สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

แบบรายงานโครงการวิจัย
โดยใช้เงินรายได้คณะวิศวกรรมศาสตร์
ประจำปี 2552

ชื่อโครงการ

เครื่องวัดระยะไกลผ่านวิทยุสื่อสาร

(Measuring machine using walkie-talkie)

RC1+
TJ
223
.T4
ว 361ค
ค.1
ตช หนู.....
เลขทะเบียน.116099
วัน,เดือน,ปี...2 พ.ค. 2554

ผู้รับผิดชอบโครงการ

หัวหน้าโครงการวิจัย.....รศ.ดร.อรรถสิทธิ์ หล่าสกุล

หน่วยงาน

สาขาวิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

พระจอมเกล้าเจ้าคุณทหารลาดกระบัง

b.12313183
i.....

บทคัดย่อ

ปัจจุบันการตรวจสอบบันทึกค่าการเปลี่ยนแปลงต่างๆ ติดต่อกันในช่วงเวลานานๆ เพื่อการคาดการณ์หรือวิเคราะห์เหตุการณ์อันอาจจะเกิดขึ้นได้ในอนาคต เป็นสิ่งจำเป็นมากต่อการพิจารณาเพื่อหาแนวทางแก้ไขหรือบรรเทาปัญหาต่างๆที่อาจเกิดขึ้นได้ ตัวอย่าง เช่น การติดตาม ค่าอุณหภูมิ, ค่าระดับน้ำ, ค่า ความชื้น เหล่านี้เป็นต้น ซึ่งหากได้มีการตรวจสอบอย่างต่อเนื่องไปเป็นระยะนานๆ ก็สามารถนำค่าเหล่านั้น มาหาสถิติความน่าจะเป็นต่างๆ เพื่อจะได้หาวิธีการแก้ไขปัญหาเหล่านั้นได้ตรงจุดที่สุด ต่อไป ดังเช่น ปัญหาน้ำท่วมซ้ำซากเป็นต้นแต่การวัดค่าเหล่านี้จำเป็นต้องมีการนำเอาเซนเซอร์ไปติดตั้งไว้ในที่ต่างๆกัน เป็นจำนวนมาก และส่วนมากก็จะเป็นสถานที่ๆ ห่างไกลจะความเจริญ เช่น แนวภูเขา ซึ่งจะไม่มีแม้สัญญาณโทรศัพท์ ในพื้นที่นั้นๆ ดังนั้น หากต้องการทราบว่าค่าที่ต้องการวัดนั้นมีค่าเท่าได ก็จำเป็นต้องออกเดินป่า เข้าไปจดค่าเหล่านั้น ซึ่งจะศูนย์เสียทั้งเวลาและงบประมาณ, กำลังบุคคล ในการบันทึกค่าเหล่านั้นในแต่ละครั้ง ยิ่งไปกว่านั้น หากต้องมีการเก็บค่าเหล่านั้นเป็นระยะบ่อยครั้งเท่าไดก็จะยิ่งทำให้ เสียค่าใช้จ่ายมากขึ้นตามไปด้วยเท่านั้น ดังนั้น ในงานวิจัยฉบับนี้จึง นำเสนอการออกแบบและสร้าง ระบบ(เครื่อง)ตรวจสอบวัดค่าระยะไกลแบบใหม่ โดยค่าต่างๆที่ต้องการวัดนั้นจะถูกวัดเก็บไว้โดยการทำงานของระบบเซนเซอร์และไมโครคอนโทรลเลอร์ที่บรรจุโปรแกรมจัดการการทำงานเพื่อตรวจวัดค่าต่างๆอยู่ในเครื่องดังกล่าวแล้ว โดยผู้ใช้งานสามารถตรวจดูค่าผลของการวัดเหล่านั้นโดยไม่ต้องเดินทางเข้าสู่พื้นที่ติดตั้งเซนเซอร์เลย ดังนั้นในการใช้งานเครื่องนี้ก็จะถูกนำไปติดตั้งในสถานที่ต่างๆหลายจุด ให้ครอบคลุมบริเวณที่ผู้ใช้งานต้องการและอยู่ในย่านกำลังส่งรับของวิทยุที่ใช้งาน และเมื่อผู้ใช้ต้องการทราบข้อมูลค่าตรวจวัดที่จุดใด ก็สามารถจะใช้ซอฟร์แวร์ที่ได้ออกแบบเป็นแบบ GUI ไว้บนคอมพิวเตอร์ทั่วไป เพื่อเข้าไปดูข้อมูลได้ทุกจุดทุกเวลาได้ตามต้องการ และโดยใช้ระบบสื่อสารที่เป็นวิทยุที่ต่อเป็นระบบอัตโนมัติแทนระบบ GSM หรือ ระบบอินเทอร์เนท ซึ่งไม่สามารถเข้าถึงพื้นที่ดังกล่าวได้ ดังนั้นด้วยการใช้ระบบสื่อสารวิทยุนี้ต่อเป็นระบบแทน จึงทำให้ไม่จำเป็นที่จะต้องส่งบุคคลเข้าไปจดบันทึกข้อมูลโดยตรง ส่งผลทำให้ลดค่าใช้จ่ายได้เป็นอย่างมาก และนอกจากนี้ ระบบยังออกแบบให้สามารถติดตั้งและใช้งานได้อย่างสะดวกอีกด้วย.

Abstract

The Monitoring system has been used for many years ago. It was utilized for recording many kinds of data for analysis. Then, these data will be used to predict what it should occur. According to data analysis's result, we can carefully plan to hand on that problem. The monitoring system, in this thesis, consists of two types. Type one for using with host computer (1 unit) and type two are clients (this thesis has 1 unit). All units consist of Microcontroller and interfacing circuit for amateur

radio. For client unit its addition some sensor (depend on purpose of measuring) and Solar system but host unit.

On the operation, all clients unit (so call beacons unit) will be installed to many positions on the certain area (usually is a very big area) in rang of propagation radio signal. User can easily access to all of client unit for getting data by only clicking on screen (GUI) of host computer. With intelligent programming software that was installed on host/client unit, all units can automatically working together. We don't need to journey to client units for checking/recording data any more. This will reduce very much of payment.

สารบัญ

หัวข้อ	หน้า
สารบัญตาราง	5
สารบัญรูป	5
บทที่ 1 บทนำ	6
บทที่ 2 เอกสารและงานวิจัยที่เกี่ยวข้อง/การทบทวนวรรณกรรม	7
บทที่ 3 วิธีดำเนินการวิจัยและผลการวิจัย	8
3.1 แผนงานระยะต่างๆของการดำเนินงานวิจัย	8
3.2 การสร้างส่วนของฮาร์ดแวร์	9
3.2.1 ส่วนของเซนเซอร์อุณหภูมิ	11
3.2.2 ส่วนของรีเลย์ตัดต่อไฟฟ้า	12
3.3 การสร้างส่วนของซอฟร์แวร์	13
3.3.1 ซอฟร์แวร์ส่วนของโปรแกรมหลักบน PC	13
3.3.2 ซอฟร์แวร์ส่วนของตัวลูก	15
3.4 ส่วนของฟอร์แมทคำสั่งในการรับส่งข้อมูล	17
3.4.1 Command Format (ส่งออกไปเพื่ออ่านข้อมูลและสั่งงานรีเลย์)	17
3.4.2 Data Format (ส่งกลับข้อมูลกลับไปสู่คอมพิวเตอร์)	18
3.5 ภาพรวมการใช้งาน	20
บทที่ 4 อภิปรายผลการวิจัยและวิจารณ์	22
บทที่ 5 สรุปและข้อเสนอแนะ	23
บรรณานุกรม	24
ภาคผนวก ส่วนฮาร์แวร์ที่สำคัญ และซอฟร์แวร์	25

สารบัญตาราง

ตารางที่ 1 แสดงช่วงเวลาการทำงาน 8

สารบัญรูป

รูปที่ 1 (a) แสดงเครื่องที่มีจำหน่ายแต่จะใช้ในการเปิดปิดอุปกรณ์ไฟฟ้าเท่านั้น 7
(b) แสดงเครื่องควบคุมการกระจายเสียงผ่านระบบสื่อสารวิทยุ
รูปที่ 2 บล็อกไดอะแกรมของส่วนแรก (ตัวแม่) 9
รูปที่ 3 แสดงวงจรของส่วน RS-232 Board และ FSK MOD 10
รูปที่ 4 บล็อกไดอะแกรมของส่วนที่สองคือตัวลูก (Client) 11
รูปที่ 5 แสดงรูปตัววัดอุณหภูมิเบอร์ DS1820 12
รูปที่ 6 แสดงตัวรีเลย์ที่ใช้งาน (5 VDC) 12
รูปที่ 7 สแดงโฟร์ชาร์ท การทำงานภาพรวมของส่วนโปรแกรมหลัก GUI บน PC 13
รูปที่ 8 แสดงหน้าจอแบบ GUI ที่ใช้รับข้อมูลคำสั่งจากผู้ใช้ 14
รูปที่ 9 แสดงโฟร์ชาร์ท การทำงานภาพรวมของโปรแกรมบนตัวลูก 16
รูปที่ 10 รูปแบบของ Command Format 17
รูปที่ 11 รูปแบบของ Data Format 18
รูปที่ 12 เส้นทางการวิ่งไปอ่านข้อมูลของ Client ตัวหมายเลข 4 19
รูปที่ 13 แสดงภาพรวมของการใช้งาน 20
รูปที่ 14 แสดงภายในตัวเครื่องต้นแบบ ตัวแม่ (main) 21
รูปที่ 15 แสดงภายในตัวเครื่องต้นแบบ ตัวลูก (Client) 21

บทที่ 1
บทนำ

ดังที่ได้กล่าวมาแล้ว ในบทคัดย่อ ว่าปัจจุบันนี้ การใช้งานคอมพิวเตอร์มีใช้อย่างกว้างขวาง และโดยเฉพาะการใช้งานโปรแกรมใดๆ หากทำให้ผู้ใช้ได้ใช้งานในระบบ GUI (Graphic User Interface) แล้ว ก็จะยิ่งทำให้บุคคลทั่วไปสามารถใช้งานได้อย่างง่ายดาย ในงานวิจัยนี้ได้ สร้างเครื่องตรวจวัด (และควบคุม) อุณหภูมิ ระยะไกลโดยอาศัยการส่งผ่านคำสั่งหรือผลลัพท์ผ่าน ระบบ วิทยุสื่อสารทั่วไป ซึ่งก็ได้จัดทำโปรแกรม ในงานวิจัยนี้ได้กำหนดคุณสมบัติพื้นฐานที่สำคัญดังนี้

คุณสมบัติมาตฐานโดยรวมเป็นดังนี้

- ใช้การสั่งงานผ่าน GUI บน คอมพิวเตอร์ทั่วไปที่ต่อกับเครื่องแม่
- ตัวลูกมี ตัวตรวจวัดอุณหภูมิจำนวน หนึ่งชุด
- ตัวลูกมี อุปกรณ์ตัดต่อไป (รีเลย์) จำนวนสองชุด
- ตัวรับส่งวิทยุใช้เป็นวิทยุสื่อสารทั่วไปที่สามารถปรับความถี่ได้
- ตัวลูกสามารถกำหนดตัวเองเป็นตัวที่หนึ่งหรือสองได้

ทั้งนี้ ทั้งหมดที่ได้กล่าวข้างต้นเป็นคุณสมบัติมาตรฐานที่จะต้องมีในเบื้องต้น ซึ่งจริงๆ แล้วในส่วนของ ตัวเซนเซอร์ที่จะต้องใช้งานนั้นจะถูกกำหนดโดยพื้นที่ๆ เรานำไปใช้งานอยู่แล้ว เช่น ในพื้นที่แห้งแล้ง ก็ควรจะมีการวัดอุณหภูมิ ส่วนในพื้นที่มีการไหลของน้ำป่าเป็นประจำก็อาจต้องมีการติดตั้ง ตัวเซนเซอร์วัดระดับน้ำ ไว้ด้วย เหล่านี้เป็นต้น ดังนั้นในส่วนของโครงงานวิจัยเบื้องต้นนี้จึงใช้ เซนเซอร์แบบอุณหภูมิเป็นหลัก หากต้องการเพิ่มชนิดของตัวเซนเซอร์ก็สามารถทำได้อย่าง อิสระและง่ายดาย เพราะจะมีการแก้ไขส่วนของฮาร์ดแวร์น้อยมาก (เพราะช่องรับสัญญาณมีอยู่มากในตัวไมโครคอนโทรลเลอร์อยู่แล้ว) ส่วนของซอฟร์แวร์ก็ได้ทำให้ยืดยุ่นปรับได้อย่า ไม่ยากทำให้โครงงานนี้สามารถนำไปต่อยอดได้อย่างมีประสิทธิภาพ

บทที่ 2
เอกสารและงานวิจัยที่เกี่ยวข้อง/การทบทวนวรรณกรรม

ในการดำเนินงานวิจัยนี้ ก็ได้มีการค้นคว้าหาข้อมูลที่เกียวข้องกับงานที่ทำทั้งในเว็ปไซต์ [3], [4] และสิ่งพิมพ์ต่างๆ แต่ยังไม่พบอุปกรณ์ฮาร์ดแวร์ที่ตรงเป้าหมายนัก ดังแสดงตัวอย่างบางชิ้นงานที่มีการทำจำหน่ายแล้ว ที่มีหน้าที่ใกล้เคียงกับงานวิจัย แสดงในรูปข้างล่างนี้

(a) (b)

รูปที่ 1 (a) แสดงเครื่องที่มีจำหน่ายแต่จะใช้ในการเปิดปิดอุปกรณ์ไฟฟ้าเท่านั้น
(b) แสดงเครื่องควบคุมการกระจายเสียงผ่านระบบสื่อสารวิทยุ

ดังนั้น ในงานวิจัยนี้จึงได้ออกแบบและสร้างระบบขึ้นมาเอง โดยให้ได้ดังจุดประสงค์ดังที่ได้กล่าวมาในบทนำ โดยในส่วนของตัวเครื่องวัดตัวลูก (Client) ได้เลือกอุปกรณ์ที่มีขนาดเล็ก เช่น เลือกหน่วยประมวลผล (CPU) ขนาดเล็กคือ MCS-51 ซึ่งหาได้ง่ายและราคาถูกและมีประสิทธิภาพที่พอเพียงต่อความต้องการ, ส่วนของการเปลี่ยนสัญญาณดิจิตอลเป็นเสียง (FSK MODULATOR) ก็เลือกใช้ ชิฟไอซี ที่มีจำหน่ายในเมืองไทย นำมาประกอบกับส่วนอื่นๆ อีกเล็กน้อย เช่น ส่วนติดต่อกับ PC (RS-232 Board), แบตเตอรี่ เหล่านี้เป็นต้น ดังรายละเอียดการสร้างและการทดลองและผลการทดลองต่างๆ ก็จะได้ นำเสนอในส่วนของ บทท้ายๆ นี้ต่อไป

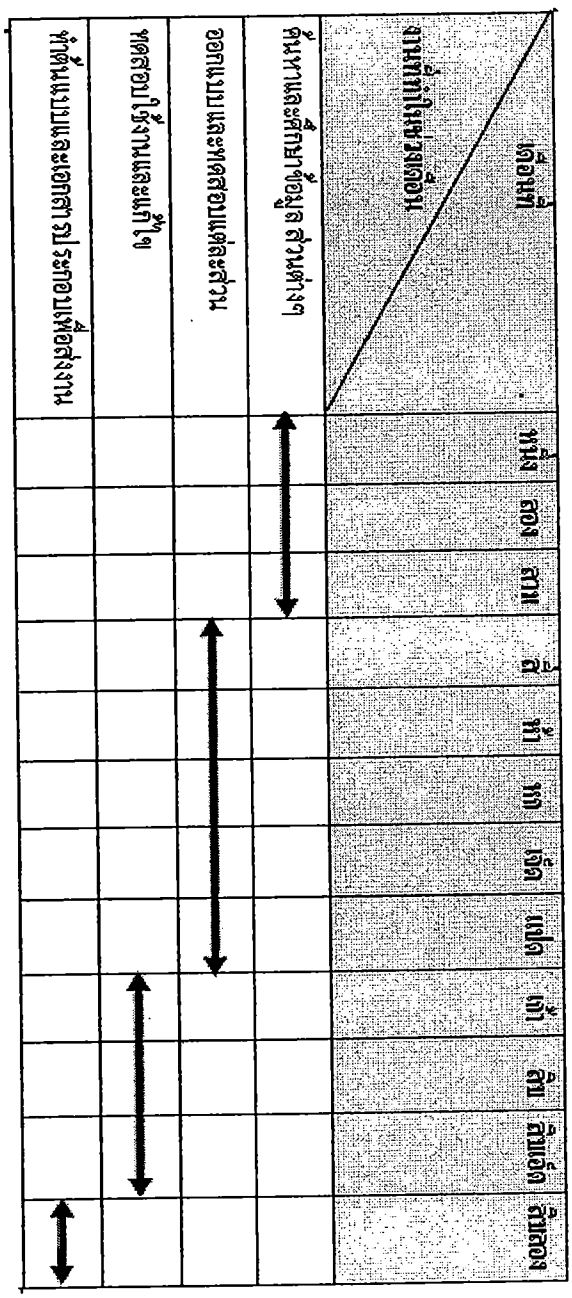
บทที่ 3

วิธีดำเนินการวิจัยและผลการวิจัย

3.1 แผนงานระยะต่างๆของการดำเนินงานวิจัย

ระยะเวลาวิจัยรวม หนึ่งปี ดังแสดงตารางช่วงเวลาการทำงาน ในแต่ละส่วน ในตารางที่ 1

หมายเหตุ เดือนที่หนึ่ง หมายถึง เดือนที่นับจากเดือนที่ได้รับอนุมัติโครงงานวิจัย และเดือนที่สิบสอง หมายถึง เดือนสุดท้ายของการทำโครงงานวิจัย

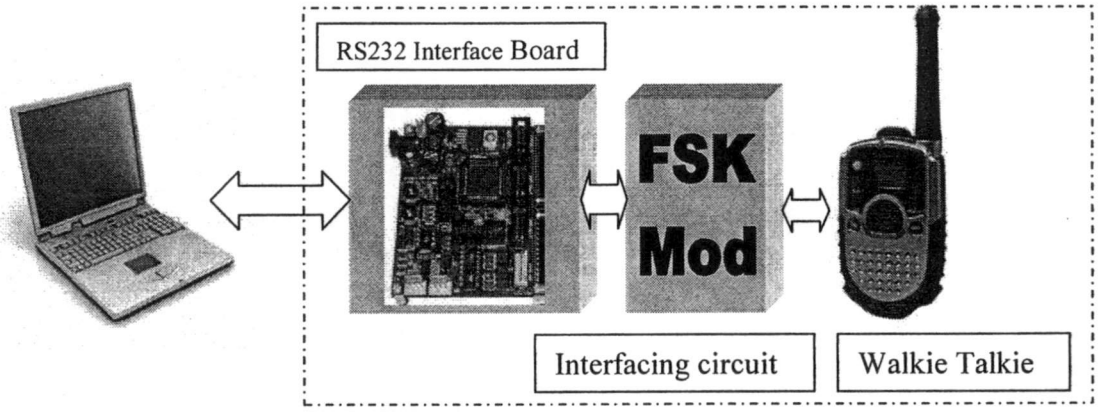


ตารางที่ 1 แสดงช่วงเวลาการทำงาน

งานวิจัยนี้ได้แบ่งออกเป็นสองส่วน คือส่วนของ ฮาร์ดแวร์และซอฟร์แวร์ ดังนี้

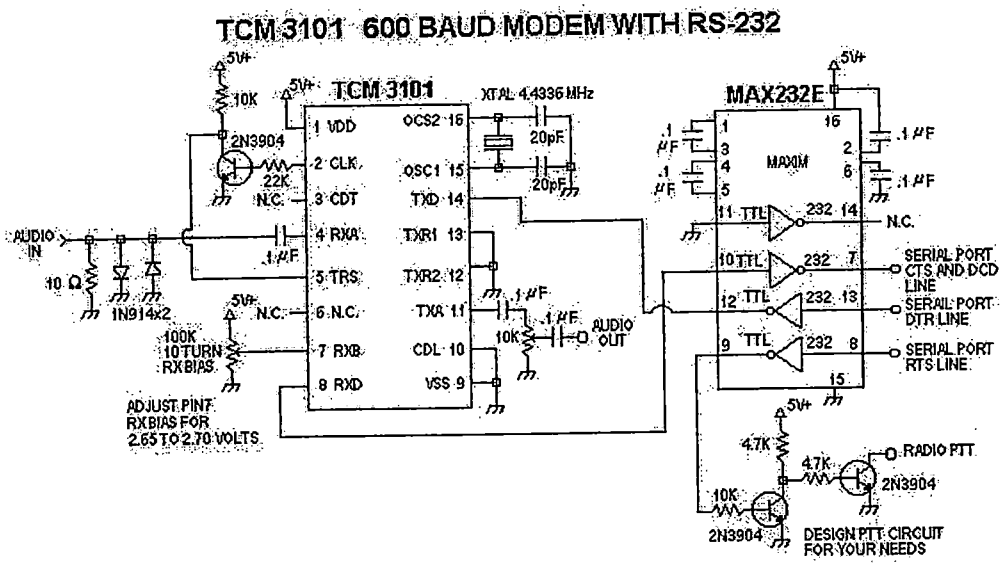
3.2 การสร้างส่วนของฮาร์ดแวร์

การสร้างจะแบ่งออกเป็นสองส่วน คือ ส่วนแรกเป็นส่วนของตัวแม่ หรือตัวหลักที่ใช้ติดต่อกับเครื่องคอมพิวเตอร์ของผู้ใช้เพื่อให้ผู้ใช้สั่งงานผ่าน ซอฟร์แวร์ ที่ออกแบบไว้ โดยในส่วนแรกนี้สามารถแสดงได้ดังรูป บล็อกไดอะแกรม ดังรูปที่ 2



รูปที่ 2 บล็อกไดอะแกรมของส่วนแรก (ตัวแม่)

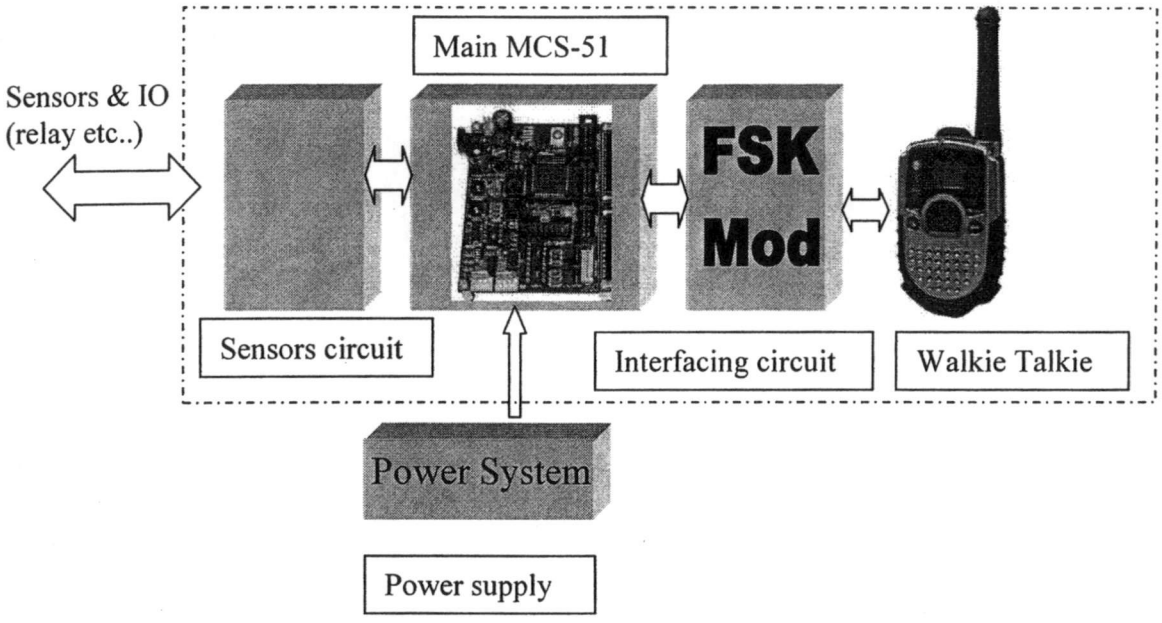
การทำงานก็คือ ผู้ใช้จะสั่งงานโดยใช้งานซอฟร์แวร์บนคอมพิวเตอร์จากนั้นคำสั่งก็จะถูกส่งผ่านสาย RS-232 เข้าสู่ RS-232 Board เพื่อนำคำสั่งไปเปลี่ยนเป็นสัญญาณเสียงโดยบอร์ด FSK MODULATOR ส่งต่อผ่านเครื่องวิทยุสื่อสารออกไป ตามลำดับ โดยวงจรของทั้งสองส่วนนั้น สามารถแสดงได้ดังรูปที่ 3 ซึ่งไปออกแบบกะทัดรัดน้อยชิ้นอุปกรณ์แต่ทำงานได้เป็นอย่างดี (จากการทดลอง) โดยเฉพาะในส่วนของ FSK MOD นั้นได้นำไอซีสำเร็จรูปเบอร์ TCM3101 ที่มีวงจรส่วนใหญ่อยู่ในตัวเรียบร้อยแล้วเพียงต่ออุปกรณ์เพิ่มเติมเล็กน้อยเท่านั้น และบอร์ดเรดที่สามารถใช้ได้ก็สูงถึง 1200 bps ซึ่งก็เพียงพอต่อการใช้งานเพราะคำสั่งที่ใช้รับส่งกันจะมีขนาดไม่ยาวมากอยู่แล้ว ส่วนในวงจร RS-232 Board นั้นนอกจากจะใช้ขาของ RX,TX ในการรับส่งปกติแล้วยังมีการนำขา RTS มาใช้งานในการเปิดหรือปิดส่วน PTT (Push To Talk) ด้วยทำให้ได้ขนาดของบอร์ดที่เล็กลง(ไม่ต้องใช้งานไมโครคอนโทรลเลอร์ในส่วนนี้) ทั้งหมดนี้สามารถแพ็คลงบนกล่องที่มีขนาดเล็กได้ (แยกส่วนของแหล่งจ่ายไฟไว้นอกกล่อง)



รูปที่ 3 แสดงวงจรของส่วน RS-232 Board และ FSK MOD

ในวงจรส่วนทำงานหลักก็จะเป็น ไอซี สำเร็จรูปเบอร์ TCM3101 ซึ่งจะมีวงจรทำสัญญาณ แบบ FSK ไว้แล้ว แต่ต้องมีการจัดวงจรที่เหมาะสมโดยตัวไอซีก็จะรับสัญญาณดิจิตอล จากไอซีแปลงสัญญาณ ดิจิตอลแบบอนุกรม (มาตรฐาน RS-232C) เบอร์ MAX232 เข้ามาเปลี่ยนเป็นสัญญาณเสียง เพื่อป้อนออกไปสู่ส่วนวิทยุรับส่งอีกต่อหนึ่ง โดยตัวของคอมพิวเตอร์ก็จะใช้ช่องทางการสื่อสารแบบอนุกรม (RS-232C) นี้ในการติดต่อกับ ตัวลูกตัวอื่นๆ เหมือนกัน ในส่วนของตัวลูกทุกตัวก็จะมีลักษณะวงจร FSK นี้เหมือนกัน ดังนั้นเมื่อทุกตัวตัวลูกรับสัญญาณจากตัวแม่ได้พร้อมกันหมด การติดต่อสื่อสารกันอย่างมีประสิทธิภาพ จึงมีต้องใช้ซอร์ฟแวร์ในการแยกแยะว่า เป็นการติดต่อกับตัวใด ซึ่งอันนี้ก็จะได้มีการอธิบายหลักการรับส่งในส่วนหัวข้อต่อไป

ส่วนต่อมาเป็นส่วนที่สองคือส่วนลูกดังแสดงในรูปข้างล่างในรูปที่ 4 ซึ่งในส่วนนี้นั้นจะมีการเพิ่มที่สำคัญก็คือ ไมโครคอนโทรลเลอร์ (MCS-51) เข้ามาทำหน้าที่ในการควบคุมและตัดสินใจทุกอย่าง เมื่อมีการับคำสั่งจากตัวแม่เข้ามา ซึ่งจะได้แสดงเป็นโฟร์ชาร์ทการทำงานหัวข้อต่อไป



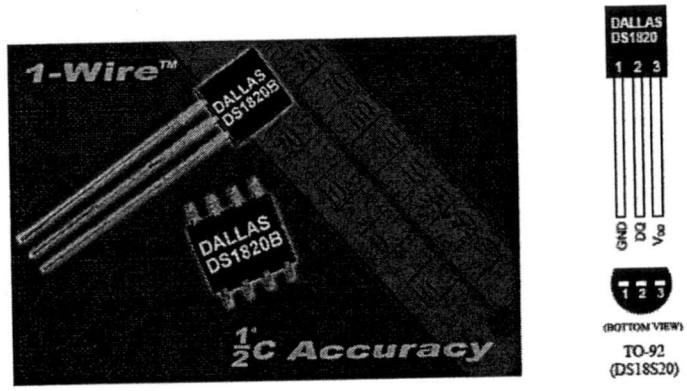
รูปที่ 4 บล็อกไดอะแกรมของส่วนที่สองคือตัวลูก (Client)

ในส่วนนี้จะมีลักษณะทั่วๆไปเหมือนกันกับแบบของตัวแม่ (main) แต่จะมีส่านของไมโครคอนโทรลเลอร์ที่ทำหน้าที่หลายอย่างคือ รับคำสั่งจากสัญญาณวิทยุที่ส่งมาจากตัวแม่ เพื่อการแยกแยะคำสั่ง (จะมีการอธิบายเป็นโฟร์ชาร์จในหัวข้อของเรื่อง ซอฟร์แวร์) จากนั้นก็ไปปฏิบัติตามคำสั่งนั้นๆ เช่นการวัดค่าของอุณภูมิ, การสั่งให้รีเลย์ตัวนั้นๆ ได้ทำงาน เปิดหรือปิด หน้าคอนแทดตัวเองเหล่านี้เป็นต้น ที่สำคัญอันหนึ่งคือส่วนนี้ หากนำไปใช้งานจริงจะต้องมีการ สร้างเพิ่มเติมในส่วนของ ตัวจ่ายกำลังไฟฟ้ากระแสตรง (แต่การวิจัยเบื้องต้นนี้ใช้แรงไฟตรงจากการแปลง จากไฟฟ้าเอซีทั่วไป) เพราะจะต้องถูกนำไปติดตั้งใน พื้นที่ๆไม่มีไฟฟ้าเอซี ที่ใช้ตามบ้านทั่วไป นั่นหมายความว่าในส่วนของตัวลูก นี้จะต้องมีการติดตั้ง ส่วนของการรับแสงอาทิตย์แปลงมาเป็นพลังงานที่สามารถจัดเก็บได้ในแบตเตอรี่ (ระบบโซล่าเซล) ที่มีกำลังพอใช้งานกับชนิดของเครื่องรับส่งที่ติดตั้งบนตัวลูกนั่นๆ นั่นเอง.

3.2.1 ส่วนของเซนเซอร์อุณหภูมิ

ในส่วนนี้ในงานวิจัยใช้ ตัววัดอุณหภูมิที่ใช้งานง่ายและอุปกรณ์เสริมน้อยชิ้นและที่สำคัญมีคุณสมบัติที่สามารถ วัดอุณหภูมิได้ในย่านที่กว้าง คือ ตัววัดอุณหภูมิในตระกูลเบอร์ DS1820 ดังแสดงในภาพรูปที่ 5 ตัวนี้เราสามารถนำมาต่อเชื่อมกับพอร์ทของไมโครคอนโทรลเลอร์ได้โดยตรง เพราะมีการสื่อสารกันแบบอนุกรม ที่มีฟอร์แมทตายตัว

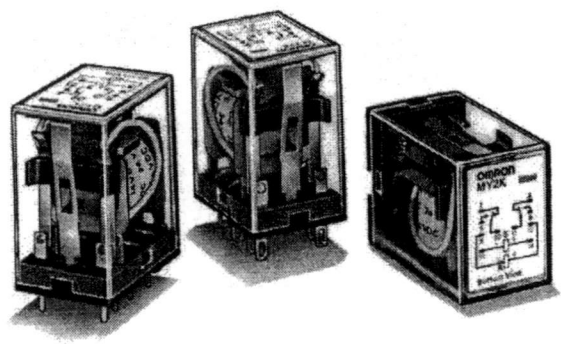
อยู่แล้ว ทางผู้วิจัยได้ออกแบบโปรแกรมเชื่อมต่อและทดลองจนได้ผลถูกต้องตามอุณหภูมิจริง นอกจากนี้ตัววัดเบอร์นี้ยังสามารถให้ผลการวัดที่รวดเร็วมากแรงไฟฟ้าที่ใช้งานก็น้อย นอกจากนี้ยังสามารถกำหนดความละเอียดของการวัดได้หลากหลาย จึงเหมาะสมอย่างยิ่งกับงานวิจัยนี้



รูปที่ 5 แสดงรูปตัววัดอุณหภูมิเบอร์ DS1820

3.2.2 ส่วนของรีเลย์ตัดต่อไฟฟ้า

ในงานวจัยใช้วงจรที่เรียบง่ายทำงานด้วยแรงดันไฟฟ้าต่ำ (5 โวลท์ดีซี) ซึ่งในส่วนนี้สามารถปรับเปลี่ยนได้ตามความต้องการของระบบว่าต้องการใช้งานกับโหลดที่ต้องการกระแสมากน้อย ขนาดไหน



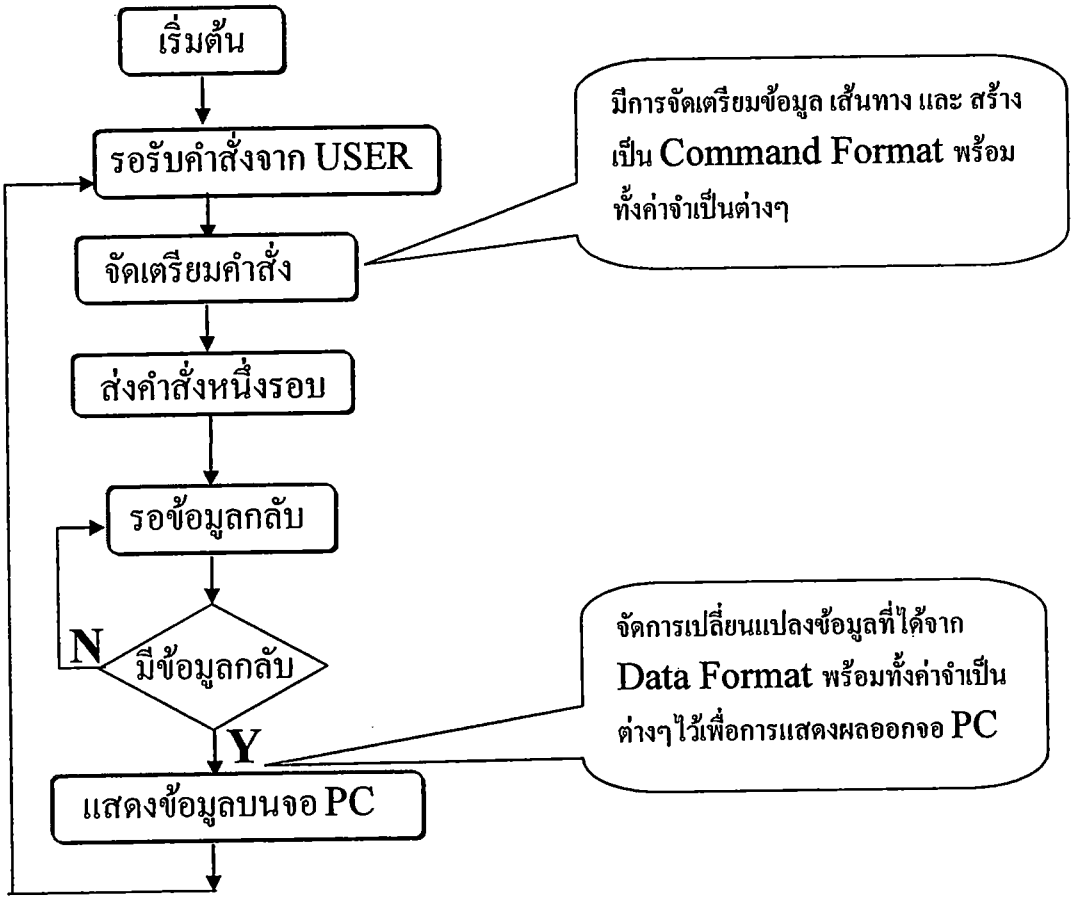
รูปที่ 6 แสดงตัวรีเลย์ที่ใช้งาน (5 VDC)

3.3 การสร้างส่วนของซอฟร์แวร์

ดังที่กล่าวมาแล้วในหัวข้อ ฮาร์ดแวร์ว่า มีการทำงานด้วยกันอยู่ส่วนส่วนคือ ส่วนควบคุมใช้งานหลักที่อยู่บนตัว PC เอง และส่วนของโปรแกรมควบคุมที่อยู่บนตัวลูก ดังนี้

3.3.1 ซอฟร์แวร์ส่วนของโปรแกรมหลักบน PC

ในส่วนนี้เป็นการเขียนโปรแกรมแบบ GUI เพื่อให้ทำงานบนเครื่อง PC ที่เป็นระบบ Windows ทั้งนี้เพื่อให้ง่ายต่อการใช้งานมากที่สุด โดยมีลักษณะของการทำงาน แสดงเป็น โฟร์ชาร์ท ได้ดังรูปที่ 7

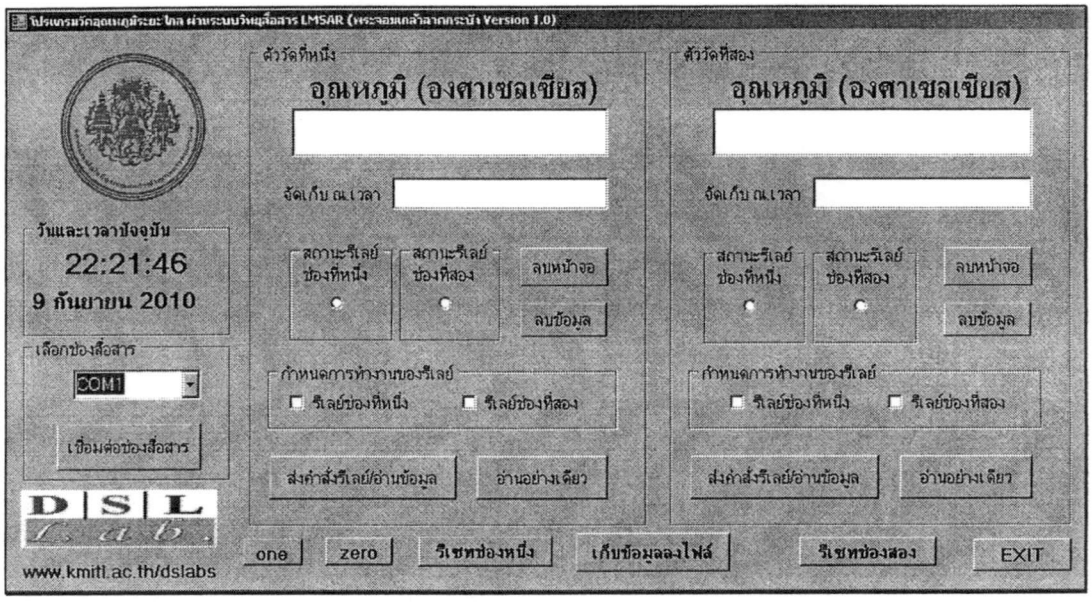


รูปที่ 7 สแดงโฟร์ชาร์ท การทำงานภาพรวมของส่วนโปรแกรมหลัก GUI บน PC

เริ่มจาก ตัวโปรแกรมจะแสดงภาพส่วน GUI บนจอมอนิเตอร์ เพื่อรับคำสั่งต่างๆจากผู้ใช้ เช่น ต้องการอ่านค่าอุณหภูมิ หรือเลือกการสั่งรีเลย์ตัวที่ต้องการ ON หรือ OFF เหล่านี้เป็นต้น เมื่อ

ตัวโปรแกรมรับคำสั่งที่ต้องการให้ทำแล้ว ก็จะจัดเรียงรูปแบบคำสั่งแล้วส่งออกผ่านสัญญาณวิทยุต่อไป จากนั้นตัวโปรแกรมก็จะทำการรอผลจากตัวลูกที่ส่งคำสั่งไปอ่านค่าต่างๆกลับมา เพื่อแสดงผลออกบนจอภาพมอนิเตอร์ต่อไป

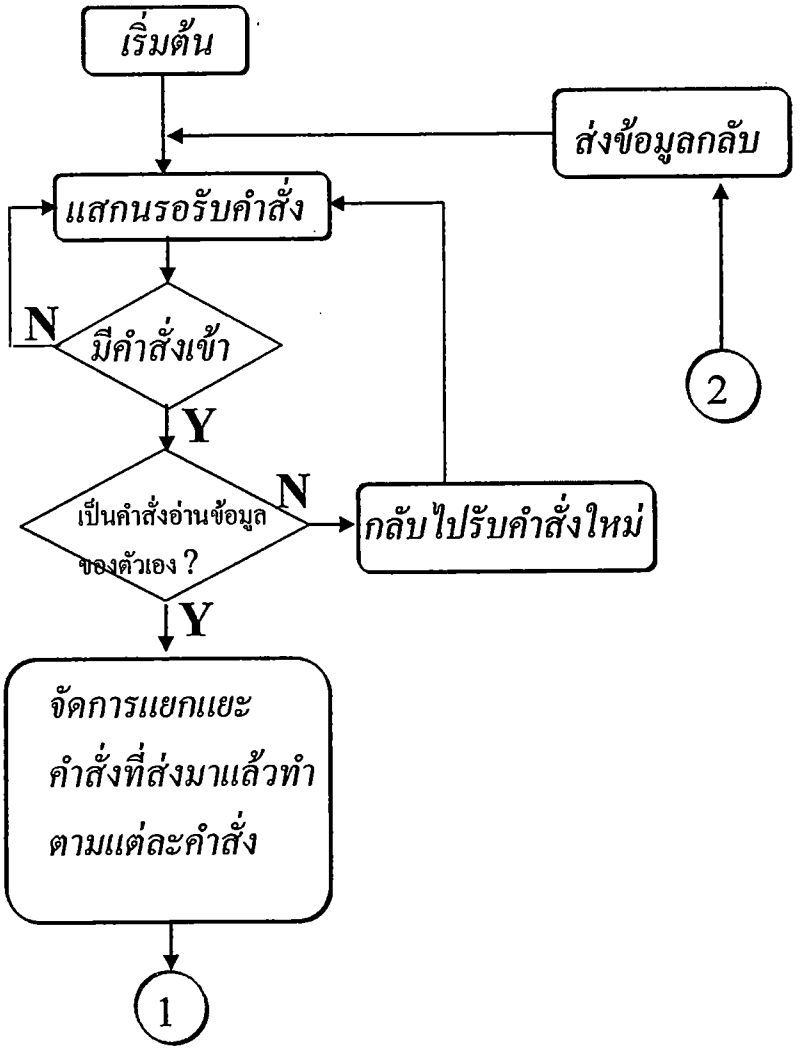
ในตอนรับข้อมูลกลับมาแล้วนั้นตัวโปรแกรมก็จะทำการแยกแยะส่วนของข้อมูลอุณหภูมิและส่วนของสถานะรีเลย์ที่ได้ทำการส่งคำสั่งอ่านค่าไป เพื่อจะได้ออกแสดงได้อย่างถูกต้อง

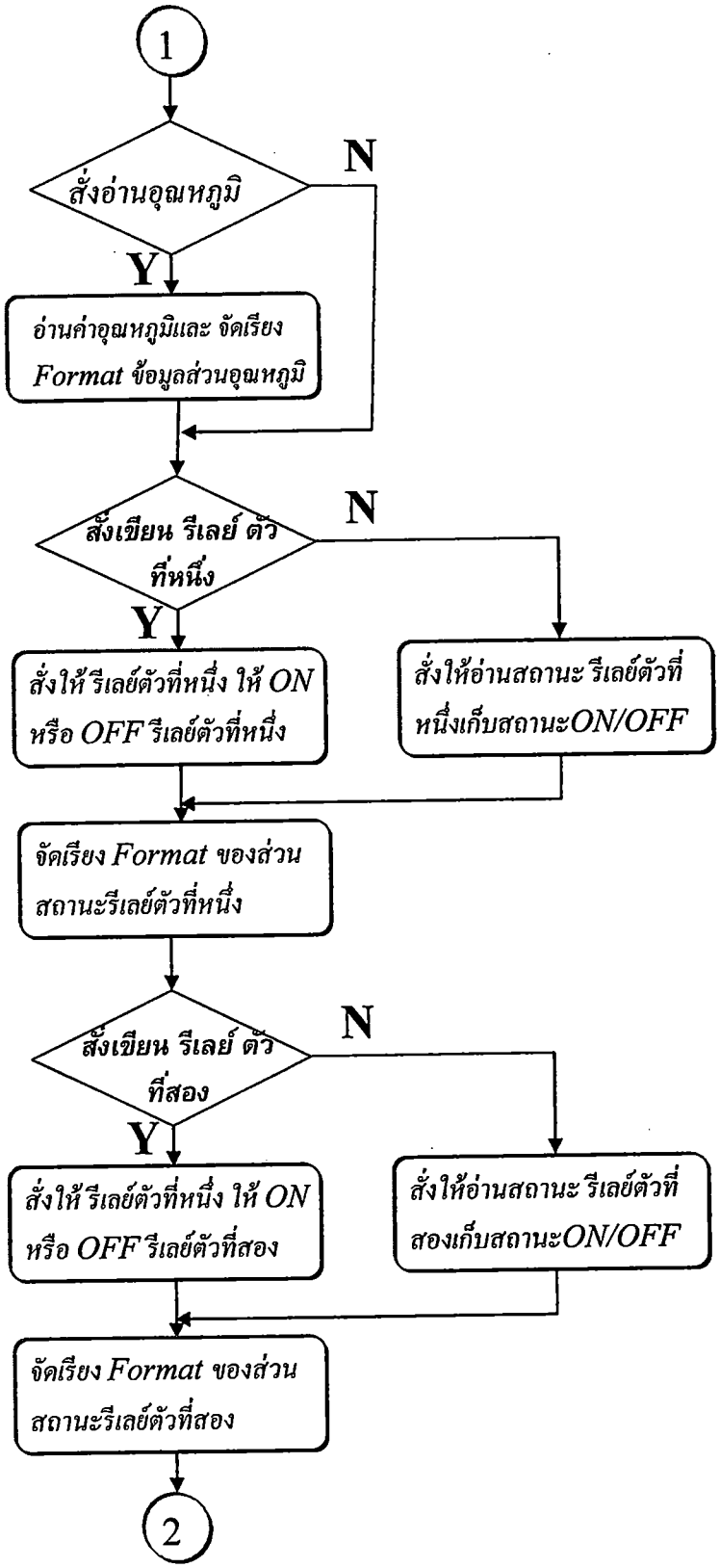


รูปที่ 8 แสดงหน้าจอแบบ GUI ที่ใช้รับข้อมูลคำสั่งจากผู้ใช้

3.3.2 ซอฟร์แวร์ส่วนของตัวลูก

ในส่วนของซอฟร์แวร์ของตัวลูกนี้ จะรับคำสั่งที่ได้จากตัวแม่ (ผ่านระบบวิทยุ) จากนั้นก็แยกแยะว่าเป็นการติดต่อกับตัวเองหรือไม่ หากใช่ก็จะแยกแยะคำสั่งต่อไปอีกว่า มีการสั่งให้อ่านค่าอุณหภูมิหรือสั่งให้ รีเลย์ตัวที่หนึ่งหรือสองทำงานในลักษณะใด (เป็นสั่งให้รีเลย์ ON หรือ OFF หรือจะเป็นการอ่านสถานะของรีเลย์ว่าอยู่ในสถานะใด)





รูปที่ 9 แสดงโฟร์ชาร์ท การทำงานภาพรวมของโปรแกรมบนตัวลูก

สำนักหอสมุดกลาง พระจอมเกล้าลาดกระบัง

โดยจะมีการทำงานที่สำคัญดังนี้ คือรับคำสั่งเข้ามา จากนั้นก็แยกแยะดูว่าเป็นคำสั่งที่ต้องการอ่านหรือเขียน (อ่านคืออ่านค่าอุณหภูมิ, เขียนคือสั่งให้รีเลย์ทำงาน) ตัวเองหรือไม่ หากไม่ใช่ก็จะไม่ทำอะไรแต่จะรอข้อมูลใหม่ไปเรื่อยๆ ส่วนหากเป็นการอ่านเขียน ตัวเอง ก็จะทำการอ่านค่าของอุณหภูมิและสั่งรีเลย์ ตามคำสั่งนั้นๆ และนำผลที่ได้มาจัดรูปแบบข้อมูล (Data format) ส่งกลับไปสู่ตัวแม่ ต่อไป

3.4 ส่วนของฟอร์แมทคำสั่งในการรับส่งข้อมูล

3.4.1 Command Format (ส่งออกไปเพื่ออ่านข้อมูลและสั่งงานรีเลย์)

เป็นฟอร์แมท (Format) ที่ใช้ในการส่งจากตัว Main ไปสู่ตัว Clients ที่เป็นตัวที่ถูกระบุให้อ่านข้อมูล เพื่อให้ส่งค่าอุณหภูมิกลับต่อไป

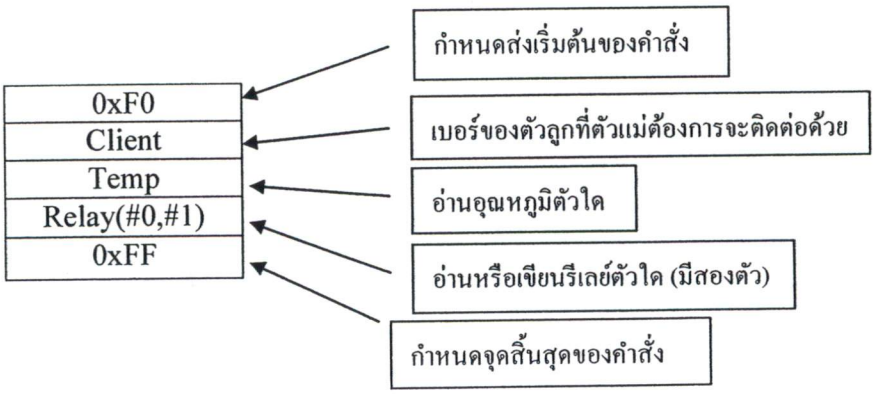
ฟอร์แมทคำสั่ง (Command Format) จะประกอบไปด้วย ไบท์ที่เรียงกันไปเป็นชั้นๆ ซึ่งแต่ละชั้นจะมีความหมายของตัวเอง โดยในงานวิจัยนี้จะมีขนาด 5 ไบท์ ดังรูปที่ 8 ดังนี้

Client ไบท์นี้จะใช้เก็บค่าที่บอกถึงหมายเลขของ Clients ตัวเป้าหมายที่จะรับคำสั่งไปเพื่ออ่านข้อมูลจากเซนเซอร์ตนเองเพื่อส่งกลับ

Temp ไบท์นี้ในแต่ละบิตจะเป็นตัวชี้บอกว่าต้องการอ่านข้อมูลจากเซนเซอร์ตัววัดอุณหภูมิตัวไหนบ้าง (ในบางครั้งตัวลูกจะมีการติดตั้งตัววัดอุณหภูมิได้หลายตัว)

Relay (#0, #1) ไบท์นี้ในแต่ละบิตจะเป็นตัวบ่งชี้ว่าจะมีการกำหนดให้รีเลย์ตัวไหน On หรือ Off บ้าง ในงานวิจัยนี้มีการติดตั้งรวม 2 ตัว (โดย On คือ สถานะบิตนั้นเป็น "1" ส่วน Off ก็จะมีสถานะบิตเป็น "0")

อีกสองไบท์ที่ปิดหัวท้ายคำสั่งคือ 0xF0 และ 0xFF มีไว้เพื่อกำหนดขอบเขตของคำสั่งเพื่อให้การรับส่งคำสั่งมีการตรวจสอบที่ปลอดภัยมากขึ้น



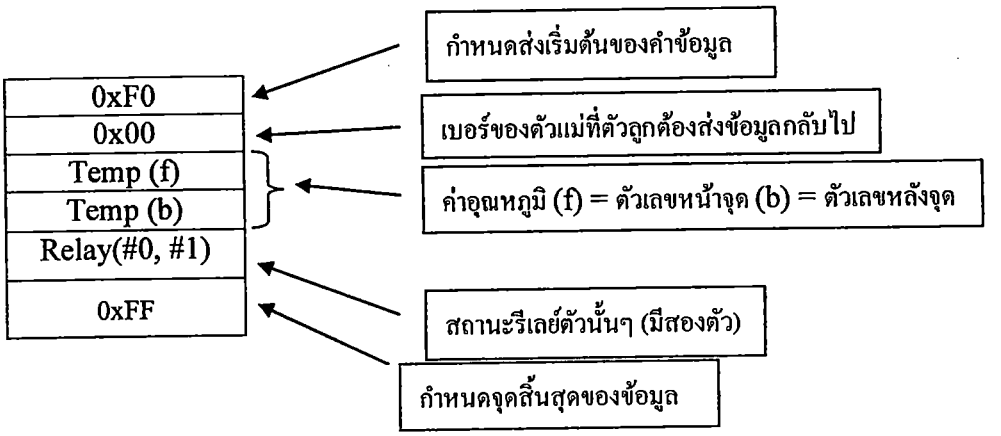
รูปที่ 10 รูปแบบของ Command Format

116099

ในการส่ง Command Format นี้ไป จะเริ่มจากผู้ใช้สั่งงานผ่าน GUI ซึ่งจะเป็นคำสั่งทั้ง ห้าไบท์นี้จะถูกส่งจากเครื่อง PC ไปสู่เครื่อง Main จากนั้นคำสั่งเดียวกันนี้ก็จะถูกส่งผ่านคลื่นวิทยุ กระจายต่อไปสู่ทุกตัวที่อยู่ในขอบเขตกระจายคลื่นวิทยุ และตัวของเครื่อง Main ก็จะอยู่ในสถานะ รอรับสัญญาณวิทยุของข้อมูลที่จะส่งกลับมา ในส่วนของข้อมูลที่จะส่งกลับมาจาก Client ตัว เป้าหมายนั้นให้ชื่อว่า Data Format ก็จะมีรูปแบบ Format ที่ใช้กำหนดใช้งาน ดังหัวข้อต่อไปนี้

3.4.2 Data Format (ส่งกลับข้อมูลกลับไปสู่คอมพิวเตอร์)

เป็นชุดของรูปแบบข้อมูลที่จะเก็บค่าต่างๆ ตามคำสั่งของ Command Formant ที่ส่งมา แล้วส่งกลับไปสู่ตัว Main โดยจะมีรูปแบบดังแสดงในรูปที่ 9



รูปที่ 11 รูปแบบของ Data Format

มีลักษณะคล้ายกันกับคำสั่งที่ส่งจากตัวแม่ไปสู่ตัวลูก โดยไบท์เริ่มต้นก็จะเป็น 0xF0 เป็นการ กำหนดการเริ่มต้นของข้อมูล และปิดท้ายข้อมูลด้วย 0xFF ส่วนข้อมูลอื่นเป็นดังนี้

0x00 คือ ไบท์แสดงเบอร์ตัวแม่ที่ตัวลูกนั้นจะทำการติดต่อส่งข้อมูลกลับด้วย

Temp (f) เป็นข้อมูลแสดงถึงค่าของข้อมูลของแต่ละตัว โดยเป็นค่าที่อยู่หน้าจุดทศนิยม (มีขนาด 8 บิต)

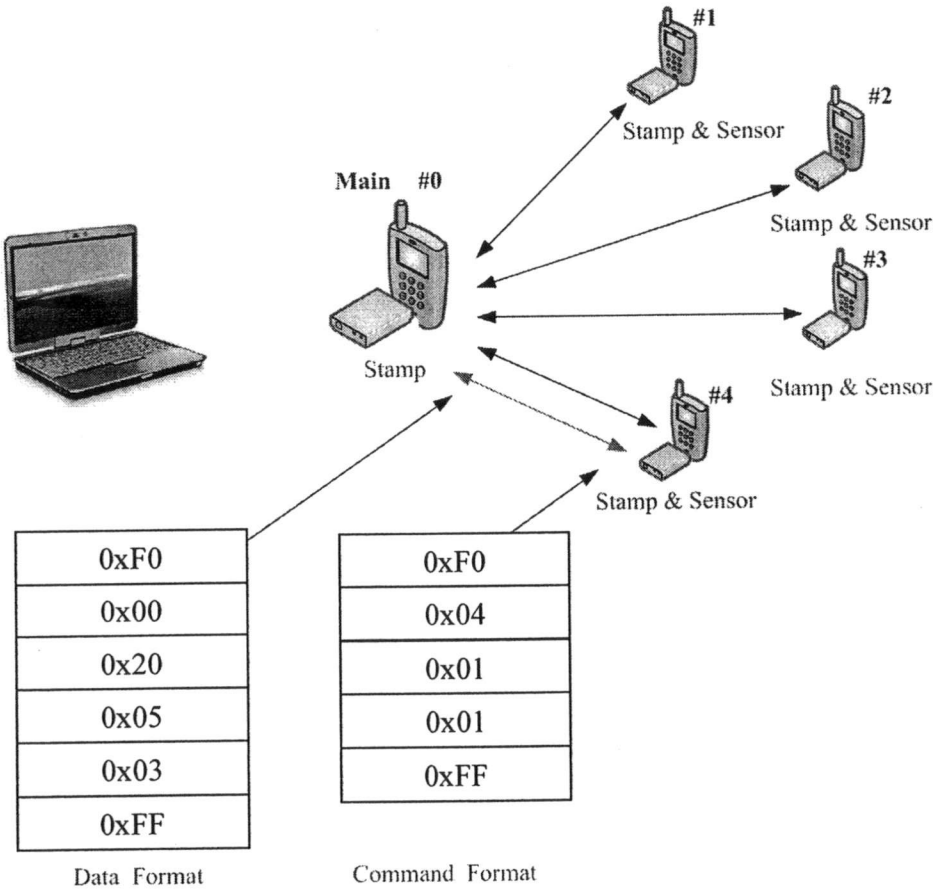
Temp (b) เป็นข้อมูลแสดงถึงค่าของข้อมูลของแต่ละตัว โดยเป็นค่าที่อยู่หลังจุด ทศนิยม (มีขนาด 8 บิต)

Relay (#0, #1) คือ ไบท์แสดงถึงสถานะของรีเลย์แต่ล่ะตัวว่าอยู่ในสถานะ On หรือ Off (โดย On จะแสดงบิตนั้นเป็น "1" และ Off จะแสดงบิตนั้นเป็น "0")

End (0xFF) เป็นไบท์แสดงถึงการสิ้นสุดของ Data Format

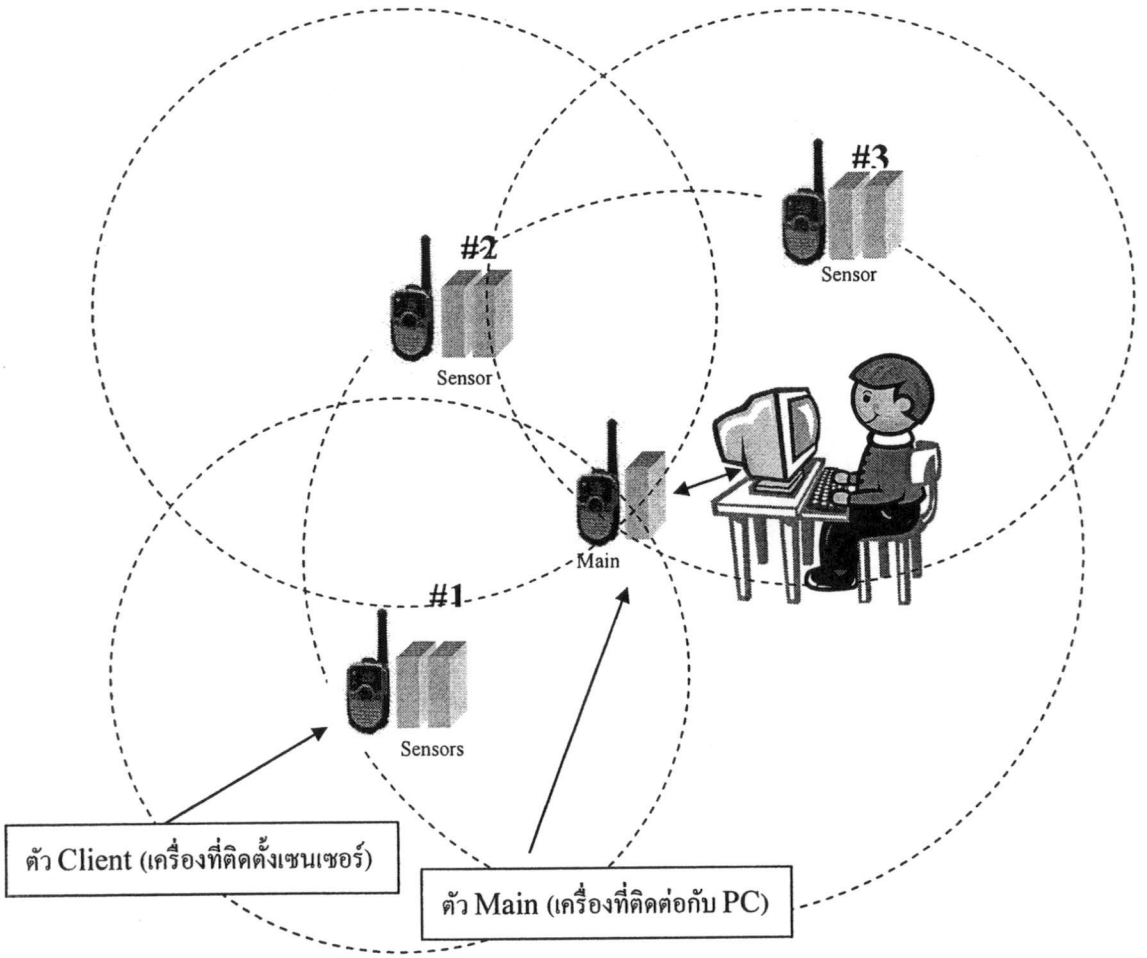
ในรูปที่ 12 แสดงเส้นทางการวิ่งไปของ Command Format เพื่อไปอ่านข้อมูลส่วนตัว Client ตัวเป้าหมาย จากตัวอย่างกำหนดให้ตัว Client หมายเลขสี่ มีการวัดอุณหภูมิเพียงตัวเดียวและมี Relay ที่ใช้งานอยู่สองตัว (กำหนดให้เป็น บิทที่ 0, 1 ตามลำดับ) เมื่อเป็นดังนี้เราก็จะได้ค่าของ Data Format ที่จะต้องส่งกลับจึงสามารถทำได้ดังรูป (ตัวอย่าง)

ไบท์ ที่ 0 => 0xF0 : เริ่มคำสั่ง
ไบท์ ที่ 1 => 0x04 : ติดต่อไปที่ตัวที่สี่
ไบท์ ที่ 2 => 0x01 : อ่านอุณหภูมิตัวที่หนึ่ง (ตอนนี้มีตัวเดียว)
ไบท์ ที่ 3 => 0x01 : บิทที่ 0 = "1", บิทที่ 1 = "0" (0 = OFF, 1 = ON)
ไบท์ ที่ 4 => 0xFF : หมดคำสั่ง



รูปที่ 12 เส้นทางการวิ่งไปอ่านข้อมูลของ Client ตัวหมายเลข 4
(stamp & sensor หมายถึง MCU ขนาดแสตมป์และมีการต่อเซนเซอร์ด้วย)

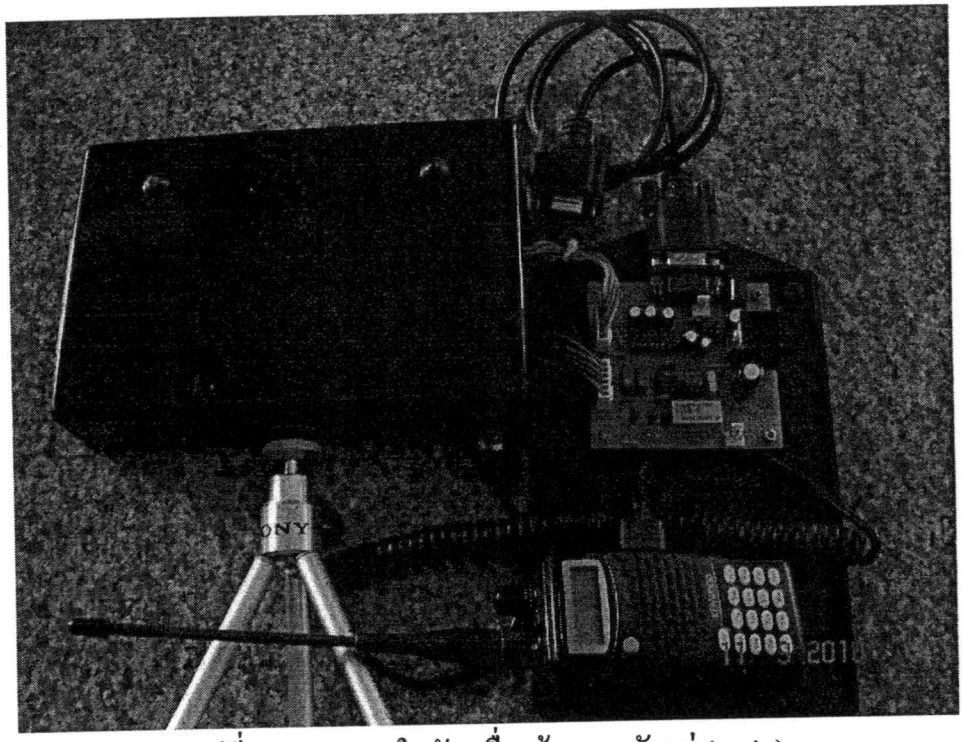
3.5 ภาพรวมการใช้งาน



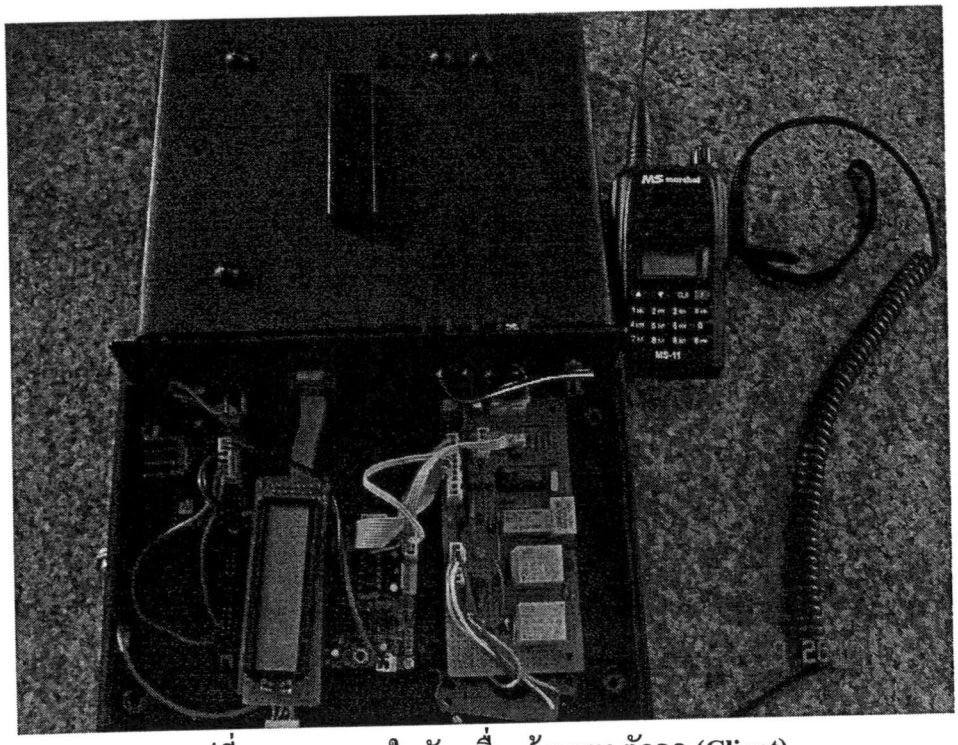
รูปที่ 13 แสดงภาพรวมของการใช้งาน

ในภาพรวมของการใช้งานนั้น แสดงได้ดังรูปที่ 13 ผู้ใช้จะสั่งควบคุมการอ่านเรื่องกำหนดค่าของรีเลย์ได้โดย ทำบนโปรแกรม GUI บนคอมพิวเตอร์ ดังรูปที่ 6 ซึ่งตัวโปรแกรมก็จะเป็น GUI ที่ง่ายเพราะสามารถเข้าใจได้ เพราะแต่ละปุ่มก็แสดงคำภาษาไทยกำกับไว้ทุกปุ่ม เช่น การอ่านอุณหภูมิของตัวลูก ตัวที่2 (#2) ก็สามารถกดที่หน้าจอ ในช่องตัวลูกที่ต้องการติดต่อได้เลย จากนั้นก็นั่งรอผลการอ่านอุณหภูมิกลับมา ส่วนของตัวลูกก็จะถูกติดตั้งไว้ในที่ต่างๆรอบรัสมีที่สามารถรับหรือส่งข้อมูลกับตัวแม่ได้ ซึ่งจากการทดลองในหลายพื้นที่ก็ระยะห่างระหว่างตัวลูกกับตัวแม่ควรจะมีระยะสั้นกว่า ข้อกำหนดของเครื่องรับส่งประมาณ 10 % ก็จะทำให้ผลการรับส่ง

ข้อมูลไม่มีผิดพลาด (ผลการการทดลอง) ทั้งนี้เพราะในพื้นที่จริงจะประกอบไปด้วยสิ่งกีดขวางมากมาย ประกอบทั้งข้อกำหนดของเครื่องที่นำมาใช้จริงก็มีโอกาศที่จะด้อยลงได้



รูปที่ 14 แสดงภายในตัวเครื่องต้นแบบ ตัวแม่ (main)



รูปที่ 15 แสดงภายในตัวเครื่องต้นแบบ ตัวลูก (Client)

บทที่ 4
อภิปรายผลการวิจัยและวิจารณ์

จากผลการวิจัย ที่ผ่านมาในบที่ 3 นั้น จะเห็นได้ว่า ส่วนของตัวเครื่องนั้นมีปุ่มให้ปรับน้อยมาก คือตัวลูกก็จะมีเพียงปุ่มสำหรับเลือกว่าจะให้เป็นหมายเลขอะไร ก่อนการเปิดเครื่องใช้งาน นอกนั้นก็ไม่ต้องการปรับแต่งอะไรเลย (นอกจากกรณีต้องการเปลี่ยนแปลงความถี่)

จากการทดลองนำไปใช้งาน ได้มีการทดลองใช้งานในหลายพื้นที่ ทั้งที่มีสิ่งกีดขวางมาก และสถานที่โล่งเพื่อให้มีภาพที่ใช้งานเหมือนจริงมากที่สุด และได้ทดลองเป็นเวลานาน หลายวัน เพื่อดูความคงทน (Stable) หรือความผิดเพี้ยนใดๆ อันอาจเกิดขึ้นได้ ก็ได้ข้อสรุปที่น่าพอใจว่า สามารถทำงานได้ดี สามารถใช้งานได้ดีและสะดวกมาก ในระดับหนึ่ง หากแต่ถ้าสามารถปรับปรุงให้ตัวลูกมีขนาดของเครื่องที่เล็กลงไปอีกก็จะเป็นประโยชน์ได้ไม่น้อย เพราะว่าในการนำไปใช้งานจริงนั้น ระบบจ่ายไฟสำคัญมาก เพราะหากนำไปติดตั้งในพื้นที่ๆห่างไกล เช่น บนภูเขา ซึ่งที่นั้นไม่มีทั้งสัญญาณใดๆ และไฟฟ้าก็ไม่มีด้วย ทำให้ต้องมีการออกแบบระบบจ่ายไฟฟ้าที่มาจากพลังงานแสงอาทิตย์ใช้งานกับเครื่องลูกนี้ และหากมีการออกแบบระบบจ่ายไฟให้สามารถจ่ายไฟฟ้าได้อย่างพอเพียง ในกรณีที่ไม่มีแสดงอาทิตย์ยาวนานเป็น สองถึงสามวัน ก็จะทำให้ระบบนี้ทำงานได้ดียิ่งขึ้น อันนี้เป็นส่วนหนึ่งที่จะได้ทำการวิจัยต่อไป

บทที่ 5
สรุปและข้อเสนอแนะ

งานวิจัยนี้ เป็นการสร้างเครื่องตรวจวัด(อุณหภูมิ)ระยะไกล โดยผ่านระบบวิทยุสื่อสาร โดยเครื่องต้นแบบจะเป็นการอ่านค่าของอุณหภูมิจากตัวลูก โดยใช้ซอฟร์แวร์ที่กำหนดให้ทำงานบนเครื่อง PC ทั่วไป ตัวเครื่องยังสามารถสั่ง เปิดปิด (รีเลย์) ได้อีกสองชุด (หรือตรวจสอบสถานะของรีเลย์ก็ได้เช่นกัน) ด้วยคุณสมบัติที่กล่าวมานี้ ก็สามารถนำไปใช้งานได้จริง

อนึ่งเครื่องต้นแบบนี้ได้มีส่วนของ ฮาร์ดแวร์และ ซอฟร์แวร์ ที่สามารถปรับปรุงแก้ไขให้สามารถเพิ่มความสามารถได้อย่างง่ายดาย เช่น เพิ่มชนิดของเซนเซอร์เข้าไปอีก อย่างเช่น เซนต์เซอร์ จำพวก วัดความชื้น, วัดระดับน้ำ หรือแม้แต่ตัววัดความเร็วลม เหล่านี้เป็นต้น และเช่นกันส่วนของ ซอฟร์แวร์ก็ออกแบบให้สามารถ แก้ไขได้ง่ายเพื่อให้สอดคล้องกับฮาร์ดแวร์ที่เปลี่ยนแปลงไป ข้อสังเกตอันหนึ่งคือ นอกจากการเพิ่มความสามารถที่กล่าวมาแล้วนั้น ยังมีส่วนที่สำคัญที่น่าทำการวิจัยต่อไปคือ การทำให้เครื่องตัวลูก ทำงานเป็นตัวกลางรับส่งสัญญาณ (รีพีทเตอร์) ได้ด้วย ซึ่งหากเป็นเช่นนี้ได้ เราก็สามารถใช้เครื่องวิทยุรับส่งที่มีกำลังต่ำในทั้งตัวลูกและตัวแม่ได้ ซึ่งจะทำให้ประหยัดทั้งกำลังไฟฟ้าที่ใช้งานและราคาต่ำลง (เครื่องลูก) ได้ด้วย ทั้งนี้เพราะ หากตัวลูกสามารถเป็นเหมือนสถานีทวนสัญญาณดังกล่าวมา ก็จะทำให้ระบบสามารถออกแบบ ซอฟร์แวร์ ให้ทำงานแบบเคลือข่ายได้ เนื่องจากในการใช้งานจริงนั้น ตัวลูกที่เป็นตัววัดนั้นจะมี จำนวนหลายตัว กระจายไปในสถานที่ต่างๆ ทั้งไกล้และไกลจากตัวแม่ กำลังส่งของตัวแม่อาจไปไม่ถึงตัวลูกที่อยู่ไกลๆ ได้ หากสามารถส่งคำสั่งผ่าน ตัวลูกตัวใกล้อื่นๆไปสู่ตัวไกลที่ต้องการติดต่อได้ก็จะเป็นประโยชน์อย่างยิ่ง นั่นเอง.

บรรณานุกรม

[1] http://www.keil.com/

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

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

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

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

[6] Han-Way Huang , "Using the MCS-51 Microcontroller" , Dec 16, 1999.

[7] ประจิน พลังสันติกุล และ ชัยวัฒน์ ลิ้มพรจิตรวิไล, " ปฏิบัติการ ไมโครคอนโทรลเลอร์ MCS-51 กับ Keil C51 ", บริษัท อินโนเวชั่น เอ็กเพอริเม้นต์ จำกัด., พ.ศ. 2550

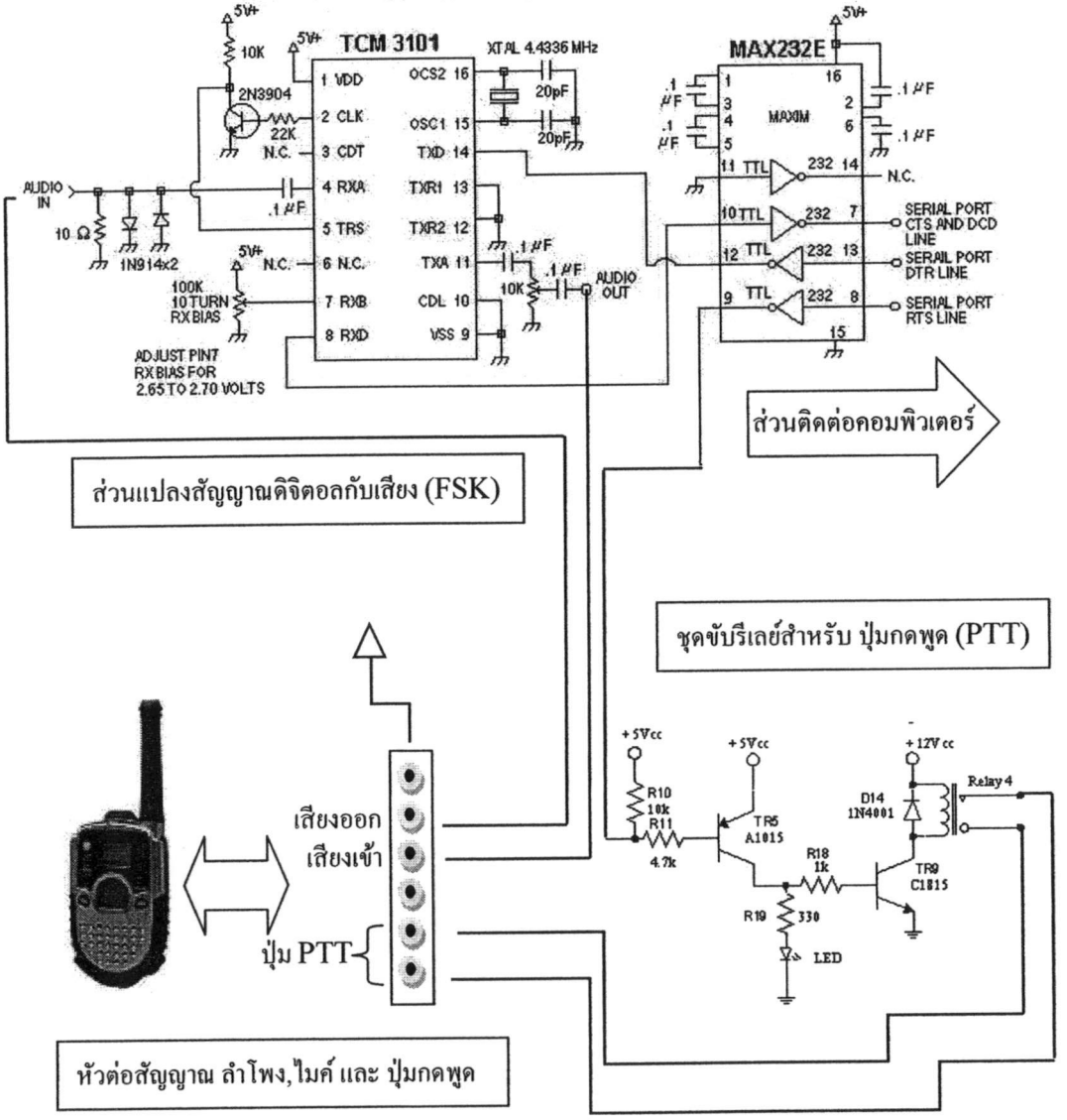
[8] สัจจะ จรัสรุ่งรวีวร , " คู่มือ Visual C# 2005 ฉบับสมบูรณ์ ", บริษัท ไอดีซีฯ, พ.ศ. 2550

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

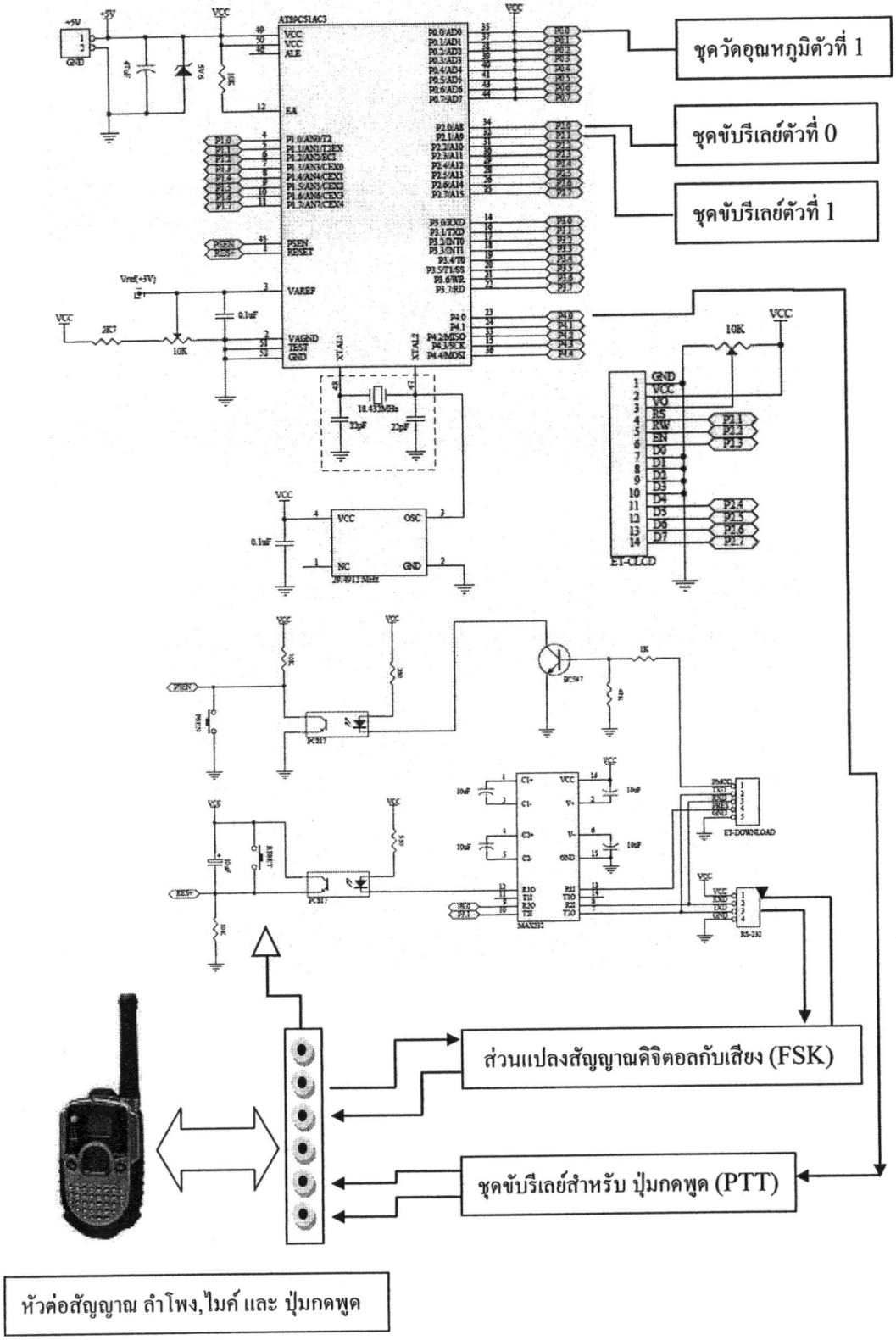
ภาคผนวก

<u>ส่วนฮาร์แวร์ที่สำคัญ</u>

ก. วงจรส่วนตัวแม่ (Main board)



ข. วงจรส่วนตัวลูก (Client)



ผู้วิจัยได้นำบอร์ดสำเร็จรูปของ บริษัท ETT รุ่น ET-MCS51 AC3 มาใช้งานเพราะมีคุณสมบัติตรงตามต้องการ ดังนั้นจึงขอนำเฉพาะส่วนที่ต้องใช้งานมาแสดงการเชื่อมต่อเท่านั้น

ค. **โปรแกรม ส่วนของเครื่องคอมพิวเตอร์ (ภาษา C#)**

โปรแกรมจะยาวมากจึงนำไปใส่ไว้ใน CD-ROM จะสะดวกต่อการนำไปพัฒนาต่อ

ง. **โปรแกรมส่วนของตัวเครื่องลูก (ภาษาซี บน MCS-51 AC3)**

โปรแกรมจะยาวมากจึงนำส่วนหลักเท่านั้นมาแสดง ส่วนอื่นจะไปใส่ไว้ใน CD-ROM จะสะดวกต่อการนำไปพัฒนาต่อ

```
/*****************************************/
/* Example Program For ET-BASE51 AC3 V1.0 */
/* MCU     : AT89C51AC3(XTAL=29.4912 MHz) */
/*         : Run X2 Mode (58.9824 MHz)    */
/* Compiler : Keil C51 (V7.50)            */
/* Write By : MR.ATTASIT LASAKUL (KMITL,Dslabs) */
/*  Descrip : Waiting for Command from Main(PC) */
/*          then process and send back the    */
/*          Information via Wireless          */
/*****************************************/

/* Include  Section */
#include <reg52.h>                  // Standard 8052 SFR : File
#include "delay_ac3.h"
#include "lcd_2line7_ac3.h"
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

/* AT89C51AC3 SFR */
sfr  CKCON = 0x8F;                  // Clock Control
sfr  P4    = 0xC0;                  // Port P4
```

```
// 7 Segmet Display  (for Testing)
/********** I/O PORT **********/
sbit MXCLK = P4^3;
sbit MXLDB = P4^1;
sbit MXDAT = P4^2;

/**** INT-RAM WORKING AREA ****/
unsigned char DISBUF[8];
char buf_Tempc[15]={'\0','\0','\0','\0','\0'
                    ,'\0','\0','\0','\0','\0'
                    ,'\0','\0','\0','\0','\0'};

sbit  Relay_sound = P1^7; // For Sound out
sbit  Relay_PTT = P4^1;   // For Control PTT
sbit  Sw_unit = P1^2;     // For Switch  : Not use by now
sbit  Relay_in0 = P2^3;     // Check  #0     : Not use by now

sbit  Relay_bit0 = P2^0;  // Out put relay #0
sbit  Relay_in1 = P2^2;   // Check  #1   : Not use by now
sbit  Relay_bit1 = P2^1;  // Out put relay #1
sbit  data_bit = P0^0;// IN/OUT PIN for  DS1820

 //*** Just for Testing ****
sbit In_sw6 = P0^1;
sbit In_sw5 = P0^2;
sbit In_sw4 = P0^3;
sbit In_sw3 = P0^4;
sbit In_sw2 = P0^5;
sbit In_sw1 = P0^6;
sbit In_sw0 = P0^7;

 // max7219 segment code
```

```
unsigned char code SEGTAB[16] = {0x7E,0x30,0x6D,0x79,
                                 0x33,0x5B,0x5F,0x70,
                                 0x7F,0x7B,0x77,0x1F,
                                 0x4E,0x3D,0x4F,0x47};

unsigned char lcd_buf[16];
unsigned char Txn;
unsigned char Txcnt,Rxcnt;
bit TX_OK,RX_OK;
unsigned char Txbuff[10];
unsigned char Rxbuff[10];

/* User Define Function */
unsigned int i;
unsigned char gencrc(unsigned char *m);
unsigned char touchbyte(unsigned char d);
unsigned char touchreset(void);
void readrom(void);
void skiprom(void);
void converttemp(void);
void delay(char i);
void readscratchpad(unsigned char *scratch);
void dmsec (unsigned int count);
void mxbyte (unsigned char add,unsigned char dat);
void mxset (void);
void mxload (void);
            //********** START MAIN PROGRAM **********//
void main()
{
  char ch_char[20];
  unsigned char ch;
      unsigned char tempbuf[10];
      int i,j,k,l;
```

```
    float tp;
CKCON = 0x01;                   // Initial X2 Mode (58.9824 MHz)
// CKCON =  0x00;               // x1 mode
    /* Initial MCS51 Serial Port */
 TMOD &= 0x0F;      // Reset old Timer1 Mode Config
 TMOD |= 0x20;      // Update Timer1 = 8 Bit Auto Reload
 SCON = 0x50;       // Serial Port Mode 1 (N,8,1)
 ES   = 1;          // Enable Serial Interupt
 ET1  = 0;          // Disable Timer1 Interrupt
 PCON &= 0x7F;      // SMOD1 = 0 (Disable Double Baudrate)
// TH1  = 0x80;     // Timer1 Baudrate 1200BPS / 58.9824 MHz
// TL1  = 0x80;
      TH1  = 0x00; // Timer1 Baudrate 600BPS / 58.9824 MHz
  TL1  = 0x00;
  TR1  = 1;         // Start Timer1 Generate Baudrate
   TI  = 1;         // Set TI to send First char of UART
   IE =  0x90;      // Enable Interupt
      Relay_in0 = 0x01;     //Set input
      Relay_in1 = 0x01;     //Set input
      Relay_bit0=0x01;      // Relay off
      Relay_bit1=0x01;      // Relay off
      Relay_PTT = 0x01;
      Relay_sound  = 0x01;
       delay_ms(2000);
       lcd_init();                      // Initial LCD in 4-bit mode 1 line
        lcd_putstr(line1,"          ");
        lcd_putstr(line2,"          ");
        lcd_putstr(line1,"-TEST PROGRAM-");
        delay_ms(2000);
            //-----  Setup Time for Temperature DS1820
             delay_ms(500);
            skiprom();
```

```
                converttemp();
                delay(25); //7 interrupt 20000
                skiprom();
                readscratchpad(tempbuf);
                Relay_PTT = 0x01;       // Off PTT
                if(Sw_unit) ch = 0x02;
                else ch = 0x01;
          for(l=0;l<5;l++)
          {
          k = 0x01;
for (j=0;j<=7;j++) {//Display G->F->E->D->C->B->A->DP
          for (i=0;i<=7;i++) DISBUF[i] = k;
                mxload ();
                dmsec (500);
                k = k << 1;
                }
          }
 // lcd_putstr(line1,"  TEST PROGRAM  ");
 // lcd_putstr(line2,"FOR LMSAR MODULE");

//---------- Start Program  Loop ----------------------------
 while(1)
 {
 // lcd_putstr(line1,"  TEST PROGRAM  ");
 // lcd_putstr(line2,"FOR LMSAR MODULE");
          Rxcnt = 0;
          RX_OK = 0;
 //Display Temp on 7 Segment while nothing comming
          while(Rxcnt!=4)
       {
                    //        lcd_putstr(line2,"              ");
                              delay_ms(500);
```

```
            skiprom();
            converttemp();
            delay(25);
            skiprom();
            readscratchpad(tempbuf);
      //    lcd_putstr(line2,"          ");
      if(tempbuf[1] == 0)
            {
                  tp = (float)tempbuf[0]/2;
                  sprintf(buf_Tempc,"%05.2f",tp);
            }
            else
            {
                  tp = ~tempbuf[0] + 1;
                  tp = tp/2;
                  sprintf(buf_Tempc,"%05.2f",tp);
            }
                  //    lcd_putstr(line2,buf_Tempc);
ch_char[0]=buf_Tempc[0];
ch_char[1]='\0';
i=atoi(ch_char);
DISBUF[0] =  SEGTAB[i];
ch_char[0]=buf_Tempc[1];
ch_char[1]='\0';
i=atoi(ch_char);
k =    SEGTAB[i];
k |= 0x80;
DISBUF[1] = k;
ch_char[0]=buf_Tempc[3];
ch_char[1]='\0';
i=atoi(ch_char);
DISBUF[2] =  SEGTAB[i];
```

```
        ch_char[0]=buf_Tempc[4];
        ch_char[1]='\0';
        i=atoi(ch_char);
        DISBUF[3] =  SEGTAB[i];
         mxload ();
        delay_ms(1000);
                }
//-- Operation if this Unit has been selected ---
if(Rxbuff[0]==ch)
{
 //lcd_putstr(line1," THIS UNIT HAS  ");
 // lcd_putstr(line2," BEEN SELECTED! ");
        lcd1_putstr("THIS WAS SELECTED");
         delay_ms(3000);
  // Read Temperature
          ch_char[0]=buf_Tempc[0];
          ch_char[1]=buf_Tempc[1];
          ch_char[2]='\0';
          i=atoi(ch_char);
          Txbuff[0]=(char)(i);
          ch_char[0]=buf_Tempc[3];
          ch_char[1]=buf_Tempc[4];
          ch_char[2]='\0';
          i=atoi(ch_char);
          if(i==0x00)
          { i=0xff; }
          Txbuff[1] = (char)(i);
    if(Rxbuff[1]==0x10)    // Read only
          {
                //Read relay #1 #2
                if (!Relay_in0 && !Relay_in1 )
```

```
            Txbuff[2]=0x01;
                    else if (!Relay_in0 && Relay_in1 )
            Txbuff[2]=0x02;
                    else if (Relay_in0 && !Relay_in1 )
            Txbuff[2]=0x03;
                    else
            Txbuff[2]=0x04;
}else
{ // Write relay #1 #2
  switch (Rxbuff[2])
       {
       case 0x01 : Relay_bit0=0x00;
            Relay_bit1=0x00;
                        break;
       case 0x02 : Relay_bit0=0x01;
            Relay_bit1=0x00;
                        break;
       case 0x03 : Relay_bit0=0x00;
            Relay_bit1=0x01;
                        break;
       case 0x04 : Relay_bit0=0x01;
            Relay_bit1=0x01;
                        break;
            default : break;
        }
  // Read relay #1 #2
       if (!Relay_in0 && !Relay_in1 )
          Txbuff[2]=0x01;
                    else if (!Relay_in0 && Relay_in1 )
          Txbuff[2]=0x02;
                    else if (Relay_in0 && !Relay_in1 )
          Txbuff[2]=0x03;
```

```
                    else
                        Txbuff[2]=0x04;
                }
                 Txbuff[3] = 0xff; //put charactor for Ending Data

                //--- Send DATA Back to PC
                Relay_PTT = 0x00;      // ON PTT
                delay_ms(2000);
                Txn = 4;// May be should put 0xF0 first for sure
                Txcnt = 1;
                TX_OK = 0;
                SBUF = Txbuff[0];
                while(!TX_OK);
                delay_ms(1000);
                Relay_PTT = 0x01;      // Off PTT
                //          lcd_putstr(line1," DATA HAS BEEN  ");
                //         lcd_putstr(line2,"SENT & FINISHED!");
                 delay_ms(1000);
                  }
          }
          while(1);          //Cloud not reach here...
}

//-----------------------------------------//
// Serial interrupt service routine
//-----------------------------------------//
void serial_isr() interrupt 4
{    if(TI)
        {    TI=0;
             if(Txcnt<Txn)
                        SBUF=Txbuff[Txcnt++];
               else
```

```
                    TX_OK = 1;
        }
        if(RI)
        {  RI=0;
           if(Rxcnt<4)
                {  Rxbuff[Rxcnt]=SBUF;
                   Rxcnt++;
                }
        }
}
//***Function for DS1820 ************************
unsigned char gencrc(unsigned char *m)
{
 char i,j;
 unsigned char crc = 0;
 unsigned char k;
 for(j=0;j<7;j++)
 {
  for(i=0;i<8;i++)
  {  k = m[j] ^ crc;
     if((k & 0x01) == 0x01)
    {    crc  = crc ^ 0x18;
         crc >>= 1;
         crc  |= 0x80;
    }
  else
    { crc >>= 1;  }
  m[j] >>= 1;
    }
 }
 return crc;
}
```

```
unsigned char touchbyte(unsigned char d)
{
 unsigned char i;
 unsigned char j;
 char c;
 for(c = 0;c < 8;c++)
 {  j >>= 1;
   if((d & 0x01) == 0x01)
   {  data_bit = 0;
      i = 4;  while(i > 0) {i--;}
      data_bit = 1;
      i = 10;  while(i > 0) {i--;}
  if(data_bit) {j |= 0x80;}
  i = 61;  while(i > 0) {i--;}
    }
   else
    {            data_bit = 0;
                 i = 4;  while(i > 0) {i--;}
                 data_bit = 0;
                 i = 10; while(i > 0) {i--;}
                 if(data_bit) {j |= 0x80;}
                 i = 90; while(i > 0) {i--;}
     }
   data_bit = 1;
        d >>= 1;
 }
 return j;
}

unsigned char touchreset(void)
{
```

```
unsigned char i;
data_bit = 0;        /* start reset pulse */
i = 255; while(i>0) {i--;}
i = 255; while(i>0) {i--;}
i = 255; while(i>0) {i--;}

data_bit = 1;        /* release data line */
 i = 230;
while(i>0)
  {
  if(!data_bit)
     {
    while(!data_bit) {i = 4;while(i>0) i--;}
    return 1;
     }
  else { i--; }
   }
return 0;
}


void readrom(void)
{
  char i;
  unsigned char rombuf[8];
  unsigned char crc;

  while(!touchreset()) {}
  touchbyte(0x33);     /*read rom command */
  for(i=0;i<8;i++)
  {
    rombuf[i] = touchbyte(0xFF);
```

```
 }
 crc = gencrc(rombuf);
 while(crc != rombuf[7]) {}
 return;
}

void skiprom(void)
{
 while(!touchreset()) {}
 touchbyte(0xCC);
 return;
}

void converttemp(void)
{
  touchbyte(0x44);
  return;
 }

 void delay(char i)
 {
   unsigned int j;

    do
     {
      for(j=0;j<1000;j++) {}
      i--;
     }while(i > 0);
  return;
  }

   void readscratchpad(unsigned char *scratch)
```

```
{
 char i;

 touchbyte(0xBE);
 for(i=0;i<9;i++)
 {
  scratch[i] = touchbyte(0xFF);
 }
 touchreset();
return;
}

/** SUB FUNCTION FOR 7 Segment Display (For Testing) ***/
void dmsec (unsigned int count) {// mSec Delay 11.0592 Mhz
   unsigned int i;                  // Keil v5.2  (Speed x 1)
   while (count) {
      i = 115; while (i>0) i--;
      count--;
   }
}

 void mxbyte (unsigned char add,unsigned char dat) {
    unsigned char i;
    for (i=1;i<=8;i++) {         // 8 bit address
       MXDAT = add & 0x80;
       add = add << 1;
       MXCLK = 1;
       MXCLK = 0;
    }
    for (i=1;i<=8;i++) {          // 8 bit data
       MXDAT = dat & 0x80;
       dat = dat << 1;
```

```
    MXCLK = 1;
    MXCLK = 0;
  }
  MXLDB = 1;              // load clock
  MXLDB = 0;
}

void mxset (void) {       // max7219 setup
  MXCLK = 0;
  MXLDB = 0;
  mxbyte (0x0f,0);        // display - normal
  mxbyte (0x0c,1);        // shutdown - normal
  mxbyte (0x09,0);        // decode - no decode
  mxbyte (0x0a,5);        // intensity
  mxbyte (0x0b,7);        // scan limit - 8 digit
}
void mxload (void) {    // max7219 load to display
  mxset ();
  mxbyte (1,DISBUF[0]);
  mxbyte (2,DISBUF[1]);
  mxbyte (3,DISBUF[2]);
  mxbyte (4,DISBUF[3]);
  mxbyte (5,DISBUF[4]);
  mxbyte (6,DISBUF[5]);
  mxbyte (7,DISBUF[6]);
  mxbyte (8,DISBUF[7]);
}

========================= END MAIN PROGTAM ======================
```

จ. คุณลักษณะของไอซี TCM3101

คู่มือฉบับเต็มจะอยู่ในแผ่น CD-ROM โครงงานวิจัย

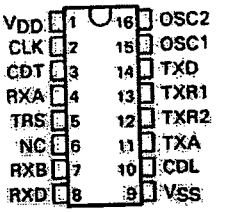
TCM3101
FSK MODEM

D2866, AUGUST 1985—REVISED FEBRUARY 1988

NOT RECOMMENDED FOR NEW DESIGN

For new design, refer to TCM3105

J
DUAL-IN-LINE PACKAGE
(TOP VIEW)



NC—No internal connection

description

The TCM3101 is a single-chip asynchronous Frequency Shift Keying (FSK) modem that uses silicon gate CMOS technology to implement a switched capacitor architecture. It is pin selectable (TXR1, TXR2, and TRS inputs) for a wide range of transmit/receive baud rates and is compatible with the applicable Bell 202 or CCITT V23 specifications. Operation is fully reversible, thereby allowing both forward and backward channels to be used simultaneously.

The transmitter is a programmable frequency synthesizer that provides two output frequencies (on TXA), representing the 'marks' and 'spaces' of the digital signal present on the TXD input.

The receive section is responsible for the demodulation of the analog signal appearing at the RXA input and is based on the principle of frequency to voltage conversion. This section contains a group delay equalizer (to correct phase distortion), automatic gain control, carrier detect level adjustment, and bias distortion adjustment, thereby optimizing performance and giving the lowest possible bit error rate.

Carrier-detect information is given to the system by means of the carrier-detect circuits, which set a flag on the CDT output if the level of received in-band energy falls below a value set on the CDL input for a specified minimum duration.

Caution. These devices have limited built-in gate protection. The leads should be shorted together or the device placed in conductive foam during storage or handling to prevent electrostatic damage to the MOS gates.

ฉ. คุณลักษณะของไอซีวัดอุณหภูมิ

คู่มือฉบับเต็มจะอยู่ในแผ่น CD-ROM โครงงานวิจัย

DALLAS SEMICONDUCTOR

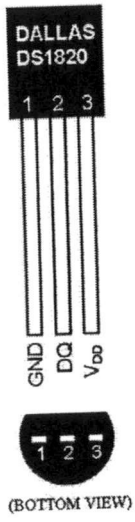
DS18S20
High Precision 1-Wire® Digital Thermometer

www.dalsemi.com

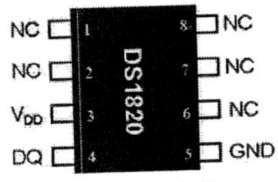
FEATURES

- Unique 1-wire interface requires only one port pin for communication
- Each device has a unique 64-bit serial code stored in an on-board ROM
- Multi-drop capability simplifies distributed temperature sensing applications
- Requires no external components
- Can be powered from data line. Power supply range is 3.0V to 5.5V
- Measures temperatures from –55°C to +125°C (–67°F to +257°F)
- ±0.5°C accuracy from –10°C to +85°C
- 9-bit thermometer resolution
- Converts temperature in 750 ms (max.)
- User-definable nonvolatile alarm settings
- Alarm search command identifies and addresses devices whose temperature is outside of programmed limits (temperature alarm condition)
- Applications include thermostatic controls, industrial systems, consumer products, thermometers, or any thermally sensitive system

PIN ASSIGNMENT



TO-92
(DS18S20)



8-pin 150-mil SOIC
(DS18S20Z)

PIN DESCRIPTION

GND - Ground
DQ - Data In/Out
V_{DD} - Power Supply Voltage
NC - No Connect

DESCRIPTION

The DS18S20 Digital Thermometer provides 9–bit centigrade temperature measurements and has an alarm function with nonvolatile user-programmable upper and lower trigger points. The DS18S20 communicates over a 1-wire bus that by definition requires only one data line (and ground) for communication with a central microprocessor. It has an operating temperature range of –55°C to +125°C and is accurate to ±0.5°C over the range of –10°C to +85°C. In addition, the DS18S20 can derive power directly from the data line ("parasite power"), eliminating the need for an external power supply.

Each DS18S20 has a unique 64-bit serial code, which allows multiple DS18S20s to function on the same 1–wire bus; thus, it is simple to use one microprocessor to control many DS18S20s distributed over a large area. Applications that can benefit from this feature include HVAC environmental controls, temperature monitoring systems inside buildings, equipment or machinery, and process monitoring and control systems.