel10
//
this.textlabel10.AutoSize = true;
this.textlabel10.Font = new System.Drawing.Font("Microsoft Sans Serif", 7F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.textlabel10.ForeColor = System.Drawing.Color.Silver;
this.textlabel10.Location = new System.Drawing.Point(5, 141);
this.textlabel10.Name = "textlabel10";
this.textlabel10.Size = new System.Drawing.Size(60, 13);
this.textlabel10.TabIndex = 167;
this.textlabel10.Text = "Next Check";
//
// imgStatusRouter3
//
this.imgStatusRouter3.Image = ((System.Drawing.Image)(resources.GetObject("imgStatusRouter3.Image")));
this.imgStatusRouter3.Location = new System.Drawing.Point(13, 100);
this.imgStatusRouter3.Name = "imgStatusRouter3";
this.imgStatusRouter3.Size = new System.Drawing.Size(16, 16);
this.imgStatusRouter3.TabIndex = 166;
this.imgStatusRouter3.TabStop = false;
//
// imgStatusRouter2
//
this.imgStatusRouter2.Image = ((System.Drawing.Image)(resources.GetObject("imgStatusRouter2.Image")));
this.imgStatusRouter2.Location = new System.Drawing.Point(13, 75);
this.imgStatusRouter2.Name = "imgStatusRouter2";
this.imgStatusRouter2.Size = new System.Drawing.Size(16, 16);
this.imgStatusRouter2.TabIndex = 165;
this.imgStatusRouter2.TabStop = false;
//
// imgStatusRouter1
//
this.imgStatusRouter1.Image = ((System.Drawing.Image)(resources.GetObject("imgStatusRouter1.Image")));
this.imgStatusRouter1.Location = new System.Drawing.Point(13, 50);
this.imgStatusRouter1.Name = "imgStatusRouter1";
this.imgStatusRouter1.Size = new System.Drawing.Size(16, 16);
this.imgStatusRouter1.TabIndex = 164;
this.imgStatusRouter1.TabStop = false;
//
// imgStatusCoordinator
//
this.imgStatusCoordinator.Image =
((System.Drawing.Image)(resources.GetObject("imgStatusCoordinator.Image")));
this.imgStatusCoordinator.Location = new System.Drawing.Point(13, 26);
this.imgStatusCoordinator.Name = "imgStatusCoordinator";
this.imgStatusCoordinator.Size = new System.Drawing.Size(16, 16);
this.imgStatusCoordinator.TabIndex = 163;
this.imgStatusCoordinator.TabStop = false;
//
// coordinatorLabel
```

```
//
this.coordinatorLabel.AutoSize = true;
this.coordinatorLabel.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.coordinatorLabel.ForeColor = System.Drawing.Color.Red;
this.coordinatorLabel.Location = new System.Drawing.Point(32, 29);
this.coordinatorLabel.Name = "coordinatorLabel";
this.coordinatorLabel.Size = new System.Drawing.Size(238, 13);
this.coordinatorLabel.TabIndex = 3;
this.coordinatorLabel.Text = "Coordinator [MAC: 0013A200 403DB431]";
//
// routerLabel3
//
this.routerLabel3.AutoSize = true;
this.routerLabel3.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.routerLabel3.ForeColor = System.Drawing.Color.Red;
this.routerLabel3.Location = new System.Drawing.Point(32, 103);
this.routerLabel3.Name = "routerLabel3";
this.routerLabel3.Size = new System.Drawing.Size(220, 13);
this.routerLabel3.TabIndex = 2;
this.routerLabel3.Text = "Router3 [MAC: 0013A200 403DC71D]";
//
// routerLabel2
//
this.routerLabel2.AutoSize = true;
this.routerLabel2.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.routerLabel2.ForeColor = System.Drawing.Color.Red;
this.routerLabel2.Location = new System.Drawing.Point(32, 78);
this.routerLabel2.Name = "routerLabel2";
this.routerLabel2.Size = new System.Drawing.Size(220, 13);
this.routerLabel2.TabIndex = 1;
this.routerLabel2.Text = "Router2 [MAC: 0013A200 403DC77D]";
//
// routerLabel1
//
this.routerLabel1.AutoSize = true;
this.routerLabel1.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.routerLabel1.ForeColor = System.Drawing.Color.Red;
this.routerLabel1.Location = new System.Drawing.Point(32, 53);
this.routerLabel1.Name = "routerLabel1";
this.routerLabel1.Size = new System.Drawing.Size(218, 13);
this.routerLabel1.TabIndex = 0;
this.routerLabel1.Text = "Router1 [MAC: 0013A200 4055B1CC]";
//
// labelSecond
//
this.labelSecond.AutoSize = true;
this.labelSecond.Font = new System.Drawing.Font("Microsoft Sans Serif", 7F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.labelSecond.ForeColor = System.Drawing.Color.Silver;
this.labelSecond.Location = new System.Drawing.Point(87, 141);
this.labelSecond.Name = "labelSecond";
this.labelSecond.Size = new System.Drawing.Size(46, 13);
this.labelSecond.TabIndex = 169;
this.labelSecond.Text = "seconds";
//
// tabRouter1
```

```
//
this.tabRouter1.Controls.Add(this.groupBox11);
this.tabRouter1.Controls.Add(this.groupBox10);
this.tabRouter1.Controls.Add(this.groupBox9);
this.tabRouter1.Controls.Add(this.groupBox8);
this.tabRouter1.Location = new System.Drawing.Point(4, 22);
this.tabRouter1.Name = "tabRouter1";
this.tabRouter1.Size = new System.Drawing.Size(607, 171);
this.tabRouter1.TabIndex = 0;
this.tabRouter1.Text = "Router1";
this.tabRouter1.UseVisualStyleBackColor = true;
//
// groupBox11
//
this.groupBox11.Controls.Add(this.statRouter1);
this.groupBox11.Location = new System.Drawing.Point(5, 123);
this.groupBox11.Name = "groupBox11";
this.groupBox11.Size = new System.Drawing.Size(96, 45);
this.groupBox11.TabIndex = 3;
this.groupBox11.TabStop = false;
this.groupBox11.Text = "Status";
//
// statRouter1
//
this.statRouter1.AutoSize = true;
this.statRouter1.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.statRouter1.ForeColor = System.Drawing.Color.Red;
this.statRouter1.Location = new System.Drawing.Point(17, 21);
this.statRouter1.Name = "statRouter1";
this.statRouter1.Size = new System.Drawing.Size(44, 13);
this.statRouter1.TabIndex = 0;
this.statRouter1.Text = "Offline";
//
// groupBox10
//
this.groupBox10.Controls.Add(this.thresholdRouter1);
this.groupBox10.Controls.Add(this.label2);
this.groupBox10.Location = new System.Drawing.Point(336, 123);
this.groupBox10.Name = "groupBox10";
this.groupBox10.Size = new System.Drawing.Size(263, 45);
this.groupBox10.TabIndex = 2;
this.groupBox10.TabStop = false;
this.groupBox10.Text = "Threshold Setting";
//
// thresholdRouter1
//
this.thresholdRouter1.AutoSize = true;
this.thresholdRouter1.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.thresholdRouter1.ForeColor = System.Drawing.Color.Red;
this.thresholdRouter1.Location = new System.Drawing.Point(164, 17);
this.thresholdRouter1.Name = "thresholdRouter1";
this.thresholdRouter1.Size = new System.Drawing.Size(41, 20);
this.thresholdRouter1.TabIndex = 188;
this.thresholdRouter1.Text = "- - - -";
//
// label2
//
this.label2.AutoSize = true;
```

```
this.label2.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.label2.Location = new System.Drawing.Point(31, 21);
this.label2.Name = "label2";
this.label2.Size = new System.Drawing.Size(131, 13);
this.label2.TabIndex = 0;
this.label2.Text = "Threshold Temp (°C) :";
//
// groupBox9
//
this.groupBox9.Controls.Add(this.btRouter1Off1);
this.groupBox9.Controls.Add(this.btRouter1On1);
this.groupBox9.Location = new System.Drawing.Point(108, 123);
this.groupBox9.Name = "groupBox9";
this.groupBox9.Size = new System.Drawing.Size(221, 45);
this.groupBox9.TabIndex = 1;
this.groupBox9.TabStop = false;
this.groupBox9.Text = "Manual Control";
//
// btRouter1Off1
//
this.btRouter1Off1.Location = new System.Drawing.Point(114, 16);
this.btRouter1Off1.Name = "btRouter1Off1";
this.btRouter1Off1.Size = new System.Drawing.Size(100, 23);
this.btRouter1Off1.TabIndex = 1;
this.btRouter1Off1.Text = "Power Off O/P:1";
this.btRouter1Off1.UseVisualStyleBackColor = true;
this.btRouter1Off1.Click += new System.EventHandler(this.btRouter1Off1_Click);
//
// btRouter1On1
//
this.btRouter1On1.Location = new System.Drawing.Point(8, 16);
this.btRouter1On1.Name = "btRouter1On1";
this.btRouter1On1.Size = new System.Drawing.Size(100, 23);
this.btRouter1On1.TabIndex = 0;
this.btRouter1On1.Text = "Power On O/P:1";
this.btRouter1On1.UseVisualStyleBackColor = true;
this.btRouter1On1.Click += new System.EventHandler(this.btRouter1On1_Click);
//
// groupBox8
//
this.groupBox8.Controls.Add(this.label49);
this.groupBox8.Controls.Add(this.label48);
this.groupBox8.Controls.Add(this.label47);
this.groupBox8.Controls.Add(this.label46);
this.groupBox8.Controls.Add(this.outputRouter1);
this.groupBox8.Controls.Add(this.label23);
this.groupBox8.Controls.Add(this.tempRouter1);
this.groupBox8.Controls.Add(this.analogRouter1);
this.groupBox8.Controls.Add(this.rawRouter1);
this.groupBox8.Controls.Add(this.label22);
this.groupBox8.Controls.Add(this.label21);
this.groupBox8.Controls.Add(this.label20);
this.groupBox8.Controls.Add(this.pictureBox2);
this.groupBox8.Controls.Add(this.pictureBox1);
this.groupBox8.Controls.Add(this.label19);
this.groupBox8.Controls.Add(this.timeConfRouter1);
this.groupBox8.Location = new System.Drawing.Point(5, 5);
this.groupBox8.Name = "groupBox8";
this.groupBox8.Size = new System.Drawing.Size(594, 115);
```

```
this.groupBox8.TabIndex = 0;
this.groupBox8.TabStop = false;
this.groupBox8.Text = "Lastest Data";
//
// label49
//
this.label49.AutoSize = true;
this.label49.ForeColor = System.Drawing.Color.Blue;
this.label49.Location = new System.Drawing.Point(133, 90);
this.label49.Name = "label49";
this.label49.Size = new System.Drawing.Size(28, 13);
this.label49.TabIndex = 187;
this.label49.Text = "FAN";
//
// label48
//
this.label48.AutoSize = true;
this.label48.ForeColor = System.Drawing.Color.Blue;
this.label48.Location = new System.Drawing.Point(77, 71);
this.label48.Name = "label48";
this.label48.Size = new System.Drawing.Size(110, 13);
this.label48.TabIndex = 186;
this.label48.Text = "0013A200 4055B1CC";
//
// label47
//
this.label47.AutoSize = true;
this.label47.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.label47.Location = new System.Drawing.Point(36, 90);
this.label47.Name = "label47";
this.label47.Size = new System.Drawing.Size(97, 13);
this.label47.TabIndex = 185;
this.label47.Text = "Output Control :";
//
// label46
//
this.label46.AutoSize = true;
this.label46.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.label46.Location = new System.Drawing.Point(36, 71);
this.label46.Name = "label46";
this.label46.Size = new System.Drawing.Size(41, 13);
this.label46.TabIndex = 184;
this.label46.Text = "MAC :";
//
// outputRouter1
//
this.outputRouter1.AutoSize = true;
this.outputRouter1.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.outputRouter1.ForeColor = System.Drawing.Color.DarkGreen;
this.outputRouter1.Location = new System.Drawing.Point(495, 83);
this.outputRouter1.Name = "outputRouter1";
this.outputRouter1.Size = new System.Drawing.Size(74, 20);
this.outputRouter1.TabIndex = 183;
this.outputRouter1.Text = "Waiting...";
//
// label23
//
```

```
this.label23.AutoSize = true;
this.label23.Location = new System.Drawing.Point(496, 67);
this.label23.Name = "label23";
this.label23.Size = new System.Drawing.Size(42, 13);
this.label23.TabIndex = 182;
this.label23.Text = "O/P 1 :";
//
// tempRouter1
//
this.tempRouter1.AutoSize = true;
this.tempRouter1.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.tempRouter1.ForeColor = System.Drawing.Color.Red;
this.tempRouter1.Location = new System.Drawing.Point(495, 40);
this.tempRouter1.Name = "tempRouter1";
this.tempRouter1.Size = new System.Drawing.Size(74, 20);
this.tempRouter1.TabIndex = 181;
this.tempRouter1.Text = "Waiting...";
//
// analogRouter1
//
this.analogRouter1.AutoSize = true;
this.analogRouter1.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.analogRouter1.ForeColor = System.Drawing.Color.Red;
this.analogRouter1.Location = new System.Drawing.Point(361, 40);
this.analogRouter1.Name = "analogRouter1";
this.analogRouter1.Size = new System.Drawing.Size(74, 20);
this.analogRouter1.TabIndex = 180;
this.analogRouter1.Text = "Waiting...";
//
// rawRouter1
//
this.rawRouter1.AutoSize = true;
this.rawRouter1.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.rawRouter1.ForeColor = System.Drawing.Color.Red;
this.rawRouter1.Location = new System.Drawing.Point(253, 40);
this.rawRouter1.Name = "rawRouter1";
this.rawRouter1.Size = new System.Drawing.Size(74, 20);
this.rawRouter1.TabIndex = 179;
this.rawRouter1.Text = "Waiting...";
//
// label22
//
this.label22.AutoSize = true;
this.label22.Location = new System.Drawing.Point(496, 24);
this.label22.Name = "label22";
this.label22.Size = new System.Drawing.Size(60, 13);
this.label22.TabIndex = 178;
this.label22.Text = "Temp (°C) :";
//
// label21
//
this.label21.AutoSize = true;
this.label21.Location = new System.Drawing.Point(362, 24);
this.label21.Name = "label21";
this.label21.Size = new System.Drawing.Size(109, 13);
this.label21.TabIndex = 177;
this.label21.Text = "Analog Voltage (mV) :";
```

```
//
// label20
//
this.label20.AutoSize = true;
this.label20.Location = new System.Drawing.Point(254, 24);
this.label20.Name = "label20";
this.label20.Size = new System.Drawing.Size(61, 13);
this.label20.TabIndex = 176;
this.label20.Text = "Raw Data :";
//
// pictureBox2
//
this.pictureBox2.Image = ((System.Drawing.Image)(resources.GetObject("pictureBox2.Image")));
this.pictureBox2.Location = new System.Drawing.Point(227, 17);
this.pictureBox2.Name = "pictureBox2";
this.pictureBox2.Size = new System.Drawing.Size(24, 24);
this.pictureBox2.TabIndex = 175;
this.pictureBox2.TabStop = false;
//
// pictureBox1
//
this.pictureBox1.Image = ((System.Drawing.Image)(resources.GetObject("pictureBox1.Image")));
this.pictureBox1.Location = new System.Drawing.Point(8, 17);
this.pictureBox1.Name = "pictureBox1";
this.pictureBox1.Size = new System.Drawing.Size(24, 24);
this.pictureBox1.TabIndex = 174;
this.pictureBox1.TabStop = false;
//
// label19
//
this.label19.AutoSize = true;
this.label19.Location = new System.Drawing.Point(35, 24);
this.label19.Name = "label19";
this.label19.Size = new System.Drawing.Size(61, 13);
this.label19.TabIndex = 173;
this.label19.Text = "Last Seen :";
//
// timeConfRouter1
//
this.timeConfRouter1.AutoSize = true;
this.timeConfRouter1.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.timeConfRouter1.ForeColor = System.Drawing.Color.RoyalBlue;
this.timeConfRouter1.Location = new System.Drawing.Point(35, 40);
this.timeConfRouter1.Name = "timeConfRouter1";
this.timeConfRouter1.Size = new System.Drawing.Size(74, 20);
this.timeConfRouter1.TabIndex = 172;
this.timeConfRouter1.Text = "Waiting...";
//
// tabRouter2
//
this.tabRouter2.Controls.Add(this.groupBox12);
this.tabRouter2.Controls.Add(this.groupBox13);
this.tabRouter2.Controls.Add(this.groupBox14);
this.tabRouter2.Controls.Add(this.groupBox15);
this.tabRouter2.Location = new System.Drawing.Point(4, 22);
this.tabRouter2.Name = "tabRouter2";
this.tabRouter2.Padding = new System.Windows.Forms.Padding(3);
this.tabRouter2.Size = new System.Drawing.Size(607, 171);
this.tabRouter2.TabIndex = 1;
```

```
this.tabRouter2.Text = "Router2";
this.tabRouter2.UseVisualStyleBackColor = true;
//
// groupBox12
//
this.groupBox12.Controls.Add(this.statRouter2);
this.groupBox12.Location = new System.Drawing.Point(5, 123);
this.groupBox12.Name = "groupBox12";
this.groupBox12.Size = new System.Drawing.Size(96, 45);
this.groupBox12.TabIndex = 7;
this.groupBox12.TabStop = false;
this.groupBox12.Text = "Status";
//
// statRouter2
//
this.statRouter2.AutoSize = true;
this.statRouter2.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.statRouter2.ForeColor = System.Drawing.Color.Red;
this.statRouter2.Location = new System.Drawing.Point(17, 21);
this.statRouter2.Name = "statRouter2";
this.statRouter2.Size = new System.Drawing.Size(44, 13);
this.statRouter2.TabIndex = 0;
this.statRouter2.Text = "Offline";
//
// groupBox13
//
this.groupBox13.Controls.Add(this.thresholdRouter2);
this.groupBox13.Controls.Add(this.label4);
this.groupBox13.Location = new System.Drawing.Point(336, 123);
this.groupBox13.Name = "groupBox13";
this.groupBox13.Size = new System.Drawing.Size(263, 45);
this.groupBox13.TabIndex = 6;
this.groupBox13.TabStop = false;
this.groupBox13.Text = "Threshold Setting";
//
// thresholdRouter2
//
this.thresholdRouter2.AutoSize = true;
this.thresholdRouter2.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.thresholdRouter2.ForeColor = System.Drawing.Color.Red;
this.thresholdRouter2.Location = new System.Drawing.Point(164, 17);
this.thresholdRouter2.Name = "thresholdRouter2";
this.thresholdRouter2.Size = new System.Drawing.Size(41, 20);
this.thresholdRouter2.TabIndex = 190;
this.thresholdRouter2.Text = "- - - -";
//
// label4
//
this.label4.AutoSize = true;
this.label4.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.label4.Location = new System.Drawing.Point(31, 21);
this.label4.Name = "label4";
this.label4.Size = new System.Drawing.Size(110, 13);
this.label4.TabIndex = 189;
this.label4.Text = "Threshold (%RH) :";
//
// groupBox14
```

```
//
this.groupBox14.Controls.Add(this.btRouter2Off1);
this.groupBox14.Controls.Add(this.btRouter2On1);
this.groupBox14.Location = new System.Drawing.Point(108, 123);
this.groupBox14.Name = "groupBox14";
this.groupBox14.Size = new System.Drawing.Size(221, 45);
this.groupBox14.TabIndex = 5;
this.groupBox14.TabStop = false;
this.groupBox14.Text = "Manual Control";
//
// btRouter2Off1
//
this.btRouter2Off1.Location = new System.Drawing.Point(114, 16);
this.btRouter2Off1.Name = "btRouter2Off1";
this.btRouter2Off1.Size = new System.Drawing.Size(100, 23);
this.btRouter2Off1.TabIndex = 1;
this.btRouter2Off1.Text = "Power Off O/P:1";
this.btRouter2Off1.UseVisualStyleBackColor = true;
this.btRouter2Off1.Click += new System.EventHandler(this.btRouter2Off1_Click);
//
// btRouter2On1
//
this.btRouter2On1.Location = new System.Drawing.Point(8, 16);
this.btRouter2On1.Name = "btRouter2On1";
this.btRouter2On1.Size = new System.Drawing.Size(100, 23);
this.btRouter2On1.TabIndex = 0;
this.btRouter2On1.Text = "Power On O/P:1";
this.btRouter2On1.UseVisualStyleBackColor = true;
this.btRouter2On1.Click += new System.EventHandler(this.btRouter2On1_Click);
//
// groupBox15
//
this.groupBox15.Controls.Add(this.label36);
this.groupBox15.Controls.Add(this.label37);
this.groupBox15.Controls.Add(this.label38);
this.groupBox15.Controls.Add(this.label39);
this.groupBox15.Controls.Add(this.outputRouter2);
this.groupBox15.Controls.Add(this.label41);
this.groupBox15.Controls.Add(this.tempRouter2);
this.groupBox15.Controls.Add(this.analogRouter2);
this.groupBox15.Controls.Add(this.rawRouter2);
this.groupBox15.Controls.Add(this.label45);
this.groupBox15.Controls.Add(this.label50);
this.groupBox15.Controls.Add(this.label51);
this.groupBox15.Controls.Add(this.pictureBox5);
this.groupBox15.Controls.Add(this.pictureBox6);
this.groupBox15.Controls.Add(this.label52);
this.groupBox15.Controls.Add(this.timeConfRouter2);
this.groupBox15.Location = new System.Drawing.Point(5, 5);
this.groupBox15.Name = "groupBox15";
this.groupBox15.Size = new System.Drawing.Size(594, 115);
this.groupBox15.TabIndex = 4;
this.groupBox15.TabStop = false;
this.groupBox15.Text = "Lastest Data";
//
// label36
//
this.label36.AutoSize = true;
this.label36.ForeColor = System.Drawing.Color.Blue;
this.label36.Location = new System.Drawing.Point(133, 90);
```

```
this.label36.Name = "label36";
this.label36.Size = new System.Drawing.Size(83, 13);
this.label36.TabIndex = 187;
this.label36.Text = "RELAY BOARD";
//
// label37
//
this.label37.AutoSize = true;
this.label37.ForeColor = System.Drawing.Color.Blue;
this.label37.Location = new System.Drawing.Point(77, 71);
this.label37.Name = "label37";
this.label37.Size = new System.Drawing.Size(112, 13);
this.label37.TabIndex = 186;
this.label37.Text = "0013A200 403DC77D";
//
// label38
//
this.label38.AutoSize = true;
this.label38.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.label38.Location = new System.Drawing.Point(36, 90);
this.label38.Name = "label38";
this.label38.Size = new System.Drawing.Size(97, 13);
this.label38.TabIndex = 185;
this.label38.Text = "Output Control :";
//
// label39
//
this.label39.AutoSize = true;
this.label39.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.label39.Location = new System.Drawing.Point(36, 71);
this.label39.Name = "label39";
this.label39.Size = new System.Drawing.Size(41, 13);
this.label39.TabIndex = 184;
this.label39.Text = "MAC :";
//
// outputRouter2
//
this.outputRouter2.AutoSize = true;
this.outputRouter2.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.outputRouter2.ForeColor = System.Drawing.Color.DarkGreen;
this.outputRouter2.Location = new System.Drawing.Point(495, 83);
this.outputRouter2.Name = "outputRouter2";
this.outputRouter2.Size = new System.Drawing.Size(74, 20);
this.outputRouter2.TabIndex = 183;
this.outputRouter2.Text = "Waiting...";
//
// label41
//
this.label41.AutoSize = true;
this.label41.Location = new System.Drawing.Point(496, 67);
this.label41.Name = "label41";
this.label41.Size = new System.Drawing.Size(42, 13);
this.label41.TabIndex = 182;
this.label41.Text = "O/P 1 :";
//
// tempRouter2
//
```

```
this.tempRouter2.AutoSize = true;
this.tempRouter2.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.tempRouter2.ForeColor = System.Drawing.Color.Red;
this.tempRouter2.Location = new System.Drawing.Point(495, 40);
this.tempRouter2.Name = "tempRouter2";
this.tempRouter2.Size = new System.Drawing.Size(74, 20);
this.tempRouter2.TabIndex = 181;
this.tempRouter2.Text = "Waiting...";
//
// analogRouter2
//
this.analogRouter2.AutoSize = true;
this.analogRouter2.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.analogRouter2.ForeColor = System.Drawing.Color.Red;
this.analogRouter2.Location = new System.Drawing.Point(361, 40);
this.analogRouter2.Name = "analogRouter2";
this.analogRouter2.Size = new System.Drawing.Size(74, 20);
this.analogRouter2.TabIndex = 180;
this.analogRouter2.Text = "Waiting...";
//
// rawRouter2
//
this.rawRouter2.AutoSize = true;
this.rawRouter2.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.rawRouter2.ForeColor = System.Drawing.Color.Red;
this.rawRouter2.Location = new System.Drawing.Point(253, 40);
this.rawRouter2.Name = "rawRouter2";
this.rawRouter2.Size = new System.Drawing.Size(74, 20);
this.rawRouter2.TabIndex = 179;
this.rawRouter2.Text = "Waiting...";
//
// label45
//
this.label45.AutoSize = true;
this.label45.Location = new System.Drawing.Point(496, 24);
this.label45.Name = "label45";
this.label45.Size = new System.Drawing.Size(86, 13);
this.label45.TabIndex = 178;
this.label45.Text = "Humidity (%RH) :";
//
// label50
//
this.label50.AutoSize = true;
this.label50.Location = new System.Drawing.Point(362, 24);
this.label50.Name = "label50";
this.label50.Size = new System.Drawing.Size(101, 13);
this.label50.TabIndex = 177;
this.label50.Text = "Analog Voltage (V) :";
//
// label51
//
this.label51.AutoSize = true;
this.label51.Location = new System.Drawing.Point(254, 24);
this.label51.Name = "label51";
this.label51.Size = new System.Drawing.Size(61, 13);
this.label51.TabIndex = 176;
this.label51.Text = "Raw Data :";
```

```
//
// pictureBox5
//
this.pictureBox5.Image = ((System.Drawing.Image)(resources.GetObject("pictureBox5.Image")));
this.pictureBox5.Location = new System.Drawing.Point(227, 17);
this.pictureBox5.Name = "pictureBox5";
this.pictureBox5.Size = new System.Drawing.Size(24, 24);
this.pictureBox5.TabIndex = 175;
this.pictureBox5.TabStop = false;
//
// pictureBox6
//
this.pictureBox6.Image = ((System.Drawing.Image)(resources.GetObject("pictureBox6.Image")));
this.pictureBox6.Location = new System.Drawing.Point(8, 17);
this.pictureBox6.Name = "pictureBox6";
this.pictureBox6.Size = new System.Drawing.Size(24, 24);
this.pictureBox6.TabIndex = 174;
this.pictureBox6.TabStop = false;
//
// label52
//
this.label52.AutoSize = true;
this.label52.Location = new System.Drawing.Point(35, 24);
this.label52.Name = "label52";
this.label52.Size = new System.Drawing.Size(61, 13);
this.label52.TabIndex = 173;
this.label52.Text = "Last Seen :";
//
// timeConfRouter2
//
this.timeConfRouter2.AutoSize = true;
this.timeConfRouter2.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.timeConfRouter2.ForeColor = System.Drawing.Color.RoyalBlue;
this.timeConfRouter2.Location = new System.Drawing.Point(35, 40);
this.timeConfRouter2.Name = "timeConfRouter2";
this.timeConfRouter2.Size = new System.Drawing.Size(74, 20);
this.timeConfRouter2.TabIndex = 172;
this.timeConfRouter2.Text = "Waiting...";
//
// tabRouter3
//
this.tabRouter3.Controls.Add(this.groupBox16);
this.tabRouter3.Controls.Add(this.groupBox19);
this.tabRouter3.Location = new System.Drawing.Point(4, 22);
this.tabRouter3.Name = "tabRouter3";
this.tabRouter3.Padding = new System.Windows.Forms.Padding(3);
this.tabRouter3.Size = new System.Drawing.Size(607, 171);
this.tabRouter3.TabIndex = 2;
this.tabRouter3.Text = "Router3";
this.tabRouter3.UseVisualStyleBackColor = true;
//
// groupBox16
//
this.groupBox16.Controls.Add(this.statRouter3);
this.groupBox16.Location = new System.Drawing.Point(5, 123);
this.groupBox16.Name = "groupBox16";
this.groupBox16.Size = new System.Drawing.Size(96, 45);
this.groupBox16.TabIndex = 11;
this.groupBox16.TabStop = false;
```

```
this.groupBox16.Text = "Status";
//
// statRouter3
//
this.statRouter3.AutoSize = true;
this.statRouter3.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.statRouter3.ForeColor = System.Drawing.Color.Red;
this.statRouter3.Location = new System.Drawing.Point(17, 21);
this.statRouter3.Name = "statRouter3";
this.statRouter3.Size = new System.Drawing.Size(44, 13);
this.statRouter3.TabIndex = 0;
this.statRouter3.Text = "Offline";
//
// groupBox19
//
this.groupBox19.Controls.Add(this.label40);
this.groupBox19.Controls.Add(this.label42);
this.groupBox19.Controls.Add(this.label43);
this.groupBox19.Controls.Add(this.label44);
this.groupBox19.Controls.Add(this.tempRouter3);
this.groupBox19.Controls.Add(this.analogRouter3);
this.groupBox19.Controls.Add(this.rawRouter3);
this.groupBox19.Controls.Add(this.label58);
this.groupBox19.Controls.Add(this.label59);
this.groupBox19.Controls.Add(this.label60);
this.groupBox19.Controls.Add(this.pictureBox7);
this.groupBox19.Controls.Add(this.pictureBox8);
this.groupBox19.Controls.Add(this.label61);
this.groupBox19.Controls.Add(this.timeConfRouter3);
this.groupBox19.Location = new System.Drawing.Point(5, 5);
this.groupBox19.Name = "groupBox19";
this.groupBox19.Size = new System.Drawing.Size(594, 115);
this.groupBox19.TabIndex = 8;
this.groupBox19.TabStop = false;
this.groupBox19.Text = "Lastest Data";
//
// label40
//
this.label40.AutoSize = true;
this.label40.ForeColor = System.Drawing.Color.Blue;
this.label40.Location = new System.Drawing.Point(133, 90);
this.label40.Name = "label40";
this.label40.Size = new System.Drawing.Size(33, 13);
this.label40.TabIndex = 187;
this.label40.Text = "None";
//
// label42
//
this.label42.AutoSize = true;
this.label42.ForeColor = System.Drawing.Color.Blue;
this.label42.Location = new System.Drawing.Point(77, 71);
this.label42.Name = "label42";
this.label42.Size = new System.Drawing.Size(112, 13);
this.label42.TabIndex = 186;
this.label42.Text = "0013A200 403DC71D";
//
// label43
//
this.label43.AutoSize = true;
```

```
            this.label43.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(222)));
            this.label43.Location = new System.Drawing.Point(36, 90);
            this.label43.Name = "label43";
            this.label43.Size = new System.Drawing.Size(97, 13);
            this.label43.TabIndex = 185;
            this.label43.Text = "Output Control :";
            //
            // label44
            //
            this.label44.AutoSize = true;
            this.label44.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(222)));
            this.label44.Location = new System.Drawing.Point(36, 71);
            this.label44.Name = "label44";
            this.label44.Size = new System.Drawing.Size(41, 13);
            this.label44.TabIndex = 184;
            this.label44.Text = "MAC :";
            //
            // tempRouter3
            //
            this.tempRouter3.AutoSize = true;
            this.tempRouter3.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
            this.tempRouter3.ForeColor = System.Drawing.Color.Red;
            this.tempRouter3.Location = new System.Drawing.Point(495, 40);
            this.tempRouter3.Name = "tempRouter3";
            this.tempRouter3.Size = new System.Drawing.Size(74, 20);
            this.tempRouter3.TabIndex = 181;
            this.tempRouter3.Text = "Waiting...";
            //
            // analogRouter3
            //
            this.analogRouter3.AutoSize = true;
            this.analogRouter3.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
            this.analogRouter3.ForeColor = System.Drawing.Color.Red;
            this.analogRouter3.Location = new System.Drawing.Point(361, 40);
            this.analogRouter3.Name = "analogRouter3";
            this.analogRouter3.Size = new System.Drawing.Size(74, 20);
            this.analogRouter3.TabIndex = 180;
            this.analogRouter3.Text = "Waiting...";
            //
            // rawRouter3
            //
            this.rawRouter3.AutoSize = true;
            this.rawRouter3.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
            this.rawRouter3.ForeColor = System.Drawing.Color.Red;
            this.rawRouter3.Location = new System.Drawing.Point(253, 40);
            this.rawRouter3.Name = "rawRouter3";
            this.rawRouter3.Size = new System.Drawing.Size(74, 20);
            this.rawRouter3.TabIndex = 179;
            this.rawRouter3.Text = "Waiting...";
            //
            // label58
            //
            this.label58.AutoSize = true;
            this.label58.Location = new System.Drawing.Point(496, 24);
            this.label58.Name = "label58";
```

```
this.label58.Size = new System.Drawing.Size(60, 13);
this.label58.TabIndex = 178;
this.label58.Text = "Temp (°C) :";
//
// label59
//
this.label59.AutoSize = true;
this.label59.Location = new System.Drawing.Point(362, 24);
this.label59.Name = "label59";
this.label59.Size = new System.Drawing.Size(109, 13);
this.label59.TabIndex = 177;
this.label59.Text = "Analog Voltage (mV) :";
//
// label60
//
this.label60.AutoSize = true;
this.label60.Location = new System.Drawing.Point(254, 24);
this.label60.Name = "label60";
this.label60.Size = new System.Drawing.Size(61, 13);
this.label60.TabIndex = 176;
this.label60.Text = "Raw Data :";
//
// pictureBox7
//
this.pictureBox7.Image = ((System.Drawing.Image)(resources.GetObject("pictureBox7.Image")));
this.pictureBox7.Location = new System.Drawing.Point(227, 17);
this.pictureBox7.Name = "pictureBox7";
this.pictureBox7.Size = new System.Drawing.Size(24, 24);
this.pictureBox7.TabIndex = 175;
this.pictureBox7.TabStop = false;
//
// pictureBox8
//
this.pictureBox8.Image = ((System.Drawing.Image)(resources.GetObject("pictureBox8.Image")));
this.pictureBox8.Location = new System.Drawing.Point(8, 17);
this.pictureBox8.Name = "pictureBox8";
this.pictureBox8.Size = new System.Drawing.Size(24, 24);
this.pictureBox8.TabIndex = 174;
this.pictureBox8.TabStop = false;
//
// label61
//
this.label61.AutoSize = true;
this.label61.Location = new System.Drawing.Point(35, 24);
this.label61.Name = "label61";
this.label61.Size = new System.Drawing.Size(61, 13);
this.label61.TabIndex = 173;
this.label61.Text = "Last Seen :";
//
// timeConfRouter3
//
this.timeConfRouter3.AutoSize = true;
this.timeConfRouter3.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.timeConfRouter3.ForeColor = System.Drawing.Color.RoyalBlue;
this.timeConfRouter3.Location = new System.Drawing.Point(35, 40);
this.timeConfRouter3.Name = "timeConfRouter3";
this.timeConfRouter3.Size = new System.Drawing.Size(74, 20);
this.timeConfRouter3.TabIndex = 172;
this.timeConfRouter3.Text = "Waiting...";
```

```
//
// groupBox4
//
this.groupBox4.Controls.Add(this.tabXbee);
this.groupBox4.Location = new System.Drawing.Point(12, 172);
this.groupBox4.Name = "groupBox4";
this.groupBox4.Size = new System.Drawing.Size(632, 231);
this.groupBox4.TabIndex = 162;
this.groupBox4.TabStop = false;
this.groupBox4.Text = "Information";
//
// notifyIcon1
//
this.notifyIcon1.Text = "notifyIcon1";
this.notifyIcon1.Visible = true;
//
// onlineTimeCheck
//
this.onlineTimeCheck.Interval = 1000;
this.onlineTimeCheck.Tick += new System.EventHandler(this.onlineTimeCheck_Tick);
//
// label24
//
this.label24.AutoSize = true;
this.label24.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.label24.ForeColor = System.Drawing.Color.Red;
this.label24.Location = new System.Drawing.Point(17, 21);
this.label24.Name = "label24";
this.label24.Size = new System.Drawing.Size(44, 13);
this.label24.TabIndex = 0;
this.label24.Text = "Offline";
//
// button3
//
this.button3.Location = new System.Drawing.Point(114, 16);
this.button3.Name = "button3";
this.button3.Size = new System.Drawing.Size(100, 23);
this.button3.TabIndex = 1;
this.button3.Text = "Power Off O/P:1";
this.button3.UseVisualStyleBackColor = true;
//
// button4
//
this.button4.Location = new System.Drawing.Point(8, 16);
this.button4.Name = "button4";
this.button4.Size = new System.Drawing.Size(100, 23);
this.button4.TabIndex = 0;
this.button4.Text = "Power On O/P:1";
this.button4.UseVisualStyleBackColor = true;
//
// label25
//
this.label25.AutoSize = true;
this.label25.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.label25.ForeColor = System.Drawing.Color.DarkGreen;
this.label25.Location = new System.Drawing.Point(495, 83);
this.label25.Name = "label25";
this.label25.Size = new System.Drawing.Size(74, 20);
```

```
            this.label25.TabIndex = 183;
            this.label25.Text = "Waiting...";
            //
            // label26
            //
            this.label26.AutoSize = true;
            this.label26.Location = new System.Drawing.Point(496, 67);
            this.label26.Name = "label26";
            this.label26.Size = new System.Drawing.Size(42, 13);
            this.label26.TabIndex = 182;
            this.label26.Text = "O/P 1 :";
            //
            // label27
            //
            this.label27.AutoSize = true;
            this.label27.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(222)));
            this.label27.ForeColor = System.Drawing.Color.Red;
            this.label27.Location = new System.Drawing.Point(495, 40);
            this.label27.Name = "label27";
            this.label27.Size = new System.Drawing.Size(74, 20);
            this.label27.TabIndex = 181;
            this.label27.Text = "Waiting...";
            //
            // label28
            //
            this.label28.AutoSize = true;
            this.label28.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(222)));
            this.label28.ForeColor = System.Drawing.Color.Red;
            this.label28.Location = new System.Drawing.Point(361, 40);
            this.label28.Name = "label28";
            this.label28.Size = new System.Drawing.Size(74, 20);
            this.label28.TabIndex = 180;
            this.label28.Text = "Waiting...";
            //
            // label29
            //
            this.label29.AutoSize = true;
            this.label29.Font = new System.Drawing.Font("Microsoft Sans Serif", 12F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(222)));
            this.label29.ForeColor = System.Drawing.Color.Red;
            this.label29.Location = new System.Drawing.Point(253, 40);
            this.label29.Name = "label29";
            this.label29.Size = new System.Drawing.Size(74, 20);
            this.label29.TabIndex = 179;
            this.label29.Text = "Waiting...";
            //
            // label30
            //
            this.label30.AutoSize = true;
            this.label30.Location = new System.Drawing.Point(496, 24);
            this.label30.Name = "label30";
            this.label30.Size = new System.Drawing.Size(60, 13);
            this.label30.TabIndex = 178;
            this.label30.Text = "Temp (°C) :";
            //
            // label31
            //
            this.label31.AutoSize = true;
```

```
this.label31.Location = new System.Drawing.Point(362, 24);
this.label31.Name = "label31";
this.label31.Size = new System.Drawing.Size(109, 13);
this.label31.TabIndex = 177;
this.label31.Text = "Analog Voltage (mV) :";
//
// label32
//
this.label32.AutoSize = true;
this.label32.Location = new System.Drawing.Point(254, 24);
this.label32.Name = "label32";
this.label32.Size = new System.Drawing.Size(61, 13);
this.label32.TabIndex = 176;
this.label32.Text = "Raw Data :";
//
// pictureBox3
//
this.pictureBox3.Image = ((System.Drawing.Image)(resources.GetObject("pictureBox3.Image")));
this.pictureBox3.Location = new System.Drawing.Point(227, 17);
this.pictureBox3.Name = "pictureBox3";
this.pictureBox3.Size = new System.Drawing.Size(24, 24);
this.pictureBox3.TabIndex = 175;
this.pictureBox3.TabStop = false;
//
// pictureBox4
//
this.pictureBox4.Image = ((System.Drawing.Image)(resources.GetObject("pictureBox4.Image")));
this.pictureBox4.Location = new System.Drawing.Point(8, 17);
this.pictureBox4.Name = "pictureBox4";
this.pictureBox4.Size = new System.Drawing.Size(24, 24);
this.pictureBox4.TabIndex = 174;
this.pictureBox4.TabStop = false;
//
// label33
//
this.label33.AutoSize = true;
this.label33.Location = new System.Drawing.Point(35, 24);
this.label33.Name = "label33";
this.label33.Size = new System.Drawing.Size(61, 13);
this.label33.TabIndex = 173;
this.label33.Text = "Last Seen :";
//
// label34
//
this.label34.AutoSize = true;
this.label34.Font = new System.Drawing.Font("Microsoft Sans Serif", 14F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.label34.ForeColor = System.Drawing.Color.RoyalBlue;
this.label34.Location = new System.Drawing.Point(35, 40);
this.label34.Name = "label34";
this.label34.Size = new System.Drawing.Size(87, 24);
this.label34.TabIndex = 172;
this.label34.Text = "Waiting...";
//
// groupBox17
//
this.groupBox17.Controls.Add(this.panel7);
this.groupBox17.Location = new System.Drawing.Point(650, 116);
this.groupBox17.Name = "groupBox17";
this.groupBox17.Size = new System.Drawing.Size(522, 513);
```

```
this.groupBox17.TabIndex = 163;
this.groupBox17.TabStop = false;
this.groupBox17.Text = "Network MAP";
//
// panel7
//
this.panel7.BackColor = System.Drawing.SystemColors.InfoText;
this.panel7.Controls.Add(this.label63);
this.panel7.Controls.Add(this.label56);
this.panel7.Controls.Add(this.label62);
this.panel7.Controls.Add(this.router3in1view);
this.panel7.Controls.Add(this.router1in1view);
this.panel7.Controls.Add(this.label64);
this.panel7.Controls.Add(this.label57);
this.panel7.Controls.Add(this.router2in1view);
this.panel7.Controls.Add(this.label55);
this.panel7.Controls.Add(this.label54);
this.panel7.Controls.Add(this.label53);
this.panel7.Controls.Add(this.label35);
this.panel7.Controls.Add(this.label7);
this.panel7.Controls.Add(this.label6);
this.panel7.Controls.Add(this.imgWarnCoor);
this.panel7.Controls.Add(this.imgWarnR3);
this.panel7.Controls.Add(this.imgWarnR2);
this.panel7.Controls.Add(this.imgWarnR1);
this.panel7.Controls.Add(this.pictureBox18);
this.panel7.Controls.Add(this.pictureBox17);
this.panel7.Controls.Add(this.pictureBox11);
this.panel7.Controls.Add(this.imgRouter3);
this.panel7.Controls.Add(this.imgRouter2);
this.panel7.Controls.Add(this.imgRouter1);
this.panel7.Controls.Add(this.imgCoordinator);
this.panel7.Controls.Add(this.pictureBox12);
this.panel7.Controls.Add(this.imgServer);
this.panel7.Location = new System.Drawing.Point(11, 21);
this.panel7.Name = "panel7";
this.panel7.Size = new System.Drawing.Size(500, 480);
this.panel7.TabIndex = 0;
//
// label62
//
this.label62.AutoSize = true;
this.label62.BackColor = System.Drawing.Color.Transparent;
this.label62.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.label62.ForeColor = System.Drawing.Color.FromArgb(((int)(((byte)(224)))), ((int)(((byte)(224)))),
((int)(((byte)(224)))));
this.label62.Location = new System.Drawing.Point(337, 189);
this.label62.Name = "label62";
this.label62.Size = new System.Drawing.Size(20, 13);
this.label62.TabIndex = 9;
this.label62.Text = "°C";
//
// router3in1view
//
this.router3in1view.AutoSize = true;
this.router3in1view.BackColor = System.Drawing.Color.Transparent;
this.router3in1view.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.router3in1view.ForeColor = System.Drawing.Color.Aqua;
```

```
this.router3in1view.Location = new System.Drawing.Point(306, 189);
this.router3in1view.Name = "router3in1view";
this.router3in1view.Size = new System.Drawing.Size(27, 13);
this.router3in1view.TabIndex = 8;
this.router3in1view.Text = "N/A";
this.router3in1view.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
//
// router1in1view
//
this.router1in1view.AutoSize = true;
this.router1in1view.BackColor = System.Drawing.Color.Transparent;
this.router1in1view.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.router1in1view.ForeColor = System.Drawing.Color.Aqua;
this.router1in1view.Location = new System.Drawing.Point(147, 25);
this.router1in1view.Name = "router1in1view";
this.router1in1view.Size = new System.Drawing.Size(27, 13);
this.router1in1view.TabIndex = 8;
this.router1in1view.Text = "N/A";
this.router1in1view.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
//
// label64
//
this.label64.AutoSize = true;
this.label64.BackColor = System.Drawing.Color.Transparent;
this.label64.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.label64.ForeColor = System.Drawing.Color.FromArgb(((int)(((byte)(224)))), ((int)(((byte)(224)))),
((int)(((byte)(224)))));
this.label64.Location = new System.Drawing.Point(436, 37);
this.label64.Name = "label64";
this.label64.Size = new System.Drawing.Size(16, 13);
this.label64.TabIndex = 10;
this.label64.Text = "%";
//
// label57
//
this.label57.AutoSize = true;
this.label57.BackColor = System.Drawing.Color.Transparent;
this.label57.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.label57.ForeColor = System.Drawing.Color.FromArgb(((int)(((byte)(224)))), ((int)(((byte)(224)))),
((int)(((byte)(224)))));
this.label57.Location = new System.Drawing.Point(179, 25);
this.label57.Name = "label57";
this.label57.Size = new System.Drawing.Size(20, 13);
this.label57.TabIndex = 7;
this.label57.Text = "°C";
//
// router2in1view
//
this.router2in1view.AutoSize = true;
this.router2in1view.BackColor = System.Drawing.Color.Transparent;
this.router2in1view.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.router2in1view.ForeColor = System.Drawing.Color.Yellow;
this.router2in1view.Location = new System.Drawing.Point(401, 37);
this.router2in1view.Name = "router2in1view";
this.router2in1view.Size = new System.Drawing.Size(27, 13);
this.router2in1view.TabIndex = 9;
```

```
this.router2in1view.Text = "N/A";
this.router2in1view.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
//
// label55
//
this.label55.AutoSize = true;
this.label55.ForeColor = System.Drawing.Color.White;
this.label55.Location = new System.Drawing.Point(52, 454);
this.label55.Name = "label55";
this.label55.Size = new System.Drawing.Size(38, 13);
this.label55.TabIndex = 19;
this.label55.Text = "Server";
//
// label54
//
this.label54.AutoSize = true;
this.label54.ForeColor = System.Drawing.Color.White;
this.label54.Location = new System.Drawing.Point(131, 454);
this.label54.Name = "label54";
this.label54.Size = new System.Drawing.Size(49, 13);
this.label54.TabIndex = 18;
this.label54.Text = "Gateway";
//
// label53
//
this.label53.AutoSize = true;
this.label53.ForeColor = System.Drawing.Color.White;
this.label53.Location = new System.Drawing.Point(125, 372);
this.label53.Name = "label53";
this.label53.Size = new System.Drawing.Size(61, 13);
this.label53.TabIndex = 17;
this.label53.Text = "Coordinator";
//
// label35
//
this.label35.AutoSize = true;
this.label35.ForeColor = System.Drawing.Color.White;
this.label35.Location = new System.Drawing.Point(267, 279);
this.label35.Name = "label35";
this.label35.Size = new System.Drawing.Size(48, 13);
this.label35.TabIndex = 16;
this.label35.Text = "Router 3";
//
// label7
//
this.label7.AutoSize = true;
this.label7.ForeColor = System.Drawing.Color.White;
this.label7.Location = new System.Drawing.Point(352, 115);
this.label7.Name = "label7";
this.label7.Size = new System.Drawing.Size(48, 13);
this.label7.TabIndex = 15;
this.label7.Text = "Router 2";
//
// label6
//
this.label6.AutoSize = true;
this.label6.ForeColor = System.Drawing.Color.White;
this.label6.Location = new System.Drawing.Point(107, 115);
this.label6.Name = "label6";
this.label6.Size = new System.Drawing.Size(48, 13);
```

```
this.label6.TabIndex = 14;
this.label6.Text = "Router 1";
//
// imgWarnCoor
//
this.imgWarnCoor.Image = ((System.Drawing.Image)(resources.GetObject("imgWarnCoor.Image")));
this.imgWarnCoor.Location = new System.Drawing.Point(91, 277);
this.imgWarnCoor.Name = "imgWarnCoor";
this.imgWarnCoor.Size = new System.Drawing.Size(54, 48);
this.imgWarnCoor.TabIndex = 13;
this.imgWarnCoor.TabStop = false;
//
// imgWarnR3
//
this.imgWarnR3.Image = ((System.Drawing.Image)(resources.GetObject("imgWarnR3.Image")));
this.imgWarnR3.Location = new System.Drawing.Point(227, 184);
this.imgWarnR3.Name = "imgWarnR3";
this.imgWarnR3.Size = new System.Drawing.Size(54, 48);
this.imgWarnR3.TabIndex = 12;
this.imgWarnR3.TabStop = false;
//
// imgWarnR2
//
this.imgWarnR2.Image = ((System.Drawing.Image)(resources.GetObject("imgWarnR2.Image")));
this.imgWarnR2.Location = new System.Drawing.Point(311, 20);
this.imgWarnR2.Name = "imgWarnR2";
this.imgWarnR2.Size = new System.Drawing.Size(54, 48);
this.imgWarnR2.TabIndex = 11;
this.imgWarnR2.TabStop = false;
//
// imgWarnR1
//
this.imgWarnR1.Image = ((System.Drawing.Image)(resources.GetObject("imgWarnR1.Image")));
this.imgWarnR1.Location = new System.Drawing.Point(64, 20);
this.imgWarnR1.Name = "imgWarnR1";
this.imgWarnR1.Size = new System.Drawing.Size(54, 48);
this.imgWarnR1.TabIndex = 10;
this.imgWarnR1.TabStop = false;
//
// pictureBox18
//
this.pictureBox18.Image = ((System.Drawing.Image)(resources.GetObject("pictureBox18.Image")));
this.pictureBox18.Location = new System.Drawing.Point(409, 53);
this.pictureBox18.Name = "pictureBox18";
this.pictureBox18.Size = new System.Drawing.Size(27, 59);
this.pictureBox18.TabIndex = 9;
this.pictureBox18.TabStop = false;
//
// pictureBox17
//
this.pictureBox17.Image = ((System.Drawing.Image)(resources.GetObject("pictureBox17.Image")));
this.pictureBox17.Location = new System.Drawing.Point(161, 40);
this.pictureBox17.Name = "pictureBox17";
this.pictureBox17.Size = new System.Drawing.Size(23, 72);
this.pictureBox17.TabIndex = 8;
this.pictureBox17.TabStop = false;
//
// pictureBox11
//
this.pictureBox11.Image = ((System.Drawing.Image)(resources.GetObject("pictureBox11.Image")));
```

```
this.pictureBox11.Location = new System.Drawing.Point(321, 204);
this.pictureBox11.Name = "pictureBox11";
this.pictureBox11.Size = new System.Drawing.Size(23, 72);
this.pictureBox11.TabIndex = 7;
this.pictureBox11.TabStop = false;
//
// imgRouter3
//
this.imgRouter3.Image = ((System.Drawing.Image)(resources.GetObject("imgRouter3.Image")));
this.imgRouter3.Location = new System.Drawing.Point(256, 231);
this.imgRouter3.Name = "imgRouter3";
this.imgRouter3.Size = new System.Drawing.Size(66, 45);
this.imgRouter3.TabIndex = 6;
this.imgRouter3.TabStop = false;
this.imgRouter3.Click += new System.EventHandler(this.imgRouter3_Click);
//
// imgRouter2
//
this.imgRouter2.Image = ((System.Drawing.Image)(resources.GetObject("imgRouter2.Image")));
this.imgRouter2.Location = new System.Drawing.Point(341, 67);
this.imgRouter2.Name = "imgRouter2";
this.imgRouter2.Size = new System.Drawing.Size(66, 45);
this.imgRouter2.TabIndex = 5;
this.imgRouter2.TabStop = false;
this.imgRouter2.Click += new System.EventHandler(this.imgRouter2_Click);
//
// imgRouter1
//
this.imgRouter1.Image = ((System.Drawing.Image)(resources.GetObject("imgRouter1.Image")));
this.imgRouter1.Location = new System.Drawing.Point(96, 67);
this.imgRouter1.Name = "imgRouter1";
this.imgRouter1.Size = new System.Drawing.Size(66, 45);
this.imgRouter1.TabIndex = 4;
this.imgRouter1.TabStop = false;
this.imgRouter1.Click += new System.EventHandler(this.imgRouter1_Click);
//
// imgCoordinator
//
this.imgCoordinator.Image = ((System.Drawing.Image)(resources.GetObject("imgCoordinator.Image")));
this.imgCoordinator.Location = new System.Drawing.Point(121, 324);
this.imgCoordinator.Name = "imgCoordinator";
this.imgCoordinator.Size = new System.Drawing.Size(66, 45);
this.imgCoordinator.TabIndex = 3;
this.imgCoordinator.TabStop = false;
//
// pictureBox12
//
this.pictureBox12.Image = ((System.Drawing.Image)(resources.GetObject("pictureBox12.Image")));
this.pictureBox12.Location = new System.Drawing.Point(133, 392);
this.pictureBox12.Name = "pictureBox12";
this.pictureBox12.Size = new System.Drawing.Size(43, 59);
this.pictureBox12.TabIndex = 2;
this.pictureBox12.TabStop = false;
//
// imgServer
//
this.imgServer.Image = ((System.Drawing.Image)(resources.GetObject("imgServer.Image")));
this.imgServer.Location = new System.Drawing.Point(19, 356);
this.imgServer.Name = "imgServer";
this.imgServer.Size = new System.Drawing.Size(96, 96);
```

```
this.imgServer.TabIndex = 0;
this.imgServer.TabStop = false;
this.imgServer.Click += new System.EventHandler(this.imgServer_Click);
//
// panel8
//
this.panel8.BackColor = System.Drawing.SystemColors.MenuText;
this.panel8.Controls.Add(this.lblClockText2);
this.panel8.Controls.Add(this.label1);
this.panel8.Controls.Add(this.lblClockText);
this.panel8.Location = new System.Drawing.Point(661, 54);
this.panel8.Name = "panel8";
this.panel8.Size = new System.Drawing.Size(500, 51);
this.panel8.TabIndex = 1;
//
// lblClockText2
//
this.lblClockText2.AutoSize = true;
this.lblClockText2.Font = new System.Drawing.Font("Microsoft Sans Serif", 32F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.lblClockText2.ForeColor = System.Drawing.Color.Silver;
this.lblClockText2.Location = new System.Drawing.Point(297, 0);
this.lblClockText2.Name = "lblClockText2";
this.lblClockText2.Size = new System.Drawing.Size(186, 51);
this.lblClockText2.TabIndex = 165;
this.lblClockText2.Text = "-- : -- : --";
//
// label1
//
this.label1.AutoSize = true;
this.label1.Font = new System.Drawing.Font("Microsoft Sans Serif", 10F, System.Drawing.FontStyle.Bold,
System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.label1.ForeColor = System.Drawing.SystemColors.AppWorkspace;
this.label1.Location = new System.Drawing.Point(2, 3);
this.label1.Name = "label1";
this.label1.Size = new System.Drawing.Size(161, 17);
this.label1.TabIndex = 0;
this.label1.Text = "Current Date / Time :";
//
// lblClockText
//
this.lblClockText.AutoSize = true;
this.lblClockText.Font = new System.Drawing.Font("Microsoft Sans Serif", 14F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.lblClockText.ForeColor = System.Drawing.Color.Silver;
this.lblClockText.Location = new System.Drawing.Point(122, 23);
this.lblClockText.Name = "lblClockText";
this.lblClockText.Size = new System.Drawing.Size(114, 24);
this.lblClockText.TabIndex = 164;
this.lblClockText.Text = "- - - - - - - - ";
//
// tmrRefresh
//
this.tmrRefresh.Tick += new System.EventHandler(this.tmrRefresh_Tick);
//
// button1
//
this.button1.Location = new System.Drawing.Point(99, 69);
this.button1.Name = "button1";
this.button1.Size = new System.Drawing.Size(85, 22);
```

```
this.button1.TabIndex = 164;
this.button1.Text = "Force Now";
this.button1.UseVisualStyleBackColor = true;
this.button1.Click += new System.EventHandler(this.button1_Click);
//
// pictureBox9
//
this.pictureBox9.Image = ((System.Drawing.Image)(resources.GetObject("pictureBox9.Image")));
this.pictureBox9.Location = new System.Drawing.Point(524, 48);
this.pictureBox9.Name = "pictureBox9";
this.pictureBox9.Size = new System.Drawing.Size(128, 128);
this.pictureBox9.TabIndex = 165;
this.pictureBox9.TabStop = false;
//
// groupBox18
//
this.groupBox18.Controls.Add(this.label3);
this.groupBox18.Controls.Add(this.refreshForceTimer);
this.groupBox18.Controls.Add(this.timeForceLabel);
this.groupBox18.Controls.Add(this.button1);
this.groupBox18.Location = new System.Drawing.Point(339, 54);
this.groupBox18.Name = "groupBox18";
this.groupBox18.Size = new System.Drawing.Size(190, 112);
this.groupBox18.TabIndex = 166;
this.groupBox18.TabStop = false;
this.groupBox18.Text = "Force Sample Data";
//
// label3
//
this.label3.AutoSize = true;
this.label3.Location = new System.Drawing.Point(8, 48);
this.label3.Name = "label3";
this.label3.Size = new System.Drawing.Size(129, 13);
this.label3.TabIndex = 172;
this.label3.Text = "(seconds) to Sample Data";
//
// refreshForceTimer
//
this.refreshForceTimer.Font = new System.Drawing.Font("Microsoft Sans Serif", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.refreshForceTimer.Location = new System.Drawing.Point(6, 69);
this.refreshForceTimer.Name = "refreshForceTimer";
this.refreshForceTimer.Size = new System.Drawing.Size(91, 22);
this.refreshForceTimer.TabIndex = 171;
this.refreshForceTimer.Text = "Refresh Timer";
this.refreshForceTimer.UseVisualStyleBackColor = true;
this.refreshForceTimer.Click += new System.EventHandler(this.refreshForceTimer_Click);
//
// timeForceLabel
//
this.timeForceLabel.AutoSize = true;
this.timeForceLabel.Font = new System.Drawing.Font("Microsoft Sans Serif", 14F,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.timeForceLabel.ForeColor = System.Drawing.Color.RoyalBlue;
this.timeForceLabel.Location = new System.Drawing.Point(6, 22);
this.timeForceLabel.Name = "timeForceLabel";
this.timeForceLabel.Size = new System.Drawing.Size(69, 24);
this.timeForceLabel.TabIndex = 171;
this.timeForceLabel.Text = "- - - - -";
this.timeForceLabel.TextAlign = System.Drawing.ContentAlignment.MiddleLeft;
```

```
//
// onlineForceCheck
//
this.onlineForceCheck.Interval = 1000;
this.onlineForceCheck.Tick += new System.EventHandler(this.onlineForceCheck_Tick);
//
// ImageBlink
//
this.ImageBlink.Enabled = true;
this.ImageBlink.Interval = 500;
this.ImageBlink.Tick += new System.EventHandler(this.ImageBlink_Tick);
//
// label56
//
this.label56.AutoSize = true;
this.label56.Font = new System.Drawing.Font("Microsoft Sans Serif", 18F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.label56.ForeColor = System.Drawing.SystemColors.AppWorkspace;
this.label56.Location = new System.Drawing.Point(255, 425);
this.label56.Name = "label56";
this.label56.Size = new System.Drawing.Size(242, 29);
this.label56.TabIndex = 20;
this.label56.Text = "Network Sensor Area";
//
// label63
//
this.label63.AutoSize = true;
this.label63.Font = new System.Drawing.Font("Microsoft Sans Serif", 10F, System.Drawing.FontStyle.Regular,
System.Drawing.GraphicsUnit.Point, ((byte)(222)));
this.label63.ForeColor = System.Drawing.SystemColors.AppWorkspace;
this.label63.Location = new System.Drawing.Point(302, 451);
this.label63.Name = "label63";
this.label63.Size = new System.Drawing.Size(191, 17);
this.label63.TabIndex = 21;
this.label63.Text = "Powered By GDI 2.0+ Library";
//
// Form1
//
this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
this.BackColor = System.Drawing.SystemColors.Menu;
this.ClientSize = new System.Drawing.Size(1184, 662);
this.Controls.Add(this.groupBox18);
this.Controls.Add(this.panel8);
this.Controls.Add(this.groupBox17);
this.Controls.Add(this.groupBox4);
this.Controls.Add(this.toolStrip);
this.Controls.Add(this.statusStrip);
this.Controls.Add(this.groupBox3);
this.Controls.Add(this.groupBox2);
this.Controls.Add(this.groupBox1);
this.Controls.Add(this.menuStrip);
this.Controls.Add(this.pictureBox9);
this.MainMenuStrip = this.menuStrip;
this.MaximizeBox = false;
this.MaximumSize = new System.Drawing.Size(1200, 700);
this.MinimumSize = new System.Drawing.Size(1200, 700);
this.Name = "Form1";
this.StartPosition = System.Windows.Forms.FormStartPosition.CenterScreen;
this.Text = "Zigbee Network Management 1.0";
```

```
this.groupBox1.ResumeLayout(false);
this.groupBox1.PerformLayout();
this.groupBox2.ResumeLayout(false);
this.groupBox2.PerformLayout();
this.groupBox3.ResumeLayout(false);
this.groupBox3.PerformLayout();
this.panel1.ResumeLayout(false);
this.panel1.PerformLayout();
this.menuStrip.ResumeLayout(false);
this.menuStrip.PerformLayout();
this.statusStrip.ResumeLayout(false);
this.statusStrip.PerformLayout();
this.toolStrip.ResumeLayout(false);
this.toolStrip.PerformLayout();
this.tabXbee.ResumeLayout(false);
this.tabOverview.ResumeLayout(false);
this.tabOverview.PerformLayout();
this.groupBox7.ResumeLayout(false);
this.groupBox7.PerformLayout();
this.panel5.ResumeLayout(false);
this.panel5.PerformLayout();
this.panel4.ResumeLayout(false);
this.panel4.PerformLayout();
this.panel6.ResumeLayout(false);
this.panel6.PerformLayout();
this.panel3.ResumeLayout(false);
this.panel3.PerformLayout();
this.panel2.ResumeLayout(false);
this.panel2.PerformLayout();
this.groupBox6.ResumeLayout(false);
this.groupBox6.PerformLayout();
this.groupBox5.ResumeLayout(false);
this.groupBox5.PerformLayout();
((System.ComponentModel.ISupportInitialize)(this.imgStatusRouter3)).EndInit();
((System.ComponentModel.ISupportInitialize)(this.imgStatusRouter2)).EndInit();
((System.ComponentModel.ISupportInitialize)(this.imgStatusRouter1)).EndInit();
((System.ComponentModel.ISupportInitialize)(this.imgStatusCoordinator)).EndInit();
this.tabRouter1.ResumeLayout(false);
this.groupBox11.ResumeLayout(false);
this.groupBox11.PerformLayout();
this.groupBox10.ResumeLayout(false);
this.groupBox10.PerformLayout();
this.groupBox9.ResumeLayout(false);
this.groupBox8.ResumeLayout(false);
this.groupBox8.PerformLayout();
((System.ComponentModel.ISupportInitialize)(this.pictureBox2)).EndInit();
((System.ComponentModel.ISupportInitialize)(this.pictureBox1)).EndInit();
this.tabRouter2.ResumeLayout(false);
this.groupBox12.ResumeLayout(false);
this.groupBox12.PerformLayout();
this.groupBox13.ResumeLayout(false);
this.groupBox13.PerformLayout();
this.groupBox14.ResumeLayout(false);
this.groupBox15.ResumeLayout(false);
this.groupBox15.PerformLayout();
((System.ComponentModel.ISupportInitialize)(this.pictureBox5)).EndInit();
((System.ComponentModel.ISupportInitialize)(this.pictureBox6)).EndInit();
this.tabRouter3.ResumeLayout(false);
this.groupBox16.ResumeLayout(false);
this.groupBox16.PerformLayout();
```

```
            this.groupBox19.ResumeLayout(false);
            this.groupBox19.PerformLayout();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox7)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox8)).EndInit();
            this.groupBox4.ResumeLayout(false);
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox3)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox4)).EndInit();
            this.groupBox17.ResumeLayout(false);
            this.panel7.ResumeLayout(false);
            this.panel7.PerformLayout();
            ((System.ComponentModel.ISupportInitialize)(this.imgWarnCoor)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.imgWarnR3)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.imgWarnR2)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.imgWarnR1)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox18)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox17)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox11)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.imgRouter3)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.imgRouter2)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.imgRouter1)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.imgCoordinator)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox12)).EndInit();
            ((System.ComponentModel.ISupportInitialize)(this.imgServer)).EndInit();
            this.panel8.ResumeLayout(false);
            this.panel8.PerformLayout();
            ((System.ComponentModel.ISupportInitialize)(this.pictureBox9)).EndInit();
            this.groupBox18.ResumeLayout(false);
            this.groupBox18.PerformLayout();
            this.ResumeLayout(false);
            this.PerformLayout();

        }

        #endregion

        private System.Windows.Forms.Button btnConnect;
        private System.Windows.Forms.ComboBox serialCombox;
        private System.Windows.Forms.ComboBox serialBaudbox;
        private System.Windows.Forms.TextBox txtData;
        private System.Windows.Forms.GroupBox groupBox1;
        private System.Windows.Forms.Button btnRefresh;
        private System.Windows.Forms.Label label9;
        private System.Windows.Forms.Label label8;
        private System.Windows.Forms.GroupBox groupBox2;
        private System.Windows.Forms.GroupBox groupBox3;
        private System.Windows.Forms.Label label5;
        private System.Windows.Forms.Panel panel1;
        private System.Windows.Forms.Label modeControlText;
        private System.Windows.Forms.RadioButton setModebt2;
        private System.Windows.Forms.RadioButton setModebt1;
        private System.Windows.Forms.MenuStrip menuStrip;
        private System.Windows.Forms.ToolStripMenuItem fileToolStripMenuItem;
        private System.Windows.Forms.ToolStripMenuItem helpToolStripMenuItem;
        private System.Windows.Forms.ToolStripMenuItem aboutToolStripMenuItem;
        private System.Windows.Forms.ToolStripMenuItem exitToolStripMenuItem;
        private System.Windows.Forms.StatusStrip statusStrip;
        private System.Windows.Forms.ToolStrip toolStrip;
        private System.Windows.Forms.ToolStripLabel toolStripLabel1;
        private System.Windows.Forms.ToolStripSeparator toolStripSeparator1;
        private System.Windows.Forms.TabControl tabXbee;
```

```
private System.Windows.Forms.TabPage tabRouter1;
private System.Windows.Forms.TabPage tabRouter2;
private System.Windows.Forms.TabPage tabRouter3;
private System.Windows.Forms.GroupBox groupBox4;
private System.Windows.Forms.TabPage tabOverview;
private System.Windows.Forms.NotifyIcon notifyIcon1;
private System.Windows.Forms.GroupBox groupBox5;
private System.Windows.Forms.Label coordinatorLabel;
private System.Windows.Forms.Label routerLabel3;
private System.Windows.Forms.Label routerLabel2;
private System.Windows.Forms.Label routerLabel1;
private System.Windows.Forms.PictureBox imgStatusCoordinator;
private System.Windows.Forms.PictureBox imgStatusRouter3;
private System.Windows.Forms.PictureBox imgStatusRouter2;
private System.Windows.Forms.PictureBox imgStatusRouter1;
private System.Windows.Forms.Label textlabel10;
private System.Windows.Forms.Label timeLabel;
private System.Windows.Forms.Timer onlineTimeCheck;
private System.Windows.Forms.Label labelSecond;
private System.Windows.Forms.GroupBox groupBox6;
private System.Windows.Forms.Label timeLabelRouter1;
private System.Windows.Forms.Label timeLabelRouter3;
private System.Windows.Forms.Label timeLabelRouter2;
private System.Windows.Forms.Label timeLabelCoordinator;
private System.Windows.Forms.GroupBox groupBox7;
private System.Windows.Forms.Label label12;
private System.Windows.Forms.Label label11;
private System.Windows.Forms.Label label10;
private System.Windows.Forms.Label label14;
private System.Windows.Forms.Label label13;
private System.Windows.Forms.Label label16;
private System.Windows.Forms.Label label15;
private System.Windows.Forms.Panel panel2;
private System.Windows.Forms.Panel panel3;
private System.Windows.Forms.Panel panel4;
private System.Windows.Forms.Panel panel5;
private System.Windows.Forms.Panel panel6;
private System.Windows.Forms.Label router3in1;
private System.Windows.Forms.Label router1in1;
private System.Windows.Forms.Label router2in1;
private System.Windows.Forms.Label label17;
private System.Windows.Forms.Label label18;
private System.Windows.Forms.Label router2out1;
private System.Windows.Forms.Label router1out1;
private System.Windows.Forms.GroupBox groupBox8;
private System.Windows.Forms.Label timeConfRouter1;
private System.Windows.Forms.PictureBox pictureBox1;
private System.Windows.Forms.Label label19;
private System.Windows.Forms.PictureBox pictureBox2;
private System.Windows.Forms.Label label20;
private System.Windows.Forms.Label label22;
private System.Windows.Forms.Label label21;
private System.Windows.Forms.Label rawRouter1;
private System.Windows.Forms.Label tempRouter1;
private System.Windows.Forms.Label analogRouter1;
private System.Windows.Forms.Label label23;
private System.Windows.Forms.Label outputRouter1;
private System.Windows.Forms.GroupBox groupBox9;
private System.Windows.Forms.GroupBox groupBox10;
private System.Windows.Forms.Button btRouter1Off1;
```

```
private System.Windows.Forms.Button btRouter1On1;
private System.Windows.Forms.GroupBox groupBox11;
private System.Windows.Forms.Label statRouter1;
private System.Windows.Forms.Label label24;
private System.Windows.Forms.Button button3;
private System.Windows.Forms.Button button4;
private System.Windows.Forms.Label label25;
private System.Windows.Forms.Label label26;
private System.Windows.Forms.Label label27;
private System.Windows.Forms.Label label28;
private System.Windows.Forms.Label label29;
private System.Windows.Forms.Label label30;
private System.Windows.Forms.Label label31;
private System.Windows.Forms.Label label32;
private System.Windows.Forms.PictureBox pictureBox3;
private System.Windows.Forms.PictureBox pictureBox4;
private System.Windows.Forms.Label label33;
private System.Windows.Forms.Label label34;
private System.Windows.Forms.Label label49;
private System.Windows.Forms.Label label48;
private System.Windows.Forms.Label label47;
private System.Windows.Forms.Label label46;
private System.Windows.Forms.GroupBox groupBox12;
private System.Windows.Forms.Label statRouter2;
private System.Windows.Forms.GroupBox groupBox13;
private System.Windows.Forms.GroupBox groupBox14;
private System.Windows.Forms.Button btRouter2Off1;
private System.Windows.Forms.Button btRouter2On1;
private System.Windows.Forms.GroupBox groupBox15;
private System.Windows.Forms.Label label36;
private System.Windows.Forms.Label label37;
private System.Windows.Forms.Label label38;
private System.Windows.Forms.Label label39;
private System.Windows.Forms.Label outputRouter2;
private System.Windows.Forms.Label label41;
private System.Windows.Forms.Label tempRouter2;
private System.Windows.Forms.Label analogRouter2;
private System.Windows.Forms.Label rawRouter2;
private System.Windows.Forms.Label label45;
private System.Windows.Forms.Label label50;
private System.Windows.Forms.Label label51;
private System.Windows.Forms.PictureBox pictureBox5;
private System.Windows.Forms.PictureBox pictureBox6;
private System.Windows.Forms.Label label52;
private System.Windows.Forms.Label timeConfRouter2;
private System.Windows.Forms.GroupBox groupBox16;
private System.Windows.Forms.Label statRouter3;
private System.Windows.Forms.GroupBox groupBox19;
private System.Windows.Forms.Label label40;
private System.Windows.Forms.Label label42;
private System.Windows.Forms.Label label43;
private System.Windows.Forms.Label label44;
private System.Windows.Forms.Label tempRouter3;
private System.Windows.Forms.Label analogRouter3;
private System.Windows.Forms.Label rawRouter3;
private System.Windows.Forms.Label label58;
private System.Windows.Forms.Label label59;
private System.Windows.Forms.Label label60;
private System.Windows.Forms.PictureBox pictureBox7;
private System.Windows.Forms.PictureBox pictureBox8;
```

```
private System.Windows.Forms.Label label61;
private System.Windows.Forms.Label timeConfRouter3;
private System.Windows.Forms.ToolStripMenuItem settingToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem thresholdSettingToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem parametersSettingToolStripMenuItem;
private System.Windows.Forms.ToolStripSeparator toolStripSeparator5;
private System.Windows.Forms.ToolStripSeparator toolStripSeparator2;
private System.Windows.Forms.ToolStripSeparator toolStripSeparator3;
private System.Windows.Forms.ToolStripMenuItem viewToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem historyToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem internetMonitoringToolStripMenuItem;
private System.Windows.Forms.ToolStripMenuItem periodRefreshNetworkToolStripMenuItem;
private System.Windows.Forms.GroupBox groupBox17;
private System.Windows.Forms.Panel panel7;
private System.Windows.Forms.Panel panel8;
private System.Windows.Forms.Label label1;
private System.Windows.Forms.Label lblClockText;
private System.Windows.Forms.Timer tmrRefresh;
private System.Windows.Forms.Label lblClockText2;
private System.Windows.Forms.Button button1;
private System.Windows.Forms.PictureBox pictureBox9;
private System.Windows.Forms.PictureBox imgServer;
private System.Windows.Forms.PictureBox pictureBox12;
private System.Windows.Forms.Button refreshTimer;
private System.Windows.Forms.ToolStripStatusLabel toolStripStatusLabel1;
private System.Windows.Forms.Label label2;
private System.Windows.Forms.Label thresholdRouter1;
private System.Windows.Forms.Label thresholdRouter2;
private System.Windows.Forms.Label label4;
private System.Windows.Forms.ToolStripMenuItem databaseToolStripMenuItem;
private System.Windows.Forms.GroupBox groupBox18;
private System.Windows.Forms.Label timeForceLabel;
private System.Windows.Forms.Button refreshForceTimer;
private System.Windows.Forms.Label label3;
private System.Windows.Forms.Timer onlineForceCheck;
private System.Windows.Forms.ToolStripMenuItem periodForceDataATISToolStripMenuItem;
private System.Windows.Forms.PictureBox imgCoordinator;
private System.Windows.Forms.PictureBox imgRouter3;
private System.Windows.Forms.PictureBox imgRouter2;
private System.Windows.Forms.PictureBox imgRouter1;
private System.Windows.Forms.PictureBox pictureBox11;
private System.Windows.Forms.PictureBox pictureBox17;
private System.Windows.Forms.PictureBox pictureBox18;
private System.Windows.Forms.PictureBox imgWarnR3;
private System.Windows.Forms.PictureBox imgWarnR2;
private System.Windows.Forms.PictureBox imgWarnR1;
private System.Windows.Forms.PictureBox imgWarnCoor;
private System.Windows.Forms.Timer ImageBlink;
private System.Windows.Forms.Label label55;
private System.Windows.Forms.Label label54;
private System.Windows.Forms.Label label53;
private System.Windows.Forms.Label label35;
private System.Windows.Forms.Label label7;
private System.Windows.Forms.Label label6;
private System.Windows.Forms.Label label62;
private System.Windows.Forms.Label router3in1view;
private System.Windows.Forms.Label router1in1view;
private System.Windows.Forms.Label label64;
private System.Windows.Forms.Label label57;
private System.Windows.Forms.Label router2in1view;
```

```
        private System.Windows.Forms.Label label56;
        private System.Windows.Forms.Label label63;
    }
}
```