สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

รายงานการวิจัย

ชื่อโครงการวิจัย

(ภาษาไทย) เครื่องวัดสภาพสิ่งแวดล้อมระยะไกลผ่านระบบวิทยุสื่อสาร.

(ภาษาอังกฤษ) Remote environment measuring machine via radio system.

ชื่อผู้วิจัย

นาย อรรถสิทธิ์ หล่าสกุล

RCH
TJ
223
.T4
๑366ค
ฉ.1

เลขหมู่.....
เลขทะเบียน..121391
วัน, เดือน, ปี..4 ก.ค. 2555

b.12179979b
i.....

ได้รับทุนสนับสนุนงานวิจัยจากเงินงบประมาณแผ่นดิน ประจำปีงบประมาณ 2554

คณะวิศวกรรมศาสตร์

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

กิตติกรรมประกาศ (Acknowledgement)

ผู้วิจัยขอขอบคุณ สำหรับ กลุ่มผู้อาศัยในรอบเมืองปทุมธานี ร.ต. หญิง รัญชนา คุ้มพุ่ม ข้าราชการบำนาญ ที่อาศัยอยู่ในย่าน คลองรังสิต และคนอื่นๆ ที่ได้ให้สถานที่เพื่อทำการติดตั้งทดสอบเครื่องวัดสิ่งแวดล้อมของผู้วิจัย และขอขอบคุณ กลุ่มนักศึกษา ในห้องปฏิบัติการ ระบบดิจิตอล หลักสูตรวิศวกรรมสารสนเทศ ที่ให้ความร่วมมือในการทดสอบการทำงานของเครื่องวัดในงานวิจัยนี้ และสุดท้าย ผู้วิจัยขอบคุณต่อ เพื่อนๆ พี่ๆอาจารย์ที่ให้ความรู้คำแนะนำ ทั้งในเชิงวิชาการและประสบการณ์ เช่น รศ.ดร.สุรพันธ์ เอื่อไพบูลย์ อาจารย์ประจำ หลักสูตรวิศวกรรมอิเลกทรอนิกส์ คณะวิศวกกรรมศาสตร์ พระจอมเกล้าลาดกระบัง รวมทั้งขอขอบพระคุณ คุณพ่อคุณแม่, ภรรยาและลูก ที่ได้ให้กำลังใจในการทำงานมาตลอด จนงานวิจัยนี้ได้ประสบผล สำเร็จด้วยดี.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทคัดย่อ

โครงงานวิจัยนี้นำเสนอการออกแบบสร้างเครื่องตรวจสอบวัดค่าอุณหภูมิระยะไกลแบบใหม่ ที่อาศัยการรับส่งข้อมูลด้วยระบบวิทยุสื่อสารแทนระบบโทรศัพท์ทั่วไปทำให้สามารถใช้งานได้ทุกพื้นที่ ระบบใหม่นี้มีการส่งผ่านสัญญาณกันเป็นทอดๆได้ ซึ่งทำให้สามารถติดตั้งเครื่องวัดของตัวลูก (Client) ได้หลายตัวเท่าที่เราต้องการครอบคลุมพื้นที่มากขึ้นโดยไม่ต้องเปลี่ยนหรือเพิ่มระบบกำลังส่งวิทยุ โดยหลักการคือ ให้ตัวลูก ซึ่งมีอยู่หลายจุดนี้ สามารถปรับตัวเองเป็นเหมือนสถานีทวนสัญญาณ (Repeater) ได้ มีการเชื่อมต่อกันคล้ายเครื่อข่าย (Network) คอมพิวเตอร์ ในการรับส่งข้อมูลระหว่างตัวแม่ (Main) กับตัวลูกที่ต้องการติดต่อด้วย โดยโปรแกรมจะควบคุมการติดต่อกันอย่างอัตโนมัติ ตั้งแต่เริ่มส่งคำสั่งจากตัวแม่ไปสู่ตัวลูกที่ต้องการติดต่อด้วยเป็นทอดๆไป โดยใช้ตัวลูกที่อยู่ในพื้นที่รัศมีกำลังส่งของตัวแม่ ทำหน้าที่เป็นตัวกลางในการรับและส่งข้อมูล และเมื่อผู้ใช้ต้องการทราบข้อมูลค่าตรวจวัดที่ตัวลูกที่จุดใด ก็สามารถจะใช้ซอฟต์แวร์ที่ได้ออกแบบเป็นแบบ GUI ทำไว้บนคอมพิวเตอร์ทั่วไป ทำให้ง่ายต่อผู้ใช้งานที่จะสามารถสั่งการเข้าไปดูข้อมูลได้ทุกจุด ทุกเวลาตามความต้องการ.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Abstract

This research presents a new ideal of designing the machine for measures temperature in long distance area with no GSM signal available. This can be done by using radio communication system (WKTK) for communicate instead. The Clients can be communicate to each other nearly same as computer networking system, this make system can install many of Clients unit cover the area more and more without changing the power propagation of the radio system. A designed program will control the connection automatically since sending the command code from main to a desire client. In some case, Client may act as the repeater unit for passing signal (client that located between main and desire client). This system will work nearly same as a computer networking system. By using the friendly program (GUI style) with is installed on general Personal Computer (PC), this make user capable to know the temperature from any clients easily.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญเรื่อง (Table of contents)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สารบัญภาพ (List of Illustrations)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภาพที่	หน้า

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

คำอธิบายสัญลักษณ์และคำย่อที่ใช้ในการวิจัย (List of Abbreviations)

- Microcontroller, Radio Packet, FSK Modulation.

สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

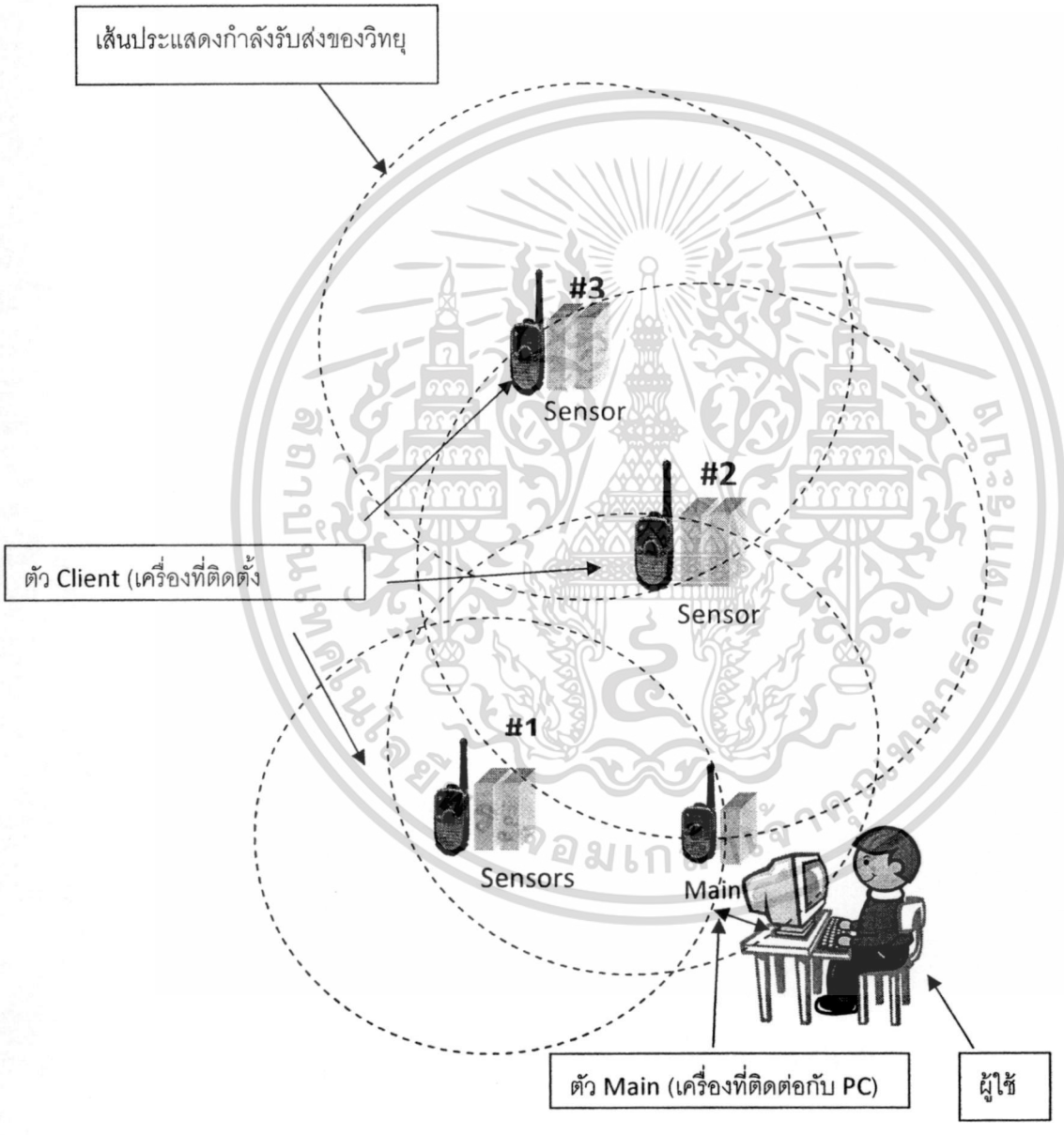
1) บทนำ

ปัจจุบันการตรวจสอบบันทึกค่าการเปลี่ยนแปลงต่างๆ ของธรรมชาติ ติดต่อกันในช่วงเวลานานๆ เพื่อการคาดการณ์หรือวิเคราะห์เหตุการณ์อันอาจจะเกิดขึ้นได้ในอนาคต เป็นสิ่งจำเป็นมากต่อการพิจารณาเพื่อหาแนวทางแก้ไขหรือบรรเทาปัญหาต่างๆที่อาจเกิดขึ้นได้ ตัวอย่าง เช่น การติดตาม ค่าอุณหภูมิ, ค่าระดับน้ำ, ค่าความชื้น เหล่านี้เป็นต้น ซึ่งหากได้มีการตรวจสอบอย่างต่อเนื่องไปเป็นระยะนานๆ ก็สามารถนำค่าเหล่านั้น มาหาสถิติความน่าจะเป็นต่างๆ เพื่อจะได้วิเคราะห์หาวิธีการแก้ไขปัญหาเหล่านั้นได้ตรงจุดที่สุดต่อไป ดังเช่น ปัญหาน้ำท่วมซ้ำซากเป็นต้น แต่การวัดค่าสาเหล่านี้จำเป็นต้องมีการนำเอาเซนเซอร์ไปติดตั้งไว้ในสถานที่ต่างๆ เป็นจำนวนมากและส่วนมากก็จะเป็นสถานที่ๆ ห่างไกลจากความเจริญ เช่น แนวภูเขา ซึ่งจะไม่มีแม้สัญญาณโทรศัพท์ ในพื้นที่นั้นๆ

ดังนั้น หากต้องการทราบว่าค่าที่ต้องการวัดนั้นมีค่าเท่าใด ก็จำเป็นต้องออกเดินป่า เข้าไปจดค่าเหล่านั้น ซึ่งจะสูนย์เสียทั้งเวลาและงบประมาณ, กำลังบุคคล ในการบันทึกค่าเหล่านั้นในแต่ละครั้ง ยิ่งไปกว่านั้น หากต้องมีการเก็บค่าเหล่านั้นเป็นระยะบ่อยครั้งเท่าใดก็จะยิ่งทำให้ เสียค่าใช้จ่ายมากขึ้นตามไปด้วยเท่านั้น เหตุนี้ในงานวิจัยนี้จึง นำเสนอการออกแบบและสร้างเครื่องตรวจสอบวัดค่าข้อมูลของสภาพสิ่งแวดล้อมระยะไกลแบบใหม่ โดยค่าต่างๆที่ต้องการวัดนั้น จะถูกวัดเก็บไว้โดยการทำงานของระบบเซนเซอร์ และไมโครคอนโทรลเลอร์ ที่บรรจุโปรแกรมจัดการการทำงานเพื่อตรวจวัดค่าต่างๆอยู่ในเครื่องดังกล่าวแล้วการรับส่งจะใช้ผ่านทางวิทยุสื่อสาร คล้ายกับการทำ Radio Packet [1] โดยผู้ใช้งานสามารถจะตรวจดูค่าผลของการวัดเหล่านั้นโดยไม่ต้องเดินทางเข้าสู่พื้นที่ติดตั้งเซนเซอร์เหล่านั้นเลย ซึ่งในการใช้งานนั้นเครื่องลูกที่บรรจุเซนเซอร์เหล่านี้ (ซึ่งจะเรียกว่า Client หรือ Beacon) ก็จะถูกนำไปติดตั้งในสถานที่ต่างๆหลายจุด ให้ครอบคลุมบริเวณที่ผู้ใช้งานต้องการ โดยตัวลูกทุกตัวถูกออกแบบให้สามารถทำงานเป็นตัวทวนสัญญานได้ ทำให้สามารถส่งได้ไกลขึ้นโดยไม่จำเป็นต้องเพิ่มกำลังส่งของตัวลูกเลย และเมื่อผู้ใช้ต้องการทราบข้อมูลค่าตรวจวัดที่จุดใด ก็สามารถทำได้อย่างง่ายดาย เพราะในส่วนของซอฟร์แวร์ที่ได้ออกแบบให้ผู้ใช้ได้ใช้งานนั้นจะเป็นแบบ GUI ที่ทำงานบนคอมพิวเตอร์ทั่วไป ดังนั้นจึงง่ายต่อการใช้งานและสามารถเข้าไปดูข้อมูลได้ทุกจุดทุกเวลาได้ตามต้องการ และโดยการรับส่งข้อมูลของแต่ละจุดจะใช้ระบบสื่อสารที่เป็นวิทยุที่ต่อเป็นระบบอัตโนมัติแทนการใช้ระบบ GSM หรือระบบอินเทอร์เนท ซึ่งไม่สามารถให้บริการในพื้นที่ดังกล่าวได้ แต่จะไม่เป็นปัญหากับระบบนี้เลย และตัวเครื่องลูกยังสามารถเพื่อเติมชนิดความต้องการของ ตัวเซนเซอร์เพื่อให้เหมาะสมกับพื้นที่นั้นๆ ได้อย่างสะดวก อีกด้วยส่งผลทำให้ลดค่าใช้จ่ายได้เป็นอย่างมาก และนอกจากนี้ตัวเครื่องลูกก็ได้ถูกออกแบบให้มี ระบบแหล่งจ่ายไฟแก่ตนเอง ด้วยพลังแสงอาทิตย์ ทำให้สามารถใช้งานใน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ทุกพื้นที่ได้นั่นเอง และด้วยแนวคิดของการสื่อสารแบบใหม่โดยการส่งข้อมูลเป็นทอดๆนี้ ซึ่งเป็นเทคนิคที่ต่อยอดมาจากงานวิจัยของผู้วิจัยเอง ที่ได้รับการสนับสนุนจากกองทุนวิจัยของพระจอมเกล้าลาดกระบัง ปี 2552 ก็จะทำให้เพิ่มทั้งประสิทธิภาพและประหยัดค่าใช้จ่ายในการสร้างเครื่องตรวจวัดดังกล่าวอีกเป็นจำนวนมาก.



รูปที่ 1 แสดงรูปภาพการใช้งานของระบบเพื่อใช้ตรวจวัดสภาพสิ่งแวดล้อมธรรมชาติ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2) ทฤษฎี สมมุติฐาน และกรอบแนวความคิดของโครงการวิจัย

ในแนวคิดของการควบคุมอุปกรณ์ไฟฟ้าด้วยการส่งผ่านวิทยุก็มีการใช้งานบ้างเช่นกัน ดังในเว็ปไซด์ [1] ซึ่งต่างจากแนวคิดของโครงงานวิจัยนี้แสดงดังรูปที่ 1 โดย ตัวลูกซึ่งติดตั้งเซนเซอร์ได้มีการติดตั้งในที่ต่างๆโดยวงกลมเส้นประแสดงถึงขอบเขตของกำลังการรับส่งวิทยุของแต่ละตัวจะเห็นได้ว่าตัวแม่ (Main) มีขอบเขตสามารถติดต่อได้กับ ตัวลูกหมายเลข #1 และ #2 ทำให้สามารถรับส่งอ่านข้อมูลได้โดยตรง แต่หากตัวแม่จะอ่านข้อมูลของตัวลูกหมายเลข #3 ซึ่งมีตำแหน่งอยู่ห่างพ้นขอบเขตกำลังรับส่งของตัวแม่ออกไป ด้วยวิธีของการส่งผ่านข้อมูลของงานวิจัยนี้ทำให้การอ่านค่าก็ยังจะสามารถทำได้ โดยส่งคำสั่งอ่านค่าผ่านตัวลูกหมายเลข #2 ซึ่งมีขอบเขตกำลังส่งต่อไปถึงตัวลูกหมายเลข #3 ได้ การทำเช่นนี้ข้อเสียมีเพียงการรับส่งข้อมูลอาจช้าไปบ้าง แต่เนื่องจากข้อมูลปกติแล้วไม่มีจำนวนมาก และในส่วนของซอฟร์แวร์ก็ได้ออกแบบให้มีการการสร้างโปรโตคอลของการติดต่อสื่อสารที่กระชับและมีระบบป้องกันปัญหาการรบกวนของสัญญาณต่างๆร่วมด้วย เช่น กรณี รับข้อมูลไม่ได้ก็จะเริ่มต้น รีเซทตัวเองใหม่ เพื่อเริ่มขบวนการใหม่ ดังนี้เป็นต้น ทำให้ปัญหานี้หมดไป ยิ่งไปกว่านั้น ในวิธีการส่งข้อมูลเป็น ทอดๆ เช่นนี้ก็จะส่งผลให้ตัวลูกแต่ละตัวไม่ต้องมีกำลังส่งที่สูง จะลดการใช้แรงดันไฟฟ้าและทุนทรัพย์การลงทุนสร้าง น้อยลงไปอย่างมาก

โครงงานวิจัยมีส่วนประกอบหลักอยู่ สามส่วนคือ

- ส่วนของเครื่องตัวแม่ (Main)

-ส่วนของเครื่องตัวลูก (Client)

-ส่วนของโปรแกรมควบคุมระบบ

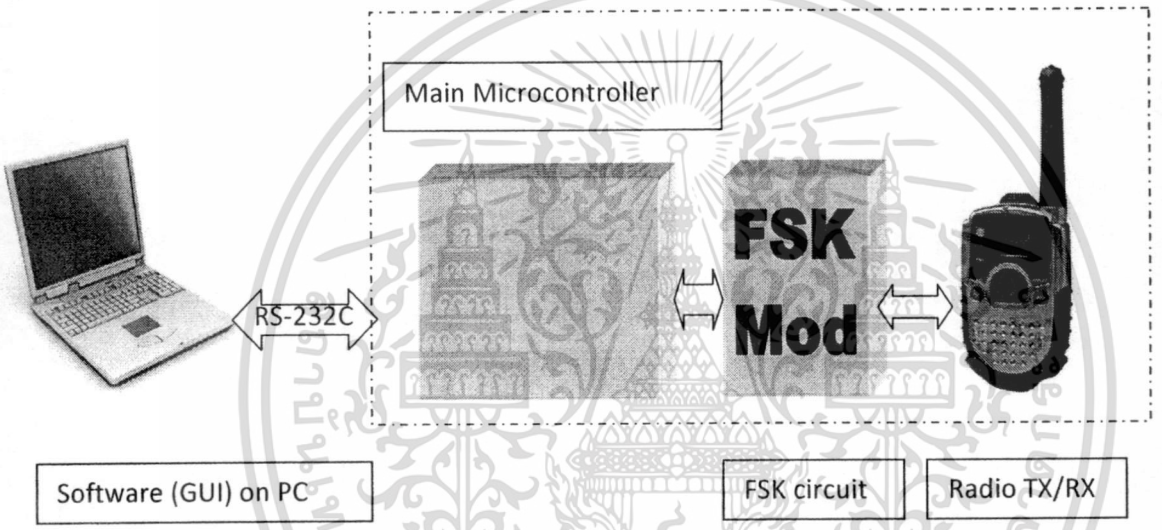
ซึ่งแต่ละส่วนก็จะได้แยกอธิบายเป็นข้อๆ ไปดังนี้

2.1) ส่วนของเครื่องตัวแม่ (Main)

นับว่าเป็นส่วนสำคัญมากเพราะเป็นส่วนที่ใช้ติดต่อกับผู้ใช้ ซึ่งหากมีการออกแบบให้ใช้งานได้ง่ายก็จะเป็นการดีมากเพราะผู้ใช้งานจะเห็นส่วนนี้เพื่อใช้งานเป็นประจำนั่นเอง ส่วนนี้จะเป็นฮาร์ดแวร์ที่ใช้ในการติดต่อส่งรับข้อมูลหรือคำสั่งจากตัวเครื่องคอมพิวเตอร์ (PC) มาสู่ตัวเองก่อนที่จะออกอากาศเป็นคลื่นวิทยุ ไปสู่ตัวเครื่องตัวลูกที่ต้องการติดต่อด้วยที่มีการบรรจุเซนเซอร์ไว้ หรือที่เรียกว่า Client หรือ Beacon นั่นเอง ซึ่งโครงสร้างแสดงได้ดังรูปที่ 2 โดยคำสั่งที่ได้จากส่วนของ โปรแกรม GUI บน PC จะ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

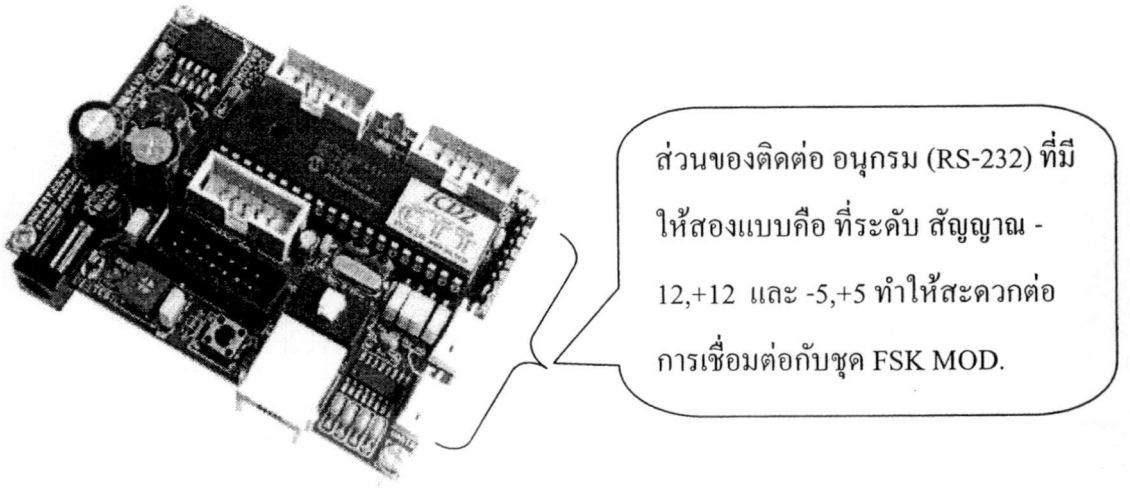
ถูกส่งผ่านช่องสัญญาณแบบอนุกรม (RS-232C) เข้าสู่ตัวเครื่องแม่ โดยจะเข้าสู่ Microcontroller (dsPIC) ที่ทำหน้าที่จัดเรียงคำสั่งใน Format ที่ถูกต้องเพื่อจะได้ส่งผ่านเข้าสู่ส่วนของ FSK MODULATOR [2] เพื่อทำหน้าที่ในการเปลี่ยนสัญญาณ ดิจิตอล ไปเป็นสัญญาณเสียง แล้วส่งสัญญาณเสียงที่ได้นี้ผ่านไปสู่วิทยุสื่อสารดังรูปที่ 2 ตัววิทยุนี้จะกระจายสัญญาณออกไปรอบทิศทางไปสู่ ตัวเครื่องลูกที่อยู่ในรัสมีการส่งต่อไป ในส่วนของตัวแม่นี้จะไม่มีเซนเซอร์ เพราะว่าเพียงทำหน้าที่ตัวกลางติดต่อกับระหว่าง PC และ Client ที่ต้องการเท่านั้น



รูปที่ 2 รูปส่วนของเครื่อง Main และการเชื่อมกับ PC

ในส่วนประกอบของ ตัวแม่นี้ ตัวควบคุมหลัก สามารถเลือกใช้เบอร์ไหนก็คงจะได้เพราะว่า ในงานวิจัยนี้ไม่ต้องการการประมวลผลที่เร็วมากนัก ทำให้สามารถเลือกได้หลายเบอร์ที่มีในท้องตลาด ผู้วิจัยเลือกใช้ dsPIC30F4011 ซึ่งให้คุณสมบัติที่ตรงที่สุด เพราะมีจำนวนของขาพอร์ทที่พอเพียง, ความเร็วที่พอเพียง และมีจุดต่อ RS-232C จำนวนสองช่อง ที่สามารถเลือกได้ว่าจะให้ผ่าน MAX232 หรือไม่ก็ได้ด้วย ทำมีการดัดแปลงบอร์ดน้อยมากเมื่อนำมาใช้กับงานวิจัยนี้ ที่สำคัญราคาถูกเพราะเป็นของบริษัทของคนไทย ที่ผลิตเองในเมืองไทย [3] ดังแสดงในรูปที่ 3

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3 ตัวบอร์ดหลัก (dsPIC 30F4011) ที่ใช้ในการควบคุมการทำงานทั้งตัวแม่และตัวลูก

ส่วนตัวของวิทยุรับส่ง สามารถใช้รุ่นไหนก็ได้ สำคัญที่ระดับกำลังงานและระบบเสาอากาศที่ใช้งาน ว่าต้องการใช้ระยะไกลของการรับส่งแค่ไหน ซึ่งหากไกลก็ต้องเสียค่าใช้จ่ายมากขึ้น นั่นเอง และอีกเรื่องหนึ่งที่สำคัญคือเรื่องของการเชื่อมต่อก็ต้องศึกษาให้ดีว่า กรณีของการกด PTT (Push To Talk) นั่นระบบเป็นเช่นไร (ในแต่ละยี่ห้อของวิทยุ จะมีลักษณะการทำงานที่ไม่เหมือนกัน) เพราะจะต้องเชื่อมโยงไปสู่การออกแบบบอร์ดใช้งานร่วมกันกับไมโครคอนโทรลเลอร์ต่อไป ในงานวิจัยนี้ผู้วิจัยเลือก แบบขนาดสูงสุด 5 Watt เป็นของ ยี่ห้อ Marshall รุ่น MS-11 ดังรูปที่ 4

รูปที่ 4 รูปของวิทยุสื่อสารที่ใช้ในงานวิจัยนี้ (Marshall MS-11)

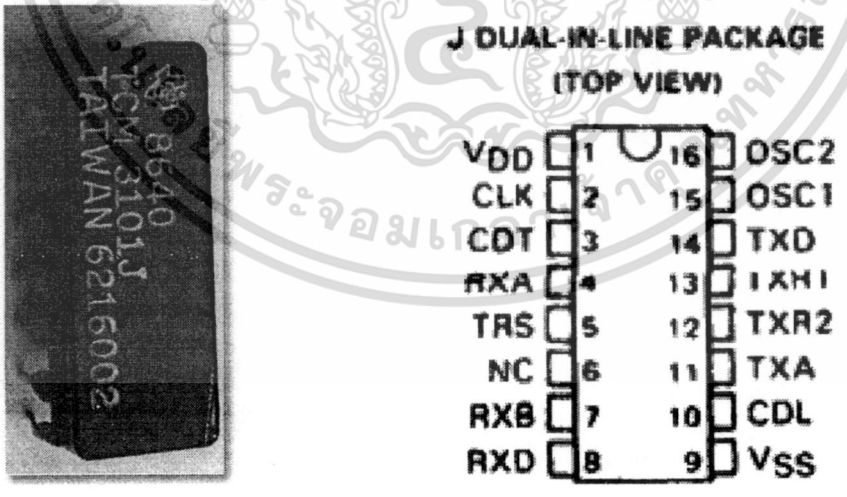
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ซึ่งลักษณะของการต่อเชื่อมส่วน PTT จะเป็นการ ต่อสัญญาณ เสียงที่ได้จาก OUTPUT ของ FSK MODULATOR ที่จะเข้าสู่ส่วน MIC ของ วิทยุสื่อสารหนึ่งขา ลง กราวด์ (ซึ่งแสดงให้เห็นได้จาก ภาคผนวก)

2.1.1 ส่วนของ FSK MODULATOR

ส่วนนี้นับเป็นส่วนที่สำคัญมาก แทบจะเรียกได้ว่าเป็นหัวใจของงานวิจัยนี้เลย เพราะใช้เพื่อการแปลงสัญญาณดิจิตอล ซึ่งก็คือสัญญาณแบบอนุกรมที่ได้ออกมาจากคอมพิวเตอร์ให้ไปเป็นแบบสัญญาณอนาล็อก หรือสัญญาณเสียง (สองระดับเสียง) เพื่อจะได้สามารถใช้งาน วิทยุสื่อสาร ส่งสัญญาณกระจายออกไปสู่เครื่องรับได้ที่ระยะไกลๆ

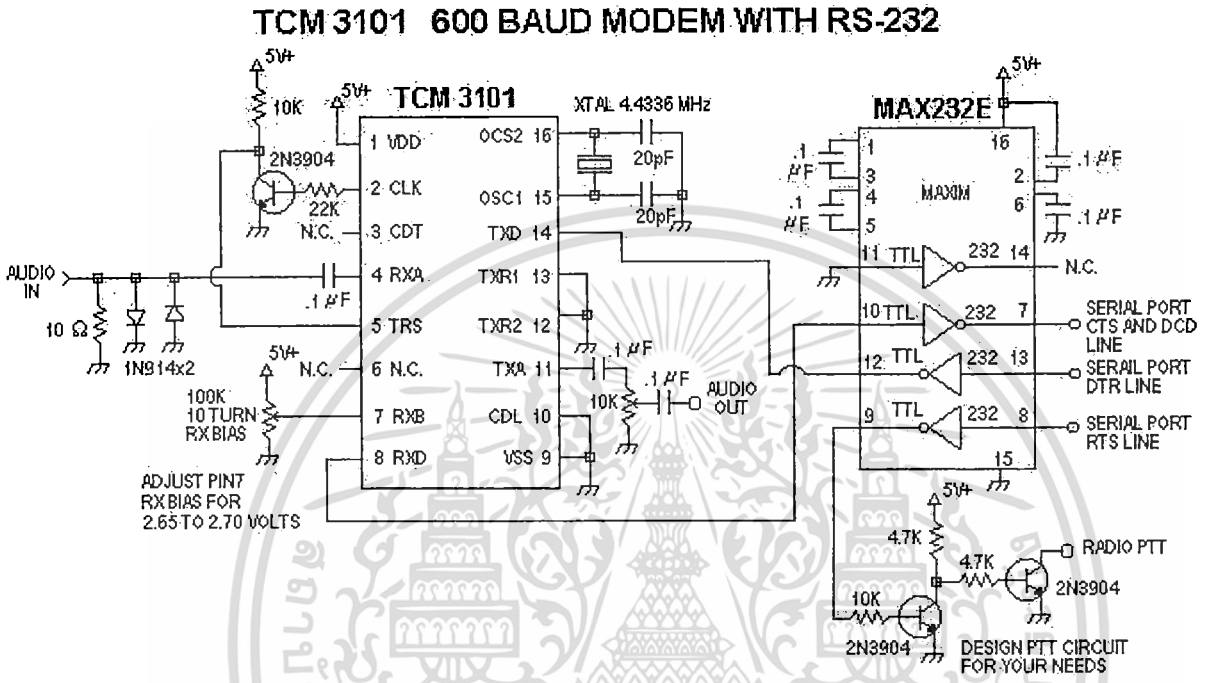
ในงานวิจัยนี้ได้เลือกใช้ตัว ไอซีสำเร็จรูป เพราะต้องการใช้มีขนาดที่เล็กและใช้แรงไฟน้อยที่สุด ซึ่งก็ได้บทสรุปคือเบอร์ TCM3101 หรือ TCM3015 เป็นของบริษัท Texas Instrument [4] ซึ่งได้มีการผลิตและนำมาใช้งานนานแล้ว และบริษัทก็ได้เลิกผลิตแล้วในปัจจุบัน แต่ ผู้วิจัยจึงได้สอบถามศูนย์จำหน่ายก็ยังมีเหลือให้สามารถใช้ได้จำนวนหนึ่ง จึงสามารถดำเนินการวิจัยต่อไปได้ รูปลักษณะของไอซี แสดงให้เห็นดังรูปที่ 5



รูปที่ 5 แสดงรูปของไอซี สำเร็จรูป FSK MODULATOR

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตัวไอซีนี้มีการใช้งานที่สะดวก เพราะได้รวมเอาส่วนต่างๆของ ชุดเข้ารหัสและถอดระหัส แบบ FSK (Frequency Shift Keying) ไว้ในตัวเดียวเลย ทำให้มีการต่อร่วมกับ อุปกรณ์อิเล็กทรอนิกส์อีกเพียงเล็กน้อยก็สามารถใช้งานได้เลย วงจรมาตฐานที่เป็นต้นแบบของการต่อใช้งานในงานวิจัยนี้ก็แสดงให้เห็นดังรูปที่ 6



รูปที่ 6 แสดงวงจรมาตฐานของการใช้งาน TCM3101, TCM3105

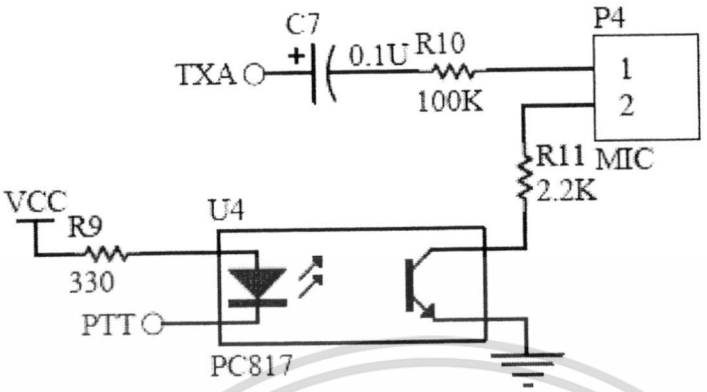
จากรูปวงจร ส่วนสำคัญคือการปรับให้ค่าของ TXA และ RXB ให้มีการรับส่ง อัตราบิทต่อวินาที ที่ถูกต้องที่สุด ซึ่ง สามารถทำได้โดยการป้อนสัญญาณสี่เหลี่ยมขนาด 600 Hz เข้าสู่วงจรภาคส่ง และใช้ออสซิลโลสโคป จับสัญญาณ ดิจิตอลออกที่ภาครับ ปรับแต่งค่าให้มีการผิดเพี้ยนของสัญญาณน้อยที่สุด ซึ่งในงานวิจัยนี้ได้ใช้ความถี่ในการรับส่งข้อมูลกันใน อัตรา 600 bps

ส่วนของการเชื่อมต่อกับไมโครคอนโทรลเลอร์จะใช้ช่องสัญญาณดิจิตอลพอร์ทอินพุทเอ้าพุท จำนวน 1 บิท คือ บิทที่ใช้สำหรับการการควบคุมการเปิดหรือปิด PTT (Push To Talk) และอีกหนึ่งช่องสำหรับการเชื่อมต่อข้อมูลแบบอนุกรม คือใช้ UART1 หรับเชื่อมต่อกับ ไมโครคอมพิวเตอร์ เพื่อรับคำสั่งจากผู้ใช้ และ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

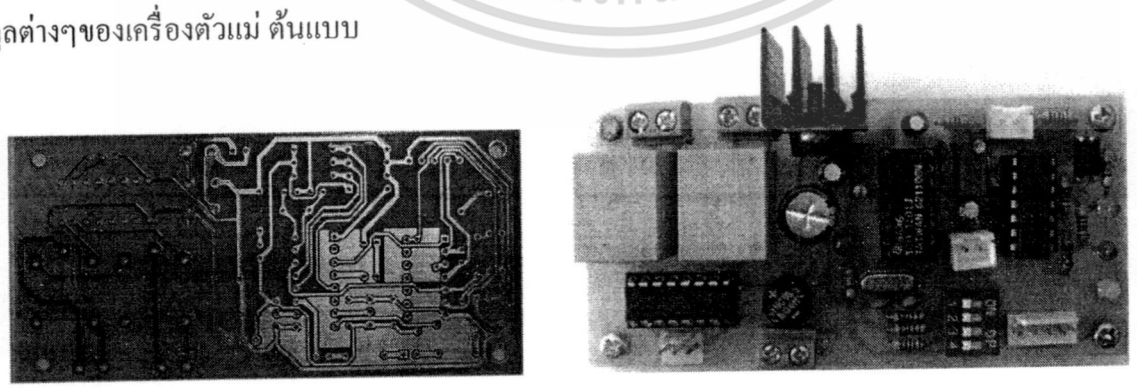
ช่อง UART2 สำหรับการเชื่อมต่อกับชุดส่วน SFK MODULATOR รูปแสดงการต่อวงจรควบคุม PTT สามารถทำได้หลายแบบ และแบบที่มีขนาดเล็กสามารถใช้งานได้ดีก็แสดงดังรูปที่ 7 ข้างล่างนี้



รูปที่ 7 วงจรใช้งาน เปิดปิด ส่วนของ PTT

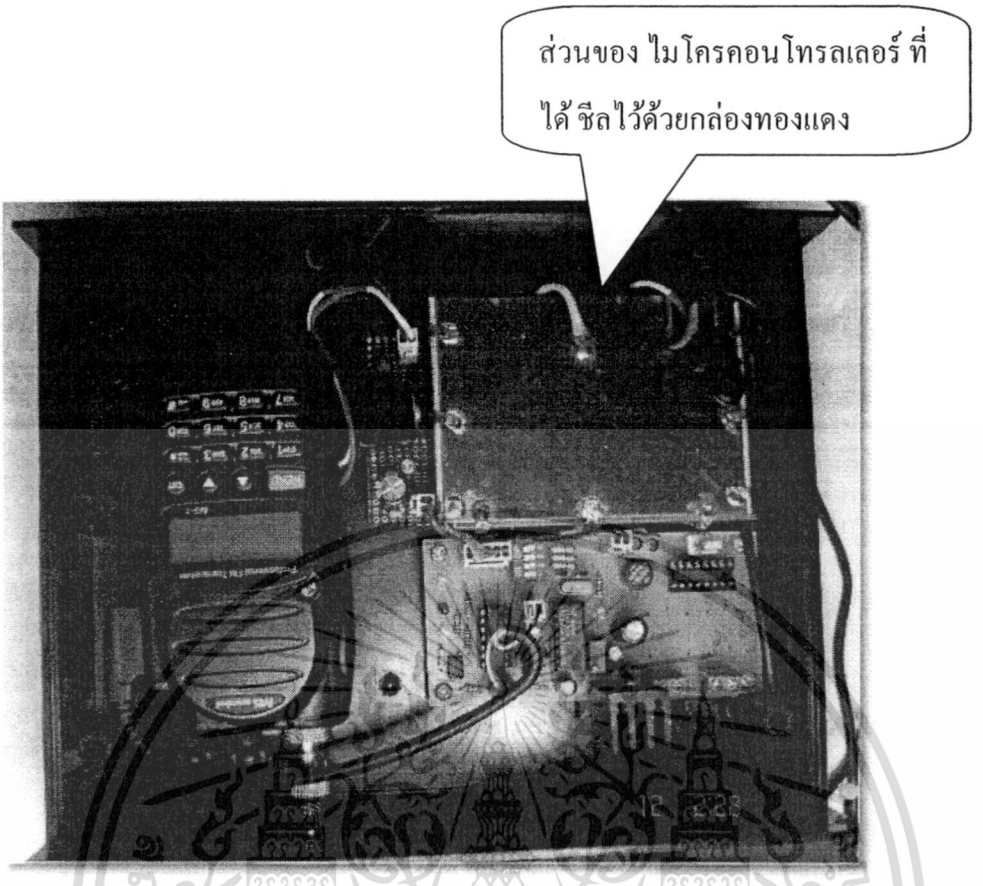
ซึ่งสามารถใช้ อ็อปโต้ไอโซเลเตอร์ขนาด 1 บิท เบอร์ PIC218 ได้ (ขนาดเล็กและราคาไม่แพง) เป็นตัวตัดต่อต่อ สัญญาณไฟฟ้าลงกราวด์ ตามสัญญาณดิจิตอลขนาด 1 บิท จากไมโครคอนโทรลเลอร์ที่สั่งมาควบคุมอีกทีหนึ่ง

ในการออกแบบผู้วิจัยได้รวมส่วนของวงจร FSK MOD และส่วนควบคุม PTT พร้อมทั้งส่วนการควบคุมรีเลย์ (สำหรับตัวลูก) ไว้ในบอร์ดเดียวกัน เพื่อทำให้มีขนาดที่เล็กที่สุด และสามารถนำไปใช้กับ เครื่องตัวลูกได้ด้วย โดยไม่ต้องมีการออกแบบใหม่ (เพียงตัวแม่ไม่ได้ใช้รีเลย์) รูปของแผ่นวงจรพิมพ์ที่ได้ออกแบบไว้แล้วแสดงดังรูปที่ 8 เป็นแผ่นวงจรพิมพ์ที่เป็นแบบหน้าเดียว และในรูปที่ 9 แสดงถึงการเชื่อมต่อกันของโมดูลต่างๆของเครื่องตัวแม่ ต้นแบบ

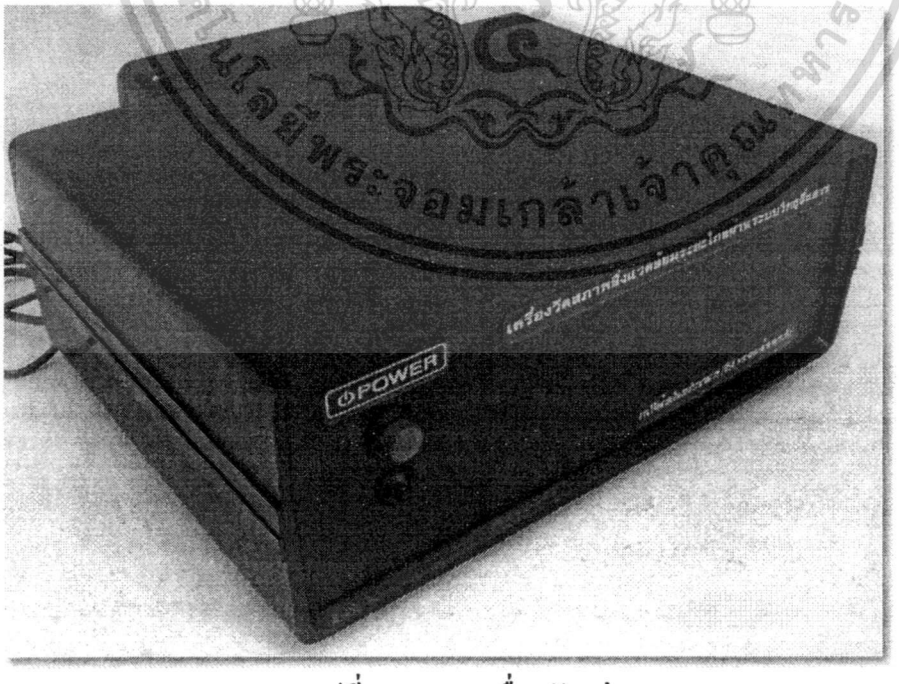


รูปที่ 8 แสดงแผ่นวงจรพิมพ์ส่วน FSK MOD และ PTT ของงานวิจัยนี้

121391
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 9 รูปการต่อรวมโมดูลต่างๆของเครื่องตัวแม่ต้นแบบ

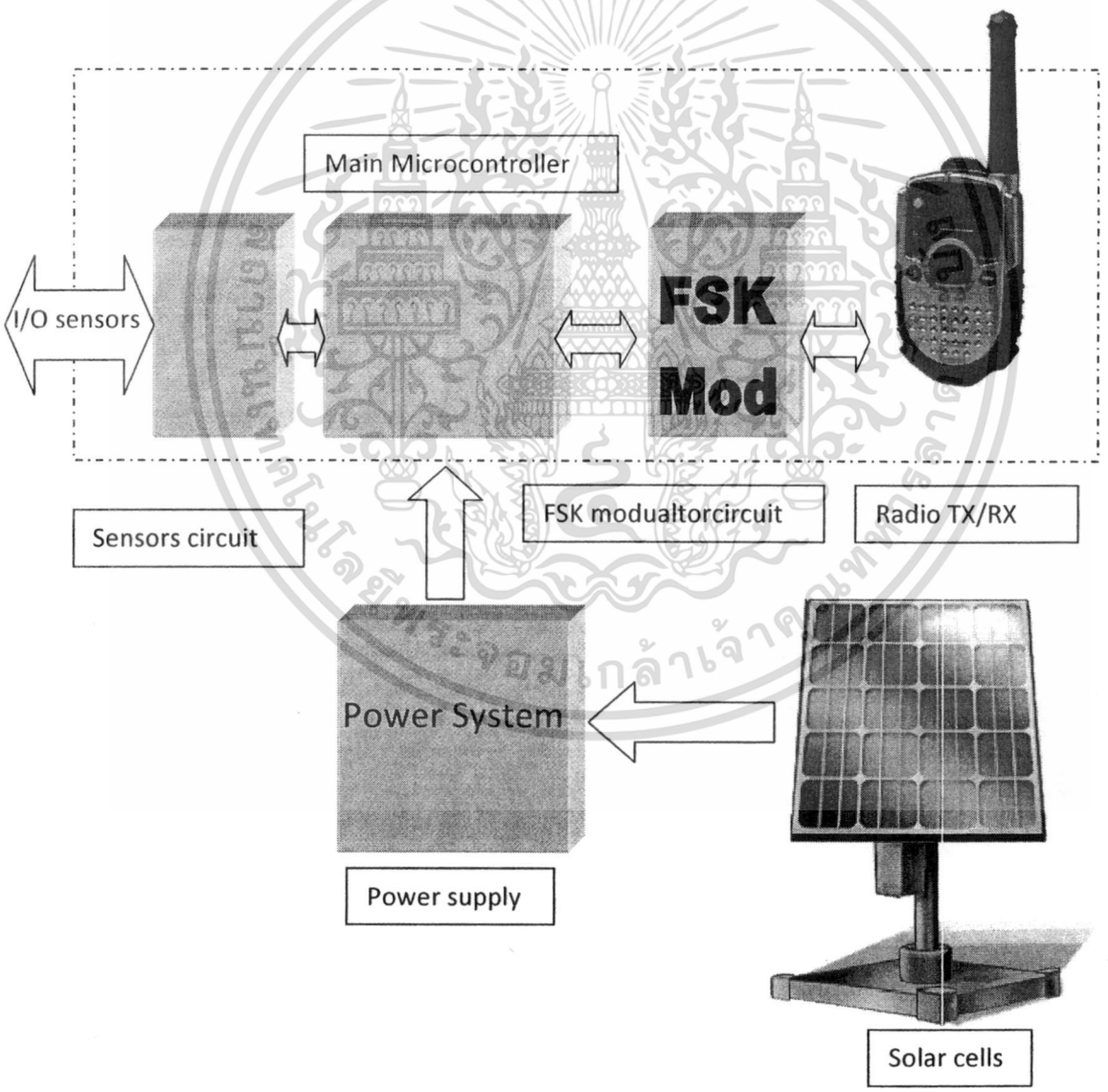


รูปที่ 10 แสดง เครื่องตัวแม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2) ส่วนของ เครื่องตัวลูก (Client)

ส่วนเครื่องตัวลูกนี้จะเป็นเครื่องที่มีการติดตั้งเซนเซอร์ต่างๆเช่น เซนเซอร์ตรวจจับ อุณหภูมิ เป็นต้น แต่นอกจากจะติดตั้งเซนเซอร์ต่างๆแล้วยังสามารถติดตั้งอุปกรณ์อื่นเช่น สวิทช์รีเลย์ได้ออกแบบลายวงจรไว้แล้ว ทั้งนี้ก็แล้วแต่การนำไปใช้งานต่างๆนั้นเอง และเช่นเดียวกันในรูปที่ 11 แสดงให้เห็นว่าจะมี FSK MODULATOR แบบเดียวกันกับส่วนของตัวแม่เพื่อใช้ในการติดต่อสื่อสารกัน และส่วนควบคุมการทำงานก็จะเป็น ไมโครคอนโทรลเลอร์แบบเดียวกับตัวแม่ และส่วนที่สำคัญที่เพิ่มมาอีกส่วนหนึ่งก็คือส่วนจัดการระบบจ่ายไฟเลี้ยงแก่ตัวเครื่อง จะเป็นลักษณะของการเก็บพลังงานจากแผงรับแสงอาทิตย์เก็บเป็นกำลังงานสำรองไว้ ให้สามารถใช้งานได้ต่อเนื่องตลอดไป หรือแม้ไม่มีแสงอาทิตย์ก็ให้สามารถทำงานได้อย่างน้อย หนึ่งวันหรือมากกว่า

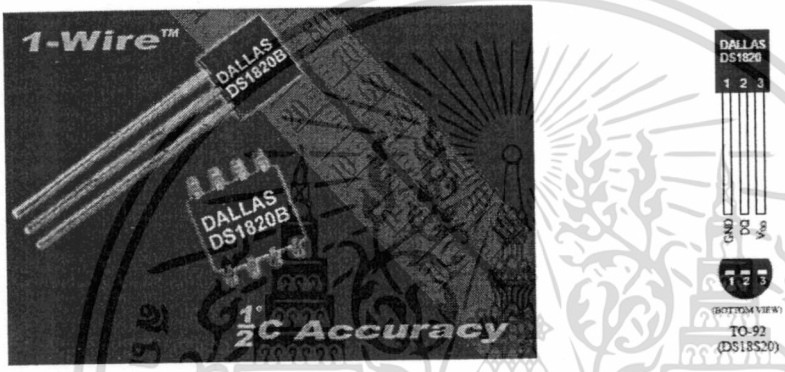


รูปที่ 11 สแดงรูปโครงสร้างของเครื่อง Client

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

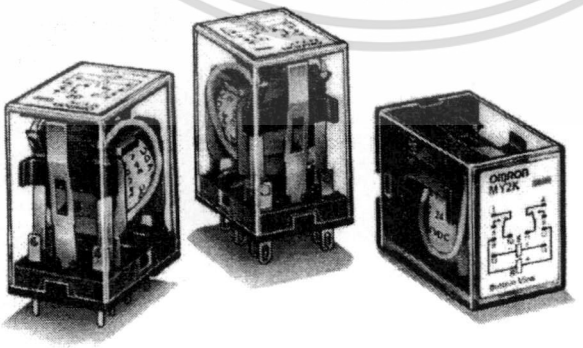
2.2.1 โมดูลตรวจจับอุณหภูมิ DS1820

ในงานวิจัยนี้ได้ใช้โมดูลสำเร็จรูป (ไอซีขนาด 3 ขา) เบอร์ DS1820 มาใช้ในการตรวจวัดอุณหภูมิ ด้วยเหตุผลที่ว่า มีขนาดที่เล็ก ใช้ไฟเลี้ยงไม่มาก และตัวโมดูลมีลักษณะเป็นพลาสติกที่เราสามารถติดตั้งบนตัวเครื่องได้ง่าย ทนต่อความร้อน ต่างๆ เพราะต้องนำไปใช้ในพื้นที่ป่าเขา สามารถโปรแกรมกำหนดการทำงานได้ทั้งแบบ อุณหภูมิที่ความละเอียดต่ำและสูง (ขึ้นอยู่กับการใช้งาน) หากตั้งไว้ที่ความระเอียดต่ำก็จะมีการเข้าถึงอ่านข้อมูลได้รวดเร็วมาก รูปร่างของตัวไอซีนี้ แสดงให้เห็นใน รูปที่ 12



รูปที่ 12 แสดงรูปตัวไอซี DS1820 ที่ใช้วัดอุณหภูมิ

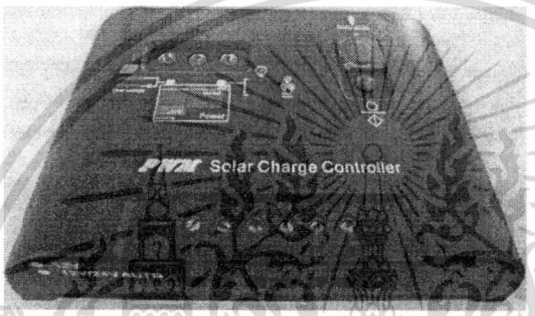
ในส่วนของตัวลูกนี้จะมีโครงสร้างเหมือนกันกับส่วนของตัวแม่ทุกประการ หากแต่ส่วนที่แตกต่างกันก็คือ มีการติดตั้งรีเลย์บนแผ่นวงจรพิมพ์ และติดตั้งตัววัดอุณหภูมิ ไว้แล้วและส่วนของซอฟร์แวร์ก็จะแตกต่างกันกับตัวแม่เล็กน้อย



รูปที่ 13 ตัวรีเลย์ที่ใช้ในงานวิจัยเป็นแบบ 5 VDC ขนาดเล็ก (ต้นแบบ)

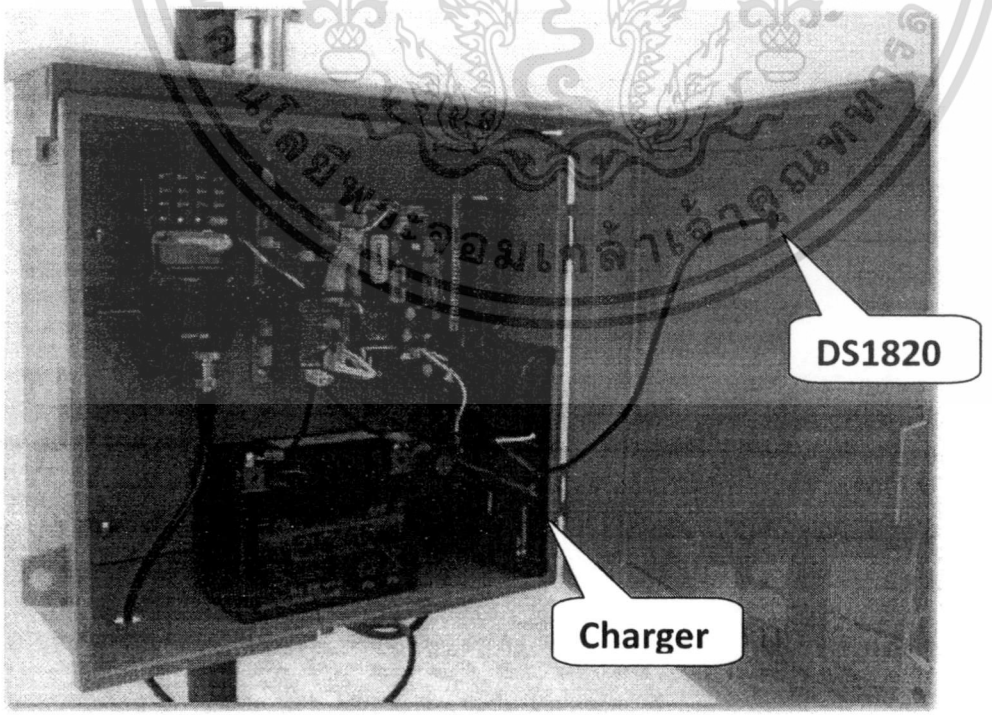
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ส่วนของภาคจ่ายไฟด้วยพลังแสงอาทิตย์ นั้นผู้วิจัยได้ใช้ โมดูลสำเร็จรูป เพราะเป็นส่วนที่ไม่คุ้มกับการสร้างใหม่ เนื่องจากต้องการเพียงคุณสมบัติที่ สามารถประจุแบตเตอร์รี่ให้มีระดับค่อนข้างเต็มตลอด ซึ่งเป็นคุณสมบัติที่มีอยู่แล้วในเครื่องชาร์จแบตเตอร์รี่จากพลังงานแผงแสงอาทิตย์ทั่วไป นอกจากนี้ ราคาก็ไม่แพงนักในคุณสมบัติที่ต้องการ ผู้วิจัยเลือกใช้ รุ่น CNP-124-20AT แสดงดังรูปที่ 14 ส่วนเรื่องของตัวแบตเตอร์รี่ ก็ได้ทดลองกับ แบตเตอร์รี่แห้งก้อนขนาด 7 AH ซึ่งเมื่อทดลองถึงระยะที่สามารถทำงานได้แล้วว่าจะนานเท่าใด ก็จะสามารถคำนวณได้ว่าจะต้องเพิ่มขนาดไปอีกเท่าใดเพื่อจะให้ได้ทำงานนานอย่างน้อย 3 วัน โดยไม่ต้องชาร์จแรงไฟจากแสดงอาทิตย์



รูปที่ 14 แสดงรูปตัวโมดูลการชาร์จไฟอัตโนมัติที่เลือกใช้งาน

และในรูปที่ 15 ก็แสดงให้เห็นถึงตัวลูกที่ประกอบไว้แล้วเพื่อการทดลอง



รูปที่ 15 ตัวเครื่องตัวลูกที่ประกอบเสร็จเพื่อทดลองใช้งาน

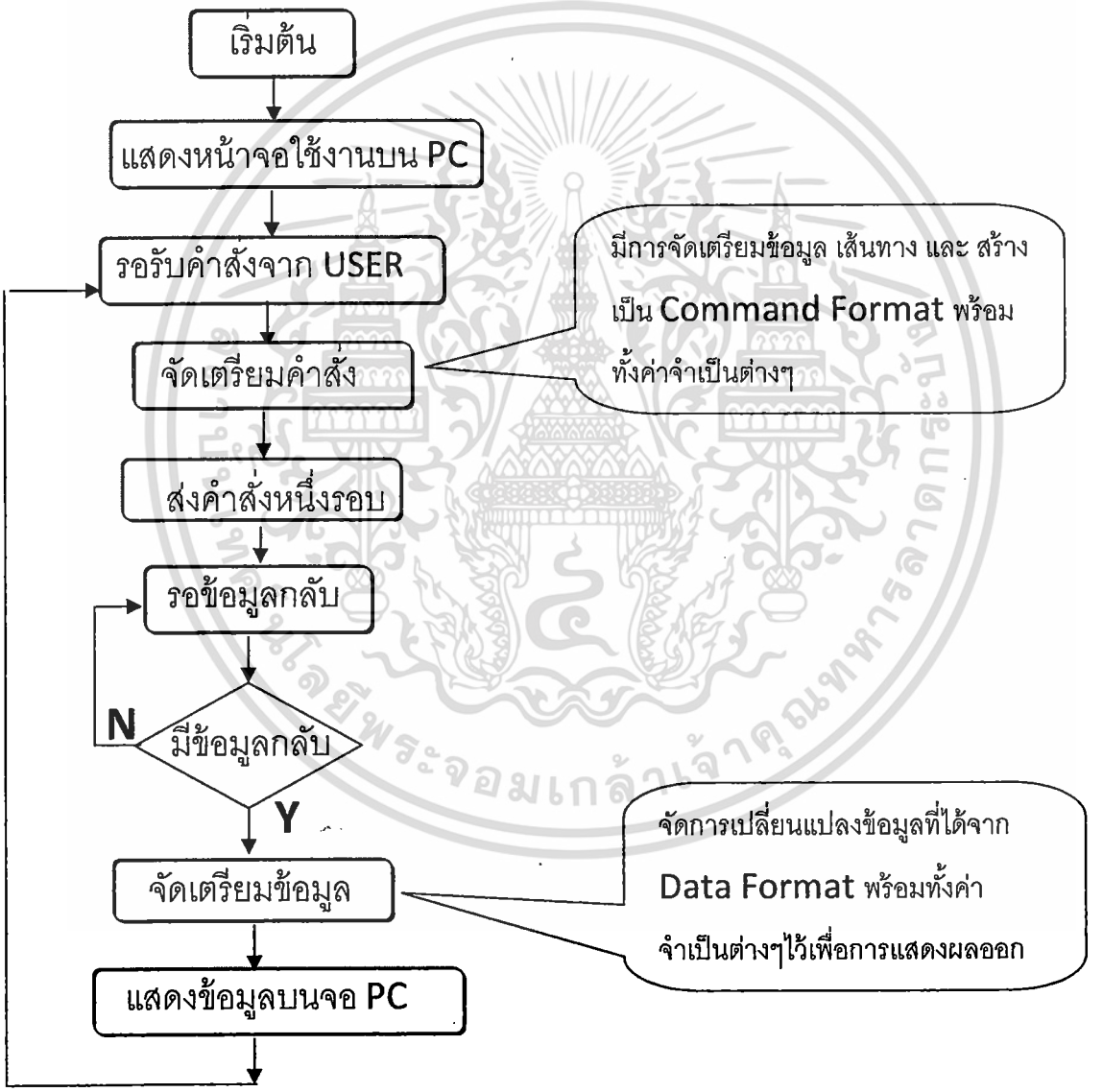
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3) ส่วนของซอฟร์แวร์

ส่วนนี้จะมีออกแบบไว้ สามโปรแกรม คือ

3.1 ซอฟร์แวร์ส่วนของบนเครื่อง PC

ในส่วนนี้เป็นการเขียนโปรแกรมที่ติดตั้งไว้บน PC เป็นแบบ GUI และสามารถทำงานบนเครื่อง PC ที่เป็นระบบ Windows ทั่วไป ทั้งนี้เพื่อให้ง่ายต่อการใช้งานมากที่สุด โดยมีลักษณะของการทำงาน แสดงเป็นโฟร์ชาร์ท ได้ดังรูปที่ 16 โปรแกรมส่วนนี้จะใช้ ภาษา C# ในการสร้าง [5][6].

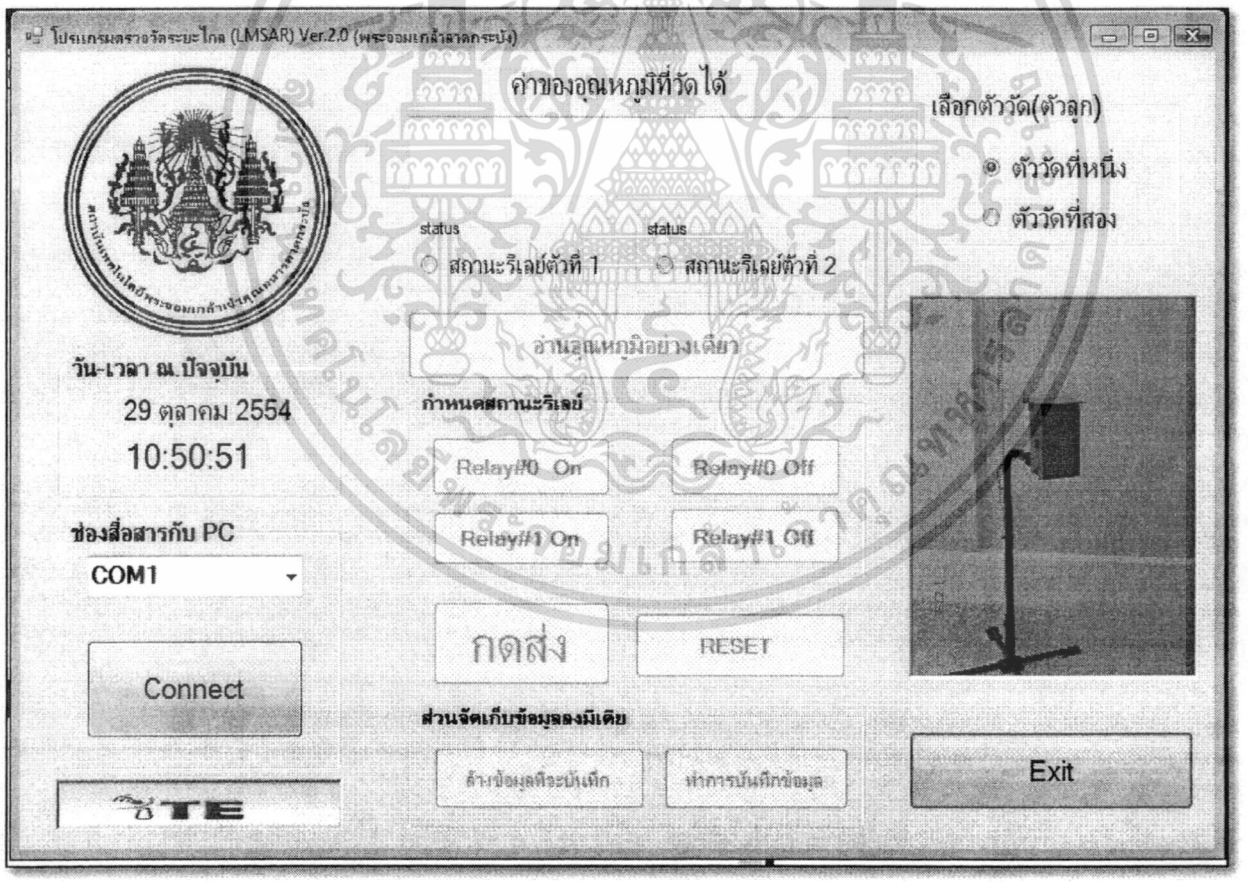


รูปที่ 16 สแดงโฟร์ชาร์ท การทำงานภาพรวมของส่วนโปรแกรม GUI บน PC

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เริ่มจาก ตัวโปรแกรมจะแสดงภาพส่วน GUI บนจอมอนิเตอร์ดังรูปที่ 17 เพื่อรับคำสั่งต่างๆจากผู้ใช้ เช่น ต้องการอ่านค่าอุณหภูมิ หรือเลือกการสั่งรีเลย์ตัวที่ต้องการตัวใดให้ ON หรือ OFF เหล่านี้เป็นต้น เมื่อตัวโปรแกรมรับคำสั่งที่ต้องการให้ทำแล้ว ก็จะจัดเรียงรูปแบบคำสั่งให้ถูกต้อง ก่อนส่งออกไปที่ตัวเครื่องตัวแม่เพื่อผ่านสัญญาณวิทยุต่อไป จากนั้นตัวโปรแกรมก็จะทำการรอผลจากตัวแม่ และเมื่อตัวแม่ได้รับข้อมูลอุณหภูมิจากตัวลูกที่ส่งคำสั่งไปอ่านค่าต่างๆกลับมาแล้ว ก็จะส่งข้อมูลต่อไปสู่ คอมพิวเตอร์ของผู้ใช้ผ่านช่องสื่อสารอนุกรม เพื่อจัดแสดงผล ออกบนจอภาพมอนิเตอร์ต่อไป

ดังนั้น เมื่อคอมพิวเตอร์ได้รับข้อมูลกลับมาแล้วจากตัวเครื่องแม่ ตัวโปรแกรมก็จะทำการแยกแยะส่วนของข้อมูลอุณหภูมิและส่วนของสถานะรีเลย์ที่ได้ทำการส่งคำสั่งอ่านค่ามา นำไปออกแสดงได้อย่างถูกต้องบน GUI (จอมอนิเตอร์)



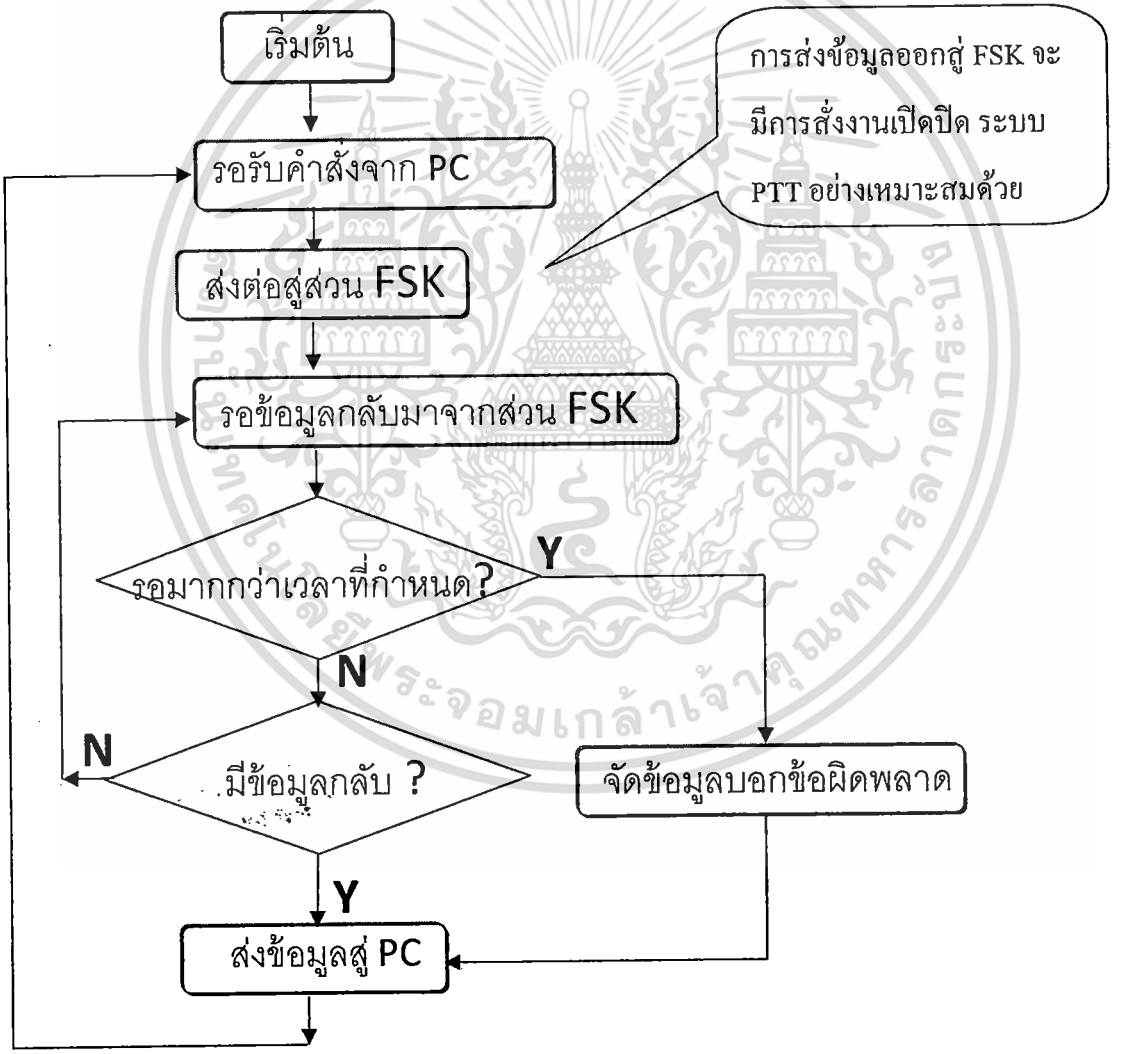
รูปที่ 17 แสดงหน้าจอแบบ GUI ที่ใช้รับข้อมูลคำสั่งจากผู้ใช้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.2 ซอฟร์แวร์ส่วนของตัวแม่

เป็นโปรแกรมที่ติดตั้งบนเครื่องตัวแม่ ที่ด้านหนึ่งติดต่อกับ คอมพิวเตอร์ของผู้ใช้และอีกด้านหนึ่งจะส่งหรือรับผ่านข้อมูลหรือคำสั่งกับเครื่องรับส่งวิทยุสื่อสาร ดังนั้น หน้าที่หลักๆคือ รับคำสั่งที่ได้จากเครื่องคอมพิวเตอร์ที่ถูกสั่งมาจากผู้ใช้อีกที มาจัดรูปแบบ (Format) ที่เหมาะสมผ่านส่วน FSK MOD ส่งต่อไปสู่วิทยุสื่อสาร หรือในทางกลับกัน จะรับสัญญาณข้อมูลที่กลับมาจากตัวลูกเปลี่ยนกลับเป็นสัญญาณดิจิตอลส่งไปสู่ คอมพิวเตอร์ของผู้ใช้เพื่อการแสดงผล และจัดเก็บข้อมูลต่อไป

ดังนั้น พอจะแสดง ไดอะแกรมได้ดังรูปที่ 18



รูปที่ 18 แสดงไดอะแกรมการรับส่งข้อมูล/คำสั่ง ผ่านตัวเครื่องแม่

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เนื่องจากว่า ในการส่งคำสั่งไปสู่ตัวลูกนั้น หากมีเหตุ อันใดที่ไม่สามารถทำงานได้ เช่น ตัวลูกตัวที่ต้องการติดต่อด้วยเสียหาย ไม่มีการส่งข้อมูลกลับ หรือ ข้อมูลไม่มีกลับมาในกรณีใดก็ตาม ก็จะทำให้การรอข้อมูลจะไม่มีสิ้นสุด ดังนั้นจึงมีส่วนของการปรับเวลาการรอ สัญญาณกลับไว้กับเครื่องตัวแม่ ซึ่งจากการทดลองจริง ก็มีข้อสรุปว่า สัญญาณจะมีการตอบกลับมาในสภาวะปกติ ช่วงละไม่เกิน 20 วินาที (คำว่า ช่วงละ หมายถึงการรับส่งข้อมูลผ่านหนึ่งทอด เท่านั้น) และในการทดลองของงานวิจัยนี้ได้ทำการทดลองมากสุดคือ สองช่วง ดังนั้น ระยะเวลาการรอสัญญาณกลับก็จะไม่เกิน 40 วินาที หากมีการรอที่มากกว่าเวลานี้ ก็ประมาณได้ว่ามีการเสียหายของอุปกรณ์แล้ว จึงกำหนดให้ตัวลูกมีการ รีเซทตัวเองรอเริ่มต้นทำงานใหม่ และให้ตัวแม่ ส่งข้อความกลับสู่ ผู้ใช้แสดงผลออก คอมพิวเตอร์ ให้ผู้ใช้ได้รับรู้ถึงความผิดพลาดต่อไป

3.3 ซอฟร์แวร์ส่วนของตัวลูก

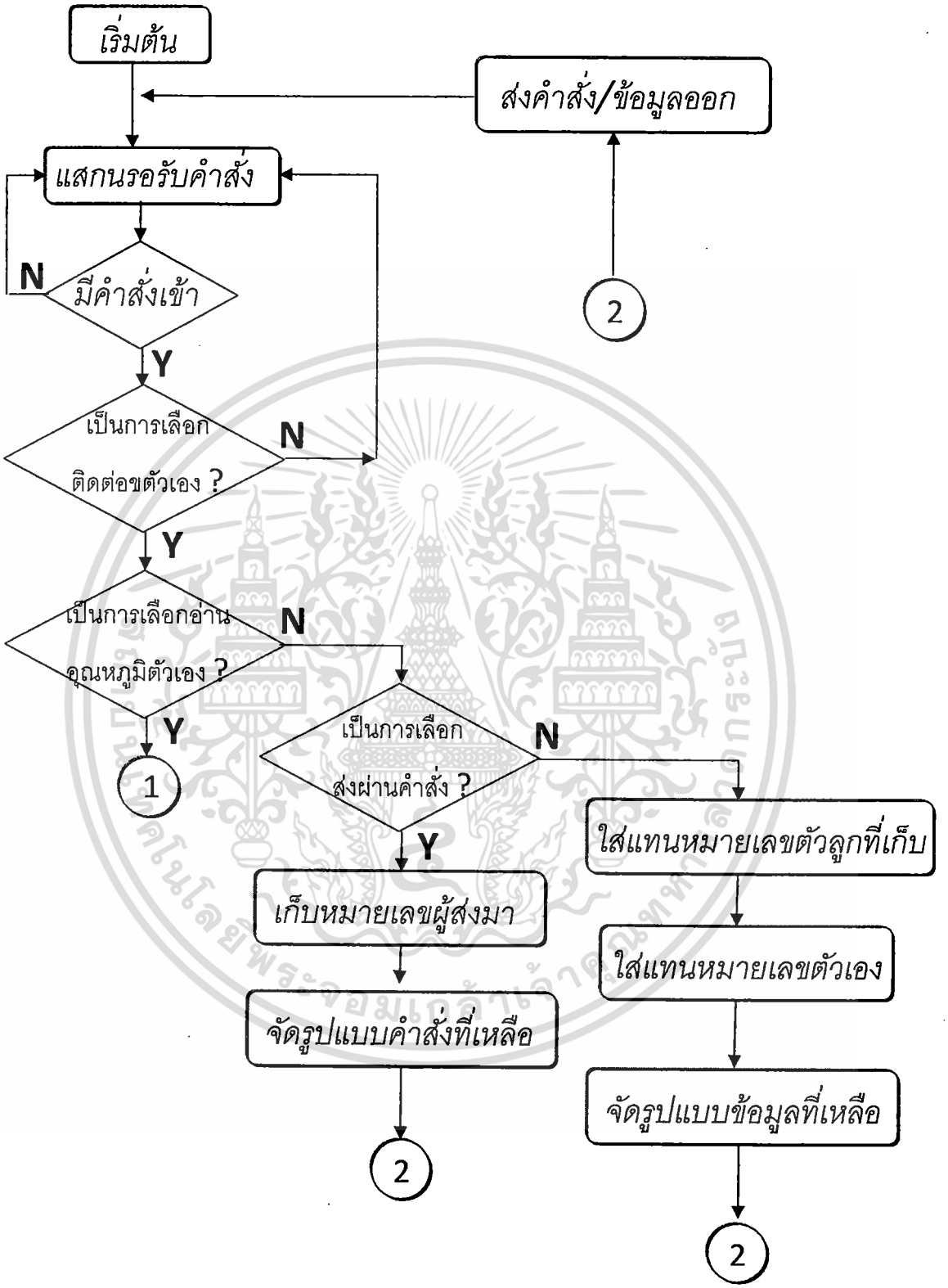
ในส่วนของซอฟร์แวร์ของตัวลูกนี้ จะรับคำสั่งที่ได้จากตัวแม่ (ผ่านระบบวิทยุ) จากนั้นก็แยกแยะว่าเป็นการติดต่อกับตัวเองหรือไม่ หากใช่ก็จะแยกแยะคำสั่งต่อไปอีกว่า มีการสั่งให้อ่านค่าอุณหภูมิหรือสั่งให้ รีเลย์ตัวที่ศูนย์หรือตัวที่หนึ่งทำงานในลักษณะใด (เป็นสั่งให้รีเลย์ ON หรือ OFF หรือจะเป็นการอ่านสถานะของรีเลย์ว่าอยู่ในสถานะใดเท่านั้น) ฉะนั้นตัวลูกแต่ละตัวหากมีสัญญาณมาถึงตัวเองและแยกแยะสัญญาณดูแล้วว่าเป็นการติดต่อกับตัวเองก็จะมีกรณีที่ต้องปฏิบัติดังนี้

กรณีส่งผ่านคำสั่งหรือข้อมูลต่อ หมายถึงตัวลูกที่รับสัญญาณแล้ว และเมื่อตรวจสอบคำสั่งดูแล้วเป็นคำสั่งที่ต้องการส่งต่อคำสั่งไปหาตัวลูกตัวอื่น หรือเป็นการส่งกลับข้อมูลไปหาตัวลูก/แม่ ตัวอื่นก็จะทำหน้าที่ส่งผ่าน คำสั่งหรือข้อมูล ดังแสดงใน ไดอะแกรม รูปที่ 19

กรณีอ่านค่าของอุณหภูมิของตัวเอง หมายถึงตัวลูกตัวเป้าหมาย เมื่อได้รับสัญญาณและแยกแยะดูแล้วว่าเป็นคำสั่งที่ต้องการ การอ่านข้อมูลอุณหภูมิ และกำหนดการทำงานของ รีเลย์ของตนเอง ก็จะทำการอ่านอุณหภูมิและกำหนดค่ารีเลย์ตามคำสั่งนั้น จากนั้นก็ส่งข้อมูลอุณหภูมิ และสถานะรีเลย์นั้น กลับไปตามเส้นทางเดิมที่ผ่านมา (เส้นทางของคำสั่ง) ดังไดอะแกรมของ รูปที่ 20

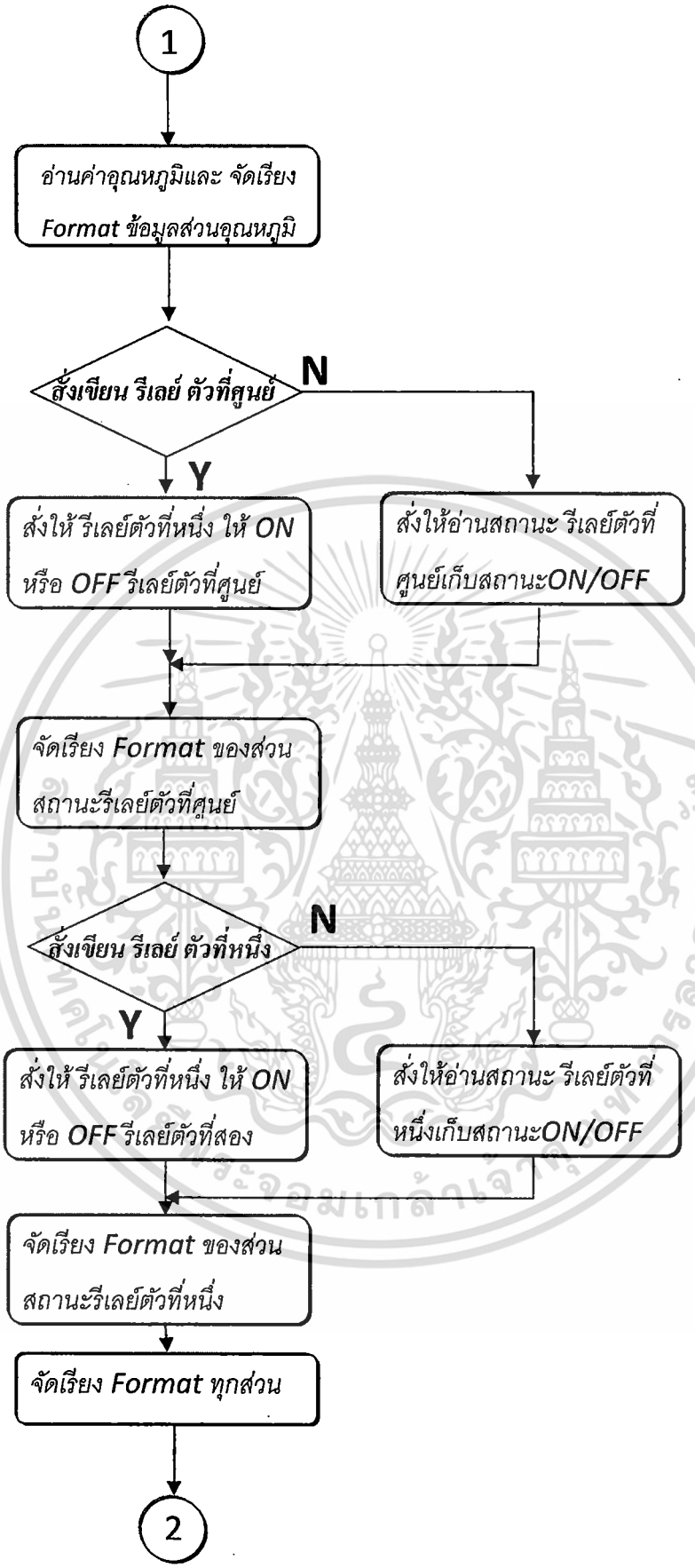
ซึ่งจะเห็นได้ว่า การทำงานของตัวลูกแต่ละตัวจะต้องรู้ว่าตนต้องทำงานรูปแบบใด อันนี้สามารถทำได้โดยการกำหนด ให้มีระหัสสถานะการทำงานเข้าไปใน รูปแบบของคำสั่ง ที่สร้างได้จากส่วนของโปรแกรม GUI ซึ่ง ระหัส ต่างๆ จะถูกกำหนดไว้ ตายตัวแล้วตามรูปแบบที่ จะได้อธิบายต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 19 แสดงโฟร์ชาร์ท การทำงานภาพรวมของโปรแกรมบนตัวลูก (กรณีส่งผ่าน)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 20 แสดงโฟร์ชาร์ท การทำงานภาพรวมของโปรแกรมบนตัวลูก (กรณีถูกเลือกอ่านอุณหภูมิ)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

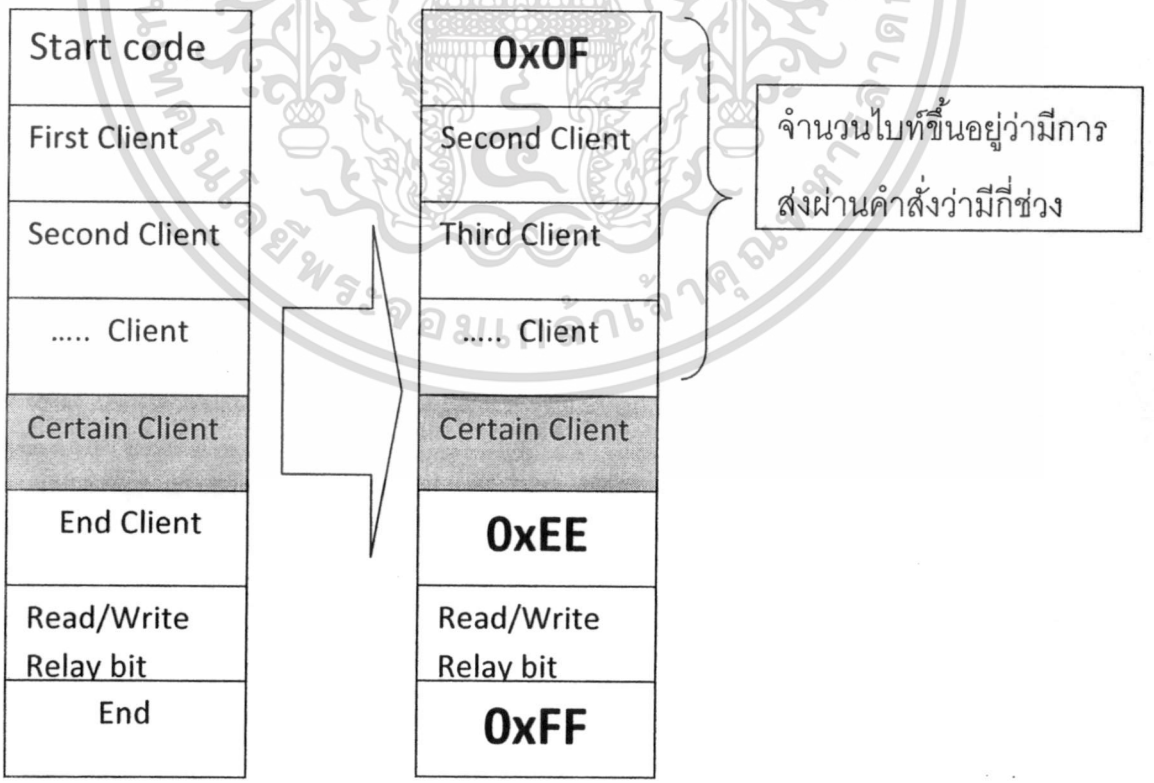
3.4 รูปแบบของคำสั่งและข้อมูล (Command & Data format)

ดังในหัวข้อที่ 3 เรื่องของ ซอฟร์แวร์ที่ได้กล่าวมา ระหัสในการตรวจสอบเพื่อกำหนดการทำงานของตัวลูกแต่ละตัว นับเป็นสิ่งสำคัญเป็นอย่างยิ่ง ทั้งนี้เพื่อให้ได้ผลทั้งความปลอดภัยของข้อมูล, ป้องกันสัญญาณรบกวนได้ในระดับหนึ่ง ทั้งนี้ เพราะในรูปแบบของคำสั่งหรือข้อมูลที่รับส่งกันนั้น จะมีการตรวจสอบว่าสัญญาณนั้นๆ เป็นสัญญาณที่ใช้ในการรับส่งจริงหรือเปล่าหรือเป็นสัญญาณรบกวนจาก เครื่อง ประชาชนทั่วไป วิธีการก็คือ ใช้ ระหัส เริ่มต้นขนาดหนึ่งไบท์ของทุก คำสั่งหรือข้อมูล ด้วย ระหัสที่ไม่มีการใช้งานในงานวิจัย เช่น ระหัส 0x0F และมี ระหัสที่ปิดบอกว่าหมดคำสั่งหรือข้อมูลด้วยคือ ระหัส 0xFF ดังนี้แล้วหากตัวลูกหรือตัวแม่มีการตรวจสอบระหัสเหล่านี้ก่อนก็จะทราบได้ว่าเป็น สัญญาณที่ใช้งานและทราบถึงจำนวนไบท์ทั้งหมดของ คำสั่งหรือข้อมูล ได้นั่นเอง ทีนี้คำสั่งและข้อมูลมีการทำงานเช่นไร จะได้กล่าวต่อไป โดยจะมีการยกตัวอย่างประกอบไปด้วย ดังนี้

3.4.1 รูปแบบคำสั่ง (Command Format)

เป็นรูปแบบที่ใช้ในการส่งจาก ตัว PC ผ่านเครื่องตัวแม่ (Main) ไปสู่แต่ละช่วงของตัวลูก (Clients) ที่เป็นตัวส่งผ่านไปจนถึงตัวลูกที่เป็นตัวที่ถูกระบุ (ตัวเป้าหมาย) ให้อ่านข้อมูลและกำหนดการทำงานของ รีเลย์ และส่งกลับผลเหล่านั้น ต่อไป

ฟอร์แมทคำสั่ง (Command Format) จะประกอบไปด้วย ไบท์ที่เรียงกันไปเป็นชั้นๆซึ่งแต่ละชั้นก็จะมีความหมายของตัวเองและจะมาน้อยกี่ชั้นเท่าใดนั้นก็แล้วแต่ความห่างไกลระหว่างตัวแม่ ถึงตัวลูกตัวเป้าหมาย ฟอร์แมทคำสั่งสามารถแสดงได้ ดังรูปที่ 21 ดังนี้



รูปที่ 21 แสดงรูปแบบของ Command format

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Start Code ไบท์นี้เป็นไบท์ที่กำหนดระหัสสำหรับการตรวจสอบว่าเป็นจุดเริ่มต้นของคำสั่งหรือข้อมูล เพื่อป้องกันสัญญาณอื่นใดที่เข้ามารวบกวนได้ ในงานวิจัยกำหนดเป็นค่า 0x0F

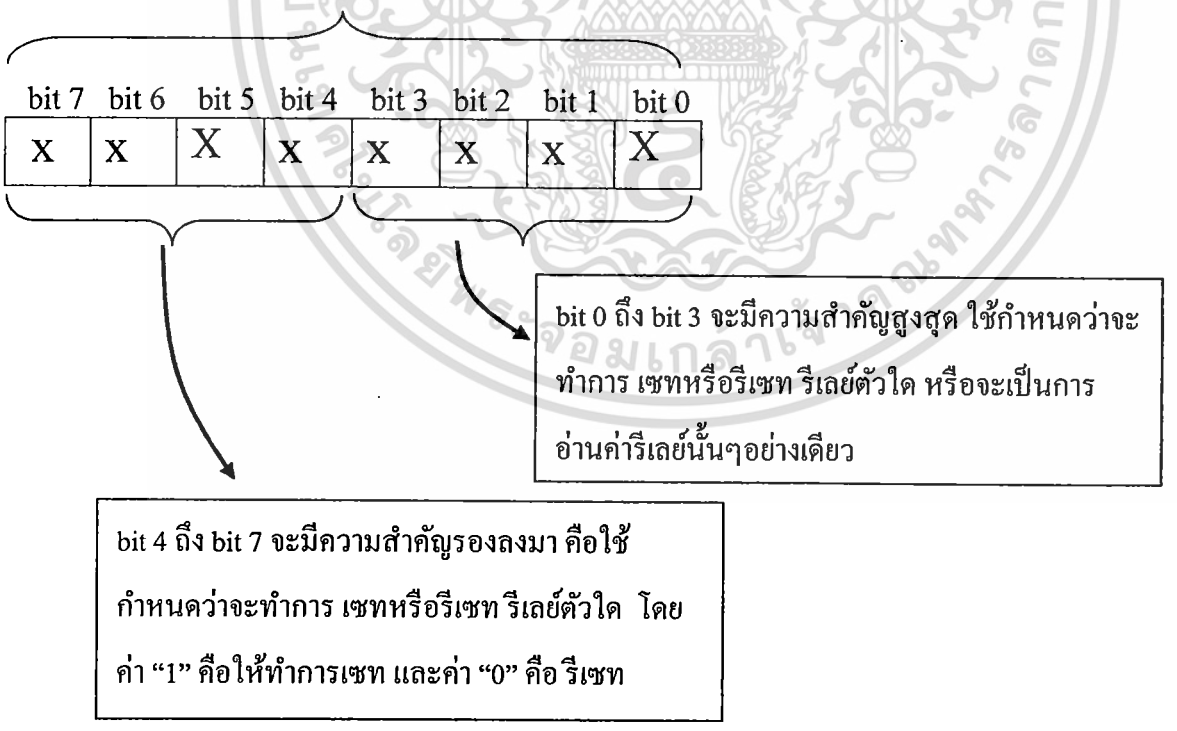
First Client ไบท์นี้จะใช้เก็บค่าที่บอกถึงหมายเลขของตัวลูกที่จะรับคำสั่งเพื่อทำการส่งต่อคำสั่งต่อไป

Seconds Client ไบท์นี้จะเก็บค่าที่บอกถึงหมายเลขของลูกตัวต่อมาจาก First Client ที่จะส่งต่อคำสั่งต่อไป

Certain Client ไบท์นี้จะเก็บค่าที่บอกถึงหมายเลขของตัวสุดท้ายหรือตัวเป้าหมายที่จะอ่านค่าข้อมูลเพื่อส่งกลับต่อไป

อนึ่ง จำนวนของตัวลูกตั้งแต่ First Client ไปจนถึง Certain Client จะมีค่ามากเท่าใด ก็แล้วแต่ว่าเส้นทางจากตัวแม่ไปถึงตัวลูก (ตัวเป้าหมาย) ว่ามีจำนวนมากเท่าใด

Read/Write Relay bit ในไบท์นี้แต่ละบิทจะบอกถึงให้ตัวลูกตัวเป้าหมายนั้นอ่านข้อมูลในลักษณะต่างๆ เช่น นอกจากจะเป็นคำสั่งอ่านอุณหภูมิแล้ว ยังมีการกำหนดสั่งให้ รีเลย์ทำงาน ON/OFF อีกสองตัว หรือจะเป็นการอ่านอย่างเดียวก็ได้ โดยการกำหนดให้มีการ ON/OFF ตามความหมายของระหัสดังแสดงในรูปที่ 22



รูปที่ 22 แสดงระหัสและความหมายของบิทที่ใช้ควบคุมการ ON/OFF ของรีเลย์

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

การใช้งานบิทต่าง

สี่บิทต่ำ คือ bit 0 ถึง bit 3 จะมีความสำคัญสูงสุด ซึ่งจะถูกตรวจสอบก่อนว่า แต่ละบิทมีค่าเป็นอะไร โดยแต่ละบิทจะใช้งานคู่กันเป็นคู่ๆกับ สี่บิทสูง ระหว่าง bit 0,1,2,3 และ bit 4,5,6,7 โดย

รีเลย์ตัวที่ศูนย์ จะใช้ bit 0 คู่กับ bit 4
รีเลย์ตัวที่หนึ่ง จะใช้ bit 1 คู่กับ bit 5
รีเลย์ตัวที่สอง จะใช้ bit 2 คู่กับ bit 6
รีเลย์ตัวที่สาม จะใช้ bit 3 คู่กับ bit 7

โดย หากบิทใดในสี่บิทต่ำนี้เป็น "0" หมายถึงเป็นคำสั่งที่กำหนดให้รีเลย์ของแต่ละบิทนั้น มีสถานะเป็น ON หรือ OFF ตามค่าใน สี่บิทบนคือ bit 4, bit 5, bit 6, bit 7 เช่น หาก bit 0 มีค่าเป็น "0" และ bit 4 มีค่าเป็น "1" หมายถึงเป็นการสั่งให้รีเลย์ตัวแรก ON นั่นเอง

ตัวอย่างเช่น
0x30 หมายถึงมีการกำหนดให้รีเลย์ตัวที่ ศูนย์ = ON, หนึ่ง = OFF, สอง = OFF, สาม = OFF
0x33 หมายถึงมีการกำหนดให้อ่านค่ารีเลย์ตัวที่ ศูนย์, หนึ่ง และให้ สอง=OFF, สาม=OFF
0x3F หมายถึงมีการกำหนดให้อ่านค่ารีเลย์ตัวที่ ศูนย์, หนึ่ง, สองและสาม

End Client เป็นไบท์ที่ใส่ค่าระหัสเป็น 0xEE บ่งบอกถึงการจบ (ตัวสุดท้ายคือโมดูลตัวที่ต้องการอ่านแล้ว)

End เป็นไบท์ที่ใส่ค่า 0xFF บ่งบอกถึงการจบของระหัสคำสั่ง

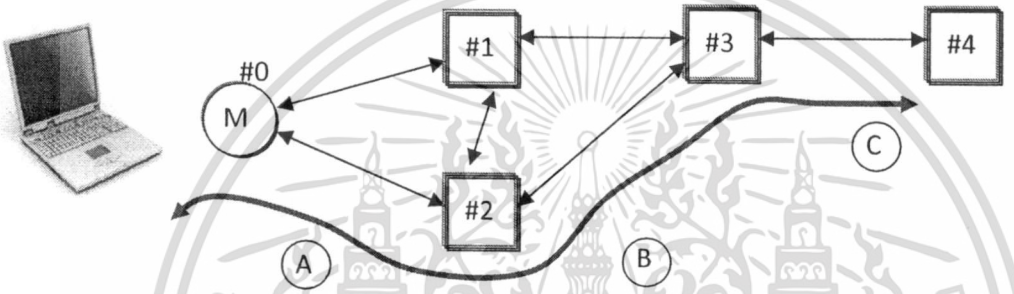
ในการส่ง Command Format นี้ไป ในแต่ละช่วงของตัวลูก ที่ต้องส่งคำสั่งต่อก็จะต้องมีการทำการปรับ Command Format นี้ไปทุกช่วงจนถึงตัวลูกเป้าหมาย ที่ต้องอ่านข้อมูลส่งกลับ ดังแสดงในตัวอย่างดังนี้

ในรูปที่ 23 แสดงเส้นทางตัวอย่างการวิ่งไปของ Command Format เพื่อไปอ่านข้อมูลของตัวลูกตัวที่สี่ กำหนดให้ตัวลูกหมายเลขสี่ มีการวัดอุณหภูมิเพียงอย่างเดียวและมี Relay ที่ใช้งานอยู่ชุดเดียวในช่องที่ศูนย์ เมื่อเป็นดังนี้เราก็จะได้ค่าคำสั่งในส่วนของ ไบท์ Read/Write bit คือ

Read/Write bit = 0x10 (หากกำหนดให้รีเลย์ตัวที่ศูนย์มีการเซทค่าเป็น ON ส่วนตัวอื่นเป็น OFF)
ส่วน ไบท์คำสั่งอื่นๆ ก็จะเป็นดังนี้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

Start Code = 0x0F
First Client = 0x00
Second Client = 0x02
Third Client = 0x03
Certain Client = 0x04
End Client = 0xEE
Read/Write bit = 0x10
End = 0xFF



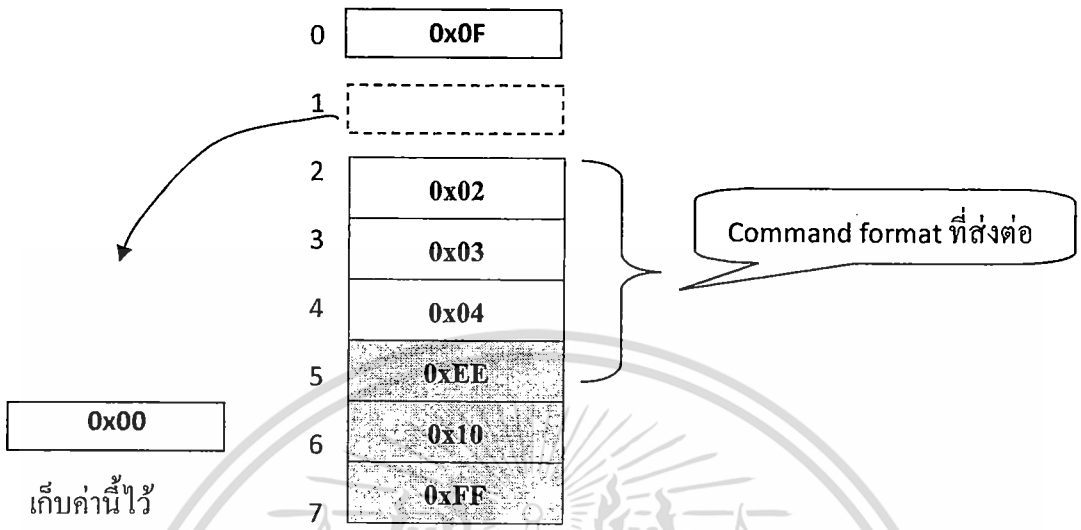
รูปที่ 23 แสดงเส้นทางการวิ่งไปอ่านข้อมูลของ Client ตัวหมายเลขสี่

ดังนั้นเราก็จะได้ Command Format ชุดแรกที่ถูกส่งออกไปจากตัวแม่ (แสดงในรูป 13 จุดที่ (A)) คือ

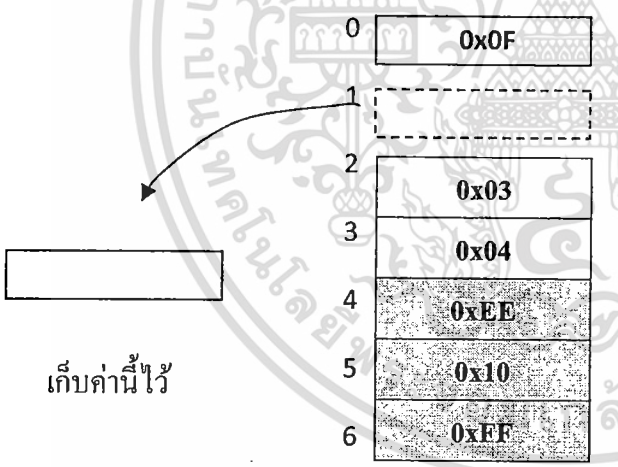
0	0x0F	เริ่มต้นระหัส
1	0x00	จากไหน
2	0x02	ตัวเอง
3	0x03	ไปไหนที่หนึ่ง
4	0x04	ไปไหนที่สอง
5	0xEE	หมดไม่ไปต่อ
6	0x10	สั่งรีเลย์
7	0xFF	จบ Command format

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

และ Command Format ก็จะถูกดัดแปลงเล็กน้อยแล้วส่งต่อไปสู่จุดต่อไปตามเส้นทางในรูป ซึ่งจะได้ Command Format ใหม่ที่จุด (B) คือ



จะเห็นได้ว่าเพียงแค่ส่วนบนหายไป หนึ่งไบท์ ดังนั้น ในจุด (C) ซึ่งเป็นจุดสุดท้ายก็จะได้เป็นดังนี้



ถึงตรงนี้ตัวลูกสุดท้ายนี้ก็จะทำการตรวจสอบคำสั่ง ว่าจะให้อ่านอุณหภูมิและสั่ง รีเลย์ตัวไดให้ทำงานอย่างไร ก็จะทำตามนั้นทันที

<u>หมายเหตุ</u> ใน Command format จะสังเกตุว่าไบท์ที่ระบุหมายเลขตัวที่สามารถรับคำสั่งได้ในแต่ละช่วงจะมีไบท์ที่อยู่ก่อนหน้าแสดงถึงว่ามาจากตัวไหนก่อนและไบท์ที่อยู่ถัดลงมาระบุว่าจะต้องส่งไปไหนต่อ แต่หากตรวจไบท์ที่สองนี้แล้วเจอระหัส 0xEE ก็แสดงว่าไม่มีการส่งต่อแล้วและตัวมันเองคือตัวเป้าหมายที่ต้องส่งค่าการวัดต่างๆ กลับในรูป Data format สู่ PC ต่อไป

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3.4.2 รูปแบบข้อมูลกลับ (Data Format)

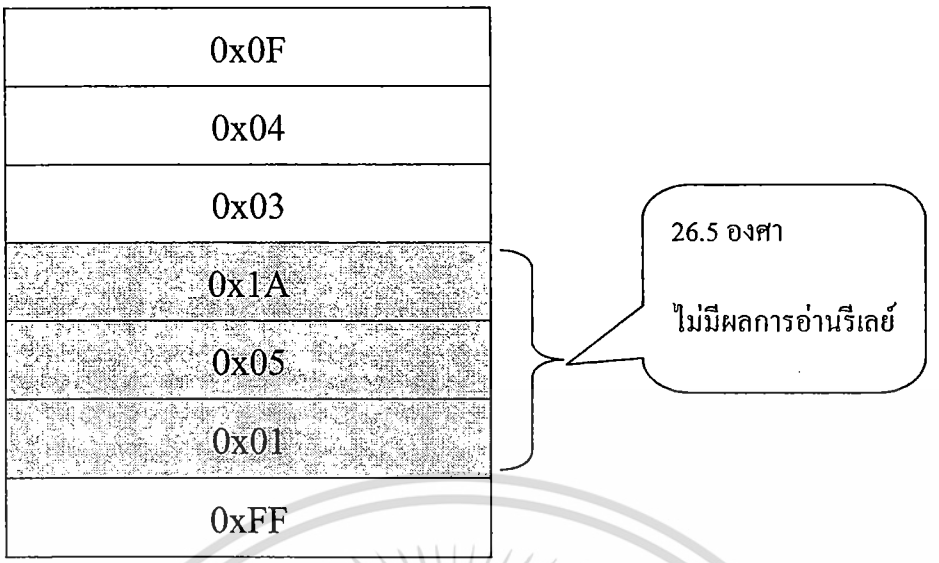
เมื่อตัวลูกเป้นหมายที่กำหนดได้ทำการอ่านค่าข้อมูลตามคำสั่งที่ต้องการแล้ว ต่อไปก็จะทำการส่งข้อมูลนั้นกลับสู่ PC อย่างถูกต้องตามเส้นทางที่กำหนด (เส้นทางเดิมตอนมา) ซึ่งจำเป็นต้องมีฟอร์แมทสำหรับการย้อนกลับได้ดังนี้ ขอเรียกว่า Data format เป็นฟอร์แมทที่ใช้ในการส่งข้อมูลกลับสู่ PC

ในรูปที่ 14 แสดงถึง Format ของ Data Format ที่จะต้องส่งกลับไปตามเส้นทางที่มา จะเห็นได้ว่านอกจากจะมีไบท์ที่เป็นระหัสเริ่มต้น (0x0F) ในไบท์แรกและไบท์สุดท้ายบอกการหมดข้อมูล (0xFF) แล้ว ก็จะมีไบท์ที่สองที่แสดงถึงหมายเลขผู้ส่ง ส่วนไบท์ที่สามคือ หมายเลขของลูก ที่จะส่งไปหา (ซึ่งไบท์นี้เราจะได้มาจากการเก็บค่าไว้ตอนได้รับ Command format) ตามด้วยข้อมูลของอุณหภูมินั้นซึ่งจะมี เลขหน้าจุดทศนิยม(8 bits) และเลขหลังจุดทศนิยม (8 bits) ตามด้วย ไบท์บอกถึงว่ามีการอ่านค่าหรือสั่งให้รีล์ช่องไหนทำงานบ้าง (ตัวอย่างนี้คือช่องที่ศูนย์) สุดท้ายก็ปิดด้วยคำสั่งจบ (END)

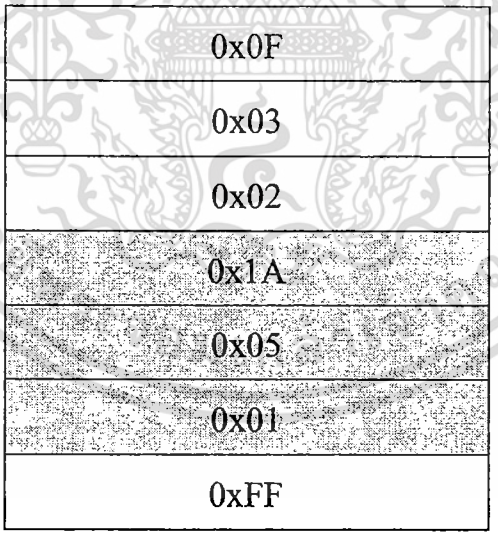
Start Code (ระหัสเริ่มต้น) **0x0F**
Own number (หมายเลขตัวเอง)
Send number (หมายเลขที่จะส่งไปหา)
อุณหภูมิเลขหน้าจุดทศนิยม (8 bits)
อุณหภูมิเลขหลังจุดทศนิยม (8 bits)
บอกสถานะของ รีเลย์
End **(0xFF)**

ดังนั้นจากตัวอย่างนี้ เราก็จะได้ Data Format ที่ส่งออกไปช่วงแรกคือ ที่จุด (C) คือ

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



เมื่อตัวลูกหมายเลข 3 ได้รับ Data Format แล้ว มันจะรู้ว่าจะส่งข้อมูลทั้งหมดไปต่อโดยจะมีการดัดแปลง Data Format เล็กน้อยดังนี้

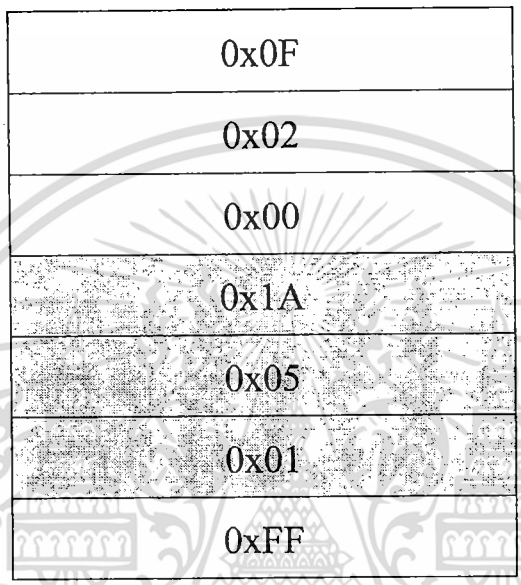


เป็น Data Format ที่จะถูกส่งออกไป ณ. จุด (B)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หมายเหตุ หมายเลขแรกที่เป็นหมายเลข 3 นั้น ได้มาจาก ในตอนที่มันรับ Command Format นั้น ก็จะเก็บหมายเลขนี้ไว้แล้ว เพื่อรอ Data Format กลับมาจะได้ใช้ในการส่งกลับได้อย่างถูกต้องนั้นเอง.

ดังนั้นในจุด (A) เราก็จะมี Data format เป็น



ถึงจุดนี้ เมื่อเครื่องแม่ได้รับ Data format นี้แล้วก็จะทำให้สามารถแยกแยะค่าข้อมูลที่ต้องการ เพื่อส่งให้ PC ทำการแสดงผล หรือเก็บบันทึกไว้ตามต้องการ

ดังนั้น การกำหนดค่าต่างๆ เช่น เส้นทางการเดินทางของ Command/Data Format ไปสู่ ตัวลูก แต่ละตัวนั้นก็จะมีความจำเป็นมากที่จะต้องกำหนดเส้นทางไว้ก่อนพร้อมทั้งกำหนดถึงการเซทหรือรีเซท รีเลย์ที่ใช้งานด้วย ซึ่งสามารถสร้าง Command format ของการส่งแต่ละครั้งได้จากการที่ผู้ใช้ คลิ๊กความต้องการต่างๆจากเมนู (โปรแกรม C# บน PC) จากนั้นเราก็นำมาจัดสร้างคำสั่งใหม่ให้สอดคล้องกับความต้องการเพื่อทำการส่งต่อไป อนึ่ง ในส่วนของซอฟร์แวร์ที่เขียน บนตัว dsPIC30F4011 ทั้งบนตัวเครื่องแม่ และตัวเครื่องลูก นั้นใช้ ภาษา C บน MPLAB [7] เหมือนกัน.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4) การทดสอบ (Experiment)

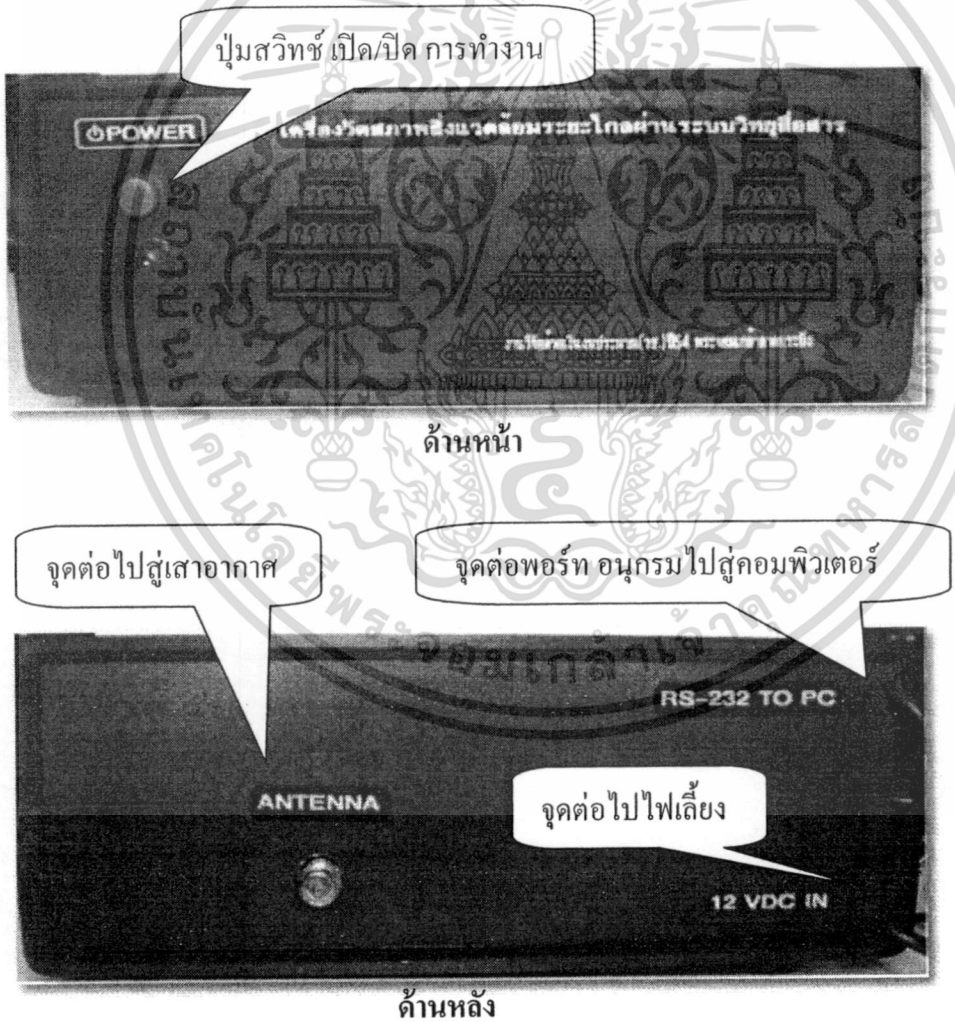
ในการทดสอบนั้น แบ่งเป็น ส่วนๆ ดังนี้

4.1 การติดตั้ง (Installation)

ลำดับการติดตั้งตัวเครื่องแม่และลูก เป็นดังขั้นตอน ต่อไปนี้

ติดตั้งตัวเครื่องแม่

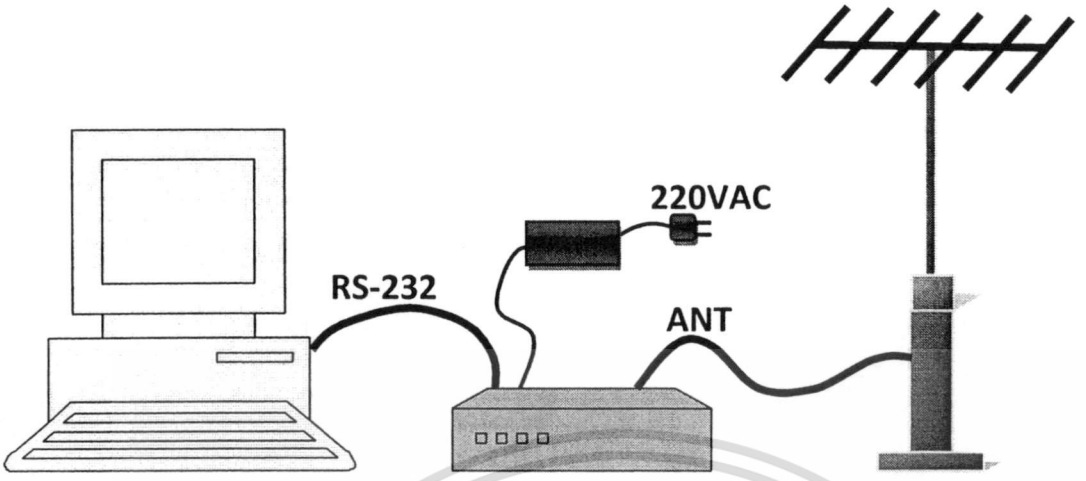
ตัวเครื่องได้ถูกออกแบบให้มีจุดที่เชื่อมต่อและการปรับแต่งให้น้อยที่สุด โดยที่ส่วนด้านหลังของตัวเครื่องก็จะมีช่องไฟฟ้าและสัญญาณ อยู่ เพียงสามจุด และด้านหน้าก็จะมีเพียง ปุ่ม สำหรับการเปิดและปิดเท่านั้นเอง ดังแสดงในรูป 24



รูปที่ 24 แสดงปุ่มสวิทช์สำหรับการเปิดใช้เครื่องด้านหน้าและจุดเชื่อมต่อ ด้านหลัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โดยไดอะแกรมการเชื่อมต่อก็จะ ง่ายๆ ดังรูปที่ 25 ข้างล่าง



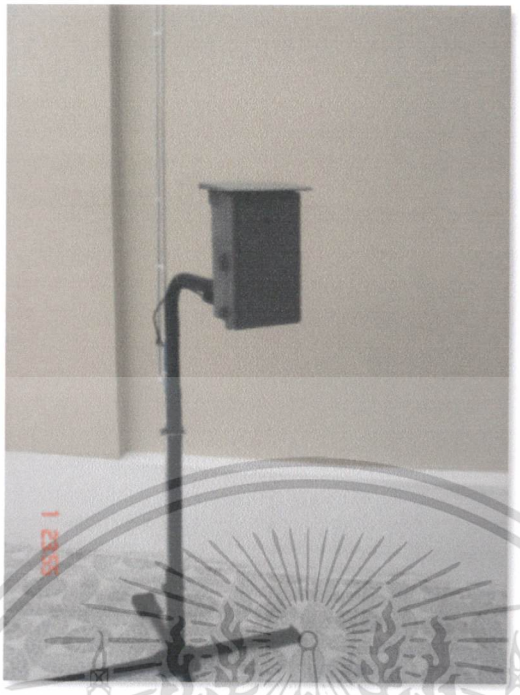
รูปที่ 25 ไดอะแกรมการเชื่อมต่อใช้งานตัวเครื่องตัวแม่

ติดตั้งเครื่องตัวลูก

ตัวลูกในงานวิจัยได้ ทำเป็นสองแบบ คือ แบบที่ติดตั้งในพื้นที่มีแรงไฟฟ้า ทำให้ไม่ต้องมีส่วนของพลังงานแสงอาทิตย์ ทำให้ขนาดที่เล็กลง สามารถนำไปติดตั้งบนอาคารสูง ได้ง่าย ในงานวิจัยได้ทำเป็นขาตั้งพร้อมติดเสาอากาศ ไว้ที่เดียวกันเลย แสดงดังรูปที่ 26 การใช้งานก็เพียง นำไปติดตั้งบนพื้นที่ๆต้องการได้เลย เพียงแต่ควรเป็นที่สูง เพราะเสาอากาศจะได้มีโอกาศรับเคลื่นสัญญาณได้ดี (เช่น บนตึกสูงต่างๆ) จากนั้นก็ต่อเชื่อมสายไฟเลี้ยงกับไฟ 220 VAC และเปิดสวิทช์ ก็เป็นการเริ่มใช้งานได้

ตัวลูกแบบที่สอง รูปทที่ 27 จะเป็นแบบที่มีระบบไฟเลี้ยงที่ได้จาก พลังงานแสงอาทิตย์ หรือระบบ Solar cell ทำให้ตัวเครื่องมีขนาดที่ใหญ่มากกว่าแบบแรก เพราะต้องมีพื้นที่เก็บ ส่วนของโมดูลประจุไฟ และส่วนของตัวแบตเตอร์รี่เก็บไว้ภายในกล่องเดียวกัน ตัวเครื่องติดตั้งบน ขาตั้งที่แข็งแรง เพราะต้องมีการติดตั้งทั้ง แผงรับแสงอาทิตย์ที่มีขนาดค่อนข้างใหญ่ และยังต้องมีการติดส่วนของ เสาอากาศด้วยบน ฐานตั้งตัวเดียวกัน ส่วนเรื่องของการใช้งานก็จะเป็นเหมือนกันกับ แบบแรกเพียงแต่เราไม่ต้องมีการการ ต่อไปเลี้ยง 220 VAC เข้า ที่ต้องทำเพียงนำไปตั้งในพื้นที่ๆต้องการ จากนั้นก็เปิดฝาเครื่องออก เพื่อ เปิดสวิทช์ไฟ ก็เป็นอันเสร็จเครื่องก็พร้อมที่จะทำงานได้แล้ว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 26 แสดงตัวเครื่องลูกแบบที่ใช้ในพื้นที่ที่มีแรงดันไฟฟ้าใช้งาน

รูปที่ 27 แสดงตัวเครื่องลูกแบบที่ใช้ในพื้นที่ที่ไม่มีแรงดันไฟฟ้าใช้งาน

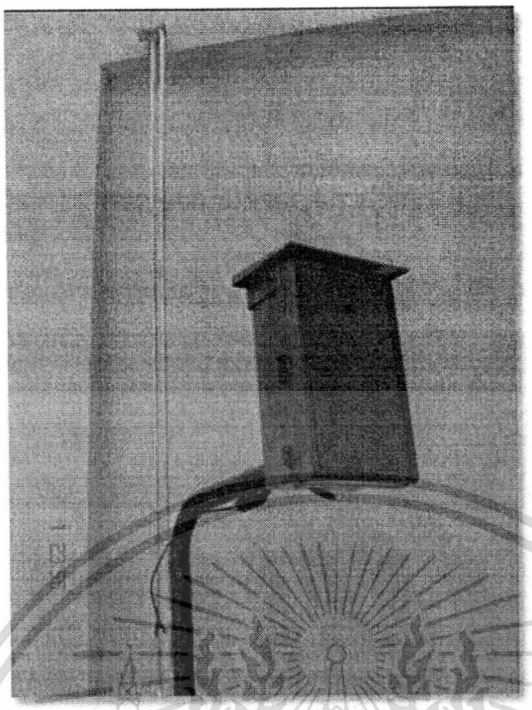
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

4.2 การทดสอบใช้งาน (Testing on field)

ในการทดสอบ ผู้วิจัยได้ทำการทดสอบหลายพื้นที่ โดย การทดลองหนึ่งในหลายพื้นที่นั้นได้ทำการติดตั้งตัวแม่ไว้ กับห้องปฏิบัติการของผู้วิจัยเองที่ อยู่ใน อาณาเขตคณะวิศวกรรมศาสตร์ พระจอมเกล้าลาดกระบัง บนตึกสูง 12 ชั้น ดังแสดงในรูปที่ 28 ส่วนตัวเครื่องตัวลูกชุดที่ไม่มีระบบไฟฟ้าแสงอาทิตย์ ก็ได้นำไปติดตั้งห่างออกไปจากคณะวิศวกรรมศาสตร์ ประมาณ 5 กิโลเมตร ในหมู่บ้านปริชาต สุวิทวงศ์ ดังแสดงในรูปที่ 29 ส่วนอีกตัวหนึ่งที่มีระบบไฟฟ้าตัวเองก็ติดตั้งห่างออกไปอีก ประมาณ 5 กิโลเมตร ช่วงไกล้ๆแถวเขตมีนบุรี และในรูปแบบเช่นเดียวกันก็ได้มีการทดลองในพื้นที่อื่นๆ อีก ที่นอกพื้นที่ของ พระจอมเกล้าลาดกระบัง เช่น ก็ได้มีการติดตั้งทดลองในพื้นที่ เขตปทุมธานี โดยวางตำแหน่งระยะห่างประมาณ 5 กิโลเมตร เช่นกัน โดยมีชาวบ้านในพื้นที่ วัดเทียนถวาย ดังรูปที่ 30 ได้กรุณาช่วยให้ความร่วมมือในการทดสอบครั้งนี้ด้วย เหตุที่ต้องติดตั้งเช่นนี้ ก็เพราะ ต้องการติดตั้งให้เป็นไปตามเงื่อนไขของงานวิจัย นั่นคือ ตัวแม่หากต้องการส่งการติดต่อไปตัวลูกตัวที่สองได้ต้องมีการติดต่อผ่าน ตัวลูกตัวที่หนึ่งที่อยู่ระหว่าง ตัวแม่กับตัวลูกตัวที่สองเท่านั้น เพราะตัวแม่ไม่สามารถส่งสัญญาณไปถึงตัวลูกตัวที่สองได้นั่นเอง (ในการทดสอบการติดต่อระหว่างตัวแม่กับตัวที่สอง โดยปิดตัวที่หนึ่งไว้ ก็จะไม่สามารถติดต่อได้)

รูปที่ 28 ตัวเครื่องตัวแม่ติดตั้งที่ตึก 12 ชั้น คณะวิศวกรรมศาสตร์ ลาดกระบัง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 29 ตัวเครื่องตัวลูกตัวที่หนึ่งติดตั้งในหมู่บ้านปริชาติ ห่างจากตัวแม่ประมาณ 5 กิโลเมตร

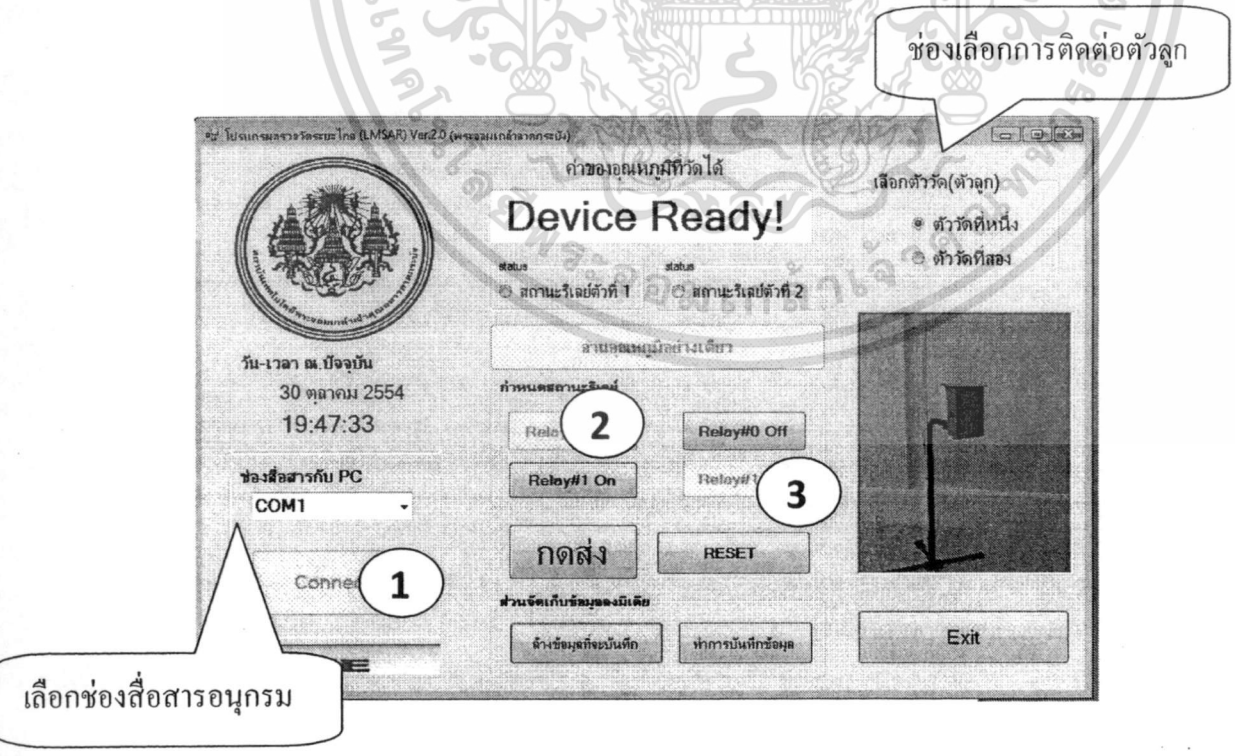
รูปที่ 30 ตัวเครื่องตัวลูกตัวที่สองติดตั้งทดลองในเขตวัดเทียนทถวย เขตปทุมธานี (กับเจ้าของสถานที่)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เมื่อได้ทำการติดตั้งตัวลูกในพื้นที่ที่ต้องการแล้ว เริ่มปิดโปรแกรมใช้งานบนคอมพิวเตอร์ผู้ใช้ก็จะได้ผลแสดงบนจอดังรูปที่ 31

รูปที่ 31 รูปบนจอคอมพิวเตอร์ของผู้ใช้เมื่อเริ่มใช้โปรแกรม

จากนั้นก็เปิดเครื่องตัวแม่ โดยกดสวิทช์ตัวเดียวเท่านั้น และเมื่อผู้ใช้ต้องการอ่านข้อมูลอุณหภูมิจากตัวลูกตัวที่หนึ่ง (ตัวอยู่ใกล้ที่ติดตั้งอยู่ใกล้เขตมีนบุรี) โดยต้องการอ่านอุณหภูมิ และต้องการกำหนดให้รีเลย์ตัวที่ศูนย์ทำงาน (ON) และตัวที่หนึ่ง ปิด (OFF) ก็ทำได้ง่ายๆ ตามรูปที่ 32



รูปที่ 32 แสดงการกดปุ่มบน GUI เพื่ออ่านค่าอุณหภูมิจากตัวลูกตัวที่หนึ่ง (ตัวที่อยู่ใกล้เขตมีนบุรี)

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

เริ่มจาก ทำการเลือกช่องการสื่อสารอนุกรมของตัวคอมพิวเตอร์กับตัวเครื่องแม่ที่ผู้ใช้ เชื่อมต่ออยู่ให้ถูกต้อง

กดปุ่ม (1) เพื่อทำการเชื่อมต่อช่องสื่อสารนั้นกับตัวเครื่องตัวแม่แบบอนุกรม

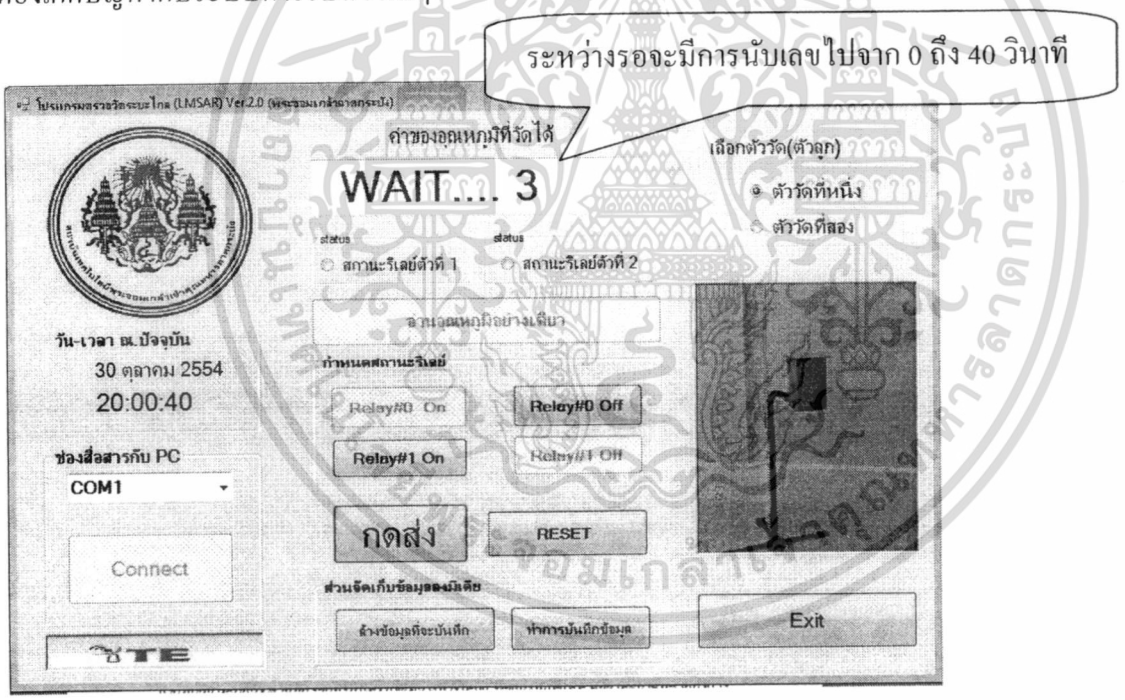
กดปุ่ม (2) เลือกให้ รีเลย์ตัวที่ศูนย์เป็นสถานะ ON

กดปุ่ม (3) เลือกให้รีเลย์ตัวที่หนึ่งให้เป็นสถานะ OFF

ตรวจดูว่าเราต้องการติดต่อกับตัวลูกตัวใดในช่องเลือกตัวลูกดังรูปเป็นเลือกติดต่อกับตัวลูกตัวที่หนึ่ง

สุดท้าย ก็กด ปุ่ม **"กดส่ง"** เพื่อเริ่มต้นการส่งสัญญาณคำสั่งออกไปสู่ตัวแม่

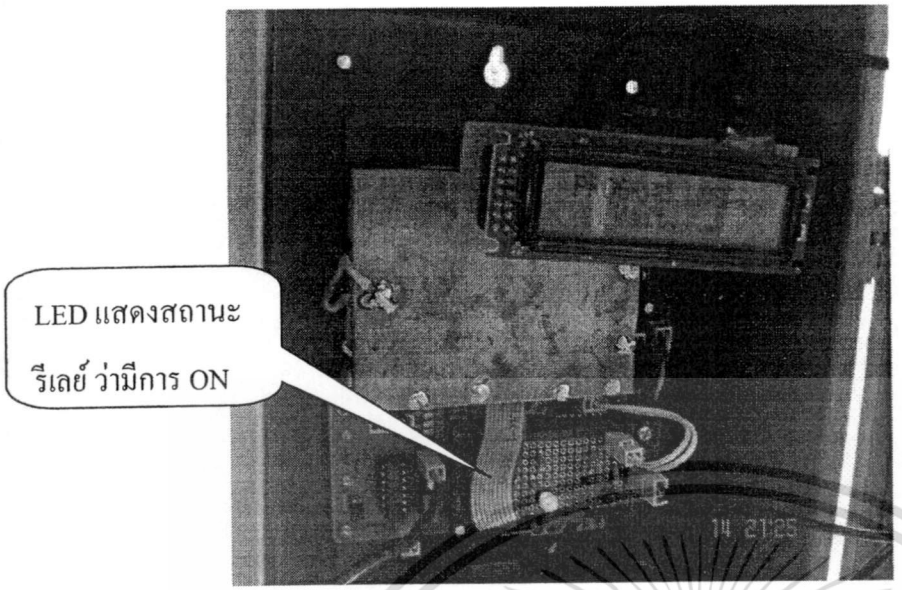
จากนั้น ก็เพียงรอข้อมูลที่กลับคืนมา โดยในขณะที่รอตัวหน้าจอของ GUI ก็จะมีการ นับตัวเลขการรอไปเรื่อยๆ จาก 0 ถึง 40 ซึ่งระยะเวลาการรอจะไม่เกิน 40 แน่นอน (จากการทดลอง) เพราะหากเกินนี้แสดงว่า จะต้องเกิดปัญหากับระบบการรับส่ง แน่ๆ



รูปที่ 33 แสดงจอ GUI ที่อยู่ระหว่างการรอข้อมูลกลับคืนมา

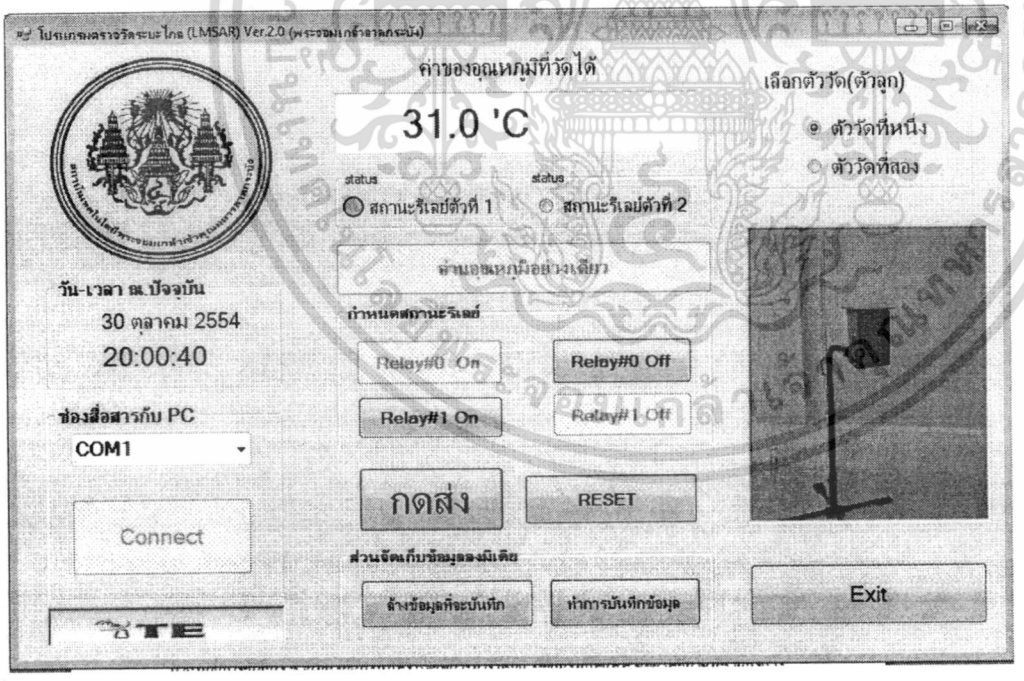
หากทุกอย่างถูกต้อง เราดูที่ตัวลูกตัวที่หนึ่ง ก็จะมีการทำงานคือ รีเลย์ตัวที่ศูนย์นั่น มีสถานะตามที่เราต้องการ คือ ON แสดงดังรูปที่ 34

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 34 แสดงผลของตัวลูกตัวที่หนึ่งที่รีเลย์มีสถานะ ON ตามคำสั่ง

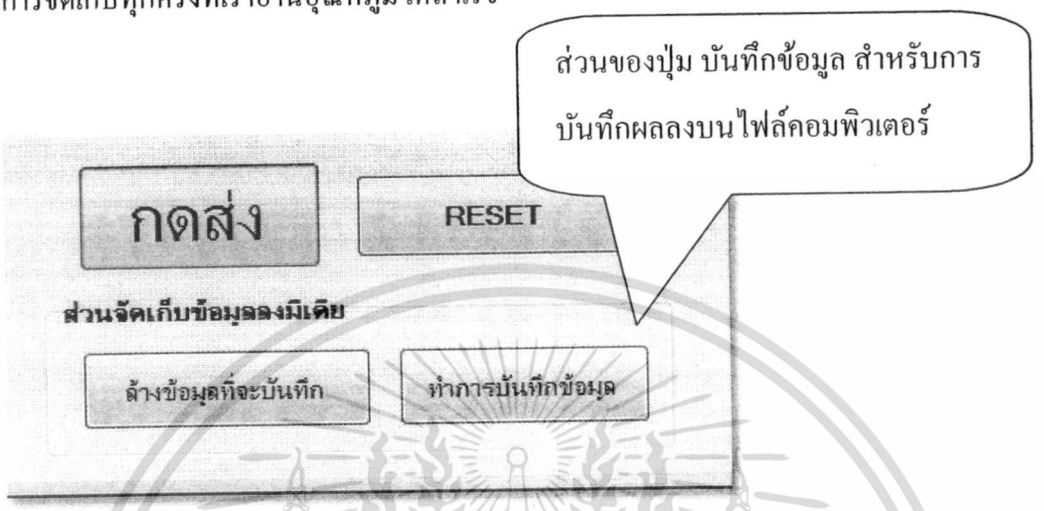
จากนั้น ไม่นานก็จะมีข้อมูลของอุณหภูมิกลับมาแสดงผลที่จอของผู้ใช้เอง ดังรูปที่ 35



รูปที่ 35 ผลของข้อมูลที่กลับมาแสดงบนจอคอมพิวเตอร์ของผู้ใช้

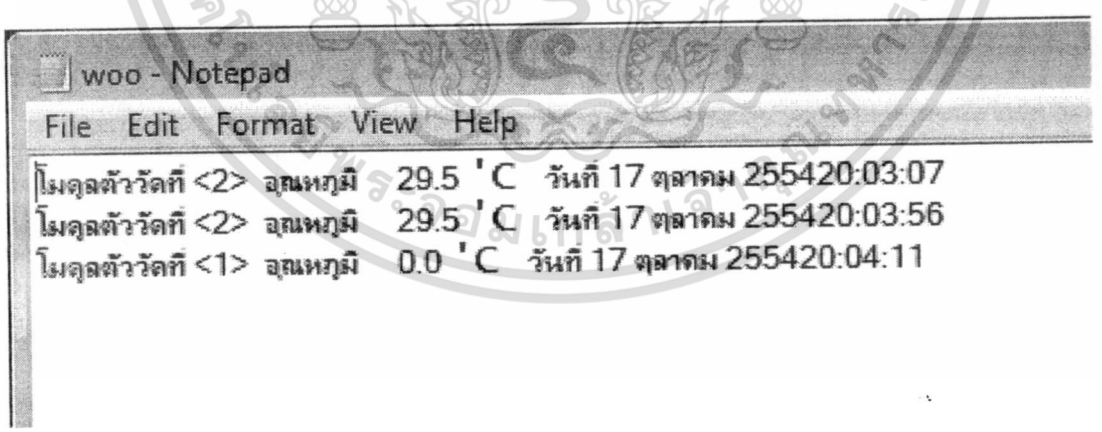
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

อนึ่งนอกจากอุณหภูมิจะได้แสดงบนจอผู้ใช้แล้วข้อมูลอย่างอื่นก็จะมีมาด้วยคือ สถานะของรีเลย์ที่ตัวลูกตัวนี้ว่าตัวใด ON หรื OFF อยู่ ก็จะแสดงบนจอ GUI ด้วยเช่นกัน นอกจากนี้เรายังสามารถจัดเก็บผลของการอ่าน แต่ละครั้งไว้บนไฟล์คอมพิวเตอร์ได้อีก โดย กดไปที่ปุ่ม "ทำการบันทึกข้อมูล" ข้างล่างดังรูปที่ 36 โดยจะมีการจัดเก็บทุกครั้งที่เราอ่านอุณหภูมิได้สำเร็จ



รูปที่ 36 ปุ่มสำหรับการจัดเก็บข้อมูลลงบนคอมพิวเตอร์

และการจัดเก็บก็จะมีรูปแบบเป็น เท็กไฟล์ ดังแสดงตัวอย่างผลที่ได้ดังรูปที่ 37

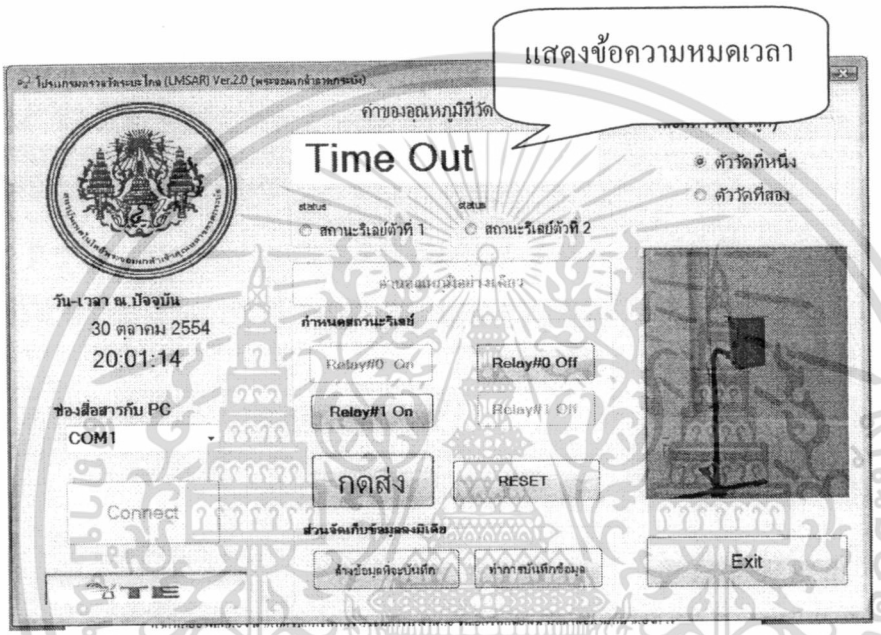


รูปที่ 37 แสดงตัวอย่างไฟล์ข้อมูลที่จัดเก็บไว้

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จะเห็นได้ว่ามีการจัดเก็บทั้ง ข้อมูลอุณหภูมิ, วัน และเวลา พร้อมทั้งบอกว่าเป็นของตัวลูกตัวไหนไว้อย่างเป็นระเบียบ ทำให้ง่ายต่อการนำไปวิเคราะห์ด้วยโปรแกรม ประมวลผลอื่นๆ เช่น Excell เป็นต้น

อนึ่ง หากรอแล้วเกิน 40 วินาที (ดูที่หน้าจอ) ที่หน้าจอก็จะแสดงคำว่าหมดเวลา ดังรูปที่ 38 เราก็จะต้องทำการส่งใหม่ได้อีก โดยการกด **" กดส่ง "** ซ้ำอีกที ก็จะเป็นการเริ่มขบวนการอ่านอุณหภูมิอีกครั้ง



รูปที่ 38 แสดงกรณีที่เกิดข้อผิดพลาดของการรับข้อมูลจากตัวลูก

จากหน้าจอ GUI เราจะเห็นได้ว่า สามารถแก้ไข คำสั่งได้ง่ายมากๆ เพียงแต่กดปุ่มตามที่มีให้บนจอ เช่น การกำหนดรีเลย์ให้ ON/OFF , การกำหนดให้เป็นคำสั่งเพียงอ่านค่าอุณหภูมิและสถานะของรีเลย์ของตัวลูกตัวเป้าหมายเท่านั้น ก็จะมีปุ่ม "**อ่านอุณหภูมิอย่างเดียว**" ให้กดใช้งานได้เลย ส่วนปุ่ม อื่นๆ ก็สามารถทำความเข้าใจได้โดย ง่าย เช่นกัน

อนึ่ง ในการทดลองนั้น ผู้วิจัยได้ทำการทดสอบทั้ง สองตัวลูก คือทั้งการติดต่อกับเครื่องตัวลูกตัวที่หนึ่ง และติดต่ออ่านอุณหภูมิของตัวลูกตัวที่สอง โดยวิธีการใช้งานก็จะเป็นเหมือนกันกับการติดต่อกับตัวเครื่องตัวที่หนึ่ง หากแต่ต้องมีการเลือก กำหนดตัวที่ต้องการติดต่อบน เมนู GUI ให้ถูกต้องก่อนการส่งคำสั่งออกไป เท่านั้นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5) ผลการทดสอบ (Result of experiments)

ในการทดลอง ผู้วิจัยได้ทำการทดลอง บนพื้นที่หลายๆ แห่ง ทำให้ได้ผลสรุปในภาพรวมได้ว่า สามารถส่งและรับข้อมูล ได้อย่างถูกต้อง ตามต้องการ ผู้วิจัยกำหนดความห่างของแต่ละตัวลูกแต่ละตัวโดยดูจากระยะที่สามารถติดต่อกับโดยการสื่อสารทดสอบด้วยเสียงให้ชัดเจนเท่านั้น เพื่อความมั่นใจว่าสัญญาณไปถึงกันได้ เพราะพื้นที่บางแห่งไม่สามารถติดตั้งห่างกันมากได้ เพราะมีสิ่งกีดขวาง และความสูงของเสาอากาศที่ไม่สามารถติดตั้งได้สูงได้ในบางพื้นที่ รวมทั้งชนิดของเสาอากาศ ก็เป็นปัจจัยด้วยเช่นกัน ในการทดสอบผู้วิจัยได้ใช้เสา อากาศแบบ สลิมจิม และแบบ วี2 ซึ่งให้ผลไม่ต่างกันนัก ระยะที่สามารถติดต่อกันได้ก็จะประมาณ 5 กิโลเมตร ทั้งนี้ดังที่ทราบปัจจัยสำคัญที่ทำให้ระยะการติดต่อเพิ่มขึ้นได้ก็อยู่ที่ ความแรงของสัญญาณส่งและระดับความสูงของเสาอากาศ เพราะจะทำให้ได้ระบบเสียงที่ชัดเจน สัญญาณรบกวนจะเบาลง และแต่ปัญหาที่พบอีกก็คือ

กรณี ส่งและรับข้อมูลกลับมาโดยปกติ หากไม่มีความผิดพลาดใดๆ นับตั้งแต่เริ่มส่งสัญญาณจนได้รับข้อมูลที่ถูกต้องกลับมาแสดงบนเครื่องของผู้ใช้

ตัวที่หนึ่ง จะใช้เวลา 15 วินาที ตัวที่สอง จะใช้เวลา 30 วินาที

กรณี มีข้อผิดพลาด จะต้องมีการ กดส่งซ้ำ ประมาณ 2 ถึง 5 ครั้งจึงสามารถจะรับข้อมูลที่ถูกต้องได้ ทั้งนี้ สาเหตุ น่าจะมาจาก การรบกวนของคลื่นวิทยุในย่านความถี่เดียวกัน เพราะ ความถี่ที่ใช้ในงานวิจัยนี้ เป็นความถี่ขนาด 144 MHz ซึ่งเป็นของ ย่าน วิทยุสมัครเล่นแห่งประเทศไทย ฉะนั้นก็เป็นไปได้ที่อาจมีการใช้ความถี่ที่ตรงกัน ดังนั้น สำหรับตัวที่อยู่ห่างไกลออกไปก็ อาจมีโอกาศสูงที่จะไม่สามารถติดต่อได้ อันเนื่องจากสัญญาณวิทยุจะอ่อนลง นั่นเอง

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

6) อภิปราย/วิจารณ์ (Discussion)

จากผลการวิจัย ที่ผ่านมาในหัวข้อที่ 5 นั้น จะเห็นได้ว่า ส่วนของตัวเครื่องนั้นมีปุ่มให้ปรับน้อยมาก คือในส่วนของตัวแม่จะมีเพียง ปุ่มเดียวที่ใช้ในการเปิดปิดเครื่อง ส่วนด้านหลังก็จะมีจุดต่อที่เข้าใจง่าย คือต่อไฟเลี้ยง, ช่องสื่อสารอนุกรมกับคอมพิวเตอร์ และจุดต่อกับเสาอากาศเท่านั้นเอง ทำให้ง่ายในการติดตั้งตัวแม่มาก ส่วนตัวลูกก็จะมีเพียงปุ่มเดียวสำหรับการเปิดปิดเช่นเดียวกัน นอกนั้นก็ไม่ต้องการปรับแต่งอะไรเลย (นอกจากกรณีต้องการเปลี่ยนแปลงความถี่) สรุปได้ว่าในส่วนของ ฮาร์ดแวร์ในการติดตั้งก็แทบไม่ต้องทำอะไรมาก เพียงแต่นำตัวลูกไปตั้งไว้ในตำแหน่งพื้นที่ๆต้องการวัด แล้วเปิดเครื่องไว้เลย ส่วนตัวแม่ก็ทำการต่อไฟเลี้ยง, เชื่อมต่อช่องสื่อสารอนุกรม (RS-232) กับคอมพิวเตอร์ และต่อเสาอากาศเข้าสู่ตัวเครื่องแม่ ก็สามารถใช้งานได้แล้ว

จากการทดลองนำไปใช้งาน ได้มีการทดลองใช้งานในหลายพื้นที่ โดยมีทั้งที่มีสิ่งกีดขวางมากและสถานที่โล่งเพื่อให้มีสภาพที่ใช้งานเหมือนจริงมากที่สุด และได้ทดลองเป็นเวลานาน หลายวันเพื่อดูความคงทน (Stable) หรือความผิดเพี้ยนใดๆ อันอาจเกิดขึ้นได้ ก็ได้ข้อสรุปที่น่าพอใจว่าสามารถทำงานได้ดี คือสามารถใช้งานได้ดีและสะดวกมาก ในระดับหนึ่ง หากแต่ถ้าสามารถปรับปรุงให้ตัวลูกมีขนาดของเครื่องที่เล็กลงไปอีกก็จะเป็นประโยชน์ได้ไม่น้อย เพราะว่าในการนำไปใช้งานจริงนั้น ระบบจ่ายไฟสำคัญมาก เพราะหากนำไปติดตั้งในพื้นที่ๆห่างไกล เช่น บนภูเขา ซึ่งที่นั้นไม่มีทั้งสัญญาณใดๆ และไฟฟ้าก็ไม่มีด้วย ทำให้ต้องมีระบบจ่ายไฟฟ้าที่มาจากพลังงานแสงอาทิตย์ใช้งานกับเครื่องลูก ซึ่งจากการทดลองที่ได้มา ด้วยระบบไฟฟ้าจากโซล่าเซลที่ใช้งานเป็นเพียงแผงเดียวคือขนาด 20 W เท่านั้น แต่ก็สามารถใช้งานต่อเนื่องได้ประมาณ 1 วัน ครึ่ง (กรณีมีการรับส่งทุกๆ ครึ่งชั่วโมง) ซึ่งพิจารณาดูแล้วอาจไม่เพียงพอต่อการใช้งานจริง เพราะอาจเป็นไปได้ที่จะไม่มีแสงอาทิตย์เลยเป็นจำนวน มากกว่า 2-3 วัน ดังนั้น ในส่วนนี้ก็แก้ปัญหาแบบตรงไปตรงมา คือการเพิ่มความจุของแบตเตอร์รี่เข้าไปอีก สัมพันธ์กับการเพิ่มขนาดของกำลังงาน และจำนวนของแผงโซล่าเซลมากขึ้นไปอีกได้ เพื่อการประจุเก็บไฟฟ้าไว้จะได้รวดเร็วและพอเพียงต่อการใช้งานที่มากขึ้นไปอีก .

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

7) สรุปและข้อเสนอแนะเกี่ยวกับการวิจัยในขั้นต่อไป (Conclusion)

จากผลของการทดลอง เราก็พอจะสรุปได้ว่า สามารถทำงานได้ในระดับที่น่าพอใจระดับหนึ่ง ส่วนปัญหาหลักๆเลยพอสรุปได้เป็นข้อๆ และแนวทางการวิจัยต่อไปดังนี้คือ

1) เรื่องของการผิดพลาดของข้อมูลขณะที่มีการรับส่ง ซึ่งจะมีความเป็นไปได้ถึงประมาณ 20-40% ที่สัญญาณอาจถูกรบกวนด้วยการใช้คลื่นวิทยุที่ความถี่เดียวกัน เหตุก็ด้วยจากการที่ผู้วิจัยได้ใช้ระบบวิทยุสื่อสารที่เป็นแบบ ประชาชนทั่วไปก็สามารถใช้งานได้ เช่น ที่ความถี่ย่าน 144 MHz ซึ่งเป็นของ ชมรมวิทยุสมัครเล่นแห่งประเทศไทย หรือจะเป็นคลื่อน CB ของประชาชนทั่วไป ก็สามารถใช้งานได้ ทำให้ไม่ว่าจะไปใช้ในพื้นที่ใดก็ตามในประเทศ ระดับการใช้งานที่ความถี่ตรงกันกับความถี่ของผู้วิจัยก็มีโอกาศเป็นไปได้ประมาณ 20 – 40% ทีเดียว อันนี้ไม่สามารถหลีกเลี่ยงไปได้

เหตุผลดังกล่าวนี้ ผู้วิจัยจึงวิเคราะห์หาทางแก้ไขหรือปรับปรุงให้ดีขึ้น ที่เป็นไปได้อยู่ สอง ทางคือ

ทางที่หนึ่งคือ ไปเสนอขอใช้ความถี่เฉพาะไปเลยกับทางราชการ ซึ่งเป็นไปได้ยากมากๆ

ทางที่สองคือ ปรับปรุงส่วนของซอฟร์แวร์บนตัวเครื่องลูกให้มีความสามารถที่ฉลาดมากขึ้นในการตรวจสอบสัญญาณของการรับส่งแต่ละครั้งว่า ได้รับข้อมูลหรือคำสั่งสมบูรณ์หรือไม่ ซึ่งอันนี้เป็นไปได้ คือต้องมีการส่งไปมาระหว่างตัวรับและตัวส่งเป็นจำนวนมากครั้งขึ้น ซึ่งจะทำให้เสียเวลาและสิ้นเปลืองพลังงาน แต่ก็น่าทำเพราะคุ้มค่า หากสามารถทำให้ข้อมูลที่ได้รับถูกต้อง

อนึ่งในแนวทางแก้ไขนี้ ผู้วิจัยได้นำเสนอเป็นโครงการวิจัยที่ต่อเนื่องไปกับ วช. แล้วในปี 56 โดยได้นำเสนอการเพิ่มประสิทธิภาพของการรับส่งที่ ให้ตัวลูกมีความเป็นอัจฉริยะมากขึ้นคือ สามารถตรวจสอบข้อมูลแล้วยังหาเส้นทางไปสู่ตัวเป้าหมายได้ด้วยตัวเอง กรณีที่หากมีตัวใดตัวหนึ่งระหว่างเส้นทางไปสู่ตัวเป้าหมายใช้งานไม่ได้ (เสีย) อันนี้ข้อยุ่งยากก็จะอยู่ที่ตัวซอฟร์แวร์ ปัญหาที่เกิดจากการรบกวนก็คาดว่าจะน้อยลง เพียงแต่ต้องศูนย์เสียเวลาของการเชื่อมต่อสื่อสารกัน มากขึ้น

2) ปัญหาการเพิ่มจำนวนของตัวลูก ซึ่งปัญหานี้เกิดขึ้นได้เพราะในส่วนของจำวนวนความยาวของรูปแบบคำสั่งนั้น จะเห็นได้ว่ายิ่งตัวลูกตัวเป้าหมาย อยู่ห่างจากตัวแม่เท่าใดก็ต้องใช้ตัวลูกตัวอื่นในการส่งผ่านสัญญาณไป อันนี้เมื่อคำสั่งมากขึ้นแวลาในการรับส่งแต่ละครั้งก็มากขึ้นด้วย ซึ่งอันตรายต่อการถูกรบกวนของคลื่นความถี่เดียวกันจากที่อื่น หรือ หากมีตัวใดตัวหนึ่งเสียหายไม่ทำงาน ก็เป็นไปไม่ได้ที่จะส่งสัญญาณไปสู่ตัวเป้าหมายได้เช่นกัน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ด้วยเหตุผลนี้ ผู้วิจัยได้เสนอทางแก้ คือ สร้างซอฟร์แวร์ใหม่ให้ตัวลูกทุกตัวสามารถตรวจสอบความพร้อมของตัวลูกตัวอื่นรอบข้างที่ตนจะใช้ในการส่งข้อมูลหรือคำสั่งต่อไป เพื่อให้มั่นใจว่าสามารถส่งคำสั่งต่อไปได้ และสามารถให้ตัวลูกจัดลำดับความสำคัญของการเลือกตัวลูกเพื่อใช้ในการสื่อสารได้เอง ซึ่งก็จะแก้ปัญหาของการส่งข้อมูลหรือคำสั่งให้ไปสู่ตัวเป้าหมายได้ ซึ่งความสามารถอันนี้ก็รวมอยู่ในข้อเสนองานวิจัยต่อไปในปี 56 เช่นกัน

3) ปัญหาเรื่องแหล่งจ่ายไฟ ปัญหาอันนี้ก็คงตรงไปตรงมาคือ วิเคราะห์จากผลการทำงานแล้วจากเดิม แบตเตอร์รี่ขนาด 7 Ah สามารถให้เครื่องทำงานได้นาน หนึ่งวันกว่า หากต้องการสามวันก็อาจใช้ระบบไฟฟ้า (โซล่าเซล) ที่ขนาดกำลังที่มากเป็นสองเท่า ก็น่าจะเพียงพอ.

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บรรณานุกรม (Bibliography)

[1] http://www.songjiangmall.com/packet_a.htm

[2] http://www.thaiamp.com/

[3] http://www.ett.co.th/

[4] http://www.datasheetarchive.com/**TCM3101**-datasheet.html

[5] สัจจะ จรัสรุ่งรวีวร , " คู่มือ Visual C# 2005 ฉบับสมบูรณ์ ", บริษัท ไอดีซีฯ, พ.ศ. 2550

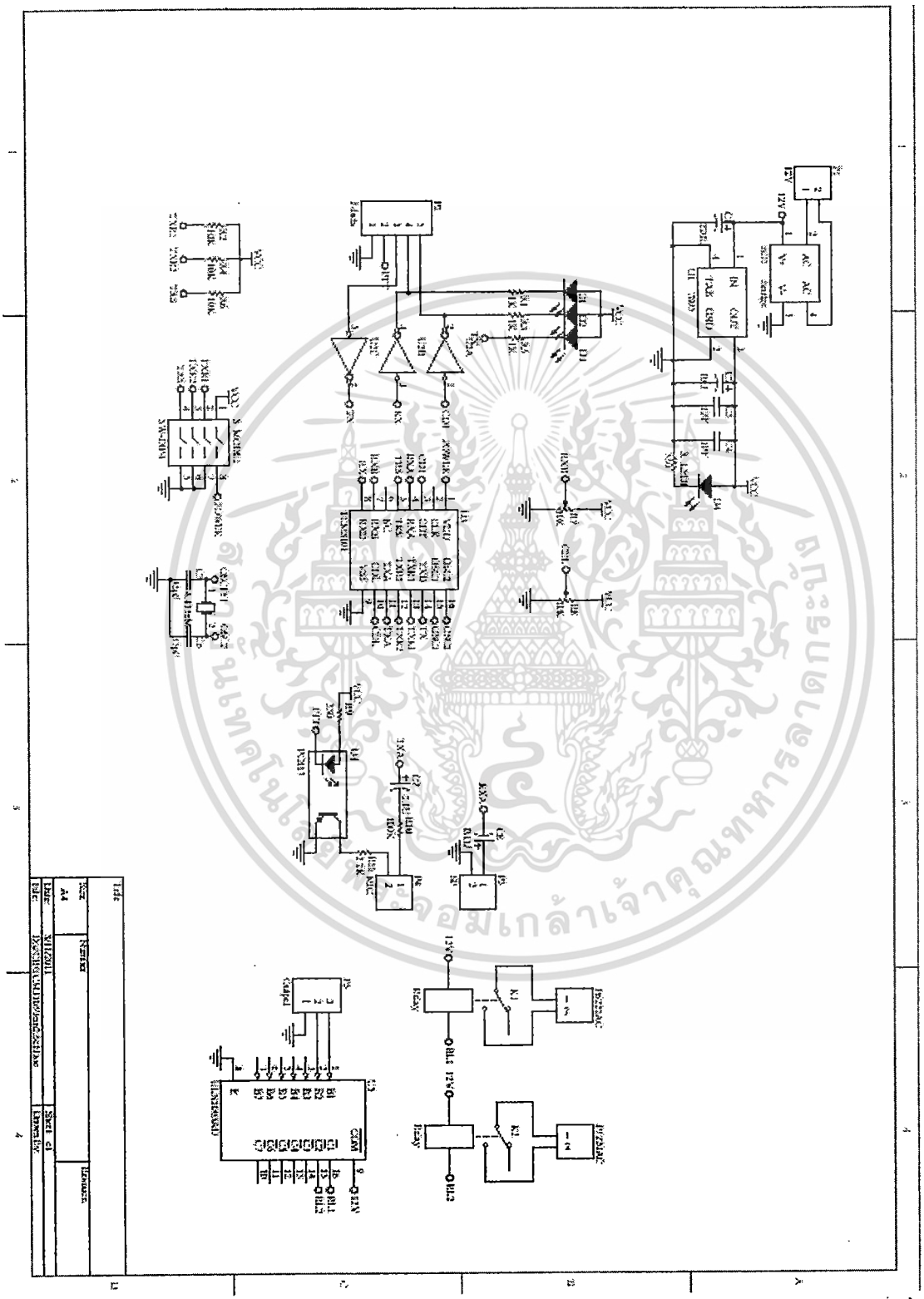
[6] ลาภลอย วานิชอังกูร, " เรียนรู้ด้วยตนเอง OOP C# ASP.NET ", บริษัท ซีเอ็ดยูเคชั่น จำกัด(มหาชน) , พ.ศ. 2550.

[7] บริษัท แอฟซอฟร์เทด " dsPIC30F Programming MPLAB C Complier" พ.ศ. 2552.

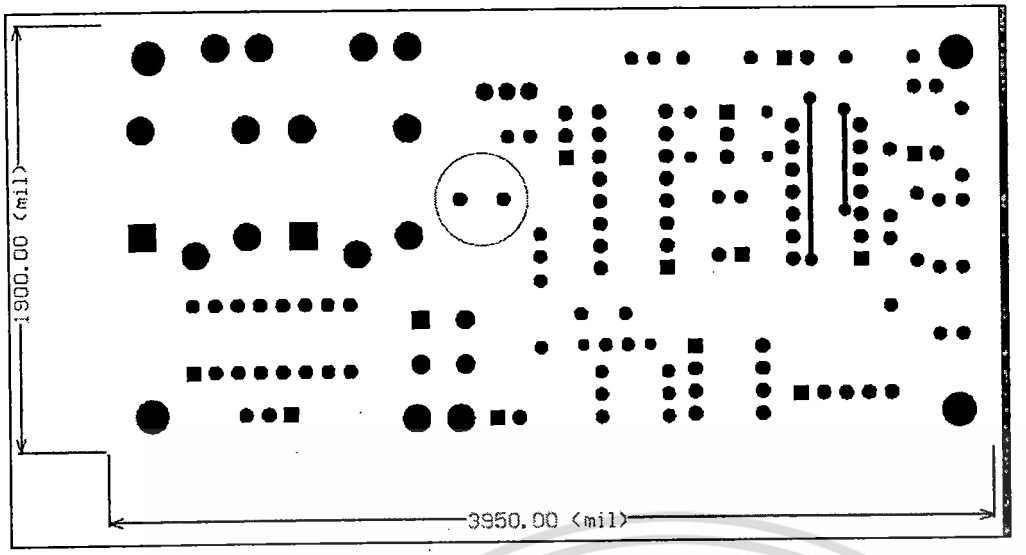
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

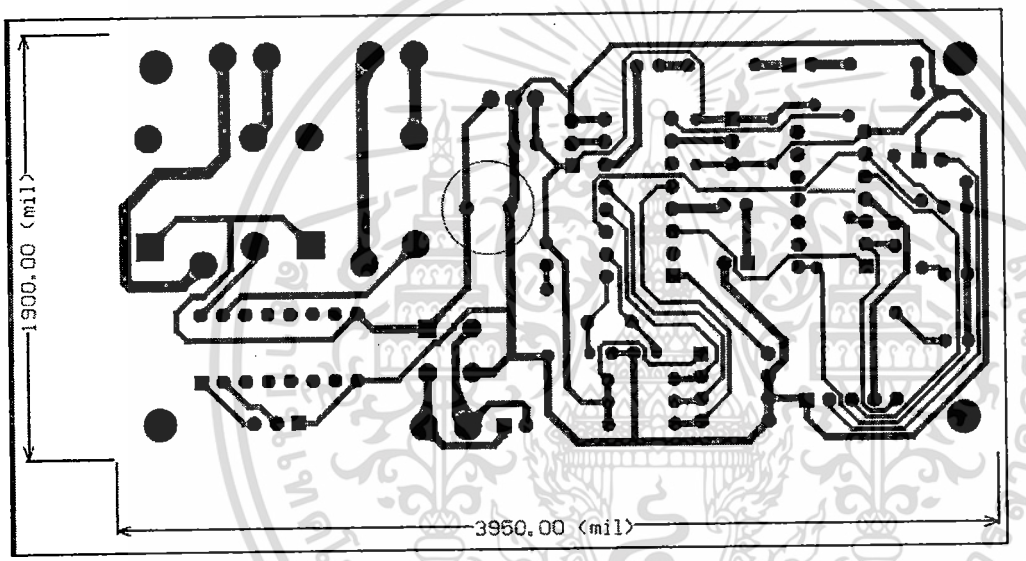
ภาคผนวก (Appendix)

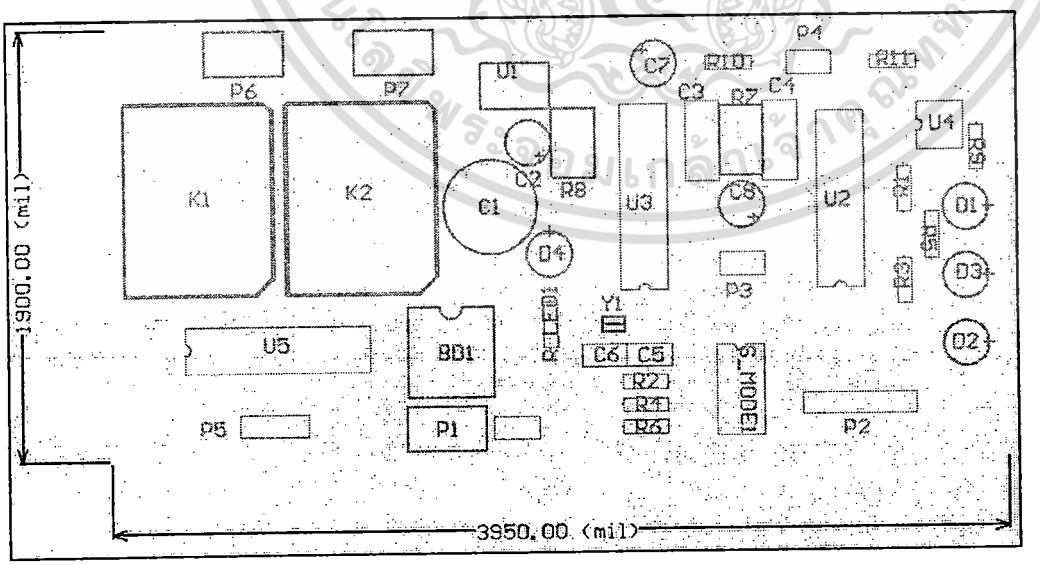
1. วงจร FSK, Relay และส่วนเชื่อมต่อไมโครคอนโทรลเลอร์



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้







เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2. ส่วนของโปรแกรม GUI

เขียนด้วย ภาษา C# 2010 แสดงในส่วนนี้เฉพาะส่วนหลักคือ Form1.cs เท่านั้น ส่วนประกอบที่เหลือทั้งหมดได้บรรจุไว้ใน CD-ROM แล้ว

```
using System;
using System.Collections.Generic;
using System.ComponentModel;
using System.Data;
using System.Drawing;
using System.Linq;
using System.Text;
using System.IO.Ports;
using System.IO;            //ใส่เพื่อใช้งาน RS-232 Port ได้
using System.Windows.Forms;

namespace Test_txrx
{
    public partial class Form1 : Form
    {
        int counter = new int();
        int end_counter = new int();
        byte[] bbyte = new byte[20];  //สำหรับไว้รับข้อมูล
        byte[] t1byte = new byte[20];  // สำหรับไว้ส่งข้อมูล ตัววัดที่หนึ่ง
        byte[] t2byte = new byte[20];  // สำหรับไว้ส่งข้อมูล ตัววัดที่สอง
        int send_num;
        int wait_end;
        bool PTT;
        string data_out;
        byte Ry;
        int time_out = new int();
        public Form1()
        {
            InitializeComponent();
        }

        private void Form1_Load(object sender, EventArgs e)
        {
            int num;

         serialPort1 .DataReceived +=new
SerialDataReceivedEventHandler(serialPort1_DataReceived);
          //กำหนดช่องเลือกการติดต่อ
         string[] comStr = SerialPort.GetPortNames();
          int i = 0;
          foreach (string port in comStr)
          {
              comboBox1.Items.Add(comStr[i]);
              i++;
          }
          comboBox1.SelectedIndex = 0;

            //ให้ช่องว่างต่างๆยังไม่มีข้อมูลในเบื้องต้น
          textBox1.Text = "0";
          textBox2.Text = "0";
          textBox3.Text = "0";
          textBox4.Text = "0";
          textBox5.Text = "0";
          textBox6.Text = "0";
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        textBox7.Text = "";
        textBox8.Text = "";
        textBox9.Text = "";
        textBox10.Text = "";
        textBox11.Text = "";
        textBox12.Text = "";

        t1byte[0] = 0x0F;
        t1byte[1] = 0x00;
        t1byte[2] = 0x01;
        t1byte[3] = 0xEE;
        t1byte[4] = 0x0F;
        t1byte[5] = 0xFF;

        t2byte[0] = 0x0F;
        t2byte[1] = 0x00;
        t2byte[2] = 0x01;
        t2byte[3] = 0x02;
        t2byte[4] = 0xEE;
        t2byte[5] = 0x0F;
        t2byte[6] = 0xFF;


        //ปิดปุ่มที่ยังไม่ใช่ก่อน
        button3.Enabled = false; //ปิดปุ่ม "ส่ง"
        button4.Enabled = false; //ปิดปุ่ม "clear"
        button5.Enabled = false; //ปิดปุ่ม "Relay0 on"
        button6.Enabled = false; //ปิดปุ่ม "Relay0 off"
        button7.Enabled = false; //ปิดปุ่ม "ตัววัดที่ 1"
        button8.Enabled = false; //ปิดปุ่ม "ตัววัดที่ 2"
        button9.Enabled = false; //ปิดปุ่ม "อ่านอย่างเดียว"
        button10.Enabled = false; //ปิดปุ่ม "Relay0 on"
        button11.Enabled = false; //ปิดปุ่ม "Relay0 off"
        button13.Enabled = false; //ปิดปุ่ม "บันทึก"
        button14.Enabled = false; //ปิดปุ่ม " RESET "

        Ry = 0x00;              //กำหนดสถานะเริ่มต้นของรีเลย์
        data_out = "";          //ล้างข้อมูลก่อน
        send_num = 6;           //กำหนดค่าเริ่มต้นของตัวนับคำสั่งเป็น 6 เพราะมีใช้ช่วงเดียวตัวเดียว
        counter = 0;            //กำหนดค่าเริ่มต้นนับเป็นศูนย์
        end_counter = 7;        //กำหนดให้รับข้อมูลจำนวน 7 ไบท์
        time_out = 0;     //กำหนดค่าเริ่มต้นของการรอสัญญาณ ใช้ใน Timer2
        wait_end = 40;    //กำหนดระยะเวลาที่ทำการรอสัญญาณกลับ
        timer1.Enabled = false;
        timer2.Enabled = false;
        timer5.Enabled = true;
        PTT = false;
       // serialPort1.DtrEnable = PTT;

          //==-==========Clera receive buffer
        for (num = 0; num < 20; num++)
            bbyte[num] = 0x00;

    }
//====================== RECEIVED DATA ================================

    private void serialPort1_DataReceived(object sender,
SerialDataReceivedEventArgs e)
    {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
      while (counter < end_counter)
        {
            bbyte[counter] = (byte)serialPort1.ReadByte();
            counter = counter + 1;
        }

}

//======================================================================

private void button1_Click(object sender, EventArgs e)  //ปุ่มคอนเน็ก & setup
{
      int num;

    if (comboBox1.Items.Count > 0)
    {
        if (!serialPort1.IsOpen)
        {
            serialPort1.PortName = comboBox1.Text;
            serialPort1.Open();
        }

    }
        textBox13.Text = " Device Ready!  ";
        button5.Enabled = true;
        button6.Enabled = true;
        button9.Enabled = true;
        button10.Enabled = true;
        button11.Enabled = true;
        button4.Enabled = true;
        button13.Enabled = true;
        button1.Enabled = false;
}

private void button3_Click(object sender, EventArgs e)//ปุ่มกดส่ง
{
    for (int i = 0; i < 20; i++)
    {
       // tbyte[i] = 0x00;
        bbyte[i] = 0x00;
    }
    if (radioButton3.Checked)
    { t1byte[4] = Ry;
    textBox13.Text = "";
    serialPort1.Write(t1byte, 0, 6);    //ส่งคำสั่ง  for #1

    }
    else
    {t2byte[5] = Ry;
    textBox13.Text = "";
    serialPort1.Write(t2byte, 0, 7);    //ส่งคำสั่ง for #2
    }

    counter = 0;
    timer1.Enabled = true;    //สั่งเปิด Timer1 เพื่อแสดงผลข้อมูลที่รับเข้ามา
    time_out = 0;
    timer2.Enabled = true;   //เปิดตัวนับรออินพุท 1-30
}

private void button2_Click(object sender, EventArgs e)//ปุ่มออก
{
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
            serialPort1.Close();

            this.Close();
        }

        private void timer1_Tick(object sender, EventArgs e)  //วนรอรับสัญญาณกลับจากโมดูล
        {
            int num = 19;
            string mo_text=" ";

            if (bbyte[6] == 0xff)
            {

                //แสดงข้อมูลบน LCD และเก็บข้อมูลไว้เพื่อเขียนลงมีเดีย
                textBox13.Text = "     " + bbyte[3].ToString() + "." +
bbyte[4].ToString() + " 'C";
                if (radioButton3.Checked == true) mo_text = "โมดูลตัววัดที่ <1> ";
                else mo_text = "โมดูลตัววัดที่ <2> ";
             data_out += mo_text +" อุณหภูมิ "+ textBox13.Text + " วันที่ "+ label1.Text +
label4.Text + "\r\n";

                //แสดงข้อมูลรีเลย์
                if (bbyte[5] == 0x00)
                {
                    radioButton1.Checked = false;
                    radioButton2.Checked = false;
                }
                else if (bbyte[5] == 0x01)
                {
                    radioButton1.Checked = true;
                    radioButton2.Checked = false;
                }
                else if (bbyte[5] == 0x02)
                {
                    radioButton1.Checked = false;
                    radioButton2.Checked = true;
                }
                else
                {
                    radioButton1.Checked = true;
                    radioButton2.Checked = true;
                }
                timer1.Enabled = false;
                timer2.Enabled = false;  //ปิดการนับ
            }
            else { textBox13.Text = "  WAIT.... "+time_out .ToString ();
            if (time_out == wait_end)
              {
                  time_out = 0;
                timer1.Enabled = false;
                timer2.Enabled = false;
                textBox13.Text = " Time Out      ";
               }
            }

        }

        private void button4_Click(object sender, EventArgs e) //ปุ่มเคลียร์
        {
            data_out = "";
        }
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
private void button5_Click(object sender, EventArgs e)  //Relay#0 On
{

    Ry |= 0x10;
        //button9.Enabled = false;
        button5.Enabled = false;  //รีเลย์หนึ่ง ON
        button6.Enabled = true;   //รีเลย์หนึ่ง OFF
        button9.Enabled = false;  //ส่งอย่างเดียว
        button3.Enabled = true;   //กดส่ง
        button14.Enabled = true;  //รีเซท
}

private void button6_Click(object sender, EventArgs e) //Relay#0 Off
{

        Ry &= 0xE0;
        button6.Enabled = false;
        button14.Enabled = true;
        button5.Enabled = true;
        //button6.Enabled = false;
        button9.Enabled = false;
        button3.Enabled = true;   // Send
}

private void button9_Click(object sender, EventArgs e)  //กำหนดอ่านอุณหภูมิอย่างเดียว
{

    textBox13.Text = "";
    Ry = 0x0F;
    button3.Enabled = true;
    button5.Enabled = false ;
    button6.Enabled = false ;
    button9.Enabled = false;
    button10.Enabled = false;
    button11.Enabled = false;
    button14.Enabled = true;

}

private void button7_Click(object sender, EventArgs e)  //เลือกตัวที่1
{
    send_num = 6;          //กำหนดจำนวนไบท์ของคำสั่งเป็น 6 ไบท์
    t1byte[0] = 0x0f;
    textBox1.Text  = t1byte[0].ToString();
    t1byte[1] = 0x00;
    textBox2.Text  = t1byte[1].ToString();
    t1byte[2] = 0x01;
    textBox3.Text  = t1byte[2].ToString();
    t1byte[3] = 0xEE;
    textBox4.Text = t1byte[3].ToString();
    t1byte[6] = 0x00;
    textBox14.Text = t1byte[6].ToString();
    //เปิดใช้งานปุ่มต่อมา

    button5.Enabled = true;
    button6.Enabled = true;
    button9.Enabled = true;
    button10.Enabled = true;
    button11.Enabled = true ;
    button7.Enabled = false;
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
    button8.Enabled = false;

}

private void button8_Click(object sender, EventArgs e) //เลือกตัวที่2
{
    send_num = 7;          //กำหนดจำนวนไบท์ของคำสั่งเป็น 7 ไบท์
    t2byte[0] = 0x00;
    textBox1.Text = t2byte[0].ToString();
    t2byte[1] = 0x01;
    textBox2.Text = t2byte[1].ToString();
    t2byte[2] = 0x02;
    textBox3.Text = t2byte[2].ToString();
    t2byte[3] = 0xEE;
    textBox4.Text = t2byte[3].ToString();

    //เปิดใช้งานปุ่มต่อมา

    button5.Enabled = true;
    button6.Enabled = true;
    button9.Enabled = true;
    button10.Enabled = true;
    button11.Enabled = true;
    button7.Enabled = false;
    button8.Enabled = false;
}

private void timer2_Tick(object sender, EventArgs e) //รอสัญญาณอินพุท
{
    time_out = time_out + 1;

}

private void button10_Click(object sender, EventArgs e)//ปุ่ม Relay#1 On
{

    Ry |= 0x20;
        button14.Enabled = true;
        button10.Enabled = false;
        button11.Enabled = true;
        button9.Enabled = false;
        button3.Enabled = true;
}

private void button11_Click(object sender, EventArgs e) //ปุ่ม Relay#1 Off
{

    Ry &= 0xd0;
        button14.Enabled = true;
        button11.Enabled = false;
        button10.Enabled = true;
        button9.Enabled = false;
        button3.Enabled = true;
}

private void groupBox1_Enter(object sender, EventArgs e)
{

}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        private void timer3_Tick(object sender, EventArgs e)
        {
            label1.Text = DateTime.Now.ToLongDateString();
            label4.Text = DateTime.Now.ToLongTimeString();
        }

        private void button14_Click(object sender, EventArgs e)
        {
            button3.Enabled = false;
            button10.Enabled = true;
            button11.Enabled = true;
            button9.Enabled = true;
            button5.Enabled = true;
            button6.Enabled = true;
            t1byte[4] = 0x00;
            textBox13.Text = "                    ";

            radioButton1.Checked = false;
            radioButton2.Checked = false;
            timer1.Enabled = false  ;
        }

        private void button13_Click(object sender, EventArgs e) //บันทึกข้อมูลลงมีเดีย
        {
            saveFileDialog1.DefaultExt = "*.txt";
            saveFileDialog1.AddExtension = false;
            saveFileDialog1.Filter = "เท็กซ์ไฟล์ | *.txt";
            if (saveFileDialog1.ShowDialog() == DialogResult.OK)
            {
                string path = saveFileDialog1.FileName;
                StreamWriter sw = new StreamWriter(path);
                sw.Write(data_out );
                sw.Close();
            }
        }

        private void timer4_Tick(object sender, EventArgs e)
        {
            if (serialPort1.CDHolding) textBox13.Text = "   LOGIC 1   ";
            else textBox13 .Text = "   LOGIC 0   ";


        }

        private void timer5_Tick(object sender, EventArgs e)
        {
            if (radioButton3.Checked)
            { pictureBox3 .ImageLocation
="D:\\WORK\\WORK_PROJ\\Project53\\LMSAR_53\\ProgamLMSAR53_C#\\M1.jpg"; }
            else { pictureBox3.ImageLocation =
"D:\\WORK\\WORK_PROJ\\Project53\\LMSAR_53\\ProgamLMSAR53_C#\\M2.jpg"; }
            pictureBox3 .Load ();
        }


    }
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

3. **ส่วนของโปรแกรมเครื่องตัวแม่และตัวลูก**

เขียนด้วย ภาษา C แสดงเฉพาะส่วน main() เท่านั้น โปรแกรมย่อยอื่นๆ หรือส่วนประกอบทั้งหมดได้บรรจุไว้ใน CD-ROM แล้ว

```
/*****************************************************/
/* Examples Program For Main (Connect to PC)         */
/* Hardware   : ET-BASE dsPIC30F4011                 */
/* Target MCU  : dsPIC30F4011                        */
/*          : X-TAL : 7.3728 MHz                     */
/*          : Run 117.9648MHz                        */
/*          : Selec OSC Mode = XT w/PLL 16x          */
/* Compiler   : MPLAB v8.40 + C30 v3.20B             */
/* Author     : MR.ATTASIT LASAKUL                   */
/*****************************************************/
#include <p30f4011.h>                  // dsPIC30F4011 MPU Register
#include "stdio.h"                     // Used "sprintf" Function
#include "uart.h"

/* Setup Configuration For ET-BASE dsPIC30F4011 */
_FOSC(CSW_FSCM_OFF & XT_PLL16);    // Disable Clock Switching,Enable Fail-Salf Clock
                                   // Clock Source = Primary XT + (PLL x 16)
_FWDT(WDT_OFF);                    // Disable Watchdog
_FBORPOR(PBOR_OFF & PWRT_64 & MCLR_EN);// Disable Brown-Out ,Power ON = 64mS,Enable
MCLR
_FGS(CODE_PROT_OFF);               // Code Protect OFF

//-----------------------------------------------:Calc Baud Rate Generator
#define Fcy          29491200.0          // Fosc 7.3728MHz*16/4
#define BAUD_RATE     1200.0             // Baud Rate 1200 bps
#define BAUD_RATE_GEN   (Fcy/(16.0*BAUD_RATE))-1  // Baud Rate Generator

/* End Configuration For ET-BASE dsPIC30F4011 */

#define CodeStop      0xFF     // Key Code บอกจบคำสั่งหรือข้อมูล
#define MY_ID         0x00     //กำหนดเลขหมายตัวเองหมายเลข '0'

//---- (Relay#0)
#define TRIS_Rly0 TRISEbits.TRISE1
#define Relay0  LATEbits.LATE1
//---- LED (Relay#1)
#define TRIS_Rly1 TRISEbits.TRISE2
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define Relay1  LATEbits.LATE2

#define TRIS_RlyLed0 TRISEbits.TRISE3
#define RlyLed0  LATEbits.LATE3
//---- LED (Relay#1)
#define TRIS_RlyLed1 TRISEbits.TRISE4
#define RlyLed1  LATEbits.LATE4

//---- (PTT)
#define TRIS_PTT TRISDbits.TRISD0
#define PTT  LATDbits.LATD0
//---- (CDT)
#define TRIS_CDT TRISDbits.TRISD1
#define CDT  LATDbits.LATD1


// Character LCD Interface Pins
#define TRIS_DATA_PIN_4    TRISBbits.TRISB0          // Direction D4
#define TRIS_DATA_PIN_5    TRISBbits.TRISB1          // Direction D5
#define TRIS_DATA_PIN_6    TRISBbits.TRISB2          // Direction D6
#define TRIS_DATA_PIN_7    TRISBbits.TRISB3          // Direction D7
#define TRIS_RS          TRISBbits.TRISB4            // Direction RS
#define TRIS_E           TRISBbits.TRISB5            // Direction E

#define DATA_PIN_4       LATBbits.LATB0              // RB0 = D4 LCD
#define DATA_PIN_5       LATBbits.LATB1              // RB1 = D5 LCD
#define DATA_PIN_6       LATBbits.LATB2              // RB2 = D6 LCD
#define DATA_PIN_7       LATBbits.LATB3              // RB3 = D7 LCD
#define RS_PIN           LATBbits.LATB4              // RB4 = RS LCD
#define E_PIN            LATBbits.LATB5              // RB5 = E  LCD

/* Display ON/OFF Control */
#define DON            0x0F                          // Display on
#define DOFF           0x0B                          // Display off
#define CURSOR_ON        0x0F                        // Cursor on
#define CURSOR_OFF       0x0D                        // Cursor off
#define BLINK_ON         0x0F                        // Cursor Blink
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define BLINK_OFF        0x0E                    // Cursor No Blink

/* Cursor or Display Shift */
#define SHIFT_CUR_LEFT    0x13                   // Cursor shifts to the left
#define SHIFT_CUR_RIGHT   0x17                   // Cursor shifts to the right
#define SHIFT_DISP_LEFT   0x1B                   // Display shifts to the left
#define SHIFT_DISP_RIGHT  0x1F                   // Display shifts to the right

/* Function Prototypes */
void Initial_4bitLCD(void);                      // Initial LCD Interface
void SetCGRamAddr(unsigned char);
void SetDDRamAddr(unsigned char);                // Set Cursor Address
void WriteCmdLCD(unsigned char);                 // Write Command
void WriteDataLCD(unsigned char);                // Write Data
void PutsLCD(unsigned char*);                    // Print Message
void Delay_tW_LCD(void);                         // Enable Pulse Delay
void Busy_LCD(void);                             // Wait LCD Busy
void Delay(unsigned long int);                   // Delay Time Function
void init_uart1200(void);
void init_uart2_1200(void);
void Uart1_Init600(void);
//---- For DS1820
#define TRIS_DS1820   TRISEbits.TRISE0           // Direction E0
#define DOUT_DS1820   LATEbits.LATE0             // OUT
#define DIN_DS1820    PORTEbits.RE0              // IN


unsigned char tempresbit(void);
void tempwrbyte(unsigned char dat);
char temprdbyte(void);
char rdtemp(void);

void Forward(void);
void Backward(void);

char buf_Tempc[20]={'\0','\0','\0','\0','\0'     // Set buffer for display Temp
                    ,'\0','\0','\0','\0','\0'
                    ,'\0','\0','\0','\0','\0'
                    ,'\0','\0','\0','\0','\0'};


//-----------------------:Global variables
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
unsigned char Buf[80];          // Received Command/data is stored in array Buf
unsigned char data_tmp[80];
unsigned char ch;
unsigned char c_buf=0;
unsigned char addr_back;        //ไว้เก็บ My เพื่อส่งกลับ
unsigned char Read_black;
unsigned char Flag;


//=============================================================
//----------------------:Main Program:----------------------
//=============================================================

int main(void)
{
        unsigned int i,j;
        unsigned long k,l;
        unsigned char r_dat,r_dat1,r_dat2,r_dat3 = 0;

        Delay(2000000);
        Flag=0;                 //กำหนดให้สถานะเริ่มให้เป็นการส่งผ่าน

        Initial_4bitLCD();      // Initial LCD 4 Bit Interface
   TRIS_PTT = 0;                //Set to Output
   TRIS_CDT = 1;                //กำหนดเป็นอินพุทสำหรับสัญญาณ CDT
   PTT = 1;                     // ปิดการส่ง PTT = OFF;

TRIS_RlyLed0 = 0;
TRIS_RlyLed1 = 0;

 Loop:
 while(1){
  //เริ่มแสดงผลจอ LCD
  SetDDRamAddr(0x00);
  PutsLCD((unsigned char *)" PROGRAM LMSAR  ");
  SetDDRamAddr(0x40);
  PutsLCD((unsigned char *)"  WAIT FOR PC   ");
  Delay(2000000);

  Uart1_Init600();              // set 600 pbs for module (No Int)
  init_uart2_1200();            //set 1200 pbs  (for PC No Int)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//receive command from PC from UART2
  do{
          while(!DataRdyUART2());     // Wait for PC IN
     r_dat = ReadUART2();
                }
        while(r_dat!=0x0f);

 //หากใช่ก็ทำการรับคำสั่งที่เหลือจนหมดที่ 0xff
 i=0;
                  do{
         while(!DataRdyUART2());
     r_dat = ReadUART2();
                  Buf[i]=r_dat;
                  ++i;
        }while(r_dat != 0xFF);

  SetDDRamAddr(0x40);
  sprintf(buf_Tempc," COMMAND(PC) IN ");
  PutsLCD((unsigned char *)buf_Tempc);
  Delay(2000000);
  //ส่งคำสั่ง ออกไปสู่ Client#1 โดยผ่าน UART1

  SetDDRamAddr(0x40);
  sprintf(buf_Tempc,"Send command to #1");
  PutsLCD((unsigned char *)buf_Tempc);

      PTT = 0;          //ON PTT
      Uart1_Init600();          //เปิดการสื่อสารอีกทีเพราะจะมีการรบกวนจากวิทยุ
      Delay(2000000);           //รอระยะหนึ่งก่อน

  putcUART1(0x0f);              //ส่งหัวระหัสไป 0x0F
  while(BusyUART1());

  for(j=0;j<(i-1);j++)          //ส่งที่เหลือไปให้หมด
   { putcUART1(Buf[j]);
     while(BusyUART1());
   }
  putcUART1(0xff);              //ส่งปิดท้าย 0xFF
  while(BusyUART1());
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        Delay(1000000);          //ยืดระยะส่งไปอีกนิดก่อนปิดการส่ง
    PTT = 1;                     //OFF PTT

 // รอข้อมูลตอบกลับมาจากโมดูลต่างๆ
Loop1:
        SetDDRamAddr(0x40);
  sprintf(buf_Tempc,"Wait for data #1");
  PutsLCD((unsigned char *)buf_Tempc);
  r_dat = 0;
        Uart1_Init600();  //กำหนดช่องรับอีกที เพราะมีการรบกวนจากวิทยุ

  do{
        k=0;
   while(CDT)          //รอตรวจสัญญาณจากโมดูลเข้ามา
  {
    k++;
   if(k== 10000000)
    {
  SetDDRamAddr(0x00);
  sprintf(buf_Tempc,"    TIME OUT    ");
  PutsLCD((unsigned char *)buf_Tempc);
  SetDDRamAddr(0x40);
  sprintf(buf_Tempc,"                ");
  PutsLCD((unsigned char *)buf_Tempc);

  for(i=0;i<80;i++) Buf[i]=0;
  Delay(3000000);
  i=0;
  k=0;
   goto Loop;
    }
   }
        k=0;
  while(!DataRdyUART1())
   {
  k++;
    if(k== 10000000)
     {
                SetDDRamAddr(0x00);
                sprintf(buf_Tempc,"    TIME OUT    ");
                PutsLCD((unsigned char *)buf_Tempc);
                SetDDRamAddr(0x40);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
        sprintf(buf_Tempc,"  No data return    ");
                PutsLCD((unsigned char *)buf_Tempc);

                for(i=0;i<80;i++) Buf[i]=0;
        Delay(3000000);
        i=0;k=0;
        goto Loop;
    }
        }
      r_dat = ReadUART1();
   }while(r_dat != 0x0f);        //หากไม่ใช่ 0x0f ก็รอให้ไปเรื่อยๆ

//หากใช่ข้อมูลก็จะรับเข้ามาทั้งหมดต่อไป
      Buf[0] = 0x0f;
      while(!DataRdyUART1());
      Buf[1] = ReadUART1();
      while(!DataRdyUART1());
      Buf[2] = ReadUART1();
      while(!DataRdyUART1());
      Buf[3] = ReadUART1();
      while(!DataRdyUART1());
      Buf[4] = ReadUART1();
      while(!DataRdyUART1());
      Buf[5] = ReadUART1();
      while(!DataRdyUART1());
      Buf[6] = ReadUART1();

//ตรวจสอบว่าเป็นการเลือกส่งข้อมูลกลับมาที่ตัวเองหรือไม่
  if (Buf[2]!= MY_ID)
{
        for(i=0;i<80;i++) Buf[i]=0;
   goto Loop1;
                        }

//ตรวจสอบว่าข้อมูลถูกต้องหรือไม่ หากถูกต้องก็ทำการส่งกลับสู่ PC ต่อไป
  if(Buf[6] != 0xff)
  {
  SetDDRamAddr(0x00);
  sprintf(buf_Tempc,"  DATA RECEIVE  ");
  PutsLCD((unsigned char *)buf_Tempc);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
   SetDDRamAddr(0x40);
   sprintf(buf_Tempc,"    ERROR     ");
   PutsLCD((unsigned char *)buf_Tempc);

  for(i=0;i<80;i++) Buf[i]=0;

   Delay(3000000);
    goto Loop;
   }
//ส่งทั้งหมดกลับสู่ pC
   SetDDRamAddr(0x00);
   sprintf(buf_Tempc," TRANSFER DATA  ");
   PutsLCD((unsigned char *)buf_Tempc);
   SetDDRamAddr(0x40);
   sprintf(buf_Tempc,"    TO PC      ");
   PutsLCD((unsigned char *)buf_Tempc);
   Delay(3000000);
   for(j=0;j<7;j++)
    { putcUART2(Buf[j]);
      while(BusyUART2());
    }
  for(i=0;i<80;i++) Buf[i]=0;
 }
 return 0;
}
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
/*******************************************************/
/* Examples Program For Clients                        */
/* Hardware   : ET-BASE dsPIC30F4011                   */
/* Target MCU  : dsPIC30F4011                          */
/*          : X-TAL : 7.3728 MHz                       */
/*          : Run 117.9648MHz                          */
/*          : Selec OSC Mode = XT w/PLL 16x            */
/* Compiler   : MPLAB v8.40 + C30 v3.20B               */
/* Author     : MR.ATTASIT LASAKUL                     */
/*******************************************************/
#include <p30f4011.h>   // generic header file for dsPIC
#include "uart.h"       // uart module
#include <string.h>     // String.h standard header
#include "stdio.h"
// Setup Configuration For ET-BASE dsPIC30F4011
_FOSC(CSW_FSCM_OFF & XT_PLL16);          // Disable Clock Switching,Enable Fail-Salf Clock
                                         // Clock Source = Primary XT + (PLL x 16)
_FWDT(WDT_OFF);                          // Disable Watchdog
_FBORPOR(PBOR_OFF & PWRT_64 & MCLR_EN);// Disable Brown-Out ,Power ON = 64mS,Enable
MCLR
_FGS(CODE_PROT_OFF);                     // Code Protect OFF

//-----------------------------------------------------:Calc Baud Rate Generator
#define Fcy          29491200.0            // Fosc 7.3728MHz*16/4
#define BAUD_RATE     1200.0              // Baud Rate 1200 bps
#define BAUD_RATE_GEN   (Fcy/(16.0*BAUD_RATE))-1  // Baud Rate Generator

#define KeyEnter      13                 // Key Code

/* End Configuration For ET-BASE dsPIC30F4011 */

#define CodeStart     0x0F    // Key Code บอกเริ่มคำสั่งหรือข้อมูล
#define CodeStop      0xFF    // Key Code บอกจบคำสั่งหรือข้อมูล
//#define MY_ID       0x02    //กำหนดเลขหมายตัวเองหมายเลข '2'
#define MY_ID         0x01    //กำหนดเลขหมายตัวเองหมายเลข '1'

//----  (Relay#0)
#define TRIS_Rly0 TRISEbits.TRISE1
#define Relay0  LATEbits.LATE1
//---- LED (Relay#1)
#define TRIS_Rly1 TRISEbits.TRISE2
#define Relay1  LATEbits.LATE2
//---- LED ---
#define TRIS_LED_Rly0 TRISEbits.TRISE3
#define LED_Relay0  LATEbits.LATE3
//---- LED (Relay#1)
#define TRIS_LED_Rly1 TRISEbits.TRISE4
#define LED_Relay1  LATEbits.LATE4

//----  (PTT)
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
#define TRIS_PTT TRISDbits.TRISD0
#define PTT  LATDbits.LATD0
//---- (CDT)
#define TRIS_CDT TRISDbits.TRISD1
#define CDT  LATDbits.LATD1

// Character LCD Interface Pins
#define TRIS_DATA_PIN_4    TRISBbits.TRISB0          // Direction D4
#define TRIS_DATA_PIN_5    TRISBbits.TRISB1          // Direction D5
#define TRIS_DATA_PIN_6    TRISBbits.TRISB2          // Direction D6
#define TRIS_DATA_PIN_7    TRISBbits.TRISB3          // Direction D7
#define TRIS_RS        TRISBbits.TRISB4              // Direction RS
#define TRIS_E         TRISBbits.TRISB5          // Direction E

#define DATA_PIN_4       LATBbits.LATB0              // RB0 = D4 LCD
#define DATA_PIN_5       LATBbits.LATB1              // RB1 = D5 LCD
#define DATA_PIN_6       LATBbits.LATB2              // RB2 = D6 LCD
#define DATA_PIN_7       LATBbits.LATB3              // RB3 = D7 LCD
#define RS_PIN         LATBbits.LATB4            // RB4 = RS LCD
#define E_PIN          LATBbits.LATB5            // RB5 = E  LCD

/* Display ON/OFF Control */
#define DON            0x0F                      // Display on
#define DOFF           0x0B                      // Display off
#define CURSOR_ON        0x0F                    // Cursor on
#define CURSOR_OFF       0x0D                    // Cursor off
#define BLINK_ON         0x0F                    // Cursor Blink
#define BLINK_OFF        0x0E                    // Cursor No Blink

/* Cursor or Display Shift */
#define SHIFT_CUR_LEFT    0x13                   // Cursor shifts to the left
#define SHIFT_CUR_RIGHT   0x17                   // Cursor shifts to the right
#define SHIFT_DISP_LEFT   0x1B                   // Display shifts to the left
#define SHIFT_DISP_RIGHT  0x1F                   // Display shifts to the right

/* Function Prototypes */
void Initial_4bitLCD(void);                      // Initial LCD Interface
void SetCGRamAddr(unsigned char);
void SetDDRamAddr(unsigned char);                // Set Cursor Address
void WriteCmdLCD(unsigned char);                 // Write Command
void WriteDataLCD(unsigned char);                // Write Data
void PutsLCD(unsigned char*);                    // Print Message
void Delay_tW_LCD(void);                         // Enable Pulse Delay
void Busy_LCD(void);                             // Wait LCD Busy
void Delay(unsigned long int);                   // Delay Time Function

/*  Function RS-232 */
void Uart1_PrintStr(unsigned char *str_uart);
void Delay_MS(unsigned int ms);
void init_uart1(void);
void init_uart2(void);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหาและต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
void  Uart1_Init600(void);

//---- For DS1820
#define TRIS_DS1820    TRISEbits.TRISE0                 // Direction E0
#define DOUT_DS1820     LATEbits.LATE0                  // OUT
#define DIN_DS1820      PORTEbits.RE0                   // IN

unsigned char tempresbit(void);
void tempwrbyte(unsigned char dat);
char temprdbyte(void);
char rdtemp(void);
void Forward(void);
void Backward(void);

char buf_Tempc[20]={'\0','\0','\0','\0','\0'                // Set buffer for display Temp
                                        ,'\0','\0','\0','\0','\0'
                                        ,'\0','\0','\0','\0','\0'
                                        ,'\0','\0','\0','\0','\0'};


//------------------------:Global variables
unsigned char Buf[80];          // Received Command/data is stored in array Buf
unsigned char data_tmp[80];
unsigned char ch;
unsigned char c_buf=0;
unsigned char addr_back;        //ไว้เก็บ My เพื่อส่งกลับ
unsigned char Read_black;
unsigned char Flag;
unsigned char tempb;
unsigned char tempf;
unsigned char r_dat;

//============================================================
//------------------------:Main Program:----------------------
//============================================================

int main(void)
{
        unsigned int i,j;
   unsigned long l,k;

        Delay(2000000);
        Flag=0;                         //กำหนดให้สถานะเริ่มให้เป็นการส่งผ่าน
        Read_black = 0;                 //กำหนดให้ค่าเริ่มต้นเป็นไม่เป็นตัวที่ถูกอ่าน
    TRIS_CDT = 1;                       //กำหนดอินพุทสำหรับ CDT
        TRIS_DS1820  = 1;               //กำหนดเป็น อินพุท สำหรับอุณหภูมิ
    Initial_4bitLCD();                  // Initial LCD 4 Bit Interface
    Uart1_Init600();                    // Initial RS-232 Uart1
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
//กำหนด รีเลย์ตัวที่หนึ่งและสองให้ใช้ช่องพอร์ท E2,E3
 TRIS_Rly0 = 0;          // Set direction control RF2 Output
 Relay0 = 1;            // Clear RF2 กำหนดให้ตัวรีเลย์อยู่ในสภาวะ OFF ก่อน
 TRIS_Rly1 = 0;          // Set direction control RF3 Output
 Relay1 = 1;            // Clear RF3 กำหนดให้ตัวรีเลย์อยู่ในสภาวะ OFF ก่อน
 TRIS_LED_Rly0 = 0;
 TRIS_LED_Rly1 = 0;
 LED_Relay0 = 1;
 LED_Relay1 = 1;
 TRIS_PTT = 0;            // Set to OUTPUT PORT
 PTT = 1;                 // OFF PTT

 //เริ่มแสดงผลจอ LCD
  SetDDRamAddr(0x00);
  PutsLCD((unsigned char *)" PROGRAM LMSAR  ");
  SetDDRamAddr(0x40);
  PutsLCD((unsigned char *)"  DS1820 TEMP   ");
  Delay(7000000);
  Relay0 = 0;
  Relay1 = 0;
  init_uart1();          // Initialize the UART1 1200 Bps
  i = 0;
  ch = 0;
 tempb=0;
 tempf=0;
  Delay(3000000);

Loop:

  for(;;)
        {
                SetDDRamAddr(0x00);
     PutsLCD((unsigned char *)" PROGRAM LMSAR  ");
                SetDDRamAddr(0x40);
                PutsLCD((unsigned char *)"   Wait....   ");

     Uart1_Init600();     //กำหนดสื่อสารเป็น 600 pbs

if(Flag == 0x01) //หากเป็นการส่งผ่านข้อมูลกลับก็ต้องมีการนับรอเวลาช่วงหนึ่งเท่านั้น
{
do{
          k=0;
   while(CDT)  //รอตรวจสัญญาณจากโมดูลเข้ามา
  {
    k++;
   if(k== 10000000)
    {
  SetDDRamAddr(0x00);
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
  sprintf(buf_Tempc,"     TIME OUT     ");
  PutsLCD((unsigned char *)buf_Tempc);
  SetDDRamAddr(0x40);
  sprintf(buf_Tempc,"                  ");
  PutsLCD((unsigned char *)buf_Tempc);

 for(i=0;i<80;i++) Buf[i]=0;

  Delay(3000000);
  i=0;
  k=0;
   goto Loop;
    }
  }
        k=0;
  while(!DataRdyUART1())
  {
k++;
   if(k== 10000000)
    {
                SetDDRamAddr(0x00);
                sprintf(buf_Tempc,"     TIME OUT     ");
                PutsLCD((unsigned char *)buf_Tempc);
                SetDDRamAddr(0x40);
         sprintf(buf_Tempc,"  No data return    ");
                PutsLCD((unsigned char *)buf_Tempc);

               for(i=0;i<80;i++) Buf[i]=0;
        Delay(3000000);
        i=0;k=0;
    Flag = 0; //คืนสถานะเป็นเริ่มใหม่(รับคำสั่ง)
        goto Loop;
    }
        }
      r_dat = ReadUART1();
   }while(r_dat != 0x0f);        //หากไม่ใช่ 0x0f ก็รอให้ไปเรื่อยๆ

  //หากใช่ข้อมูลก็จะรับเข้ามาทั้งหมดต่อไป
      Buf[0] = 0x0f;
      i = 1;
      do{
     while(!DataRdyUART1());
      r_dat = ReadUART1();
      Buf[i]=r_dat;
      i++;
                }while(r_dat!=0xff);

}
 else
  {
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
do{
      while(CDT);  //รอตรวจสัญญาณจากโมดูลเข้ามา
   while(!DataRdyUART1());
    r_dat = ReadUART1();
 }while(r_dat != 0x0f);        //หากไม่ใช่ 0x0f ก็รอให้ไปเรื่อยๆ

              //หากใช่ข้อมูลก็จะรับเข้ามาทั้งหมดต่อไป
    Buf[0] = 0x0f;
    i = 1;
    do{
   while(!DataRdyUART1());
    r_dat = ReadUART1();
    Buf[i]=r_dat;
    i++;
              }while(r_dat!=0xff);
     }

                          i=0;
                          ch=0;
       //ตรวจว่าเป็นการเลือกใช้งานตัวเองเป็นตัวอ่านหรือผ่านข้อมูลหรือไม่?
            if (Buf[2]== MY_ID)
      {
   //ตรวจสอบว่าเป็นการอ่านค่าตัวเองหรือไม่ หากใช่ก็ทำการอ่านแล้วส่งกลับทันที
    if (Buf[3]==0xEE)  //หากเป็นการส่งกลับข้อมูลปกติ ในช่องตรงนี้จะมีค่าเป็นอุณหภูมิ
              //ซึ่งเป็นค่าที่ไม่มีทางถึง EE = 238 ได้
     {
            Flag = 0x01;       //เป็นการกำหนดให้ส่งกลับข้อมูล(ทำงานกับข้อมูล)
       Read_black = 0x01;  //กำหนดโค๊ด 0x01 ให้เป็นการอ่านค่าข้อมูลตัวเองกลับไป
     }
     else  if (Flag != 0x00) //เป็นการอยู่ในสถานะส่งกลับ
         {
              Read_black = 0x00;  //ไม่มีการอ่านข้อมูลตนเองกลับไป
            }

     switch (Flag)
      {
          case 0x00 :
            Flag = Flag+1;  //กลับสถานะของแฟกเพื่อบอกการทำงานกับคำสั่ง(0)หรือข้อมูล(1)
          SetDDRamAddr(0x00);
                                sprintf(buf_Tempc," TRANSFERING  ");
                                PutsLCD((unsigned char *)buf_Tempc);
                                SetDDRamAddr(0x40);
                          sprintf(buf_Tempc,"COMMAND FORWARD ");
                                PutsLCD((unsigned char *)buf_Tempc);
       Delay(2000000);
                          Forward();
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

```
                    break;
            case 0x01 :
             Flag = Flag -1 ;  //กลับสถานะของแฟกเพื่อบอกการทำงานกับคำสั่ง(0)หรือข้อมูล(1)
        SetDDRamAddr(0x00);
                                sprintf(buf_Tempc," TRANSFERING  ");
                                PutsLCD((unsigned char *)buf_Tempc);
                                SetDDRamAddr(0x40);
                            sprintf(buf_Tempc," DATA BACKWARD  ");
                                PutsLCD((unsigned char *)buf_Tempc);
             Delay(2000000);
                    Backward();
                                 break;
            default   : break;
    }
        }

  Delay(100000);
 }
 return 0;
}
```

4. **ส่วนของ ข้อมูลคุณสมบัติอุปกรณ์**

ข้อมูลเฉพาะของอุปกรณ์ที่สำคัญ ที่ใช้ในงานวิจัยได้ จัดเป็น ไฟล์ pdf บรรจุไว้ใน CD-ROM แล้ว.

- ไมโครคอนโทรลเลอร์เบอร์ dsPIC30F4011
- ไอซีวัดอุณหภูมิเบอร์ DS1820
- ไอซี FSK MODOLATOR เบอร์ TCM3101J

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้