

การศึกษาการวัดอุณหภูมิและความชื้นแสดงผลผ่านจอ VGA โดยใช้ μ VGA II
A STUDY OF TEMPERATURE AND HUMIDITY THROUGH VGA DISPLAY
BY μ VGA II



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

การศึกษาการวัดอุณหภูมิและความชื้นแสดงผลผ่านจอ VGA โดยใช้ μ VGA II
A STUDY OF TEMPERATURE AND HUMIDITY THROUGH VGA DISPLAY
BY μ VGA II



ปริญญานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรบัณฑิต
สาขาวิชาวิศวกรรมการวัดคุม
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ปีการศึกษา 2556

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

A STUDY OF TEMPERATURE AND HUMIDITY THROUGH VGA DISPLAY
BY μ VGA II



A THESIS SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE DEGREE OF
BACHELOR OF ENGINEERING IN INSTRUMENTATION ENGINEERING
FACULTY OF ENGINEERING
KING MONGKUT'S INSTITUTE OF TECHNOLOGY LADKRABANG
ACADEMIC YEAR 2013

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้


ปริญญาานิพนธ์ปีการศึกษา 2556
คณะวิศวกรรมศาสตร์
สถาบันเทคโนโลยีพระจอมเกล้าเจ้าคุณทหารลาดกระบัง
ใบรับรองปริญญาานิพนธ์

.....

หัวข้อปริญญาานิพนธ์ การศึกษาการวัดอุณหภูมิและความชื้นแสดงผลผ่านจอ VGA โดยใช้ μ VGA II
A STUDY OF TEMPERATURE AND HUMIDITY THUOUGH VGA DISPLAY
BY μ VGA II

นักศึกษาผู้จัดทำ นายณัฐพล ต้นสังวรณธ์ รหัสนักศึกษา 53010417
นายณัฐกร ภาวจันทิก รหัสนักศึกษา 53010442

ปริญญา วิศวกรรมศาสตรบัณฑิต
สาขาวิชา วิศวกรรมการวัดคุม
ปีการศึกษา 2556

| อาจารย์ผู้ควบคุมปริญญาานิพนธ์ | ลายมือชื่อ |
|-------------------------------|--------------------------------------------------------------------------------------|
| รศ.วิริยะ กองรัตน์ |  |

หัวข้อปริญญานิพนธ์ การศึกษาการวัดอุณหภูมิและความชื้นแสดงผลผ่านจอ VGA โดยใช้ μ VGA II

A STUDY OF TEMPERATURE AND HUMIDITY THUOGH VGA DISPLAY BY μ VGA II

นักศึกษาผู้จัดทำ นายณัฐพล ตันสังวรณม์ รหัสนักศึกษา 53010417
นายณัฐกร ภาวจันทิก รหัสนักศึกษา 53010442
อาจารย์ที่ปรึกษา รศ.วิริยะ กองรัตน์
ปีการศึกษา 2556

บทคัดย่อ

ปริญญานิพนธ์นี้ได้นำเสนอการออกแบบและสร้างเครื่องวัดอุณหภูมิและความชื้น แสดงผลผ่านจอ VGA โดยใช้แผงควบคุมไมโครวีจีเอ 2 (μ VGA II) ใช้โปรแกรม 4DGL ในการจัดการภาพบนหน้าจอคอมพิวเตอร์แบบแอลซีดี (LCD) สามารถติดต่อสื่อสารข้อมูลกับอุปกรณ์ตรวจวัด SHT 15 เพื่อนำค่าที่วัดได้มาแสดงผลบนหน้าจอ ที่มีขนาดใหญ่สามารถมองเห็นได้ชัดเจน จากนั้นทำการทดลองอุปกรณ์ตรวจวัดอุณหภูมิและความชื้นจำนวน 2 ตำแหน่ง กับอุปกรณ์ตรวจวัดมาตรฐาน ภายในห้องที่มีการควบคุมอุณหภูมิ ได้ผลการทดลองคือ

การแสดงค่าของอุณหภูมิและความชื้นบนหน้าจอ VGA โดยใช้แผงควบคุมไมโครวีจีเอ 2 ได้อย่างชัดเจน ค่าเปอร์เซ็นต์ความผิดพลาดเฉลี่ยของอุปกรณ์ตรวจรู้ SHT 15 ทั้งสองตำแหน่งมีค่าดังนี้

เปอร์เซ็นต์ความผิดพลาดเฉลี่ยของอุปกรณ์ตรวจรู้ SHT 15 ตัวที่ 1 เทียบกับอุปกรณ์มาตรฐาน

| | | |
|---------------------|--------------|--------|
| ด้านอุณหภูมิ | มีค่าเท่ากับ | 0.91 % |
| ด้านความชื้น | มีค่าเท่ากับ | 1.47 % |
| ด้านอุณหภูมิน้ำค้าง | มีค่าเท่ากับ | 4.18 % |

เปอร์เซ็นต์ความผิดพลาดเฉลี่ยของอุปกรณ์ตรวจรู้ SHT 15 ตัวที่ 2 เทียบกับอุปกรณ์มาตรฐาน

| | | |
|---------------------|--------------|--------|
| ด้านอุณหภูมิ | มีค่าเท่ากับ | 2.00 % |
| ด้านความชื้น | มีค่าเท่ากับ | 1.69 % |
| ด้านอุณหภูมิน้ำค้าง | มีค่าเท่ากับ | 5.11 % |

Thesis Title A STUDY OF TEMPERATURE AND HUMIDITY THUOUGH VGA
DISPLAY BY μ VGA II

Authors Mr.Natapon Tansangworn
Mr.Natakorn Pawajantuk

Thesis Advisor Assoc.Prof. Wiriya Gongrat

Year 2013

ABSTRACT

This thesis presents the design of temperature and humidity Though VGA display by μ VGA II. 4DGL programmed to manage images on the LCD screen (LCD). It can communicate with sensors SHT 15. To be measured and displayed on a screen that large can be seen clearly. Then test the temperature and humidity sensor number 2 position on the standard equipment. (in room with controlled temperature). The experimental results are

The values of temperature and humidity on the VGA monitor by using by μ VGA II is clearly. The percentage of the average error of the sensor SHT 15, it tested two positions.

Percent of the average error of the sensor SHT 15 no. 1 compared to standard equipment.

Temperature = 0.91 %

Humidity = 1.47 %

Dew point = 4.18 %

Percent of the average error of the sensor SHT 15 no. 2 compared to standard equipment.

Temperature = 2.00 %

Humidity = 1.69 %

Dew point = 5.11 %

กิตติกรรมประกาศ

ขอขอบพระคุณ รศ.วิริยะ กองรัตน์ อาจารย์ประจำภาควิชาวิศวกรรมการวัดคุม อาจารย์ประจำภาควิชาวิศวกรรมการวัดคุมทุกท่าน ที่ให้คำปรึกษาในการทำปริญญาานิพนธ์นี้จนสำเร็จลุล่วงไปได้ด้วยดีรวมทั้งที่ได้ให้ความรู้ในการศึกษา ขอขอบคุณเจ้าหน้าที่ของภาควิชาวิศวกรรมการวัดคุมทุกท่าน ที่ให้ความร่วมมือและช่วยเหลือในด้านต่าง ๆ

ขอกราบขอบพระคุณคุณพ่อ แม่ และญาติพี่น้อง รวมทั้งเพื่อนๆ พี่ๆ ที่น่ารักทุกคนที่คอยให้กำลังใจ และให้การสนับสนุนช่วยเหลือในทุกด้านทั้งเรื่องการเรียนรู้ การเงิน และเรื่องส่วนตัวจนทำให้คณะผู้จัดทำประสบความสำเร็จในการจัดทำปริญญาานิพนธ์ โดยไม่มีอุปสรรคใด ๆ

สุดท้ายนี้คุณประโยชน์อันที่พึงมีจากปริญญาานิพนธ์นี้ ผู้เขียนขอบแต่ผู้มีพระคุณทุกท่าน และขอให้ทุกท่านจงมีแต่ความสุขตลอดไป



คณะผู้จัดทำ

สารบัญ

| | หน้า |
|---------------------------------------------------------------------------------------|----------|
| บทคัดย่อภาษาไทย | I |
| บทคัดย่อภาษาอังกฤษ..... | II |
| กิตติกรรมประกาศ..... | III |
| สารบัญ..... | IV |
| สารบัญตาราง..... | VII |
| สารบัญภาพ..... | VIII |
| | |
| บทที่ 1 บทนำ..... | 1 |
| 1.1 ความเป็นมาและความสำคัญของปัญหา..... | 1 |
| 1.2 วัตถุประสงค์ของการศึกษา..... | 1 |
| 1.3 ขอบเขตของปริญญานิพนธ์..... | 2 |
| 1.4 ขั้นตอนของการศึกษา..... | 2 |
| | |
| บทที่ 2 ทฤษฎีและหลักการที่เกี่ยวข้อง..... | 3 |
| 2.1 คอมพิวเตอร์กราฟฟิก (Computer Graphic)..... | 3 |
| 2.1.1 ความละเอียดของภาพ (Resolution)..... | 3 |
| 2.1.2 Channel Color | 4 |
| 2.2 แผงควบคุมไมโครวีจีเอ (μ VGA-II Graphics Engine)..... | 5 |
| 2.2.1 คุณสมบัติของแผงควบคุมไมโครวีจีเอ..... | 6 |
| 2.3 4D Graphics Language (4DGL)..... | 9 |
| 2.4 ความชื้น (Humidity)..... | 9 |
| 2.5 อุณหภูมิ (Temperature)..... | 10 |
| 2.6 บอร์ดวัดอุณหภูมิและความชื้น (AP-1701)..... | 10 |
| 2.6.1 คุณสมบัติทั่วไปของบอร์ด AP-1701..... | 11 |
| 2.7 อุปกรณ์ตรวจรู้อุณหภูมิและความชื้นสัมพัทธ์ (Humidity and Temperature Sensor)..... | 11 |
| 2.8 การส่งข้อมูลแบบอนาลอก (Analog Transmission)..... | 12 |
| 2.9 การส่งสัญญาณข้อมูลแบบดิจิตอล (Digital Transmission)..... | 12 |

สารบัญ (ต่อ)

| | หน้า |
|-----------------------------------------------------------------------------------------------|-----------|
| 2.10 การแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก (Digital to Analog Conversion)..... | 13 |
| 2.11 การแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล (Analog to Digital Conversion)..... | 13 |
| 2.12 ความรู้เบื้องต้นของระบบบัส I2C..... | 13 |
| 2.12.1 คุณสมบัติโดยทั่วไปของบัส I2C..... | 13 |
| 2.12.2 หลักการของบัส I2C..... | 14 |
| 2.12.3 สภาวะที่เกิดขึ้นบนบัส I2C..... | 15 |
| 2.13 มาตรฐานการเชื่อมต่อแบบอนุกรม RS-485..... | 18 |
| บทที่ 3 วิธีการดำเนินการ..... | 20 |
| 3.1 ศึกษาการทำงานของอุปกรณ์ตรวจรู้อุณหภูมิและความชื้น และแผงควบคุมไมโครวีจีเอ 2..... | 20 |
| 3.2 ออกแบบการแสดงผลหน้าจอ VGA..... | 21 |
| 3.3 เขียนโปรแกรมการแสดงผลหน้าจอ VGA จากคอมพิวเตอร์ส่วนบุคคล ผ่านแผงควบคุมไมโครวีจีเอ 2..... | 22 |
| 3.3.1 เขียนผังงาน (Flowcharts)..... | 22 |
| 3.3.2 เขียนโปรแกรมโดยใช้ 4D Workshop4 IDE..... | 24 |
| 3.4 ขั้นตอนการสร้างอุปกรณ์ตรวจรู้อุณหภูมิและความชื้น ผ่านแผงควบคุมไมโคร วีจีเอ 2..... | 46 |
| 3.4.1 แหล่งจ่ายไฟตรง..... | 46 |
| 3.4.2 การต่อวงจรอุปกรณ์ตรวจรู้อุณหภูมิและความชื้น..... | 47 |
| 3.5 ทดสอบโปรแกรมการแสดงผลกับอุปกรณ์ตรวจรู้อุณหภูมิและความชื้น ผ่านแผงควบคุมไมโครวีจีเอ 2..... | 49 |
| บทที่ 4 ผลการทดลอง..... | 52 |

สารบัญ (ต่อ)

| | หน้า |
|-------------------------------------------|------|
| บทที่ 5 สรุปลผลการทดลองและข้อเสนอแนะ..... | 56 |
| 5.1 สรุปลผลการทดลอง..... | 56 |
| 5.2 ปัญหาที่พบในการทดลอง..... | 56 |
| 5.3 ข้อเสนอแนะและแนวทางการพัฒนา..... | 57 |
| บรรณานุกรม..... | 58 |



สารบัญตาราง

| ตารางที่ | หน้า |
|-------------------------------------------------------------------------------------------------|------|
| 2.1 การกำหนดขาสัญญาณอนุกรมอซิงโครนัส..... | 7 |
| 2.2 การกำหนดขาสัญญาณออก..... | 8 |
| 2.3 การกำหนดขาสัญญาณออกแบบขนานและขาสัญญาณ I2C..... | 8 |
| 4.1 ผลการวัดของอุปกรณ์ตรวจรู้ SHT 15 ตัวที่ 1 ณ ที่อุณหภูมิต่างๆเทียบกับอุปกรณ์มาตรฐาน...50 | |
| 4.2 เปอร์เซนต์ความผิดพลาดเฉลี่ยของอุปกรณ์ตรวจรู้ SHT 15 ตัวที่ 1 เทียบกับอุปกรณ์มาตรฐาน..... | 51 |
| 4.3 ผลการวัดของอุปกรณ์ตรวจรู้ SHT 15 ตัวที่ 2 ณ ที่อุณหภูมิต่างๆเทียบกับอุปกรณ์มาตรฐาน...52 | |
| 4.4 เปอร์เซนต์ความผิดพลาดเฉลี่ยของอุปกรณ์ตรวจรู้ SHT 15 ตัวที่ 2 เทียบกับอุปกรณ์มาตรฐาน..... | 52 |



บทที่ 1

บทนำ

1.1 ความเป็นมาและความสำคัญของปัญหา

ในปัจจุบันการแสดงผลอุณหภูมิและความชื้นของเครื่องมือวัด ที่ใช้งานในห้องปฏิบัติการต่างๆ มีรูปแบบการแสดงผลที่จำกัด โดยเฉพาะเครื่องมือวัดส่วนมากจะมีหน้าจอแสดงผลในตัวเครื่องมือ จึงแสดงผลได้เพียงอุณหภูมิหรือแสดงผลได้เพียงความชื้นในพื้นที่หนึ่งหน้าจอนั้น ค่าที่แสดงในหน้าจอมีขนาดเล็กไม่สามารถอ่านค่าจากระยะไกลได้ ดังนั้นจึงได้มีความคิดที่จะการสร้างอุปกรณ์แสดงผลอุณหภูมิและความชื้นผ่านจอ VGA ซึ่งปัจจุบันมีขนาดของจอให้เลือกตั้งแต่ 19-24 นิ้ว ซึ่งจัดได้ว่ามีขนาดที่ใหญ่พอสมควร เมื่อเทียบกับจอ LCD ขนาด 16 ตัวอักษร 2 บรรทัดที่ใช้แสดงค่าต่างๆของไมโครคอนโทรลเลอร์ แต่สำหรับปริญญาานิพนธ์นี้จะอาศัยแผงควบคุมไมโครวีจีเอ 2 (uVGA II) ซึ่งใช้ไมโครคอนโทรลเลอร์ PICASO เป็นหน่วยประมวลผลกลางในการจัดการจอภาพ VGA ขนาดใหญ่ และสามารถอ่านค่าอุณหภูมิและความชื้น โดยเป็นการสร้างภาพจุดสีเล็ก ๆ จำนวนมากมาวางเรียงกันบนจอภาพ จนทำให้เกิดเห็นเป็นภาพต่าง ๆ ตามความต้องการ และเพื่อให้มีความเหมาะสมกับขนาดของจอภาพ ที่จะต้องแสดงผลทั้งอุณหภูมิและความชื้น

1.2 วัตถุประสงค์ของการศึกษา

จุดมุ่งหมายของปริญญาานิพนธ์นี้ เพื่อทำการออกแบบและจัดสร้างอุปกรณ์แสดงผลอุณหภูมิและความชื้นผ่านจอ VGA โดยใช้แผงควบคุมไมโครวีจีเอ 2 (uVGA II) โดยมีวัตถุประสงค์ดังต่อไปนี้

- 1) เพื่อศึกษาการทำงานของแผงควบคุมไมโครวีจีเอ 2
- 1) เพื่อเป็นการศึกษาขั้นตอนการแสดงผลอุณหภูมิและความชื้นจากอุปกรณ์ตรวจวัด
- 2) ออกแบบและสร้างอุปกรณ์แสดงผลอุณหภูมิและความชื้นขึ้นใช้เอง
- 3) เพื่อเป็นการพัฒนาเทคโนโลยีในการแสดงผลให้มีความหลากหลาย
- 4) สามารถปรับเปลี่ยนการแสดงผลอุณหภูมิและความชื้นตามความต้องการของผู้ใช้งาน

ได้

1.3 ขอบเขตของปริญญาโท

- 1) สามารถสร้างเครื่องอุณหภูมิและความชื้นแสดงผลผ่านจอ VGA โดยใช้แผงควบคุมไมโครวีจีเอ 2 (μ VGA II) โดยสามารถวัดอุณหภูมิและความชื้นได้ 2 ตำแหน่ง
- 2) สามารถเขียนโปรแกรม 4DGL แสดงผลอุณหภูมิและความชื้นได้จำนวน 2 ตำแหน่ง

1.4 ขั้นตอนของการศึกษา

ปริญญาโทนี้เป็นงานวิจัยที่จัดทำขึ้นใหม่ โดยใช้แผงควบคุมไมโครวีจีเอ 2 ทำงานร่วมกับอุปกรณ์วัดอุณหภูมิและความชื้น ผ่านช่องทางการสื่อสารแบบอนุกรม RS485 โดยได้เพิ่มเติมการแสดงผลบนหน้าจอ VGA ให้มีรูปแบบการแสดงผลที่หลากหลายมากขึ้น โดยสามารถออกแบบรูปแบบการแสดงผลได้จากคอมพิวเตอร์ส่วนบุคคลกับโปรแกรมกราฟิก 4DGL ในปริญญาโทนี้ได้แบ่งขั้นตอนการศึกษาและจัดสร้างอุปกรณ์แสดงผลอุณหภูมิและความชื้นดังต่อไปนี้

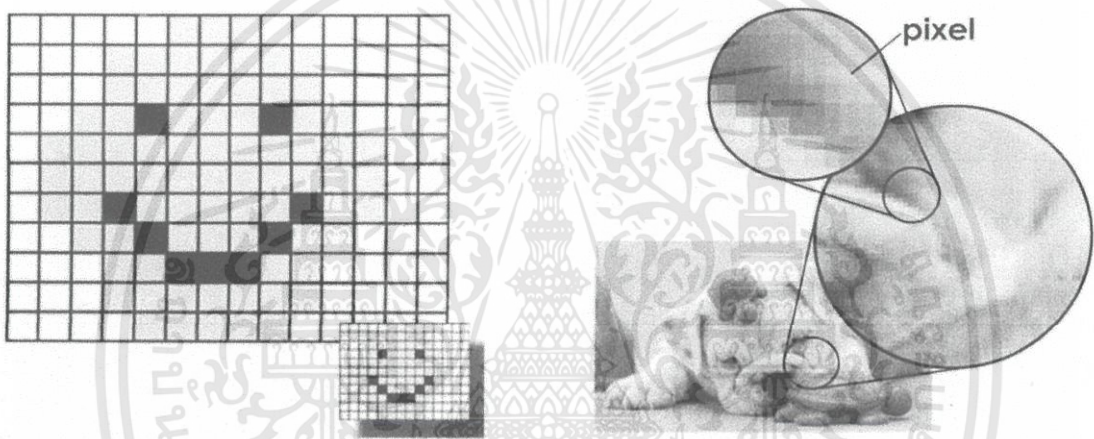
- 1) ศึกษาการทำงานของอุปกรณ์ตรวจอุณหภูมิและความชื้นและแผงควบคุมไมโครวีจีเอ 2
- 2) ออกแบบการแสดงผลหน้าจอ VGA
- 3) เขียนโปรแกรมการแสดงผลหน้าจอ VGA ด้วยคอมพิวเตอร์ส่วนบุคคลผ่านแผงควบคุมไมโครวีจีเอ 2
- 4) ทดสอบโปรแกรมการแสดงผลกับอุปกรณ์ตรวจอุณหภูมิและความชื้น ผ่านแผงควบคุมไมโครวีจีเอ 2
- 5) ทดสอบอุณหภูมิและความชื้น เปรียบเทียบอุปกรณ์ตรวจรุ่นต้นแบบ
- 6) วิเคราะห์ผลที่ได้จากการทดสอบ

บทที่ 2

ทฤษฎีและหลักการที่เกี่ยวข้อง

2.1 คอมพิวเตอร์กราฟฟิก (Computer Graphic)

ภาพในการทำงานของเครื่องคอมพิวเตอร์ เกิดจากการสร้างจุดเล็ก ๆ จำนวนมากมาวางเรียงต่อกัน จนทำให้เกิดเห็นเป็นภาพต่าง ๆ เราเรียกจุดสีเล็กๆ เหล่านี้เรียกว่า พิกเซล (Pixel)



รูปที่ 2.1 จำนวนพิกเซลของภาพที่แสดงบนจอคอมพิวเตอร์

จากรูปที่ 2.1 จุดสีเหลืองและสีดำหลายจุดถูกวางเรียงต่อกันจนเกิดเป็นภาพขึ้นมา สำหรับในกรณีที่ภาพเป็นภาพที่มีสีสวยงามเหมือนจริง มีรายละเอียดมากๆ เช่นรูปถ่าย ก็ยังคงอาศัยหลักการนี้เหมือนกัน เพียงแต่ใช้จุดพิกเซลในภาพจำนวนมากกว่า และจำนวนเฉดสีของจุดพิกเซล ที่ใช้งานก็มีความแตกต่างหลากหลายกว่า

2.1.1 ความละเอียดของภาพ (Resolution)

Resolution ก็คือความละเอียดของภาพ หรืออีกนัยหนึ่งอาจหมายถึงขนาดของภาพก็ได้ วิธีการกำหนด Resolution ของภาพก็คือ การกำหนดว่าด้านกว้างและด้านยาวของภาพจะประกอบไปด้วยกี่จุดพิกเซล เช่น ภาพที่มี Resolution = 640 x 480 Pixel ก็คือภาพนี้มีความกว้างเท่ากับ 640 จุดพิกเซลวางต่อกัน และด้านความยาวจะเท่ากับ 480 จุดพิกเซลวางต่อกัน

อีกปัจจัยหนึ่งที่มีผลต่อความละเอียดของภาพ ก็คือค่า PPI (Pixel Per Inch) หรือจำนวนจุดพิกเซลต่อนิ้ว เช่น ค่า PPI = 72 ก็จะเท่ากับว่าในภาพนั้นจะมีจำนวนจุดพิกเซล 72 จุดต่อเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

1 นิ้ว อย่างไรก็ตามค่านี้อาจไม่มีผลมากนักสำหรับงานตัดต่อภาพ เนื่องจากภาพหรือจอโทรทัศน์แทบทั้งหมดจะใช้ค่า PPI = 72 เหมือนกันหมด (สำหรับงานสิ่งพิมพ์จะใช้ภาพ หรืองานที่มีค่า PPI = 300 ขึ้นไป)

2.1.2 Channel Color

เนื่องจากจำนวนเฉดสี (Channel Color) ต่างๆ ตามธรรมชาติมีนับล้าน ๆ เฉดสี จึงเป็นเรื่องสุดความสามารถเช่นกันที่เครื่องคอมพิวเตอร์ จะมีความสามารถในการจดจำหรือรู้จักเฉดสีตามธรรมชาติได้ทั้งหมด ดังนั้นวิธีการที่คอมพิวเตอร์จะทำการสังเคราะห์บันทึกหรือจดจำเฉดสีต่างๆ จึงจำเป็นต้องอาศัยหลักของการผสมแม่สี โดยจะทำการจดจำปริมาณของแม่สีต่าง ๆ เอาไว้ก่อนที่จะเอาแม่สีเหล่านั้นมาผสมกันจนเกิดเป็นเฉดของพิกเซลหนึ่ง

แต่แม่สีในระบบคอมพิวเตอร์นั้นแตกต่างไปจากแม่สีของสีจริง ๆ ที่เราเคยเรียนในชั่วโมงเรียนศิลปะ เพราะแม่สีของระบบคอมพิวเตอร์นั้นเป็นเรื่องของแสงที่ถูกปล่อยออกมาแล้ว ไปผสมกับบนจอภาพ ทำให้เกิดสีต่าง ๆ ดังนั้นแม่สีในคอมพิวเตอร์ จึงเป็นเรื่องของการผสมสีของแสงเข้าหากัน แม่สีหรือวิธีที่จะผสมแม่สีแบบต่าง ๆ เข้าหากันแบบนี้ เรียกว่าระบบสี ซึ่งก็มีใช้งานอยู่ด้วยกันหลายแบบ เช่น สีระบบ RGB, Lab Color หรือ CMYK เป็นต้น แต่สำหรับระบบสีหลักที่ใช้ในการทำงานกับงานตัดต่อภาพ ก็คือสีในระบบ RGB ซึ่งเป็นระบบสีที่ใช้งานกันในระบบจอภาพทั่วไป

ระบบสีแบบ RGB ประกอบไปด้วยแม่สี 3 สีด้วยกันคือ สีแดง (R), สีเขียว (G), และสีน้ำเงิน (B) ค่าปริมาณของแม่สีทั้ง 3 นี้จะผสมกันในสัดส่วนต่าง ๆ จนเกิดเป็นเฉดสีต่าง ๆ ขึ้นมาจำนวนมาก

ระบบสีแบบ Lab Color เป็นระบบสีที่ให้ความกว้างของเฉดสีมากกว่า RGB มักนำไปใช้กับงานภาพถ่าย ในระบบดิจิทัล ระบบสีแบบนี้จะประกอบไปด้วยแม่สี 3 ชนิด คือ ค่าความสว่าง (Lightness), ค่าวงจรมแม่สีในกลุ่ม a (a Component) และค่าวงจรมแม่สีในกลุ่ม b (b Component)

ระบบสีแบบ CMYK เป็นแม่สีสำหรับงานสิ่งพิมพ์โดยเฉพาะ โดยเรียนแบบมาจากแม่สีของหมึกพิมพ์ ในระบบการพิมพ์ออฟเซต ซึ่งจะประกอบไปด้วยแม่สี 4 ชนิด คือ สีน้ำเงินอมเขียว (Cyan), สีบานเย็น (Magenta), สีเหลือง (Yellow), สีดำ (Black)

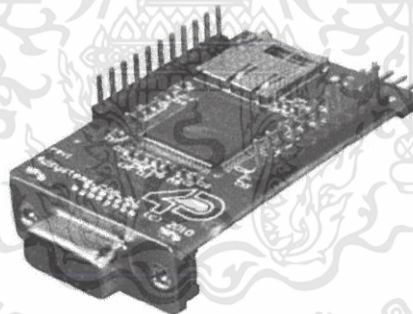
วิธีการเก็บข้อมูลสีของภาพ คอมพิวเตอร์จะอาศัยการจัดเก็บข้อมูลสีแบบแยกเฉพาะแต่ละ พิกเซล โดยในแต่ละพิกเซลเองก็จะมีเก็บข้อมูลตามจำนวนของแม่สีที่ใช้งานกับภาพนั้น เราเรียกการแบ่งเก็บข้อมูลแม่สีของภาพว่าจำนวนเฉดสีของภาพ เช่น ภาพในระบบ RGB ที่ใช้งานกันทั่วไปจะแบ่งข้อมูลสีเป็น 3 เฉดสี คือ เฉดสีแดง เฉดสีเขียว และ เฉดสีน้ำเงิน

วิธีเก็บค่าเฉดสีของแม่สีก็อาศัยการเก็บค่าความเข้มของแม่สีในแต่ละจุดพิกเซลนั้น โดยอาศัยแม่สีที่มีความเข้มแตกต่างกันก็จะเกิดเป็นสีต่าง ๆ ขึ้นมา ระดับการแบ่งส่วนความเข้มเอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ของแม่สีมีผลต่อจำนวนเฉดสีที่พิกเซลสามารถแสดงออกมาได้ ยิ่งแม่สีสามารถแสดงความแตกต่างของความเข้มแม่สีได้มากระดับเท่าไร เราจะได้ภาพที่มีความสามารถในการแสดงเฉดสีมากขึ้นตามไปด้วย การที่แม่สีจะแสดงระดับความแตกต่างของความเข้มขั้นนี้เรียกว่าค่าความลึกของสี (Color Depth) เช่นภาพ RGB แบบ 24 Bit ซึ่งเป็นภาพมาตรฐานใช้งานทั่วไป ด้วยจำนวนเฉดสีของแม่สีแบบ R G และ B โดยในแต่ละจำนวนเฉดสีของแม่สีจะเก็บข้อมูลสีด้วยหน่วยความจำขนาด 8 บิตต่อแม่สี ซึ่งสามารถแสดงความเข้มขั้นของแม่สีได้ 256 เฉด (มีค่าความเข้มของแม่สีเป็น 0-255) เมื่อนำเอาแม่สี 256 เฉดทั้ง 3 เฉดสีมาผสมกันก็จะสามารถแสดงเฉดสีได้ถึง 16.7 ล้านสีเลยทีเดียว

2.2 แผงควบคุมไมโครวีจีเอ (μVGA-II Graphics Engine)

จากรูปที่ 2.2 แสดงภาพของแผงควบคุมไมโครวีจีเอ หรือ μVGA-II นี้มีขนาดเล็กกระทัดรัดง่ายต่อการนำมาประยุกต์ใช้งาน สามารถแสดงผลกราฟฟิกได้อย่างมีประสิทธิภาพ ในรูปแบบ QVGA / VGA / WVAG ด้วยไมโครคอนโทรลเลอร์ PICASO-GFX อย่างเช่น ข้อความ (Text) รูปภาพ (Image) ภาพเคลื่อนไหว (Animation)



รูปที่ 2.2 แผงควบคุมไมโครวีจีเอ (μVGA-II Graphics Engine)

ไมโครคอนโทรลเลอร์ PICASO-GFX เป็นหน่วยประมวลผลกลาง ที่มีความสามารถทางด้านกราฟฟิก และยังสามารถเชื่อมต่อกับอุปกรณ์ต่าง ๆ ผ่านช่องทางอนุกรม (Serial Ports) มีระบบการจัดเก็บข้อมูลที่ประมวลผลได้ลงในหน่วยความจำแบบ ไมโครเอสดี (Micro-SD Memory Card) มีขาสัญญาณแบบดิจิทัลที่เป็นอินพุตและเอาต์พุต (General Purpose I/O Pins : GPIO) เพื่อใช้ในการติดต่อสื่อสารกับอุปกรณ์ภายนอก จึงทำให้ไมโครคอนโทรลเลอร์ PICASO-GFX เป็นหัวใจหลักในการทำงานของแผงควบคุมไมโครวีจีเอ การสั่งงานแผงควบคุมไมโครวีจีเอ ทำได้โดยการเขียนโปรแกรมที่มีชื่อเรียกว่า ภาษากราฟฟิก 4ดี (4D Graphics Language : 4DGL)

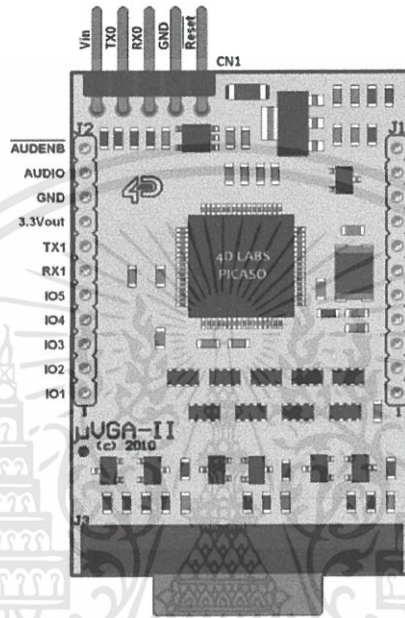
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.2.1 คุณสมบัติของแผงควบคุมไมโครวีจีเอ

- 1) สามารถแสดงผลตามมาตรฐานสำหรับจอภาพสี (Video Graphics Array: VGA) ได้ทั้ง หน้าจอภาพแสดงผลทั่วไปและจอภาพแสดงผลแบบ LCD
- 2) รองรับความละเอียดของภาพหรือขนาดของภาพ (Resolutions) ทั้งแบบ 320 × 240 (QVGA) , 640 × 480 (VGA) , 800 × 480 (WVGA) และยังสามารถตั้งค่าความละเอียดของภาพ ในรูปแบบของ $x \times y = 405KB$
- 3) รองรับระบบสีแบบ RGB 65K
- 4) มี 5 ขาสัญญาณที่สามารถเชื่อมต่อกับอุปกรณ์อนุกรม (Serial) คือ Vcc,TX,RX ,GND ,RESET
- 4) ไมโครคอนโทรลเลอร์ PICASO-GFX เป็นหน่วยประมวลผลกลาง
- 5) มีหน่วยความจำแบบ Flash ขนาด 15KB สำหรับเขียนโปรแกรม และหน่วยความจำแบบ SRAM ขนาด 14KB สำหรับใช้ในงานอื่น ๆ
- 6) พอร์ตอนุกรมอซิงโครนัส (Asynchronous Serial Ports) จำนวน 2 พอร์ต คือ COM0 และ COM1 มีช่องทางการเชื่อมต่อแบบ TTL ด้วยความเร็ว 300 – 256K baud
- 7) มีช่องทางการเชื่อมต่อแบบ I2C
- 8) มีช่องใส่หน่วยความจำไมโครเอสดีการ์ด (Micro-SD Memory Card) บนบอร์ดเพื่อการจัดเก็บข้อมูลมัลติมีเดียและเก็บข้อมูลได้ถึง 4 GB
- 9) ตัวนับเวลา (Timer) ขนาด 16 บิต ความละเอียด 1 mS จำนวน 8 ตัว
- 10) มีขาสัญญาณแบบ GPIO จำนวน 13 ขา และสามารถใช้เป็นแบบระบบ Bus ส่งข้อมูลด้วยความเร็วพร้อมกันทั้ง 8 บิต (8-bit parallel)
- 11) มี Micro-SD Memory Card รองรับความจุในการเก็บข้อมูลได้ 4 GB
- 12) รองรับการจัดเก็บแฟ้มข้อมูลแบบ DOS (Disk Operating System) ที่มีรูปแบบเป็น FAT16
- 13) มีขาสัญญาณเสียง PWM (Plus Width Modulation) จำนวน 1 ขา
- 14) รองรับภาษา 4D Graphics Language
- 15) รูปภาพ (Images) ภาพเคลื่อนไหว (Animation) และภาพวิดีโอ (Video Clips) มีการแสดงผลแบบ Full Color
- 16) รองรับรูปแบบตัวหนังสือทุกแบบ
- 17) มีขาสัญญาณจำนวน 2×11 ขา ขนาดของขาสีบ 2.54 mm

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

- 18) มีช่องต่อสัญญาณมาตรฐาน VGA แบบ 15 pin D-type ซึ่งสามารถเชื่อมต่อกับจอภาพแบบ VGA
- 19) ใช้ได้กับไฟฟ้ากระแสตรง 4.0 โวลต์ ถึง 5.5 โวลต์



รูปที่ 2.3 แสดงขาสัญญาณต่าง ๆ ของแผงควบคุมไมโครวิจีโอ

จากรูปที่ 2.3 แสดงขาสัญญาณต่าง ๆ ของแผงควบคุมไมโครวิจีโอ ซึ่งมีรายละเอียดการทำงานตามตารางที่ 2.1 ,2.2 และ 2.3 ดังนี้

ตารางที่ 2.1 การกำหนดขาสัญญาณอนุกรมอซิงโครนัส

| ขา | สัญลักษณ์ | I / O | คำอธิบาย |
|----|-----------|-------|----------------------------------------------------------|
| 1 | Vin | P | ขาจ่ายไฟหลัก 4.0 โวลต์ ถึง 5.5 โวลต์ ทั่วไปใช้ 5.0 โวลต์ |
| 2 | TX0 | O | ขาอนุกรมอซิงโครนัสส่งสัญญาณ |
| 3 | RX0 | I | ขาอนุกรมอซิงโครนัสรับสัญญาณ |
| 4 | GND | P | ขาต่อสายดิน |
| 5 | Reset | I | ขารีเซ็ตหลัก |

ตารางที่ 2.2 การกำหนดขาสัญญาณออก

| ขา | สัญลักษณ์ | I / O | คำอธิบาย |
|----|-----------------|-------|-------------------------------------------------|
| 1 | IO1 | I/O | ขาเอาต์พุตที่ 1 |
| 2 | IO2 | I/O | ขาเอาต์พุตที่ 2 |
| 3 | IO3 | I/O | ขาเอาต์พุตที่ 3 |
| 4 | IO4 / BUS_RD | I/O | ขาเอาต์พุตที่ 4 หรือ บัสเขียน (Bus Read) |
| 5 | IO5 / BUS_WR | I/O | ขาเอาต์พุตที่ 5 หรือ บัสอ่าน (Bus Write) |
| 6 | RX1 | I | อซิงโครนัสพอร์ตที่ 1 ขารับสัญญาณ (COM1 Rx) |
| 7 | TX1 | O | อซิงโครนัสพอร์ตที่ 1 ขาส่งสัญญาณ (COM1 Tx) |
| 8 | 3.3Vout | P | เอาต์พุต 3.3 โวลต์ กระแสสูงสุด 400 มิลลิแอมแปร์ |
| 9 | GND | P | สายดิน |
| 10 | AUDIO | O | PWM Audio เอาต์พุต |
| 11 | AUDENB | O | Audio Enable เอาต์พุต |

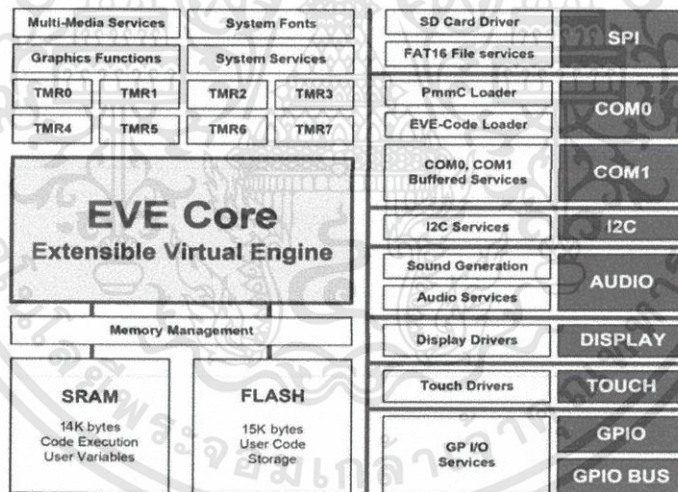
ตารางที่ 2.3 การกำหนดขาสัญญาณออกแบบขนานและขาสัญญาณ I2C

| ขา | สัญลักษณ์ | I / O | คำอธิบาย |
|----|-----------|-------|--------------------------|
| 1 | BUS7 | I/O | IO บัส (BUS0..7) บิต 7 |
| 2 | BUS6 | I/O | IO บัส (BUS0..7) บิต 6 |
| 3 | BUS5 | I/O | IO บัส (BUS0..7) บิต 5 |
| 4 | BUS4 | I/O | IO บัส (BUS0..7) บิต 4 |
| 5 | BUS3 | I/O | IO บัส (BUS0..7) บิต 3 |
| 6 | BUS2 | I/O | IO บัส (BUS0..7) บิต 2 |
| 7 | BUS1 | I/O | IO บัส (BUS0..7) บิต 1 |
| 8 | BUS0 | I/O | IO บัส (BUS0..7) บิต 0 |
| 9 | SCL | O | เอาต์พุตสัญญาณนาฬิกา I2C |
| 10 | SDA | I/O | ข้อมูลอนุกรม I2C |
| 11 | Vin | O | ขาจ่ายไฟหลัก |

2.3 4D Graphics Language (4DGL)

ไมโครโปรเซสเซอร์ของ 4D-Labs (GOLDELOX-GFX, PICASO-GFX และ DIABLO-GFX) มีประสิทธิภาพมากในการจัดการกราฟฟิก ด้วยเทคโนโลยีการประมวลผลแบบ E.V.E (Extensible Virtual Engine) โดยการแปลความหมายของโปรแกรม (Complied) ด้วย 4DGL ซึ่งเป็นภาษาระดับสูง (High Level Language) เรียนรู้และทำความเข้าใจง่าย

4DGL จัดเป็นภาษาโปรแกรมกราฟฟิกเชิงวัตถุ (Graphics Oriented Language) ทำให้พัฒนาโปรแกรมได้ง่ายและรวดเร็ว มีฟังก์ชันในการจัดการกราฟฟิก ข้อความ และระบบจัดการเพิ่มข้อมูลที่มีประสิทธิภาพ โครงสร้างของโปรแกรมเข้าใจง่ายเพราะคล้ายกับ ภาษา C ภาษา Basic ภาษา Pascal ทำให้ผู้เขียนโปรแกรมไม่ต้องทำการศึกษาเพิ่มเติมมาก จากรูปที่ 2.4 แผนผังของการประมวลผลคำสั่งภายในของ PICASO-GFX เช่น การจัดการหน่วยความจำแบบ SRAM และ Flash คำสั่งสำหรับจัดการกับจอภาพ (Display Drivers) คำสั่งสำหรับจัดการกับเสียง (Audio Drivers) เป็นต้น



รูปที่ 2.4 แผนผังของการประมวลผลคำสั่งภายในของ PICASO-GFX

2.4 ความชื้น (Humidity)

ความชื้น หมายถึงจำนวนไอน้ำที่มีอยู่ในอากาศ ความชื้นของอากาศมีการเปลี่ยนแปลงอยู่ตลอดเวลาจะมากหรือน้อยขึ้นอยู่กับความดันและอุณหภูมิ ความชื้นสัมพัทธ์ (Relativity Humidity) หมายถึง “อัตราส่วนของปริมาณไอน้ำที่มีอยู่จริงในอากาศต่อปริมาณไอน้ำที่จะทำให้อากาศอิ่มตัว ณ อุณหภูมิเดียวกัน” หรือ “อัตราส่วนของความดันไอน้ำที่มีอยู่จริงต่อความดันไอน้ำอิ่มตัว” ค่าความชื้นสัมพัทธ์แสดงในรูปของร้อยละ (%) ปริมาณของไอน้ำในอากาศ ขึ้นอยู่กับอุณหภูมิของอากาศร้อน สามารถเก็บไอน้ำได้มากกว่าอากาศเย็น ดังนั้นหากเราลดอุณหภูมิของอากาศจนถึงจุด ๆ หนึ่ง เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

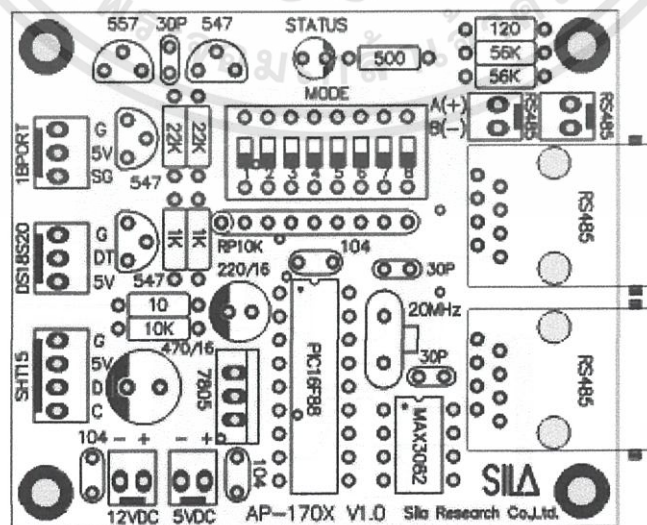
จะเกิด “อากาศอิ่มตัว” (Saturated Air) อากาศไม่สามารถเก็บกักไอน้ำไว้ได้มากกว่านี้ หรือกล่าวได้ว่า อากาศมีความชื้นสัมพัทธ์ 100% ดังนั้นหากอุณหภูมิยังคงลดต่ำลงอีก ไอน้ำจะเปลี่ยนสถานะเป็นของเหลว อุณหภูมิที่ทำให้เกิดการควบแน่นนี้เรียกว่า “จุดน้ำค้าง” (Dew Point)

2.5 อุณหภูมิ (Temperature)

อุณหภูมินับเป็นค่าการวัดพื้นฐานหรือพารามิเตอร์ตัวหนึ่งที่ต้องทำการวัดค่า เพื่อนำไปใช้ประโยชน์ในระบบการควบคุมให้เป็นไปตามความต้องการ คำว่า อุณหภูมิและความร้อน (Heat) มีความหมายใกล้เคียงกันมาก แต่อุณหภูมิจะหมายถึงระดับของความร้อน (Degree of Heat) คือ อุณหภูมิเป็นตัวแทนของความร้อน ส่วนความร้อนหมายถึง ปริมาณพลังงานความร้อน (Quantity of Heat Energy) หน่วยวัดอุณหภูมิมียุคหลายแบบที่นิยมใช้กัน คือ หน่วยเซลเซียส (Celsius) นิยมใช้กันทั่วไปมากเช่น หน่วยฟาเรนไฮต์ (Fahrenheit), และหน่วยเคลวิน (Kelvin) ซึ่งเป็นหน่วยมาตรฐานสากล

2.6 บอร์ดวัดอุณหภูมิและความชื้น (AP-1701)

AP-1701 คือบอร์ดวัดอุณหภูมิและความชื้น มีหัววัดแบบดิจิตอลความละเอียดสูง ดังรูปที่ 2.5 ใช้วัดในบรรยากาศเน้นการวัดระยะไกลโดยผ่านการสื่อสาร RS485 สามารถต่อสายพ่วงกันเป็น Network ได้อย่างสะดวก สามารถตั้ง Address แยกต่างกันได้ถึง 31 Node ด้วย Dip-Switch ตัวบอร์ดมีขนาดเล็ก หัววัดสามารถต่อได้ไกลถึง 100 เมตร ประยุกต์ใช้งานได้หลากหลาย โดยเฉพาะการวัดหลาย ๆ จุด และระยะไกล



รูปที่ 2.5 บอร์ดวัดอุณหภูมิและความชื้นแบบ RS485

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

2.6.1 คุณสมบัติทั่วไปของบอร์ด AP-1701

- 1) ทำงานด้วยไมโครคอนโทรลเลอร์เบอร์ PIC16F88 ความถี่ 20 MHz
- 2) มี LED สีแดงแสดงสถานะการสื่อสาร RS485
- 3) มี Dip-Switch 8 จุด เพื่อตั้งคุณสมบัติต่าง ๆ คือ ชนิดของหัววัด, ความเร็วการสื่อสาร, Node Address
- 4) RS485 มีขั้วต่อแบบ 2 Pin 2 ตัว และขั้ว RJ45 8 Pin 2 ตัว
- 5) เลือกใช้ไฟเลี้ยงแบบ 5 VDC หรือ 12 VDC ในกรณี 12 VDC สามารถจ่ายเข้าทางขั้ว RJ45
- 6) มี I/O พอร์ต (Port) อีกระยะ 1 Bit ด้วย และขั้วต่อแบบ 3 Pin ใช้เป็นอินพุตและเอาต์พุต (Input / Output) แบบดิจิตอล (Digital) ได้
- 7) สามารถตั้งความเร็วการสื่อสารได้ 2 ระดับ คือ 9600 และ 19200 บิตต่อวินาที
- 8) สามารถตั้ง Node Address ของบอร์ดได้ตั้งแต่ 0 – 31 (0 คือไม่มี Address) ทำให้ต่อพ่วงกันได้สูงสุด 31 บอร์ดด้วยสายสัญญาณ RS485 Network

2.7 อุปกรณ์ตรวจรู้อุณหภูมิและความชื้นสัมพัทธ์ (Humidity and Temperature Sensor)



รูปที่ 2.6 อุปกรณ์ตรวจรู้อุณหภูมิและความชื้นสัมพัทธ์

จากรูปที่ 2.6 SHT15 เป็นอุปกรณ์ตรวจรู้ที่ผลิตจากบริษัท Sensirion โดยมีแผ่นวงจรอิเล็กทรอนิกส์ที่มีตัวตรวจจับอุณหภูมิกับความชื้นสัมพัทธ์ยึดติดที่แผ่นวงจร (Surface Mount Technology : SMT) ภายในอุปกรณ์ตรวจรู้มีระบบประมวลผลสัญญาณพัลส์ (Plus Signal Processing) มีสัญญาณออกเป็นสัญญาณดิจิตอล การวัดความชื้นสัมพัทธ์ใช้อุปกรณ์ตรวจรู้แบบคาปาซิทีฟ (Capacitive Sensor) และการวัดอุณหภูมิทำการวัดโดย Band-gap Sensor มีการรับประกันความเชื่อถือจาก CMOSens® Technology เซนเซอร์ทั้งคู่เป็นการรวมกันทำงานที่ลงตัวโดยทำการแปลงสัญญาณจากอนาล็อกเป็นดิจิตอลขนาด 14 บิต และมีวงจรอินเทอร์เฟซแบบอนุกรม ด้วยสิ่งนี้จึงทำให้ได้รับสัญญาณที่มีคุณภาพที่เหนือกว่า เวลาในการตอบสนองเร็วและมีสัญญาณรบกวนน้อย เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

หัววัด SHT 15 แต่ตัวจะถูกปรับแต่งในตัวควบคุมความชื้นที่มีความเที่ยงตรง การปรับแต่งค่าสัมประสิทธิ์จะถูกเก็บไว้ในหน่วยความจำ OTP (One Time Programming) ใช้วงจรอนุกรม 2 สาย (2 Wire Serial Interface) ในการส่งสัญญาณค่าที่ตรวจวัดได้ สามารถควบคุมแรงดันภายในตัวตรวจจับได้และเร็วมีขนาดเล็ก ใช้กำลังไฟฟ้าน้อย มีความสามารถวัดอุณหภูมิตั้งแต่ -40 ถึง 120 องศาเซลเซียส ความละเอียด 0.1 องศาและวัดความชื้นตั้งแต่ 10 ถึง 90 เปอร์เซ็นต์ ความละเอียด 0.1 เปอร์เซ็นต์

2.8 การส่งข้อมูลแบบอนาล็อก (Analog Transmission)

เป็นการส่งสัญญาณแบบอนาล็อก โดยไม่สนใจในสิ่งที่ถูกบรรจุรวมกันอยู่ในสัญญาณเลย สัญญาณจะแทนข้อมูลอนาล็อก (เช่น สัญญาณเสียง) หรือ ข้อมูลดิจิทัล (เช่น ข้อมูลไบนารีผ่านโมเด็ม) สัญญาณอนาล็อกที่ทำการส่งออกไปพลังงานจะอ่อนลงไปเรื่อยๆ เมื่อระยะทางเพิ่มขึ้น ดังนั้นในการส่งสัญญาณอนาล็อกไประยะไกล ๆ จึงต้องอาศัยเครื่องขยายสัญญาณ หรือ แอมพลิไฟเออร์ (Amplifier) เพื่อเพิ่มพลังงานให้กับสัญญาณ แต่ในการใช้เครื่องขยายสัญญาณจะมีการสร้างสัญญาณรบกวนขึ้น (Noise) รวมกับสัญญาณข้อมูลด้วย ดังนั้นยิ่งระยะทางไกลมากเท่าไร ก็ยังมีสัญญาณรบกวนมากขึ้นเท่านั้น การส่งสัญญาณอนาล็อกจึงต้องการวงจรกรองสัญญาณ (Filter) เพื่อกรองเอาสัญญาณรบกวนออกอีก

2.9 การส่งสัญญาณข้อมูลแบบดิจิทัล (Digital Transmission)

ส่วนในการส่งสัญญาณแบบดิจิทัล จะสนใจทุกสิ่งทุกอย่างมาบรรจุในสัญญาณ เมื่อระยะทางเพิ่มขึ้นมากขึ้น จะทำให้สัญญาณดิจิทัลจางหายไปได้ จึงจำเป็นต้องใช้อุปกรณ์ทบทวนสัญญาณหรือ รีพีตเตอร์ (Repeater) เพื่อกู้ (Recover) รูปแบบของสัญญาณที่มีลักษณะ เป็น " 1 " และ " 0 " เสียก่อน แล้วจึงส่งสัญญาณใหม่ต่อไป เราสามารถนำเอาอุปกรณ์ทบทวนสัญญาณมาใช้ในการส่งสัญญาณอนาล็อกที่มีข้อมูลเป็นแบบดิจิทัลได้ เครื่องทบทวนสัญญาณจะทำการกู้ข้อมูลดิจิทัลจากสัญญาณอนาล็อกและสร้างสัญญาณขึ้นมาใหม่ แล้วลบสัญญาณอนาล็อกที่ส่งมาด้วยออกไป ดังนั้นจะไม่มีสัญญาณรบกวนที่ติดมากับสัญญาณอนาล็อกหลงเหลืออยู่เลย คำถามคือว่าเราจะเลือก ใช้วิธีการส่งสัญญาณข้อมูลเป็นแบบอนาล็อกหรือแบบดิจิทัลดี คำตอบก็ขึ้นอยู่กับระยะทางในการส่งข้อมูลนั้น ใกล้หรือไกล ถ้าเป็นระยะทางใกล้ ๆ สามารถเดินสายสื่อสารดิจิทัลได้ก็ควรจะเลือก ใช้การส่งสัญญาณแบบดิจิทัล ส่วนการส่งสัญญาณข้อมูลในระยะทางไกล ๆ การสื่อสารของไทยเรายังคงเป็นระบบอนาล็อกอยู่ เช่น ระบบโทรศัพท์ หรือระบบโทรเลข ดังนั้นจึงควรเลือก ใช้วิธีการส่งสัญญาณข้อมูลเป็นแบบอนาล็อก

2.10 การแปลงสัญญาณดิจิทัลเป็นสัญญาณอนาล็อก (Digital to Analog Conversion)

ในปัจจุบันการส่งสัญญาณข้อมูลดิจิทัลโดยผ่านช่องทางสื่อสารแบบอนาล็อกที่เราคุ้นเคยกันได้แก่ การส่งข้อมูลคอมพิวเตอร์ผ่านทางเครือข่าย โทรศัพท์สาธารณะ เครือข่ายโทรศัพท์มือถือแบบมาเพื่อทำการสลับสวิตช์ และส่งสัญญาณอนาล็อกซึ่งเป็นย่านความถี่ของเสียง หรือประมาณ 300-400 Hz อุปกรณ์อิเล็กทรอนิกส์ที่ทำหน้าที่แปลงสัญญาณข้อมูลดิจิทัลให้เป็นสัญญาณอนาล็อกย่านความถี่เสียง เราเรียกว่าโมเด็ม (MODEM หรือ Modulator-Demodulator)

สำหรับเทคนิคการแปลงสัญญาณข้อมูลดิจิทัล ให้เป็นสัญญาณอนาล็อกนั้นมีอยู่ด้วยกันทั้งหมด 3 วิธีคือ

- 1) การมอดูเลตเลขทางแอมพลิจูด (Amplitude-Shift Keying หรือ ASK)
- 2) การมอดูเลตเชิงเลขความถี่ (Frequency-Shift Keying หรือ FSKK)
- 3) การมอดูเลตเลขเชิงทางเฟส (Phase-Shift Keying หรือ PSK)

2.11 การแปลงสัญญาณอนาล็อกเป็นสัญญาณดิจิทัล (Analog to Digital Conversion)

ในกรณีถ้าระบบเครือข่ายของเราเป็นแบบดิจิทัล ก็สามารถส่งผ่านสัญญาณดิจิทัลสู่ช่องทางสื่อสารดิจิทัลได้โดยตรง เช่น ในระบบเครือข่าย ISDN หรือไปรษณีย์อิเล็กทรอนิกส์ (E-Mail or Electronic Mail) เป็นต้น เราสามารถส่งสัญญาณดิจิทัล ที่ออกจากคอมพิวเตอร์สู่เครือข่ายได้ โดยตรง ไม่ต้องผ่านโมเด็มและในทำนองเดียวกัน เราสามารถส่งสัญญาณอนาล็อกผ่านเข้าไปในระบบเครือข่ายดิจิทัลได้ โดยการเปลี่ยนแปลงสัญญาณอนาล็อกให้เป็นสัญญาณดิจิทัลเสียก่อน โดยใช้อุปกรณ์ที่ทำงานตรงกันข้ามกับโมเด็มคือ โคเดก (CODEC หรือ COder/DECoder)

เทคนิคในการเปลี่ยนแปลงสัญญาณอนาล็อก เป็นสัญญาณดิจิทัล แบ่งออกเป็น 2 วิธีคือ

- 1) การมอดูเลตทางแอมพลิจูดของพัลส์ หรือ PAM (Pulse Amplitude Modulation)
- (2) การมอดูเลตแบบรหัสพัลส์ หรือ PCM (Pulse Code Modulation)

2.12 ความรู้เบื้องต้นของระบบบัส I2C

I2C ย่อมาจาก Inter-IC Communication หมายถึง การติดต่อสื่อสารระหว่างไอซี โดยบัส I2C ได้รับการพัฒนาขึ้นโดยฟิลิปส์ (Philips) ด้วยจุดมุ่งหมายหลักคือ ต้องการให้ไอซีหรือโมดูลสามารถติดต่อ สั่งงาน และควบคุมภายใต้สายสัญญาณเพียง 2 เส้น เส้นหนึ่งคือ สายข้อมูล อีกเส้นหนึ่งคือ สายสัญญาณนาฬิกาที่ใช้ในการกำหนดจังหวะการทำงาน การต่อร่วมกันของอุปกรณ์บนบัส

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

I2C ทำได้ง่ายมาก เพียงต่อสายข้อมูลและสายสัญญาณนาฬิกาของอุปกรณ์แต่ละตัวขนานกันไป ส่วนการกำหนดแอดเดรสสำหรับติดต่ออุปกรณ์แต่ละตัว จะใช้รหัสข้อมูลรวมกับการกำหนดสถานะลอจิกที่ขาแอดเดรส

สายข้อมูลบนบัส I2C มีชื่อเรียกอย่างเป็นทางการว่า สายข้อมูลอนุกรม หรือ SDA (Serial Data Line) ส่วนสายสัญญาณนาฬิกามีชื่อเรียกว่า สายสัญญาณนาฬิกาอนุกรม หรือ SCL (Serial Clock Line)

2.12.1 คุณสมบัติโดยทั่วไปของบัส I2C

สาย SDA และ SCL เป็นสายสัญญาณ 2 ทิศทาง (Bi-directional Line) ต้องมีการต่อตัวต้านทานพูลอัพกับแรงดัน +5V ไว้ตลอดเวลา เพื่อให้สายมีสถานะลอจิกสูงในขณะที่ไม่มีการติดต่อใช้งาน ทั้งยังช่วยในการป้องกันสัญญาณรบกวนที่อาจมีเข้ามาในสายสัญญาณทั้งสองวงจรเอาท์พุทของอุปกรณ์ที่ต่ออยู่บนบัส I2C ต้องมีลักษณะเป็นวงจรทรานเปิด (Open-drain) หรือคอลเล็กเตอร์เปิด (Open-collector)

อัตราการถ่ายเทข้อมูลบนระบบบัส I2C สูงถึง 100 กิโลบิตต่อวินาทีในโหมดปกติ (Standard Mode) และสูงถึง 400 กิโลบิตต่อวินาทีในโหมดความเร็วสูง (Fast Mode) อุปกรณ์ที่ต่ออยู่บนบัส I2C จะต้องมีค่าความจุไฟฟ้ารวมที่เกิดขึ้นระหว่างสาย SDA และ SCL ไม่เกิน 400pF การเข้าถึงอุปกรณ์บนบัส I2C ใช้ข้อมูลสำหรับการเข้าถึง 2 ค่าคือ 7 บิต (7-bit addressing) หรือ 10 บิต (10-bit addressing)

2.12.2 หลักการของบัส I2C

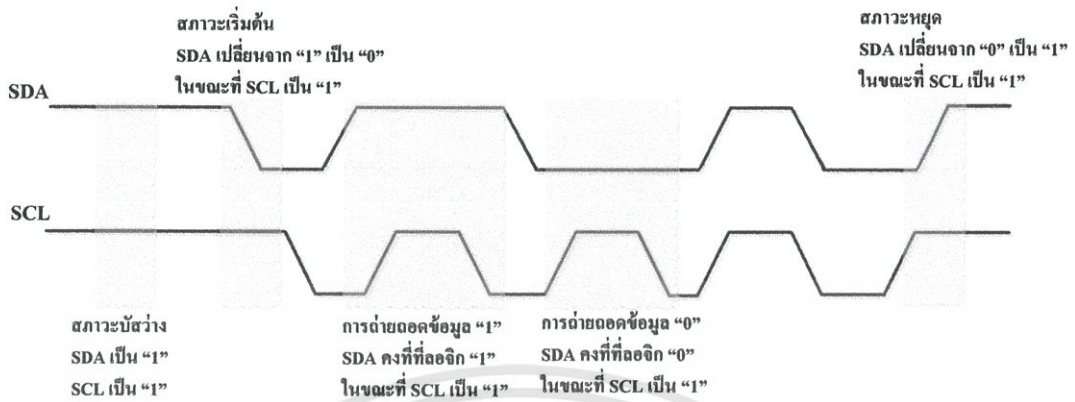
บัส I2C ประกอบด้วยสายสัญญาณ 2 เส้น คือ SDA และ SCL อุปกรณ์ต้องพ่วงบนบัสสามารถมีได้มากมาย ดังนั้นจึงต้องมีการกำหนดรูปการของการติดต่อบนบัส หรือเรียกว่า โพรโตคอล (Protocol) เพื่อให้ผู้ใช้งานทราบว่า ขณะนี้อุปกรณ์ใดติดต่อกันอยู่ และอุปกรณ์ตัวใดเป็นตัวรับหรือตัวส่ง ต่อไปนี้จะขออธิบายลักษณะ หน้าที่ และนิยามของอุปกรณ์ที่ต่ออยู่บนบัส I2C เพื่อเป็นข้อตกลงพื้นฐานก่อนที่จะอธิบายการทำงานของบัส I2C ต่อไปคือ อุปกรณ์ที่เป็นผู้รับข้อมูลหรือส่งข้อมูล เรียกว่า ตัวส่ง (Transmitter) และ อุปกรณ์ที่เป็นผู้รับข้อมูล เรียกว่า ตัวรับ (Receiver)

สำหรับอุปกรณ์บนระบบบัส I2C สามารถเป็นได้ทั้งตัวรับและตัวส่ง บางอุปกรณ์ทำหน้าที่เป็นตัวรับเพียงอย่างเดียว จะไม่มีอุปกรณ์ใดบนระบบบัส I2C ที่ทำหน้าที่เป็นตัวส่งเพียงตัวเดียว อุปกรณ์ที่ทำหน้าที่ควบคุมจังหวะการทำงานหรือการติดต่อบนระบบบัส I2C เรียกว่า มาสเตอร์ (Master) อุปกรณ์ที่ถูกควบคุมหรืออุปกรณ์ที่ต่อพ่วงเข้าไปบนระบบบัส I2C เรียกว่า สเลฟ (Slave)

ข้อกำหนดของการติดต่อบนระบบบัส I2C คือ การถ่ายทอดข้อมูลจะเกิดขึ้นเมื่อบัสว่างเท่านั้น และในระหว่างการถ่ายทอดข้อมูล เมื่อใดก็ตามที่สาย SCL มีสถานะเป็นลอจิกสูง สายข้อมูลต้องรักษาข้อมูลไว้อย่าให้เกิดการเปลี่ยนแปลงขึ้นเด็ดขาด มิฉะนั้น สัญญาณที่เกิดขึ้นจะได้รับการแปลความหมายเป็นสัญญาณควบคุมแทน

2.12.3 สภาวะที่เกิดขึ้นบนบัส I2C มีด้วยกัน 5 สภาวะ ดังนี้

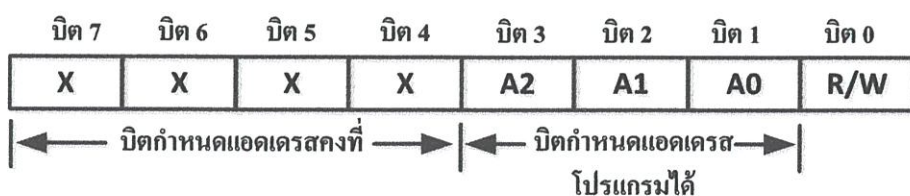
- 1) บัสว่าง (Bus Not Busy) สภาวะนี้เกิดขึ้นเมื่อสถานะลอจิกบนสาย SDA และ SCL เป็นลอจิกสูงทั้งคู่ นั่น หมายความว่า การถ่ายทอดข้อมูลสามารถเริ่มต้นขึ้นได้
- 2) เริ่มต้นการถ่ายทอดข้อมูล (Start Data Transfer) เกิดขึ้นเมื่อสาย SDA มีการเปลี่ยนแปลงระดับลอจิกจสูงไปลอจิกต่ำ ในขณะที่สาย SCL มีสถานะลอจิกสูงเรียกสภาวะที่เกิดขึ้นนี้ว่า สภาวะเริ่มต้น
- 3) ข้อมูลดำรงอยู่บนบัส (Data Valid) สภาวะนี้เกิดขึ้นถัดจากสภาวะเริ่มต้น โดยสถานะลอจิกที่เกิดขึ้นบนสาย SDA ก็คือข้อมูลที่กำลังถ่ายทอด เมื่อสาย SCL เป็นลอจิกสูง และสถานะที่สาย SDA ต้องคงที่ เพื่อให้อุปกรณ์รับรู้ข้อมูลในจังหวะนั้นว่าเป็น “0” หรือ “1” ข้อมูลอาจเกิดการเปลี่ยนแปลงได้ในขณะที่สาย SCL เป็นลอจิกต่ำ แต่เมื่อใดก็ตามที่ต้องการให้เกิดการถ่ายทอดข้อมูลอย่างสมบูรณ์ สถานะลอจิกที่ขา SDA ต้องคงที่ตลอดช่วงเวลาที่ยสาย SCL มีสถานะลอจิกสูง หากเกิดการเปลี่ยนแปลงสถานะลอจิกในขณะที่สาย SCL มีลอจิกสูงอยู่นั้น อุปกรณ์มาสเตอร์ที่ทำการควบคุมการถ่ายทอดข้อมูล จะแปลความหมายเป็นสภาวะหยุดหรือสภาวะเริ่มต้นก็ได้ ทำให้ข้อมูลที่ทำการถ่ายทอดนั้นเกิดความผิดพลาดขึ้น
- 4) รับรู้ข้อมูล (Acknowledge) เกิดขึ้นหลังจากที่การถ่ายทอดข้อมูลจากตัวส่งมายังตัวรับเกิดขึ้นอย่างสมบูรณ์ โดยตัวส่งจะทำการส่งข้อมูลมา 1 บิตเรียกว่า บิตรับรู้ (Acknowledge Bit) มีสถานะเป็นลอจิกสูงหลังจากส่งข้อมูลมาครบถ้วน ส่วนอุปกรณ์มาสเตอร์จะทำการส่งสัญญาณรับรู้พิเศษซึ่งสัมพันธ์กับสัญญาณนาฬิกา อุปกรณ์สเลฟที่ถูกอ้างถึงในการติดต่อ หรือกำลังติดต่ออยู่ในขณะนั้น ก็จะกำเนิดบิตรับรู้ที่มีสถานะลอจิกต่ำ เพื่อตอบสนองให้ทราบว่าได้รับข้อมูลในแต่ละไบต์เรียบร้อยแล้ว
- 5) หยุดการถ่ายทอดข้อมูล (Stop Data Transfer) จะเกิดขึ้นในขณะที่สาย SDA มีการเปลี่ยนแปลงระดับลอจิกต่ำไปลอจิกสูง และสาย SCL มีสถานะลอจิกสูง เรียกสภาวะที่เกิดขึ้นนี้ว่า สภาวะหยุด (STOP)



รูปที่ 2.7 สถานะการทำงานของระบบบัส I2C

จากรูปที่ 2.7 ก่อนที่จะเริ่มต้นการถ่ายโอนข้อมูลระหว่างอุปกรณ์ต่าง ๆ ที่ต่ออยู่บนบัส ต้องมีการอ้างถึงอุปกรณ์เสียก่อน โดยการอ้างถึงอุปกรณ์บนบัส I2C นั้นจะใช้การอ้างถึงแบบ 7 บิต (7-bit addressing) หรือ 10 บิต (10-bit addressing) ในกรณีที่มียุกรณ์ต่ออยู่บนบัสไม่มาก ใช้การอ้างถึงแบบ 7 บิตก็เพียงพอ แต่ถ้ามียุกรณ์ต่ออยู่บนบัสมากกว่า 127 แอดเดรส จำเป็นต้องใช้การอ้างถึงแบบ 10 บิต หลังจากที่ติดต่ออุปกรณ์แต่ละตัวได้เรียบร้อยแล้ว ก็จะเริ่มต้นการถ่ายโอนข้อมูลกันต่อไป

1) การอ้างถึงแบบ 7 บิต (7-bit addressing) ข้อมูลไบต์แรกที่เกิดขึ้นหลังจากสถานะเริ่มต้น คือข้อมูลที่ใช้ในการอ้างถึงอุปกรณ์ที่ต้องการติดต่อ โดยมีรูปแบบแสดงในรูปที่ 2.8 ใน 7 บิต บนรวมทั้งบิต MSB ด้วยจะเป็นข้อมูลแอดเดรสของอุปกรณ์สเลฟที่ต้องการติดต่อ โดยแบ่งเป็น บิตกำหนดแอดเดรสคงที่ (fixed address bit) จำนวน 4 บิต ซึ่งข้อมูลนี้ยูกรณ์แต่ละตัวจะถูกกำหนดมาจากผู้ผลิต ไม่สามารถเปลี่ยนแปลงแก้ไขได้ ถัดมาอีก 3 บิตเป็นบิตกำหนดแอดเดรสที่สามารถโปรแกรมได้ (programmable address bit) โดยผู้ใช้งานต้องกำหนดสถานะลอจิกให้แก่ขา A0-A2 ของอุปกรณ์ที่มีการเชื่อมต่อแบบระบบบัส I2C ส่วนในบิต LSB เป็นบิตที่ใช้กำหนดการอ่านหรือเขียนข้อมูลกับอุปกรณ์สเลฟตัวนั้น ๆ หากบิต LSB เป็น "0" หมายถึงต้องเขียนข้อมูลไปยังอุปกรณ์นั้น ถ้าเป็น "1" จะเป็นการอ่านข้อมูลจากอุปกรณ์สเลฟ



รูปที่ 2.8 การกำหนดบิตแอดเดรสแบบ 7 บิต

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ข้อมูลในไบต์ต่อมาคือ ข้อมูลควบคุม (Control Byte) ในอุปกรณ์แต่ละตัวมีการกำหนดข้อมูลควบคุมที่แตกต่างกันไป ยกตัวอย่าง ไอซีขยายพอร์ตมีข้อมูลควบคุมที่ใช้กำหนดว่า บิตใดเป็นอินพุต บิตใดเป็นเอาต์พุต ในขณะที่ไอซี ADC/DAC ต้องการข้อมูลควบคุมเพื่อกำหนดให้ทำงานเป็นวงจรร ADC หรือ DAC เป็นต้น ข้อมูลในไบต์ต่อมาคือ ข้อมูลที่ทำการถ่ายทอดจริง

หลังจากที่มีการถ่ายทอดข้อมูลในแต่ละไบต์ อุปกรณ์สเลฟที่ได้รับการติดต่อต้องส่งสัญญาณรับรู้ตอบกลับมาด้วยทุกครั้ง ในรูปที่ 2.9 แสดงรูปแบบข้อมูลอนุกรมที่เกิดขึ้นในการติดต่อบนบัส I2C ของการอ้างถึงแบบ 7 บิต



รูปที่ 2.9 ข้อมูลอนุกรมที่เกิดขึ้นในการติดต่อบนบัส I2C ของการอ้างถึงแบบ 7 บิต

2) การอ้างถึงแบบ 10 บิต (10-bit addressing) การอ้างถึงแบบ 10 บิต จะมีข้อมูลเพิ่มเติมขึ้นมาเล็กน้อย โดยในไบต์แรกหลังจากเกิดสถานะเริ่มต้นต้องกำหนดให้ 5 บิตบนมีข้อมูลเป็น 11110 ส่วนอีก 2 บิตถัดมาเป็นบิตแอดเดรสของอุปกรณ์ที่ต้องการติดต่อ ในบิต LSB ของข้อมูลไบต์แรกยังคงเป็นการกำหนดว่า ต้องการอ่านหรือเขียนข้อมูลกับอุปกรณ์สเลฟตัวที่ต้องการติดต่อด้วย ข้อมูลไบต์ต่อมาเป็นข้อมูลแอดเดรสในไบต์ที่ 2 ของอุปกรณ์ที่ต้องการติดต่อด้วย ข้อมูลไบต์ถัดไปจึงเป็นข้อมูลควบคุม หลังจากนั้นก็จะจะเป็นข้อมูลจริงที่ใช้ในการติดต่อ

เช่นเดียวกับการอ้างถึงแบบ 7 บิต หลังจากถ่ายทอดข้อมูลครบทุกไบต์ ต้องมีสถานะรับรู้เกิดขึ้น เพื่อให้กระบวนการถ่ายทอดข้อมูลสามารถดำเนินต่อไปได้ ในรูปที่ 2.10 แสดงรูปแบบข้อมูลอนุกรมของการอ้างถึงแบบ 10 บิต



รูปที่ 2.10 แสดงรูปแบบข้อมูลอนุกรมของการอ้างถึงแบบ 10 บิต

2.13 มาตรฐานการเชื่อมต่อแบบอนุกรม RS-485

RS-485 เป็นหนึ่งในมาตรฐานการสื่อสารแบบอนุกรม (Serial Communication) ที่ถูกกำหนดขึ้นมาโดยกลุ่มพันธมิตรอุตสาหกรรมอิเล็กทรอนิกส์ในประเทศสหรัฐอเมริกา (Electronic Industries Alliance: EIA) เพื่อตอบสนองต่อความต้องการใช้งานที่ต้องการเชื่อมต่ออุปกรณ์หลาย ๆ ตัว บนสายสัญญาณที่มีแบบ 2 สายและแบบ 4 สาย โดยมีระยะทางการสื่อสารที่ไกลขึ้น และมีความเร็วรับส่งข้อมูลที่สูงขึ้น เมื่อเทียบกับมาตรฐานการสื่อสาร RS-232 ที่ถูกกำหนดขึ้นมาโดยกลุ่มพันธมิตรอุตสาหกรรมอิเล็กทรอนิกส์เช่นเดียวกัน

เหตุที่การสื่อสาร RS-485 สามารถรับส่งสัญญาณข้อมูลได้ไกลขึ้นและเร็วขึ้นนั้น เป็นเพราะ RS-485 ใช้เทคนิคสัญญาณรับส่งแบบดิฟเฟอเรนเชียล (Differential Mode) ขณะที่ RS-232 ใช้เทคนิคสัญญาณรับส่งแบบคอมมอน (Common Mode) สัญญาณรับส่งแบบคอมมอนนั้นจะใช้สัญญาณกราวด์ (Ground Signal) เป็นตัวเปรียบเทียบปัญหาจะเกิดขึ้น เมื่อระดับสัญญาณกราวด์ของตัวรับและตัวส่งไม่เท่ากัน ยิ่งระดับสัญญาณแตกต่างกันมากเท่าไร ก็ยิ่งมีผลต่อความผิดพลาดในการสื่อสารมากขึ้นเท่านั้น เพราะการตีความข้อมูลที่รับเข้ามาว่าเป็นศูนย์หรือหนึ่งนั้น จะดูจากระดับความแตกต่างระหว่างสัญญาณกราวด์กับสัญญาณข้อมูลที่รับเข้ามา และยิ่งเมื่อมีสัญญาณรบกวนสอดแทรกเข้ามาในสายสัญญาณมากเท่าไร ก็ยิ่งทำให้การตีความสัญญาณมีโอกาสผิดพลาดสูงมากยิ่งขึ้น พุดง่าย ๆ ก็คือระดับความผิดพลาดในการรับส่งข้อมูลจะสูงขึ้นจนไม่สามารถสื่อสารกันได้ หรือต้องลดระดับความเร็วในการสื่อสารลงมา

ขณะที่สัญญาณรับส่งแบบดิฟเฟอเรนเชียลนั้น จะไม่ใช่สัญญาณกราวด์เป็นระดับอ้างอิงหรือเปรียบเทียบ แต่จะดูที่ความต่างของสัญญาณของคู่สายเป็นสำคัญ ทำให้ปัญหาเรื่องความแตกต่างของสัญญาณกราวด์ระหว่างอุปกรณ์รับและอุปกรณ์ส่งหมดไป (เพราะไม่ถูกนำมาใช้อ้างอิง) และถ้าหากคู่สายที่ใช้รับส่งสัญญาณพันกันเป็นเกรียว ก็จะไม่ส่งผลให้สัญญาณรบกวนจากภายนอกที่สอดแทรกเข้ามาในคู่สายถูกกำจัดออกไปได้โดยง่าย เป็นผลให้การสื่อสาร RS-485 ทนต่อสัญญาณรบกวน

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ภายนอกได้สูง สามารถรับส่งสัญญาณได้ไกลขึ้น และเร็วขึ้นเมื่อเปรียบเทียบกับเทคนิคสัญญาณรับส่งแบบคอมมอน

การเชื่อมต่ออุปกรณ์ RS-485 เป็นเครือข่าย (network) ได้เป็นอีกหนึ่งเหตุผลที่ทำให้มาตรฐานการสื่อสาร RS-485 เป็นที่นิยมนำมาใช้งานในงานควบคุมและตรวจรู้ อุปกรณ์รับส่งสัญญาณ RS-485 โดยทั่วไปนั้น จะสามารถเชื่อมต่ออุปกรณ์บนเครือข่ายได้ 32 อุปกรณ์ ถ้าความต้านทานขาเข้า (Input Resistance) ของอุปกรณ์ดังกล่าวมีค่าอยู่ที่ 12 K Ω ปัจจุบันมีอุปกรณ์รับส่งสัญญาณ RS-485 ที่มีความต้านทานสูงสามารถเชื่อมต่ออุปกรณ์บนเครือข่ายเดียวกันได้ถึง 256 อุปกรณ์ ด้วยอุปกรณ์ทวนสัญญาณ RS-485 เราสามารถเพิ่มจำนวนอุปกรณ์บนเครือข่ายได้มาก

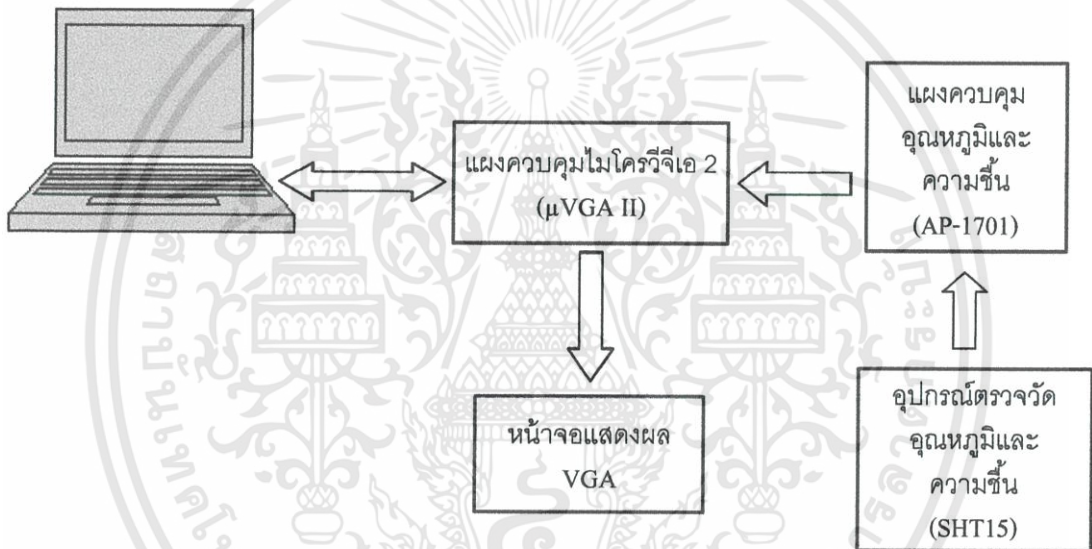


บทที่ 3

การออกแบบและการทดลอง

3.1 ศึกษาการทำงานของอุปกรณ์ตรวจวัดอุณหภูมิและความชื้น และแผงควบคุม ไมโครวีจีเอ 2

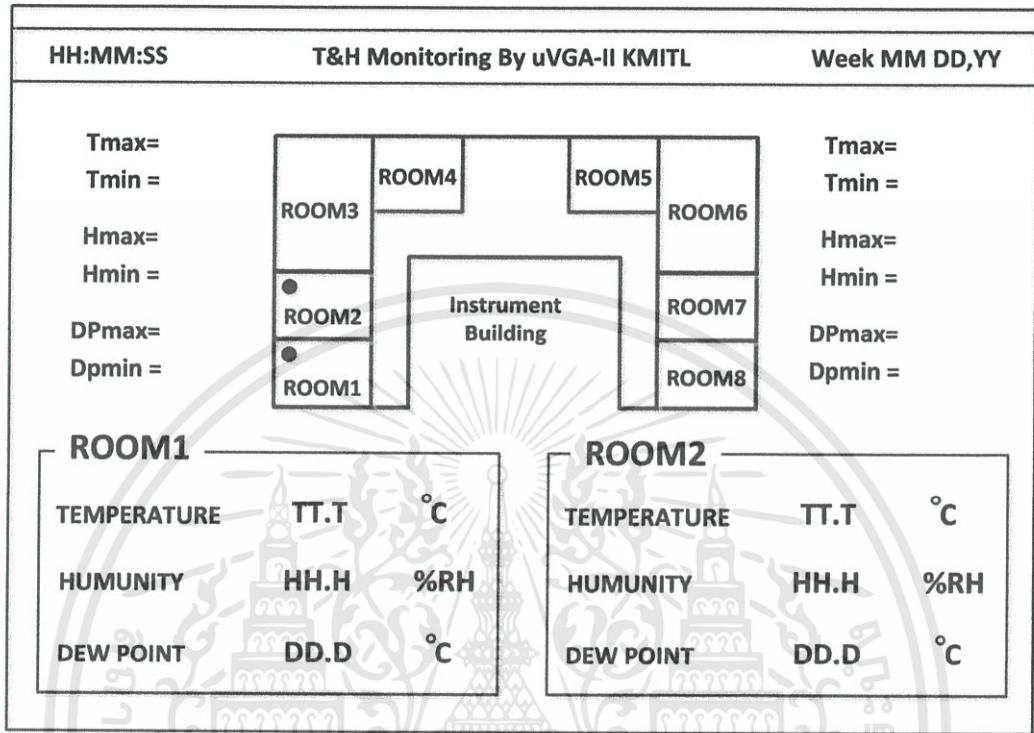
จากข้อมูลเอกสารและทฤษฎีที่เกี่ยวข้องในบทที่ 2 สามารถที่จะสรุปการทำงานของอุปกรณ์ตรวจวัดอุณหภูมิและความชื้น (SHT15) โดยการประมวลผลของแผงควบคุมไมโครวีจีเอ 2 กับแผงควบคุมอุณหภูมิและความชื้น (AP-1701) ได้ดังนี้



รูปที่ 3.1 แผนผังการตรวจวัดอุณหภูมิและความชื้นแสดงผลหน้าจอ VGA

จากรูป 3.1 เมื่อทำการตรวจวัดอุณหภูมิและความชื้น ด้วยอุปกรณ์ตรวจวัดอุณหภูมิและความชื้น (SHT15) ซึ่งอุปกรณ์ดังกล่าวจะทำการแปลงสัญญาณอนาล็อกที่อ่านค่าได้ เป็นสัญญาณดิจิทัลส่งข้อมูลผ่านแผงควบคุมอุณหภูมิและความชื้น (AP-1701) ทำการประมวลผลข้อมูล แล้วทำการติดต่อสื่อสารข้อมูลแบบอนุกรม RS-485 กับแผงควบคุมไมโครวีจีเอ 2 จากนั้น แผงควบคุมไมโครวีจีเอ 2 ก็จะส่งข้อมูลของอุณหภูมิและความชื้นไปแสดงผลที่จอ VGA ซึ่งแสดงผลทั้งหมดบนหน้าจอ VGA จะถูกจัดการด้วยโปรแกรมกราฟฟิก 4DGL

3.2 ออกแบบการแสดงผลหน้าจอ VGA



รูปที่ 3.2 ภาพหน้าจอแสดงผล อุณหภูมิและความชื้น

จากรูปที่ 3.2 เป็นรูปแบบการแสดงผลอุณหภูมิ ความชื้นและอุณหภูมิจุดน้ำค้าง พร้อมแสดงจุดปริบการเปลี่ยนแปลงอุณหภูมิของแต่ละอุปกรณ์ตรวจจำนวน 2 ตัวที่แผนผังของ Instrument Building (ROOM1 และ ROOM2) โดยมีรายละเอียดดังนี้

| | | |
|---------------|----------|-----------------------------------------------------------------------------------------------|
| HH:MM:SS | แสดงเวลา | ชั่วโมง:นาที:วินาที |
| Week MM DD,YY | แสดง | วันของสัปดาห์ เดือน วัน,ปี |
| Tmax | แสดง | อุณหภูมิสูงสุด |
| Tmin | แสดง | อุณหภูมิต่ำสุด |
| Hmax | แสดง | ความชื้นสูงสุด |
| Hmin | แสดง | ความชื้นต่ำสุด |
| DPmax | แสดง | อุณหภูมิจุดน้ำค้างสูงสุด |
| DPmin | แสดง | อุณหภูมิจุดน้ำค้างต่ำสุด |
| ROOM1 | แสดง | อุณหภูมิ (TT.T) ความชื้น (HH.H) และอุณหภูมิจุดน้ำค้าง (DD.D) ห้องที่ 1 จากอุปกรณ์ตรวจตัวที่ 1 |

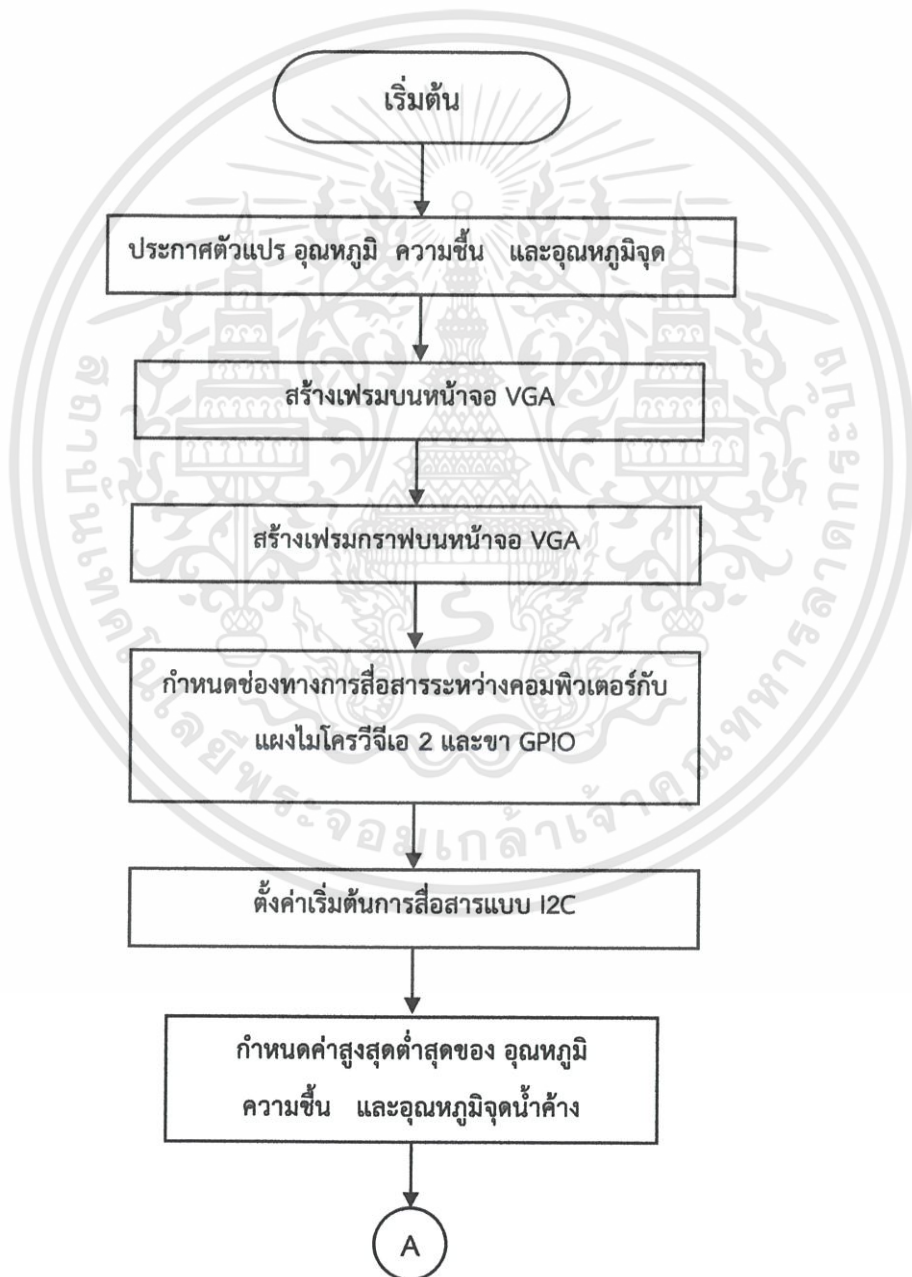
เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ROOM2

แสดง อุณหภูมิ (TT.T) ความชื้น (HH.H) และอุณหภูมิจุด
น้ำค้าง (DD.D) ห้องที่ 1 จากอุปกรณ์ตรวจรู้ตัวที่ 2

3.3 เขียนโปรแกรมการแสดงผลหน้าจอ VGA จากคอมพิวเตอร์ส่วนบุคคลผ่านแผงควบคุมไมโครวีจีเอ 2

3.3.1 เขียนผังงาน (Flowcharts)



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.3 แผนผังการเขียนโปรแกรมแสดงผลการวัดอุณหภูมิและความชื้น

3.3.2 เขียนโปรแกรมโดยใช้ 4D Workshop4 IDE

1) ประกาศตัวแปร ค่าคงที่ เพื่อใช้ในการแสดงผล อุณหภูมิ ความชื้น และอุณหภูมิจุดน้ำค้าง

โปรแกรมที่ 1.1 การประกาศตัวแปร ค่าคงที่

```
#platform "uVGA-II"
#inherit "4DGL_16bitColours.fnc"
#constant SECTORSIZE 512
#constant SECONDS 0
#constant MINUTES 1
#constant HOURS 2
#constant DAYS 3
#constant DATES 4
#constant MONTHS 5
#constant YEARS 6
#constant DS1307 0xD0
#constant WR 1
#DATA
byte values 69,75,79,82,84,86,88,89,91,92,93,94,95,96,96,97,98,98,99,100
#END
var TEMP_MAX;
var TEMP_MIN;
var TEMP_MAX2;
var TEMP_MIN2;
var HUM_MAX;
var HUM_MIN;
var HUM_MAX2; HUM_MIN2;
var DP_MAX;
var DP_MIN;
var DP_MAX2; DP_MIN2;
```

โปรแกรมที่ 1.1 การประกาศตัวแปร ค่าคงที่ (ต่อ)

```
var buf[100];
var buf2[100];
var n;
var seconds;
var address;
var r;
```

2) เขียนฟังก์ชัน (User Defined Function)

โปรแกรมที่ 1.2 ฟังก์ชันพื้นที่การแสดงผลอุณหภูมิ ความชื้น และอุณหภูมิจุดน้ำค้าง

```
func CreateFrame()
    gfx_RectangleFilled(0,0,639,15,WHITE);
    gfx_Rectangle(1,50,639,50,WHITE);
    txt_MoveCursor(2,1);
    txt_Set(TEXT_COLOUR,LIGHTBLUE);txt_Set(TEXT_HEIGHT,2);
    print("          T&H Monitoring By uVGA-II KMITL ");
endfunc
```

โปรแกรมที่ 1.3 ฟังก์ชันพื้นที่การแสดงตัวหนังสือ หมายเลขห้อง ชื่ออาคาร

```
func CreateText()
    txt_MoveCursor(7,34);
    txt_Set(TEXT_COLOUR,LIGHTBLUE);
    txt_Height(1);
    txt_Width(1);
    print(" Instrument");
    txt_MoveCursor(17,36);
    txt_Set(TEXT_COLOUR,LIGHTBLUE);
    txt_Height(1);
    txt_Width(1);
    print("building");
```

โปรแกรมที่ 1.3 ฟังก์ชันพื้นที่การแสดงตัวหนังสือ หมายเลขห้อง ชื่ออาคาร (ต่อ)

```
//-----ROOM 1-----
txt_MoveCursor(20,25);
txt_Set(TEXT_COLOUR,WHITE);
txt_Height(1);
txt_Width(1);
print("ROOM 1");

//-----ROOM 2-----
txt_MoveCursor(16,25);
txt_Set(TEXT_COLOUR,WHITE);
txt_Height(1);
txt_Width(1);
print("ROOM 2");

//-----ROOM 3-----
txt_MoveCursor(10,25);
txt_Set(TEXT_COLOUR,WHITE);
txt_Height(1);
txt_Width(1);
print("ROOM 3");

//-----ROOM 4-----
txt_MoveCursor(8,31);
txt_Set(TEXT_COLOUR,WHITE);
txt_Height(1);
txt_Width(1);
print("ROOM 4");

//-----ROOM 5-----
txt_MoveCursor(8,44);
txt_Set(TEXT_COLOUR,WHITE);
txt_Height(1);
txt_Width(1);
print("ROOM 5");
```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ 1.3 ฟังก์ชันพื้นที่การแสดงตัวหนังสือ หมายเลขห้อง ชื่ออาคาร (ต่อ)

```
//-----ROOM 6-----
txt_MoveCursor(10,50);
txt_Set(TEXT_COLOUR,WHITE);
txt_Height(1);
txt_Width(1);
print("ROOM 6");

//-----ROOM 7-----
txt_MoveCursor(16,50);
txt_Set(TEXT_COLOUR,WHITE);
txt_Height(1);
txt_Width(1);
print("ROOM 7");

//-----ROOM 8-----
txt_MoveCursor(20,50);
txt_Set(TEXT_COLOUR,WHITE);
txt_Height(1); txt_Width(1);
print("ROOM 8");

endfunc
```

โปรแกรมที่ 1.4 ฟังก์ชันการแสดงผลภาพตัวอาคาร

```
func CreateBuilding()
gfx_Rectangle(195,70,245,170,GREEN);
gfx_Rectangle(195,170,245,220,GREEN);
gfx_Rectangle(195,220,245,270,GREEN);
gfx_Rectangle(245,70,295,120,GREEN);

gfx_Line(295,70,345,70,GREEN);

gfx_Rectangle(395,70,445,170,GREEN);
gfx_Rectangle(395,170,445,220,GREEN);
```

โปรแกรมที่ 1.4 ฟังก์ชันการแสดงผลภาพตัวอาคาร (ต่อ)

```
gfx_Rectangle(395,220,445,270,GREEN);
gfx_Rectangle(345,70,395,120,GREEN);
gfx_Line(245,270,270,270,GREEN);
gfx_Line(270,270,270,145,GREEN);
gfx_Line(270,145,370,145,GREEN);
gfx_Line(370,270,370,145,GREEN);
gfx_Line(370,270,395,270,GREEN);
```

```
endfunc
```

โปรแกรมที่ 1.5 ฟังก์ชันการสร้างตารางแสดงค่า

```
func CreateTable()
  gfx_Rectangle(50,300,310,470,WHITE);
  gfx_Rectangle(330,300,590,470,WHITE);
  //-----Text1-----
  txt_MoveCursor(14,9);
  txt_Height(2);
  txt_Width(1);
  txt_Set(TEXT_COLOUR,LIGHTBLUE);
  print("TEMPERATURE");
  txt_MoveCursor(16,9);
  txt_Height(2);txt_Width(1);
  txt_Set(TEXT_COLOUR,YELLOW);
  print("HUMUNITY");
  txt_MoveCursor(18,9);
  txt_Height(2);txt_Width(1);
  txt_Set(TEXT_COLOUR,RED);
  print("DEW POINT");
  //-----Text2-----
  txt_MoveCursor(14,44);
```

โปรแกรมที่ 1.5 ฟังก์ชันแสดงการสร้างตารางแสดงค่า (ต่อ)

```

txt_Height(2);
txt_Width(1);
txt_Set(TEXT_COLOUR,LIGHTBLUE);
print("TEMPERATURE");
txt_MoveCursor(16,44);
txt_Height(2);
txt_Width(1);
txt_Set(TEXT_COLOUR,YELLOW);
print("HUMUNITY");
txt_MoveCursor(18,44);
txt_Height(2);
txt_Width(1);
txt_Set(TEXT_COLOUR,RED);
print("DEW POINT");

//-----HEADER-----
txt_MoveCursor(12,7);
txt_Height(2);
txt_Width(1);
txt_Set(TEXT_COLOUR,WHITE);
print("ROOM 1");
txt_MoveCursor(12,42);
txt_Height(2);
txt_Width(1);
txt_Set(TEXT_COLOUR,WHITE);
print("ROOM 2");
endfunc

```

โปรแกรมที่ 1.6 ฟังก์ชันการสื่อสารแบบ I2C

```

/*
    I2C_Open("speed");        // Configure the I2C module, speed can be
                               //I2C_SLOW, I2C_MED, I2C_FAST (100khz, 400khz, 1mhz)
    I2C_Close();              // Disable the I2C1 module.
    I2C_Start();              // Generate a Start condition.
    I2C_Stop();               // Generate a Stop condition.
    I2C_Restart();            // Generate a Restart condition.
    I2C_Read();               // Read a single byte from the I2C Bus.
    I2C_Write("byte");        // Write a byte to the I2C bus, Returns TRUE if ok
    I2C_Ack();                // Generate the acknowledge condition.
    I2C_Nack();               // Generate the negative acknowledge condition.
    I2C_DataRdy();            // Return TRUE if the I2C receive buffer is full.
    I2C_Idle();               // Wait until the I2C Bus is Inactive.
    I2C_Gets("buffer", "length"); // Read length bytes from the I2C device to a buffer
    I2C_Puts("buffer");       // Write an ascii string to the I2C device
*/

func I2C_Initialize()
    var c;
    I2C_Open(I2C_SLOW);        // Configure speed the I2C module
    c:=ReadRTC(0) & 0x7F;      // ensure CH bit is clear else clock wont run
    WriteRTC(0, c);
    WriteRTC(7, 0x10);        // enable square wave output
    pause(1000);
endfunc

```

โปรแกรมที่ 1.7 ฟังก์ชันอ่านค่าสัญญาณนาฬิกา (Real Time Clock : ReadRTC)

```

func ReadRTC(var addr)
    I2C_Start();                // Generate Start condition
    I2C_Write(DS1307);         // send slave address
    I2C_Write(LObyte(addr));   // Send register address
    I2C_Restart();             // Generate Restart
    I2C_Write(DS1307+WR);     // send control byte for Read
    r := I2C_Read();           // read the byte
    I2C_Nack();                // finished reading, send Not Ack
    I2C_Stop();                // Send Stop Condition
    return r;                  // return the register value
endfunc

func WriteRTC(var register, var value)
    I2C_Start();                // Generate Start condition
    I2C_Write(DS1307);         // send slave address
    I2C_Write(register);       // select the register
    I2C_Write(value);          // save the value in selected register
    I2C_Stop();                // finished with bus
endfunc

```

โปรแกรมที่ 1.8 ฟังก์ชันแสดงผลวัน เวลา

```

func DisplayDate()
    var day,month;
    txt_MoveCursor(2,66);
    txt_Set(TEXT_COLOUR,WHITE);
    txt_Set(TEXT_HEIGHT,2);
    day := ReadRTC(DAYS);
    if (day == 1) print ("SUN ");
    if (day == 2) print ("MON ");
    if (day == 3) print ("TUE ");
    if (day == 4) print ("WED ");

```

โปรแกรมที่ 1.8 ฟังก์ชันแสดงผลวัน เวลา (ต่อ)

```

if (day == 5) print ("THU ");
if (day == 6) print ("FRI ");
if (day == 7) print ("SAT ");
month := ReadRTC(MONTHS);
if (month == 1) print ("JAN ");
if (month == 2) print ("FEB ");
if (month == 3) print ("MAR ");
if (month == 4) print ("APR ");
if (month == 5) print ("MAY ");
if (month == 6) print ("JUN ");
if (month == 7) print ("JUL ");
if (month == 8) print ("AUG ");
if (month == 9) print ("SEP ");
if (month == 16) print ("OCT ");
if (month == 17) print ("NOV ");
if (month == 18) print ("DEC ");
print( [HEX2] ReadRTC(DATES)," ",
      [HEX2] ReadRTC(YEARS) );
txt_Height(1);txt_Width(1);
endfunc

```

โปรแกรมที่ 1.9 ฟังก์ชัน แสดงผล อุณหภูมิ (TEMP) ความชื้น (HUM) และอุณหภูมิจุดน้ำค้าง (DPT) บนหน้าจอ VGA จากอุปกรณ์ตรวจรู้ตัวที่ 1

```

func DisplayTEMP(var Temp, var Tempfrac)
  txt_MoveCursor(28,25);
  txt_Height(2);
  txt_Width(1);
  txt_Set(TEXT_COLOUR,LIGHTBLUE);
  Temp := ABS(Temp);
  if (Temp < 10)

```

โปรแกรมที่ 1.9 ฟังก์ชัน แสดงผล อุณหภูมิ (TEMP) ความชื้น (HUM) และอุณหภูมิจุดน้ำค้าง (DPT) บนหน้าจอ VGA จากอุปกรณ์ตรวจวัดตัวที่ 1 (ต่อ)

```

    print (" ",Temp);
else
    if (Temp < 100)
        print (" ",Temp);
    else
        print(Temp);
endif
endif
txt_Height(2);
txt_Width(1);
print(".",Tempfrac);
txt_MoveCursor(14,33);
txt_Height(1);
txt_Width(1);
print("O");
txt_Height(2);
txt_Width(1);
print("C");
if ((Temp*10)+Tempfrac > TEMP_MAX) TEMP_MAX := (Temp*10)+Tempfrac;
if ((Temp*10)+Tempfrac < TEMP_MIN) TEMP_MIN := (Temp*10)+Tempfrac;
if (TEMP_MAX >= 999) TEMP_MAX := 999;
if (TEMP_MIN < 0) TEMP_MIN := 0;
txt_Height(2);txt_Width(1);
txt_MoveCursor(3,9);
print("Tmax=",TEMP_MAX/10,".",TEMP_MAX % 10);
txt_MoveCursor(4,9);
print("Tmin=",TEMP_MIN/10,".",TEMP_MIN % 10);
txt_Height(1);txt_Width(1);
endfunc

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ 1.9 ฟังก์ชัน แสดงผล อุณหภูมิ (TEMP) ความชื้น (HUM) และอุณหภูมิจุดน้ำค้าง (DPT) บนหน้าจอ VGA จากอุปกรณ์ตรวจวัดตัวที่ 1 (ต่อ)

```

func DisplayHUM(var HUM, var Humfrac)
    txt_MoveCursor(32,25);
    txt_Height(2);
    txt_Width(1);
    txt_Set(TEXT_COLOUR,YELLOW);
    HUM := ABS(HUM);
    if (HUM < 10)
    print (" ",HUM);
    else
    if (HUM < 100) print (" ",HUM);
    endif
    txt_Height(2);
    txt_Width(1);
    print(".",Humfrac);
    txt_MoveCursor(16,33);
    txt_Height(2);
    txt_Width(1);
    print("%RH");
    if ((HUM*10)+Humfrac > HUM_MAX) HUM_MAX := (HUM*10)+Humfrac;
    if ((HUM*10)+Humfrac < HUM_MIN) HUM_MIN := (HUM*10)+Humfrac;
    if (HUM_MAX >= 999) HUM_MAX := 999;
    if (HUM_MIN < 0) HUM_MIN := 0;
    txt_Height(2);txt_Width(1);
    txt_MoveCursor(6,9);
    print("Hmax=",HUM_MAX/10,".",HUM_MAX % 10);
    txt_MoveCursor(7,9);
    print("Hmin=",HUM_MIN/10,".",HUM_MIN % 10);
    txt_Height(1);txt_Width(1);
endfunc

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ 1.9 ฟังก์ชัน แสดงผล อุณหภูมิ (TEMP) ความชื้น (HUM) และอุณหภูมิจุดน้ำค้าง (DPT) บนหน้าจอ VGA จากอุปกรณ์ตรวจรู้ตัวที่ 1 (ต่อ)

```
func DisplayDPT(var DPT, var DPTfrac)
    txt_MoveCursor(36,25);
    txt_Height(2);
    txt_Width(1);
    txt_Set(TEXT_COLOUR,RED);
    DPT := ABS(DPT);
    if (DPT < 10)
    print (" ",DPT);
    else
    if (DPT < 100) print (" ",DPT);
    endif
    txt_Height(2);
    txt_Width(1);
    print(".",DPTfrac);
    txt_MoveCursor(18,33);
    txt_Height(1);
    txt_Width(1);
    print("O");
    txt_Height(2);
    txt_Width(1);
    print("C");
    if ((DPT*10)+DPTfrac > DP_MAX) DP_MAX := (DPT*10)+DPTfrac;
    if ((DPT*10)+DPTfrac < DP_MIN) DP_MIN := (DPT*10)+DPTfrac;
    if (DP_MAX >= 999) DP_MAX := 999;
    if (DP_MIN < 0) DP_MIN := 0;
    txt_Height(2);txt_Width(1);
    txt_MoveCursor(9,9);
    print("DPmax=",DP_MAX/10,".",DP_MAX % 10);
```

โปรแกรมที่ 1.9 ฟังก์ชัน แสดงผล อุณหภูมิ (TEMP) ความชื้น (HUM) และอุณหภูมิจุดน้ำค้าง (DPT) บนหน้าจอ VGA จากอุปกรณ์ตรวจรู้ตัวที่ 1 (ต่อ)

```

txt_MoveCursor(10,9);
print("DPmin=",DP_MIN/10,".",DP_MIN % 10);
txt_Height(1);txt_Width(1);
endfunc
func DisplayDPT2(var DPT2, var DPTfrac2)
txt_MoveCursor(36,60);
txt_Height(2);
txt_Width(1);
txt_Set(TEXT_COLOUR,RED);
DPT2 := ABS(DPT2);
if (DPT2 < 10)
print (" ",DPT2);
else
if (DPT2 < 100) print (" ",DPT2);
endif
txt_Height(2);
txt_Width(1);
print(".",DPTfrac2);
txt_MoveCursor(18,68);
txt_Height(1);
txt_Width(1);
print("O");
txt_Height(2);
txt_Width(1);
print("C");
if ((DPT2*10)+DPTfrac2 > DP_MAX2) DP_MAX2 := (DPT2*10)+DPTfrac2;
if ((DPT2*10)+DPTfrac2 < DP_MIN2) DP_MIN2 := (DPT2*10)+DPTfrac2;
if (DP_MAX2 >= 999) DP_MAX2 := 999;
if (DP_MIN2 < 0) DP_MIN2 := 0;

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ 1.9 ฟังก์ชัน แสดงผล อุณหภูมิ (TEMP) ความชื้น (HUM) และอุณหภูมิจุดน้ำค้าง (DPT) บนหน้าจอ VGA จากอุปกรณ์ตรวจรู้ตัวที่ 1 (ต่อ)

```

txt_Height(2);
txt_Width(1);
txt_MoveCursor(9,61);
print("DPmax=",DP_MAX2/10,".",DP_MAX2 % 10);
txt_MoveCursor(10,61);
print("DPmin=",DP_MIN2/10,".",DP_MIN2 % 10);
txt_Height(1);
txt_Width(1);
endfunc

```

โปรแกรมที่ 1.10 ฟังก์ชัน แสดงผล อุณหภูมิ (TEMP) ความชื้น (HUM) และอุณหภูมิจุดน้ำค้าง (DPT) บนหน้าจอ VGA จากอุปกรณ์ตรวจรู้ตัวที่ 2

```

func DisplayTEMP2(var Temp2, var Tempfrac2)
txt_MoveCursor(28,60);
txt_Height(2);
txt_Width(1);
txt_Set(TEXT_COLOUR,LIGHTBLUE);
Temp2 := ABS(Temp2);
if (Temp2 < 10)
print (" ",Temp2);
else
if (Temp2 < 100)
print (" ",Temp2);
else
print(Temp2);
endif
endif
txt_Height(2);
txt_Width(1);

```

โปรแกรมที่ 1.10 ฟังก์ชัน แสดงผล อุณหภูมิ (TEMP) ความชื้น (HUM) และอุณหภูมิจุดน้ำค้าง (DPT) บนหน้าจอ VGA จากอุปกรณ์ตรวจรู้ตัวที่ 2 (ต่อ)

```

    print(".",Tempfrac2);
    txt_MoveCursor(14,68);
    txt_Height(1);
    txt_Width(1);
    print("O");
    txt_Height(2);txt_Width(1);
    print("C");
    if ((Temp2*10)+Tempfrac2 > TEMP_MAX2) TEMP_MAX2 := (Temp2*10)+Tempfrac2;
    if ((Temp2*10)+Tempfrac2 < TEMP_MIN2) TEMP_MIN2 := (Temp2*10)+Tempfrac2;
    if (TEMP_MAX2 >= 999) TEMP_MAX2 := 999;
    if (TEMP_MIN2 < 0) TEMP_MIN2 := 0;
    txt_Height(2);
    txt_Width(1);
    txt_MoveCursor(3,61);
    print("Tmax=",TEMP_MAX2/10,".",TEMP_MAX2 % 10);
    txt_MoveCursor(4,61);
    print("Tmin=",TEMP_MIN2/10,".",TEMP_MIN2 % 10);
    txt_Height(1);
    txt_Width(1);
endfunc
func DisplayHUM2(var HUM2, var Humfrac2)
    txt_MoveCursor(32,60);
    txt_Height(2);
    txt_Width(1);
    txt_Set(TEXT_COLOUR,YELLOW);
    HUM2 := ABS(HUM2);
    if (HUM2 < 10)
    print (" ",HUM2);
    else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ 1.10 ฟังก์ชัน แสดงผล อุณหภูมิ (TEMP) ความชื้น (HUM) และอุณหภูมิจุดน้ำค้าง (DPT) บนหน้าจอ VGA จากอุปกรณ์ตรวจวัดตัวที่ 2 (ต่อ)

```

    if (HUM2 < 100) print (" ",HUM2);
    endif
    txt_Height(2);txt_Width(1);
    print(".",Humfrac2);
    txt_MoveCursor(16,68);
    txt_Height(2);txt_Width(1);
    print("%RH");
    if ((HUM2*10)+Humfrac2 > HUM_MAX2) HUM_MAX2 := (HUM2*10)+Humfrac2;
    if ((HUM2*10)+Humfrac2 < HUM_MIN2) HUM_MIN2 := (HUM2*10)+Humfrac2;
    if (HUM_MAX2 >= 999) HUM_MAX2 := 999;
    if (HUM_MIN2 < 0) HUM_MIN2 := 0;
    txt_Height(2);
    txt_Width(1);
    txt_MoveCursor(6,61);
    print("Hmax=",HUM_MAX2/10,".",HUM_MAX2 % 10);
    txt_MoveCursor(7,61);
    print("Hmin=",HUM_MIN2/10,".",HUM_MIN2 % 10);
    txt_Height(1);
    txt_Width(1);
endfunc
func DisplayDPT2(var DPT2, var DPTfrac2)
    txt_MoveCursor(36,60);
    txt_Height(2);
    txt_Width(1);
    txt_Set(TEXT_COLOUR,RED);
    DPT2 := ABS(DPT2);
    if (DPT2 < 10)
    print (" ",DPT2);
    else

```

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า
ไม่ว่ากรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

โปรแกรมที่ 1.10 ฟังก์ชัน แสดงผล อุณหภูมิ (TEMP) ความชื้น (HUM) และอุณหภูมิจุดน้ำค้าง (DPT) บนหน้าจอ VGA จากอุปกรณ์ตรวจรู้ตัวที่ 2 (ต่อ)

```

if (DPT2 < 100) print (" ",DPT2);
endif
txt_Height(2);txt_Width(1);
print(".",DPTfrac2);
txt_MoveCursor(18,68);
txt_Height(1);txt_Width(1);
print("O");
txt_Height(2);
txt_Width(1);
print("C");
if ((DPT2*10)+DPTfrac2 > DP_MAX2) DP_MAX2 := (DPT2*10)+DPTfrac2;
if ((DPT2*10)+DPTfrac2 < DP_MIN2) DP_MIN2 := (DPT2*10)+DPTfrac2;
if (DP_MAX2 >= 999) DP_MAX2 := 999;
if (DP_MIN2 < 0) DP_MIN2 := 0;
txt_Height(2);
txt_Width(1);
txt_MoveCursor(9,61);
print("DPmax=",DP_MAX2/10,".",DP_MAX2 % 10);
txt_MoveCursor(10,61);
print("DPmin=",DP_MIN2/10,".",DP_MIN2 % 10);
txt_Height(1);txt_Width(1);
endfunc

```

โปรแกรมที่ 1.11 ฟังก์ชันการรับค่าอุณหภูมิ (TEMP) ความชื้น (HUM) และอุณหภูมิจุดน้ำค้าง (DPT) บนหน้าจอ VGA จากอุปกรณ์ตัวรู้ตัวที่ 1

```

func TXMessage()
var ch;
pin_HI(IO1_PIN);
pause(2);

```

โปรแกรมที่ 1.11 ฟังก์ชันการรับค่าอุณหภูมิ (TEMP) ความชื้น (HUM) และอุณหภูมิจุดน้ำค้าง (DPT) บนหน้าจอ VGA จากอุปกรณ์ตัวรู้ตัวที่ 1 (ต่อ)

```

serout1(':');
pause(1);
serout1('1');
pause(1);
serout1(0x0D);
pause(1);
pin_LO(IO1_PIN);
pause(2);
n := 0;
repeat
ch := serin1();
if (ch >= 0)
buff[n] := 0x0000 | ch;
n:=n+1;
endif
until (ch == 0x0D);
endfunc

```

โปรแกรมที่ 1.12 ฟังก์ชันการรับค่าอุณหภูมิ (TEMP) ความชื้น (HUM) และอุณหภูมิจุดน้ำค้าง (DPT) บนหน้าจอ VGA จากอุปกรณ์ตัวรู้ตัวที่ 2

```

func TXMessage2()
var ch2;
pin_HI(IO1_PIN);
pause(2);
//-----config Address :AAC[X]<cr>-----
serout1(':');pause(1);
serout1('1');// A == node Address
pause(1);

```

โปรแกรมที่ 1.12 ฟังก์ชันการรับค่าอุณหภูมิ (TEMP) ความชื้น (HUM) และอุณหภูมิจุดน้ำค้าง (DPT) บนหน้าจอ VGA จากอุปกรณ์ตัวรู้ตัวที่ 2 (ต่อ)

```

serout1('2');// A == node address
pause(1);
serout1('1');// C = code [ 0 = check ,1 = read Temp-Humi , 2 = Read-Write I/O]
pause(1);
serout1(0x0D);// <cr>
pause(2);
pin_LO(IO1_PIN);
pause(2);
//-----
n := 0;
repeat
ch2 := serin1();
if (ch2 >= 0)
buf2[n] := 0x0000 | ch2;
n:=n+1;
endif
until (ch2 == 0x0D);
endfunc

```

โปรแกรมที่ 1.13 ฟังก์ชันการแปลงค่าตัวเลขฐานสองเป็นตัวเลขฐานสิบ (Binary to BCD code)

```

func bin2bcd(var binValue)
return (binValue>>4)*10 + (binValue&15);
endfunc

```

โปรแกรมที่ 1.14 ฟังก์ชันตรวจสอบการอ่านค่าเซนเซอร์จากแผง AP1701

```

func Sensor1() // แผง AP1701 ตำแหน่ง 11H
var TEMP,TEMP_FRAC,HUM,HUM_FRAC,N,Coeff,DPT,DPT_FRAC,A,B,C,D;
if (bin2bcd(ReadRTC(SECONDS)) & 0b0000000000000001 != 0)
TXMessage();

```

โปรแกรมที่ 1.14 ฟังก์ชันตรวจสอบการอ่านค่าเซนเซอร์จากแผง AP1701 (ต่อ)

```

    gfx_CircleFilled(200,176,2,RED);
else
    gfx_CircleFilled(200,176,2,LIGHTGREEN);
endif

TEMP := ((buf[1]-0x30)*100) + ((buf[2]-0x30)*10) + (buf[3]-0x30);
TEMP_FRAC := buf[5]-0x30;
HUM := ((buf[8]-0x30)*10) + (buf[9]-0x30);
HUM_FRAC := buf[11]-0x30;
N := HUM/5;
Coeff := values[N-1];
A := 1120+(9*TEMP); //MAX=2020
C := (Coeff)*(A/10); //MAX=18584
B := (TEMP - 1120)*10;
DPT := (C+B)/100;
DPT_FRAC := ((C+B)%100)/10;
DisplayTEMP(TEMP,TEMP_FRAC);
DisplayHUM(HUM,HUM_FRAC);
DisplayDPT(DPT,DPT_FRAC);
endfunc

func Sensor2() // แผง AP1701 ตำแหน่ง 12H
var TEMP2,TEMP_FRAC2,HUM2,HUM_FRAC2,N2,Coeff2,DPT2,DPT_FRAC2,A2,B2,C2,E;
if (bin2bcd(ReadRTC(SECONDS)) & 0b0000000000000001 != 0)
    TXMessage2();
    gfx_CircleFilled(200,226,2,RED);
else
    gfx_CircleFilled(200,226,2,LIGHTGREEN);
endif

TEMP2 := ((buf2[1]-0x30)*100) + ((buf2[2]-0x30)*10) + (buf2[3]-0x30);
TEMP_FRAC2 := buf2[5]-0x30;

```

โปรแกรมที่ 1.14 ฟังก์ชันตรวจสอบการอ่านค่าเซนเซอร์จากแผง AP1701 (ต่อ)

```

HUM2 := ((buf2[8]-0x30)*10) + (buf2[9]-0x30);
HUM_FRAC2 := buf2[11]-0x30;
N2 := HUM2/5;
Coeff2 := values[N2-1];
A2 := 1120+(9*TEMP2);           //MAX=2020
C2 := (Coeff2)*(A2/10);        //MAX=18584
B2 := (TEMP2 - 1120)*10;
DPT2 := (C2+B2)/100;
DPT_FRAC2 := ((C2+B2)%100)/10;
DisplayTEMP2(TEMP2,TEMP_FRAC2);
DisplayHUM2(HUM2,HUM_FRAC2);
DisplayDPT2(DPT2,DPT_FRAC2);
Endfunc

```

2) เขียนฟังก์ชันหลัก (function main)

โปรแกรมที่ 1.15 เขียนฟังก์ชันหลัก

```

func main()
  pause(2000);
  CreateFrame();
  CreateBuilding();
  CreateTable();
  CreateText();
  com_SetBaud(COM1, 960);
  pin_Set(0,IO1_PIN);
  I2C_Initialize();
  TEMP_MAX := 200;
  TEMP_MIN := 500;
  TEMP_MAX2 := 200;
  TEMP_MIN2 := 500;

```

โปรแกรมที่ 1.15 เขียนฟังก์ชันหลัก (ต่อ)

```
HUM_MAX := 500;
HUM_MIN := 1000;
HUM_MAX2 := 500;
HUM_MIN2 := 1000;
DP_MAX := 100;
DP_MIN := 200;
DP_MAX2 := 100;
DP_MIN2 := 200;

repeat
  Sensor1();
  Sensor2();
  DisplayClock();
  DisplayDate();
  pause(900);
forever

endfunc
```

3.4 ขั้นตอนการสร้างอุปกรณ์ตรวจรู้อุณหภูมิและความชื้น ผ่านแผงควบคุมไมโครวีจีเอ 2

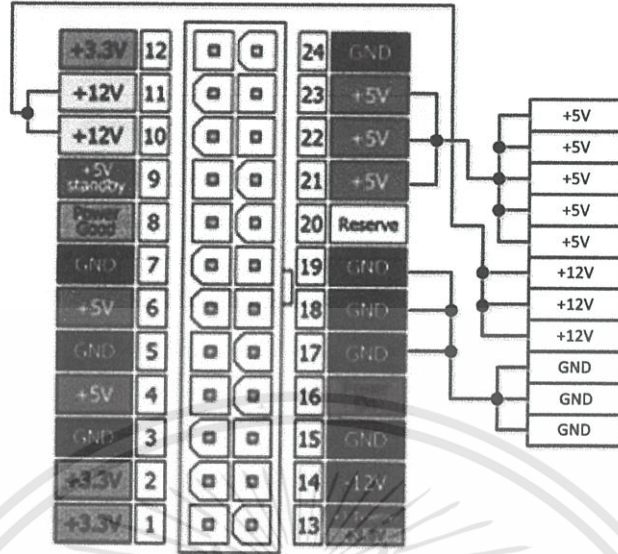
3.4.1 แหล่งจ่ายไฟตรง

แหล่งจ่ายไฟตรงสำหรับจ่ายพลังให้กับอุปกรณ์ตรวจรู้อุณหภูมิและความชื้น ได้ปรับปรุงแหล่งจ่ายไฟตรงของเครื่องคอมพิวเตอร์ ขนาด 450 วัตต์ แบบ ATX (Advance Technology Extended) มีแรงดันไฟตรง 4 ค่า คือ +3.3 โวลต์ +5 โวลต์ \pm 12 โวลต์ ดังแสดงตามรูปที่ 3.4



รูปที่ 3.4 การจัดขาแรงดันไฟตรงของแหล่งจ่ายไฟคอมพิวเตอร์ แบบ ATX

จากนั้นก็ทำการเลือกแรงดันไฟตรงที่ต้องการ จ่ายให้กับแผงไมโครวีจีเอ 2 และแผงตรวจวัดอุณหภูมิและความชื้น (AP1701) สายเซนเซอร์ (Sensor) STH15 ซึ่งทั้งหมดจะใช้แรงดันไฟตรง +5 โวลต์ กับ + 12 โวลต์ แล้วจัดทำคอนเน็กเตอร์เพื่อความสะดวกในการใช้งาน ดังรูปที่ 3.5



รูปที่ 3.5 แสดงการต่อแรงดันไฟตรงเพื่อใช้กับอุปกรณ์ตรวจรู้อุณหภูมิและความชื้น

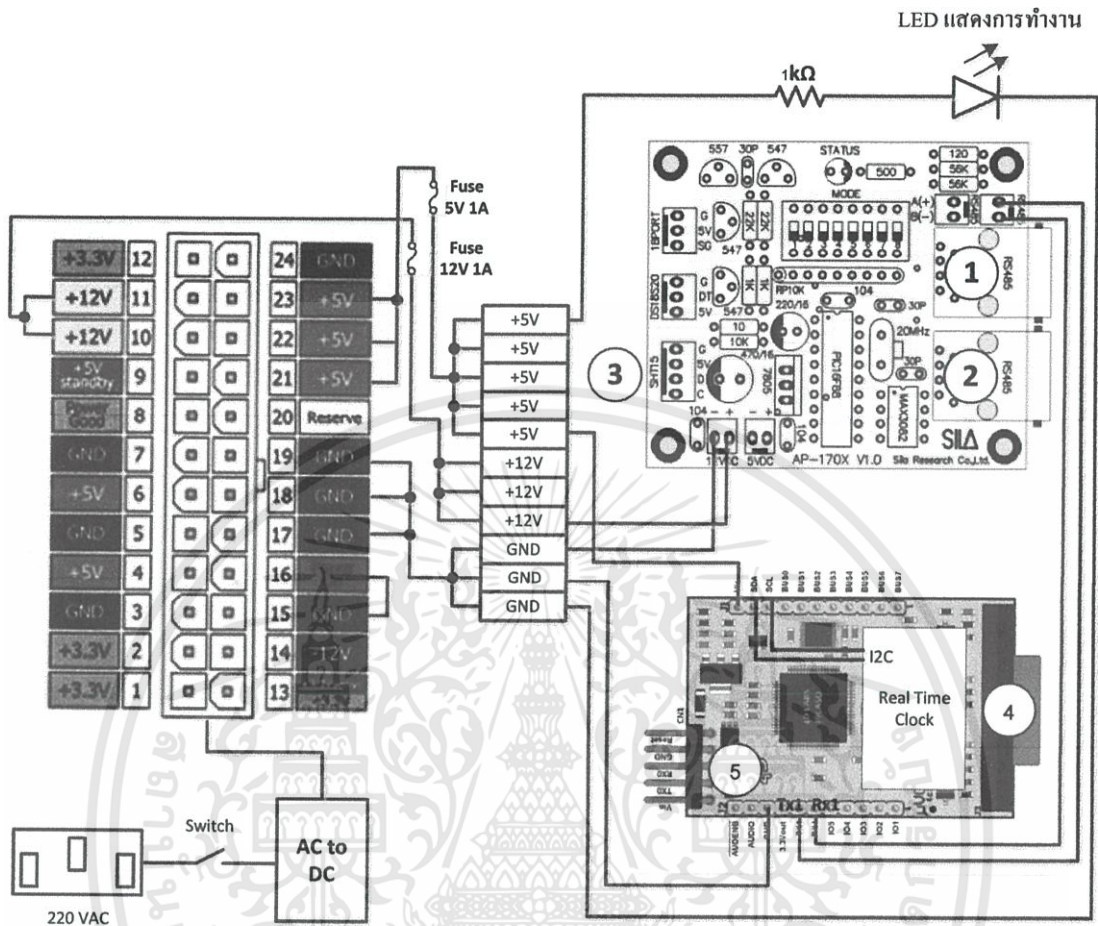
เมื่อต่อแรงดันไฟตรงเสร็จเรียบร้อยแล้ว จะได้แหล่งจ่ายไฟตรงพร้อมใช้งานดังรูปที่ 3.6



รูปที่ 3.6 แหล่งจ่ายไฟตรงพร้อมใช้งาน

3.4.2 การต่อวงจรอุปกรณ์ตรวจรู้อุณหภูมิและความชื้น

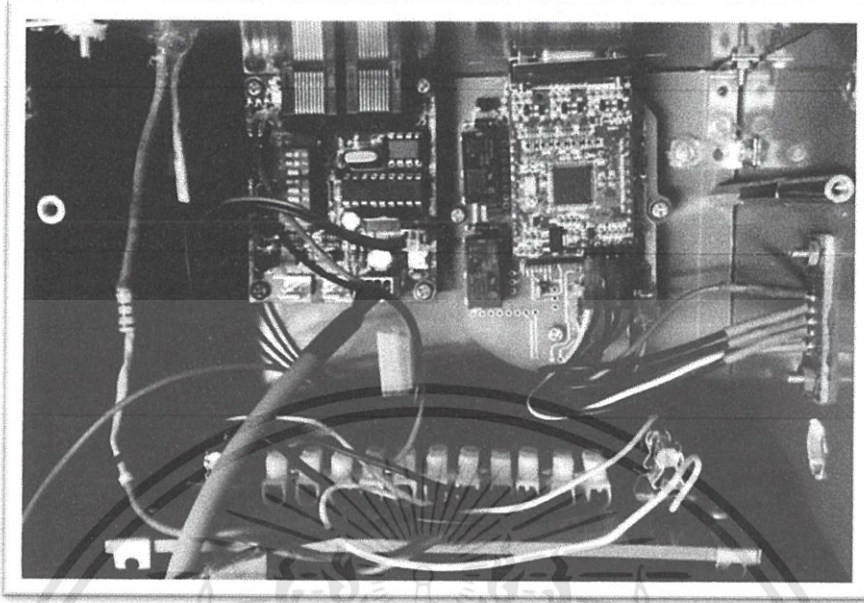
สำหรับขั้นตอนนี้จะเริ่มทำการต่อวงจรจากแหล่งจ่ายไฟตรง ไปแผงควบคุมวีจีเอ และแผงอุปกรณ์ตรวจรู้อุณหภูมิและความชื้น ดังแสดงรายละเอียดการต่อวงจรดังรูปที่ 3.7



รูปที่ 3.7 วงจรการทำงานของอุปกรณ์ตรวจอุณหภูมิและความชื้น

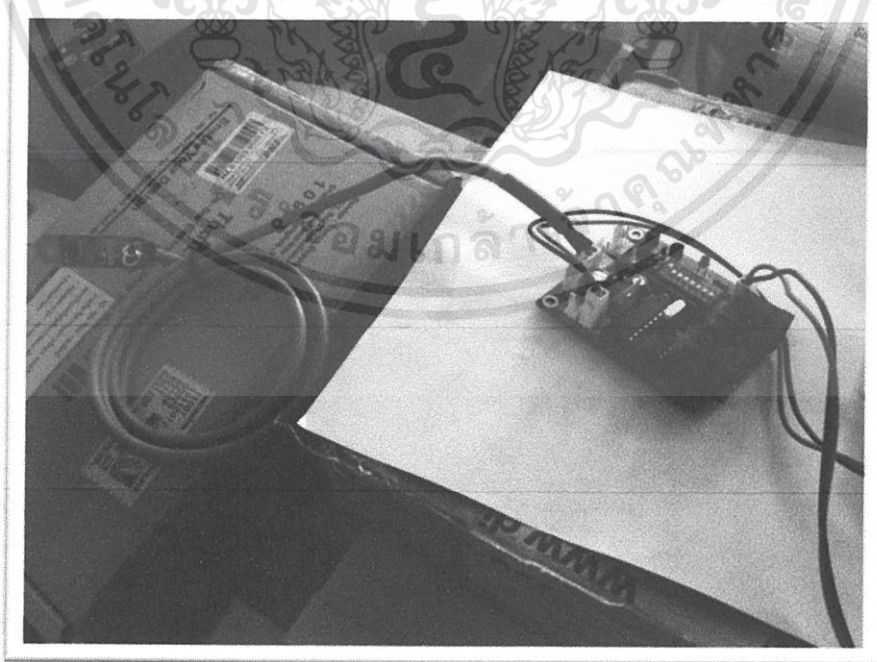
จากรูปที่ 3.7 แสดงวงจรการทำงานของอุปกรณ์ตรวจอุณหภูมิและความชื้น แสดงผลบนจอภาพ VGA มีการต่อวงจรดังนี้ จ่ายแรงดันไฟฟ้าสลับ 220 โวลต์ เข้ากล่องแหล่งจ่ายไฟตรง ซึ่งจะผ่านวงจรแปลงสัญญาณไฟสลับเป็นสัญญาณไฟตรง เลือกใช้เฉพาะแรงดันไฟตรง +12 โวลต์และ +5 โวลต์ โดยที่แรงดันไฟตรง +12 โวลต์จ่ายให้กับแผงตรวจวัดอุณหภูมิและความชื้น AP1701 แรงดันไฟตรง +5 โวลต์ จ่ายให้กับแผงควบคุมไมโครวีจีเอ 2 มีระบบป้องกันกระแสไฟฟ้าเกินและลัดวงจร ด้วยฟิวส์ 12 V 1 A และ 5 V 1 A ใช้หลอดไฟ LED แสดงการทำงานของอุปกรณ์ตรวจวัดอุณหภูมิและความชื้น หมายเลข 1 และ 2 สำหรับขยายตำแหน่งแผงตรวจวัดอุณหภูมิและความชื้น AP1701 หลายตำแหน่ง หมายเลข 3 สำหรับต่อเซ็นเซอร์วัดอุณหภูมิและความชื้น SHT15 หมายเลข 4 ช่องต่อสัญญาณภาพ VGA เพื่อต่อกับจอภาพ LCD หมายเลข 5 ช่องต่อสัญญาณเชื่อมต่อกับคอมพิวเตอร์ กับแผงควบคุมวีจีเอ 2 เพื่อแก้ไขโปรแกรม รูปที่ 3.8 ภาพวงจรภายในอุปกรณ์ตรวจอุณหภูมิและความชื้น ผ่านแผงควบคุมไมโครวีจีเอ 2

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.8 ภาพวงจรภายในอุปกรณ์ตรวจรู้อุณหภูมิและความชื้น ผ่านแผงควบคุมไมโครวีจีเอ 2

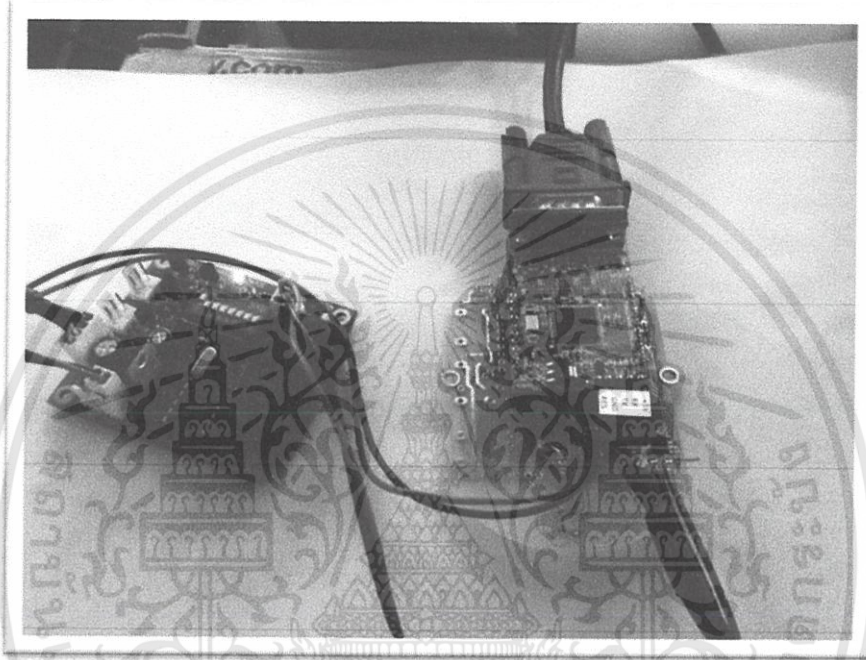
3.5 ทดสอบโปรแกรมการแสดงผลกับอุปกรณ์ตรวจรู้อุณหภูมิและความชื้น ผ่านแผงควบคุมไมโครวีจีเอ 2



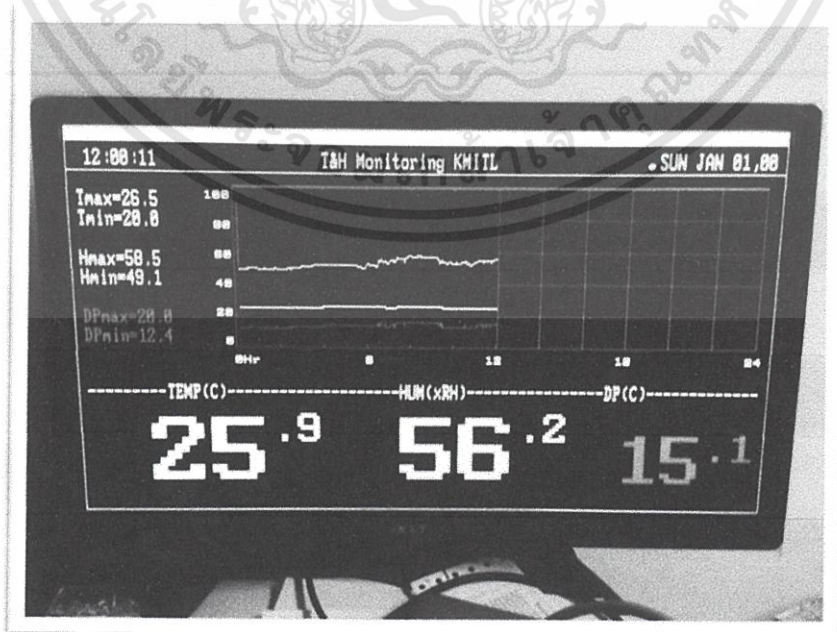
รูปที่ 3.9 การต่อบอร์ดวัดอุณหภูมิและความชื้นแบบ RS458 กับ SHT15

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากรูปที่ 3.9 แสดงการต่อบอร์ดวัดอุณหภูมิและความชื้นแบบ RS458 กับ SHT15 เพียง 1 ชุด ปรับตำแหน่งของ Dip-Switch 8 ตำแหน่งเป็น 11H เพื่อทำการทดสอบการส่งรหัสอ่านค่าอุณหภูมิและความชื้น ผ่านช่องทางการสื่อสารแบบอนุกรม RS458 และส่งข้อมูลอุณหภูมิและความชื้น โดยต่อเข้ากับ TX1 และ RX1 ของแผงควบคุมไมโครวีจีเอ 2 ดังรูปที่ 3.10 และทำการแสดงผลอุณหภูมิและความชื้นเพียงค่าเดียวบนหน้าจอ VGA ดังรูปที่ 3.11



รูปที่ 3.10 การต่อบอร์ดวัดอุณหภูมิและความชื้นแบบ RS458 กับ แผงควบคุมไมโครวีจีเอ 2



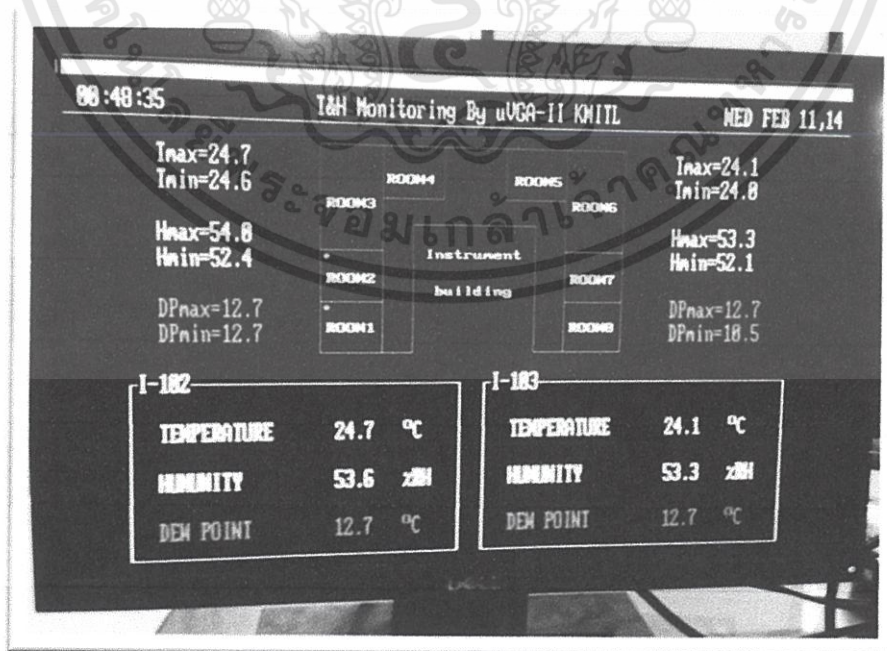
รูปที่ 3.11 การแสดงผลของอุณหภูมิและความชื้นโดยการวัดเพียงตำแหน่งเดียว

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้



รูปที่ 3.12 แสดงการทำงานของอุปกรณ์ตรวจอุณหภูมิและความชื้นผ่านแผงควบคุมไมโครวีจีเอ 2

จากรูปที่ 3.12 เป็นการทดสอบการทำงานของอุปกรณ์ตรวจอุณหภูมิและความชื้น ผ่านแผงควบคุมไมโครวีจีเอ 2 เมื่อเขียนโปรแกรม 4DGL เสร็จเรียบร้อยโดยต่อแผงตรวจวัดอุณหภูมิและความชื้น AP1701 จำนวน 2 แผง ตั้งค่าตำแหน่งของแผงเป็น 11H และ 12H ตามลำดับ และต่อเซนเซอร์ SHT15 จำนวน 2 ตัว ซึ่งภาพที่แสดงหน้าจอ VGA จะแสดงอุณหภูมิ ความชื้น อุณหภูมิจุดน้ำค้างของห้อง I102 และ I103 พร้อมแสดงค่าสูงสุดและต่ำสุด มีแผงผังอาคารคณะวิศวกรรมศาสตร์ ให้สามารถตรวจสอบว่าวัดที่พื้นที่ห้องใด และมีการแสดงวันเวลาในการวัดด้วย ดังรูปที่ 3.13



รูปที่ 3.13 การแสดงผลของอุณหภูมิและความชื้นผ่านจอ VGA โดยใช้ μ VGA II

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 4

ผลการทดลอง

การศึกษาการวัดอุณหภูมิและความชื้นแสดงผลผ่านจอ VGA โดยใช้แผงควบคุมไมโครวีจีเอ สามารถวัดอุณหภูมิและความชื้นได้มากถึง 31 ตำแหน่ง ด้วยแผงวัดอุณหภูมิและความชื้น AP1701 โดยใช้การสื่อสารข้อมูลอนุกรมเพียง 2 เส้น (RS485) สำหรับการศึกษาครั้งนี้ได้ทำการทดลองวัดอุณหภูมิและความชื้นเพียง 2 ตำแหน่งเท่านั้น โดยทดลองวัดอุณหภูมิและความชื้นของอุปกรณ์ตรวจรู้ SHT 15 ที่ตำแหน่งเดียวกันกับอุปกรณ์มาตรฐาน ภายในห้องที่ใช้เครื่องปรับอากาศเป็นอุปกรณ์ควบคุมอุณหภูมิ

ตารางที่ 4.1 ผลการวัดของอุปกรณ์ตรวจรู้ SHT 15 ตัวที่ 1 ณ ที่อุณหภูมิต่างๆเทียบกับอุปกรณ์มาตรฐาน

| ค่าที่กำหนดโดยอุปกรณ์มาตรฐาน | | | ค่าที่ได้จากอุปกรณ์ตรวจรู้ (SHT15) ตัวที่ 1 | | | เปอร์เซ็นต์ความผิดพลาด จากอุปกรณ์ตรวจรู้ (SHT15)ตัวที่ 1 เทียบกับอุปกรณ์มาตรฐาน | | |
|------------------------------|--------------|-----------------------|---------------------------------------------|--------------|-----------------------|---------------------------------------------------------------------------------|--------------|-----------------------|
| อุณหภูมิ (°C) | ความชื้น (%) | อุณหภูมิ น้ำค้าง (°C) | อุณหภูมิ (°C) | ความชื้น (%) | อุณหภูมิ น้ำค้าง (°C) | อุณหภูมิ (°C) | ความชื้น (%) | อุณหภูมิ น้ำค้าง (°C) |
| 16.0 | 45.2 | 6.2 | 16.3 | 46.0 | 6.4 | 1.88 | 1.78 | 3.23 |
| 18.0 | 45.7 | 7.1 | 18.2 | 46.3 | 7.8 | 1.11 | 1.31 | 9.86 |
| 20.0 | 46.3 | 9.0 | 20.2 | 46.7 | 9.6 | 1.00 | 0.86 | 6.67 |
| 22.0 | 46.5 | 11.0 | 22.3 | 47.2 | 11.2 | 1.36 | 1.51 | 1.81 |
| 24.0 | 48.0 | 13.9 | 24.1 | 47.8 | 13.3 | 0.42 | 0.42 | 4.32 |
| 26.0 | 49.2 | 14.8 | 26.2 | 48.5 | 15.1 | 0.77 | 1.42 | 2.03 |
| 28.0 | 51.4 | 16.9 | 28.2 | 49.8 | 17.1 | 0.71 | 3.11 | 1.18 |
| 30.0 | 52.3 | 18.4 | 30.0 | 51.6 | 19.2 | 0.00 | 1.34 | 4.35 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

ตารางที่ 4.2 เปอร์เซนต์ความผิดพลาดเฉลี่ยของอุปกรณ์ตรวจรู้ SHT 15 ตัวที่ 1เทียบกับอุปกรณ์มาตรฐาน

| เปอร์เซนต์ความผิดพลาดเฉลี่ยของอุปกรณ์ตรวจรู้ SHT 15 ตัวที่ 1เทียบกับอุปกรณ์มาตรฐาน | | | | | | | | | |
|------------------------------------------------------------------------------------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|
| ปัจจัยด้าน | ค่าที่ 1 | ค่าที่ 2 | ค่าที่ 3 | ค่าที่ 4 | ค่าที่ 5 | ค่าที่ 6 | ค่าที่ 7 | ค่าที่ 8 | ค่าเฉลี่ย |
| อุณหภูมิ | 1.88 | 1.11 | 1.00 | 1.36 | 0.42 | 0.77 | 0.71 | 0.00 | 0.91 |
| ความชื้น | 1.78 | 1.31 | 0.86 | 1.51 | 0.42 | 1.42 | 3.11 | 1.34 | 1.47 |
| อุณหภูมิ น้ำค้าง | 3.23 | 9.86 | 6.67 | 1.81 | 4.32 | 2.03 | 1.18 | 4.35 | 4.18 |

จากตารางที่ 4.2 คำนวณหาเปอร์เซนต์ความผิดพลาดของอุปกรณ์ตรวจรู้ SHT 15 ตัวที่ 1เทียบกับอุปกรณ์มาตรฐาน (อุณหภูมิ, ความชื้น และ อุณหภูมิน้ำค้าง) หาได้จากสมการที่ 4.1 และคำนวณหาเปอร์เซนต์ความผิดพลาดเฉลี่ยของอุปกรณ์ตรวจรู้ SHT 15 ตัวที่ 1 เทียบกับอุปกรณ์มาตรฐาน จำนวน 8 ค่าได้จากสมการที่ 4.2

$$\text{เปอร์เซนต์ความผิดพลาด} = \frac{\text{ค่าที่อ่านได้} - \text{ค่าอุปกรณ์มาตรฐาน}}{\text{ค่าอุปกรณ์มาตรฐาน}} \times 100 \quad (4.1)$$

$$\text{เปอร์เซนต์ความผิดพลาดเฉลี่ยของอุปกรณ์ตรวจรู้} = \frac{\text{เปอร์เซนต์ความผิดพลาดค่าที่ 1} + \text{ค่าที่ 2} + \text{ค่าที่ 3} \dots + \text{ค่าที่ 8}}{8} \quad (4.2)$$

ตารางที่ 4.3 ผลการวัดของอุปกรณ์ตรวจรู้ SHT 15 ตัวที่ 2 ณ ที่อุณหภูมิต่างๆเทียบกับอุปกรณ์มาตรฐาน

| ค่าที่กำหนดโดยอุปกรณ์มาตรฐาน | | | ค่าที่ได้จากอุปกรณ์ตรวจรู้ (SHT15) ตัวที่ 2 | | | เปอร์เซ็นต์ความผิดพลาด จากอุปกรณ์ตรวจรู้ (SHT15)ตัวที่ 2 เทียบกับอุปกรณ์มาตรฐาน | | |
|------------------------------|--------------|-----------------------|---------------------------------------------|--------------|-----------------------|---------------------------------------------------------------------------------|--------------|-----------------------|
| อุณหภูมิ (°C) | ความชื้น (%) | อุณหภูมิ น้ำค้าง (°C) | อุณหภูมิ (°C) | ความชื้น (%) | อุณหภูมิ น้ำค้าง (°C) | อุณหภูมิ (°C) | ความชื้น (%) | อุณหภูมิ น้ำค้าง (°C) |
| 16.0 | 45.2 | 6.2 | 16.4 | 46.0 | 6.3 | 2.55 | 1.77 | 1.61 |
| 18.0 | 45.7 | 7.1 | 18.3 | 46.1 | 7.8 | 1.67 | 0.87 | 9.86 |
| 20.0 | 46.3 | 9.0 | 20.4 | 46.5 | 9.2 | 2.00 | 0.43 | 2.22 |
| 22.0 | 46.5 | 11.0 | 22.5 | 47.3 | 11.5 | 2.27 | 1.72 | 4.54 |
| 24.0 | 48.0 | 13.9 | 24.2 | 47.7 | 13.2 | 0.83 | 0.63 | 5.04 |
| 26.0 | 49.2 | 14.8 | 26.5 | 48.6 | 15.1 | 1.92 | 1.21 | 6.08 |
| 28.0 | 51.4 | 16.9 | 28.7 | 49.2 | 17.2 | 2.50 | 4.28 | 1.78 |
| 30.0 | 52.3 | 18.4 | 30.7 | 50.9 | 20.2 | 2.33 | 2.68 | 9.78 |

ตารางที่ 4.4 เปอร์เซนต์ความผิดพลาดเฉลี่ยของอุปกรณ์ตรวจรู้ SHT 15 ตัวที่ 2 เทียบกับอุปกรณ์มาตรฐาน

| เปอร์เซนต์ความผิดพลาดเฉลี่ยของอุปกรณ์ตรวจรู้ SHT 15 ตัวที่ 2 เทียบกับอุปกรณ์มาตรฐาน | | | | | | | | | |
|-------------------------------------------------------------------------------------|----------|----------|----------|----------|----------|----------|----------|----------|-----------|
| ปัจจัยด้าน | ค่าที่ 1 | ค่าที่ 2 | ค่าที่ 3 | ค่าที่ 4 | ค่าที่ 5 | ค่าที่ 6 | ค่าที่ 7 | ค่าที่ 8 | ค่าเฉลี่ย |
| อุณหภูมิ | 2.55 | 1.67 | 2.00 | 2.27 | 0.83 | 1.92 | 2.50 | 2.33 | 2.00 |
| ความชื้น | 1.77 | 0.87 | 0.43 | 1.72 | 0.63 | 1.21 | 4.28 | 2.68 | 1.69 |
| อุณหภูมิ น้ำค้าง | 1.61 | 9.86 | 2.22 | 4.54 | 5.04 | 6.08 | 1.78 | 9.78 | 5.11 |

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

จากตารางที่ 4.4 คำนวณหาเปอร์เซ็นต์ความผิดพลาดของอุปกรณ์ตรวจรู้ SHT 15 ตัวที่ 2 เทียบกับ อุปกรณ์มาตรฐาน (อุณหภูมิ, ความชื้น และ อุณหภูมิน้ำค้าง) หาได้จากสมการที่ 4.1 และคำนวณหา เปอร์เซ็นต์ความผิดพลาดเฉลี่ยของอุปกรณ์ตรวจรู้ SHT 15 ตัวที่ 2 เทียบกับอุปกรณ์มาตรฐาน จำนวน 8 ค่า ได้จากสมการที่ 4.2



เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดๆทั้งสิ้น อีกทั้งห้ามมิให้ดัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

บทที่ 5

สรุปผลการทดลองและข้อเสนอแนะ

ปริญญานิพนธ์ได้นำเสนอการออกแบบ และการสร้างเครื่องตรวจรู้อุณหภูมิและความชื้น ที่สามารถแสดงบนจอ VGA โดยใช้แผงควบคุมไมโครวีจีเอ โดยใช้จอคอมพิวเตอร์ชนิดแอลซีดี (LCD) โดยใช้โปรแกรม 4DGL ในการจัดการภาพบนหน้าจอแอลซีดี และติดต่อสื่อสารข้อมูลกับอุปกรณ์ตรวจรู้ เพื่อให้ได้ข้อมูลนำมาแสดงผลบนหน้าจอที่มีขนาดใหญ่สามารถมองเห็นได้ชัดเจนโดยได้ทำการทดสอบค่าที่วัดได้จากอุปกรณ์ตรวจรู้ อุณหภูมิและความชื้น จำนวน 2 ตำแหน่ง ภายในห้องที่มีการควบคุมอุณหภูมิ

5.1 สรุปผลการทดลอง

การแสดงค่าของอุณหภูมิและความชื้นบนหน้าจอ VGA โดยใช้แผงควบคุมไมโครวีจีเอ 2 ได้อย่างชัดเจน จากการทดลองตารางที่ 4.2 และตารางที่ 4.4 ค่าเปอร์เซ็นต์ความผิดพลาดเฉลี่ยของอุปกรณ์ตรวจรู้ SHT 15 ทั้งสองตำแหน่งมีค่าดังนี้

5.1.1 เปอร์เซนต์ความผิดพลาดเฉลี่ยของอุปกรณ์ตรวจรู้ SHT 15 ตัวที่ 1 เทียบกับอุปกรณ์มาตรฐาน

ด้านอุณหภูมิ มีค่าเท่ากับ 0.91 %

ด้านความชื้น มีค่าเท่ากับ 1.47 %

ด้านอุณหภูมิน้ำค้าง มีค่าเท่ากับ 4.18 %

5.1.2 เปอร์เซนต์ความผิดพลาดเฉลี่ยของอุปกรณ์ตรวจรู้ SHT 15 ตัวที่ 2 เทียบกับอุปกรณ์มาตรฐาน

ด้านอุณหภูมิ มีค่าเท่ากับ 2.00 %

ด้านความชื้น มีค่าเท่ากับ 1.69 %

ด้านอุณหภูมิน้ำค้าง มีค่าเท่ากับ 5.11 %

5.2 ปัญหาที่พบในการทดลอง

ในการทดลองที่ทำการวัดเปรียบเทียบภายในห้องที่ควบคุมอุณหภูมิ ซึ่งตั้งค่าอุณหภูมิอยู่ในช่วง 16-30 องศาเซนเซียล เปอร์เซนต์ค่าผิดพลาดที่คำนวณได้ไม่เป็นเชิงเส้น จึงจำเป็นต้องหาค่าเฉลี่ยของเปอร์เซนต์ค่าผิดพลาด

เอกสารนี้เป็นเอกสารที่สงวนไว้สำหรับการใช้งานเพื่อการศึกษาเท่านั้น ไม่อนุญาตให้นำไปใช้ประโยชน์ด้านการค้า ไม่ว่าจะกรณีใดทั้งสิ้น อีกทั้งห้ามมิให้ตัดแปลงเนื้อหา และต้องอ้างอิงถึงเจ้าของเอกสารทุกครั้งที่มีการนำไปใช้

5.3 ข้อเสนอแนะและแนวทางการพัฒนา

5.3.1 ความสามารถในการแสดงผลกราฟิกของแผงควบคุมไมโครวีจีเอ โดยโปรแกรม 4DGL ยังมีอีกมากมาย เช่นการทำสไลด์บาร์ ปุ่มกดแบบกราฟิก เป็นต้น หากผู้ที่สนใจจะนำไปพัฒนา ควรศึกษาคำสั่งภายในของ PICASO GFX (Internal Command PICASO GFX) เพิ่มเติม

5.3.2 เมื่อทำการทดลองเปรียบเทียบการวัดอุณหภูมิห้องกับอุปกรณ์ตรวจรู้มากกว่า 2 ตัวผ่านสาย RJ45 ควรใช้ไฟตรง 12 โวลท์



บรรณานุกรม

- [1] กฤษดา ใจเย็น และชัยวัฒน์ ลิ้มพรจิตรวิไล. เขียนโปรแกรมภาษา C สำหรับไมโครคอนโทรลเลอร์ PIC ด้วย mikro C คอมไพเลอร์และการทดลองเบื้องต้น บริษัทอินโนเวทีฟ เอ็กเพอริเมนต์ จำกัด, กรุงเทพฯ, 2537
- [2] มงคล เดชนครินทร์. พจนานุกรมศัพท์วิศวกรรมไฟฟ้า ฉบับ ว.ส.ท., บริษัท โกลบอล กราฟฟิค จำกัด, กรุงเทพฯ, 2550
- [3] พูนศักดิ์ ฐนพันธ์พานิช. คู่มือการใช้งาน After Effects, บริษัท เอส.พี.ซี. บุ๊คส์ จำกัด, กรุงเทพฯ, 2551
- [4] 4D System Turning Technology into Art. 2013. **4DGL Programmers Reference Manual. (Online). Available :** <http://www.4dsystems.com.au/downloads/Software/4D-Workshop4-IDE/Docs/Designer/4DGL-Programmers-Reference-Manual-REV5.3.pdf>
- [5] 4D System Turning Technology into Art. 2013. **PICASO-4DGL-Internal-Functions. (Online). Available :** <http://www.4dsystems.com.au/downloads/Software/4D-Workshop4-IDE/Docs/Designer/PICASO-4DGL-Internal-Functions-REV6.17.pdf>
- [6] 4D System Turning Technology into Art. 2013. **Workshop-4-User-Guide. (Online). Available :** <http://www.4dsystems.com.au/downloads/Software/4D-Workshop4-IDE/Docs/Workshop-4-User-Guide-REV1.3.pdf>
- [7] 4D System Turning Technology into Art. 2013. **uVGA-II(GFX). (Online). Available :** [http://old.4dsystems.com.au/downloads/micro-VGA/uVGA-II\(GFX\)/Docs/uVGA-II\(GFX\)-DS-rev3.pdf](http://old.4dsystems.com.au/downloads/micro-VGA/uVGA-II(GFX)/Docs/uVGA-II(GFX)-DS-rev3.pdf)
- [8] 4D System Turning Technology into Art. 2013. **PICASO-Processor-Datasheet. (Online). Available :** <http://www.4dsystems.com.au/downloads/Semiconductors/PICASO/Docs/PICASO-Processor-Datasheet-REV1.1.pdf>